

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 22

2016

№ 7

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

СОДЕРЖАНИЕ

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Федосов В. В., Федосов А. В. О параметризации сетки в алгоритмах полубесконечной оптимизации для задач промышленной экологии 483
- Зеленко Г. В., Иванников А. Д., Коган Д. Ю., Рощин А. В. Проблемы функциональной устойчивости сенсорной локальной сети 489

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

- Холопов Ю. А., Ле Ба Чунг, Нгуен Тхань Чунг, Чан Ван Хань. Особенности реализации распределенных систем управления на основе аппаратных сетевых решений 494
- Романов А. Ю. Исследование сетей на кристалле с топологией *mesh* с помощью модели *NoCTweak* 498

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

- Кухаренко Б. Г., Солнцева-Чалей М. О. Использование методов выделения ярких областей при сегментировании телеметрических изображений для идентификации групп объектов 504
- Гиацинтов А. М., Мамросенко К. А. Методы отображения трехмерных объектов при применении отложенной визуализации 510

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Стемпковский А. Л., Тельпухов Д. В., Соловьев Р. А. Повышение сбоеустойчивости логических схем на основе частичного ресинтеза схемы 515

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Асратян Р. Э., Лебедев В. Н., Орлов В. Л. Эффективность защищенных каналов взаимодействия на основе применения протокола HTTPS в прокси-серверах . . 523
- Грибова В. В., Иванова А. В. Концепция программного комплекса для управления безопасностью информационных систем 527

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

- Подбельский В. В., Максименкова О. В., Бабич К. С. Об обеспечении интероперабельности программных средств поддержки тестирования в образовании . . 535

ДИСКУССИОННЫЙ КЛУБ

- Вяткин В. Б. Интегративно-кодированная взаимосвязь комбинаторного, вероятностного и синергетического подходов к определению количества информации . 542

Журнал в журнале

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- Галушкин А. И. Новые технологии микроэлектроники и разработки перспективных нейрокомпьютеров 550
- Дам В. Н. Автоматическое распознавание цифровой модуляции радиосигналов с помощью многослойной нейронной сети по кумулянтным признакам . . . 555

Главный редактор:

СТЕМПКОВСКИЙ А. Л.,
акад. РАН, д. т. н., проф.

Зам. главного редактора:

ИВАННИКОВ А. Д., д. т. н., проф.
ФИЛИМОНОВ Н. Б., д. т. н., с.н.с.

Редакционный совет:

БЫЧКОВ И. В., акад. РАН, д. т. н.
ЖУРАВЛЕВ Ю. И.,
акад. РАН, д. ф.-м. н., проф.

КУЛЕШОВ А. П.,
акад. РАН, д. т. н., проф.

ПОПКОВ Ю. С.,
чл.-корр. РАН, д. т. н., проф.

РУСАКОВ С. Г.,
чл.-корр. РАН, д. т. н., проф.

РЯБОВ Г. Г.,
чл.-корр. РАН, д. т. н., проф.

СОЙФЕР В. А.,
чл.-корр. РАН, д. т. н., проф.

СОКОЛОВ И. А., акад.
РАН, д. т. н., проф.

СУЕТИН Н. В., д. ф.-м. н., проф.

ЧАПЛЫГИН Ю. А.,
чл.-корр. РАН, д. т. н., проф.

ШАХНОВ В. А.,
чл.-корр. РАН, д. т. н., проф.

ШОКИН Ю. И.,
акад. РАН, д. т. н., проф.

ЮСУПОВ Р. М.,
чл.-корр. РАН, д. т. н., проф.

Редакционная коллегия:

АВДОШИН С. М., к. т. н., доц.
АНТОНОВ Б. И.

БАРСКИЙ А. Б., д. т. н., проф.

ВАСЕНИН В. А., д. ф.-м. н., проф.

ВИШНЕКОВ А. В., д. т. н., проф.

ГАЛУШКИН А. И., д. т. н., проф.

ДИМИТРИЕНКО Ю. И., д. ф.-м. н., проф.

ДОМРАЧЕВ В. Г., д. т. н., проф.

ЗАБИДУЛЛИН Р. Ш., к. т. н., доц.

ЗАРУБИН В. С., д. т. н., проф.

КАРПЕНКО А. П., д. ф.-м. н., проф.

КОЛИН К. К., д. т. н., проф.

КУЛАГИН В. П., д. т. н., проф.

КУРЕЙЧИК В. М., д. т. н., проф.

ЛЬВОВИЧ Я. Е., д. т. н., проф.

МИХАЙЛОВ Б. М., д. т. н., проф.

НЕЧАЕВ В. В., к. т. н., проф.

ПОЛЕШУК О. М., д. т. н., проф.

СОКОЛОВ Б. В., д. т. н., проф.

ТИМОНИНА Е. Е., д. т. н., проф.

УСКОВ В. Л., к. т. н. (США)

ФОМИЧЕВ В. А., д. т. н., проф.
ШИЛОВ В. В., к. т. н., доц.

Редакция:

БЕЗМЕНОВА М. Ю.

ГРИГОРИН-РЯБОВА Е. В.

ЛЫСЕНКО А. В.

ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.

Журнал включен в систему Российского индекса научного цитирования.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

CONTENTS**MODELING AND OPTIMIZATION**

Fedosov V. V., Fedosova A. V. Parameterization of the Mesh Algorithms Semi-Infinite Optimization for the Problems of Industrial Ecology 483

Zelenko G. V., Ivannikov A. D., Kogan D. Yu., Roshchin A. V. Problems of Functional Stability in Sensor Network 489

COMPUTING SYSTEMS AND NETWORKS

Holopov Ju. A., Le Ba Chung, Nguyen Thanh Trung, Tran Van Khanh. Features of Implementation of Distributed Control Systems Based on Hardware Network Solutions 494

Romanov A. Yu. The Research of Network-on-Chip with Mesh Topology by Using *NoCTweak Model* 498

DIGITAL PROCESSING OF SIGNALS AND IMAGES

Kukhareenko B. G., Solntseva-Chalei M. O. Use of Salient Region Detecting Methods after Telemetric Image Segmentation for Object Group Identification 504

Giatsintov A. M., Mamrosenko K. A. Methods of 3D Objects Visualization with Deferred Rendering 510

CAD-SYSTEMS

Stempkovskiy A. L., Telpukhov D. V., Solovyev R. A. Enhancing Reliability of Logic Circuits with Partial Resynthesis Method 515

CRYPTOSAFETY INFORMATION

Asratian R. E., Lebedev V. N., Orlov V. L. Efficiency of a Secure Communication Channels Based on the Use of the HTTPS in Proxy Servers 523

Gribova V. V., Ivanova A. V. Software for Security Control of Information Systems 527

INFORMATION TECHNOLOGIES IN EDUCATION

Podbelskiy V. V., Maksimenkova O. V., Babich K. S. On Educational Testing Software Interoperability Support 535

DISCUSSION CLUB

Vyatkin V. B. Integrative-Code Interrelation of Combinatorial, Probabilistic and Synergetic Approaches to Determining the Amount of Information 542

**Journal-in-journal
NEUROTECHNOLOGIES**

Galushkin A. I. New Microelectronics Technology and the Development of Advanced Neurocomputers 550

Dam Van Nhich. Automatic Digital Modulation Recognition of Radio Signals Using Multilayer Neural Network with Cumulant Features 555

Editor-in-Chief:

Stempkovskiy A. L., Member of RAS,
Dr. Sci. (Tech.), Prof.

Deputy Editor-in-Chief:

Ivannikov A. D., Dr. Sci. (Tech.), Prof.
Filimonov N. B., Dr. Sci. (Tech.), Prof.

Chairman:

Bychkov I. V., Member of RAS,
Dr. Sci. (Tech.), Prof.
Zhuravljov Yu. I., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Kuleshov A. P., Member of RAS,
Dr. Sci. (Tech.), Prof.
Popkov Yu. S., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Rusakov S. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Ryabov G. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Soifer V. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Sokolov I. A., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Suetin N. V.,
Dr. Sci. (Phys.-Math.), Prof.
Chaplygin Yu. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Shakhnov V. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Shokin Yu. I., Member of RAS,
Dr. Sci. (Tech.), Prof.
Yusupov R. M., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.

Editorial Board Members:

Avdoshin S. M., Cand. Sci. (Tech.), Ass. Prof.
Antonov B. I.
Barsky A. B., Dr. Sci. (Tech.), Prof.
Vasenin V. A., Dr. Sci. (Phys.-Math.), Prof.
Vishnekov A. V., Dr. Sci. (Tech.), Prof.
Galushkin A. I., Dr. Sci. (Tech.), Prof.
Dimitrienko Yu. I., Dr. Sci. (Phys.-Math.), Prof.
Domrachev V. G., Dr. Sci. (Tech.), Prof.
Zaborovsky V. S., Dr. Sci. (Tech.), Prof.
Zagidullin R. Sh., Cand. Sci. (Tech.), Ass. Prof.
Zarubin V. S., Dr. Sci. (Tech.), Prof.
Karpenko A. P., Dr. Sci. (Phys.-Math.), Prof.
Kolin K. K., Dr. Sci. (Tech.)
Kulagin V. P., Dr. Sci. (Tech.), Prof.
Kurechik V. M., Dr. Sci. (Tech.), Prof.
Ljvovich Ya. E., Dr. Sci. (Tech.), Prof.
Mikhailov B. M., Dr. Sci. (Tech.), Prof.
Nechaev V. V., Cand. Sci. (Tech.), Ass. Prof.
Poleschuk O. M., Dr. Sci. (Tech.), Prof.
Sokolov B. V., Dr. Sci. (Tech.)
Timonina E. E., Dr. Sci. (Tech.), Prof.
Uskov V. L. (USA), Dr. Sci. (Tech.)
Fomichev V. A., Dr. Sci. (Tech.), Prof.
Shiulov V. V., Cand. Sci. (Tech.), Ass. Prof.

Editors:

Bezmenova M. Yu.
Grigorin-Ryabova E. V.
Lysenko A. V.
Chugunova A. V.

Complete Internet version of the journal at site: <http://novtex.ru/IT>.

According to the decision of the Higher Certifying Commission of the Ministry of Education of Russian Federation, the journal is inscribed in "The List of the Leading Scientific Journals and Editions wherein Main Scientific Results of Theses for Doctor's or Candidate's Degrees Should Be Published"

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ MODELING AND OPTIMIZATION

УДК 519.683, 519.856

В. В. Федосов, канд. техн. наук, доц., e-mail: vlr.fdsfv@gmail.com, Москва,
А. В. Федосова, канд. физ.-мат. наук, доц., e-mail: afedosova@unal.edu.co,
Национальный университет, Богота, Колумбия (Universidad Nacional de Colombia)

О параметризации сетки в алгоритмах полубесконечной оптимизации для задач промышленной экологии

На примере задачи оптимального управления загрязнением территории выбросами группы источников в составе алгоритма полубесконечной оптимизации выполнен численный эксперимент с параметризацией по дискретности сетки. Установлено значительное и устойчивое влияние дискретности на аппроксимации результатов. Выбор дискретности области должен достаточно качественно проявить (описать) характерные особенности общей функции загрязнения, но остается эвристической процедурой.

Ключевые слова: дискретность сетки, полубесконечная оптимизация, аппроксимация, нелинейное программирование, источники выбросов, нормативы загрязняющей нагрузки

Введение

Во многих областях (техника, материаловедение, промышленная экология, финансы) существуют задачи оптимизации, содержащие параметры пространства или времени, которые в силу неопределенной дискретизации считают бесконечными, а задачи с их участием — полубесконечными (*SIP*, *Semi-Infinite Programming*).

SIP-задачи формулируются как задачи математического программирования в составе целевой функции (1) и системы ограничений (2):

$$f(x) \rightarrow \min_{x \in X^0}; \quad (1)$$

$$g(x, y) \leq 0, \quad \forall y \in Y^0, \quad (2)$$

где $f(x)$ и $g(x, y)$ предполагаются непрерывно дифференцируемыми на $X^0 \times Y^0$ — выпуклых компактах; $X^0 \subset \mathfrak{R}^k$ and $Y^0 \subset \mathfrak{R}^l$, k и l — размерности пространства.

Бесконечность точек пространства u означает бесконечность множества ограничений $g(x, y) \leq 0$, связанных с физической сущностью задачи, где вектор x является управляющим воздействием и может применяться в системах с обратной связью.

Теория SIP рассматривает и находит выход из проблемы бесконечности. Метод аппроксимации сводит исходную задачу P^0 с бесконечным числом ограничений к решению последовательности аппроксимирующих задач $P(Y_i)$, $i = 1, 2, \dots$, в которых множества Y_i , задающие систему критических ограничений, являются конечными $|Y_i| < \infty$. Критические ограничения находят максимизацией функ-

ции ограничения от старта, соответствующего точке очередного нарушения ограничения. Для поиска нарушений применяется стохастическая процедура.

Аппроксимирующие задачи $P(Y_i)$ — это уже обычные задачи математического программирования с конечными числами переменных и ограничений.

Алгоритм SIP организует итерации, на которых определяется вектор аппроксимации x , действующий на следующей итерации, поэтому SIP считают "методом внешних аппроксимаций". Его обоснование приведено в работе [1], а рабочие алгоритмы — в работе [2].

Однако при переходе к практике SIP необходимо оперировать точными координатами точек и размерами выделенной области, поэтому необходим выбор размерности сетки, так как исходная сетка по умолчанию бесконечна. Число точек при объявленной дискретности несопоставимо мало в сравнении с бесконечной, но влияет на объемы счетных массивов данных, определяет время прохождения алгоритма и ограничено мощностью вычислительных средств.

В работах [3, 4] отмечены трудности аппроксимации моделей в экстремальной постановке, связанные с вынужденными приближенными описаниями, и сделан вывод, что задачи параметрической оптимизации обычно удовлетворительны по точности решения лишь при достаточной размерности сеток.

Объявление "достаточной" дискретности сетки не определено. Дискретность меняет приближения (а следовательно, и интегральные характеристики

функций), что предполагает приближения результатов SIP и отражается на их графическом представлении.

Для выяснения роли конечной дискретности сетки в сходимости оптимального решения алгоритмов SIP и качестве вывода графики необходим сравнительный численный эксперимент с сохранением всех существенных процедур алгоритма, что составило цель работы. Диапазоны дискретности выбраны близкими к разрешениям экранов мониторов, у которых 640×480 — считается низким разрешением, а 1600×1200 — уже высоким.

Взята одна из формулировок задачи промышленной экологии, связанной с требованием соблюдения экологических нормативов загрязнения ограниченной территории выбросами группы источников.

Программирование и графика выполнены в *MATLAB*.

Постановка эксперимента

На условной двумерной области \mathbf{d} размещена группа из N источников промышленных выбросов, координаты размещения которых \mathbf{tN} и мощности \mathbf{H} известны.

Если отсутствуют надежные данные мониторинга загрязнений от выбросов источников, их описывают функционально, считая такую адекватность приемлемой [5, 6]. Наиболее простыми представляются параболические функции, немногочисленные параметры которых имеют ясную трактовку для данного случая, хотя допустимы иные, связанные, например, с функциями Гаусса [6, 7].

Каждый источник генерирует пятно загрязнения с максимумом в точке выброса и понижением значений загрязнения к границам пятна, описываемым локальной функцией параболического типа:

$$f(j, s) = -a(j)r(j, s)^2 + H(j),$$

где $j = 1, \dots, n$ — номер источника в группе; r — удаление j -го источника от точки территории $s(X, Y)$ с условной мощностью выброса $H(j)$, $r = \sqrt{(tN(1, j) - X)^2 + (tN(2, j) - Y)^2}$; $a(j)$ — коэффициент разброса j -го источника.

В результате наложения (суммирования) выбросов группы промышленных источников с разными координатами размещения функция общего загрязнения области (территории) $F(s)$ обычно становится многоэкстремальной. Если заданы нормативы экологической безопасности области, то задачей управления (оптимизации) загрязнением такой территории будет поиск ограничений выбросов источников, при которых эти нормативы не нарушены.

Функции наложения выбросов отдельных источников, описываемых приближениями за счет конечной дискретности, также становятся приближениями.

Особенность задач управления (оптимизации) загрязнением территорий источниками промышленных выбросов, которые описываются многоэкстремальными функциями, не в обнаружении глобального и всех локальных максимумов (хотя последнее алгоритм предусматривает), а в поиске такого вектора подавления источников выбросов, который во всех точках рассматриваемой территории будет обеспечивать заданные нормативы экологических объектов (зон), также размещенных на территории.

В любой точке s территории декларировано соблюдение ограничения:

$$g(s) = \sum_{j=1}^N (-a(j)r(s, j)^2 + (1 - x(j))H(j)) - \text{norm}(s) \leq 0, \quad (3)$$

где $x(j)$ — элемент реформирования мощности j -го источника; $\text{norm}(s)$ — норматив загрязнения для точки s .

Ограничение (3) сопоставляет суммарный уровень загрязнения от источников с заявленным нормативом в каждой точке территории.

Целевую функцию конструировали в упрощенном варианте как предельную сумму затрат (потерь) в результате понижения мощностей источников по результатам оптимизации:

$$C(x) = \sum_{j=1}^N m(j)x(j)H(j), \quad (4)$$

где \mathbf{m} — вектор затрат на единицу понижения мощности источников.

Данную модель определения избыточных загрязнений выбросами группы источников применяли для сравнения переменной дискретности сетки в процедурах поиска решений алгоритма SIP с учетом графического оформления.

Численные эксперименты

Генерировали случайную выборку параметров 13 точечных источников выбросов, размещенных на области размерами $\mathbf{d}[400, 200]$ с координатами \mathbf{tN} (табл. 1, вариант 1). В вариантах $v = 1..4$ параметры мощностей выбросов \mathbf{H} и разброса \mathbf{a} не менялись. Пересчету подлежали:

- размеры области: $\mathbf{d} = \mathbf{d} \times \mathbf{u}$, где $\mathbf{u} = [1, 2, 4, 8]$ — вектор пересчета;
- координаты размещения источников: $\mathbf{tn} = \mathbf{tN} \times \mathbf{u}$. Это привело к значительным изменениям дискретности области в вариантах (табл. 1).

Дискретность представляет число точек (пикселей), а выражается количеством значащих цифр $0..9$ в числе независимо от того, целое оно или с плавающей точкой. Например, дискретности области $[7291 \times 536]$ и $[7,291 \times 53,6]$ по числу точек одинаковы, но сфокусированы на разных интервалах по

Параметры источников выбросов в вариантах $\nu = 1-4$

Параметры источников	Номера источников												
	1	2	3	4	5	6	7	8	9	10	11	12	13
H	1812	1460	2088	2087	1054	1506	1550	1505	774	1581	1087	601	1609
tN ($\nu = 1$) [400 × 200]	51 183	112 110	63 194	320 29	317 192	340 187	297 79	283 7	39 165	380 7	306 159	178 129	111 136
tN ($\nu = 2$) [800 × 400]	102 366	224 220	126 388	640 58	634 384	680 374	594 158	566 14	78 330	760 14	612 318	356 258	222 272
tN ($\nu = 3$) [1600 × 800]	204 732	448 440	252 776	1280 116	1268 768	1360 748	1188 316	1132 28	156 660	1520 28	1224 636	712 516	444 544
tN ($\nu = 4$) [3200 × 1600]	408 1464	896 880	504 1552	2560 232	2536 1536	2720 1496	2376 632	2264 56	312 1320	3040 56	2448 1272	1424 1032	888 1088
a	0,059	0,015	0,063	0,036	0,061	0,012	0,051	0,019	0,012	0,027	0,031	0,037	0,051

Таблица 2

Векторы x -решений для вариантов $\nu = 1-4$

ν	Номера источников												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0,7591	0,9856	1	0,5113	0,7796	1	1	1	0	0,8617	1	1	0,9791
2	1	0,4839	1	0,9427	1	0,8585	1	0,7777	0,3081	0,2283	0	0	0,8877
3	0,7166	0,1644	0,9553	0,8597	1	0,1899	0,5511	0,2492	0	0,2283	0,8842	0	0,9481
4	0,6012	0,1644	0,7799	0,4925	0,7602	0,1899	0,2129	0,1894	0	0,2283	0,2042	0	0,7801

X , Y . В работе для большей наглядности дискретность выражали целыми числами, так как в этом случае количество точек и фокусировка области совпадают.

Принят единый экологический норматив по загрязнению для любой точки области $\text{norm} = 1220$.

Дискретность точек в условиях наложения функций разброса источников порождает непредсказуемые варианты общей функции $F(s)$. Для сопоставления проектируемых вариантов дискретности генерировали в 3D общие функции загрязнения области (рис. 1–4, см. третью сторону обложки), что немедленно выявило их существенные различия.

При низкой дискретности сетки (см. рис. 1 на третьей стороне обложки) локальные максимумы себя не проявляют, в результате чего наложения выбросов источников нивелированы. Общая функция сильно сглажена в виде общего пятна, превышения ограничений над нормативом размыты на значительных участках области. Грубые точечные оценки загрязнений приводят к также грубым оценкам функции наложения $F(s)$. Завышенные оценки означают значительные нарушения ограничений на больших площадях области и, очевидно, требуют также значительных сокращений в предстоящей SIP-оптимизации.

Повышение дискретности сетки начинает проявлять локальные максимумы загрязнений, однако они

также еще сглажены. Визуально по координате Z интеграл функции $F(s)$ начинает падать.

Дальнейший рост дискретности (рис. 2, 3) приводит к росту числа максимумов с также более высокими градиентами. Явно наблюдается появление участков области с невысокими уровнями загрязнений (и даже с их отсутствием).

При высокой дискретности сетки (рис. 4) источники значительно локализованы, видны их градиенты, понижение эффекта наложения общей функции $F(s)$ и ее интеграла.

Варианты по табл. 1 использованы в качестве входных данных разработанного в MATLAB программного комплекса на базе алгоритма SIP. В табл. 2 представлены достигнутые аппроксимации по результатам оптимизации.

Вектор x трактуется как долевое понижение исходных единичных мощностей источников выбросов. Например, для источника 7 в варианте 1 $x(7) = 1$. Это означает, что вся его мощность для решения задачи признана избыточной. Напротив, для источника 9 в варианте 4 $x(9) = 0$. Поэтому его мощность при оптимизации может быть сохранена в исходном объеме. Соответственно $x(2) = 0,4839$ в варианте 2 рекомендует понижение исходной мощности источника 2 примерно вдвое. Таким образом, векторы решений дифференцированно отменяют, сохраняют или реформируют мощности всех источников.

На практике не обязательно понижать или закрывать источники с избыточными мощностями, что может привести к неприемлемым экономическим потерям. В качестве замены применяют улавливающие, фильтрующие или иные технологии. Тогда решения x будут означать дифференцированные объемы затрат по каждому источнику на их осуществление.

Обращает внимание более высокая "жесткость решений" при понижении дискретностей области. Так, число элементов в решениях, требующих кардинального закрытия источников выбросов, возрастает. Напротив, число элементов решений, требующих частичных ограничений мощностей источников возрастает в сторону более высоких дискретностей.

В целом работает логика: грубые приближения функций — грубые решения алгоритма SIP.

Если повышение дискретности сетки ведет к более качественным (распределенным по большинству элементов) аппроксимациям (табл. 3), то порогом при выборе достаточной дискретности может стать исчезновение из решения элементов с $x(j) = 1$. Судя по третьему столбцу табл. 3, такой момент наступает между вариантами 3 и 4. Варианты сеток 1 и 2 содержат значительное число единичных элементов и по такому принципу должны быть отклонены.

Эксперимент показал, что алгоритм SIP обеспечил устойчивые решения в широких диапазонах дискретности области для задачи, потенциально содержащей в ограничении многоэкстремальные функции. Также видно значительное влияние дискретности области на оптимальные аппроксимации.

На рис. 5 (см. четвертую сторону обложки) показаны 2D-карты загрязнения области в результате наложения выбросов источников после оптимизации.

Карты построены в соответствии с оптимальными сокращениями избыточных мощностей источников выбросов по табл. 2. Выведены зоны загрязнений территории для уровней вектора: $z = [30, 130, 280, 430, 600, 790, 950, 1220, 1400]$. Включение в вектор норматива 1220 показывает, что превышающих остаточных зон в картах нет, поэтому оптимизация достигается независимо от дискретности для всех вариантов. Таким образом, алгоритм SIP обеспечивает качественное решение независимо от дискретизации области.

Таблица 3

Количественные показатели вариантов решений

v	Число точек сетки ($\times 10^{-3}$, ед.)	Число элементов в аппроксимации $(\sum_{j=1}^n x(j),$ если $x(j) = 1)$	Число элементов в аппроксимации $(\sum_{j=1}^n x(j),$ если $0 < x(j) < 1)$
1	80	6	6
2	320	4	7
3	1280	1	10
4	5120	—	11

Грубая дискретность обеспечивает "смазанное" представление остаточного загрязнения территории. Высокая дискретность значительно ближе к реальной картине загрязнения, а ее оптимальное решение более точно очерчивает зоны остаточного загрязнения. По площади такие зоны значительно меньше на фоне полностью свободных от загрязнения.

Варианты 3, 4 исходно имели выраженные локальные пики загрязнений и оптимальные решения "срезали" их избыточные превышения (рис. 4, 5).

В вариантах 1, 2 с более распределенным типом исходных загрязнений оптимизация выполнила то же самое, но в больших объемах, причем остатки остались распределенными.

Сделана попытка ввода дополнительных критериев, связывающих параметры дискретности с результатами SIP-оптимизации.

Простая оценка может быть связана с изменением интеграла общей функции загрязнения для разной дискретности области. Если достижимо точное значение интеграла общей функции для ограниченной области, то отношение реального значения к точному назовем коэффициентом приближения функции W . Такую возможность представляет известное выражение для объема параболоида $V = \pi H^2 / (2a)$ в качестве интеграла параболической функции с бесконечным числом точек области.

Коэффициент $W = [0..1]$ выражает долевую часть численного значения интеграла общей функции загрязнения территории относительно аналитического значения интеграла:

$$W = \lim_{P \rightarrow \infty} \frac{\sum_{p=1}^P \sum_{j=1}^N (H(j) - a(j)r(j, p))^2}{\pi \sum_{j=1}^N (H(j)^2 / 2a(j))} \rightarrow 1, \quad (5)$$

где P — множество точек (дискретность) области ($P = d(1)d(2)$); $r(j, p)$ — удаление точки p от j -го источника.

Такой коэффициент в корреляции с дискретностью области, очевидно, будет постоянным для любых точек области в конкретной задаче, но пригоден только для оценки тренда приближений функции.

Обобщения решений (см. табл. 2) с помощью взвешенного пересчета приводят к коэффициенту остатков мощностей источников. Коэффициент взвешенных остатков мощностей группы источников (E) по результатам оптимизации составил

$$E = \frac{\sum_{j=1}^N (1 - x(j))H(j)}{\sum_{j=1}^N H(j)}, \quad (6)$$

где x — векторы решений в вариантах по табл. 2.

Достигнутые значения целевой функции в вариантах фиксировались по завершению алгоритма SIP.

Связи введенных критериев по результатам оптимизации с переменной дискретностью в вариантах представлены на рис. 6.

По варианту 1 (см. рис. 1) исходные загрязнения достигали высоких (до 8000...9000) значений, причем охватывали почти всю область, поэтому избыточность по сравнению с нормативом 1220 была значительной. Взвешенный остаток общей мощности от исходной 1 составил $E(1) = 0,1462$.

Напротив, по варианту 4 (см. рис. 4) выраженные пики загрязнений исходно соответствовали интервалу 1500...3500, что, естественно, означало меньшую избыточность. Решение по варианту 4 привело к взвешенному остатку мощностей $E(4) = 0,5978$.

Варианты 2, 3 со средними параметрами дискретностей и исходными загрязнениями по рис. 2, 3 привели также к средним взвешенным остаткам по результатам оптимизации: $E(2) = 0,2696$ и $E(3) = 0,4259$.

Высокая дискретность (вариант 4) примерно удваивает точность приближения функций. Конкретно, при отображении области количеством пикселей >5 млн приближение входит в зону $W \sim 0,9$, что можно считать приемлемым.

Затратная целевая функция C за счет высокой дискретности падает вдвое, что объясняется значительно меньшей "жесткостью" решений.

Темп нарастания взвешенных аппроксимаций E наиболее значителен (примерно четырехкратный), что говорит о большой чувствительности решений SIP к изменениям дискретности

Будучи упрощением, более низкая дискретность ведет к потере адекватности критериев, но с неопределенной мерой выбора.

Проведенный эксперимент показал, что алгоритм SIP дает достаточно хорошие решения в рассмотренных и, очевидно, в иных случаях вариативности дискретностью. Однако результаты оптимизации или введенные критерии указывают только на направления их изменений, а не на конкретные точки предпочтений. С позиции отсутствия формализации выбора дискретности в численных расчетах SIP можно отнести к эвристическим алгоритмам.

Допустимую дискретность области можно отождествить с мерой адекватности общей функции загрязнения. Из-за непредсказуемых вариантов наложения локальных выбросов группы источников выбор дискретности области оптимизации ложится на опыт исследователя, его умение анализировать и даже предвидеть результаты, на использование вероятностных оценок или косвенных критериев.

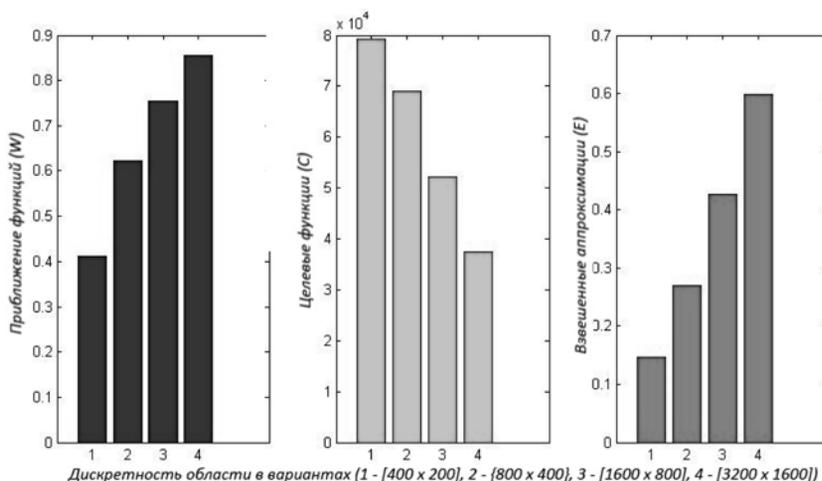


Рис. 6. Влияние дискретности области на результаты оптимизации

Следует представлять последствия низкой дискретности области. Очевидным улучшением будет заметное сокращение объемов вычислений, однако "плата" за такое сокращение весьма ощутима:

- потеря реальной геометрии функции общего загрязнения области;
- отставание интегральных критериев функций распределения загрязнений в сравнении с их значениями для бесконечной дискретности;
- растущее искажение вектора оптимального SIP-решения.

Главный недостаток недостаточной дискретности области — потеря адекватности функции наложения. Особенно это относится к многоэкстремальным функциям, которые появляются при разреженном размещении источников выбросов и наличии у локальных максимумов функций больших градиентов.

Выбор конечной дискретности не только обеспечивает запуск алгоритма SIP. Дискретность сохраняется и не может меняться до конца выполнения алгоритма. Поэтому ее связь с качеством итогового решения очевидна.

Сдерживают рост дискретности сетки разреженные сетки, однако их информативное качество падает. Мощным средством вычислительного эксперимента являются ЛП_τ-сетки (последовательности), которые заключаются в отказе от равномерных сеток [4]. Исключают дублирование получаемой информации, так как число малоинформативных точек с ростом дискретности катастрофически нарастает.

Роль дискретности для качества проявления функций наложения по аналогии близка проблеме качества визуализации изображений в экранных устройствах (мониторы, гаджеты), для которых прогресс размерностей экранов протекал в диапазонах от (640 × 480) до (3200 × 1800). Такой диапазон дискретностей представляет интерес для вычисли-

тельной оптимизации и был рассмотрен в работе. Не исключено, что соотношения размеров сторон области оптимизации также следует предпочитать уже сложившимся для экранных устройств (взамен (4×3) или (5×4) переходить на (16×9)).

В работе установлена чувствительность параметров дискретности пространства к прохождению и результатам алгоритма полубесконечной оптимизации. Приведенные обсуждения о роли и косвенные рекомендации по выбору дискретности в какой-то мере продвигают ситуацию к пониманию.

Подтверждено, что проблему ухода от бесконечности решает именно выбор дискретности сетки, а аппроксимации SIP лишь значительно понижают объемы вычислений при оптимизации методами математического программирования.

Выводы

Дискретность сетки области в численной реализации алгоритмов полубесконечной оптимизации значительно влияет на конечные аппроксимации.

Изменения дискретности сетки не приводят к потере сходимости аппроксимаций.

При возрастании дискретности взвешенный критерий групповой аппроксимации уверенно понижается.

Выбор дискретности сетки области должен достаточно качественно проявить (описать) характерные особенности общей функции для конкретной задачи, но остается эвристической процедурой.

Список литературы

1. Volkov Y. V., Zavriv S. K. A General Stochastic Outer Approximations Methods // SIAM Journal on Control and Optimization. 1997. Vol. 35. P. 1387—1421.
2. Завриев С. К., Новикова Н. М., Федосова А. В. Стохастический алгоритм решения выпуклых задач полубесконечной оптимизации с ограничениями равенствами и неравенствами // Вестник Московского университета. Сер. 15. Вычислит. матем. и кибернетика. 2000. № 4. С. 30—35.
3. Рапопорт Э. Я., Пleshivцева Ю. Э. Модели и методы полубесконечной оптимизации в обратных задачах теплопроводности // Сб. трудов 5-го Минского международного форума по тепло- и массообмену. Тезисы докл. и сообщ. Минск: Институт тепло- и массообмена им. Лыкова НАНБ. 2004. Т. 1. С. 274—276.
4. Антонова Г. М. Сеточные методы равномерного зондирования для исследования и оптимизации динамических стохастических систем. М.: ФИЗМАТЛИТ, 2007. 224 с.
5. Баранова М. Е., Гаврилов А. С. Методы расчетного мониторинга загрязнения атмосферы мегаполисов // Естественные и технические науки. 2008. № 4. С. 221—225.
6. Замай С. С., Якубайлик О. Э. Модели оценки и прогноза загрязнения атмосферы промышленными выбросами в информационно-аналитической системе природоохранных служб крупного города: учеб. пособие. Красноярск: Красноярский гос. ун-т, 1998. 109 с.
7. Vaz A. Ismael F., Ferreira E. C. Air pollution control with semi-infinite programming, *Applied Mathematical Modelling*. ELSEVIER, 2009, no. 33, pp. 1957—1969.

V. V. Fedosov, Moscow

A. V. Fedosova, Universidad Nacional de Colombia, Bogotá, Colombia

Parameterization of the Mesh Algorithms Semi-Infinite Optimization for the Problems of Industrial Ecology

The industrial ecology are typical management tasks pollution emission sources, the territory of the group. If you set the standards of environmental safety territory (scalar, vector, or function), the optimization should detect the redundancy of such emissions.

The task is complicated potential functions multiextremal general pollution of the territory as a result of imposition of emission sources, as well as the presence of the infinite parameter — the number of points of the territory, which leads to an infinite number of constraints.

Algorithms for semi-infinite optimization (SIP, Semi-Infinite Programming) reduces the initial formulation of the problem with an infinite number of constraints to the sequence of approximating problems with a finite number of critical constraints. Critical limits are maximizing function limitation from the start, the corresponding point of the next constraint violation. To search for violations of applied stochastic procedure. Approximating the final task it is usual formulation of nonlinear mathematical programming.

However, a theoretical departure from the infinity does not negate choice discrete grid area. The numerical experiment with parameterization by a discrete grid. Range Resolution made by $[400 \times 200]$ to $[3200 \times 1600]$.

A significant and sustainable impact on the approximation of the discrete nature of the results. Changes in grid discrete nature do not lead to a loss of convergence of approximations. Increase grid discreteness leads to better quality (distributed on most items) approximations.

Introduced and analyzed several surrogate grid discrete nature (approximation dirt overlay function, the objective function weighted approximations). The logic of their trend is well aligned with the variable grid increments.

Selecting an area must first demonstrate sufficient quality (to describe) the characteristics of the overall pollution of functions, but it remains a heuristic procedure.

Keywords: discrete grid, semi-infinite optimization, approximation, nonlinear programming, emission sources, pollution standards for areas

References

1. Volkov Y. V., Zavriev S. K. A General Stochastic Outer Approximations Methods, *SIAM Journal on Control and Optimization*, 1997, vol. 35, pp. 1387–1421.
2. Zavriev S. K., Novikova N. M., Fedosova A. V. Stohasticheskiy algoritm resheniya vypuklykh zadach polubeskonechnoy optimizatsii s ogranichenijami ravenstvami i neravenstvami, *Vestnik Moskovskogo universiteta. Ser. 15. Vychislit. matem. i kibernetika*, 2000, no. 4, pp. 30–35.
3. Rapoport Je. Ja., Pleshivceva Ju. Je. Modeli i metody polubeskonechnoy optimizatsii v obratnykh zadachah teploprovodnosti, *Sb. drudov: 5-yj Minskij mezhdunarodnyj forum po teplo- i massoobmenu. Tezisy dokl. i soobshh.* Minsk: Institut teplo i massoobmena im. Lykova NANB, 2004, no. 1, pp. 274–276.
4. Antonova G. M. *Setochnye metody ravnomernogo zondirovaniya dlja issledovanija i optimizatsii dinamicheskikh stohasticheskikh sistem*, Moscow: FIZMATLIT, 2007. 224 p.
5. Baranova M. E., Gavrilov A. S. Metody raschetnogo monitoringa zagriznenija atmosfery megapolisov, *Estestvennye i tehnicheckie nauki*, 2008, no. 4, pp. 221–225.
6. Zamaj S. S., Jakubajlik O. Je. *Modeli ocenki i prognoza zagriznenija atmosfery promyshlennymi vybrosami v informacionno-analiticheskoj sisteme prirodoohrannyh sluzhb krupnogo goroda*: Ucheb. posobie / Krasnojarskij gos. un-t. Krasnojarsk, 1998. 109 p.
7. Vaz A. Ismael F., Ferreira E. C. Air pollution control with semi-infinite programming, *Applied Mathematical Modelling*. ELSEVIER, 2009, no. 33, pp. 1957–1969.

УДК 004.052.3

Г. В. Зеленко, канд. техн. наук, проф.,

Московский государственный университет информационных технологий, радиотехники и электроники,

А. Д. Иванников, д-р техн. наук, зам. директора по научной работе,

Институт проблем проектирования в микроэлектронике РАН,

Д. Ю. Коган, ведущий инженер, ООО "Элитекс",

А. В. Роцин, канд. техн. наук, проф., e-mail: aleksey_roschin@mail.ru,

Московский государственный университет информационных технологий, радиотехники и электроники

Проблемы функциональной устойчивости сенсорной локальной сети

Предложен критерий оценки функциональной устойчивости сенсорной локальной сети, основанный на показателях достижимости любого узла сети из любого другого. Предложен алгоритм идентификации сенсорной локальной сети по результатам прохождения реальных или тестовых сообщений.

Ключевые слова: сенсорная сеть, достижимость, функциональная устойчивость

Введение

Широкое использование сенсорных локальных сетей для создания распределенных систем контроля и управления требует наличия эффективных методов и способов оценки способности таких систем выполнять свое функциональное назначение. Понятие *функциональная устойчивость*, используемое в настоящей статье, подразумевает возможность исполнения сенсорной локальной сетью своего предназначения, а именно, прохождения информационных пакетов от обычных узлов сети к выделенным управляющим узлам и управляющих пакетов от выделенных управляющих узлов к обычным [1, 2].

Проблема оценки функциональной устойчивости в различных формулировках исследуется уже достаточно давно [3–9], однако чаще всего такие исследования проводятся применительно к человеку или организационным системам. Есть работы [3], где функциональная устойчивость привязывается к понятию динамической устойчивости, однако это требует наличия аппарата динамического опи-

сания сенсорных локальных сетей (например, на основе динамических графов), которое в настоящее время недостаточно проработано.

В работе [4] рассматривается функциональная устойчивость навигационно-информационной системы (НИС): "...НИС представляет собой многообъектную распределенную иерархическую систему автоматизированного управления. При этом сами объекты управления НИС изменяют во времени под действием внешних воздействий и внутренних факторов не только свое состояние, но и местоположение в пространстве, что влечет за собой, вследствие пространственных неоднородностей телекоммуникационной подсистемы и навигационного поля, изменения структурно-функциональных связей внутри самой системы. Кроме того, воздействие внешней среды (преднамеренное или непреднамеренное) на элементы системы может существенно влиять на ее работоспособность. В связи с этим необходима способность сохранять или восстанавливать (полностью или частично) возможность выполнения возложенных на нее функций в условиях воздействия деструктивных

факторов. Такую способность будем трактовать как функциональную устойчивость НИС".

В работах [5, 6] говорится, что "Технологической основой обеспечения функциональной устойчивости стало создание бортовых информационно-управляющих комплексов, которые позволили комплексировать ресурсы системы и осуществлять их перераспределение. Только комплексирование всех каналов систем автоматического управления в единый информационно-управляющий комплекс, где информационно-измерительная подсистема будет включать все источники информации, устройства перекодирования и сопряжения, вычислительная система — все вычислители, а энергетическая подсистема — все исполнительные механизмы и источники энергии, позволит обеспечить ее функциональную устойчивость. Следует учитывать, что возможность комплексного использования ресурсов и обеспечения функциональной устойчивости систем ограничивается условиями устойчивости динамической системы, ее управляемости и наблюдаемости по Калману. На сегодняшний день достаточно полное исследование условий обеспечения функциональной устойчивости проведено для линейных нестохастических стационарных моделей. Поэтому задачи обеспечения функциональной устойчивости можно рассматривать как вид задач адаптивного оптимального управления "в большом", предполагающих оптимальное использование на каждом этапе или режиме функционирования системы всех располагаемых ресурсов (энергетических, информационных, вычислительных), для достижения главной для этого этапа цели при соблюдении множества ограничений."

В работе [7] понятие функциональной устойчивости строится на способности системы "гарантированно выполнять заданные функции, причем значения показателей, характеризующих качество выполнения системой функций, соответствуют установленным требованиям. Уровень гарантии при этом определяется выполнением заданных требований к качеству выполнения функций".

В работе [8] понятие функциональной устойчивости привязывается к понятию устойчивости системы по Ляпунову. При этом говорится, что "Качество функционирования СЭОУ в большой степени зависит от чувствительности показателей качества к изменениям параметров системы. Такого рода чувствительность определяет поведение системы в условиях нежелательной вариации ее параметров, и высокая чувствительность становится причиной того, что СЭОУ оказывается функционально неустойчивой."

К сожалению, конструктивных критериев функциональной устойчивости в указанных работах предложено не было.

Формирование критерия достижимости для сенсорной сети

В данной статье определяется конструктивный критерий функциональной устойчивости, основанный на достижимости любого узла сети из любого другого [1]. В работе [2] показано также, что сенсорные локальные сети достаточно хорошо описываются графом.

Как и любой другой, граф сенсорной локальной сети описывается парой — множеством вершин $X = [x_1, x_2, \dots, x_N]$, описываемым множеством узлов сети, и присоединенной матрицей A , описывающей связи между вершинами [9, 10]. Нумерация узлов сенсорной локальной сети в общем случае может быть произвольной. Удобнее, однако, начинать с выделенного (управляющего) узла, который собирает данные со всей сети и может координировать и конфигурировать ее узлы. Если выделенных узлов несколько, имеет смысл давать им младшие номера.

Как уже было сказано, необходимым условием функционирования сенсорной локальной сети является безусловная достижимость любого узла сети любым другим. То есть понятие функциональной устойчивости основано на следующей достижимости: сенсорная локальная сеть является функционально устойчивой, если существует путь длиной от 1 до N (где N — число узлов сети) из любого узла сети до любого другого. При этом запас функциональной устойчивости определяется числом таких путей. Чем больше путей возможно между узлами, тем более функционально устойчивой является сенсорная локальная сеть.

Критерий функциональной устойчивости сенсорной сети

Для формирования конструктивного критерия функциональной устойчивости рассмотрим присоединенную матрицу графа сети. В присоединенной матрице $A = \{a_{ij}\}_{N \times N}$ a_{ij} описывает возможность передачи сообщения от элемента i элементу j [1]. В качестве значений элементов a_{ij} могут выбираться значения, пропорциональные эффективной скорости передачи, определяемой числом потерянных пакетов, потребовавших повторной передачи. Для начала мы будем рассматривать простейший вариант — бинарные значения: 1 — есть связь, 0 — нет связи. Так как обычно в качестве узлов сенсорной локальной сети используются элементы, мощность передающей части которых может изменяться, разные узлы сети могут характеризоваться различной мощностью передатчика. В результате может возникнуть ситуация, при которой из узла i в узел j сообщение может быть передано, а из j в i — не может. В результате присоединенная матрица графа, описывающего сенсорную локальную сеть, в общем случае является несимметричной.

Для начала рассмотрим случай, когда все узлы сенсорной локальной сети имеют одинаковую мощность передающей части. В этом случае присоединенная матрица графа сети является симметричной. Тогда возможные переходы в сети за один шаг описываются присоединенной матрицей $A = \{a_{ij}\}_{N \times N}$. Все возможные переходы от узла к узлу за два шага описываются квадратом присоединенной матрицы A^2 , за три шага — третьей степенью и т. д. Все возможные переходы в сети за любое число шагов от 1 до N определяются суммой всех степеней присоединенной матрицы от 1 до N .

Таким образом, достижимость узлов сенсорной локальной сети определяется матрицей

$$D = \sum_{i=1}^N A^i, \quad (1)$$

которую назовем простой матрицей N -достижимости. Понятно, что в (1) A — присоединенная матрица графа, описывающего исследуемую сенсорную локальную сеть. В этом случае критерий достижимости i -го узла сети из любого другого, и наоборот, описывается условием

$$d_{ij} \neq 0, \quad i \neq j, \quad j = \overline{1, N}, \quad (2)$$

где N — число узлов исследуемой сети; d_{ij} — элемент, находящийся на пересечении i -й строки и j -го столбца матрицы D (1).

Критерием функциональной устойчивости сенсорной локальной сети является условие полной достижимости всех узлов сети, причем число возможных путей для каждого целевого узла должно быть более одного:

$$d_{ij} > 1, \quad i \neq j, \quad j = \overline{1, N}, \quad \dots, \quad i = \overline{1, N}. \quad (3)$$

В общем случае мощность передающей части различных узлов сети может быть различной, а следовательно, присоединенная матрица имеет произвольный вид. Для того чтобы оценить достижимость узлов сенсорной локальной сети с различной мощностью передатчиков узлов, необходимо сформировать более сложную матрицу:

$$C = \sum_{k=0}^{2N-1} \prod_{j=0}^{N-1} M_j(k), \quad (4)$$

где

$$M_j(k) = \begin{cases} A, & B_j(k) = 1; \\ A^T, & B_j(k) = 0; \end{cases} \quad (5)$$

$$B_j(k) = \begin{cases} 1, & k \& 2^j \neq 0; \\ 0, & k \& 2^j = 0; \end{cases} \quad (6)$$

A^T — транспонированная матрица A ;

$\&$ — побитовая операция над двоичными значениями (в данном случае, k и 2^j);

N — общее число узлов сенсорной локальной сети; $k = 0, \dots, 2^{N-1}$ — индекс, описывающий все возможные комбинации переходов в несимметричном графе сети (в сети с различной мощностью передатчиков в узлах).

Таким образом, комплексная матрица достижимости, построенная для произвольной присоединенной матрицы графа сети, определяется всеми возможными сочетаниями произведения прямой и транспонированной присоединенной матрицы графа сети.

Критерием функциональной устойчивости сенсорной локальной сети в случае произвольной присоединенной матрицы также является условие полной достижимости всех узлов сети, причем опять число возможных путей для каждого целевого узла должно быть более одного:

$$c_{ij} > 1, \quad i \neq j, \quad j = \overline{1, N}, \quad \dots, \quad i = \overline{1, N}, \quad (7)$$

где c_{ij} — элемент, находящийся на пересечении i -й строки и j -го столбца матрицы C (4).

Так как сенсорная локальная сеть строится из элементов, которые позволяют ей самоконфигурироваться, оценить структуру реальной сенсорной сети достаточно трудно, поскольку структура графа, описывающего такую сеть, зависит не только от числа и расположения узлов, но и от условий прохождения радиосигнала между узлами. Дополнительно усложняют оценку реальной структуры сенсорной сети различные значения мощности передатчиков узлов, которые могут быть установлены в процессе настройки сети управляющим (или управляемыми) узлом (узлами). Для получения объективной оценки показателей достижимости и функциональной устойчивости реальной сенсорной сети необходимо провести идентификацию ее структуры по результатам реальной работы. Для этого следует выполнить анализ всех пакетов (как информационных, передаваемых от обычных узлов — датчиков к выделенному управляющему узлу, так и управляющих пакетов, передаваемых в обратном направлении) за определенный интервал времени. Временной интервал должен быть достаточен для "проявления" в сенсорной сети по возможности всех возможных маршрутов прохождения пакетов. Для проведения такой идентификации пакеты, передаваемые в сети, должны быть снабжены информационным кадром, включающим в себя номер узла-источника и номера всех узлов, через который пакет прошел. Длина информационного кадра фиксирована — одно значение для узла-источника и N значений для элементов маршрута. В принципе, при передаче пакетов возможно образование циклов. Для исключения их в каждом узле должен анализироваться зафиксированный маршрут и, если в

маршруте обнаруживается собственный номер, что говорит о возникновении цикла, пакет ликвидируется. В этом случае N значений позволяют описать любой возможный маршрут в сети без циклов.

Идентификация структуры сенсорной сети для оценки ее функциональной устойчивости

Выполнение идентификации сенсорной локальной сети проводится следующим образом [2]:

- для каждого сообщения из выборки строятся цепочки прохождения узлов сети;
- на основании этих цепочек строится граф сенсорной сети по следующему правилу:
 - к элементу оценки ассоциированной матрицы графа сети \hat{a}_{ij} добавляется единица, если в исследуемом маршруте имеется переход из i -го узла сети в j -й;
 - предыдущий пункт выполняется для всех переходов всех тестовых маршрутов;
 - после анализа всех переходов всех маршрутов все элементы полученной ассоциированной матрицы делятся на число проанализированных маршрутов;
- в результате получается оценка ассоциированной матрицы графа исследуемой сенсорной локальной сети.

Понятно, что предложенный метод идентификации дает в результате достаточно грубую оценку ассоциированной матрицы графа исследуемой сети, так как из-за возможного наличия множественных путей в каждом маршруте реализуется только один конкретный. Для идентификации полной ассоциированной матрицы графа сенсорной сети необходимо предусмотреть поочередное отключение отдельных узлов сети и их комбинаций [2].

Реализация алгоритма идентификации сенсорной сети для оценки ее функциональной устойчивости

Для реализации предложенного алгоритма была разработана программа на языке C++.

Для работы программы необходим набор тестовых или реальных маршрутов сообщений, прошедших через исследуемую сеть. В случае идентификации реальной сенсорной локальной сети используются реальные наборы маршрутов. Для тестового прогона программы использовался выход модели сенсорной локальной сети, также реализованной в виде программы [2].

На рис. 1 показано окно для ввода матрицы модели сенсорной локальной сети. На рис. 2 представлен результат работы программы идентификации.

Как видно из рисунков, в идентифицированной матрице присутствуют не все элементы исходной матрицы, а некоторые присутствующие имеют меньшие значения, чем в исходной.

	1	2	3	4	5	6
1	0	1	1	0	0	0
2	1	0	1	1	0	0
3	1	1	0	1	0	0
4	0	1	1	0	1	1
5	0	0	0	1	0	0
6	0	0	0	1	0	0

Рис. 1. Ввод матрицы в программу моделирования сети

	1	2	3	4	5	6
1	0.00	0.50	0.50	0.00	0.00	0.00
2	0.13	0.00	0.25	0.63	0.00	0.00
3	0.13	0.25	0.00	0.62	0.00	0.00
4	0.00	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	1.00	0.00	0.00

Рис. 2. Результат идентификации ассоциированной матрицы графа сенсорной сети

Как уже было сказано, причины этого заключаются в том, что не все узлы и не всегда участвовали в передаче тестовых сообщений.

В том случае, когда проводится идентификация реальной сенсорной локальной сети, по результатам идентификации могут быть построены критерии достижимости и функциональной устойчивости исследуемой сети.

Заключение

В статье предложен критерий оценки функциональной устойчивости сенсорной локальной сети, основанный на показателях достижимости любого узла сети из любого другого. Предложен алгоритм идентификации сенсорной локальной сети по результатам прохождения реальных или тестовых сообщений.

Использование поочередного отключения отдельных узлов сети и их комбинаций позволит оценить критические места сенсорной сети. Эти оценки могут использоваться для выработки стратегии коррекции конфигурации сети.

Список литературы

1. Баранов М. А., Зеленко Г. В., Рошин А. В. Проблемы маршрутизации сообщений в сенсорной локальной сети // Информационные технологии. 2014. № 11. С. 8–12.
2. Баранов М. А., Зеленко Г. В., Рошин А. В., Степанова И. В. Проблемы идентификации сенсорной локальной сети. Актуальные проблемы аппаратно-программного и информационного обеспечения науки, образования, культуры и бизнеса // VII международная научно-практическая интернет-конференция, 28–30 апреля 2015. М.: МГУПИ, 2015. [Электронный ресурс]. URL: http://it-4-mgupi.ru/conference/iconf_about.html
3. Семеняк М. В., Федоров В. К. Вариация энтропии как аналог функции Ляпунова в статистическом анализе функциональной устойчивости. [Электронный ресурс]. URL: vestnik.omgau.ru/wp-content/files/7_12.pdf
4. Королев А. Н., Тарасов А. А. О функциональной устойчивости навигационно-информационных систем. [Электронный ресурс]. URL: <http://studik.net/o-funkcionalnoj-ustojchivosti-navigacionno-informacionnyx-sistem/>
5. Машков О. А., Самчишин О. В. Современные проблемы развития теории функционально устойчивых сложных систем управления. [Электронный ресурс]. URL: [dspace.nbu.gov.ua>bitstream/handle/123456789/](http://dspace.nbu.gov.ua/bitstream/handle/123456789/)
6. Машков О. А. О функциональной устойчивости бортовых информационно-управляющих комплексов // Вопросы повышения эффективности и качества систем управления полетом и навигации воздушных судов. К.: КИИГА, 1990. С. 79–83.
7. Бородакий Ю. В., Тарасов А. А. О функциональной устойчивости информационно-вычислительных систем // Изв. Южного федерального университета. Технические науки. 2006. Т. 62. Вып. № 7. [Электронный ресурс]. URL: <http://cyberleninka.ru/article/n/o-funktsionalnoj-ustojchivosti-informatsionno-vychislitelnyh-sistem>
8. Айзерман М. А., Гусев Л. А., Смирнова И. М., Петров С. В. Динамический подход к анализу структур, описываемых графами (основы графодинамики) // Автоматика и телемеханика. 1977. № 7. С. 135–151.
9. Саати Т. Л. Математические модели конфликтных ситуаций. М.: Советское радио. 1977. 304 с.
10. Оре О. Теория графов. М.: Наука, 1980. 336 с.

G. V. Zelenko, Professor,

Moscow State University of Information Technologies, Radio Engineering and Electronics,

A. D. Ivannikov, Deputy Director for Research of the Institute for Design Problems
in Microelectronics of Russian Academy of Sciences,

D. Yu. Kogan, Chief Engineer of "Elitex" LLC,

A. V. Roshchin, Professor, e-mail: aleksey_roschin@mail.ru,

Moscow State University of Information Technologies, Radio Engineering and Electronics

Problems of Functional Stability in Sensor Network

The paper proposes the criterion for evaluating the functional stability of the sensor network, based on the reachability of any network node from any other. The existing papers of the systems' functional stability research were analyzed.

It is shown that cited papers did not propose suitable constructive criteria of functional stability. Functional stability criteria are considered for the sensor network with symmetrical and asymmetrical attached matrix of the network graph. It is shown that for an objective evaluation of functional stability and attainability of real sensor network, one must identify its structure as a result of real work. To do this, an analysis of all packets sent over a certain period of time is required. The analysis period should be sufficient for the development of all possible routes of the network packets. Since the transmission of packets in a sensor network can create loops, the special measures are provided to avoid those loops. An algorithm for the identification of the sensor network as a result of the passage of real or test messages is proposed.

Keywords: sensor network, reachability, functional stability

References

1. Baranov M. A., Zelenko G. V., Poshchin A. V. Problemy marshrutizatsii soobshhenij v sensornoj lokal'noj seti, *Informacionnyye tehnologii*, 2014, no. 11, pp. 8–12.
2. Baranov M. A., Zelenko G. V., Roshchin A. V., Stepanova I. V. Problemy identifikatsii sensornoj lokal'noj seti. Aktual'nye problemy apparatno-programmnogo i informacionnogo obespecheniya nauki, obrazovaniya, kul'tury i biznesa. *VII mezhdunarodnaja nauchno-prakticheskaja internet-konferencija*, 28–30 aprelja 2015. M.: MGUPI, 2015. [Elektronnyj resurs]. URL: http://it-4-mgupi.ru/conference/iconf_about.html
3. Semenjak M. V., Fedorov V. K. *Variacija entropii kak analog funktsii Lyapunova v statisticheskom analize funkcional'noj ustojchivosti*. [Elektronnyj resurs]. URL: vestnik.omgau.ru/wp-content/files/7_12.pdf
4. Korolev A. N., Tarasov A. A. *O funkcional'noj ustojchivosti navigacionno-informacionnyh sistem*. [Elektronnyj resurs]. URL: <http://studik.net/o-funkcionalnoj-ustojchivosti-navigacionno-informacionnyx-sistem/>
5. Mashkov O. A., Samchishin O. V. *Sovremennye problemy razvitiya teorii funkcional'no ustojchivykh slozhnykh sistem upravlenija*. [Elektronnyj resurs]. URL: dspace.nbu.gov.ua/bitstream/handle/123456789/
6. Mashkov O. A. *O funkcional'noj ustojchivosti bortovykh informacionno-upravljajushchih kompleksov, Voprosy povysheniya effektivnosti i kachestva sistem upravlenija poletom i navigacii vozdušnykh sudov*. Kiev: KIIGA, 1990, pp. 79–83.
7. Borodakij Ju. V., Tarasov A. A. *O funkcional'noj ustojchivosti informacionno-vychislitel'nyh sistem. Izvestija Juzhnogo federal'nogo universiteta. Tehnicheskie nauki*, 2006, vol. 62, vyp. 7. [Elektronnyj resurs]. URL: <http://cyberleninka.ru/article/n/o-funktsionalnoj-ustojchivosti-informatsionno-vychislitelnyh-sistem>
8. Ajzerman M. A., Gusev L. A., Smirnova I. M., Petrov S. V. *Dinamicheskij podhod k analizu struktur, opisываемых графами (osnovy grafodinamiki)*, *Avtomatika i telemehanika*, 1977, no. 7, pp. 135–151.
9. Saati T. L. *Matematicheskie modeli konfliktnyh situacij*. Moscow: Sovetskoe Radio. 1977. 304 p.
10. Ore O. *Teorija grafov*. Moscow: Nauka, 1980. 336 p.

УДК 004.713, 004.75

Ю. А. Холопов¹, вед. инженер, Ле Ба Чунг², аспирант, e-mail: chungbaumanvietnam@gmail.com,
Нгуен Тхань Чунг², аспирант, Чан Ван Хань¹, канд. техн. наук, инженер,

¹Институт точной механики и вычислительной техники им. С. А. Лебедева РАН,

²Московский физико-технический институт (ГУ)

Особенности реализации распределенных систем управления на основе аппаратных сетевых решений

Рассмотрены аспекты внедрения сетевых технологий в распределенные системы управления. Описаны различные приемы аппаратной поддержки сетевых решений, обеспечивающие повышение пропускной способности информационной среды за счет исключения арбитражных процедур, исключения управляющей информации из пакетов, организации обмена центрального процессора и периферийных устройств через укрупненные пакеты состояния и управления. Реализация на FPGA предложенных технологий позволяет достичь минимальных и стабильных задержек передачи данных в бортовых сетях.

Ключевые слова: распределенная система управления, сетевые интерфейсы, коммутатор, конвертор, регистратор, CAN, Ethernet

Введение

В настоящее время распределенные системы управления (PCY) представляют собой сложные, высокодинамичные системы с разнесением компонентов в пространстве на десятки метров. Эти системы могут включать в себя большое число датчиков, исполнительных устройств, узлов управления-вычисления, связанных между собой через сетевую информационную среду.

В современных PCY отмечается тенденция использования разных цифровых сетевых интерфейсов на всех структурных уровнях. Например, функции связи между модулями, внутри электронных блоков системы могут выполнять интерфейсы SPI, I2C, USB. Для подключения к системе различных датчиков, исполнительных устройств используют такие распространенные последовательные интерфейсы, как CAN, UART, Profibus, Modbus, LIN, HART. В комплексах бортового оборудования высокоуровневые связи между сложными системами и связи с системой управления верхнего уровня организуются с помощью интерфейсов Ethernet, CAN, ARINC, STANAG и т.п.

1. Проблемы использования сетевых технологий в распределенных системах управления

Под информационной средой (ИС) распределенной системы управления понимается совокупность технических и программных средств для хранения, обработки и передачи информации. В нее входят следующие компоненты: центральный про-

цессор, оконечные устройства, коммутаторы и совокупность физических связей узлов (сеть). В системах управления используется централизованная сетевая информационная среда, спецификой которой являются гетерогенная структура абонентов с резервированным ведущим и наличие средств документирования трафика для послеаварийного анализа работы системы — регистратора. На каждом уровне иерархии PCY используют разные, обычно стандартные типы сетей, максимально соответствующие функции межсоединений данного уровня. Однако использование разных сетей для обеспечения связей между компонентами системы управления порождает следующие проблемы:

- ✓ наличие разных типов сетей на разных уровнях PCY требует решения задачи их объединения;
- ✓ существующие форматы пакетов не оптимальны для использования в распределенных системах управления вследствие избыточности управляющей информации;
- ✓ в универсальной сети присутствуют механизмы управления обменами, которые не нужны для PCY, например, избыточная размерность системы адресации, механизм контроля доступа к среде передачи данных и т.п.;
- ✓ при проектировании некоторых управляющих сетей, существует "проблема масштаба", когда необходимо комплексировать большое число датчиков и исполнительных устройств, но отсутствует единая методика, которая позволила бы сделать это, не ухудшая динамические характеристики системы управления;

- ✓ наличие большого числа абонентов в ИС РСУ усложняет задачу их контроля и мониторинга стандартными средствами;
- ✓ необходимость использования в РСУ готовых периферийных устройств с разными сетевыми интерфейсами требует разработки типовой технологии их комплексирования.

Современные бортовые сети позволяют не только обеспечить процесс обмена информацией между компонентами РСУ, но и являются ее информационной основой. Для сетевой ИС распределенной системы управления основным требованием является минимальное и известное время передачи информации [1]. Обеспечение прозрачного механизма взаимодействия между узлами сети напрямую сказывается на качестве функционирования всей РСУ. Фактически, будет решаться задача создания бесконфликтной коммуникационной среды между компонентами распределенных систем управления.

Актуальность поставленной задачи обусловлена не только постоянным развитием сложности объектов управления, но и повышенными требованиями к качеству собственно управления объектами. Практическая ценность оптимизации сетевой информационной среды РСУ заключается в том, что она позволяет обеспечить повышенный уровень точности управления и надежности. В единой сетевой ИС РСУ появляется дополнительная возможность децентрализованного протоколирования практически всех информационных событий. При этом незначительно усложняется ее структура, не разрушается основная функциональность системы. Далее рассмотрим методы решения описанных выше проблем на конкретных примерах.

2. Объединение разнообразных сетей в высокодинамичных распределенных системах управления

Существует широкий спектр устройств, позволяющих обеспечить решение задачи объединения разных сетей в режиме доступа к абонентам в адресном пространстве одной объединенной сети, это, например, конвертеры HD67290 (CAN/RS232), HD67048 (CAN/Ethernet), HD67390-U-D1 (CAN/USB), и т.п.

Недостатки таких универсальных готовых решений заключаются в том, что они имеют невысокий уровень настройки параметров комплексирования через сетевой интерфейс верхнего уровня. В режиме настройки через технологический интерфейс эти универсальные решения избыточны, а поэтому дороги и трудно настраиваемы.

Наиболее целесообразной для объединения двух сетей с разными интерфейсами представляется реализация в рамках системы управления специализированного конвертора, подходящего не только по функциям, но и по способу комплексирования, обладающего возможностью расширения набора функций и внешних интерфейсов. При этом конвертор будет играть роль ведущего узла для одного типа сети (например, сети 1) и ведомого для другой сети

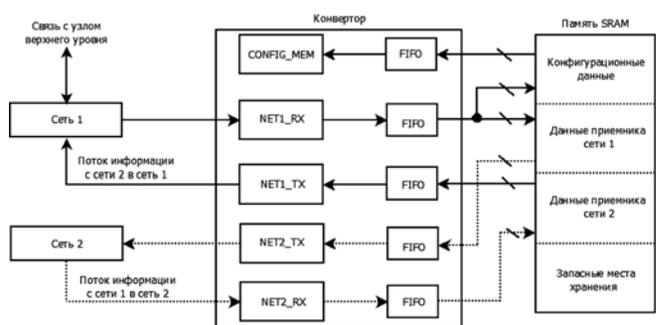


Рис. 1. Обобщенная функциональная схема реализации конвертора, объединяющего две сети



Рис. 2. Обобщенный алгоритм функционирования конвертора

(сети 2). Обобщенная функциональная схема аппаратной реализации конвертора показана на рис. 1.

Комплексирование разных сетей заключается в переформатировании пакетов, соответствующих протоколам сетей 1 и 2, в общей памяти. Буферы FIFO используются для пересинхронизации и согласования скоростей потоков сетевой информации и порта доступа к общей памяти. Обобщенный алгоритм функционирования конвертора показан на рис. 2.

На основе предложенного алгоритма функционирования и обобщенной функциональной схемы аппаратной реализации был разработан конвертор промышленных сетей Profibus и Modbus, как часть логики центрального вычислителя системы управления [2]. В данном случае конвертор играет роль ведущего узла в сети Profibus и ведомого в сети Modbus. Конвертор позволяет объединить одну сеть Profibus с одновременно несколькими равноскоростными сетями Modbus. Аппаратное решение обладает необходимыми значениями компактности, гибкости, экономичности по объему имеющихся логических ресурсов.

3. Обеспечение минимального времени передачи данных в высокодинамичных распределенных системах управления

Еще одной проблемой, возникающей в высокодинамичных РСУ, является необходимость обеспечения минимального и стабильного времени передачи данных между узлами системы. Решение данной задачи позволяет не только повысить частоту обработки данных в системе управления, но и уменьшить фазовые искажения (устаревание информации). Для решения данной проблемы требуется применение комплексных методов в сетевом аппаратном и программном обеспечении. Рассмотрим некоторые методы аппаратных решений.

Построение слабосвязанной распределенной системы управления. Одним из путей решения про-

блемы "быстрых передач" является построение слабосвязанной РСУ. Распределенная система управления строится таким образом, чтобы все сильные связи, все фазы процесса управления, необходимые для передачи большого объема информации с высокой частотой, замыкались внутри локальных подсистем через высокоскоростные сети. А для межуровневого взаимодействия можно использовать сети с меньшей скоростью передачи информации. По сети связи между подсистемами передаются только высокоуровневые команды управления и укрупненные параметры состояния. Модель слабосвязанной распределенной системы управления является одной из широко используемых концепций разработки бортовых систем управления.

В работе [3] описано решение задачи обеспечения работоспособности системы подвески автомобиля на основе концепции слабосвязанной РСУ. Информационное взаимодействие контуров управления узлами подвески (горизонтальные связи) реализовано через локальную высокоскоростную сеть передачи информации. Связь с узлом управления верхнего уровня не требует такой высокой скорости передачи информации, и поэтому может быть реализована через сеть другого типа.

Модификация промышленных универсальных сетевых протоколов. Другой путь решения проблемы "быстрых передач" — это упрощение формата пакета данных, т.е. модификация универсальных сетевых протоколов. Разработанные на сегодняшний день универсальные протоколы не учитывают специфику информационных процессов в РСУ. Избыточность управляющей информации в формате пакетов состояния и управления может ухудшить качество РСУ с точки зрения эффективности и помехоустойчивости системы. Предлагаемая коррекция протоколов базируется на отличных от универсальных сетей характеристиках информационных обменов в РСУ: статичности состава абонентов в системе и цикличности информационных процессов. Эти характеристики означают наличие возможности взаимодействия вычислителя с периферийными устройствами системы по фиксированному расписанию. Отсутствие асинхронности во взаимодействии с периферийными компонентами системы позволяет упростить адресацию абонентов и исключить конфликты между абонентами сети. Рассмотрим подробно эту идею на примере задачи аппаратной реализации коммутатора Fast Ethernet.

Благодаря высокой скорости передачи данных (порядка 100 Мбит/с) и широкому спектру аппаратных решений, Ethernet часто используют в различных современных системах управления. Однако достаточно высокое значение и недетерминированность времени задержки в процессе передачи данных между узлами верхнего и нижнего уровней через обычный коммутатор ограничивает возможности его применения для высокодинамичных РСУ. Предлагается решение данной проблемы путем реализации специального конвертора, связанного с

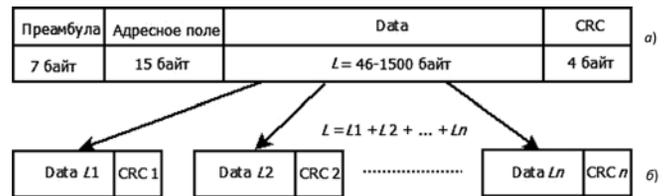


Рис. 3. Форматы пакетов: а — протокола Fast Ethernet; б — нижнего уровня

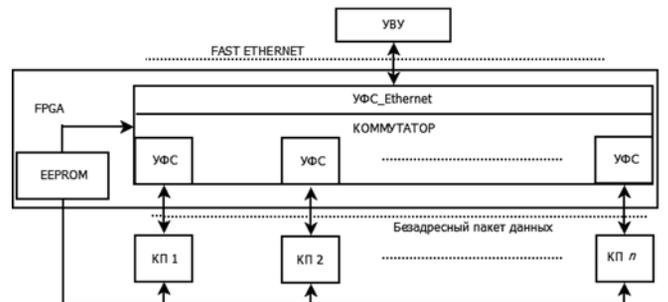


Рис. 4. Структурная схема аппаратной реализации конвертора

узлом верхнего уровня через стандартный протокол Ethernet, и использующего сокращенный формат пакетов для обмена данными с контроллерами нижнего уровня. Фактически, на нижнем уровне РСУ нет необходимости в функциях сетевого протокола, а сетевой протокол вырождается в бездресный пакетный интерфейс: адресация выполняется по "подключению" к каналам конвертора, границы пакетов определяются по паузе между циклами регулирования, процедура доступа к сети отсутствует ввиду режима работы "по расписанию". Формат пакетов протокола Fast Ethernet и формат пакетов нижнего уровня показаны на рис. 3, где L — длина поля данных пакетов протокола Fast Ethernet, а L_1, L_2, \dots, L_n — длины полей данных пакетов нижнего уровня. Процессы упаковки и распаковки пакетов в конверторе сопровождаются формированием и анализом контрольных кодов. Структурная схема аппаратной реализации конвертора показана на рис. 4.

Здесь УВУ — узел верхнего уровня; КП 1, КП 2, ..., КП n — контроллеры периферии; УФС — узел физического сопряжения с линией связи (приемопередатчик). Минимальная задержка достигается благодаря аппаратному преобразованию стандартных пакетов верхнего уровня в несколько сокращенных пакетов для нижестоящих узлов с упрощенной адресацией последних по подключению, что обеспечивается заранее составленным расписанием.

Аппаратное обеспечение подключения к бортовой сети разнообразных периферийных устройств. В современных распределенных системах управления широко используют разнообразные периферийные устройства, работающие с разной скоростью и с разными сетевыми протоколами и цифровыми интерфейсами. Поэтому актуальной задачей является разработка типовой технологии комплексирования разнообразных периферийных устройств к бортовой сети. Практические реализации на при-

мере контроллеров широко используемых сетей Profibus DP и CAN [4] показали, что аппаратное решение обеспечивает подключение к сети большого числа периферийных устройств с высокой скоростью передачи данных, в оптимальном формате пакетов данных и с минимальными требованиями к объему аппаратуры. Структурная схема аппаратной реализации контроллера периферии с CAN-интерфейсом показана на рис.5.

В качестве цифровых интерфейсов периферийных устройств могут служить UART, SPI, I2C и т.п. Реализация контроллера периферии зависит от конкретной бортовой сети и интерфейсов периферийных устройств.

Отметим некоторые принципиальные особенности реализации контроллера периферии, порождаемые характеристиками PCY и выбранной стратегией управления контроллером.

- ✓ Благодаря статичности характеристик PCY заранее известны конфигурационные параметры контроллера периферии.
- ✓ Все модули управления периферийными устройствами могут быть реализованы автономно, и большинство из них функционирует независимо от типов интерфейсов бортовой сети.
- ✓ Данные, полученные от разных периферийных устройств, постоянно обновляются в блоке формирования пакетов и всегда готовы для пересылки.
- ✓ Процесс согласования скоростей разнообразных модулей в составе контроллера периферии выполняется либо на основе использования буферов, либо по запросам от сети.

4. Концепция контроля, мониторинга и обеспечения качества распределенных систем управления на основе реализации автономного регистратора

Через сетевую информационную среду PCY передаются все параметры, характеризующие состояние объекта управления в данном цикле регулирования. Одним из абонентов сетевой ИС может быть узел регистрации и контроля (УРК), следящий за всеми обменами (рис. 6).

Узел регистрации и контроля не адресуется в цикле регулирования и является устройством, фиксирующим все обмены независимо от адресации, и работающим автономно вне зависимости от цикла регулирования системы управления. Обращение к УРК возможно только в технологическом режиме при отладке системы. В данном узле сохраняются мгновенные параметры, фиксируемые в каждом цикле регулирования PCY [5].

УРК может быть настроен на запись параметров, выбираемых по заранее заданной маске. Выбор параметров для контроля и мониторинга осуществляет разработчик при настройке УРК через стандартные интерфейсы, например CAN, I2C и т.п. Подключение к коммуникационной среде PCY позволяет независимо от структуры внутренней сис-

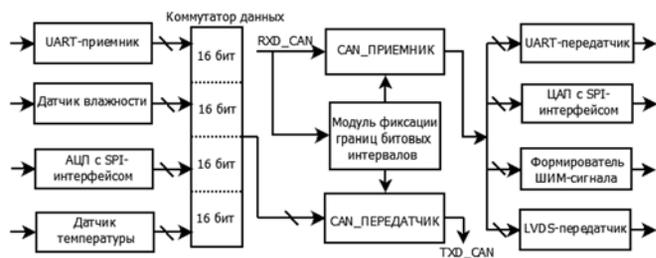


Рис. 5. Структурная схема контроллера периферии с CAN-интерфейсом



Рис. 6. Схема подключения регистратора к системе управления

темы и алгоритмов ее функционирования получить различные параметры системы и данные, циркулирующие внутри системы управления.

Заключение

В распределенных системах управления актуальной тенденция замены неструктурированных цифровых связей между элементами системы на сетевые интерфейсы. Использование универсальных сетевых протоколов в ИС PCY избыточно. Оптимизация форматов и способов организации связей в информационной среде PCY основана на статичных характеристиках обменов информации между составными частями распределенных систем управления. В обычных PCY синхронизация узлов недостаточно широко распространена. Предлагаемый подход обеспечивает синхронность работы всех составных частей системы управления. Сетевая ИС как однотипное коммуникационное пространство PCY позволяет использовать дополнительные распределенные по объекту управления узлы протоколирования, контроля и функционирования PCY. Реализация предложенных приемов аппаратной поддержки сетевых решений на FPGA необходима для обеспечения максимальной точности механизма синхронизации обменов. Сокращение задержек передачи данных между узлами PCY повышает качество управления объектом.

Список литературы

1. Преображенский Н. Б., Ле Ба Чунг, Чан Ван Хань, Дам Чонг Нам. Некоторые особенности реализации сетевых решений в системах управления // Международная конференция Инжиниринг & Телекоммуникации — En & T 2014: тезисы докладов. Москва/Долгопрудный, 26–28 ноября 2014. М.: Изд-во МФТИ, 2014. С. 217–218.
2. Дам Чонг Нам. Комплексирование MODBUS и PROFBUS в промышленных сетях. // Сборник "XL ГАГАРИНСКИЕ ЧТЕНИЯ. Научные труды Международной молодежной научной конференции в 9 т. Москва, 7–11 апреля 2014 г." М.: Изд-во МАТИ. 2014. Т. 5. С. 218–219.

3. **Холопов Ю. А., Чан Ван Хань, Преображенский Н. Б.** Аппаратная оптимизация бортовой распределенной системы управления подвеской автомобиля на основе ее свободных степеней информационных связей // Глобальный научный потенциал. 2014. № 11 (44). С. 113—116.

4. **Ле Ба Чунг.** Контроллер периферии с CAN-интерфейсом // Труды 56-й всероссийской научной конференции МФТИ "Актуальные проблемы фундаментальных и прикладных наук в совре-

менном информационном обществе. Москва, 25—30 ноября 2013 г.". М.: МФТИ, 2013. С. 59—60.

5. **Чан Ван Хань.** Исследование и разработка метода построения высокоавтономного регистратора параметров в цифровой системе управления // XXXIX ГАГАРИНСКИЕ ЧТЕНИЯ: научные труды Международной молодежной научной конференции в 9 т. Москва, 9—13 апреля 2013 г.". М.: МАТИ, 2013. Т. 4. С. 183—185.

Ju. A. Holopov¹, Leader Engineer, **Le Ba Chung^{1, 2}**, Graduate Student, **Nguyen Thanh Trung^{1, 2}**, Graduate student, **Tran Van Khanh^{1, 2}**, PhD,
¹Lebedev Institute of Precision Mechanics and Computer Engineering
²Moscow Institute of Physics and Technology (State University),
E-mail: chungbaumanvietnam@gmail.com

Features of Implementation of Distributed Control Systems Based on Hardware Network Solutions

The article discusses aspects of the implementation of network technology in distributed control systems. The described various approaches of hardware support for network solutions provide higher throughput capacity in the information environment by: exclusion of arbitration procedures, elimination of control information from the data packages, the organization of exchange between the CPU and peripherals based on a unified packages of state and control. The proposed technology, implemented on FPGA, allows to achieve minimal and stable delay of transfer data in on-board networks.

Keywords: distributed control system, network interfaces, switch, converter, recorder, CAN, Ethernet

References

1. **Преображенский Н. В., Ле Ба Чунг, Тран Ван Хань, Дам Trong Нам.** Некоторые особенности реализации сетевых решений в системах управления, *Международная конференция Инжиниринг & Телекоммуникации — En & T 2014, Tezisy dokladov*, Moscow/Dolgoprudnyj, 26—28 November 2014, Moscow: MFTI, 2014, pp. 217—218.
2. **Дам Trong Нам.** Комплексирование MODBUS и PROFBUS в промышленных сетях, *XL GAGARINSKIE ChTENIJA. Nauchnye trudy Mezhdunarodnoj molodezhnoj nauchnoj konferencii v 9 vol.* Moscow, 7—11 April 2014, Moscow: MATI, 2014, vol. 5, pp. 218—219.
3. **Холопов Ю. А., Тран Ван Хань, Преображенский Н. В.** Аппаратная оптимизация бортовой распределенной системы управления

подвеской автомобиля на основе ее свободных степеней информационных связей, *Global'nyj nauchnyj potencial*, 2014, no. 11 (44), pp. 113—116.

4. **Ле Ба Чунг.** Контроллер периферии с CAN-интерфейсом, *Труды 56-ой всероссийской научной конференции МФТИ "Актуальные проблемы фундаментальных и прикладных наук в современном информационном обществе"*, Moscow, 25—30 November 2013. М.: МФТИ, 2013, pp. 59—60.

5. **Тран Ван Хань.** Исследование и разработка метода построения высоко-автономного регистратора параметров в цифровой системе управления, *XXXIX GAGARINSKIE ChTENIJA. Nauchnye trudy Mezhdunarodnoj molodezhnoj nauchnoj konferencii v 9 vol.* Moscow, 9—13 April 2013. Moscow, MATI, 2013, vol. 4, pp. 183—185.

УДК 004.942

А. Ю. Романов, канд. техн. наук, ст. преп., e-mail: a.romanov@hse.ru,
Национальный исследовательский университет "Высшая школа экономики"

Исследование сетей на кристалле с топологией *mesh* с помощью модели *NoCTweak*

Проанализировано влияние геометрической формы топологии и размещения "горячих точек" на эффективность сетей на кристалле с помощью модели NoCTweak на базе языка SystemC. Результаты моделирования показали, что использование топологий, наиболее близких по форме к квадратной, позволяет до 24,7 % увеличить пропускную способность сети на кристалле, а центральное размещение "горячих точек" дает возможность до 9 % уменьшить задержки прохождения пакетов и до 15,2 % — энергозатраты на передачу каждого из них при увеличении до 19,5 % пропускной способности сети.

Ключевые слова: сеть на кристалле, система на кристалле, регулярная топология сети на кристалле, проектирование сетей на кристалле, язык программирования SystemC, модель сети на кристалле, топология mesh, "горячие точки"

Одной из основных тенденций развития современных систем на кристалле (СнК) считается появление многоядерных систем, где в рамках одного кристалла интегрируются десятки вычислительных узлов, периферийных модулей и модулей памяти. Насущная необходимость в объединении различных компонентов таких СнК обуславливает широкое использование сетевых решений и применение сетей на кристалле (СтнК). СтнК представляет собой множество вычислительных узлов и вспомогательных модулей, объединенных общей подсистемой связи, состоящей из маршрутизаторов и соединений между ними. Подсистема связи СтнК, занимая значительные ресурсы кристалла, является энергозатратной [1, 2]. Всевозрастающие требования к ресурсоемкости и скорости передачи данных по сети диктуют необходимость анализа факторов, влияющих на эффективность СтнК (топология сети, структура маршрутизаторов и алгоритмы маршрутизации данных, распределение вычислительной нагрузки по сети и др.).

Так, последний фактор связан с распределением наиболее емких задач по вычислительной нагрузке и обмену данными на узлах СтнК. Это так называемые "горячие точки" (*hot spots*) — узлы, с которыми сетевой обмен наиболее интенсивен. "Горячая точка" — это управляющий или интерфейсный узел, через который поступают и извлекаются данные в сети. Поскольку в отличие от компьютерных сетей в СтнК вычислительные ядра располагаются компактно и обмен данными происходит на высоких частотах, распределение "горячих точек" имеет определяющее значение.

Как указывает В. Делли [2, 3], наиболее ценными из ресурсов при реализации СтнК являются физические каналы передачи, буферная память, а также коммутационная часть маршрутизаторов (роутеров). Поэтому важной характеристикой СтнК является ее топология, которая, будучи одним из основных факторов эффективности СтнК, в значительной степени определяет структуру роутеров, алгоритм маршрутизации и расходы соединительных ресурсов. В общем случае топология СтнК представляет собой неориентированный связный граф, состоящий из вершин (роутеров) и ребер (физических линий связи между ними), и фактически описывает, каким образом роутеры соединены между собой и сколько линий связи задействовано для этого [1]. В настоящее время наибольшее распространение получили регулярные топологии, такие как *mesh*, *torus*, *hypercube*, *spidergon*, *chordal ring*, *butterfly fat tree (BFT)* и т.п. [3–6]. Это обусловлено тем, что регулярные топологии имеют простую структуру роутеров, соединений между ними и заранее определенные алгоритмы маршрутизации.

Цель статьи путем использования высокоуровневого моделирования с применением модели *NoCTweak* на основе языка *SystemC* провести анализ влияния топологии и расположения "горячих точек" на эффективность СтнК.

Очевидным решением для анализа эффективности СтнК является реализация их в рамках кристалла по технологии *ASIC* или на *FPGA*-чипе. Но ввиду сложности реализаций СтнК и больших затрат на изготовление многоядерных чипов поиск оптимальных решений для СтнК с помощью прототипирования не целесообразен [1]. Для выполнения таких задач используют модели-симуляторы СтнК, которые позволяют осуществлять тестирование лучших архитектурных решений при построении СнК, и в настоящее время именно они являются неотъемлемой составляющей конструирования и оптимизации СтнК. Среди моделей СтнК выделяют аналитические и высокоуровневые/низкоуровневые имитационные модели. Каждый из приведенных подходов к моделированию СтнК характеризуется различным уровнем абстракции и, соответственно, имеет различную точность и затраты ресурсов на проведение моделирования [5].

Низкоуровневое моделирование — эмуляция работы сети на уровне логических вентилей. Компоненты модели разрабатывают с применением языков описания аппаратуры (например, *Verilog* или *VHDL*) или собирают из стандартных блоков в специализированных средах проектирования, например, *Altera Quartus II* или *Xilinx ISE*. При этом функционирование разрабатываемых компонентов модели устройства анализируется с помощью специализированных программ моделирования аппаратных средств (например, пакет *ModelSim*). Такая модель может быть синтезирована с помощью специализированных САПР. Недостатком данного подхода являются значительные затраты времени на построение модели и самомоделирование. Его целесообразно использовать для оценки эффективности отдельных решений при построении СтнК.

Высокоуровневое моделирование — это симуляция работы сети на уровне распространения потоков данных. Этот подход характеризуется скоростью разработки, гибкостью настроек и значительно меньшим временем моделирования. Данные модели делят на синтезируемые и несинтезируемые.

Синтезируемые модели обычно разрабатывают с использованием языка *SystemC* [7, 8]. Этот язык проектирования и верификации моделей системного уровня реализован в виде *C++* библиотеки с открытым кодом. Библиотека содержит ядро событийного моделирования и позволяет получить исполняемую модель устройства. Язык *SystemC* используют для построения транзакционных и поведенческих моделей, а также для высокоуровневого синтеза устройств. В *SystemC* используются понятия, подобные тем, которые присущи языкам описания аппаратуры — *VHDL* и *Verilog* — интерфейсы, процессы, сигналы, событийность и иерархия модулей. Благодаря этому *SystemC* пригоден как для поведенческого моделирования, так и для *RTL*-синтеза (на уровне регистров).

Применение языка *SystemC* позволяет уменьшить влияние недостатков и объединить преимущества низкоуровневого и высокоуровневого подходов, чем и обеспечивается его широкая популярность при создании симуляторов работы СтнК. На основе *SystemC* создана библиотека *Xpipes* [9], что дает возможность осуществлять полный цикл моделирования и синтеза СтнК [10, 11]. В работе [12] предложена высокоуровневая *SystemC*-модель *ARTS* для сравнительного моделирования сетевого и шинного методов построения систем на кристалле. С использованием *SystemC* разработаны достаточно известные симуляторы *Noxim* [13], *NIRGAM* [14] и др.

Популярность *SystemC* обусловлена также и тем, что он базируется на языке *C/C++*, однако по своей природе *C/C++* — последовательный язык, в то время как процессы в аппаратуре происходят одновременно и параллельно. Это требует освоения новой парадигмы программирования, а также специфических инструментов, таких как процессы, события, сигналы и др. [7].

Описание модели СтнК *NoCTweak*

Одним из симуляторов, основанных на *SystemC*, является *NoCTweak* [15]. Эта модель СтнК характеризуется открытым исходным кодом и предназначена для исследования производительности и энергетической эффективности сетей на кристалле, а использование связки языков *SystemC* и *C++* в *NoCTweak* дает возможность проводить моделирование с высокой скоростью на уровне циклов. Симулятор ориентирован на моделирование процесса передачи данных в подсистеме связи СтнК с топологией *mesh* и имеет большое число настраиваемых параметров.

Открытый исходный код *NoCTweak* позволяет выполнять его модификацию, оптимизацию и настройку под конкретную прикладную задачу. Так, поскольку симулятор был разработан для функционирования в *Unix*-подобных операционных средах, он был адаптирован под *MS Windows*. Были исправлены некоторые ошибки и оптимизирован исходный код *NoCTweak*, а также разработаны *bash*-скрипты для автоматического запуска нескольких прогонов симулятора и программный код в среде *MATLAB* для сбора и анализа статистики и по-

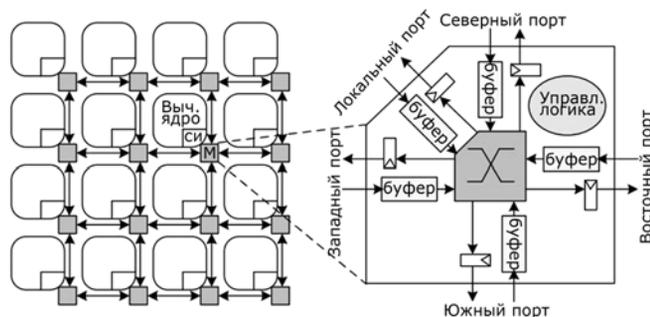


Рис. 1. Упрощенная структура подсистемы связи СтнК в *NoCTweak* [15]

строения сводных графиков изменения характеристик СтнК.

Посредством *NoCTweak* выполняется моделирование работы сети на кристалле, включающей в себя множество ядер, связанных между собой с помощью сети маршрутизаторов с топологией *mesh* (рис. 1). Каждый узел сети состоит из вычислительного узла (ядра и сетевых интерфейсов) и связанного с ним маршрутизатора. Каждый вычислительный узел генерирует пакеты данных и посылает их в сеть через маршрутизаторы. Пакеты перемещаются по сети согласно алгоритму маршрутизации и поглощаются при достижении пункта назначения (другого вычислительного узла).

Основным преимуществом симулятора *NoCTweak* по сравнению с аналогами является его достаточно простая структура и большое число конфигурируемых параметров, которые можно как задавать в конфигурационном файле перед началом симуляции, так и изменять непосредственно в исходном коде. К основным параметрам относят геометрические пропорции сети и ее размер. От того, какую форму имеет СтнК, зависят значения диаметра и среднего расстояния между узлами, ширина бисекции и т. д. Анализ влияния геометрической формы топологии на производительность СтнК является важной задачей проектирования.

NoCTweak поддерживает два типа режимов симуляции: встроенные тесты и синтетические тесты [15]. Встроенные тесты — это заранее подготовленные файлы конфигураций, отвечающих реальным объектам (например, *MPEG4*-декодер с 12 задачами или 802.11a *WiFi*-передатчик с 25 задачами). Для их моделирования достаточно передать файл конфигурации в программу и указать лишь несколько основных опций или оставить настройки по умолчанию. Другой вид тестов (синтетические тесты) предполагает ручную настройку всех параметров моделирования, среди которых основными являются тип трафика, размер сети, частота генерации флитов (*flit* — составная часть пакета данных), а также расположение "горячих точек". Как уже было сказано, "горячие точки" — это узлы с наиболее интенсивным сетевым трафиком; обычно на них приходится до 90 % сгенерированных и принятых пакетов в сети. Правильный выбор "горячих точек" во многом определяет эффективность проекции характеристического графа задачи на сеть и ее конечную производительность.

Влияние геометрической формы топологии на характеристики СтнК

Влияние геометрической формы топологии на характеристики СтнК было исследовано на примере сетей с 99 и 100 узлами. Обычно используют топологию *mesh* квадратной формы, но это возможно только для сетей с числом узлов, кратным степени натуральных чисел (например, 4, 9, 16, ...). В остальных случаях используют топологии прямоугольной

формы, что приводит к увеличению диаметра и уменьшению ширины бисекции, а если число узлов представлено простым числом (например, 5, 7, 11, ...), задача усложняется еще больше, поскольку приходится отказываться от детерминистических алгоритмов маршрутизации (главного преимущества *mesh*-топологии) или брать топологию для большего числа узлов, оставляя один из маршрутизаторов как вспомогательный и не присоединенный к вычислительному узлу [16].

Для проведения моделирования были взяты две сети на кристалле с числом узлов, равным 100: одна сеть — квадратной формы (10 × 10); другая — прямоугольной (5 × 20). Кроме того, была взята СтНК с меньшим числом узлов, равным 99, и формой 9 × 11, приближенной к квадратной. Длительность прогонов модели *NoCTweak* выбрана в 30 000 циклов моделирования, время разогрева сети (*warm-up time*) — 5000 циклов. Прогоны проводили для различной частоты генерации флитов (*fir*, флит/цикл/узел) — от 0,05 до 0,3, а режим генерации пакетов был установлен как "случайный". Полученные зависимости средней задержки прохождения пакета (L_{av} , цикл) (рис. 2, а, б) и средней пропускной способности (T_{av} , флит/цикл/узел) (рис. 2, в) от частоты генерации флитов приведены ниже.

Анализ полученных результатов моделирования (рис. 2) свидетельствует о том, что СтНК с топологией прямоугольной формы имеет значительно худшие показатели задержки прохождения пакетов и пропускной способности. Так, насыщение сети 5 × 20 происходит при $fir = 0,15$, $L_{av} = 567,3$ и достигает $T_{av} = 0,121$. При этом на графиках рис. 3, а, б видно, что и при меньшем *fir* показатель задержки прохождения пакетов сети 5 × 20 в разы больше, чем у сети 10 × 10 с топологией квадратной формы. Это происходит вследствие значительно большей удаленности узлов друг от друга — диаметр сети возрос с 18 до 23 хопов (*hop* — переход между маршрутизаторами). Для сети 10 × 10 насыщение происходит при $fir = 0,17$, когда L_{av} достигает 475,6 и в дальнейшем резко возрастает. При этом максимальное значение $T_{av} = 0,16$, что на 24,4 % больше, чем у топологии с неоптимальной прямоугольной формой.

Анализ формы графиков пропускной способности при дальнейшем увеличении частоты генерации флитов показывает, что если пропускная способность топологии 5 × 20 начиная с $fir = 0,17$ стабилизируется на уровне 0,07 и остается на нем вне зависимости от *fir*, то для топологии 10 × 10 снижение является более плавным (так, для $fir = 0,3$ $T_{av} = 0,097$, что на 24,7 % больше, чем значение $T_{av} = 0,073$ для топологии 5 × 20) и только при $fir = 0,5$ достигает уровня топологии 5 × 20. Это сви-

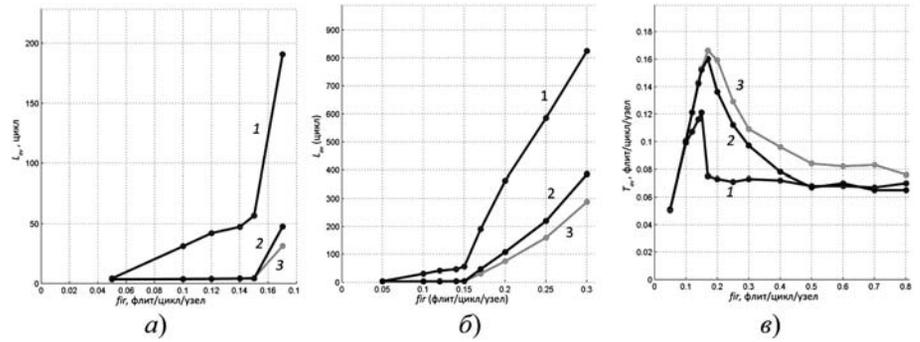


Рис. 2. Результаты моделирования СтНК топологии *mesh* в *NoCTweak*: 1 — при размерности сети 5 × 20; 2 — при размерности сети 10 × 10; 3 — при размерности сети 9 × 11

детельствует о значительно большей устойчивости подсистемы связи СтНК к повышенной нагрузке за счет применения топологии квадратной формы.

Отдельно на рис. 2 показаны графики, относящиеся к топологии 9 × 11. Их форма практически повторяет форму графиков топологии 10 × 10: пропускная способность является ненамного большей, а задержка прохождения пакетов — ненамного меньшей (при $fir = 0,17$ $L_{av} = 313,9$, а максимальная $T_{av} = 0,166$, что на 3,6 % больше, чем у топологии 10 × 10). Такие результаты объясняются тем, что СтНК с топологией 9 × 11 содержит на один узел меньше, чем СтНК с топологией 10 × 10. При этом форма топологии 9 × 11 близка к квадратной, за счет чего негативное влияние на характеристики сети не оказывается. Данные результаты согласуются с подобными, полученными при моделировании СтНК с использованием высокоуровневой модели *OCNS* в работе [17].

Влияние расположения "горячих точек" на характеристики СтНК

Как уже было отмечено, топология *mesh* является одной из наиболее распространенных в СтНК, что и стало определяющим для ее использования в модели *NoCTweak*. Собственно, это единственная топология, поддерживаемая симулятором в его основной конфигурации, хотя данное ограничение может быть снято путем расширения набора поддерживаемых топологий благодаря открытости кода симулятора. При необходимости в модель достаточно легко могут быть добавлены топологии *torus* и *hypercube*, близкие по своей структуре к топологии *mesh*.

В данном исследовании интерес представляла именно топология *mesh*. При такой топологии "горячие точки" могут располагаться или в центре сети (*central*), или по краям (*corner*). Для оценки влияния "горячих точек" на характеристики сети была проведена серия прогонов модели *NoCTweak* для сети с топологией *mesh* размером 5 × 5 (25 узлов); длительность каждого прогона моделирования составляла 100 000 циклов; время разогрева сети было установлено на 10 000 циклов. Прогоны проводились для различной частоты генерации флитов —

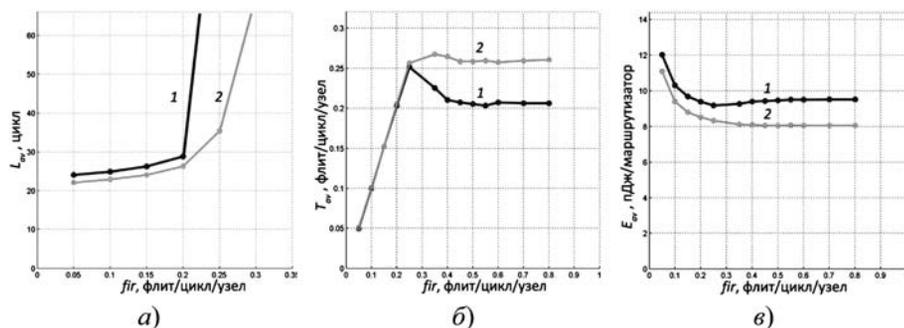


Рис. 3. Результаты моделирования СтНК для 25 узлов в *NoCTweak*. Расположение "горячих точек":
1 — по краям сети; 2 — в центре

от 0,05 до 0,8. Полученные зависимости средней задержки прохождения пакета (см. рис. 3, а), средней пропускной способности (рис. 3, б) и среднего распределения энергии на пакет (E_{av} , пДж/маршрутизатор) (рис. 3, в) от частоты генерации флитов приведены ниже.

Анализ результатов моделирования (см. рис. 3) свидетельствует о том, что при размещении "горячих точек" на периферии насыщение сети происходит при $fir = 0,2$ и $L_{av} = 28,8$ и уже при $fir = 0,25$ L_{av} возрастает в разы, достигая значения 514,0. График же средней задержки пакетов для центрального размещения "горячих точек" проходит значительно ниже. Так, при $fir = 0,2$ средняя задержка прохождения пакета $L_{av} = 26,2$ (что на 9,0 % меньше), а насыщение наступает при $fir = 0,25$ и $L_{av} = 35,3$. При дальнейшем увеличении fir происходит резкое возрастание задержки пакетов, которая при $fir = 0,35$ достигает порядка 10 000 циклов для обоих способов размещения "горячих точек". Данные наблюдения подтверждаются и графиками на рис. 3, б: сначала пропускная способность T_{av} возрастает линейно с увеличением fir , но при достижении значения 0,25 происходит насыщение, и T_{av} стабилизируется на уровне 0,257 и 0,207 (данные для $fir = 0,6$) для центрального (что на 19,5 % больше) и периферийного размещения "горячих точек" соответственно. Общую картину подтверждает и график на рис. 3, в: по достижении $fir = 0,25$ средние затраты энергии на передачу одного пакета стабилизируются и составляют 8,04 и 9,48 для центрального (что на 15,2 % меньше) и периферийного размещения "горячих точек" соответственно. Данные результаты согласуются с результатами, полученными другими авторами для трехмерных регулярных топологий [18].

Заключение

Таким образом, в результате моделирования СтНК с числом узлов, равным 100, в симуляторе *NoCTweak* была выявлена высокая чувствительность сетей с топологией *mesh* к тому, какие геометрические пропорции имеет топология сети.

Чем форма топологии ближе к квадратной, тем выше основные характеристики СтНК, что продемонстрировано на примере сетей размерности 10×10 , 9×11 и 5×20 : у первых двух топологий характеристики примерно одинаковы ввиду близости их формы к квадратной; различия в 3,6 % по пропускной способности проявляются вследствие разницы в числе их узлов на единицу. При сравнении топологий 10×10 и 5×20 различие в пропускной способности достигает 24,4...24,7 % в пользу топологии

10×10 как в момент насыщения СтНК, так в установившемся режиме.

Также с помощью моделирования СтНК с 25 узлами в симуляторе *NoCTweak* получены результаты, характеризующие влияние размещения "горячих точек" в сетях с топологией *mesh* на их производительность. Размещение задач, требующих большого объема сетевого обмена на центральных узлах, позволяет до 9 % уменьшить задержки прохождения пакетов и до 15,2 % — энергозатраты на передачу каждого пакета данных, при этом до 19,5 % увеличивая пропускную способность сети.

Перспективой дальнейших исследований является модификация симулятора *NoCTweak* для моделирования СтНК с топологиями *torus*, *hypercube* и нерегулярными топологиями для оценки влияния на них геометрической формы топологии и расположения "горячих точек".

Список литературы

1. Axel J., Hannu T. Networks on Chip. Dordrecht: Kluwer Academic Publishers, 2003. 303 p.
2. Dally W., Towles B. Principles and practices of interconnection networks. Elsevier, 2004. 550 p.
3. Dally W., Aoki H. Deadlock-free adaptive routing in Multi-computer Networks using virtual channels // IEEE Transactions on Parallel Distributed Systems. 1993. Vol. 4. N. 4. P. 466—475.
4. Balfour J., Dally W. Design Tradeoffs for Tiled CMP On-Chip Networks // ICS'06: Proceedings of the 20th Annual International Conference on Supercomputing. ACM Press, 2006. P. 187—198.
5. Saldana M., Shannon M., Chow P. The Routability of Multi-processor Network Topologies in FPGAs // Proc. of the 2006 international workshop on System-level interconnect prediction (SLIP'06). NY: ACM Press, 2006. P. 49—56.
6. Benini L., Bertozzi D. Network-on-chip architectures and design methods // Computers and Digital Techniques. IEEE. 2005. Vol. 152, N. 2. P. 261—272.
7. Романов О. Ю., Лисенко О. М. Обґрунтування вибору методу та засобів моделювання мереж на кристалі // Наукоємні технології. Збірник наукових праць. Київ: НАУ. 2014. № 1 (21). С. 49—54.
8. Genko N., Atienza D., De Micheli G., Benini L. Feature-NoC emulation: a tool and design flow for MPSoC // IEEE Circuits and Systems Magazine. 2007. Vol. 7, N. 4. P. 42—51.
9. Bertozzi D., Benini L. Xpipes: A network-on-chip architecture for gigascale systems-on-chip // IEEE Circuits and Systems Magazine. 2004. Vol. 4, N. 2. P. 18—31.
10. Murali S., De Micheli G. Bandwidth-constrained mapping of cores onto NoC architectures // Proc. of the Conference on Design,

Automation and Test in Europe, 2004 (DATE'04). Paris, 2004. Vol. 2. P. 16–20.

11. **Bertozzi D., Murali S., Jalabert A.** NoC synthesis flow for customized domain specific multiprocessor systems-on-chip // *IEEE Transactions on Parallel and Distributed Systems*. 2005. Vol. 16, N. 2. P. 113–129.

12. **Mahadevan S., Virk K., Madsen J.** ARTS: A SystemC-based framework for multi-processor Systems-on-Chip modeling // *Design Automation for Embedded Systems*. Springer, 2007. Vol. 11, N. 4. P. 285–311.

13. **Fazzino F., Palesi M., Patti D.** Noxim: Network-on-chip simulator. Режим доступа: <http://noxim.sourceforge.net/> (Доступ свободный). Загл. с экрана.

14. **Jain L.** NIRGAM: A Simulator for NoC Interconnect Routing and Application Modeling. Version 1.1. 2007. 27 p. URL: <http://nirgam.ecs.soton.ac.uk/Documentation.php>.

15. **Tran A. N.** On-Chip Network Designs for Many-Core Computational Platforms: Ph.D. thesis. USA, Davis: University of California, 2012. 156 p.

16. **Romanov O., Lysenko O.** The Comparative Analysis of the Efficiency of Regular and Pseudo-optimal Topologies of Networks-on-Chip Based on Netmaker // 1st Mediterranean Conference on Embedded Computing, MECO 2012. Advances and Challenges in Embedded Computing: Proceedings. Montenegro, Bar: IEEE, 2012. P. 13–16.

17. **Феськов Д. О., Романов О. Ю., Короткий С. В.** Программная модель сетей на кристалле с нерегулярными топологиями // Проблемы информатизации та управління. Збірник наукових праць. Київ: НАУ. 2013. № 2 (42). С. 118–123.

18. **Gebali F., Elmiligi H., Watheq M.** Networks-on-Chips: Theory and Practice. FL: CRC Press, 2009. 389 p.

A. Yu. Romanov, Senior Lecturer, e-mail: a.romanov@hse.ru,
National Research University Higher School of Economics, Moscow

The Research of Network-on-Chip with Mesh Topology by Using NoCTweak Model

This article gives an analysis of the effect of geometric shape of the topology and "hot spots" placement on the effectiveness of networks-on-chip. For this reason, a review of the main approaches to the modeling of networks-on-chip was made and the approach for networks-on-chip modeling by using SystemC NoCTweak networks-on-chip simulator was chosen. The analysis of the simulation results makes it possible to formulate the following conclusion: the high sensitivity of mesh network topology to its geometric proportions is found; the use of topologies which are close to the square form allows to increase the network capacity up to 24,7 %, and the central location of "hot spots" allows up to 9 % reduction in the delay of packets, up to 15,2 % — in energy consumption for each packet transmission and up to 19,5 % increase of the network capacity.

Keywords: network-on-chip, system-on-chip, network-on-chip regular topology, networks-on-chip design, SystemC programming language, network-on-chip model, mesh topology, "hot spots"

References

1. **Axel J., Hannu T.** *Networks on Chip*, Dordrecht, Kluwer Academic Publishers, 2003, 303 p.

2. **Dally W., Towles B.** Principles and practices of interconnection networks. Elsevier, 2004. 550 p.

3. **Dally W., Aoki H.** Deadlock-free adaptive routing in Multi-computer Networks using virtual channels, *IEEE Transactions on Parallel Distributed Systems*, 1993, vol. 4, no. 4, pp. 466–475.

4. **Balfour J., Dally W.** Design Tradeoffs for Tiled CMP On-Chip Networks, *ICS'06: Proceedings of the 20th Annual International Conference on Supercomputing*, ACM Press, 2006, pp. 187–198.

5. **Saldana M., Shannon M., Chow P.** The Routability of Multi-processor Network Topologies in FPGAs, Proceedings of the 2006 international workshop on System-level interconnect prediction (SLIP'06), NY, ACM Press, 2006, pp. 49–56.

6. **Benini L., Bertozzi D.** Network-on-chip architectures and design methods, *Computers and Digital Techniques*, IEEE, 2005, vol. 152, no. 2, pp. 261–272.

7. **Romanov O. Yu., Lysenko O. M.** Obg'runtuvannja vyboru metodu ta zasobiv modeljuvannja mrezh na krystali [Justification of the choice of methods and means of networks-on-chip modeling], *Naukojenni tehnologii, Zbirnyk naukovykh prac'*, Kyiv, NAU, 2014, no. 1 (21), pp. 49–54.

8. **Genko N., Bertozzi D., De Micheli G., Benini L.** Feature-NoC emulation: a tool and design flow for MPSoC, *IEEE Circuits and Systems Magazine*, 2007, vol. 7, no. 4, pp. 42–51.

9. **Bertozzi D., Benini L.** Xpipes: A network-on-chip architecture for gigascale systems-on-chip, *IEEE Circuits and Systems Magazine*, 2004, vol. 4, no. 2, pp. 18–31.

10. **Murali S., De Micheli G.** Bandwidth-constrained mapping of cores onto NoC architectures, *Proceedings of the Conference on De-*

sign, Automation and Test in Europe, 2004 (DATE'04), Paris, 2004, vol. 2, pp. 16–20.

11. **Bertozzi D., Murali S., Jalabert A.** NoC synthesis flow for customized domain specific multiprocessor systems-on-chip, *IEEE Transactions on Parallel and Distributed Systems*, 2005, vol. 16, no. 2, pp. 113–129.

12. **Mahadevan S., Virk K., Madsen J.** ARTS: A SystemC-based framework for multi-processor Systems-on-Chip modeling, *Design Automation for Embedded Systems*. Springer, 2007, vol. 11, no. 4, pp. 285–311.

13. **Fazzino F., Palesi M., Patti D.** Noxim: Network-on-chip simulator. URL: <http://noxim.sourceforge.net/>

14. **Jain L.** NIRGAM: A Simulator for NoC Interconnect Routing and Application Modeling, Version 1.1, 2007, 27 p., URL: <http://nirgam.ecs.soton.ac.uk/Documentation.php>.

15. **Tran A. N.** On-Chip Network Designs for Many-Core Computational Platforms: Ph. D. thesis, USA, Davis: University of California, 2012, 156 p.

16. **Romanov O., Lysenko O.** The Comparative Analysis of the Efficiency of Regular and Pseudo-optimal Topologies of Networks-on-Chip Based on Netmaker, *1st Mediterranean Conference on Embedded Computing, MECO 2012. Advances and Challenges in Embedded Computing: Proceedings. Montenegro, Bar: IEEE*, 2012, pp. 13–16.

17. **Fes'kov D. O., Romanov O. Yu., Korotkyj Je. V.** Programna model' mrezh na krystali z nereguljarnymy topologijamy [Networks-on-chip program model with irregular topologies], *Problemy informatyzacii ta upravlinnja. Zbirnyk naukovykh prac'*, Kyiv, NAU, 2013, no. 2 (42), pp. 118–123.

18. **Gebali F., Elmiligi H., Watheq M.** *Networks-on-Chips: Theory and Practice*, FL: CRC Press, 2009, 389 p.

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ DIGITAL PROCESSING OF SIGNALS AND IMAGES

УДК 519.245; 519.674

Б. Г. Кухаренко, канд. физ.-мат. наук, ст. науч. сотр., вед. науч. сотр., e-mail: kukharenkobg@gmail.com,
Институт машиноведения РАН, г. Москва,
М. О. Солнцева-Чалей, аспирант, e-mail: solnceva.chalei@gmail.com,
Московский физико-технический институт (ГУ)

Использование методов выделения ярких областей при сегментировании телеметрических изображений для идентификации групп объектов

Для ускорения идентификации групп объектов и слежения в режиме реального времени необходимо удалить неинформативную часть фона и выделить наиболее яркие области изображения. Яркие объекты выделяются сегментированием изображения по методу сдвига среднего в пространстве характеристик пикселей и построением карты яркости. Численные эксперименты показывают, что процесс обнаружения ярких объектов не может быть только низкоуровневым. Поэтому используется унифицированная модель, интегрирующая характеристики пикселей изображения низкого уровня и знание высокого уровня. Эффективность подхода демонстрируется на примере изображения с яркими объектами.

Ключевые слова: анализ изображений, сегментирование, метод сдвига среднего, карта яркости

Введение

Для сокращения времени обработки при решении задачи наблюдения и идентификации групп объектов в режиме реального времени сегментирование телеметрических изображений низкого качества посредством алгоритма сдвига среднего выполняется после удаления неинформативной части (фона) с помощью метода швов [1]. После сегментирования определение ярких (*salient*) областей позволяет быстро и точно выделить яркие объекты из окружающего фона, что способствует значительному сокращению времени обработки изображений [2–5]. Таким образом, после сегментирования изображений методы определения ярких областей выявляют группы объектов [6], обеспечивают слежение за объектами (*object tracking*) [7], редактирование изображений [8], целевой поиск в базах данных изображений (*image retrieval*) [9].

Наблюдатель выделяет особенности отдельных областей изображения в связи с решением конкретной задачи [10]. Процесс выявления ярких областей определяется стадиями зрительного восприятия: сначала выполняется быстрый обзор изображения "снизу-вверх" (*bottom-up*) ([11–13]), затем более медленное рассмотрение кадра "сверху-вниз" (*top-down*) [14, 15]. При рассмотрении изображения "снизу-вверх" визуальное восприятие области изображения связано с ее цветом, градиентом цвета и

границами этой области на фоне остального изображения [16, 17]. После сегментирования изображения этот процесс моделируется на основе методов декомпозиции матричного представления изображения на матрицу низкого ранга, представляющую фон, и разреженную матрицу, представляющую яркие области [17, 18].

Алгоритмы определения ярких областей обычно используют ограничительные контуры (*bounding boxes*) [19, 20], сегментирование бинарного переднего плана и фона (*binary foreground and background segmentation*) [15, 21] и карты яркости (*saliency maps*), определяющие принадлежность каждого пикселя к яркой области [13]. Существующие методы определения ярких областей недостаточно совершенны. Не всегда удается использовать полный спектр пространственных частот исходного изображения [22]. Низкое разрешение карты яркости приводит к неточному определению ярких областей [23]. Кроме того, в зависимости от способа определения яркой области некоторые карты могут давать недостаточно определенные границы объектов [24]. Также существуют методы, которые точно выделяют только малые яркие области, но способны выделять более крупные. В настоящей работе задача точного выделения ярких областей решается на основе методов, учитывающих спектр пространственных частот сегментированного изображения [22, 25].

1. Выделение ярких объектов на основе спектра пространственных частот сегментированного изображения

Детектор (фильтр) карты яркости для сегментированного изображения должен обеспечивать: 1) выделение наиболее крупных ярких объектов; 2) выделение ярких объектов как связанных однородных областей; 3) точное определение границ ярких объектов, не чувствительное к высоким частотам, обусловленным текстурой, шумом и артефактами; 4) создание карты яркости полного разрешения. Обозначим пороговые значения низких и высоких пространственных частот как ω_{lc} и ω_{hc} , соответственно. Для выделения крупных ярких объектов ω_{lc} должна быть достаточно низкой (первый критерий). Это также выделяет яркие объекты как однородные области (второй критерий). Для получения хорошо определенных границ ярких объектов ω_{hc} должна быть достаточно высокой (третий критерий). Но для удаления текстурных паттернов, шума, артефактов кодирования следует отбросить самые высокие пространственные частоты (четвертый критерий). Поэтому для создания карты яркости следует объединить несколько узкополосных фильтров, покрывающих требуемый интервал частот $[\omega_{lc}, \omega_{hc}]$.

Объединение разностей Гауссовых фильтров. В качестве узкополосного фильтра выбираем разность Гауссовых фильтров (*Difference of Gaussians — DoG*)

$$DoG(x, y) = G(x, y, \sigma_1) - G(x, y, \sigma_2) = \frac{1}{2\pi} \left[\frac{1}{\sigma_1^2} \exp\left(-\frac{x^2 + y^2}{2\sigma_1^2}\right) - \frac{1}{\sigma_2^2} \exp\left(-\frac{x^2 + y^2}{2\sigma_2^2}\right) \right], \quad (1)$$

поскольку она хорошо аппроксимирует широко применяемый при определении локальных особенностей изображения лапласиан Гауссового фильтра (*Laplacian of Gaussian — LoG*), когда стандартные отклонения σ_1 и σ_2 двух Гауссовых фильтров, контролирующие ширину полосы, имеют отношение 1:1,6 [26—29].

Если ввести обозначение $\sigma_1 = \rho\sigma$ и $\sigma_2 = \sigma$, то результат суммирования L узкополосных *DoG*-фильтров

$$\sum_{l=0}^{L-1} G(x, y, \rho^{l+1}\sigma) - G(x, y, \rho^l\sigma) \equiv G(x, y, \sigma\rho^L) - G(x, y, \sigma), \quad (2)$$

равно разности двух Гауссовых фильтров с отношением стандартных отклонений $P = \rho^L$, поскольку все члены за исключением первого и последнего при сложении дают ноль. Если $P \gg 1$, то применение *DoG*-фильтра (2) суммирует результат детекторов краев (фильтров) с $\rho = 1,6$ в нескольких масштабах изображения (поэтому яркие области открыты полностью).

Для полосы частот $[\omega_{lc}, \omega_{hc}]$ *DoG*-фильтра (1) с $\sigma_1 > \sigma_2$ нижняя граница ω_{lc} определяется σ_1 , а верх-

няя граница $\omega_{hc} = \sigma_2$. Однако на практике, чтобы задать большое значение отношения стандартных отклонений, σ_1 полагается бесконечно большой. Это позволяет калибровать пороговые значения частот при сохранении всех остальных частот. Чтобы устранить высокие частоты шума и текстуры используется малое Гауссово ядро. В дискретном случае малое Гауссово ядро хорошо аппроксимирует биномиальный фильтр. У биномиального фильтра с элементами $(1/6) [1, 4, 6, 4, 1]$ верхняя граница частот $\omega_{hc} = \pi/2,75$.

Определение карты яркости на основе характеристик низкого уровня для пикселей сегментированного изображения. Карта яркости $S[i, j]$ для сегментированного изображения $L[i, j]$ размерности $M \times N$ вычисляется по формуле

$$S[i, j] = |I_\mu - I_{\omega_{hc}}[i, j]|, \quad (3)$$

где I_μ — среднее значение пикселей изображения и $I_{\omega_{hc}}[i, j]$ — результат свертки изображения $L[i, j]$ с Гауссовым фильтром. Так как используется исходное изображение, карта яркости имеет полное разрешение (четвертый критерий в разд. 1). Формула (3) обобщается с учетом других характеристик низкого уровня, например интенсивности и цветов в пикселе:

$$S[i, j] = \|I_\mu - I_{\omega_{hc}}[i, j]\|_2, \quad (4)$$

где I_μ — вектор усредненных характеристик изображения, $I_{\omega_{hc}}[i, j]$ — вектор характеристик пикселя изображения в пространстве интенсивности и цветов Luv , свернутого с Гауссовым ядром (используется аппроксимация разделимым биномиальным ядром размерности 5×5), и $\|\cdot\|_2$ — Евклидова норма.

Чтобы выделить яркие объекты, на основе карты яркости сегментированного изображения строится бинарная карта, которая получается путем приписывания единиц пикселям ярких объектов и нулей остальным пикселям фона [22]. Для этого вводится адаптивный порог, который определяется как удвоенная средняя яркость $S[i, j]$ (4) сегментированного изображения:

$$T_a = \frac{2}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} S[i, j].$$

В настоящей работе для сегментирования изображения применяется алгоритм сдвига среднего (*mean-shift*), который улучшает границы объектов (в пространстве интенсивности и цветов Luv) [30]. Полный вектор характеристик пикселей изображения в формуле (4) может состоять из нескольких десятков компонент, например, дискретные координаты (i, j) пикселя на плоскости изображения, цветовые характеристики (три цветовых канала RGB или другие каналы цветов, яркость (*hue*) и насыщенность), регулируемые пирамиды (*steerable*

pyramids), дающие 12 откликов фильтров в каждом пикселе (четыре направления по трем различным масштабам), и фильтры Габора (*Gabor filters*) (12 ориентаций и три масштаба, ширина полосы наименьшего фильтра выбирается равной восьми, а фактор масштабирования — двум) (в общей сложности размерность пространства характеристик пикселей изображения равна 55). При сегментировании изображения посредством алгоритма сдвига среднего (*mean-shift* [30]) выбираются полосы пространственных частот и характеристик (*spatial and features bandwidth*), и в результате даже для визуально однородного изображения формируются многочисленные сегменты $\{C_i, i = \overline{1, K}\}$, где $K \gg 1$ — число сегментов. Пусть $I_p, i = \overline{1, K}$ — вектор характеристик центра сегмента $C_p, i = \overline{1, K}$. Поэтому процесс обнаружения ярких объектов не может быть в чистом виде низкоуровневым.

Для обнаружения ярких объектов требуется унифицированная модель, способная интегрировать характеристики изображения низкого уровня и знание высокого уровня [17].

2. Унифицированная модель карты яркости, интегрирующая характеристики низкого уровня и знания высокого уровня

Матричное представление характеристик низкого уровня сегментированного изображения для определения карты яркости. Матричное представление сегментированного изображения имеет вид

$$F = [I_1, \dots, I_K]. F \in \mathbb{R}^{D \times N}, \quad (5)$$

где D — размерность вектора характеристик (например, $D = 55$). Как отмечается во введении (см. [17]), яркие объекты выделяют на относительно однородном фоне сегментированного изображения, используя декомпозицию матричного представления F (5) сегментированного изображения

$$F = L + S, \quad (6)$$

где L — матрица низкого ранга, представляющая фон в пространстве меньшей размерности; S — разреженная матрица, представляющая яркие области. Задача оценки матриц L и S имеет вид

$$(L^*, S^*) = \underset{L, S}{\operatorname{argmin}} (\operatorname{rank}(L) + \lambda \|S\|_0) \quad (7)$$

при условии (6), где $\|\cdot\|_0$ — псевдонорма L_0 , т.е. число ненулевых компонент матрицы. Поскольку (6) — NP -сложная задача оптимизации, то в качестве альтернативы решается упрощенная выпуклая задача

$$(L^*, S^*) = \underset{L, S}{\operatorname{argmin}} (\|L\|_* + \lambda \|S\|_1), \quad (8)$$

при условии (6), где $\|L\|_* = \sum_i \sigma_i(L)$, т.е. сумма сингулярных собственных чисел, — ядерная норма (*nuclear norm*) матрицы и $\|\cdot\|_1$ обозначает L_1 -норму.

В работе [31] показано, что L и S могут быть достаточно точно оценены как решение (8) посредством устойчивого анализа главных компонент (*Robust PCA*). После того как определена матрица S , L_1 -норма столбца $s_i, i = \overline{1, K}$, в $S = [s_1, \dots, s_K]$ используется как оценка яркости сегмента $C_i, i = \overline{1, K}$.

Чем больше $\|s_i\|_1, i = \overline{1, K}$, тем большее значение яркости назначается сегменту $C_i, i = \overline{1, K}$, изображения. Карта яркости, генерируемая и нормированная таким образом, должна быть серо-шкальным изображением. Чтобы гарантировать законность модели (6), необходимо выполнить обучение линейного преобразования пространства характеристик пикселей сегментированного изображения.

Обучение линейного преобразования характеристик сегментированного изображения. Рассмотрим линейные преобразования T исходного пространства характеристик пикселей изображения

$$J_i = T I_p, \text{ где } T \in \mathbb{R}^{D \times D}. G = [J_1, \dots, J_K] = T F \quad (9)$$

и уравнение (6) принимает вид

$$T F = L + S.$$

В правильном пространстве характеристик большая часть векторов фона изображения должна лежать в подпространстве меньшей размерности, которое представляется матрицей низкого ранга. Для обнаружения такого подпространства характеристик можно использовать обучающие изображения, в которых яркие объекты отмечены вручную и выделены в прямоугольную рамку. При заданной выделяющей яркой объект рамке q_i указывает принадлежность вектора I_i в выражении (5) к яркому объекту ($q_i = 0$, если соответствующая область является яркой, $q_i = 1$ — в противном случае). Такая информация может быть представлена в виде диагональной матрицы $Q = \operatorname{diag}(q_1, \dots, q_K)$. Отличие матрицы $T F Q \in \mathbb{R}^{D \times D}$ от $T F$ состоит в том, что векторы характеристик, соответствующие ярким областям, в $T F Q$ становятся равными нулю, так как они умножаются на $q_i = 0$. Поэтому матрица $T F Q$ содержит только информацию о фоне и должна иметь низкий ранг при заданном правильном преобразовании T (9). Следовательно, задача оценки T имеет вид

$$T^* = \underset{T}{\operatorname{argmin}} \left(\frac{1}{I} \sum_{k=1}^I \|T F_k Q_k\|_* - \gamma \|T\|_* \right), \quad (10)$$

при условии $\|T\|_2 = c$.

Здесь F_k и Q_k — представление характеристик и индикатор яркости для k -го обучающего изображения соответственно (I — число обучающих изображений), c — константа. Член $-\gamma \|T\|_*$ позволяет избежать тривиального решения, когда ранг матрицы T становится малым настолько, что ранг матрицы $T F Q$ сильно превышает ранг T , и γ — весовой параметр. Ограничение $\|T\|_2 = c$ позволяет избежать того, чтобы

норма матрицы \mathbf{T} становилось слишком большой или малой.

Локально-оптимальное решение (10) обеспечивается методом спуска по градиенту. Градиент

$$O(\mathbf{T}) = \frac{1}{I} \sum_{k=1}^I \|\mathbf{T}\mathbf{F}_k\mathbf{Q}_k\|_* - \gamma\|\mathbf{T}\|_*$$

относительно \mathbf{T} имеет вид

$$\begin{aligned} \frac{\partial O(\mathbf{T})}{\partial \mathbf{T}} &= \frac{1}{I} \sum_{k=1}^I \frac{\partial \|\mathbf{T}\mathbf{F}_k\mathbf{Q}_k\|_*}{\partial \mathbf{T}} - \gamma \frac{\partial \|\mathbf{T}\|_*}{\partial \mathbf{T}} = \\ &= \frac{1}{I} \sum_{k=1}^I \frac{\partial \|\mathbf{T}\mathbf{F}_k\mathbf{Q}_k\|_*}{\partial (\mathbf{T}\mathbf{F}_k\mathbf{Q}_k)} (\mathbf{F}_k\mathbf{Q}_k)^T - \gamma \frac{\partial \|\mathbf{T}\|_*}{\partial \mathbf{T}}. \end{aligned} \quad (11)$$

Частная производная ядерной нормы произвольной матрицы \mathbf{X} может быть получена с помощью разложения сингулярных чисел (*singular value decomposition* — *SVD*) этой матрицы $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ в виде

$$\frac{\partial \|\mathbf{X}\|_*}{\partial \mathbf{X}} = \mathbf{U}\mathbf{V}^T + \mathbf{W}, \quad (12)$$

где \mathbf{W} — матрица такая, что $\mathbf{U}^T\mathbf{W} = 0$, $\mathbf{V}\mathbf{W} = 0$ и $\|\mathbf{W}\| \leq 1$. При использовании метода спуска по градиенту для (10) начальное значение $\mathbf{T} = \mathbf{I}$, где \mathbf{I} — единичная матрица. Используя (11), итерации

$\mathbf{T}_{t+1} = \mathbf{T}_t - \alpha \frac{\partial O(\mathbf{T})}{\partial \mathbf{T}}$, где α — шаг (на каждой итерации нормируется $\|\mathbf{T}_{t+1}\|_2 = c$), выполняются до достижения локального минимума, который дает оценку правильного преобразования \mathbf{T} (9).

Интегрирование априорных распределений высокого уровня в модель карты яркости. Особенности человеческого восприятия высокого уровня интегрируются в модель карты яркости. Поскольку наибольшее внимание привлекают объекты, близкие к центру изображения, при построении карты яркости используется априорное Гауссово распределение

$$p_c(\mathbf{x}) = \exp(-d(\mathbf{x}, \mathbf{c})/\sigma_1^2),$$

на основе расстояния $d(\mathbf{x}, \mathbf{c})$ от пикселей до центра изображения. Для учета особенностей восприятия цветов (например, красного и желтого) строится $2D$ -распределение гистограммы $H(S)$ в $nR-nG$ пространстве цветов

$$nR = \frac{R}{R+G+B}, \quad nG = \frac{G}{R+G+B},$$

для отмеченных ярких объектов. Аналогично строится распределение гистограммы цветов фона $H(B)$. Для каждого дискретного цвета (*quantized color*) значения, полученные из двух гистограмм $H(S)$ и $H(B)$, обозначаются как h_S и h_B . Затем устанавливается априорное распределение цветов для карты яркости

$$p_c(\mathbf{x}) = \exp((h_S(c_x) - h_B(c_x))/\sigma_3^2),$$

где c_x показывает цвет в положении \mathbf{x} . Далее эти априорные распределения перемножаются и дают окончательную априорную карту яркости.

Априорная вероятность для сегмента быть ярким определяется положением центра сегмента и обозначается p_i . Для всех сегментов изображения эта априорная вероятность может быть представлена диагональной матрицей $\mathbf{P} = \text{diag}(p_1, \dots, p_K)$, и оптимизация

$$(\mathbf{L}^*, \mathbf{S}^*) = \underset{\mathbf{L}, \mathbf{S}}{\text{argmin}} (\|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1) \quad (13)$$

выполняется при условии (6) в виде

$$\mathbf{T}\mathbf{F}\mathbf{P} = \mathbf{L} + \mathbf{S}.$$

Отметим, что в матрице \mathbf{P} большинство компонент p_i малы. Следовательно, векторы характеристик, умноженные на p_i , в матрице низкого порядка \mathbf{L} являются посторонними (*outliers*), и наиболее вероятно они будут включены в разреженную матрицу \mathbf{S} . Элементы матрицы \mathbf{S} увеличиваются при увеличении p_i так, что на карте яркости области с большими p_i имеют большую яркость. Как показывают численные эксперименты, при включении в модель (6) априорных распределений высокого уровня вероятность точного определения карты яркости при решении (13) повышается в 2 раза.

3. Численный эксперимент

Обработка изображения (рис. 1, а [32], см. четвертую сторону обложки) посредством сегментирования изображения методом сдвига среднего (*mean-shift* [30]) (параметры: $\text{sigma}S = 7$, $\text{sigma}R = 10$ и $\text{min-Region} = 20$) (рис. 1, б) и построения карты яркости сегментированного изображения (рис. 1, в) выделяет группу трех ярких объектов (рис. 1, г).

Заключение

Показано, что комбинация метода сегментирования на основе алгоритма сдвига среднего и методов выделения ярких областей, интегрирующие характеристики низкого уровня для пикселей изображения и знания высокого уровня, обеспечивают точное выделение ярких объектов в изображении.

Список литературы

1. Кухаренко Б. Г., Солнцева М. О. Использование методов сокращения фона при сегментировании телеметрических изображений для идентификации групп объектов // Информационные технологии. 2014. № 2. С. 3–8.
2. Borji A., Sihite D. N., Itti L. Salient object detection: A benchmark / Computer Vision—ECCV 2012. Lecture Notes in Computer Science. V. 7573. Berlin, Heidelberg: Springer, 2012. P. 414–429.
3. Ren T., Liu Y., Ju R., Wu G. How important is location information in saliency detection of natural images // Multimedia Tools and Applications. 2015. V. 74, Is. 20. P. 1–22.
4. Lang C., Liu G., Yu J., Yan S. Saliency detection by multitask sparsity pursuit // IEEE Transactions on Image Processing. 2012. Vol. 21, N. 3. P. 1327–1338.
5. Goferman S., Zelnik-Manor L., Tal A. Context-aware saliency detection // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2012. Vol. 34, N. 10. P. 1915–1926.

6. **Cheng M. M., Mitra N. J., Huang X., Hu S. M.** Salient-shape: Group saliency in image collections // The Visual Computer. Springer. 2014. Vol. 30, Is. 4. P. 443–453.
7. **Ren T., Qui Z., Yu T., Liu Y., Bei J.** Soft-assigned bag of features for object tracking // Multimedia Systems. 2014. Vol. 21, N. 2. P. 189–205.
8. **Cheng M. M., Mitra N. J., Huang X., Hu S. M.** RepFinder: Finding approximately repeated scene elements for image editing // ACM Transactions on Graphics (TOG). Proc. of ACM SIGGRAPH 2010. 2010. Vol. 29, N. 4. Article No. 83.
9. **Gao Y., Wang M., Zha Z. G., Shen J., Li X., Wu X.** Visual-textual joint relevance learning for tag-based social image search // IEEE Transactions on Image Processing, 2013. Vol. 22, N. 1. P. 363–376.
10. **Borji A., Itti L.** State-of-the-art in visual attention modeling // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2013. Vol. 35, N. 1. P. 185–207.
11. **Cheng M., Mitra N. J., Huang X., Torr P. H.S., Hu S.** Global contrast based salient region detection // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2015. Vol. 37, Is. 3. P. 569–582.
12. **Qiu Y., Sun X., She M. F.** Saliency detection using hierarchical manifold learning // Neurocomputing. 2015. Vol. 168. P. 538–549.
13. **Jiang P., Vasconcelos N., Peng J.** Generic promotion of diffusion-based salient object detection // Proc. of the IEEE International Conference on Computer Vision. 2015. P. 217–225.
14. **Yang J., Yang M. H.** Top-down visual saliency via joint CRF and dictionary learning // Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2012), IEEE. 2012. P. 2296–2303.
15. **Liu T., Yuan Z., Sun J., Wang J., Zheng N., Tang X., Shum H. Y.** Learning to detect a salient object // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2011. Vol. 33, Is. 2. P. 353–367.
16. **Borji A., Itti L.** State-of-the-art in visual attention modeling // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2013. Vol. 35, Is. 1. P. 185–207.
17. **Shen X., Wu Y.** A unified approach to salient object detection via low rank matrix recovery // Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2012). IEEE, 2012. P. 853–860.
18. **Yang C., Zhang L., Lu H., Ruan X., Yang M. H.** Saliency detection via graph-based manifold ranking // Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2013). IEEE, 2013. P. 3166–3173.
19. **Chang K. Y., Liu T. L., Chen H. T., Lai S. H.** Fusing generic objectness and visual saliency for salient object detection // Proc. of the IEEE International Conference on Computer Vision (ICCV'2011). IEEE, 2011. P. 914–921.
20. **Feng J., Wie Y., Tao L., Zhang C., Sun J.** Salient object detection by composition // Proc. of the IEEE International Conference on Computer Vision (ICCV'2011). IEEE. 2011. P. 1028–1035.
21. **Lu Y., Zhang W., Lu H., Xue X.** Salient object detection using concavity context // Proc. of the IEEE International Conference on Computer Vision (ICCV'2011). IEEE, 2011. P. 233–240.
22. **Achanta R., Estrada F., Wils P., Süssstrunk S.** Frequency-tuned salient region detection // Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2009). IEEE, 2009. P. 1597–1604.
23. **Hou X., Zhang L.** Saliency detection: A spectral residual approach // Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07). IEEE, 2007. P. 1–8.
24. **Itti L., Koch C.** A saliency-based search mechanism for overt and covert shifts of visual attention // Vision Research. 2000. Vol. 40, N. 10. P. 1489–1506.
25. **Achanta R., Estrada F., Wils P., Süssstrunk S.** Salient region detection and segmentation / Proceedings of the International Conference on Computer Vision Systems (ICVS '08). Springer Lecture Notes in Computer Science. V. 5008. Berlin, Heidelberg: Springer. 2008. P. 66–75.
26. **Marr D.** Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. San Francisco, CA: W. H. Freeman and Company. 1982.
27. **Low D. G.** Distinctive image features from scale-invariant keypoints // International Journal of Computer Vision. 2004. Vol. 60, N. 2. P. 91–110.
28. **Harel J., Koch C., Perona P.** Graph-based visual saliency. Advances in Neural Information Processing Systems. Vol. 19 / Ed. Schölkopf B., Platt J. C., Hoffman T. Cambridge, MA: The MIT Press. 2006. P. 545–552.
29. **Itti L., Koch C., Niebur E.** A model of saliency-based visual attention for rapid scene analysis // IEEE Transactions on Pattern Analysis & Machine Intelligence. 1998. Is. 11. P. 1254–1259.
30. **Comaniciu D., Meer P.** Mean shift: A robust approach toward feature space analysis // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002. Vol. 24, Issue 5. P. 603–619.
31. **Wright J., Ganesh A., Rao S., Peng Y., Ma Y.** Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization / Bengio Y., Schuurmans D., Lafferty J. D., eds. Advances in Neural Information Processing Systems. V. 22. Cambridge, MA: The MIT Press. 2009. P. 2080–2088.
32. **Chambers G.** Oxford's WWII Harvard Airplanes. URL: <http://www.whatravelwriterssay.com/oxfordcounty4feb09gene.html>

B. G. Kukhareno, Leading Research Scientist, e-mail: kukharenskobg@gmail.com,
Blagonravov Mechanical Engineering Research Institute of the RAS,
M. O. Solntseva-Chalei, e-mail: solnceva.chalei@gmail.com,
Post-Graduate Student, Moscow Institute of Physics and Technology (SU)

Use of Salient Region Detecting Methods after Telemetric Image Segmentation for Object Group Identification

In this paper we describe how to accelerate object group identification and tracking in real time. It is in need to remove non-informational part of image backward and to detect highly salient objects. The salient objects are detected by mean-shift segmentation in image pixel feature high dimensional space and by determining the segmented image saliency map. The frequency-tuned salient region detection method is in use, which produces segmented image full resolution saliency map displaying salient object boundaries. The method preserves the object boundaries by retaining main frequency content from the segmented image and exploits features of color and luminance. Since salient object detection is not a pure low-level process, next a unified model is in use to incorporate the image low-level features with higher-level knowledge. In the model, a special image matrix in the feature space is decomposed as a low-rank matrix representing background plus sparse matrix indicating the salient regions. To ensure the model validity, a linear transform of the image feature space has to be learned. Next, the higher-level knowledge is fused to compose a prior map, and is treated as a prior term in the objective function. The approach efficiency is demonstrated on sample image with salient objects

Keywords: image analysis, segmentation, mean-shift, salient map

References

1. **Kukharenskiy B. G., Solntseva-Chalei M. O.** Ispolzovanie metodov sokraschenia fona pri segmentirovanii telemektricheskikh izobrazheniy dlya identifikatsii grupp objectov, *Informatsionnye tehnologii*, 2014, no. 2, pp. 3–8.
2. **Borji A., Sihite D. N., Itti L.** Salient object detection: A benchmark, *Computer Vision — ECCV 2012. Lecture Notes in Computer Science*, vol. 7573, Berlin, Heidelberg: Springer, 2012. P. 414–429.
3. **Ren T., Liu Y., Ju R., Wu G.** How important is location information in saliency detection of natural images, *Multimedia Tools and Applications*, 2015, vol. 74, issue 20, pp. 1–22.
4. **Liang C., Liu G., Yu J., Yan S.** Saliency detection by multitask sparsity pursuit, *IEEE Transactions on Image Processing*, 2012, vol. 21, no. 3, pp. 1327–1338.
5. **Goferman S., Zelnik-Manor L., Tal A.** Context-aware saliency detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012, vol. 34, no. 10, pp. 1915–1926.
6. **Cheng M. M., Mitra N. J., Huang X., Hu S. M.** Salienshape: Group saliency in image collections, *The Visual Computer*, Springer, 2014, vol. 30, issue 4, pp. 443–453.
7. **Ren T., Qui Z., Yu T., Liu Y., Bei J.** Soft-assigned bag of features for object tracking, *Multimedia Systems*, 2014, vol. 21, no. 2, pp. 189–205.
8. **Cheng M. M., Mitra N. J., Huang X., Hu S. M.** RepFinder: Finding approximately repeated scene elements for image editing, *ACM Transactions on Graphics (TOG), Proceedings of ACM SIGGRAPH — 2010*, 2010, vol. 29, no. 4, article No. 83.
9. **Gao Y., Wang M., Zha Z. G., Shen J., Li X., Wu X.** Visual-textual joint relevance learning for tag-based social image search, *IEEE Transactions on Image Processing*, 2013, vol. 22, no. 1, pp. 363–376.
10. **Borji A., Itti L.** State-of-the-art in visual attention modeling, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, vol. 35, no. 1, pp. 185–207.
11. **Cheng M., Mitra N. J., Huang X., Torr P. H.S., Hu S.** Global contrast based salient region detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, vol. 37, issue 3, pp. 569–582.
12. **Qiu Y., Sun X., She M. F.** Saliency detection using hierarchical manifold learning, *Neurocomputing*, 2015, vol. 168, pp. 538–549.
13. **Jiang P., Vasconcelos N., Peng J.** Generic promotion of diffusion-based salient object detection, *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 217–225.
14. **Yang J., Yang M. H.** Top-down visual saliency via joint CRF and dictionary learning, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2012)*, IEEE, 2012, pp. 2296–2303.
15. **Liu T., Yuan Z., Sun J., Wang J., Zheng N., Tang X., Shum H. Y.** Learning to detect a salient object, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011, vol. 33, issue 2, pp. 353–367.
16. **Borji A., Itti L.** State-of-the-art in visual attention modeling, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, vol. 35, issue 1, pp. 185–207.
17. **Shen X., Wu Y.** A unified approach to salient object detection via low rank matrix recovery, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2012)*, IEEE, 2012, pp. 853–860.
18. **Yang C., Zhang L., Lu H., Ruan X., Yang M. H.** Saliency detection via graph-based manifold ranking, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2013)*, IEEE, 2013, pp. 3166–3173.
19. **Chang K. Y., Liu T. L., Chen H. T., Lai S. H.** Fusing generic objectness and visual saliency for salient object detection, *Proceedings of the IEEE International Conference on Computer Vision (ICCV'2011)*, IEEE, 2011, pp. 914–921.
20. **Feng J., Wic Y., Tao L., Zhang C., Sun J.** Salient object detection by composition, *Proceedings of the IEEE International Conference on Computer Vision (ICCV'2011)*, IEEE, 2011, pp. 1028–1035.
21. **Lu Y., Zhang W., Lu H., Xue X.** Salient object detection using concavity context, *Proceedings of the IEEE International Conference on Computer Vision (ICCV'2011)*, IEEE, 2011, pp. 233–240.
22. **Achanta R., Estrada F., Wils P., Süsstrunk S.** Frequency-tuned salient region detection, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2009)*, IEEE, 2009, pp. 1597–1604.
23. **Hou X., Zhang L.** Saliency detection: A spectral residual approach, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*, IEEE, 2007, pp. 1–8.
24. **Itti L., Koch C.** A saliency-based search mechanism for overt and covert shifts of visual attention, *Vision Research*, 2000, vol. 40, no. 10, pp. 1489–1506.
25. **Achanta R., Estrada F., Wils P., Süsstrunk S.** Salient region detection and segmentation, *Proceedings of the International Conference on Computer Vision Systems (ICVS '08)*. Springer Lecture Notes in Computer Science, V. 5008. Berlin, Heidelberg: Springer. 2008. P. 66–75.
26. **Marr D.** *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, San Francisco, CA: W. H. Freeman and Company, 1982.
27. **Lowe D. G.** Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, 2004, vol. 60, no. 2, pp. 91–110.
28. **Harel J., Koch C., Perona P.** Graph-based visual saliency / Schölkopf B., Platt J. C., Hoffman T., *Advances in Neural Information Processing Systems*, vol. 19, Cambridge, MA: The MIT Press, 2006, pp. 545–552.
29. **Itti L., Koch C., Niebur E.** A model of saliency-based visual attention for rapid scene analysis, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 1998, issue 11, pp. 1254–1259.
30. **Comaniciu D., Meer P.** Mean shift: A robust approach toward feature space analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002, vol. 24, issue 5, pp. 603–619.
31. **Wright J., Ganesh A., Rao S., Peng Y., Ma Y.** Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization, Bengio Y., Schuurmans D., Lafferty J. D., eds., *Advances in Neural Information Processing Systems*, vol. 22, Cambridge, MA: The MIT Press, 2009, pp. 2080–2088.
32. **Chambers G.** Oxford's WWII Harvard Airplanes, URL: <http://www.whatstravelwriterssay.com/oxfordcounty4feb09gene.html>

А. М. Гиацинтов, науч. сотр., e-mail: algts@inbox.ru,
 К. А. Мамросенко, канд. техн. наук, зав. отделом, e-mail: kirillam@ya.ru
 Центр визуализации и спутниковых информационных технологий,
 Научно-исследовательский институт системных исследований РАН

Методы отображения трехмерных объектов при применении отложенной визуализации

Описан метод отложенной визуализации, применяемый на современных аппаратных платформах. Определены этапы формирования изображения, необходимые при отложенной визуализации. Рассмотрен метод устранения артефактов итогового изображения (сглаживания), основанный на поиске граней и реализуемый в качестве этапа постобработки. Представлен разработанный метод отображения полупрозрачных объектов при применении отложенной визуализации.

Ключевые слова: тренажерно-обучающая система, подсистема визуализации, тренажер, рендеринг, отложенная визуализация, освещение, прозрачность

Введение

Тренажерно-обучающая система (далее ТОС) оператора сложной технической системы — техническое средство для подготовки операторов сложных технических систем, отвечающее требованиям методик подготовки, обеспечивающее получение знаний, навыков и умений, реализующее модель таких систем и осуществляющее контроль над действиями обучаемого, а также используемое для исследований. ТОС могут применять для формирования индивидуальных профессиональных навыков и умений, а также для отработки групповых операций [1].

Подсистема визуализации ТОС обеспечивает отображение результатов моделирования внешней среды и объекта управления с помощью устройств отображения информации. Подсистема должна обеспечивать воспроизведение визуальной картины с достаточно подробным содержанием, позволяющим операторам ТОС успешно выполнять поставленные задачи. Важную роль в достижении реалистичности отображаемой сцены играет освещение.

Зачастую для освещения трехмерной сцены требуется несколько источников света. При традиционном подходе (*forward rendering*) этапы прорисовки геометрических объектов и их освещения совмещены. Упрощенно прорисовка объектов осуществляется следующим образом: информация о геометриче-

ских объектах (вершинах, нормалях, текстурных координатах) передается сначала в вершинный шейдер, затем, при необходимости, в геометрический шейдер. Далее происходит растеризация геометрических объектов. Полученные при растеризации данные обрабатываются пиксельными шейдерами и выводятся в кадровый буфер. Схема прорисовки геометрии при традиционном подходе представлена на рис. 1. Расчет освещения определяет влияние каждого источника света в сцене на геометрические объекты и проводится для каждой вершины в трехмерной сцене и каждого пикселя конечного изображения, выводимого на экран. Для динамических источников света и динамической геометрии сцены требуются расчеты пересечения источников света и геометрических объектов, например, с помощью ограничивающих сфер, кубов [2]. Кроме того, расчеты будут проведены даже для скрытых и перекрывающихся поверхностей, а также для тех пикселей, которые могут не попасть в итоговое изображение. Соответственно, при большом числе источников света производительность подсистемы визуализации может быть существенно снижена.

Метод отложенной визуализации

Одним из решений данной проблемы является применение методов так называемой отложенной визуализации, в частности, отложенного освещения. Первый метод был разработан Майклом Дирингом и его коллегами в 1988 г. В своей работе [3] авторы предложили систему, в которой процессоры обработки треугольников растеризируют геометрию, а затем шейдерные процессоры применяют затенение по Фонгу к обрабатываемым геометрическим объектам с использованием многочисленных источников света. Существенный вклад в разработку технологии отложенной визуализации внесла ра-

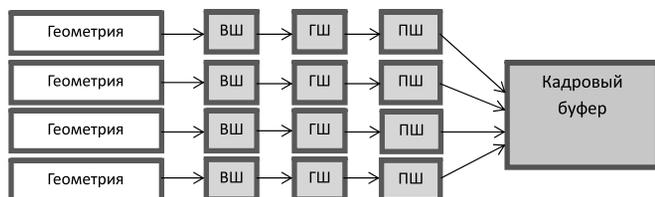


Рис. 1. Процесс прорисовки изображения с помощью традиционного метода визуализации; ВШ — вершинный шейдер, ГШ — геометрический шейдер, ПШ — пиксельный шейдер

бота Сайто и Такахаша в 1990 г. [4]. В своей статье авторы предлагают способ визуализации 3D-изображений, основанный на распознавании геометрических фигур и их последовательностей. Для оптимизации процесса геометрические свойства поверхностей сохраняются в геометрических буферах (G-Buffers). Применение геометрических буферов для хранения промежуточных результатов позволяет отделить этап распознавания фигур от этапов геометрической обработки (проецирования, удаления невидимых граней, затенения, применения текстур), и использовать его для постобработки.

Рассмотренные методы отложенной визуализации требовали применения узкоспециализированных аппаратных модулей. В настоящее время методы разрабатываются с учетом применения вычислительных мощностей центральных процессоров и графических адаптеров.

Рассмотрим подробнее базовый метод отложенной визуализации, применяемый на современных аппаратных платформах. Его основной идеей является отделение этапа геометрической обработки, во многом схожего с этапом прорисовки геометрии в традиционном подходе, от этапа освещения трехмерной сцены.

Прорисовка изображения осуществляется в несколько этапов. Все геометрические объекты сцены прорисовываются один раз, при этом информация о цвете, нормалях и глубине прорисовки для каждого пикселя сохраняется в промежуточный G-Buffer. Далее, с использованием сохраненной информации рассчитывается освещение. Затем итоговое изображение копируется в кадровый буфер (рис. 2). Всего этапов четыре: этап геометрической обработки, этап освещения, этап постобработки и заключительный этап [5].

Каждый этап использует функционал программируемого графического конвейера, применяя вершинные и пиксельные шейдеры. Все расчеты, проводимые на каждом этапе, сохраняются в памяти видеокарты в виде текстур [6].

Этап геометрической обработки является единственным этапом, который работает с данными трехмерных объектов. Входной информацией для данного этапа является трехмерная сетка обрабатываемого объекта, а выходом — изображение в G-Buffer. Для заполнения G-Buffer информацией буфер необходимо сделать активным. Первой обрабатывается информация о вершинах, текстурных координатах и материалах — она записывается в текстуру глубины G-Buffer в памяти видеокарты. Далее геометрические данные и информация о материалах обрабатываются пиксельными шейдерами. Результат их работы загружается в остальные текстуры буфера — текстуры нормалей, цвета и отражений.

В текстуру глубины записывается информация глубины для каждого пикселя итогового изображения. Значение глубины находится в отрезке 0—1, где 0 — ближняя граница прорисовки виртуальной камеры, 1 — дальняя граница прорисовки. Если один трехмерный объект перекрывает другой, то в текстуру глубины записывается значение глубины, соответствующее ближайшему объекту. Во время этапа освещения возможно восстановить информацию о позиции объекта, зная значение глубины и направление виртуальной камеры.

Далее проводится заполнение текстур нормалей, цвета и отражений. Каждый объект трехмерной сцены прорисовывается с учетом заданного материала, в котором указываются используемые вершинные и пиксельные шейдеры, определяющие алгоритм обработки данного объекта.

После заполнения всех текстур G-Buffer (нормалей, цвета, отражения и глубины) начинается этап освещения. Сначала делается активной текстура цвета, для которой рассчитывается влияние рассеянного света. Затем для каждого источника света в трехмерной сцене определяется ограничивающий прямоугольник, в рамках которого учитывается влияние источника света. Прямоугольник рассчитывается в экранных координатах, его размер равен разрешению экрана. Благодаря этому достигается полное отделение этапа освещения от этапа геометрической обработки, что приводит к значительному повышению производительности при большом числе (более 8) источников света.

При использовании метода отложенной визуализации нет необходимости знать, какие геометрические объекты будут освещены каким источником света, при этом возможна обработка всех источников света, которые освещают видимые наблюдателем объекты. Для уменьшения числа обрабатываемых источников света могут быть применены различные методы оптимизации, например, могут быть исключены из обработки источники света, которые закрыты другими объектами.

Как и при визуализации традиционным способом, существует несколько вариантов расчета

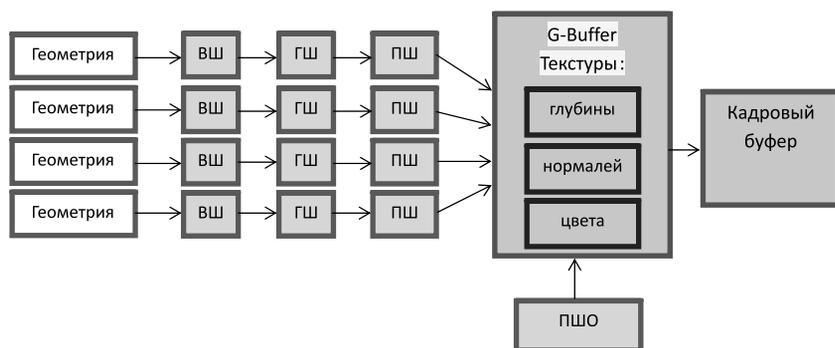


Рис. 2. Процесс прорисовки изображения с помощью отложенного метода визуализации; ВШ — вершинный шейдер, ГШ — геометрический шейдер, ПШ — пиксельный шейдер без процедур расчета освещения, ПШО — пиксельный шейдер, рассчитывающий освещение пикселя

влияния источников света на трехмерную сцену, например, последовательная обработка каждого источника света (более простой вариант в реализации, но более затратный по производительности) или же обработка сразу множества источников света за один проход. Рассмотрим процесс последовательной обработки источников света. Описанные ниже действия также могут быть использованы при создании более сложных шейдеров, реализующих обработку нескольких источников света за один проход.

Первым шагом обработки является активация созданных ранее текстур глубины, цвета, нормалей и отражений и передача параметров об источнике света (позиция, цвет, радиус) в пиксельный шейдер. Далее необходимо рассчитать влияние рассеянного света на пиксели итогового изображения (*ambient pass*). Те пиксели, на которые не оказывают влияния источники света, будут прорисованы только с учетом рассеянного освещения, для остальных пикселей рассчитывается рассеянное освещение с учетом влияния источников света. Достигается это путем прорисовки полноэкранного прямоугольника. По сути, данный этап осуществляет "заливку" прорисовываемых трехмерных объектов выбранным цветом. Если у объектов текстуры присутствуют, то происходит смешивание цветов.

При последовательной обработке источников света для каждого источника будет прорисован полноэкранный прямоугольник, определяющий его влияние на трехмерные объекты. Возможно ввести ограничение размера прорисовываемого прямоугольника. Так как при прорисовке полноэкранного прямоугольника каждый пиксель итогового изображения будет обработан шейдером освещения, то ограничение его размера, т.е. отбрасывание всех пикселей, на которые обрабатываемый источник света не оказывает влияния, позволит существенно сократить вычислительную нагрузку.

Например, имеется трехмерная сцена с тремя источниками света. Если размер итогового изображения составляет 512×512 (262 144) пикселей, то без оптимизации для обработки трех источников света шейдер обработки будет вызван 786 432 раза. Применение оптимизации позволяет сократить число обрабатываемых пикселей для каждого источника света приблизительно на 50 % в зависимости от расположения источников света и их радиуса.

Расчет ограничивающего прямоугольника выполняется на центральном процессоре перед прорисовкой каждого источника света. Одним из возможных методов расчета прямоугольника является проецирование восьми вершин ограничивающего куба источника света (AABB — Axis-Aligned Bounding Box) на экранную плоскость и нахождение ограничивающего 2D-прямоугольника из этих точек.

Следующим шагом расчета освещения является расчет влияния источников света (*illumination pass*). Для этого при обработке каждого источника света

прорисовывается прямоугольник с размером, совпадающим с размером итогового изображения. Обработка начинается с поиска позиции текущего пикселя в видовой системе координат. Значения ближней и дальней границ прорисовки, вектор направления виртуальной камеры и сохраненное нормализованное значение глубины для конкретного пикселя передаются в шейдерную программу, написанную на языке GLSL (OpenGL Shading Language) для расчета положения пикселя. Чтобы вектор направления виртуальной камеры был доступен в шейдере для каждого пикселя изображения, его необходимо передавать как текстурные координаты для каждой из четырех вершин прорисовываемого прямоугольника. Вектор будет линейно интерполирован при обработке пикселей прорисовываемого прямоугольника. Для компенсации ошибок линейной интерполяции на этапе растеризации в шейдере необходимо проводить нормализацию вектора.

Первым этапом расчета положения пикселя является определение компоненты z из нормализованного значения глубины, сохраненного в текстуре глубины. Для этого в шейдере выполняются следующие расчеты:

$$\text{float pos.z} = \text{far_plane} * \text{near_plane} / (\text{d} * (\text{far_plane} - \text{near_plane}) - \text{far_plane}),$$

где pos.z — значение компоненты z пикселя в системе координат виртуальной камеры; near_plane — значение ближней границы прорисовки; far_plane — значение дальней границы прорисовки; d — значение глубины для текущего пикселя.

После расчета координаты z выполняется расчет X и Y координат позиции пикселя:

$$\text{pos.xy} = \text{view.xy} / \text{view.z} * \text{pos.z},$$

где view.xy , view.z — компоненты вектора направления виртуальной камеры, pos.xy , pos.z — позиции пикселя в координатах виртуальной камеры.

Теперь позиция пикселя получена и после активации текстур цвета, отражений и нормалей возможно применение освещения по Фонгу к пикселю. Цвет пикселя также учитывает освещение от других источников света в трехмерной сцене путем аддитивного смешивания.

После окончания этапа освещения имеется готовое изображение, которое можно отобразить на экране. Но при необходимости также можно применить к изображению такие эффекты постобработки, как сглаживание (*antialiasing*) и засветка (*bloom*).

Так как одним из недостатков отложенной визуализации является отсутствие поддержки аппаратного сглаживания изображения (MSAA — Multi-sample Anti-Aliasing) при прорисовке в основной кадровый буфер, то сглаживание реализуется в качестве этапа постобработки (фильтра). Одним из вариантов реализации сглаживания как этапа постобработки является поиск граней в изображении. Для этого используется шейдер, который будет при-

менять размытие к пикселям итогового изображения, если соответствующие значения нормали и/или глубины скачкообразно изменяются. Фильтр сглаживания складывает квадраты расстояний от текущего пикселя до соседних пикселей для расчета коэффициента размытия пикселей. Осуществляется обработка восьми соседних пикселей вокруг целевого пикселя для поиска скачкообразных изменений по всем направлениям.

Формула расчета коэффициента размытия пикселя:

$$K(u, v) = \sum_{i=1}^8 (p_{(u + \text{сдвиг}[i].x)(v + \text{сдвиг}[i].y)} - p_{uv})^2, \quad (1)$$

где p_{uv} — целевой пиксель с координатами (u, v) , сдвиг $[i]$ — таблица сдвигов для изменения позиции соседних пикселей вокруг целевого пикселя.

После определения коэффициента размытия пикселя (с помощью текстур нормалей и/или глубины) рассчитывается окончательный цвет пикселя. Так, в тех участках изображения, где грани не были обнаружены, сдвиг будет равен нулю, т.е. размытие проводиться не будет. В случае обнаружения грани к цвету пикселя добавляются цвета соседних пикселей в соответствии с рассчитанным коэффициентом размытия.

Отображение полупрозрачных объектов при применении отложенной визуализации

Рассмотренный метод отложенной визуализации не предусматривает наличия в сцене полупрозрачных объектов. Это обусловлено тем, что при прорисовке геометрии применяется так называемый тест видимости для исключения объектов, которые перекрываются другими объектами, для повышения скорости визуализации. Однако из-за аппаратных ограничений графических адаптеров для каждого пикселя за один проход теста видимости может быть сохранена только информация о ближайшем объекте, что усложняет процесс визуализации — для правильной прорисовки полупрозрачных объектов требуется информация о видимости каждого объекта, перекрываемого полупрозрачным объектом. Из существующих способов расчета видимости пикселей наиболее часто применяются зависимый и независимый от порядка прорисовки способы. Для реализации второго способа требуется, чтобы все прорисовываемые объекты были отсортированы от дальнего к ближнему, что позволяет получить результат без искажений [7].

Одним из методов прорисовки полупрозрачных объектов при отложенной визуализации является метод создания слоев прозрачности (*depth peeling*) [8]. Он позволяет отображать полупрозрачные объекты независимо от порядка прорисовки, последовательно обрабатывая слои полупрозрачных пикселей с помощью теста глубины [9].

По сравнению с другими методами прорисовки полупрозрачных объектов, например, отложенной визуализации с разделением изображения на секции (*tiled deferred rendering*) *depth peeling* не требует применения вычислительных шейдеров (*compute shader*) для своей работы. Соответственно, метод отложенной визуализации с использованием *depth peeling* может быть реализован на графических картах уровня DirectX 10 (OpenGL 3.3). По сравнению с методом отложенной визуализации с индексированием источников света (*Light Indexed Deferred Rendering*) *depth peeling* не требует предварительной сортировки источников света и не имеет проблем с пересекающимися источниками света. Однако метод *depth peeling* может требовать большего количества видеопамати для хранения промежуточных результатов визуализации.

Разработанный метод прорисовки полупрозрачных поверхностей основывается на описанном выше методе отложенной визуализации.

Во время инициализации программируемого конвейера вместе с основным G-Buffer создается так называемый буфер прозрачности, который идентичен G-Buffer. Так же, как и в базовом алгоритме отложенной визуализации, расчет итогового изображения разделен на несколько этапов: прорисовка геометрических объектов, освещение, прорисовка прозрачных объектов, вывод на экран.

На этапе геометрической обработки в G-Buffer помещаются все данные о геометрии трехмерной сцены, за исключением данных о полупрозрачных объектах. Далее на этапе освещения выполняются переключение на буфер прозрачности, а также очистка его текстур глубины и цвета. Текстура глубины из G-Buffer копируется в текстуру глубины буфера прозрачности шейдером копирования данных. Для этого требуется прорисовка полноэкранный прямоугольника поверх буфера отрисовки. Затем происходит отсоединение текстуры глубины и присоединение текстур цвета, нормалей, отражений из G-Buffer. Присоединенные текстуры прорисовываются в соответствующие текстуры буфера прозрачности шейдером обработки за счет прорисовки полноэкранный прямоугольника. После этого осуществляются освещение прорисованных объектов и отсоединение всех буферов.

Следующим этапом является прорисовка и освещение полупрозрачных объектов.

Последним этапом является перемещение информации из буфера прозрачности в кадровый буфер. Проводится переключение на кадровый буфер и его очистка. Далее цветовая текстура буфера прозрачности копируется в кадровый буфер с помощью шейдера копирования. После этого изображение в кадровом буфере считается завершенным и может быть отображено на экране.

Работа выполняется в рамках проекта № 2.9 программы фундаментальных исследований ОНИТ РАН.

Использование методов отложенной визуализации позволяет сократить объем вычислений при наличии более восьми источников света по сравнению с традиционными методами визуализации за счет исключения из обработки всех пикселей, которые не попадут в итоговое изображение. Одну из основных проблем отложенной визуализации — отсутствие возможности обработки полупрозрачных объектов, решает разработанный авторами метод. Преимуществами метода являются применимость вне зависимости от порядка прорисовки объектов и относительная простота реализации. Недостатком является значительный объем используемой видеопамати, который увеличивается при повышении разрешения итогового изображения. Несмотря на указанный недостаток разработанный метод применим для визуализации высокореалистичных трехмерных сцен в тренажерно-обучающих системах.

1. **Гиацинтов А. М., Мамросенко К. А., Решетников В. Н.** Инструментальные средства предтренажерной и тренажерной подготовки операторов сложных технических систем // Программные продукты системы и алгоритмы. 2014. № 1. URL: <http://swsys-web.ru/simulator-training-operators.html>.
2. **Решетников В. Н.** Тригонометрические воспоминания. Серия Космические телекоммуникации. Санкт-Петербург: ООО "РИП СПб", 2015. 138 с.
3. **Deering M., Scediwy B., Duffy C.** The triangle processor and normal vector shader: a VLSI system for high performance graphics // SIGGRAPH'88 Proceedings of the 15th annual conference on Computer graphics and interactive techniques., 1988. P. 21—30.
4. **Saito T., Takahashi T.** Comprehensible Rendering of 3-D Shapes // Computer Graphics. 1990. N. 4. P. 197—206.
5. **Alamia M.** Coding Labs. Simple OpenGL Deferred Rendering Tutorial [Электронный ресурс]. URL: http://www.codinglabs.net/tutorial_simple_def_rendering.aspx (дата обращения: 08.07.2015).
6. **Fonseca F.** Deferred Shading Tutorial [Электронный ресурс]. URL: <http://gamedevs.org/uploads/deferred-shading-tutorial.pdf>.
7. **Боресков А.** steps3D — Tutorials — Deferred Shading. [Электронный ресурс]. URL: <http://steps3d.narod.ru/tutorials/ds-tutorial.html> (дата обращения: 08.07.2015).
8. **Persson E.** Deep deferred shading [Электронный ресурс]. URL: <http://www.humus.name/index.php?page=3D&ID=75> (дата обращения: 08.07.2015).
9. **Everitt C.** Interactive Order-Independent Transparency [Электронный ресурс]. URL: http://www.eng.utah.edu/~cs5610/handouts/order_independent_transparency.pdf (дата обращения: 18.02.2016).

A. M. Giatsintov, Researcher, e-mail: algts@inbox.ru,

K. A. Mamrosenko, Head of Department, e-mail: kirillam@ya.ru,

The center of visualization and satellite information technologies, SRISA RAS

Methods of 3D Objects Visualization with Deferred Rendering

Article depicts a method of deferred rendering utilized on modern hardware platforms. The use of deferred rendering methods can significantly reduce the amount of required calculations when there are more than eight lights in 3d scene. In contrast with traditional forward rendering methods, deferred rendering also discards pixels that would not complement to the final image. The phases of image drawing, required with the use of deferred rendering, are determined. A method of antialiasing, based on edge detection and implemented as a post-process effect, is described. Article presents a developed method of visualizing semi-transparent 3d objects with deferred rendering.

Keywords: training simulation system, visualization subsystem, rendering, trainer, lighting, transparency, deferred rendering

References

1. **Giatsintov A. M., Mamrosenko K. A., Reshetnikov V. N.** Instrumental'nyye sredstva predtrenazhernoy i trenazhernoy podgotovki operatorov slozhnykh tekhnicheskikh sistem, *Programmnye produkty, sistemy i algoritmy*, 2014, no. 1 [Web resource]. URL: <http://swsys-web.ru/simulator-training-operators.html>.
2. **Reshetnikov V. N.** *Trigonometricheskie vospominaniya. Kosmicheskie telekommunikatsii*. SPb: RIP SPB, 2015. 138 p.
3. **Deering M., Scediwy B., Duffy C.** The triangle processor and normal vector shader: a VLSI system for high performance graphics, *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, New York, USA, 1988, pp. 21—30.
4. **Saito T., Takahashi T.** Comprehensible Rendering of 3-D Shapes, *Computer Graphics*, 1990, no. 4, pp. 197—206.
5. **Alamia M.** Coding Labs. Simple OpenGL Deferred Rendering Tutorial [Web resource]. URL: http://www.codinglabs.net/tutorial_simple_def_rendering.aspx
6. **Fonseca F.** *Deferred Shading Tutorial*. [Web resource]. URL: <http://gamedevs.org/uploads/deferred-shading-tutorial.pdf>.
7. **Boreskov A.** *Deferred Shading*. [Web resource]. URL: <http://steps3d.narod.ru/tutorials/ds-tutorial.html>
8. **Persson E.** *Deep deferred shading*. [Web resource]. URL: <http://www.humus.name/index.php?page=3D&ID=75>
9. **Everitt C.** *Interactive Order-Independent Transparency* [Web resource]. URL: http://www.eng.utah.edu/~cs5610/handouts/order_independent_transparency.pdf

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ CAD-SYSTEMS

УДК 004.272.2

А. Л. Стемпковский, академик РАН, директор, Д. В. Тельпухов, канд. техн. наук, зав. отд.,
Р. А. Соловьев, канд. техн. наук, вед. науч. сотр., e-mail: ZF-Turbo@yandex.ru
Институт проблем проектирования в микроэлектронике РАН (ИППМ РАН)

Повышение сбоеустойчивости логических схем на основе частичного ресинтеза схемы

Ряд разнородных факторов из области микроэлектронной промышленности в последнее время существенно актуализировал разработки в области повышения сбоеустойчивости комбинационных схем. В настоящее время в современных системах автоматизированного проектирования отсутствуют средства для оценки или повышения сбоеустойчивости комбинационных схем на логическом уровне. В данной работе была сделана попытка создания методологии для реализации подобных программных средств. Предложен маршрут ресинтеза, позволяющий без внедрения существенной избыточности повысить логическую устойчивость комбинационных схем к одиночным сбоям. Были проведены вычислительные эксперименты, демонстрирующие высокую эффективность предложенного метода.

Ключевые слова: комбинационные схемы, наблюдаемость вентиля, коэффициент чувствительности, ресинтез

Введение

Современные тенденции, заключающиеся в увеличении тактовых частот, росте степени интеграции, уменьшении порогового напряжения и напряжения питания, приводят к экспоненциальному росту одиночных сбоев в комбинационных схемах, обуславливая острую необходимость в обеспечении сбоеустойчивости не только модулей памяти, но и комбинационных схем. Для процессоров и иных микроэлектронных устройств, где последовательные элементы уже имеют встроенные механизмы защиты, комбинационные схемы становятся одним из основных источников одиночных сбоев [1].

Существующие методы повышения отказоустойчивости комбинационных схем эксплуатируют одно из маскирующих свойств логических схем: логическое, электрическое, временное.

- **Логическое маскирование:** сбой возникает в участках схемы, не оказывающих влияния на выход схемы в условиях заданных входных воздействий.
- **Электрическое маскирование:** сбой затухает и не распространяется до выхода схемы благодаря передаточным характеристикам электрических цепей логических вентилях.
- **Временное маскирование:** сбой не проявляется ввиду того, что его длительность не позволяет зафиксировать ошибку на выходном регистре.

В настоящем исследовании учитывалось лишь логическое маскирование. Это обусловлено тем, что данное свойство не зависит от технологического базиса и вносит наибольший вклад в общую интен-

сивность сбоев схемы [2]. Кроме того, логическое маскирование является наиболее трудоемким свойством для моделирования и характеристики [3].

В настоящее время за основу чаще всего берется модель независимых вентиляльных сбоев в трактовке Фон Неймана [4], в рамках которой считается, что все вентили имеют одинаковую независимую вероятность сбоя, в то время как число ошибок в схеме не ограничено. В рамках такой постановки задачи было разработано большое число различных мажоритарных подходов, защищающих схемы на архитектурном уровне, таких как каскадное тройное резервирование (*cascade triple modular redundancy (CTMR)*), логические схемы с четырехкратным резервированием (*quadded logic*), случайно-переплетенная логика (*random interwoven redundancy*) и т.д. Однако несмотря на большое количество научных публикаций, на практике для защиты комбинационных схем до сих пор используют архаичные методы тройного резервирования.

В настоящей работе для повышения сбоеустойчивости комбинационной схемы предлагается использовать метод ресинтеза, который подразумевает итерационное замещение уязвимых участков схем на более устойчивые аналоги.

Методы оценки логической устойчивости схемы к одиночным сбоям

Дестабилизирующие воздействия и помехи на выходе логических вентилях моделируют как двоичный симметричный канал связи. Это означает, что

каждый вентиль имеет постоянную независимую вероятность сбоя $p < 0,5$. Эта вероятность интерпретируется как совокупность различных дестабилизирующих эффектов и источников помех, включая перекрестные помехи, наземное космическое излучение, электромагнитные помехи и т.д. В общем случае свойства логического маскирования полностью определяются функцией вероятности ошибки на выходе схемы $F(p)$ от вероятности сбоя вентиля p [5].

Введем некоторые обозначения. Пусть Ω обозначает набор всех вентилях в схеме, в то время как N и M — число первичных входов и число вентилях соответственно. Будем обозначать ошибки, возникающие на элементах как e_i , причем $e_i = 1$, если на i -м элементе возникла ошибка. В случае отсутствия ошибки $e_i = 0$. Вектор $\mathbf{e} = (e_1, e_2, \dots, e_M)$ будем называть вектором ошибки, по аналогии с вектором входных значений $\mathbf{X} = (x_1, x_2, \dots, x_N)$.

В качестве примера рассмотрим схему в базисе элементов NAND, реализующую функцию прямой импликации (рис. 1), и на основе его расширенной таблицы истинности получим основные формулы для нахождения $F(p)$. Построим расширенную таблицу истинности (табл. 1) для всех комбинаций входных векторов и векторов ошибок (\mathbf{X}, \mathbf{e}).

Здесь $E(\mathbf{X}, \mathbf{e})$ обозначает характеристическую функцию от пар векторов (входной вектор \mathbf{X} и вектор ошибки \mathbf{e}):

$$E(\mathbf{X}, \mathbf{e}) = \begin{cases} 1, & \text{если набор } (\mathbf{X}, \mathbf{e}) \text{ приводит к ошибке} \\ 0, & \text{иначе.} \end{cases}$$

В последнем столбце указаны вероятности появления вектора ошибки при условии того, что вероятность возникновения ошибки на каждом вентиле равна p . Например, вероятность появления вектора ошибки $\mathbf{e} = (1, 0)$ равна $p(1 - p)$, вследствие

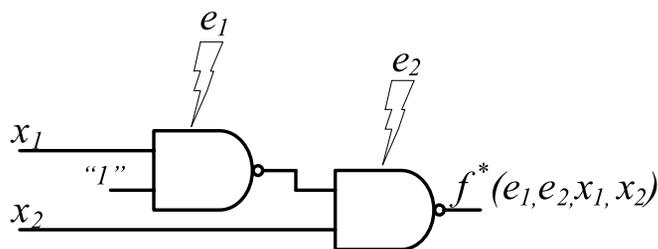


Рис. 1. Прямая импликация в базисе {NAND}

независимости двух случайных событий: возникновение ошибки на первом вентиле (p) и отсутствие ошибки на втором вентиле ($1 - p$).

Нас интересуют строки, в которых значение функции не совпадает с эталонным, а иными словами — строки, в которых характеристическая функция $E(\mathbf{X}, \mathbf{e})$ равна единице. Обратившись к строчке под номером 5 в табл. 1 и учитывая тот факт, что вероятность любой комбинации двух входов равна $\frac{1}{4}$, получим вероятность наступления этого

события: $\frac{1}{4}(1 - p)p$. Если для каждой строчки, в которой $E(\mathbf{X}, \mathbf{e}) = 1$ найти вероятность ее возникновения, то просуммировав все эти выражения, получим полином ошибки, который характеризует вероятность несовпадения результата работы схемы с эталонным при вероятности ошибки на вентиле, равной p . Для рассмотренного примера полином будет выглядеть следующим образом:

$$F(p) = \frac{1}{4}(0 \cdot (1 - p)^2) + \frac{1}{4}(4 \cdot (1 - p)p) + \frac{1}{4}(2 \cdot p(1 - p)) + \frac{1}{4}(2 \cdot p^2);$$

$$F(p) = 1,5p - p^2. \quad (1)$$

Таблица 1

Таблица истинности по всем (\mathbf{X}, \mathbf{e}) для схемы прямой импликации

№	e_1	e_2	x_1	x_2	$f(x_1, x_2)$	$f^*(x_1, x_2, e_1, e_2)$	$E(\mathbf{X}, \mathbf{e})$	Вероятность появления вектора ошибки
0	0	0	0	0	1	1	0	$(1 - p)^2$
1	0	0	0	1	0	0	0	
2	0	0	1	0	1	1	0	
3	0	0	1	1	1	0	0	$(1 - p)p$
4	0	1	0	0	1	0	1	
5	0	1	0	1	0	1	1	
6	0	1	1	0	1	0	1	
7	0	1	1	1	1	0	1	$p(1 - p)$
8	1	0	0	0	1	1	0	
9	1	0	0	1	0	1	1	
10	1	0	1	0	1	1	0	
11	1	0	1	1	1	0	1	p^2
12	1	1	0	0	1	0	1	
13	1	1	0	1	0	0	0	
14	1	1	1	0	1	0	1	
15	1	1	1	1	1	1	0	

Теперь выведем формулу для расчета полинома ошибки в общем виде для произвольной логической схемы. Учитывая, что вероятность появления на входе конкретного вектора входных сигналов \mathbf{X} длины N (в предположении равновероятности всех таких наборов) равна $\frac{1}{2^N}$, а вероятность возникновения

вектора ошибки \mathbf{e} длины M и веса $|\mathbf{e}|$ равна $p^{|\mathbf{e}|}(1-p)^{M-|\mathbf{e}|}$, получаем вероятность ошибки на выходе схемы (вес вектора равен числу его ненулевых элементов):

$$F(p) = \frac{1}{2^N} \sum_{\mathbf{X}, \mathbf{e}} E(\mathbf{X}, \mathbf{e}) p^{|\mathbf{e}|} (1-p)^{M-|\mathbf{e}|}. \quad (2)$$

Из формулы (2) видно, что вычислительная сложность аналитического расчета полинома ошибки экспоненциально зависит от числа входов и числа элементов, что делает этот метод неприменимым даже для сравнительно небольших схем. В настоящее время все методы оценки маскирующих свойств комбинационной логики балансируют между высокой вычислительной сложностью и точностью результатов характеристик сбоеустойчивости [6].

В рамках настоящей работы для оценки сбоеустойчивости логических схем предлагается использовать некоторый обобщенный коэффициент логической чувствительности схемы:

$$\alpha = \frac{1}{2^N} \sum_{\mathbf{X}, \mathbf{e}, |\mathbf{e}|=1} E(\mathbf{X}, \mathbf{e}), \quad (3)$$

где $|\mathbf{e}| = 1$ говорит о том, что суммирование идет только по векторам ошибки с весом, равным единице.

Эта метрика сбоеустойчивости имеет ряд преимуществ по сравнению с предложенными ранее метриками [6]. Во-первых, вычислительная сложность метода линейная относительно числа элементов, что вкупе с методами бит-параллельного моделирования и методами Монте-Карло позволяет использовать эту метрику для сравнительно больших схем. Во-вторых, предлагаемый коэффициент не зависит от вероятности сбоя вентиля, что позволяет использовать его на ранних этапах проектирования сбоеустойчивых схем, а также методов повышения сбоеустойчивости, когда не определена элементная база и условия эксплуатации схемы. В-третьих, для большинства практических применений, в условиях, когда вероятность сбоя вентиля стремится к нулю — эта аппроксимация является наиболее точной, являясь касательной к графику полинома ошибки в точке ноль. Кроме того, предлагаемая метрика может быть

модифицирована для оценки участков подсхем в составе больших комбинационных схем. Для этого достаточно в формуле (2) учесть два дополнительных коэффициента, характеризующих вероятность возникновения входной комбинации и вероятность наблюдаемости конкретной ошибки подсхемы на выходах основной схемы.

Разработка маршрута ресинтеза комбинационных схем в целях повышения логической устойчивости к одиночным сбоям

Для больших логических схем, для которых невозможно в разумные сроки посчитать параметры сбоеустойчивости и сгенерировать более устойчивую к сбоям схему, был разработан подход, который заключается в локальной перезаписи небольших участков схемы. Суть метода заключается в итерационном изменении схемы, где каждая итерация гарантированно увеличивает уровень логического маскирования схемы. Предлагаемый подход не гарантирует нахождение оптимума по параметру сбоеустойчивости, однако позволяет за разумное время повысить устойчивость схемы к единичным сбоям. Общая схема процесса ресинтеза представлена на рис. 2.

Как видно из рис. 2, в контексте задачи ресинтеза можно выделить три больших подзадачи:

- 1) подсчет характеристик;
- 2) выбор уязвимой подсхемы;
- 3) генерация набора эквивалентных подсхем и оценка их отказоустойчивости.

Рассмотрим последовательно пути решения обозначенных задач.

Задача предлагаемого итеративного подхода к ресинтезу заключается в том, чтобы после замены подсхемы на более надежную сбоеустойчивость всей схемы гарантированно увеличилась. Если для исход-

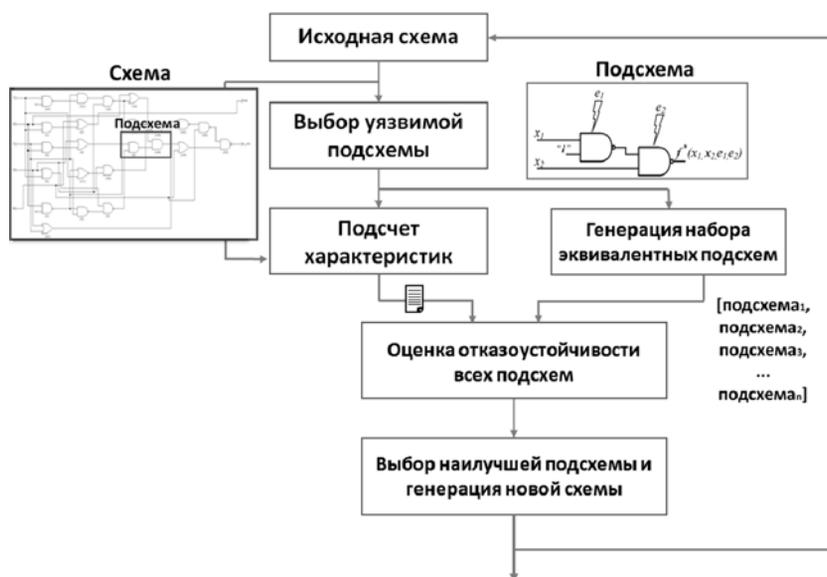


Рис. 2. Общий маршрут ресинтеза логической схемы

ной схемы и для подсхем применять разработанную метрику, то в общем случае это не всегда так. Эксперименты показали, что зачастую замена подсхемы на более надежный эквивалент приводит к деградации сбоеустойчивости исходной схемы. В результате исследований было выявлено два основных фактора, которые влияют на возникновение подобных ситуаций.

Первый фактор связан с предположением о равновероятности входных комбинаций, которое было принято при выводе формулы полинома ошибки. И если для исходной схемы это допущение является оправданным, то в случае подсхемы оно становится источником существенных погрешностей. Решением этой проблемы является модификация метрики для оценки отказоустойчивости подсхем таким образом, чтобы там учитывалось входное распределение входных узлов. Для этого необходимо вычислить таблицы вероятностей входных комбинаций для подсхемы, что приведет к появлению дополнительного коэффициента в формуле

$$\alpha = \sum_{\mathbf{X}, \mathbf{e}, |\mathbf{e}|=1} p(\mathbf{X}) E(\mathbf{X}, \mathbf{e}).$$

Второй фактор связан с учетом маскирующих свойств исходной схемы. Дело в том, что разные ошибки на выходах подсхемы имеют разные вероятности распространения по схеме и влияния на ее выходы. Этот факт также не отражен в предложенной метрике. Для того чтобы учесть наблюдаемость ошибок на выходных узлах подсхемы, необходимо построить двумерную квадратную матрицу наблюдаемости ошибок размерностью $2^n \times 2^n$, где n — число выходов подсхемы. В соответствующих ячейках хранятся вероятности того, что ошибка вида $abcd \rightarrow abce$ на выходах подсхемы даст ошибку на выходах всей схемы. С учетом наблюдаемости возможных ошибок на выходе схемы формула для подсчета сбоеустойчивости подсхемы примет следующий вид:

$$\alpha = \sum_{\mathbf{X}, \mathbf{e}, |\mathbf{e}|=1} E(\mathbf{X}, \mathbf{e}) p(\mathbf{X}) p(f(\mathbf{X}) \rightarrow f^*(\mathbf{X}, \mathbf{e})),$$

где $p(\mathbf{X})$ — вероятность возникновения входного набора; $p(f(\mathbf{X}) \rightarrow f^*(\mathbf{X}, \mathbf{e}))$ — вероятность того, что ошибка, возникшая на выходе подсхемы, окажется наблюдаемой для исходной схемы.

Используя такую расширенную метрику, можно с большой вероятностью гарантировать, что если предлагаемый коэффициент подсхемы A больше коэффициента подсхемы B , то сбоеустойчивость всей схемы в случае использования подсхемы A будет больше, чем при использовании подсхемы B . Более того, количественные значения прироста предлагаемого коэффициента чувствительности при замене подсхемы примерно равны приросту коэффициента чувствительности всей схемы целиком. Точного соответствия принципиально невозможно достичь ввиду того, что мы считаем два вышеописанных

фактора и соответствующие им вероятности независимыми, что не соответствует действительности. Для того чтобы учесть существующую зависимость потребуется выполнить такое количество вычислений, которое равно количеству вычислений, необходимых для подсчета коэффициента логической устойчивости исходной схемы. В этом случае теряется смысл оценки подсхемы, поскольку с такой же вычислительной нагрузкой можно после каждой замены оценивать сбоеустойчивость всей схемы целиком. С этих позиций данное допущение (о независимости вероятностей) является разумным компромиссом между быстродействием и точностью получаемых предсказаний.

Важно отметить, что рассматриваемые характеристики — вероятность возникновения входного набора и наблюдаемость ошибки на выходе подсхемы — не зависят от структуры подсхемы, а являются исключительно параметрами логической структуры "до" и "после" рассматриваемого участка. Следствием этого является возможность предварительного подсчета характеристик для дальнейшего использования в вычислениях коэффициента чувствительности для подсхем разной конфигурации. Это нашло отражение в разработанном маршруте ресинтеза комбинационных схем.

Рассмотрим конкретный пример. Рассмотрим подсхему с тремя входами и двумя выходами (рис. 3). В этом случае $P(\mathbf{X})$ будет состоять из восьми значений вероятностей для каждой из комбинаций входных воздействий.

В табл. 2 приведена матрица наблюдаемости ошибок. Здесь в помеченной ячейке хранится значение вероятности для события, что ошибка вида $(1\ 0) \rightarrow (0\ 1)$ окажется наблюдаемой на выходе схемы.

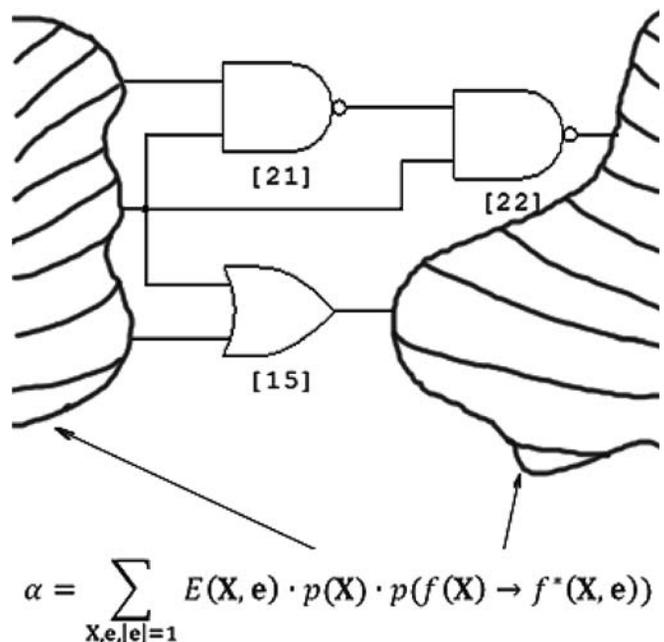


Рис. 3. Подсчет характеристик подсхемы

Таблица 2
Наблюдаемость ошибок подсхемы на выходах основной схемы

	0 0	0 1	1 0	1 1
0 0	q00	q01	q02	q03
0 1	q10	q11	q12	q13
1 0	q20	q21	q22	q23
1 1	q30	q31	q32	q33

Следующим важным этапом в разработке маршрута ресинтеза комбинационных схем является задача выбора уязвимых участков схемы. В общем случае можно выбирать произвольные подсхемы и пытаться отыскать более устойчивые аналоги. Однако такой подход может негативно отразиться на эффективности алгоритма ресинтеза с точки зрения скорости его сходимости. Но даже в такой упрощенной постановке задачи возникает множество вопросов относительно того, как именно и какие именно подсхемы следует выбирать. Во-первых, необходимо определиться с размером подсхем таким образом, чтобы можно было точно посчитать все необходимые характеристики: распределение логических значений на входах, матрицу наблюдаемости ошибок для выходов и коэффициент чувствительности схемы. Эксперименты показали, что на заданные подсхемы необходимо наложить следующие ограничения: число входов не более 10 и число выходов не более 12. В этом случае размер логических характеристик на входах не превосходит 1024, а размерность матрицы ошибок на выходах подсхемы будет не более 4096×4096 .

Для генерации случайной подсхемы используется следующий эвристический алгоритм:

Шаг 0. Создаем пустую подсхему.

Шаг 1. Выбираем случайный элемент в схеме и добавляем его в список элементов подсхемы.

Шаг 2. Случайно выбираем произвольный узел, являющийся входом подсхемы. Добавляем к подсхеме все элементы, которые подсоединены к этому узлу.

Шаг 3. Считаем число входов в схеме. Если оно незначительно превысило или равно заданному числу входов, возвращаем подсхему, иначе переходим к Шагу 2. Если число входов и выходов превысило лимиты в 10 входов или 12 выходов, переходим к Шагу 0.

Рассмотрим этот процесс на детальном примере. Пусть нам требуется получить произвольную подсхему с пятью входами из заданной на рис. 3.

На первом шаге выбирается случайный вентиль N19 (рис. 4). Далее выбираем случайный вход, который соединен с вентилями N16 и N19,

трех входов и одного выхода. Положим дальше из трех входов подсхемы, мы выбрали узел, ведущий к вентилю N12. В этом случае в подсхему также требуется добавить элемент N13, который подключен к тому же узлу. Подсхема теперь состоит из четырех элементов N12, N13, N16 и N19, пяти входов и двух выходов. Мы достигли лимита в пять входов, следовательно, требуемая подсхема получена.

Кроме того, было разработано несколько модификаций данного метода, которые позволяют получать наиболее уязвимые подсхемы с точки зрения их наблюдаемости на основных выходах исходной схемы. Для этого перед началом алгоритма генерации случайной подсхемы выполняем построение карты уязвимости схемы. Иными словами, определяем значения наблюдаемости для каждого вентиля в схеме. Имея такую информацию, можно выбирать первоначальный вентиль для ветвления с учетом его уязвимости, а на втором шаге алгоритма ветвиться в сторону более уязвимых элементов. Для этого был использован метод асимметричного колеса рулетки. Предлагаемый метод направленного поиска уязвимой подсхемы позволяет ускорить сходимость алгоритма за счет того, что каждая успешная итерация вносит больший прирост для сбоеустойчивости схемы. Стоит также отметить, что задача построения "карты уязвимости" схемы является базовой процедурой при подсчете коэффициента чувствительности исходной схемы, так как он, по сути, является суммой наблюдаемостей всех элементов схемы. Таким образом, предложенный метод направленного поиска уязвимой подсхемы практически не приносит дополнительных вычислительных затрат, так как вся необходимая для него информация в неявном виде рассчитывается на каждой итерации алгоритма ресинтеза.

Наиболее сложной является задача синтеза эквивалентной подсхемы, имеющей наименьшую наблюдаемость для исходной схемы. Задача осложняется тем, что при подсчете коэффициента чувствительности используются параметры, связанные с ис-

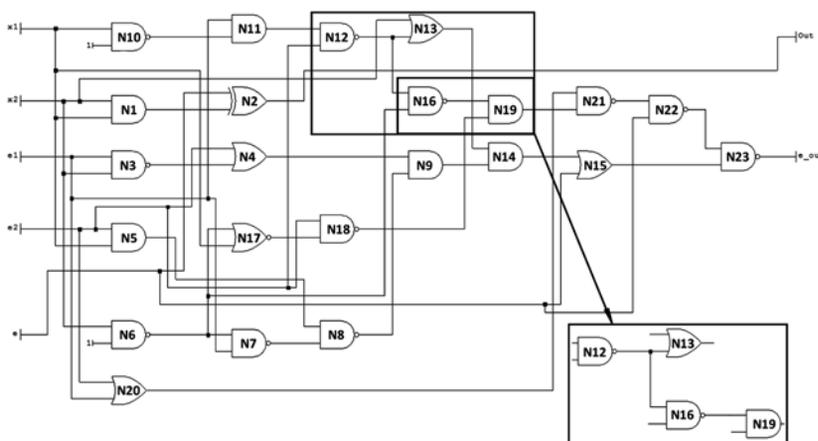


Рис. 4. Тестовая схема для задачи ресинтеза

ходной схемой. Это приводит к тому, что одна и та же подсхема может иметь абсолютно разные коэффициенты чувствительности, будучи помещенной в разные части схемы. Этот факт не позволяет использовать традиционные методы с использованием заранее просчитанной библиотеки стандартных замен. На текущий момент задача решается с помощью генерации большого набора подсхем различной конфигурации с последующим выбором подсхемы с наименьшим параметром коэффициента чувствительности. В данной работе использовано несколько подходов по автоматической генерации подсхем, эквивалентных заданной:

- несколько эвристических методов генерации подсхемы на базе минимизации логической функции с помощью Espresso [7];
- синтез оптимизированной схемы с помощью свободного ПО для синтеза схем Yosys-ABC [8];
- тройное резервирование (TMR) для подсхемы.

Отсутствие методов направленного синтеза подходящей подсхемы выливается в очень большое число холостых прогонов алгоритма: доля успешных итераций в текущей реализации составляет всего 1—2 %. Задача по увеличению этой доли остается в качестве основного направления дальнейших исследований.

Результаты вычислительных экспериментов

В ходе исследований был разработан программный пакет на языке *Python*, с критическими частями, написанными на *C* в целях увеличения скорости вычислений. Программа свободно доступна по ссылке [9].

Для первого вычислительного эксперимента использовали схемы ISCAS85 [10], предварительно синтезированные в базе библиотеки, состоящей из семи базовых элементов: INV, NAND, NOR, AND, OR, XOR, XNOR. Синтез проводили с помощью программы *Synopsys Design Compiler* (с оптимизацией по задержке). В табл. 3 приведены значения коэффициента чувствительности для схемы до и после

ресинтеза. Итерация считается успешной, если удалось найти более надежный аналог выбранной случайной подсхемы. Если не было успешных замен за последние 1000 итераций, работа алгоритма завершилась.

Замечание: в алгоритме не был использован метод TMR для подсхем ввиду серьезного увеличения площади схемы при относительно незначительном уменьшении коэффициента чувствительности. Метод TMR имеет смысл использовать для замены сразу больших участков схем, при условии, что нельзя выполнить TMR сразу над всей схемой ввиду жестких ограничений по площади. Все сгенерированные подсхемы имели площадь, сравнимую с площадью изначальной подсхемы.

Как видно из табл. 3, коэффициент чувствительности для схемы уменьшается иногда более чем на 18 %. При этом площадь изменяется незначительно и иногда даже уменьшается. То есть маскирование ошибок происходит не за счет увеличения избыточности, а за счет оптимальной логики работы схемы.

Длина критического пути для схемы после ресинтеза в рамках данного вычислительного эксперимента увеличивается, что снижает скорость работы схемы. Это связано с тем, что для моделирования использовали схемы, максимально оптимизированные по скорости работы (*compile_ultra*), и любое изменение логической структуры с большой вероятностью приводит к увеличению критического пути. Вторая вероятная причина — на более длинном пути прохождения сигнала маскируется больше сбоев. Дополнительный эксперимент на c432 показал, что задержка для подсхемы и ее замены в среднем уменьшается на 5 %, однако общая задержка схемы после всех замен выросла на 31 %.

В следующем эксперименте была сделана попытка проанализировать, как влияет начальный синтез схемы на эффективность последующего ресинтеза. Для этого использовали различные варианты синтеза c432: структурная версия до синтеза, синтез в *Synopsys* с оптимизацией по задержке,

Таблица 3

Результаты работы алгоритма ресинтеза на схемах ISCAS85

Схема	c432	c1355	c1908	c3540	c5315	c6288	c7552
Число входов	36	41	33	50	178	32	207
Коэффициент чувствительности (до)	63,36	91,08	140,71	299,63	641,38	1476,69	566,43
Коэффициент чувствительности (после)	51,76	90,94	121,19	278,88	612,55	1376,68	514,22
Улучшение, %	18,31	0,2	13,9	6,9	4,5	6,8	9,2
Число логических элементов (до)	204	234	287	958	1351	1776	1369
Число логических элементов (после)	192	238	284	1000	1421	1700	1486
Процент от начального, %	94,12	101,7	99,0	104,4	105,2	95,7	108,6
Задержка на критическом пути, нс (до)	19	9	17	24	16	55	17
Задержка на критическом пути, нс (после)	25	9	20	29	18	66	26
Изменение задержки, %	31,58	0	17,7	20,8	12,5	20	52,9
Общее число итераций	5128	5000	4540	7805	11 841	11 572	16 534
Число успешных замен	46	6	32	101	110	256	165
Успешные замены, %	0,9	0,1	0,7	1,3	0,9	2,2	1

Результаты ресинтеза разных версий схемы c432

Схема	c432 Initial	c432 Min Path	c432 Min Area	c432 ABC
Коэффициент чувствительности (до)	71,71	63,36	54,49	107,66
Коэффициент чувствительности (после)	54,53	51,76	50,82	56,23
Улучшение, %	23,96	18,31	6,74	47,77
Число логических элементов (до)	216	204	166	301
Число логических элементов (после)	197	192	171	400
Процент от начального, %	91,2	94,12	103,01	132,89
Задержка на критическом пути, нс (до)	44	19	28	43
Задержка на критическом пути, нс (после)	34	25	29	58
Изменение задержки, %	-22,73	31,58	3,57	34,88
Общее число итераций	4575	5128	2483	9759
Число успешных замен	62	46	20	156
Успешные замены, %	1,36	0,9	0,81	1,6

Таблица 5

Результаты работы алгоритма ресинтеза на схемах с малым числом входов из набора LGSynth89

Схема	5xp1	9sym	alu2_synth	alu4_synth	bw	f51m_synth	ldd_synth	misex3
Число входов	7	9	10	14	5	8	9	14
Коэффициент чувствительности (до)	54,8	18,7	145,8	281,8	107,8	61,8	73,1	155,2
Коэффициент чувствительности (после)	50,5	17,4	78,5	163,6	98,5	43,7	51,2	104,5
Улучшение, %	7,85	6,95	46,16	41,94	8,63	29,29	29,96	32,67
Число логических элементов (до)	135	252	386	741	195	111	85	1509
Число логических элементов (после)	132	255	752	1490	189	103	81	2142
Процент от начального, %	97,8	101,2	194,8	201,1	96,9	92,8	95,3	141,9
Задержка на критическом пути, нс (до)	11	16	30	37	10	26	14	22
Задержка на критическом пути, нс (после)	11	15	45	53	9	27	12	26
Изменение, %	0	-6,25	50	43,24	-10	3,9	-14,3	18,2
Общее число итераций	4151	2945	26 026	29 846	3176	6618	3004	18 077
Число успешных замен	23	12	316	589	22	60	46	267
Успешные замены, %	0,55	0,41	1,21	1,97	0,69	0,91	1,53	1,48

синтез в Synopsys с оптимизацией по площади, и стандартный синтез в системе *Yosys-ABC*. Результаты работы ресинтеза для всех четырех схем приведены в табл. 4.

Еще один эксперимент был проведен для схем из набора LGSynth89 [11]. Были выбраны схемы с небольшим числом входов и выходов, чтобы статистические параметры для схемы считались точно, а не на базе метода Монте-Карло, как в эксперименте со схемами ISCAS85. Результаты приведены в сводной табл. 5.

Заключение

В результате проведенного исследования были получены практические результаты, свидетельствующие о том, что предложенный метод ресинтеза может быть эффективно использован в любом маршруте проектирования на этапе постсинтеза. Его можно применять как самостоятельно, так и в дополнение к традиционным методам кратного резервирования, поскольку он сам незначительно влияет на площадь конечного устройства. К основному недостатку метода можно отнести маленький процент успешных итераций (1–2 %), что ведет к относительно медленной работе метода. Однако для устранения этого и других недостатков оста-

лись широкие возможности для исследований, которые не были охвачены в данной работе.

1. Генерация схем на замену проводится случайным образом, если же генерировать схему сразу с учетом входного распределения и матрицы ошибок, то можно увеличить частоту успешных итераций, а также повысить общий процент улучшения коэффициента чувствительности.

2. Выбор случайной подсхемы в данный момент, очевидно, не является оптимальным.

3. Скорость работы методов, основанных на методе Монте-Карло, довольно низкая — требуется изучить возможности использования более быстрых методов.

4. Необходима разработка методов предварительной оценки возможности улучшения показателей чувствительности схемы методом ресинтеза еще до начала моделирования.

5. В настоящее время на схемах с малым числом логических уровней (например, c1355) метод дает маленький процент улучшения. Необходимо исследовать фундаментальные причины этого явления.

6. Необходима доработка алгоритма для учета ограничений на общую площадь схемы, а также длину критического пути.

Для удобства разработчиков ресинтез лучше использовать в составе пакета логического синтеза. Среди свободных решений есть открытый пакет для логического синтеза — Yosys. Планируется реализация модуля ресинтеза для Yosys, после обработки маршрута на Python.

Список литературы

1. Mahatme N. N., Jagannathan S., Loveless T. D., Massengill L. W., Bhuvan B. L., Wen S. J. et al. Comparison of combinational and sequential error rates for a deep submicron process // *IEEE Trans Nucl Sci (TNS)*. 2011. Vol. 58 (6): 27. P. 19–25.
2. Asadi H., Tahoori M. B., Fazeli M., Miremadi S. G. Efficient algorithms to accurately compute derating factors of digital circuits // *Microelectronics Reliability*. 2012. Vol. 52 (6): 12. P. 15–26.
3. George N., Lach J. Characterization of logical masking and error propagation in combinational circuits and effects on system vulnerability // *Dependable Systems Networks (DSN)*. 2011 IEEE/IFIP 41st International Conference. 2011. P. 323–334.

4. Von Neumann J. Probabilistic logics and the synthesis of reliable organisms from unreliable components // *Automata Studies* / C. E. Shannon, J. McCarthy, Eds. Princeton, NJ: Princeton Univ. Press, 1956. P. 43–98.
5. Стемповский А. Л., Телпухов Д. В., Соловьев Р. А., Соловьев А. Н., Мячиков М. В. Моделирование возникновения неисправностей для оценки надежностных характеристик логических схем // *Информационные технологии*. 2014. № 11. С. 30–36.
6. Xiao R., Chen C. Gate-level circuit reliability analysis: A survey // *VLSI Design*. 2014. P. 1–12.
7. Espresso — a Multi-valued PLA minimization. URL: <http://embedded.eecs.berkeley.edu/pubs/downloads/espresso/>
8. Wolf C., Glaser J. Yosys—A Free Verilog Synthesis Suite // *Proceedings of the 21st Austrian Workshop on Microelectronics (Austrochip)*, Linz, Austria, 2013. Vol. 10. 6 p.
9. Reliability-aware resynthesis. URL: <https://github.com/ID-MIPPM/reliability-aware-resynthesis>
10. Bryan D. The ISCAS'85 benchmark circuits and netlist format // North Carolina State University, 1985. 4 p.
11. Yang S. Logic synthesis and optimization benchmarks user guide: version 3.0 // Microelectronics Center of North Carolina (MCNC), 1991. 45 p.

A. L. Stempkovskiy, Academician, Director, D. V. Telpukhov, Ph. D., Head of the Department,
R. A. Solovyev, Ph. D., Chief Researcher, ZF-Turbo@yandex.ru
Institute for Design Problems in Microelectronics, Moscow (IPPM RAS)

Enhancing Reliability of Logic Circuits with Partial Resynthesis Method

A number of diverse factors in the area of microelectronics industry significantly actualized the developments in the field of fault-tolerant combinational circuits design recently. Currently in modern CAD systems there are no means to assess and improve logic masking effects in combinational circuits. In this paper, an attempt was made to create a methodology for the implementation of such software. This paper proposes re-synthesis design flow, providing a logical masking capability for the combinational circuits resistance to a single event upsets without the introduction of substantial redundancy. The details of the method including the development of reliability metrics, the selection of vulnerable sub-schemes, as well as methods of generating reliable circuits are presented. Computational experiments were conducted to demonstrate the high efficiency of the proposed method.

Keywords: combinational circuit, gate observability, sensitivity factor, resynthesis

References

1. Mahatme N. N., Jagannathan S., Loveless T. D., Massengill L. W., Bhuvan B. L., Wen S. J. et al. Comparison of combinational and sequential error rates for a deep submicron process, *IEEE Trans Nucl Sci (TNS)*, 2011, vol. 58 (6): 27, pp. 19–25.
2. Asadi H., Tahoori M. B., Fazeli M., Miremadi S. G. Efficient algorithms to accurately compute derating factors of digital circuits, *Microelectronics Reliability*, 2012, vol. 52 (6): 12, pp. 15–26.
3. George N., Lach J. Characterization of logical masking and error propagation in combinational circuits and effects on system vulnerability, *Dependable Systems Networks (DSN)*, 2011, *IEEE/IFIP 41st International Conference*, 2011, pp. 323–334.
4. Von Neumann J. Probabilistic logics and the synthesis of reliable organisms from unreliable components, *Automata Studies*, C. E. Shannon, J. McCarthy, Eds. Princeton, NJ: Princeton Univ. Press, 1956, pp. 43–98.

5. Stempkovskij A. L., Tel'pukhov D. V., Solov'ev R. A., Solov'ev A. N., Mjachikov M. V. Modelirovanie vzniknovenija neispravnostej dlja ocenki nadezhnostnyh harakteristik logicheskikh shem, *Informacionnye tehnologii*, 2014, no. 11, pp. 30–36 (in Russian).
6. Xiao R., Chen C. Gate-level circuit reliability analysis: A survey, *VLSI Design*, 2014, pp. 1–12.
7. Espresso — a Multi-valued PLA minimization, URL: <http://embedded.eecs.berkeley.edu/pubs/downloads/espresso/>
8. Wolf C., Glaser J. Yosys — A Free Verilog Synthesis Suite, *Proceedings of the 21st Austrian Workshop on Microelectronics (Austrochip)*, Linz, Austria, 2013, vol. 10, 6 p.
9. Reliability-aware resynthesis, URL: <https://github.com/ID-MIPPM/reliability-aware-resynthesis>
10. Bryan D. *The ISCAS'85 benchmark circuits and netlist format*, North Carolina State University, 1985, 4 p.
11. Yang S. *Logic synthesis and optimization benchmarks user guide: version 3.0*, Microelectronics Center of North Carolina (MCNC), 1991, 45 p.

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ CRYPTOSAFETY INFORMATION

УДК 004.738

Р. Э. Асратян, канд. техн. наук, вед. науч. сотр., e-mail: rea@ipu.ru,
В. Н. Лебедев, канд. техн. наук, зав. лаб., e-mail: lebvini@ipu.ru,
В. Л. Орлов, канд. техн. наук, вед. науч. сотр., e-mail: ovl@ipu.ru
Институт проблем управления им. В. А. Трапезникова РАН

Эффективность защищенных каналов взаимодействия на основе применения протокола HTTPS в прокси-серверах

Статья посвящена экспериментальному исследованию временных характеристик нового подхода к организации безопасного информационного взаимодействия в мультисетевой среде, основанного на применении протокола HTTPS и технологии прокси-серверов. Суть подхода заключается в организации защищенных HTTPS-соединений между прокси-серверами для безопасной передачи информационных запросов из одной частной локальной сети в другую через глобальную сеть. Приводятся результаты экспериментов, позволяющих определить область эффективного применения подхода.

Ключевые слова: распределенные системы, Интернет-технологии, сетевые протоколы, прокси-серверы, информационное взаимодействие, информационная безопасность, web-сервисы

Введение

Интерес разработчиков информационных систем к сетевому протоколу HTTPS [1] как к удобному и доступному средству организации защищенного информационного взаимодействия в глобальных сетях не ослабевает с течением времени. Отчасти это связано с появлением возможности защиты информационных запросов к электронным сервисам в рамках технологии .NET [2, 3] на его основе (протокол SOAP "поверх" HTTPS), отчасти же — с широким распространением распределенных информационных систем и постоянным возрастанием требований к безопасности данных [4, 5] в таких системах.

Аспект информационной безопасности особенно важен в разработках распределенных систем, ориентированных на работу в сложных мультисетевых средах, включающих в себя одну или несколько глобальных сетей и множество частных локальных сетей предприятий различного размера и различной административной подчиненности. Разработчики таких систем зачастую сталкиваются с серьезными проблемами при решении задач маршрутизации и защиты информационных запросов, направленных из одной частной сети в другую.

В работе [6] описан подход к организации безопасного взаимодействия между клиентскими и серверными компонентами, размещенными в разных частных сетях, основанный на применении прокси-серверов — серверов-посредников, размещаемых "на границах" частных сетей и обеспечивающих создание защищенных HTTPS-тоннелей через гло-

бальную сеть (рис. 1). Преимущество данного подхода по сравнению с известными (например, с технологиями NAT или VPN [7, 8]) заключается в том, что на прокси-серверы можно возложить дополнительные функции, связанные с учетом (регистрацией), контролем, фильтрацией, маршрутизацией и защитой информационных запросов. При данном подходе удаленные клиентские и серверные программы пользуются протоколом HTTP и, как и при применении VPN, нисколько не заботятся об информационной защите, а на прокси-серверы возлагаются дополнительные функции шлюзов "HTTP → HTTPS" и "HTTPS → HTTP". Особен-

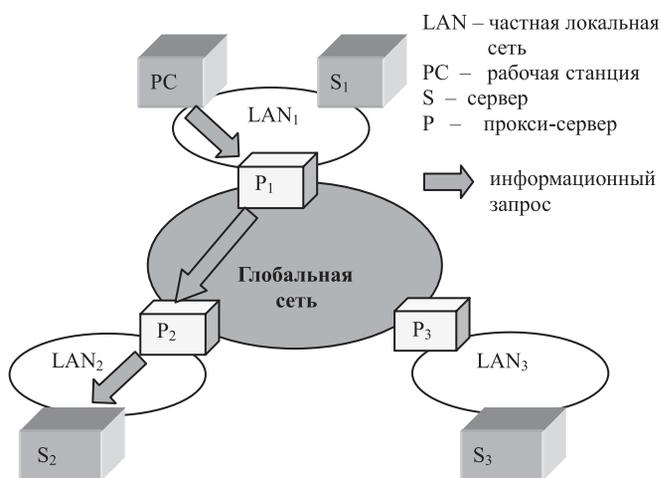


Рис. 1. Маршрутизация запросов в мультисетевой среде

ность подхода заключается в том, что в этом случае прокси-серверы "видят" проходящие через них информационные запросы клиентов и ответы серверов в открытой, дешифрованной форме и способны обеспечить необходимый контроль и высокоуровневую маршрутизацию потока запросов по символическим именам (URL) информационных ресурсов.

Однако данный подход имеет принципиальный недостаток — дополнительные временные задержки, вносимые прокси-серверами в обработку запросов между удаленными частными сетями. Так как эти задержки в значительной степени определяют область эффективного применения подхода, их оценка является необходимым условием его целенаправленного использования. В данной работе приводятся основные результаты экспериментов, направленных на оценку дополнительных накладных расходов времени, связанных с применением защищенных HTTPS-тоннелей на основе прокси-серверов.

Краткое описание принципов организации HTTPS-тоннеля

Суть рассматриваемого подхода заключается в соединении возможностей технологии HTTPS и технологии прокси-серверов для решения двух взаимосвязанных задач:

- обеспечения информационной безопасности данных в условиях угроз, исходящих от глобальной сети;
- маршрутизации запросов между клиентскими и серверными программами, функционирующими в разных частных сетях (см. рис. 1).

Главная особенность подхода заключается в обеспечении "высокоуровневых" средств защиты и маршрутизации, основанных на заголовках и данных HTTP-запросов, а не отдельных IP-пакетов (отметим, что решение этих задач на сетевом уровне, например на основе VPN, может оказаться невозможным или совершенно неэффективным, если число частных сетей исчисляется десятками).

Организация защищенного HTTPS-тоннеля с помощью двух прокси-серверов, выполняющих функции шлюзов "HTTP → HTTPS" и "HTTPS → HTTP", проиллюстрирована на рис. 2. Как видно из рисунка, каждый из прокси-серверов оснащается про-

граммной поддержкой SSL/TLS-технологии, обеспечивающей построение защищенного двустороннего канала взаимодействия "поверх" обычного TCP-соединения.

Как видно из рис. 1, каждый запрос, направленный из одной частной сети в другую, проходит по крайней мере через два прокси-сервера и через два этапа маршрутизации. Из ближайшего прокси-сервера запрос сначала передается в удаленный прокси-сервер по глобальной сети, а уже потом — к адресуемому информационному ресурсу. На каждом этапе маршрутизация проводится на основе символического имени адресуемого ресурса, содержащегося в HTTP-заголовке запроса, с помощью двух таблиц маршрутизации: "глобальной", связывающей Интернет-имя ресурса с адресом удаленного прокси-сервера, и "локальной", связывающей Интернет-имя ресурса с его адресом в частной сети.

Скорость обработки и эффективность применения подхода

К принципиальным недостаткам описанного подхода следует отнести дополнительные временные задержки, вносимые защищенными HTTPS-каналами на основе технологии прокси-серверов. Для оценки этих задержек была проведена серия экспериментов с опытной реализацией шлюзов "HTTP → HTTPS" и "HTTPS → HTTP". Цель экспериментов заключалась в сравнении времен выполнения запросов к модельным электронным сервисам в условиях прямого обращения (без шлюзов) и при обращении через защищенный HTTPS-канал (т.е. через шлюзы). Важно подчеркнуть, что измерялось не только время выполнения одиночных запросов, но и скорость обработки (число обработанных запросов в секунду), достигаемая при одновременной обработке пакета информационных запросов. Другими словами, мы попытались провести сравнение с учетом распараллеливания обработки в множестве программных потоков (и в прокси-сервере, и в службе электронных сервисов для обработки каждого запроса создается отдельный программный поток). Скорость обработки вычислялась как частное от деления числа запросов в пакете на полное время его выполнения.

Эксперименты проводились в условиях скоростного (100 Мбит/с) Ethernet в локальной сети. Модельные web-сервисы [1, 2] с временами выполнения 500, 1000 и 2000 мс были установлены на четырехъядерном сервере приложений с тактовой частотой 3,4 ГГц и 4 Гбайт оперативной памяти (оценка скоростных характеристик выполнялась для каждого электронного сервиса в отдельности). Что же касается прокси-серверов, то для их установки использовались как доста-

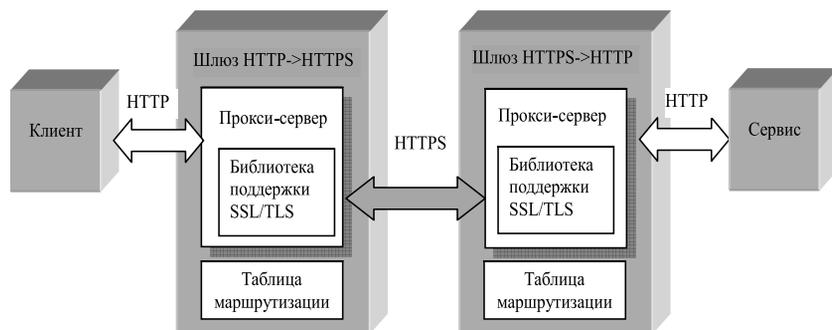


Рис. 2. Принципы организации HTTPS-тоннеля

точно скромные одноядерные компьютеры с тактовой частотой 2,8 ГГц, так и более современные четырехъядерные с тактовой частотой 3,4 ГГц (чтобы оценить, насколько важно использовать серверы с высокими характеристиками в реализации защищенного канала).

На рис. 3 показан характерный график, который дает представление о дополнительных задержках, вносимых защищенным HTTPS-каналом, при обращении к модельному web-сервису с фиксирован-

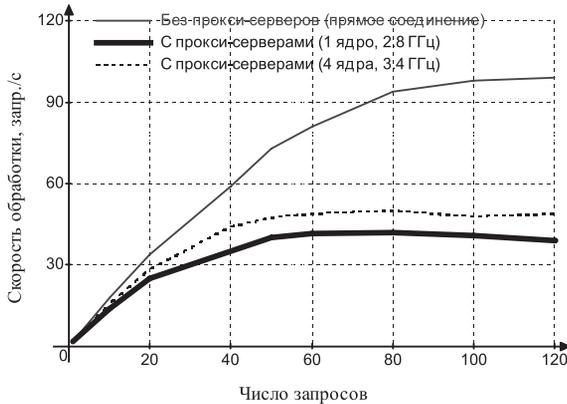


Рис. 3. График зависимости скорости обработки от числа одновременных запросов при задержке web-сервиса 500 мс

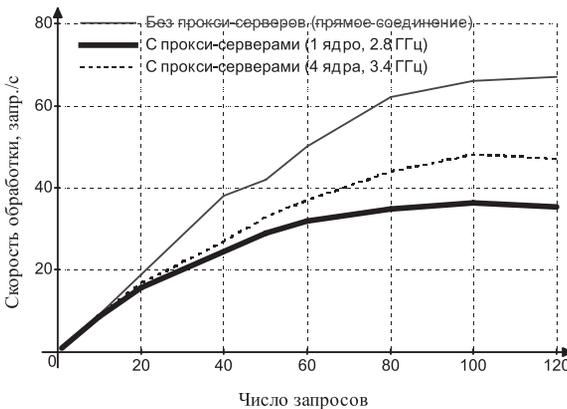


Рис. 4. График зависимости скорости обработки от числа одновременных запросов при задержке web-сервиса 1000 мс

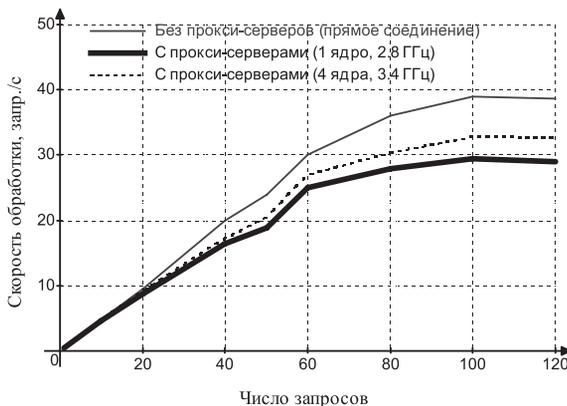


Рис. 5. График зависимости скорости обработки от числа одновременных запросов при задержке web-сервиса 2000 мс

ном временем выполнения (задержкой), равным 0,5 с. В эксперименте измерялась скорость обработки (число обработанных запросов в секунду), достигаемая при обработке группы запросов, т.е. при одновременном обращении сразу нескольких клиентов. По горизонтальной оси графика отложено число запросов в группе, а по вертикальной — скорость обработки. Кривая, прорисованная тонкой сплошной линией, отражает скорость обработки при прямом обращении к сервису (без шлюзов), а кривые, прорисованные толстой и прерывистой линиями, отражают скорость обработки со шлюзами, функционирующими на одноядерном и четырехъядерном серверах соответственно.

При подаче одиночного запроса на вход сервиса со временем выполнения 0,5 с полное время обработки составило 515 мс без шлюзов, и 546 мс со шлюзами (1,94 и 1,88 мс в секунду соответственно). Как видно из графика, при росте числа одновременных запросов растет и скорость обработки, что объясняется "многоканальной" природой и прокси-серверов, и электронного сервиса, т.е. положительным эффектом от распараллеливания обработки в нескольких программных потоках. Например, при одновременной подаче 10 запросов время их выполнения составило 544 мс без шлюзов, 627 мс со шлюзами, реализованными в четырехъядерных серверах и 697 мс со шлюзами, реализованными в одноядерных серверах (18,3, 15,9 и 14,3 запросов в секунду соответственно). Как видно из графика, при превышении числа запросов в группе значения 30...40, рост кривых сильно замедляется, а при превышении значения 60...80 — прекращается вовсе, достигая предельной производительности.

Как видно из рис. 3, при обращениях к быстрому web-сервису (0,5 с) применение прокси-серверов существенно замедляет обработку практически при любом числе запросов в группе. При увеличении числа запросов это замедление возрастает и становится более чем двукратным. Характерно, что переход от использования одноядерных серверов в шлюзах к использованию четырехъядерных не дает значительного эффекта (это тем более удивительно, что аналогичный переход в сервере приложений повышает скорость обработки в несколько раз). По-видимому, это связано с тем, что основные временные затраты в работе прокси-сервера связаны с выполнением сетевых функций (прежде всего функции connect) и мало зависят от числа ядер и тактовой частоты.

На рис. 4 и 5 показаны аналогичные кривые для ситуаций, в которых время выполнения модельного web-сервиса составляет 1 и 2 с соответственно. Можно констатировать, что общие закономерности в поведении кривых сохраняются и в этом случае, но эффект замедления обработки вследствие применения шлюзов "HTTP → HTTPS" и "HTTPS → HTTP" заметно сглаживается.

На рис. 6, 7 и 8 приведены графики зависимости среднего времени выполнения информационного

запроса от числа запросов в пакете при обращении к тем же трем модельным сервисам и в тех же обозначениях. В отличие от рис. 3, 4 и 5, здесь все кривые демонстрируют устойчивый рост. Легко видеть, что и по этому важному показателю влияние шлюзов HTTP → HTTPS и "HTTPS → HTTP" является довольно ощутимым при обращении к более быстрому сервису (0,5 с) и менее ощутимым при обращении к более медленным сервисам (1 и 2 с). Например, при поступлении 20 одновременных запросов соотношение значений среднего времени обра-

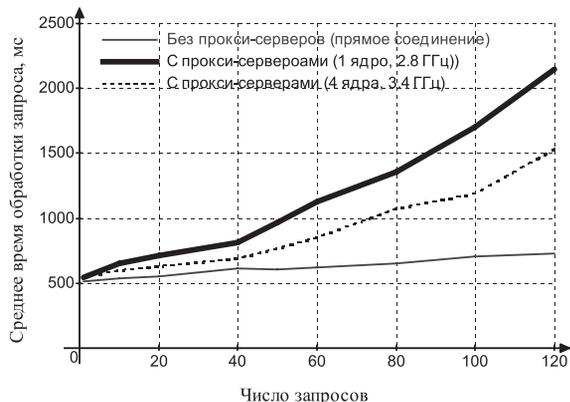


Рис. 6. График зависимости среднего времени обработки запроса от числа одновременных запросов при задержке web-сервиса 500 мс

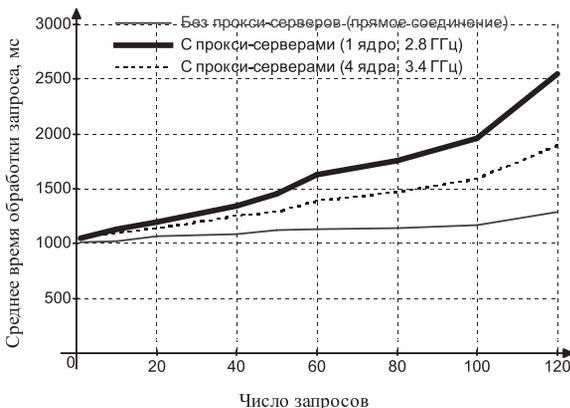


Рис. 7. График зависимости среднего времени обработки запроса от числа одновременных запросов при задержке web-сервиса 1000 мс

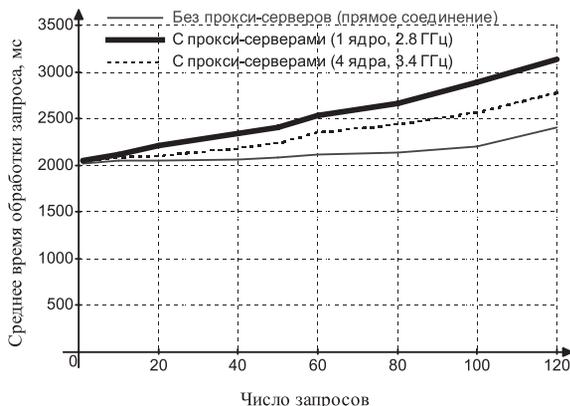


Рис. 8. График зависимости среднего времени обработки запроса от числа одновременных запросов при задержке web-сервиса 2000 мс

ботки запроса со шлюзами и без шлюзов равно 1,29 в первом случае, 1,12 во втором и 1,08 в третьем (при использовании одноядерных серверов в шлюзах).

Заключение

В целом результаты проведенных экспериментов позволяют сформулировать следующие выводы.

- Применение шлюзов "HTTP → HTTPS" и "HTTPS → HTTP" в целом сохраняет положительный эффект от многопоточного распараллеливания обработки.
- Построенный по вышеописанным принципам защищенный HTTPS-канал вполне позволяет поддерживать скорость обработки до нескольких десятков запросов в секунду, что обычно бывает достаточным для большинства информационно-управляющих систем.
- Чем больше время выполнения электронного сервиса, тем менее заметна задержка, вносимая шлюзами. В частности, в проведенных экспериментах наблюдалось заметное снижение скорости обработки при одновременном обращении сразу нескольких клиентов к быстрым (со временем выполнения менее 1 с) электронным сервисам через защищенный HTTPS-канал и не столь существенное при обращении к более медленным (со временем выполнения более 1 с).

Существенный недостаток описанного подхода заключается в том, что в пределах частной локальной сети данные передаются в открытом виде. Поэтому его применение ограничено ситуациями, в которых частные сети можно рассматривать как "зоны доверия", а единственным источником угроз является глобальная сеть. В этих условиях он может обеспечить "централизованную" защиту информационных запросов при обращении к относительно "медленным" электронным сервисам (со временем выполнения более 1 с) без значительного снижения скорости обработки.

Список литературы

1. Кришнамурти Б., Рэкфорд Д. Web протоколы. М.: Бинном, 2010. 592 с.
2. Шапошников И. В. Web-сервисы Microsoft.NET. СПб.: БХВ-Петербург, 2002. 336 с.
3. Мак-Дональд М., Шпуста М. Microsoft ASP.NET 3.5 с примерами на C# 2008 и Silverlight 2 для профессионалов. М.: Вильямс, 2009. 1408 с.
4. Згоба А. И., Маркелов Д. В., Смирнов П. И. Кибербезопасность: угрозы, вызовы, решения // Вопросы кибербезопасности. 2014. № 5. С. 30–38.
5. Щеглов А. В. Защита компьютерной информации от несанкционированного доступа. СПб.: Наука и техника, 2004. 384 с.
6. Асратян Р. Э., Лебедев В. Н., Орлов В. Л. Организация защищенных каналов взаимодействия на основе применения протокола HTTPS в прокси-серверах // Информационные технологии. 2015. Т. 21, № 9. С. 670–674.
7. Хант К. TCP/IP. Сетевое администрирование. СПб.: Питер, 2007. 816 с.
8. Андреев А. Г., Беззубов Е. Ю., Емельянов и др. Windows 2000: Server и Professional. СПб.: БХВ-Санкт-Петербург, 2001. 1055 с.

Efficiency of a Secure Communication Channels Based on the Use of the HTTPS in Proxy Servers

The article is devoted to experimental study of characteristics of the new method of secure interactions organization in multi-network environment. The method is based on the use of HTTPS protocol and proxy server technology. The essence of approach consists in the organization of the protected HTTPS connections between proxy servers for safe transfer of information queries from one private local network in another through a global network while client and server components of distributed system are released from of information protection functions. In this approach client and server programs use HTTP protocol and proxy servers have additional functions of "HTTP -> HTTPS" and "HTTPS -> HTTP" gateways. The main advantage of approach over known (for example, VPN) consists in providing "high-level" means of HTTP queries routing and filtration the based on headers of HTTP queries, but not on separate IP packages. The application of approach is limited to situations in which private local networks can be considered as "a trust zone", and the only source of threats is the global network. The experimental study of efficiency of the offered approach in terms of influence of protection gateways on the speed of electronic services calls processing is conducted. This research showed that the area of effective application of the approach is made by electronic services with performance time over 1–2 s.

Keywords: distributed systems, Internet technologies, network protocols, proxy servers, remote interactions, data security, web-services

References

1. Krishnamurti B., Reksford D. *Web protokoly*. Moscow: Binom, 2010. 592 p.
2. Shaposhnikov I. V. *Web-servisy Microsoft.NET*. SPb: BHV-Peterburg, 2002. 336 p.
3. Mak-Donal'd M., Shpushta M. *Microsoft ASP.NET 3.5 s primerami na C# 2008 i Silverlight 2 dlja professionalov*. Moscow: Vil'jams, 2009. 1408 p.
4. Zgoba A. I., Markelov D. V., Smirnov P. I. Kiberbezopasnost': ugrozy, vyzovy, reshenija, *Voprosy kiberbezopasnosti*, 2014, no. 5, pp. 30–38.
5. Shheglov A. V. *Zashhita komp'yuternoj informacii ot nesankcionirovannogo dostupa*. SPb.: Nauka i tehnika, 2004, 384 p.
6. Asratjan R.Je., Lebedev V. N., Orlov V. L. Organizacija za-shhishhennyh kanalov vzaimodejstviya na osnove primeneniya protokola HTTPS v proksi-serverah, *Informacionnye tehnologii*, 2015, vol. 21, no. 9, pp. 670–674.
7. Hant K. *TCP/IP. Setevoe administrirovanie*. SPb.: Piter, 2007. 816 p.
8. Andreev A. G., Bezzubov E. Ju., Emel'janov i dr. *Windows 2000: Server i Professional*. SPb.: BHV-Sankt-Peterburg, 2001. 1055 p.

УДК 004.45

В. В. Грибова, д-р техн. наук, зам. директора по научной работе, e-mail: gribova@iacp.dvo.ru,

А. В. Иванова, аспирант, e-mail: 2395146@gmail.com,

Институт автоматизации и процессов управления Дальневосточного отделения Российской Академии наук

Концепция программного комплекса для управления безопасностью информационных систем

Рассмотрены существующие подходы к созданию систем защиты, их достоинства и недостатки. Описаны основные требования, принципы и концептуальная архитектура программного комплекса для управления безопасностью информационных систем, состоящая из двух подсистем: среды управления информационными ресурсами и клиентской среды управления безопасностью информационной системы. Описаны основные компоненты программного комплекса и их функции.

Ключевые слова: информационная безопасность, управление, интеллектуальные системы, базы данных, онтологии, базы знаний, защита информации, информационные системы

Введение

Обеспечение защиты информации в информационных системах является очень актуальной задачей. Исследование компании Dell, проведенное в 2014 г., показало [1], что из 1440 опрошенных

компаний подавляющее большинство (90 % специалистов) отметили, что информационная безопасность становится одним из главных приоритетов в планируемом бюджете, и примерно 20 % всех затрат на информационные технологии (ИТ) будет

потрачено на всевозможные сервисы и продукты по защите информации. Три четверти опрошенных планируют увеличить расходы на информационную безопасность в этом году, причем 58 % из них планируют приобрести недешевые системы информационной защиты. 40 % ИТ-профессионалов все еще сомневаются в эффективности существующих решений. Общие взгляды на управление информационной безопасностью в целом и без учета специфических технических особенностей защищаемых платформ сформированы достаточно давно и закреплены в стандарте [2], а такие новые понятия, как "виртуализация", "облака", "кластеры", "распределенные вычисления" и т.д., только начинают вхождение в законодательный фундамент.

Можно выделить два основных подхода к защите информации в информационных системах всех типов [3].

Первый подход заключается в использовании единого средства защиты. Такое решение комплексно решает проблему обеспечения безопасности информационной системы (ИС)¹, обладает централизованным интерфейсом, что упрощает управление для администратора ИС [4], однако имеет и ряд существенных недостатков [5]: отсутствие совместимости с другими системами защиты, высокая стоимость конечного решения, невозможность отключения и последующей замены отдельных элементов на решения от другого разработчика (к примеру, отключение антивируса из комплекса и установка другого более репутационного антивируса или имеющего более высокий сертификат на класс выше), нарушение работоспособности всех подсистем системы защиты информации по окончании срока действия сертификата, компрометация лицензионного ключа, сбой процедур обновления и т.д.

Второй, и наиболее популярный, подход заключается в использовании комплекса средств защиты информации (СЗИ). Отдельные компоненты СЗИ специфичны, узконаправлены, предназначены для нейтрализации сходных угроз и могут быть разделены по типам: антивирусные средства, СЗИ от несанкционированного доступа, межсетевые экраны, системы обнаружения вторжений и др. Для обеспечения защиты ИС может быть выбран различный набор таких средств. При всей гибкости такого подхода он также обладает набором существенных недостатков [5]: отсутствие единого интерфейса управления и механизмов оперативного контроля (необходимо напрямую вмешиваться в настройки СЗИ), возможная несовместимость СЗИ между собой (после установки одного СЗИ другое может перестать функционировать, либо обнаруживаются конфликты совместного использования). Кроме того, перед администратором возникает ряд дополни-

¹ Под информационной системой будем понимать любую информационную систему вне зависимости от типа платформы (локально-вычислительная сеть, автоматизированное рабочее место, распределенная облачная платформа и т.д.).

тельных задач по обеспечению информационной безопасности ИС: ему необходимо знать и определять требования к ИС, в том числе законодательные, для правильного выбора набора СЗИ, их последующей настройки, поддержания непрерывного функционирования системы, разрешения конфликтных ситуаций, изучения возможностей улучшения системы и повышения ее эффективности [6].

В связи с тем, что подход, основанный на использовании комплекса разнородных СЗИ, применяется в подавляющем большинстве ИС, актуальной задачей является разработка программных средств, обеспечивающих администратора ИС единой средой для управления и мониторинга разнородными СЗИ, рекомендациями по их выбору и настройке в соответствии с конфигурацией ИС и ее конкретными особенностями. Такое решение обеспечит высокую эффективность технологий выявления, предупреждения и предотвращения атак и выполнение законодательных требований. Целью данной работы является разработка общих принципов, концепции и архитектуры комплекса программных средств, обеспечивающих такую функциональность.

Принципы построения программного комплекса

На основе анализа литературы, опыта работы в профилированной компании, приобретенных практических навыков, с учетом условий современной стадии развития технологий защиты информации, преимуществ и недостатков подхода, основанного на использовании набора разнопрофильных узконаправленных СЗИ, а также законодательных требований к защите ИС, можно выделить ключевые требования к функциональности и использованию комплекса программных средств.

- Программный комплекс должен обеспечивать администратора конкретной ИС помощью в определении класса ИС, требований к ней, а также рекомендациями по выбору наборов СЗИ с учетом конфигурации ИС.
- Набор СЗИ для конкретной ИС должен включать рекомендации по настройке, обеспечивающие бесконфликтность их функционирования в ИС.

Выбор СЗИ в соответствии с конкретной конфигурацией ИС и требованиями к уровню ее защиты, а также настройка каждого СЗИ таким образом, чтобы не возникало конфликтов совместимости между различными СЗИ, являются достаточно сложной задачей для любого администратора и часто требуют приглашения сторонних специалистов, как правило, из профилированных компаний, которые осуществляют функции консультирования по определению набора СЗИ в соответствии с конфигурацией ИС и требованиями к ее защите, настройку комплекса СЗИ, а также сопровождение в процессе эксплуатации.

- Программный комплекс должен поддерживать функцию анализа и обнаружения возникших конф-

ликтных ситуаций при функционировании разнородных СЗИ в единой ИС.

При функционировании разнородных СЗИ возможно возникновение конфликтных ситуаций. Их анализ, обнаружение, информирование администратора о таких конфликтах и возможных способах их устранения поможет администратору оперативно решать возникающие в процессе эксплуатации ИС проблемы.

• **Наличие единого интерфейса для управления разнородными СЗИ.** Такой подход обеспечит администратора ИС средствами оперативного управления набором разнородных СЗИ для централизованного сбора и отображения сведений о текущем состоянии ИС, полученных от различных СЗИ.

Учитывая, что архитектура ИС, требования к обеспечению их защиты, сами СЗИ и их функции постоянно изменяются, предлагаются следующие **основные принципы** создания программного комплекса:

1. Представление информации о стандартах, требованиях к ИС и их классификации, СЗИ и их функциях, методах устранения конфликтов СЗИ в форме, одинаково понятной как экспертам для модификации, исправления ошибок и неточностей, а также дальнейшего развития в связи с изменениями в данной предметной области, так и компьютерным системам для их обработки.

2. Создание централизованной "среды", аккумулирующей знания в едином информационном пространстве с целью обеспечить их накопление, совместное развитие и использование для исключения повторной разработки готовых решений и минимизации человеческих ресурсов в существующих условиях кризиса высококвалифицированных специалистов в области информационной безопасности и сокращения расходов на сопровождение комплекса СЗИ.

3. Размещение всех информационных ресурсов и сервисов в облаке для обеспечения их широкой доступности как для использования, так и модифицирования в процессе жизненного цикла.

4. Использование методов искусственного интеллекта для формирования и сопровождения данных и знаний. В качестве таких методов предлагается онтологический подход, обеспечивающий пользователя системой понятий (онтологией) для формирования целевых баз данных и баз знаний и исключающий необходимость использования посредников в лице инженеров знаний для преобразования опыта и навыков экспертов в форматы хранения, "понимаемые" программными системами.

Концептуальная архитектура программного комплекса

Концептуальная архитектура программного комплекса для управления безопасностью ИС представлена на рис. 1.

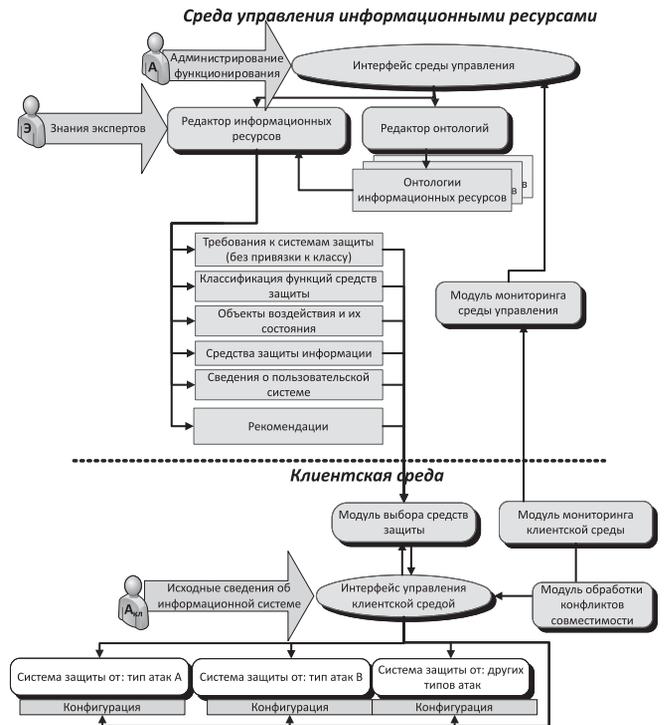


Рис. 1. Концептуальная архитектура программного комплекса для управления безопасностью информационных систем

Программный комплекс состоит из среды управления информационными ресурсами и множества клиентских сред управления безопасностью ИС.

Среда управления информационными ресурсами состоит из программных и информационных ресурсов. Информационными ресурсами являются онтологии, базы данных и базы знаний, программными — структурные редакторы онтологий, база данных и база знаний, а также "Модуль мониторинга среды управления", который предназначен для сбора, систематизации и анализа служебной информации обо всех подключенных к глобальной среде клиентских средах управления безопасностью ИС и выполняемых ими функциях. Данная информация необходима для получения новых знаний о совместном функционировании разнородных СЗИ для выработки рекомендаций по их настройке, устранению конфликтных ситуаций при совместном функционировании.

Редактор онтологий предназначен для создания онтологий баз данных и баз знаний, по которым эксперты предметной области с помощью редактора будут формировать и сопровождать соответствующие информационные ресурсы — базы данных и базы знаний. Информационные ресурсы используются в клиентских системах для помощи администратору в классификации ИС, определении требований к ней, выборе и настройке СЗИ, обеспечении их бесконфликтной работы. Информационные ресурсы состоят из множества баз данных и баз знаний. Базами данных являются: "Требования к системам защиты (без привязки к классу)",

"Классификация функций средств защиты", "Объекты воздействия и их состояния", "Средства защиты информации", "Сведения о пользовательской системе". Базы знаний содержат знания об информационных системах, их классификации, а также знания о разрешении конфликтов совместного функционирования СЗИ. Подробное описание информационных ресурсов дано в разделе "Информационные ресурсы".

В качестве средства реализации среды управления информационными ресурсами выбрана облачная платформа IASaaS [7]. Она представляет собой программно-информационный Интернет-комплекс для обеспечения поддержки разработки, управления и удаленного использования прикладных и инструментальных (системных) мультиагентных облачных сервисов (прежде всего интеллектуальных) и их компонентов. Комплекс основан на технологии облачных вычислений и обеспечивает удаленный доступ конечным пользователям к интеллектуальным системам, а разработчикам и администраторам — к средствам создания интеллектуальных систем и управления ими. В состав платформы входит универсальный двухуровневый редактор для формирования данных и знаний. Он предназначен для описания метаданных — онтологии базы данных или базы знаний (первый уровень), по которой затем автоматически генерируется специализированный интерфейс для эксперта (второй уровень). Наличие универсального редактора, во-первых, не требует разработки специализированных редакторов информационных ресурсов, во-вторых, позволяет экспертам предметной области без инженеров знаний формировать и модифицировать базы знаний и базы данных в терминах онтологии.

Все информационные ресурсы (метаданные и объектная информация) представляются в форме семантической сети понятий (иерархического однородного бинарного ориентированного графа с возможными петлями и циклами). Таким образом, использование платформы IASaaS обеспечивает всю необходимую инфраструктуру для хранения и наполнения фонда информационных ресурсов.

Клиентская среда управления безопасностью информационной системы (клиентская среда) предназначена для мониторинга состояния подконтрольных СЗИ, оперативного централизованного управления ими (без необходимости внесения изменений администратором в каждое СЗИ), а также для определения требований к клиентской ИС, формированию минимального набора требуемых функций и СЗИ, реализующих их выполнение. Доступ к данным функциям реализуется через *Интерфейс управления клиентской средой*.

Программными элементами комплекса помимо редакторов являются: *Модуль выбора средств защиты*, *Модуль мониторинга клиентской среды*, *Модуль обработки конфликтов совместимости*.

Модуль выбора средств защиты предназначен для определения набора требований на основании входных параметров клиентской ИС. Данный модуль также предусматривает помощь в выборе СЗИ, которые удовлетворяют предъявленным требованиям и могли бы быть использованы в конкретной ИС.

Модуль мониторинга клиентской среды предназначен для сбора актуальных сведений о состоянии системы защиты ИС. Указанный модуль взаимодействует со встроенными средствами аудита средств защиты информации и *Интерфейсом управления клиентской средой*, который позволяет собирать уточненные данные, основанные на типе СЗИ. Например, функции аудита, доступные в межсетевом экране, позволяют собрать более расширенную статистику о трафике, чем аналогичные функции аудита в антивирусном средстве. Помимо расширенной статистики осуществляется сбор информации об ошибках в системе, а также о произведенных администратором клиентской ИС описаниях параметров конкретной системы.

Модуль обработки конфликтов совместимости предназначен для выявления конфликтных ситуаций совместного функционирования СЗИ по собранной *Модулем мониторинга клиентской среды* информации об ошибках в системе. Ввиду того, что не каждая ошибка является результатом конфликта настройки или конфликта совместимости СЗИ, система осуществляет анализ полученной информации и в случае выявления конфликта уведомляет о событии администратора клиентской среды через *Интерфейс управления клиентской средой*. Также указанный модуль предусматривает определение специфики конфликта и возможные варианты его решения: в автоматическом либо интерактивном режиме, о чем также уведомляет администратора клиентской среды. В случае "автоматического" решения конфликта требуемые изменения параметров вносятся через доступные элементы управления для конкретного СЗИ (командная строка, изменения значений реестра и т.д.) на основе "Базы знаний о конфликтах". При выборе интерактивного режима решения конфликта администратору клиентской среды выдаются рекомендации по внесению изменений в настройки через штатный интерфейс СЗИ на основе "Интерфейса управления клиентской средой".

Разрабатываемый инструментальный комплекс предусматривает устранение конфликтных ситуаций в клиентской информационной системе в автоматическом режиме при наличии такой возможности, а в случае ее отсутствия — выдачу рекомендаций в *Интерфейс управления клиентской средой* для внесения изменений в настройки в ручном режиме.

Информационные ресурсы

Информационными ресурсами являются базы данных, базы знаний и их онтологии.

Блок базы данных (БД) включает в себя следующие информационные ресурсы:

1. **"Сведения о пользовательской системе"** описывает структуру данных, по которой администратор может описать характеристики клиентской информационной системы (рис. 2), необходимые для определения требований к ней.

- 1. Характер обрабатываемой информации
 - ⇨ 1.1. Персональные данные
 - ⇨ 1.2. Конфиденциальная информация
 - ⇨ 1.3. Для служебного пользования
- 2. Число автоматизированных рабочих мест
- 3. Расположение в сети
 - ⇨ 3.1. В составе ЛВС
 - ⇨ 3.2. Автономное рабочее место

Рис. 2. Фрагмент базы данных "Сведения о пользовательской системе"

2. **"Требования к системе защиты (без привязки к классу)"** включает в себя формализованное описание перечня законодательных и пользовательских требований к защите ИС. Используется при определении требуемого набора функций защиты клиентской ИС на основе ее конфигурации (рис. 3).

- 1. Наличие межсетевого экрана
 - ⇨ 1.1. 3 класса
 - ⇨ 1.2. 4 класса
- 2. Наличие средства защиты от несанкционированного доступа
 - ⇨ 2.1. 3 уровень контроля недеklarированных возможностей
 - ⇨ 2.2. 4 уровень контроля декларированных возможностей
- 3. Наличие модуля доверенной загрузки
- 4. Классификация информационной системы
 - ⇨ 4.1. 1Г
 - ⇨ 4.2. 2Б
 - ⇨ 4.3. первый уровень защищенности персональных данных
 - ⇨ 4.4. второй уровень защищенности персональных данных
 - ⇨ 4.5. третий уровень защищенности персональных данных

Рис. 3. Фрагмент базы данных "Требования к системам защиты (без привязки к классу)"

3. **"Классификация функций средств защиты"** является многоуровневой структурой и включает в себя классификацию функций СЗИ (рис. 4), определяемую нормативными документами [8], а также дополнительные функции, встраиваемые разработчиками в программное обеспечение. Используется при выявлении конфликтов, связана с БД "Средства защиты информации", которая содержит сводную информацию о всех функциях СЗИ.

- 1. Межсетевые экраны
 - ⇨ 1.1. Регистрация событий
 - ⇨ 1.1.1. Вход администратора межсетевого экрана в систему
 - ⇨ 1.1.2. Выход администратора межсетевого экрана из системы
 - ⇨ 1.1.3. Загрузка системы
- 2. Средства защиты от несанкционированного доступа
 - ⇨ 2.1. Аудит
 - ⇨ 2.1.1. Генерация записи аудита для следующих событий, потенциально подвергаемых аудиту
 - ⇨ 2.1.1.1. Запуск и завершение выполнения функций аудита
 - ⇨ 2.1.1.2. Все события, подвергаемые аудиту на уровне аудита
 - ⇨ 2.1.1.2.1. Минимальный
 - ⇨ 2.1.1.2.2. Базовый

Рис. 4. Фрагмент базы данных "Классификация функций средств защиты"

- 2. Secret Net
 - ⇨ 1.1. Средство защиты от несанкционированного доступа
 - ⇨ 1.2. Функции: 2.1., 2.1.1., 2.1.1.1., 2.1.1.2., 2.1.1.2.1., 2.1.1.2.2.*
 - ⇨ 1.3. Уровень контроля недеklarированных возможностей: 3

* Функции из базы данных "Классификация функций средств защиты"

Рис. 5. Фрагмент базы данных "Средства защиты информации"

4. **"Средства защиты информации"** содержит информацию о наборе функций конкретных СЗИ, представленных на рынке сертифицированных продуктов для обеспечения информационной безопасности (Kaspersky Endpoint Security, Dr. Web, SecretNet, TrustAccess и др.). Используется для определения перечня СЗИ, необходимых для установки в клиентской ИС, для определения набора функций, которые фактически будут активны с привязкой к конкретным СЗИ (рис. 5).

Структура БД имеет порядковую нумерацию, в качестве вложений используются цифровые обозначения из других БД, в частности, из БД "Классификация функций средств защиты" вместо текстового наименования функции используется ее цифровой код. Так, в п. 1.1 тип СЗИ указывается в соответствии с БД "Классификация функций средств защиты" (Пример: СЗИ от НСД; антивирусное средство или межсетевой экран), в п. 1.2 перечисляются только цифровые обозначения имеющихся функций (рис. 5).

Часть функций, встроенных в качестве обязательных в одно СЗИ, может использоваться в качестве дополнительных функций в СЗИ другого профиля защиты.

5. **"Объекты воздействия и их состояния"** включает в себя перечень возможных объектов воздействия, их атрибуты и допустимые значения функций СЗИ над этими объектами (рис. 6).

1. Сетевой порт (номер)

⇨1.1. Функции: (2.1., 2.1.1., 2.1.1.1., 2.1.1.2., 2.1.1.2.1.)

⇨1.2. Состояния

⇨1.2.1. Открыт

⇨1.2.2. Закрыт

Рис. 6. Фрагмент базы данных "Объекты воздействия и их состояния"

Блок "Базы знаний" (БЗ) включает в себя базу знаний о классификации и базу знаний о конфликтах.

1. **База знаний о классификации** содержит знания о классификации пользовательской информационной системы на основе ее конфигурации (рис. 7). Класс ИС необходим для определения требований к информационной системе, определения набора СЗИ и их функций. Классификация определяется на основе регламентирующих документов [9, 10].

```
<режим доступа> =многопользовательский AND
<конфиденциальность носителей информации> = различная AND
<уровень полномочий> = различный AND
<наивысший уровень конфиденциальности информации> =персональные данные AND
<характер персональных данных> = (сотрудники оператора OR внешние контрагенты) AND
<категория персональных данных> = специальная AND
<число субъектов персональных данных> > [100 000] AND
<тип актуальные угрозы> = (третий OR второй) =>
<класс системы> = 1B

(<наивысший уровень конфиденциальности информации>= персональные данные AND
<характер персональных данных> = сторонние контрагенты AND
<категория персональных данных> = общедоступные AND
<количество субъектов персональных данных> < [100 000] AND
<тип актуальные угрозы> = третий =>
<класс системы> = У34
```

Рис. 7. Фрагмент базы знаний о классификации

2. **База знаний о конфликтах** содержит знания о конфликтных ситуациях в конкретных СЗИ с учетом специфики произошедшего события и конфигурации ИС: разрядность и версия операционной системы (ОС), наличие выхода в Интернет, установленное программное обеспечение и др. Очевидно, что не каждая ошибка является результатом конфликта настройки или конфликта совместимости СЗИ. Для возникновения конфликта необходимо, чтобы два различных процесса (СЗИ, штатный процесс ОС и др.) в один момент времени запросили доступ к одному объекту доступа, в результате которого система повела себя таким образом, который не предусмотрен установленной политикой безопасности: отказ при предусмотренном предоставлении прав либо, наоборот, предоставление доступа при предусмотренном запрете. Ввиду того, что при возникновении конфликтной ситуации система может повести себя нештатным образом (рис. 8), фактический результат выполнения операции может не соответствовать зафиксированному журналами ОС и журналами СЗИ. Из рисунка видно, что в один момент времени (22:28:54) в систему осуществлено несколько мгновенных попыток доступа ("Открытие каталога"), результатом каждой из которых являются две операции "Закрытие объекта", т.е. фактически доступ к объекту не предоставлен несмотря на фиксацию ус-

Время	Пользователь	Файл	Результат	Операция
21.05.2015 22:28:54	Администратор	Flash(32400757):\	OK	Закрытие объекта
21.05.2015 22:28:54	Администратор	Flash(32400757):\	OK	Закрытие объекта
21.05.2015 22:28:54	Администратор	Flash(32400757):\	OK	Открытие каталога
21.05.2015 22:28:54	Администратор	Flash(32400757):\	OK	Закрытие объекта
21.05.2015 22:28:54	Администратор	Flash(32400757):\	OK	Закрытие объекта
21.05.2015 22:28:54	Администратор	Flash(32400757):\	OK	Открытие каталога
21.05.2015 22:28:54	Администратор	Flash(32400757):\	OK	Закрытие объекта
21.05.2015 22:28:54	Администратор	Flash(32400757):\	OK	Закрытие объекта
21.05.2015 22:28:54	Администратор	Flash(32400757):\	OK	Открытие каталога
21.05.2015 22:28:54	Администратор	Flash(32400757):\	OK	Закрытие объекта

Рис. 8. Нештатное поведение системы защиты

```
<число процессов> = 2 AND
<процесс 1 > = avr.exe AND
<процесс 2> = dl.exe AND
<время. процесс1> = <время. процесс 2> AND
<объект воздействия процесс 1> = локальный OR устройство OR внешнее устройство OR
USB устройство OR съемный носитель AND
<объект воздействия процесс 1> = <объект воздействия процесс 2>
<название операции. процесс 1> = чтение AND
<название операции. процесс 2> = запись => <состояние конфликта> = потенциальный
конфликт
```

Рис. 9. Фрагмент базы знаний о конфликтах

пешного результата операции. Для точного выявления конфликтов необходимо предусматривать такие ситуации. Для этого вводится понятие потенциального конфликта: ситуация, когда два различных процесса в один момент времени запрашивают доступ к одному объекту доступа.

Параллельно анализируя другие системные журналы, можно сделать вывод о том, является ли потенциальный конфликт фактическим конфликтом. Если результат операции доступа двух процессов к одному объекту в один момент времени соответствует заданной политике доступа, то потенциальный конфликт *не является конфликтом*.

Потенциальный конфликт может возникнуть при настройке схожих функций защиты различных СЗИ, установленных в одной информационной системе. К примеру, функция контроля доступа есть во многих СЗИ и выполняет не только операцию чтения прав доступа к объекту, но и операцию записи в случае их изменения. Конфликт может возникнуть в ситуации, когда к одному объекту воздействия будет осуществляться доступ нескольких функций в целях установки противоположных (противоречащих друг другу) состояний объекта. Результат этой операции может зависеть от множества факторов: приоритетов назначенных прав, режимов доступа (дискретный, мандатный), типов СЗИ, конкретных объектов доступа и т. д. Знания об устранении таких конфликтов описаны в базе знаний о конфликтах (рис. 9).

Программные компоненты

Основными программными компонентами являются редакторы информационных ресурсов, модули мониторинга, а также модули выбора СЗИ и обработки конфликтов совместимости. Редакторы

информационных ресурсов являются компонентами платформы IASaaS, их функциональность и особенности подробно изложены в работе [11]. Функциональность модулей мониторинга достаточно понятна, поэтому в данном разделе более детально описаны два программных компонента — модуль выбора средства защиты информации и модуль обработки конфликтов совместимости.

Модуль выбора средств защиты информации последовательно решает следующие основные задачи:

- определение множества требований к ИС на основе ее конфигурации;
- определение множества функций СЗИ и их параметров, которые должны функционировать в конкретной ИС;
- перечень СЗИ, которые требуется установить, с указанием возможных вариантов, а также с учетом эффективности тех СЗИ, которые уже установлены в ИС;
- параметры СЗИ, которые необходимо активировать.

Стоит отметить, что на основе сформированной администратором клиентской среды конфигурации информационной системы нельзя сразу определить исчерпывающий перечень требований. Некоторые требования являются причинно-следственными. К примеру: в конфигурации ИС заявлено, что она подключена к сети. В качестве требования следствием будет "1. Межсетевой экран". Однако определить требуемый класс межсетевого экрана можно только на основе классификации информационной системы. Если указано, что характер обрабатываемой информации — персональные данные, можно присвоить один из уровней защищенности, к примеру, "4.5. Третий уровень защищенности" (конкретизация определяется также и другими параметрами конфигурации ИС). Используя установленный класс, а также дополнительные сведения о конфигурации, можно определить требуемый класс межсетевого экрана для конкретной ИС — "1.2 4 класс", а также дополнительные требования к устанавливаемому ПО — "2.2. 4 уровень контроля НДВ". Соответствия между конфигурацией ИС и требованиями к этой ИС описываются в БЗ "О классификации". Таким образом, можно сказать, что эта база последовательно и циклично использует определяемые характеристики для получения исчерпывающего перечня требований к ИС. Общая схема алгоритма работы модуля выбора СЗИ представлена на рис. 10.

Модуль обработки конфликтов совместимости предназначен для выявления конфликтных ситуаций. Общий алгоритм модуля представлен на рис. 11. На первом шаге работы алгоритма определяется наличие потенциального или истинного конфликта. Задачей второго шага работы алгоритма является выдача рекомендаций администратору по устранению возникшего конфликта либо предупреждение о потенциальной возможности конфликта с

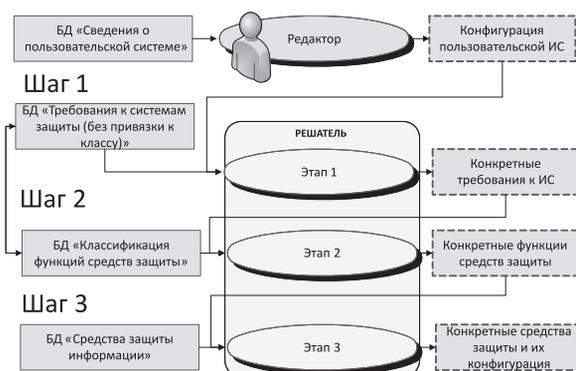


Рис. 10. Алгоритм работы модуля выбора СЗИ

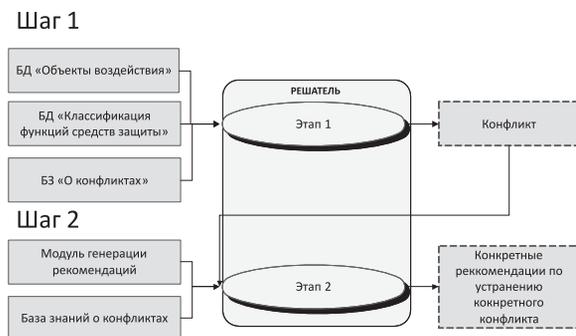


Рис. 11. Алгоритм работы модуля обработки конфликтов совместимости

<p>Формальное описание В рамках назначенной политики [значение политики] для [название процесса] по операции [название операции] для объекта [название объекта] необходимо выполнить следующие действия: [описание действий, схемы, картинки, инструкции]</p> <p>Фактическое представление В рамках назначенной политики [Разрешительная] для [dl.exe] по операции [Запись] для объекта [Prod&220HW] необходимо выполнить следующие действия: [Открыть консоль администратора Dallas...]</p>

Рис. 12. Структура описания рекомендаций

объяснением, какие СЗИ, процессы, операции и др. могут быть источниками конфликтов.

Генерация рекомендаций осуществляется соответствующим модулем, структура выдачи рекомендаций представлена на рис. 12.

С учетом того, что пользователь может несвоевременно осуществлять настройку системы защиты в соответствии с рекомендациями, предлагается использовать механизм выявления дубликатов конфликтов. Для этого при взаимодействии модуля мониторинга клиентской среды и модуля обработки конфликтов совместимости выявляются такие конфликты, которые полностью дублируют друг друга лишь с разницей во времени.

Заключение

В настоящей статье рассмотрены существующие подходы к созданию систем защиты (комплекс СЗИ и монолитные решения), достоинства и недостатки указанных подходов. Обосновывается необходимость создания программного обеспечения, объединяющего разнородные СЗИ в единую сис-

тему. Описаны основные требования, принципы и концептуальная архитектура программного комплекса, состоящая из двух подсистем: среды управления информационными ресурсами и клиентской среды управления безопасностью информационной системы. Описаны основные компоненты указанных сред и их функции.

На сегодняшний день в рамках решения данной задачи разработаны базы данных, ведется наполнение баз знаний и реализация программных компонентов программного комплекса на платформе IACPaaS.

Работа выполнена при частичной финансовой поддержке РФФИ, грант 16-07-00340, программы "Дальний Восток".

Список литературы

1. **Securing the cloud in a BYOD world: "No one really has this figured out"** [Электронный ресурс]. Business Cloud News. Лондон. 2014. URL: <http://www.businesscloudnews.com/2014/02/21/securing-the-cloud-in-a-byod-world-no-ones-really-has-this-figured-out/>
2. **Управление** технического регулирования и стандартизации Федерального агентства по техническому регулированию и метрологии, ГОСТ Р ИСО/МЭК 27001—2006. Государственный стандарт. 2006. 31 с.
3. **PCMagazine**. Курс лекций "Вирусы и борьба с ними" // Лаборатория Касперского. 2007. [Электронный ресурс]. URL: http://www.pcmag.ru/elearning/course/lesson.php?COURSE_ID=10&ID=62

http://www.pcmag.ru/elearning/course/lesson.php?COURSE_ID=10&ID=62

4. **Код безопасности**. Security Studio Endpoint Protection. Сертифицированная защита компьютера от сетевых вторжений, вредоносных программ и спама // Код безопасности — 2015. [Электронный ресурс]. URL: http://www.securitycode.ru/products/security_studio_endpoint_protection/
5. **Rhodes-Ousley M.** Information Security. The Complete Reference, Second Edition. California: Silicon Valley, 2014. P. 578—595.
6. **Меры** защиты информации в государственных информационных системах. М.: Федеральная служба по техническому и экспортному контролю (ФСТЭК России), 2014. P. 5—15.
7. **Gribova V. V., Kleshchev A. S., Krylov D. A.** Project IACPaaS. Complex for intelligent software based on cloud computing // Artificial Intelligence and Decision Making. 2011. N. 1. P. 27—35.
8. **Документы** по сертификации средств защиты информации и аттестации объектов информатизации по требованиям безопасности информации. М.: Федеральная служба по техническому и экспортному контролю. России — 2015. [Электронный ресурс]. URL: <http://fstec.ru/tehnicheskaya-zashchita-informatsii/dokumenty-po-sertifikatsii/120-normativnye-dokumenty>.
9. **Автоматизированные** системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации. Руководящий документ. Решение председателя Гостехкомиссии России от 30 марта 1992 г.
10. **Об утверждении** требований к защите персональных данных при их обработке в информационных системах персональных данных. Постановление Правительства РФ от 01.11.2012 № 1119.
11. **Грибова В. В., Клешев А. С.** Технология разработки интеллектуальных сервисов, ориентированных на декларативные предметные базы знаний. Часть I. Информационные ресурсы // Информационные технологии. 2013. № 9. С. 7—11.

V. V. Gribova, Research Deputy Director, e-mail: gribova@iacp.dvo.ru,

A. V. Ivanova, e-mail: 2395146@gmail.com, PhD Student,

Institute of Automation and Control Processes for Eastern Branch of the Russian Academy of Sciences

Software for Security Control of Information Systems

This article reviewed modern approaches to the creation of information security systems, the advantages and disadvantages of these approaches. The motivation to create a software that incorporates heterogeneous information security systems into an integrated system is explained. The basic requirements, principles and architecture of software including two subsystems are described. These subsystems are: a subsystem for control of information resources and a client subsystem for security control of an information system. The subsystem for control of information resources is implemented on the cloud platform IACPaaS. The cloud platform is an Internet software for development, control and usage of intelligent services. The client subsystem is intended for monitoring the state of installed systems, as well as for determine the security requirements for the client information system and a security configuration tool in accordance with requirements. By now data and knowledge bases are realized, development of software components is in progress.

Keywords: security system, intelligent system, databases, ontologies, knowledge bases, data protection, information systems, data security solution

References

1. **Securing the cloud in a BYOD world: "No one really has this figured out"** [Electronic resource]. Business Cloud News — London, 2014. URL: <http://www.businesscloudnews.com/2014/02/21/securing-the-cloud-in-a-byod-world-no-ones-really-has-this-figured-out/>.
2. **Управление** технического регулирования и стандартизации Федерального агентства по техническому регулированию и метрологии *GOST R ISO MEK 27001—2006*. Gosudarstvennyj standart. 2006. 31 p.
3. **PCMagazine**. Course of lectures "Viruses i bor'ba s nimi". Course of lectures "Viruses and opposing them". Moscow, 2007. URL: http://www.pcmag.ru/elearning/course/lesson.php?COURSE_ID=10&ID=62.
4. **Код** Безопасности. Security Studio Endpoint Protection. Сертифицированная защита компьютера от сетевых вторжений, вредоносных программ и спама, *Kod bezopasnosti*, 2015. [Elektronnyj resurs]. URL: http://www.securitycode.ru/products/security_studio_endpoint_protection/.
5. **Rhodes-Ousley M.** Information Security. The Complete Reference, Second Edition. Silicon Valley: California, 2014, pp. 578—595.
6. **Меры** защиты информации в государственных информационных системах, Moscow: Federal'naya sluzhba po tehnicheskomu i ehksportnomu kontrolyu (FSTEHK Rossii), 2014. pp. 5—15.

7. **Gribova V. V., Kleshchev A. S., Krylov D. A.** Project IACPaaS. Complex intelligent software based on cloud computing, *Artificial Intelligence and Decision Making*, 2011, no. 1, pp. 27—35.

8. **Документы** по сертификации средств защиты информации и аттестации объектов информатизации по требованиям безопасности информации, Федеральная служба по техническому и экспортному контролю, FSTEHK Rossii — 2015. [Elektronnyj resurs]. URL: <http://fstec.ru/tehnicheskaya-zashchita-informatsii/dokumenty-po-sertifikatsii/120-normativnye-dokumenty>.

9. **Автоматизированные** системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации. Руководящий документ. Решение председателя Гостехкомиссии России от 30 марта 1992 г.

10. **Об утверждении** требований к защите персональных данных при их обработке в информационных системах персональных данных. Постановление Правительства РФ от 01.11.2012 N 1119.

11. **Gribova V. V., Kleshchev A. S.** Technology of Intelligent Services Development Oriented on Declarative Domain Knowledge Bases. Part 1. Information Resources, *Information Technologies*, 2013, no. 9, pp. 7—11.

УДК 004.414.38

В. В. Подбельский, проф., проф. департамента программной инженерии, e-mail: vpodbelskiy@hse.ru,

О. В. Максименкова, мл. науч. сотр., e-mail: omaksimenkova@hse.ru,

К. С. Бабич, студент, e-mail: kirill.babich@mail.ru,

Национальный исследовательский университет "Высшая школа экономики", Москва

Об обеспечении интероперабельности программных средств поддержки тестирования в образовании

Обсуждаются вопросы обеспечения интероперабельности программных средств поддержки тестирования в образовании. Проведен анализ наиболее распространенных форматов представления тестовых заданий и программных средств конвертации тестовых заданий, представленных в различных форматах. По результатам анализа сформулированы требования к архитектуре конвертера тестовых заданий и разработан прототип web-API, отвечающий данным требованиям.

Ключевые слова: компьютерное тестирование, форматы тестовых заданий, конвертер тестовых заданий, QTI/API-совместимость

Введение

Проектирование и разработка программных средств поддержки учебного процесса и его частей в настоящее время является одной из актуальных задач программной инженерии. Уже много лет существуют средства поддержки учебного процесса *LMS (Learning Management System)*, платформы сопровождения дистанционного обучения, системы поддержки массовых открытых онлайн курсов *MOOC (Massive Open Online Courses)* и проч. Многие из названных многопрофильных средств включают в себя модули контроля знаний, зачастую представленные тестовыми системами или их фрагментами.

Отметим, что популяризация компьютерных форм тестирования тесно связана с развитием компьютерной техники и сетевых компьютерных технологий. Так, с 70-х годов XX века к настоящему времени рост вычислительных мощностей привел к развитию методов и алгоритмов, а также к внедрению в педагогическую практику компьютерного тестирования (*computer testing*) и компьютерного адаптивного тестирования (*computer adaptive testing*) [1, 2]. Это связано с возможностью использования для обработки результатов тестирований методов современной теории тестирования (*item response theory, IRT*) [3] и появившейся в начале XXI века и требующей еще более серьезных вычислительных мощностей теории тестлетов (*testlet response theory, TRT*) [4–6].

Существует много программных средств автоматизации процессов разработки и сохранения тестовых заданий, а также проведения тестирования и анализа результатов. Данное многообразие является

одной из причин проблем со свободным трансфером тестовых заданий между различными системами, что в свою очередь замедляет и усложняет миграцию пользователей и перенос данных на более современные платформы [7]. Подобная задача возникает, например, в тех случаях, когда уже разработанный банк тестовых заданий предполагается использовать при проведении тестирования в новой среде или в случае изменения инструментального программного обеспечения при неизменной системе проведения тестирования.

Программное обеспечение (ПО), связанное с педагогическим тестированием, нацелено на автоматизацию одного (или нескольких) из следующих процессов:

- разработка тестовых заданий;
- экспертиза и калибровка тестовых заданий;
- формирование банка тестовых заданий;
- трансфер тестов и тестовых заданий;
- проведение тестирования и доставка тестов (*test delivering*) и тестовых заданий (*test item delivering*)¹;
- анализ результатов тестирования.

Пользователи тестовых систем в большинстве своем не имеют квалификации программистов и не имеют возможности создавать собственные утилиты для преобразования тестовых заданий из формата в формат. Таким образом, вопросы поддержки процессов передачи тестовых заданий от системы к системе на данный момент являются весьма акту-

¹ Под доставкой тестов и тестовых заданий понимается форма представления тестовых материалов, например, бланковая или электронная, а также способ передачи этих материалов от администраторов тестирования к организаторам.

альными, особенно на уровне конечного пользователя. Кроме того, средства разработки тестовых заданий, создаваемые вне конкретной системы контроля знаний или *LMS*, могут быть усовершенствованы за счет современных технологий разработки.

Форматы представления тестовых заданий в компьютерных системах

Проблемы внутрисистемного представления тестовых заданий, а также повышения удобства импорта/экспорта тестов и тестовых заданий в системы тестового контроля знаний не новы. Известны и используются различные форматы представления тестовых заданий (табл. 1), достаточно подробная их подборка представлена на сайте [8].

Отметим, что большинство из них является частными форматами отдельных систем, нацеленными на возможность разработки тестовых заданий преподавателями в отдельных файлах и их импорта в тестовые системы, например *MoodleXML*, *WebCT*, *Blackboard*.

Отдельные форматы получили широкое распространение. Например, форматы, исходно разработанные для опросников (*quiz*) *LMS Moodle*, поддерживаются и другими *LMS* и тестовыми системами *AIKEN*, *GIFT*, *MoodleXML*.

Системное решение проблемы совместимости форматов тестовых заданий было предложено консорциумом *IMS Global learning* [16]. Специалиста-

ми *IMS* к 2012 г. была разработана спецификация *IMS Question and Test interoperability specification (QTI v2.1)* [15], регламентирующая совместимость тестов и тестовых заданий для банков и тестовых систем, разработанных участниками альянса *QTI/APIP*. Основанный на *QTI v2.1* протокол *Accessible Portable Item Protocol (APIP v1.0)* [17] представляет собой модель данных, стандартизирующую формат файлов для передачи (трансфера) тестовых заданий.

Несмотря на проработанность и наличие полноценного методического сопровождения, *QTI/APIP*-совместимость пока не стала повсеместной практикой. В настоящее время большинство продуктов обеспечивает *Learning Tools Interoperability (LTI v1.0)* и (*LTI v1.1*), т. е. обеспечивает возможность использования единого протокола для переноса учебных курсов между *LMS*. Однако среди более 100 разработчиков образовательного программного обеспечения, представленных на сайте *IMS*, совместимость на основе спецификации *QTI/APIP* в своих продуктах обеспечивают не более десяти.

Обзор программных средств конвертации форматов тестовых заданий

В настоящее время в свободном доступе находятся некоторые средства поддержки процесса разработки тестовых заданий, позволяющие экспортировать тестовые задания в распространенные форматы.

Таблица 1

Некоторые форматы представления тестовых заданий

Название формата	Краткое описание	Примеры поддерживающего ПО
AIKEN [9]	Текстовый формат <i>импорта</i> тестовых заданий альтернатив ¹	LMS Moodle, LMS E-Front
GIFT [10]	Текстовый UTF-8 формат <i>импорта</i> тестовых заданий, поддерживающий представление альтернатив, опций, отображений, а также простых редакторов (короткий текст и число)	LMS Moodle, LMS E-Front
MoodleXML [11]	XML-представление для <i>экспорта/импорта</i> тестовых заданий. Формат поддерживает те же виды тестовых заданий, что и GIFT, но допускает импорт изображений	LMS Moodle Quiz module
Blackboard [12]	Текстовый формат <i>импорта</i> тестовых заданий и элементов опросников. Поддерживает тестовые задания: альтернативы, опции, отображения, простые редакторы (короткий текст и число), а также несколько заданий комбинированного и специального вида, например шкала Ликерта	Blackboard, LMS Moodle
SCORM/AICC [13]	XML-формат <i>импорта/экспорта</i> данных между электронными учебными курсами, включающий, среди прочего, представление некоторых тестовых заданий: альтернатив, опций и простых редакторов (короткий текст)	LMS Moodle, Blackboard, IMS Content package in Cecil 7, SharePoint LMS, Scorm assessment engine
WebCT [14]	Текстовый формат <i>импорта</i> тестовых заданий: альтернатив, опций и простых редакторов (короткий текст, число, эссе). Формат предусматривает включение рисунков в текст заданий. Для короткого текста возможно использование нескольких пропусков	Blackboard, LMS Moodle
QTI/APIP [15]	Спецификация формата <i>импорта/экспорта</i> тестовых заданий, позволяет описать любое тестовое задание или задание опросника, в том числе задания, содержащие графику, мультимедиа, а также задания для адаптивного тестирования	Respondus 4.0, TAO v.2.6.5 with MathJax, ONYX Testsuite v5.3.1

¹В данной работе задания с выбором одного или нескольких верных ответов классифицируются как задания альтернативы (выбор ровно одного верного ответа) и задания опции (выбор нескольких верных ответов).

Среди русскоязычного программного обеспечения известен конвертер *AMV Converter* [<http://amv-tst.narod.ru>]. Конвертер поддерживает форматы *NT-II*, *JSTest*, *GIFT*, *HTML*, *DOC*, *MoodleXML* и в одной из первых версий — ПРОМЕТЕЙ. К сожалению, данное приложение не имеет сетевой версии, недоступно для скачивания с сайта и не обновляется с 2008 г.

Преобразовать задания с одним верным ответом, задания с несколькими верными ответами и задания с кратким ответом из текстового представления в формат *GIFT* можно на сайте *Moodle test creator* [<http://text2gift.host-ed.me/Text2GiftConverter.html>]. Для каждого задания размещенный на сайте скрипт создает отдельный текстовый файл с заданием, размеченным в *GIFT*-формате. Отметим, что типы тестовых заданий, доступных для преобразования из формата в формат, серьезно ограничены. Например, не поддерживаются такие виды заданий закрытого типа, как задания на установление соответствий и восстановление верной последовательности. Интерфейс *Moodle test creator* реализован на двух языках: английском и испанском, но доступных средств локализации не содержит.

Преобразование тестовых заданий из текста в формат *MoodleXML* может быть организовано на сайте [<http://vletools.com/>]. Преобразование может быть проведено как самостоятельно (зарегистрированным пользователем), так и с помощью службы поддержки. Конвертер бесплатно доступен для пользователей, имеющих Facebook-аккаунт, регистрация платная и составляет (на момент написания данной работы) 15 евро. При заказе услуги преобразования тестовых заданий из форматов *TXT*, *DOC* и *XLS/XLSX* стоимость в зависимости от сложности колеблется от 20 до 50 евроцентов. Конвертер поддерживает все задания закрытого типа, задания с кратким и развернутым ответами. Интерфейс сайта реализован только на английском языке и доступных средств локализации не содержит.

Поскольку формат *MoodleXML* предусмотрен не только для тестовых заданий, но и для глоссариев и учебных дисциплин, существуют специальные программные средства, например, расширение для *Microsoft Office* [http://sourceforge.net/projects/moodletoolkit/?source=typ_redirect] или редактор на сайте открытых образовательных ресурсов [<https://www.oercommons.org/community/moodle-xml-converter/view>], также экспортирующие учебные единицы в указанный формат. Отметим, что пока среди их инструментов отсутствуют специфические, связанные с тестовыми заданиями.

Краткий обзор ситуации с текущими реализациями показывает, что находящиеся в свободном доступе средства конвертации тестовых заданий имеют следующие ограничения:

- серьезно ограничен набор поддерживаемых типов тестовых заданий;

- большинство конвертеров не позволяет свободно преобразовать тестовые задания из одного формата в другой. Например, есть хорошие конвертеры заданий из текстового представления, не являющегося специальным форматом, в *MoodleXML* или *GIFT*, но невозможно преобразовать задание из формата *MoodleXML* в формат *GIFT* или в обратном направлении;
- рассмотренные программные средства не поддерживают русскоязычный интерфейс и имеют очень низкие возможности по его локализации.

Постановка задачи

С учетом ограничений рассмотренных в предыдущем разделе программ и актуальности задачи было принято решение разработать программное средство для конвертации тестовых заданий между форматами, соответствующее следующим требованиям:

- доступность через глобальную сеть Интернет;
- простота интерфейса и удобство воплощения основных сценариев;
- наличие русскоязычного интерфейса и возможностей локализации;
- поддержка как можно большего числа форматов тестовых заданий, распространенных среди российского академического сообщества;
- бесплатность и отсутствие скрытых и явных платных сервисов;
- открытость.

Web-приложение для конструирования тестовых заданий закрытого типа

В рамках решения поставленной задачи была спроектирована архитектура (рис. 1) на основе *RESTful web-API* [18], позволяющая легко добавлять новые форматы и типы тестовых заданий.

В приложении с рабочим названием *Конвертер тестовых заданий* (КТЗ) [<http://testconstructor.azurewebsites.net>] реализован первый прототип *web-API* для конструирования тестовых заданий с закрытым ответом и их экспорта в форматы *MoodleXML*, *GIFT* и *AIEKEN*. Форматы были выбраны с учетом

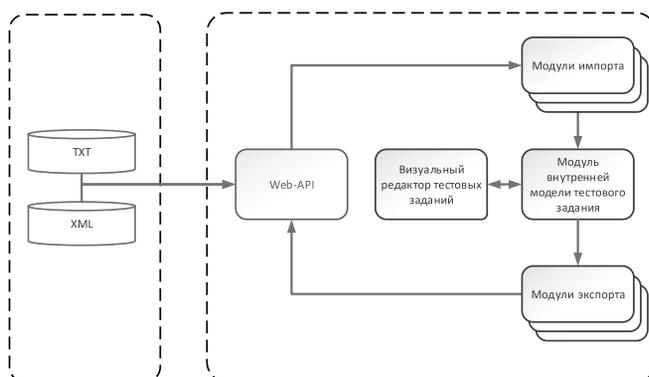


Рис. 1. Архитектура приложения

Таблица 2

Технологии, использованные при реализации прототипа

Цель использования	Технологии	Источники
Среда разработки	<i>Microsoft Visual Studio 2013</i>	https://msdn.microsoft.com/ru-ru/vstudio/default.aspx
Базовый язык разработки	<i>C#</i>	https://msdn.microsoft.com/enus/vstudio/hh341490.aspx
Web-инфраструктура	<i>ASP.NET MVC</i>	http://www.asp.net/mvc
Язык описания скриптов	<i>JavaScript</i>	https://www.javascript.com
WYSIWIG-редактор HTML-кода	<i>TinyMCE</i>	http://www.tinymce.com
Регламентирующие стандарты консорциума W3C	<i>HTML5, CSS3</i>	http://www.w3.org/TR/html5/ , http://www.w3.org/Style/CSS/

используемых в учебном процессе НИУ ВШЭ тестовых модулей *LMS Moodle* и *eFront*.

Основной акцент в прототипе был сделан на разработке редактора и визуальном представлении тестового задания в графическом интерфейсе пользователя (*Graphic User Interface, GUI*).

Технологии, использованные при создании прототипа

Для разработки прототипа была использована мощная, широко поддерживаемая и динамично развивающаяся web-инфраструктура *Microsoft ASP.NET MVC*. В рамках поставленной задачи существенно, что в *ASP.NET* заложены гибкие возможности по локализации и интернационализации.

Список основных технологий, использованных при разработке прототипа приложения, приведен в табл. 2.

Интерфейс с пользователем

При разработке интерфейса с пользователем реализован архитектурный шаблон *Model-View-Controller (MVC)*, поддерживаемый инфраструктурой *ASP.NET MVC*. Кроме того, интерфейс оптимизирован для современных браузеров посредством использования *HTML5* и *JavaScript*.

Пользовательский интерфейс прототипа построен вокруг двух главных сценариев использования: *создание нового тестового задания* и *преобразование существующего задания*.

Поддерживаемые прототипом типы тестовых заданий

Тестовые задания и ограничения для них соответствуют требованиям системы управления учебным процессом *Moodle*. Под тестовыми заданиями с закрытым ответом в рамках данного web-приложения подразумеваются [19] задания следующих типов:

1. *Множественный выбор* — задание, предусматривающее один поставленный вопрос и множество

Входные данные			
<i>Тестовое задание со множественным выбором в формате GIFT</i>			
<pre> \\ question: 1 name: Q1 ::Q1::[html]<p>Пробное тестовое задание со множественным выбором в формате GIFT</p>{ ~%100%<p>Неверный ответ</p> ~%100%<p>Неверный ответ</p> ~%50%<p>Верный ответ</p> ~%50%<p>Верный ответ</p> }</pre>			
Выходные данные			
<i>Результат преобразования в формат MoodleXML</i>			
<pre> <?xml version="1.0" encoding="utf-8"?> <quiz> <question type="category"> <category> <text /> </category> </question> <question type="multichoice"> <name> <text>Q2</text> </name> <questiontext format="html"> <text><![CDATA[<p>Пробное тестовое задание со множественным выбором в формате GIFT</p>]]></text> </questiontext> <generalfeedback format="html"> <text></text> </generalfeedback> <defaultgrade>1</defaultgrade> <penalty>0</penalty> <hidden>0</hidden> <single>false</single> <shuffleanswers>false</shuffleanswers> <answernumbering>123</answernumbering> <correctfeedback format="html"> <text></text> </correctfeedback> <partiallycorrectfeedback format="html"> <text></text> </partiallycorrectfeedback> <incorrectfeedback format="html"> <text></text> </incorrectfeedback> <shownumcorrect></shownumcorrect> <answer fraction="-100" format="html"> <text><![CDATA[<p>Неверный ответ</p>]]></text> <feedback format="html"> <text></text> </feedback> </answer> <answer fraction="-100" format="html"> <text><![CDATA[<p>Неверный ответ</p>]]></text> <feedback format="html"> <text></text> </feedback> </answer> <answer fraction="50" format="html"> <text><![CDATA[<p>Верный ответ</p>]]></text> <feedback format="html"> <text></text> </feedback> </answer> <answer fraction="50" format="html"> <text><![CDATA[<p>Верный ответ</p>]]></text> <feedback format="html"> <text></text> </feedback> </answer> </question> </quiz></pre>			
Входные данные			
<i>Тестовое задание «истина/ложь» в формате MoodleXML</i>			
<pre> <?xml version="1.0" encoding="utf-8"?> <quiz> <question type="category"> <category> <text /> </category> </question> <question type="truefalse"> <name> <text>Q3</text> </name> <questiontext format="html"> <text><![CDATA[<p>Задание истина/ложь в формате MoodleXML</p>]]></text> </questiontext> <generalfeedback format="html"> <text></text> </generalfeedback> <defaultgrade>1</defaultgrade> <penalty>0</penalty> <hidden>0</hidden> <answer fraction="100" format="moodle_auto_format"> <text>false</text> <feedback format="html"> <text></text> </feedback> </answer> <answer fraction="0" format="moodle_auto_format"> <text>true</text> <feedback format="html"> <text></text> </feedback> </answer> </question> </quiz></pre>			
Выходные данные			
<i>Результат преобразования в формат GIFT</i>			
<pre> \\ question: 1 name: Q3 ::Q3::[html]<p>Задание истина/ложь в формате MoodleXML</p>{ FALSE }</pre>			

Рис. 2. Пример входных и выходных данных

ответов на него. Число правильных ответов не ограничено, однако каждому ответу должен быть присвоен свой собственный вес — положительное или отрицательное значение, измеряемое в процентах из диапазона $[-100; 100]$, показывающее часть общего балла за задание, выставляемого при проверке.

2. *Выбор одного правильного ответа* — задание, которое является производным от *Множественного выбора*, но имеет единственный верный ответ. Шкала весов для заданий данного типа отсутствует.

3. *Истина/ложь* — задание, в котором требуется определить истину или ложность приведенного высказывания.

4. *Установление соответствия* — задание, предусматривающее список объектов, каждому из которых должен соответствовать объект из другого списка.

5. *Упорядочивание* — задание, производное от *Установления соответствия*. Для данного типа задания один из списков является упорядоченным, например числовым рядом.

Примеры входных и выходных данных приведены на рис. 2.

Режимы работы с тестовыми заданиями

Пользовательский интерфейс КТЗ реализует два основных сценария использования. На рис. 3 приведен интерфейс точки входа в приложения.

Для создания и преобразований доступны все типы заданий, описанные в разделе "Поддерживаемые прототипом типы тестовых заданий" (рис. 4).

Для обеспечения простоты использования в прототипе реализованы два режима конструирования тестовых заданий: *упрощенный* и *расширенный*. При создании нового тестового задания по умолчанию запускается упрощенная версия редактора, которая состоит из трех основных блоков: *основные параметры задания*, *ответы* и *элементы управления*. Пример интерфейса упрощенной версии редактора



Рис. 3. Основной экран прототипа

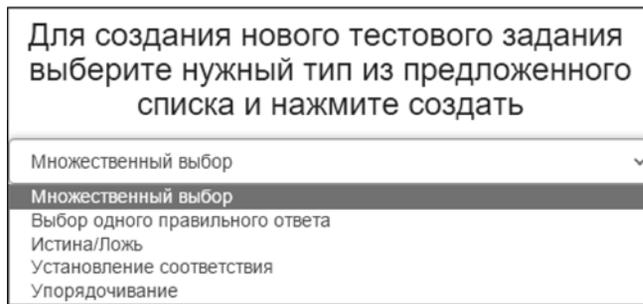


Рис. 4. Выбор типа тестового задания

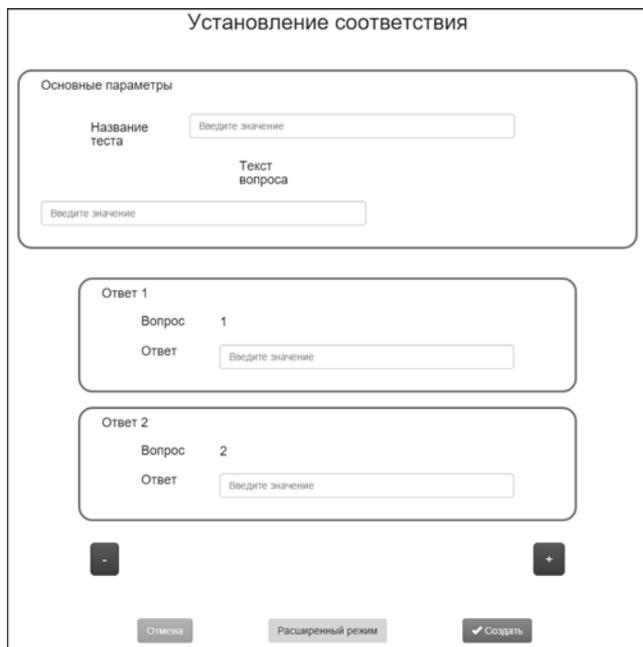


Рис. 5. Упрощенный редактор задания на установление соответствия

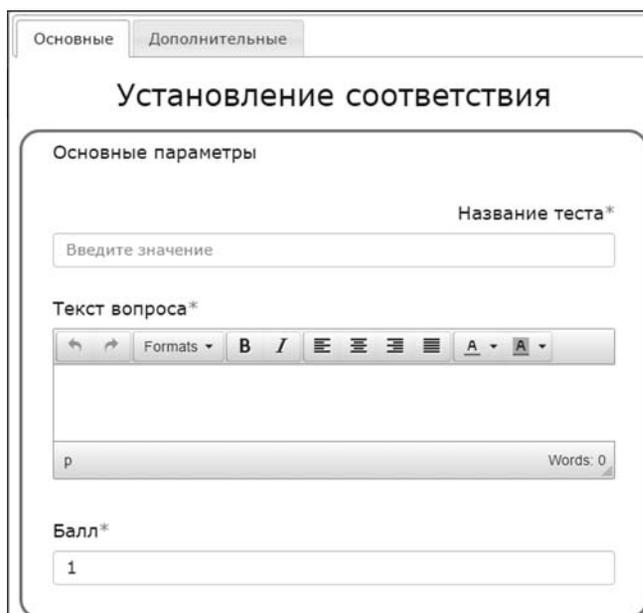


Рис. 6. Расширенный редактор задания на установление соответствия

для задания на установление соответствия представлен на рис. 5.

Расширенная версия редактора (форма интерфейса приведена на рис. 6) содержит два раздела для управления дополнительными характеристиками заданий: основной раздел для форматирования текста задания, вариантов ответа и весов вариантов ответов и дополнительный для добавления комментариев к ответам.

Планы развития и следующие версии программы

В следующих версиях программы планируется:

1. Обеспечить поддержку тестовых заданий со свободным ответом, а также заданий, содержащих мультимедийный контент (например, изображения).
2. Расширить поддерживаемые форматы тестовых заданий другими распространенными текстовыми форматами, XML-форматами и форматом *QTI/APIP*.
3. Обеспечить возможность групповой обработки тестовых заданий.
4. Спроектировать и реализовать *RESTful API*.
5. Усовершенствовать механизм отправки тестового задания на почту и повысить его безопасность.
6. Добавить механизм локализации.
7. Дополнить прототип механизмами авторизации и идентификации пользователей.
8. Выполнить тестовую интеграцию с одной из существующих *LMS* или тестовых систем.

Заключение

Работа посвящена особенностям автоматизации разработки, сохранения и обмена тестовыми заданиями в программных средствах поддержки этих процессов.

На основе обзора и анализа наиболее распространенных форматов импорта/экспорта тестовых заданий, а также программных средств, внешних по отношению к системам управления учебным процессом и контроля знаний, предназначенных для разработки тестовых заданий и преобразования их форматов, сделан вывод о необходимости создания общедоступного специализированного программного средства конвертации тестовых заданий.

Сформулированы требования и поставлена задача разработки доступного по сети Интернет программного средства для конвертации тестовых заданий, представленных в различных форматах.

Согласно сформулированным требованиям разработана архитектура и реализован прототип *Web-API* с рабочим названием *Конвертер тестовых заданий*. В настоящее время прототип, доступный по адресу

[<http://testconstructor.azurewebsites.net>], проходит апробацию, тестирование и отладку. Существующая версия *Конвертера тестовых заданий* позволяет создавать тестовые задания закрытого типа и поддерживает их сохранение и преобразование в форматах *AIKEN*, *GIFT* и *MoodleXML*.

Список литературы

1. **Van der Linden W. J., Glas G. A.W.** Computerized Adaptive Testing: Theory and Practice. Dordrecht: Kluwer academic publishers, 2000. 324 p.
2. **Van der Linden W. J., Glas C. A.W.** Elements of adaptive testing. Dordrecht: Springer, 2010. 428 p.
3. **Rasch G.** Probabilistic models for some intelligence and attainment tests. Chicago: University of Chicago press, 1980. 199 p.
4. **Bradlow E. T., Wainer H., and Wang X.** A Bayesian random effects model for testlets, *Psychometrika*. 1999. Vol. 64, N. 2. P. 153–168.
5. **Wainer H., Bradlow E. T., and Wang X.** Testlet response theory and its application. New York: Cambridge University press, 2007. 280 p.
6. **Wang X., Bradlow E. T., and Wainer H.** A general Bayesian model for testlets: theory and applications // *Applied Psychological Measurement*. 2002. N. 16. P. 309–325.
7. **Интероперабельность** информационных систем. Сб. материалов. М.: INFO-FOSS, 2008. 128 с.
8. **Help** // Cruz roja Mexicana. URL: <http://www.cruzrojamexicanasonora.org.mx/capacitacion-en-linea/help.php?module=quiz&file=import.html> (дата обращения: 23.06.2015).
9. **Aiken** format. 2015. URL: https://docs.moodle.org/29/en/Aiken_format (дата обращения: 12.02.2015).
10. **GIFT** format. 2015. URL: https://docs.moodle.org/29/en/GIFT_format (дата обращения: 12.02.2015).
11. **Moodle XML** format. 2015. URL: https://docs.moodle.org/29/en/Moodle_XML_format (дата обращения: 25.05.2015).
12. **Blackboard Help**. Uploading questions. 2015. URL: https://help.blackboard.com/en-us/Learn/9.1_SP_10_and_SP_11/Instructor/070_Tests_Surveys_Pools/106_Uploading_Questions#question_format (дата обращения: 24.07.2015).
13. **SCORM** Assessment Engine. Documentation. 2015. URL: <https://docs.google.com/document/d/1zzLk8sp3MHsnLCNeAJbq-RcMv-DIg8Y4J0oRAUro5gbk/edit> (дата обращения: 24.07.2015).
14. **WebCT** format. 2015. URL: https://docs.moodle.org/29/en/Import_questions#WebCT_format (дата обращения: 21.05.2015).
15. **IMS** Question & Test Interoperability™ Specification // IMS Global learning consortium. 2012. URL: <http://www.imsglobal.org/question/> (дата обращения: 24.07.2015).
16. **IMS** Global learning consortium [Электронный ресурс] // IMS Global learning consortium: [сайт]. [2001–2015]. URL: <http://www.imsglobal.org/> (дата обращения: 24.07.2015).
17. **Accessible** Portable Item Protocol® (APIP®) // IMS Global learning consortium. 2014. URL: <http://www.imsglobal.org/apiip/> (дата обращения: 24.07.2015).
18. **Richardson L., Amundsen M., and Ruby S.** RESTful Web APIs. O'Reilly Media, 2013. 406 p.
19. **Аванесов В. С.** Композиция тестовых заданий. М: Центр тестирования, 2002. 240 с.
20. **Gulliksen H.** Theory of Mental Tests. New York: John Wiley & Sons, 1950. 504 p.

V. V. Podbelskiy, Professor, of School of Software Engineering, vpodbelskiy@hse.ru,
O. V. Maksimenkova, Junior Research Fellow,
International Laboratory for Intelligent Systems and Structural Analysis, omaksimenkova@hse.ru,
K. S. Babich, Student, Faculty of Computer Science
National Research University Higher School of Economics, Moscow

On Educational Testing Software Interoperability Support

Nowadays the computer adaptive testing plays the main role in massive educational testing. This partly causes the growing interest to computerized forms of testing. In its turn, data transfer between item banks and test systems has become the topic of high actuality this last few years. This paper address questions about the interoperability of educational testing systems. The most popular text and XML-formats of the test questions (including rapidly developed QTI/APIP) were studied. The review of existing converters demonstrated that there is no freely accessed converter, which supports several questions' types, has adequate Russian interface or allows internationalization, and offers authoring tools for the questions of different types. Consequently, the architectural and functional software requirements on test questions' converter were determined and listed. As a result, the paper presents the web-API prototype, which was design and implemented according to collected requirements.

Keywords: computer-based testing, test questions formats, test items converter, QTI/APIP compatibility

References

1. **Van der Linden W. J., Glas G. A.W.** *Computerized Adaptive Testing: Theory and Practice*. Dordrecht: Kluwer academic publishers, 2000. 324 p.
2. **Van der Linden W. J., Glas C. A.W.** *Elements of adaptive testing*. Dordrecht: Springer, 2010. 428 p.
3. **Rasch G.** *Probabilistic models for some intelligence and attainment tests*. Expanded ed. Chicago: University of Chicago press, 1980. 199 p.
4. **Bradlow E. T., Wainer H., and Wang X.**, A Bayesian random effects model for testlets, *Psychometrika*, 1999, vol. 64, no. 2, pp. 153–168.
5. **Wainer H., Bradlow E. T., and Wang X.** *Testlet response theory and its application*. New York: Cambridge University press, 2007. 280 p.
6. **Wang X., Bradlow E. T., and Wainer H.**, A general Bayesian model for testlets: theory and applications, *Applied Psychological Measurement*, 2002, no. 16, pp. 309–325.
7. **Interoperabel'nost' informacionnyh sistem** (Information systems interoperability). Proceedings. Moscow: INFO-FOSS, 2008. 128 pp.
8. **Help.** *Cruz roja Mexicana*. URL: <http://www.cruzrojamexicana-nasonora.org.mx/capacitacion-en-linea/help.php?module=quiz&file=import.html> (Accessed: 23.06.2015).
9. **Aiken format**. 2015. URL: https://docs.moodle.org/29/en/Aiken_format (Accessed: 12.02.2015).
10. **GIFT format**. 2015. URL: https://docs.moodle.org/29/en/GIFT_format (Accessed: 12.02.2015).
11. **Moodle XML format**. 2015. URL: https://docs.moodle.org/29/en/Moodle_XML_format (Accessed: 25.05.2015).
12. **Uploading questions**, *Blackboard Help*. 2015. URL: https://help.blackboard.com/en-us/Learn/9.1_SP_10_and_SP_11/Instructor/070_Tests_Surveys_Pools/106_Uploading_Questions#question_format (Accessed: 24.07.2015).
13. **SCORM Assessment Engine. Documentation**. 2015. URL: <https://docs.google.com/document/d/1zzLk8sp3MHsnLCNeAJbqRcMv-Dlg8Y4J0oRAUro5gbk/edit> (Accessed: 24.07.2015).
14. **WebCT format**. 2015. URL: https://docs.moodle.org/29/en/Import_questions#WebCT_format (Accessed: 21.05.2015).
15. **IMS Question & Test Interoperability™ Specification**, *IMS Global learning consortium*. 2012. URL: <http://www.imsglobal.org/question/> (Accessed: 24.07.2015).
16. **IMS Global learning consortium, IMS Global learning consortium**. [2001–2015]. URL: <http://www.imsglobal.org/> (Accessed: 24.07.2015).
17. **Accessible Portable Item Protocol® (APIP®)**, *IMS Global learning consortium*. 2014. URL: <http://www.imsglobal.org/api/> (Accessed: 24.07.2015).
18. **Richardson L., Amundsen M., and Ruby S.** *RESTful Web APIs*. O'Reilly Media, 2013. 406 p.
19. **Avanesov V. S.** *Kompozicija testovyh zadaniy* (Test questions' composition). Moscow: Testing center, 2002. 240 p.
20. **Gulliksen H.** *Theory of Mental Tests*. New York: John Wiley & Sons, 1950. 504 p.

В. Б. Вяткин, канд. техн. наук, e-mail: vbvbbv@yandex.ru,
г. Екатеринбург

Интегративно-кодовая взаимосвязь комбинаторного, вероятностного и синергетического подходов к определению количества информации

Рассматриваются интегративные коды элементов дискретных систем и обосновывается, что в общем случае они могут быть разделены на групповую и системную части. Показано, что через средневзвешенную длину этих частей выражаются информационные меры комбинаторного, вероятностного и синергетического подходов к определению количества информации, что является доказательством непосредственной взаимосвязи последних. Сделано заключение, что с информационно-генетических позиций синергетический подход является первичным относительно комбинаторного и вероятностного подходов.

Ключевые слова: количество информации, интегративный код, синтропия, энтропия, конечное множество, дискретная система

Введение

Традиционными подходами к определению количества информации являются комбинаторный и вероятностный [2], в которых за информацию принята снимаемая неопределенность выбора одной из множества возможностей. Эти подходы широко используют в различных предметных областях, но в то же время они не позволяют корректно решать информационные по своей сущности задачи, связанные с анализом отражения друг через друга системных образований, представленных конечным множеством элементов [3–5]. Данный факт обусловил разработку нового — синергетического — подхода к определению количества информации, в котором за информацию приняты сведения о конечном множестве как едином целом [5, 6].

Развитие указанного подхода в виде одноименной теории информации [7] показало, что между ним и традиционными подходами существует непосредственная взаимосвязь. Эта взаимосвязь выразилась в том, что установленная в синергетическом подходе энтропия отражения, представляющая собой такую информацию о дискретной системе, которая не воспроизводится через совокупность ее частей, оказалась математически тождественной энтропии Шеннона, лежащей в основе вероятностного подхода. При этом содержательная сторона отмеченной взаимосвязи различных подходов не была раскрыта.

В настоящей работе проводится анализ интегративных кодов элементов дискретных систем, разделенных на части по значениям произвольного признака. На основе этого анализа раскрывается природа взаимосвязи различных подходов к определению количества информации и показывается, что синергетический подход является генетически первичным относительно комбинаторного и

вероятностного подходов. При этом для лучшего восприятия материала сначала дается общее описание каждого из подходов в контексте рассуждений их авторов.

1. Комбинаторный, вероятностный и синергетический подходы к определению количества информации

Комбинаторный подход. Среди известных подходов к определению количества информации наиболее ранним является комбинаторный подход, разработанный Хартли в 1928 г. [8]. Решая задачу определения количества информации (H_0), передаваемой по техническим каналам связи, Хартли исходил из того, что при передаче сообщения с помощью N -символьного алфавита каждый символ сообщения является результатом выбора одной из N возможностей. Соответственно, для того чтобы передать сообщение, состоящее из n символов, необходимо осуществить n таких выборов. Беря сказанное за основу, Хартли постулировал, что "количество информации пропорционально числу выборов" [8, с. 11], и проводя после этого несложные математические операции, получил следующую меру информации:

$$H_0 = \log N^n. \quad (1)$$

Вероятности появления различных символов в сообщении при этом во внимание не принимали, т. е. по умолчанию считались одинаковыми, а основание логарифма было принято считать произвольным.

Наибольшую популярность комбинаторная мера (1) получила в том своем частном виде, когда выбор из N возможностей осуществляется один раз ($n = 1$), а основание логарифма равно двум:

$$H_0 = \log_2 N. \quad (2)$$

Более того, с выражения (2) сейчас начинается, как правило, и собственно рассмотрение комбинаторного подхода [2]. При этом использование двоичного основания логарифма не имеет под собой какого-либо теоретического обоснования, а обусловлено лишь удобством оперирования двоичными логарифмами при передаче, хранении и обработке информации.

Вероятностный подход. В 1948 г. Шеннон распространил идею Хартли о связи количества информации с выбором из множества возможностей на общий случай, когда возможности имеют различную вероятность [9]. Вывод соответствующей меры информации начинался при этом с допущения, что "имеется некоторое множество возможных событий, вероятности осуществления которых суть p_1, p_2, \dots, p_N " [9, с. 259]. После этого был поставлен вопрос: "Можно ли найти меру того, насколько велик "выбор" из такого набора событий или сколь неопределен для нас его исход?" [9, с. 259]. Отвечая на поставленный вопрос, Шеннон обосновал, что такой мерой (H) является следующая функция:

$$H = -K \sum_{i=1}^N p_i \log p_i, \quad (3)$$

где K — некоторая положительная постоянная, которая зависит от основания логарифма и определяет просто выбор единицы измерения.

Форма функции (3) проявила определенную степень подобия с термодинамической энтропией Больцмана, на основании чего Шеннон назвал эту функцию энтропией множества вероятностей, утверждая, что "она является разумной количественной мерой возможности выбора или мерой количества информации" [9, с. 262].

Наибольшее распространение энтропийная мера Шеннона (3), как и комбинаторная мера Хартли (1), получила при использовании двоичных логарифмов. В этом случае принимается, что $K = 1$ и, соответственно, имеем

$$H = - \sum_{i=1}^N p_i \log_2 p_i. \quad (4)$$

При этом, сравнивая (2) и (4), нетрудно видеть, что $p_i = \text{const} \Rightarrow H = H_0$. Это говорит о том, что во взаимоотношениях вероятностного и комбинаторного подходов соблюдается принцип соответствия.

Следует также отметить, что вероятностная мера (4) может быть представлена как математическое ожидание случайной величины ($-\log_2 p$): $H = M[\log_2 p]$. То есть информация, получаемая в результате снятия неопределенности выбора одной из N возможностей, равна среднему значению логарифмов вероятности осуществления этих возможностей. На этом основании в вероятностном подходе вводится также понятие частной информации I_i [10], как индивидуальной информационной характеристики i -й возможности:

$$I_i = -\log_2 p_i. \quad (5)$$

В содержательном плане выражение (5) говорит о том, что, чем меньше вероятность наступления какого-либо события, тем больше информации дает его осуществление.

Синергетический подход. В основе этого подхода [5, 6] лежит задача оценки синтропии отражения, постановка которой выглядит следующим образом.

Пусть в составе некоторой системы $D = \{d\}$ по отличительным признакам P_A и P_B выделены два конечных множества $A = \{a|P_A(a)\} = \{d|P_A(d)\}$ и $B = \{b|P_B(b)\} = \{d|P_B(d)\}$, такие, что $A \cap B = K, K \neq \emptyset$. Число элементов в составе каждого из множеств равно $|A|, |B|, |K|$. Требуется определить, чему равна синтропия отражения I_{AB} , т. е. количество информации, которую множества A и B отражают друг о друге как едином целом. (До 2012 г. автор подхода называл эту информацию негэнтропией отражения [11].)

Так как при $A = B = K$ отражение множеств A и B друг через друга не отличается от их отражения через самих себя, то решение поставленной задачи начинается с определения количества информации I_A , которую произвольное конечное множество A отражает о самом себе как о целостном образовании. При этом заметим, что ранее [5—7] информация I_A называлась просто самоотражаемая множеством информации, что создавало определенные неудобства. Поэтому, поскольку эта информация связана только с самим множеством A и совместно с признаком P_A является его атрибутивной характеристикой, будем в дальнейшем называть ее атрибутивной информацией конечного множества или просто атрибутивной информацией.

Вывод формулы атрибутивной информации I_A базируется на двух аксиомах.

1. **Аксиома монотонности.** *Атрибутивная информация конечного множества является монотонно возрастающей функцией от общего числа его элементов, т. е. для любых двух конечных множеств A и B с числом элементов $|A|$ и $|B| = |A| + 1$ имеет место неравенство*

$$I_A < I_B. \quad (6)$$

2. **Аксиома интегративности.** *Показателем конечного множества A как единого целого является интегративный код его элементов, представляющий собой индивидуальную для каждого элемента последовательность символов какого-либо алфавита, число которых L_A (длина кода) является функцией от общего числа элементов $|A|$ в составе множества.*

На основе принятых аксиом рассматривается процесс увеличения числа элементов $|A|$ в составе множества A , который представляется в виде роста ориентированного дерева, совокупность висячих вершин которого взаимно-однозначно соответствует множеству элементов $a \in A$, а максимальное число дуг, выходящих из одной вершины, равно числу символов n алфавита, выбранного для составления интегративных кодов. При этом каждой из смежных дуг в алфавитном порядке ставится в соответствие свой символ и, как следствие, в качестве индивидуального интегративного кода какого-либо элемента выступает последовательность символов, находящаяся на пути движения из начальной вершины дерева в соответствующую данному элементу висячую вершину. Пример такого дерева кодов при $n = 2$ и использовании в качестве алфавита упорядоченной пары символов $\langle 0, 1 \rangle$ приведен на рис. 1.

В результате указанного рассмотрения за количество атрибутивной информации конечного множества при-

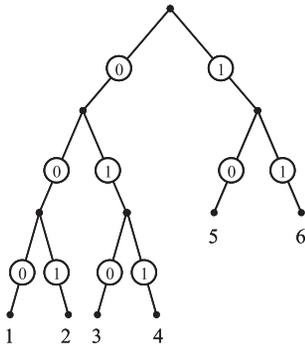


Рис. 1. Дерево интегративных кодов при $n = 2$ и $|A| = 6$

нимается средняя длина интегративного кода его элементов

$$I_A = \bar{L}_A. \quad (7)$$

Так как при $|A| \leq n$ величина \bar{L}_A не изменяется и тем самым нарушается аксиома монотонности (6), то утверждается, что интегративные коды элементов множества могут быть составлены только с помощью двоичного алфавита ($n = 2$). Соответствующая формула средней длины кода получена в следующем виде [5, 6]:

$$\bar{L}_A = x + 2 - \frac{2^{x+1}}{|A|}, \quad (8)$$

где x — целочисленная часть $\log_2 |A|$.

При этом установлено, что разность $\bar{L}_A - \log_2 |A|$ ограничена постоянной величиной

$$\psi = \sup_{|A| \in [1, \infty)} (\bar{L}_A - \log_2 |A|) = 0,0860713... \quad (9)$$

В силу выражения (9) величина $\log_2 |A|$ может служить аппроксимацией средней длины интегративного кода (8). Поэтому для удобства теоретических построений и практических расчетов, формула атрибутивной информации I_A принята в следующем виде:

$$I_A = \log_2 |A|. \quad (10)$$

После этого решение поставленной задачи по оценке синтропии отражения I_{AB} проводится на основе анализа передачи атрибутивной информации по системе связи, в которой множества A и B поочередно выступают в качестве источника и приемника информации, а множество K играет роль передающей среды или канала связи. Формула синтропии I_{AB} при этом получена в следующем виде:

$$I_{AB} = \frac{|K|^2}{|A||B|} \log_2 |K|. \quad (11)$$

В том случае, когда одно множество является подмножеством другого, синтропия отражения (11) сводится (на примере $B \subset A$) к своему частному виду:

$$B \subset A \Rightarrow I_{AB} = \frac{|B|}{|A|} \log_2 |B|. \quad (12)$$

Из формулы (11) следует, что, когда $A = B = K$ и множества полностью отражаются друг через друга, синтропия отражения принимает свое максимальное значение, равное атрибутивной информации каждого из множеств,

т. е. $I_{AB}^{\max} = I_A = I_B = I_K$. При этом заметим, что если в данной ситуации для информационной оценки взаимного отражения множеств A и B (определения величины I_{AB}) попытаться использовать комбинаторный и вероятностный подходы, то мы придем к противоречию здравому смыслу в виде равенства $H_0 = H = 0$, которое обусловлено отсутствием какого-либо выбора при $A = B = K$. Это лишний раз говорит о том, что информация, рассматриваемая в синергетическом подходе, существует независимо от управления, т. е. не связана с процедурой выбора, как это принято в традиционных подходах.

2. Равенство энтропии Шеннона и энтропии отражения

При анализе отражения дискретной системы $A = \{a|P_A\}$ через совокупность своих частей B_1, B_2, \dots, B_N , выделенных по признаку $P_B = P_{B_1}, P_{B_2}, \dots, P_{B_N}$, таких, что $\bigcap_{i=1}^N B_i = \emptyset$, $\bigcup_{i=1}^N B_i = A$, $\sum_{i=1}^N |B_i| = |A|$, установлено следующее [7].

Аддитивная синтропия отражения системы I_Σ через N ее частей, определяемая на основе (12) в виде суммы частных синтропий

$$I_\Sigma = \sum_{i=1}^N I_{AB_i} = \sum_{i=1}^N \frac{|B_i|}{|A|} \log_2 |B_i|, \quad (13)$$

при $N > 1$ всегда меньше атрибутивной информации I_A , отражаемой системой как целостным образованием, т. е. $I_\Sigma < I_A$. При этом та часть информации I_A , которая остается неотраженной, характеризует неопределенность отражения системы и в силу этого называется энтропией отражения S . Формула этой энтропии получена в следующем виде [6]:

$$S = I_A - I_\Sigma = - \sum_{i=1}^N \frac{|B_i|}{|A|} \log_2 \frac{|B_i|}{|A|}. \quad (14)$$

Так как с позиций теории вероятностей отношение $|B_i|/|A|$ представляет собой вероятность p_i встречи элементов, обладающих i -м значением признака P_B среди общего числа элементов системы A , то из сравнения выражений (4) и (14) следует, что с математических позиций энтропия отражения и энтропия Шеннона равны друг другу:

$$S = H. \quad (15)$$

Равенство (15) свидетельствует о том, что между синергетическим и традиционными подходами к определению количества информации существует непосредственная взаимосвязь. При этом в силу того, что энтропия отражения S является мерой неотраженной информации и выводится после определения аддитивной синтропии I_Σ , на формальном уровне суждений можно говорить о том, что в информационно-генетическом отношении синергетический подход является первичным относительно комбинаторного и вероятностного подходов.

В нижеследующем изложении раскрывается природа указанной взаимосвязи и показывается, что синергетический подход действительно является генетически первичным.

3. Интегративно-кодовая взаимосвязь различных подходов к определению количества информации

Каждая часть описанной выше дискретной системы A представляет собой автономное множество элементов B_i со средней длиной интегративного кода \bar{L}_{B_i} . Очевидно, что когда N таких множеств объединяются в систему и выступают как единое целое, средняя длина интегративного кода элементов каждого множества возрастает на значение разности $\bar{L}_A - \bar{L}_{B_i}$. При этом в содержательном плане можно говорить о том, что указанная разность по отношению к

множеству B_i представляет собой системный эффект от его вхождения в состав системы A в качестве ее части.

Из сказанного следует, что интегративный код любого элемента системы A в общем случае может быть разделен на две части. При этом первая часть образуется за счет множества элементов, имеющих одинаковое значение признака $P_B = P_{B_1}, P_{B_2}, \dots, P_{B_N}$, а вторая часть является надстройкой первой и обусловлена объединением в систему совокупности множеств B_1, B_2, \dots, B_N . Так как любое множество B_i в составе системы A можно также называть однородной по признаку P_B группой элементов, то дадим указанным частям интегративных кодов следующие названия: первой — *групповая часть* кода, второй — *системная часть* кода. При этом длину частей кода будем обозначать следующими символами: ΔL_G — длина групповой части; ΔL_S — длина системной части.

Таким образом, длину L интегративного кода элементов дискретной системы в общем случае можно представить в виде следующего уравнения:

$$L = \Delta L_G + \Delta L_S. \quad (16)$$

Чтобы иметь наглядное представление о делении интегративных кодов на групповую и системную части обратимся к рис. 2, где приведены кодовые деревья дискретной системы A с числом элементов $|A| = 8$. При этом на рисунке показаны три ситуации, которые отражают два полярных (рис. 2, а, в) и один общий (рис. 2, б) случаи деления системы на N частей.

На рис. 2, а непосредственное деление системы на части отсутствует ($N = 1$), т. е. все элементы системы характеризуются одним и тем же значением признака P_B и соответственно все коды представлены только групповой частью. Противоположная ситуация показана на рис. 2, в, где число частей системы равно числу ее элементов ($N = |A|$), т. е. в данном случае каждый элемент системы имеет индивидуальное значение признака P_B и его интегративный код состоит только из системной части. На рис. 2, б приведен наиболее общий случай деления системы на части ($1 < N < |A|$), когда части соотносятся между собой по числу элементов произвольным образом, т. е., когда элементы системы по значениям признака P_B могут образовывать множества с числом элементов от $|B_i| = 1$ до $|B_i| = |A| - 1$. При этом, когда выполняется не-

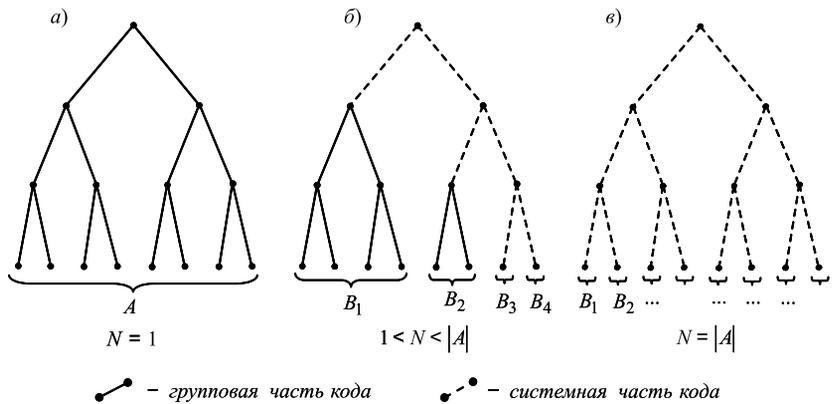


Рис. 2. Деревья интегративных кодов дискретной системы A

равенство $1 < |B_i| \leq |A| - 1$ (части B_1 и B_2 на рис. 2, б), то в структуре кода выделяются как групповая, так и системная части, а если $|B_i| = 1$ (части B_3 и B_4 на рис. 2, б), то коды представлены только системной частью.

Рассмотрим теперь количественные аспекты интегративных кодов элементов, образующих систему A , и начнем с групповой части и общей длины кода.

Из выражений (7) и (10) следует, что средняя длина \bar{L}_A интегративного кода элементов системы A и среднее значение $(\Delta L_G)_{B_i}$ групповой части кода элементов любой ее части B_i равны

$$\bar{L}_A = \log_2 |A|; \quad (17)$$

$$(\Delta L_G)_{B_i} = \log_2 |B_i|. \quad (18)$$

Средневзвешенное значение $(\Delta L_G)_A$ длины групповой части кода по всем частям системы A , в свою очередь, выражается формулой

$$(\Delta L_G)_A = \frac{\sum_{i=1}^N |B_i| \log_2 |B_i|}{\sum_{i=1}^N |B_i|}.$$

Так как $\sum_{i=1}^N |B_i| = |A|$, то окончательно для $(\Delta L_G)_A$

имеем

$$(\Delta L_G)_A = \sum_{i=1}^N \frac{|B_i|}{|A|} \log_2 |B_i|. \quad (19)$$

Определяя теперь среднее значение $\bar{\Delta L}_S$ длины системной части кода, сначала на основе уравнения (16) с очевидностью отметим, что это значение для элементов B_i части системы будет равно разности между средним значением общей длины кода всех элементов системы (17) и средним значением групповой части кода элементов данной части системы (18):

$$\begin{aligned} (\Delta L_S)_{B_i} &= \bar{L}_A - (\Delta L_G)_{B_i} = \\ &= \log_2 |A| - \log_2 |B_i| = -\log_2 \frac{|B_i|}{|A|}. \end{aligned} \quad (20)$$

Соответственно, средневзвешенное значение $(\overline{\Delta L_S})_A$ системной части интегративного кода по всем элементам системы равно

$$(\overline{\Delta L_S})_A = \frac{-\sum_{i=1}^N |B_i| \log_2 \frac{|B_i|}{|A|}}{\sum_{i=1}^N |B_i|} = -\sum_{i=1}^N \frac{|B_i|}{|A|} \log_2 \frac{|B_i|}{|A|}. \quad (21)$$

В том случае, когда все части системы равны между собой по числу элементов, т. е., когда $|B_i| = |A|/N$, имеем

$$(\overline{\Delta L_S})_A \Big|_{|B_i| = \text{const}} = \log_2 N. \quad (22)$$

Проводя теперь сравнительный анализ полученных характеристик интегративных кодов с информационными мерами комбинаторного, вероятностного и синергетического подходов к определению количества информации, можно сказать следующее.

Комбинаторный и вероятностный подходы. Наиболее известная мера информации — энтропийная мера Шеннона (4) — представляет собой средневзвешенное значение системной части интегративного кода элементов дискретной системы (21), разделенной по значениям произвольного признака P_B на N частей:

$$H = (\overline{\Delta L_S})_A. \quad (23)$$

В том случае, когда все части системы равны между собой по числу элементов (22), через средневзвешенное значение системной части интегративного кода выражается также комбинаторная мера Хартли (2):

$$H_0 = (\overline{\Delta L_S})_A \Big|_{|B_i| = \text{const}}. \quad (24)$$

Частная информация (5), фигурирующая в вероятностном подходе, в свою очередь, представляет собой среднее значение системной части кода (20) тех элементов системы, которые имеют одно и то же значение признака P_B :

$$I_i = (\overline{\Delta L_S})_{B_i}. \quad (25)$$

Синергетический подход. Первоначально выводимая в этом подходе мера атрибутивной информации конечного множества (10) по определению представляет собой среднюю длину интегративного кода всех элементов системы (7), которая при $N = 1$ также может рассматриваться в качестве средней длины групповой части интегративного кода элементов, обладающих признаком P_A , т. е.

$$I_A = \bar{L}_A = (\overline{\Delta L_G})_A \Big|_{N=1}. \quad (26)$$

Частная синтропия отражения системы I_{AB_i} какой-либо ее частью, определяемая по формуле (12), с интегративно-кодовых позиций представляет собой соответствующее слагаемое в формуле средневзвешенного значения групповой части кода элементов системы (19) и с учетом равенства (18) может быть представлена в следующем виде:

$$I_{AB_i} = \frac{|B_i|}{|A|} (\overline{\Delta L_G})_{B_i}. \quad (27)$$

Аддитивная синтропия отражения (13), в свою очередь, равна средневзвешенному значению групповой части кода всех элементов системы (19):

$$I_\Sigma = (\overline{\Delta L_G})_A. \quad (28)$$

Энтропия отражения системы через совокупность своих частей (14), так же как и энтропия Шеннона (4), равна средневзвешенному значению системной части интегративного кода элементов системы (21), т. е.

$$S = (\overline{\Delta L_S})_A. \quad (29)$$

Обобщением сказанного является таблица, в которой вместе с информационными мерами комбинаторного (2), вероятностного (4), (5) и синергетического (10), (12—14) подходов к определению количества информации приведены результаты их сравнительного анализа (23)—(29) с различными характеристиками интегративных кодов элементов дискретных систем.

Приведенная таблица наглядно показывает, что *информационные меры комбинаторного, вероятностного и синергетического подходов по своей сущности являются количественными характеристиками структурных особенностей интегративных кодов элементов дискретных систем.* Именно это следует считать глубинной природой взаимосвязи данных подходов к определению количества информации. При этом информационные меры комбинаторного и вероятностного подходов функционально связаны только с системной частью интегративных кодов, в то время как в мерах синергетического подхода фигурируют обе части кода, а также его общая длина.

Так как системная часть кодов является надстройкой групповой части и образуется только тогда, когда система делится на части по значениям какого-либо признака, а до этого деления интегративные коды элементов представлены только групповой частью, то на основе сказанного можно сделать следующий вывод. *Информация, фигурирующая в синергетическом подходе (сведения*

Информационные меры и их интегративно-кодовая интерпретация

Подход к определению количества информации	Информационная мера	
	Оригинальный вид	Интегративно-кодовая интерпретация
Комбинаторный	$H_0 = \log_2 N$	$H_0 = (\overline{\Delta L_S})_A \Big _{ B_i = \text{const}}$
Вероятностный	$H = -\sum_{i=1}^N p_i \log_2 p_i$ $I_i = -\log_2 p_i$	$H = (\overline{\Delta L_S})_A$ $I_i = (\overline{\Delta L_S})_{B_i}$
Синергетический	$I_A = \log_2 A $ $I_{AB_i} = \frac{ B_i }{ A } \log_2 B_i $ $I_\Sigma = \sum_{i=1}^N \frac{ B_i }{ A } \log_2 B_i $ $S = -\sum_{i=1}^N \frac{ B_i }{ A } \log_2 \frac{ B_i }{ A }$	$I_A = \bar{L}_A = (\overline{\Delta L_G})_A \Big _{N=1}$ $I_{AB_i} = \frac{ B_i }{ A } (\overline{\Delta L_G})_{B_i}$ $I_\Sigma = (\overline{\Delta L_G})_A$ $S = (\overline{\Delta L_S})_A$

о конечном множестве), является генетически первичной по отношению к информации, измеряемой в комбинаторном и вероятностном подходах (снятая неопределенность выбора). Это лишний раз подтверждает ранее сделанное формальное заключение о первичности синергетического подхода, основанное на последовательности получения формул атрибутивной информации, аддитивной синтропии и энтропии отражения, последняя из которых равна энтропии Шеннона.

4. Примеры генетической первичности информации синергетического подхода

Покажем на конкретном примере как можно практически представить себе генетическую первичность информации синергетического подхода и вторичность информации комбинаторного и вероятностного подходов.

Известно [2], что при передаче текстовых сообщений по каналам технической связи среднее количество информации, приходящейся на одну букву, равно энтропии Шеннона, определенной на основе вероятностей появления букв в языке, на котором составлено сообщение. Эти вероятности выражают статистическую структуру языка и определяются на основе анализа представительного по объему текста, написанного на данном языке. То есть прежде чем вычислять энтропию Шеннона, нужно провести предварительную работу по определению общего числа букв анализируемого текста (система A с числом элементов $|A|$, в качестве которых выступают буквы) и числа появлений в тексте каждой буквы используемого N -буквенного алфавита (части $B_i \subset A$ с числом букв $|B_i|$, $i = 1, 2, \dots, N$). Только после этого можно оценить вероятность появления в тексте каждой буквы ($p_i = |B_i|/|A|$) и перейти к непосредственному вычислению энтропии Шеннона. При этом нетрудно видеть, что уже на предварительном этапе работ мы можем сразу определить атрибутивную информацию анализируемого текста (I_A) и такую же информацию каждой из его частей (I_{B_i}). Кроме того, последовательно определяя вероятность появления различных букв, мы каждый раз можем параллельно вычислять синтропию отражения всего текста каждой буквой (I_{AB_i}).

В разд. 1 было показано, что использование энтропии Шеннона для оценки синтропии отражения может приводить к результату, противоречащему здравому смыслу. В то же самое время синтропия отражения, взятая сама по себе, позволяет решать определенный круг задач, связанных с вычислением данной энтропии. Приведем конкретный пример такой задачи и покажем ее решение, как с помощью энтропии Шеннона, так и с помощью синтропии отражения.

В работе [12] приведена следующая задача, решаемая с помощью энтропии Шеннона: "Имеются две урны, содержащие по 20 шаров — 10 белых, 5 черных и 5 красных в первой и 8 белых, 8 черных и 4 красных во второй. Из каждой урны вытаскивают по одному шару. Исход какого из этих двух опытов следует считать более неопределенным?" [12, с. 76]. При этом задача решается через определение энтропии Шеннона для каждого из опытов. В соответствии с формулой (4) имеем: энтропия первого опыта — $H^1 = 1,5$; второго опыта — $H^2 = 1,52$. На основе полученных значений авторы пишут: "Если оценивать

степень неопределенности опыта его энтропией, то надо считать, что исход второго опыта является более неопределенным, чем исход первого" [12, с. 77].

Решая эту задачу с помощью синтропии отражения, предварительно отметим, что аддитивная синтропия I_Σ является противоположностью энтропии отражения S , которая математически равна энтропии Шеннона H и, соответственно, ее можно рассматривать в качестве меры определенности исхода того или иного опыта. Поэтому, отвечая на поставленный в задаче вопрос, можно сказать, что более неопределенный исход имеет тот опыт, аддитивная синтропия системы шаров у которого имеет меньшее значение. Собственно решение задачи при этом выглядит следующим образом. Используя формулу (13), получаем, что аддитивная синтропия системы шаров первого опыта — $I_\Sigma^1 = 2,82$; второго опыта — $I_\Sigma^2 = 2,8$. В результате имеем неравенство $I_\Sigma^1 > I_\Sigma^2$ и, соответственно, можно утверждать, что неопределенность исхода второго опыта больше, чем первого. То есть мы пришли к такому же решению задачи, как и в случае использования энтропии Шеннона.

Заключение

В работе проведен анализ интегративных кодов элементов дискретных систем, разделенных на части по значениям произвольного признака, и показано, что в общем случае эти коды делят на групповую и системную части. При этом групповая часть кода обусловлена множеством элементов с одинаковым значением признака, а системная часть является надстройкой групповой части и образуется в результате объединения в одну систему множеств с различными значениями признака.

Установлено, что через средневзвешенные значения групповой и системной частей интегративного кода в точности выражаются информационные меры комбинаторного, вероятностного и синергетического подходов к определению количества информации. Причем, если меры синергетического подхода выражаются через средневзвешенное значение как групповой, так и системной частей кода, а также через его общую длину, то в мерах комбинаторного и вероятностного подходов фигурирует средневзвешенное значение только системной части. На этом основании сделаны выводы о том, что, во-первых, между указанными подходами существует интегративно-кодовая взаимосвязь и, во-вторых, что информация в виде сведений о конечном множестве, измеряемая в синергетическом подходе, является генетически первичной по отношению к информации как снятой неопределенности выбора, с которой оперируют комбинаторный и вероятностный подходы.

В конце прошлого века академик Колмогоров, анализируя положение дел в традиционной теории информации, говорил, — "не видно, почему теория информации должна столь существенно основываться на теории вероятностей, как это представляется по большинству руководств" [13, с. 29]. И утверждал при этом, что "теория информации должна предшествовать теории вероятностей, а не опираться на нее" [13, с. 35]. Представленная в статье интегративно-кодовая интерпретация

различных информационных мер актуализирует эти слова признанного авторитета в области теории информации и позволяет говорить о необходимости радикального пересмотра базовых основ данной теории, порожденных, главным образом, работами Хартли и Шеннона, посвященными передаче сообщений по системам технической связи. По мнению автора статьи, рано или поздно, но такой пересмотр обязательно произойдет.

Список литературы

1. **Вяткин В. Б.** К вопросу взаимосвязи комбинаторного, вероятностного и синергетического подходов к определению количества информации // Научный журнал КубГАУ. 2015. № 4 (108). С. 1374—1408.
2. **Колмогоров А. Н.** Три подхода к определению понятия "количество информации" // Проблемы передачи информации. 1965. Т. 1, № 1. С. 3—11.
3. **Берлянт А. М.** Образ пространства: карта и информация. М.: Мысль, 1986. 240 с.
4. **Вяткин В. Б.** К вопросу информационной оценки признаков при прогнозно-геологических исследованиях // Извест-

тия Уральского горного института. Сер.: Геология и геофизика. 1993. Вып. 2. С. 21—28.

5. **Вяткин В. Б.** Математические модели информационной оценки признаков рудных объектов: Дисс. ... канд. техн. наук: 05.13.18: Екатеринбург, 2004. 129 с.
6. **Вяткин В. Б.** Синергетический подход к определению количества информации // Информационные технологии. 2009. № 12. С. 68—73.
7. **Вяткин В. Б.** Введение в синергетическую теорию информации // Информационные технологии. 2010. № 12. С. 67—73.
8. **Хартли Р. В. Л.** Передача информации // Сб.: Теория информации и ее приложения. М.: Физматгиз, 1959. С. 5—35.
9. **Шеннон К.** Работы по теории информации и кибернетике. М.: Изд. иностр. лит., 1963. 830 с.
10. **Вентцель Е. С.** Теория вероятностей. М.: Наука, 1969. 576 с.
11. **Вяткин В. Б.** Синергетическая теория информации: пояснения и терминологические замечания // Научный журнал КубГАУ. 2012. № 6 (080). С. 557—592.
12. **Яглом А. М., Яглом И. М.** Вероятность и информация. М.: Наука, 1973. 512 с.
13. **Колмогоров А. Н.** Комбинаторные основания теории информации и исчисления вероятностей // УМН. 1983. Т. 38, вып. 4. С. 27—36.

V. B. Vyatkin, PhD, vbv@yandex.ru

Integrative-Code Interrelation of Combinatorial, Probabilistic and Synergistic Approaches to Determining the Amount of Information

Joint analysis of combinatorial, probabilistic and synergistic approaches to determining the amount of information on the basis of the consideration of integrative codes of discrete system elements, divided into parts according to the values of arbitrary sign, have been carried out in this article. (Integrative code is an individual description for each element of the system and presents a sequence of symbols of any alphabet, the length of which is a function of the total number of elements.) It is shown that, in general, integrative codes are divided into a group and system parts. The group part of code is specified by a set of elements which have identical value of attribute, system part is superstructure over group part and it is result of summing of sets having different characteristic into integrated system.

It was found that information measures of combinatorial, probabilistic and synergistic approaches to determining the amount of information are exactly expressed through the weighted average of the group and system parts of an integrative code. Moreover, if the measures of synergetic approach are expressed through the weighted average of the group and system parts of a code, the measures of combinatorial and probabilistic approaches are functionally linked only with the weighted average of the system parts. Following conclusions are made on this basis. First of all information measures of combinatorial, probabilistic and synergistic approaches are essentially the quantitative characteristics of the structural features of the integrative codes of an elements of a discrete systems. Secondly, the information appearing in a synergetic approach (information about a finite set, as a single entity), is genetically primary in relation to the information measured in combinatorial and probabilistic approaches (removed uncertainty of selection one of the many opportunities).

Keywords: amount of information, integrative code, syntropy, entropy, finite set, discrete system

References

1. **Vyatkin V. B.** K voprosu vzaimosvjazi kombinatornogo, verojatnostnogo i sinergeticheskogo podhodov k opredeleniju kolichestva informacii, *Nauchnyj zhurnal KubGAU*, KubGAU, 2015, no. 4 (108), pp. 1374—1408 (in Russian).
2. **Kolmogorov A. N.** Tri podhoda k opredeleniju ponjatija "kolichestvo informacii", *Problemy peredachi informacii*, 1965, vol. 1, no. 1, pp. 3—11 (in Russian).
3. **Berljant A. M.** *Obraz prostranstva: karta i informacija*, Moscow, Mysl', 1986, 240 p. (in Russian).
4. **Vyatkin V. B.** K voprosu informacionnoj ocenki priznakov pri prognozno-geologicheskikh issledovanijah, *Izvestija Ural'skogo gornogo instituta. Ser.: Geologija i geofizika*, 1993, no. 2, pp. 21—28 (in Russian).
5. **Vyatkin V. B.** *Matematicheskie modeli informacionnoj ocenki priznakov rudnyh ob'ektov*: Diss. ... kand. tehn. nauk: 05.13.18: Ekaterinburg, 2004, 129 p. (in Russian).

6. **Vyatkin V. B.** Sinergeticheskij podhod k opredeleniju kolichestva informacii, *Informacionnye tehnologii*, 2009, no. 12, pp. 68—73 (in Russian).
7. **Vyatkin V. B.** Vvedenie v sinergeticheskiju teoriju informacii, *Informacionnye tehnologii*, 2010, no. 12, pp. 67—73 (in Russian).
8. **Hartli R. V. L.**, *Peredacha informacii*, Sб.: *Teorija informacii i ee prilozhenija*. Moscow, Fizmatgiz, 1959, pp. 5—35 (in Russian).
9. **Shennon K.** *Raboty po teorii informacii i kibernetike*, Moscow: Izd. inostr. lit., 1963, 830 p. (in Russian).
10. **Ventcel' E. S.** *Teorija verojatnostej*, Moscow, Nauka, 1969, 576 p. (in Russian).
11. **Vyatkin V. B.**, Sinergeticheskaja teorija informacii: pojasnenija i terminologicheskie zamechanija, *Nauchnyj zhurnal KubGAU*, 2012, no. 6 (080), pp. 557—592 (in Russian).
12. **Jaglom A. M., Jaglom I. M.** *Verojatnost' i informacija*, Moscow, Nauka, 1973. 512 p. (in Russian).
13. **Kolmogorov A. N.** Kombinatornye osnovanija teorii informacii i ischislenija verojatnostej, *UMN*, 1983, vol. 38, no. 4, pp. 27—36 (in Russian).

ЖУРНАЛ В ЖУРНАЛЕ



**НЕЙРОСЕТЕВЫЕ
ТЕХНОЛОГИИ**

№ 7
ИЮЛЬ
2016

Главный редактор:

ГАЛУШКИН А. И.

Редакционная коллегия:

АВЕДЬЯН Э. Д.
БАЗИЯН Б. Х.
БЕНЕВОЛЕНСКИЙ С. Б.
БОРИСОВ В. В.
ГОРБАЧЕНКО В. И.
ЖДАНОВ А. А.
ЗЕФИРОВ Н. С.
ЗОЗУЛЯ Ю. И.
КРИЖИЖАНОВСКИЙ Б. В.
КУДРЯВЦЕВ В. Б.
КУЛИК С. Д.
КУРАВСКИЙ Л. С.
РЕДЬКО В. Г.
РУДИНСКИЙ А. В.
СИМОРОВ С. Н.
ФЕДУЛОВ А. С.
ЧЕРВЯКОВ Н. И.

**Иностранные
члены редколлегии:**

БОЯНОВ К.
ВЕЛИЧКОВСКИЙ Б. М.
ГРАБАРЧУК В.
РУТКОВСКИЙ Л.

Редакция:

БЕЗМЕНОВА М. Ю.
ГРИГОРИН-РЯБОВА Е. В.
ЛЫСЕНКО А. В.
ЧУГУНОВА А. В.

Галушкин А. И.

Новые технологии микроэлектроники и разработки перспективных нейрокомпьютеров 550

Дам В. Н.

Автоматическое распознавание цифровой модуляции радиосигналов с помощью многослойной нейронной сети по кумулянтным признакам 555

А. И. Галушкин, д-р техн. наук, проф.,
 начальник центра нейросетевых технологий, e-mail: neurocomputer@yandex.ru;
 Международный центр по информатике и электронике (Интер ЭВМ)

Новые технологии микроэлектроники и разработки перспективных нейрокомпьютеров

Описаны основные проблемы и трудности, которые имеются в настоящее время и, вероятно, будут присутствовать в будущем при развитии и внедрении новых технологий микроэлектроники в разработки перспективных нейрокомпьютеров.

Ключевые слова: нейросетевые технологии, новые технологии микроэлектронники, нейронные сети и нейрочипы с частотно-импульсной модуляцией сигналов, мемристоры

Введение

Нейросетевые технологии, развиваемые с 40-х годов прошлого века, в настоящее время получили активное развитие. Это связано с ростом сложности решаемых формализуемых и неформализуемых задач, что в свою очередь требует отказа от классической фон-Неймановской архитектуры вычислительных систем при переходе к распределенным, в том числе нейросетевым, архитектурам.

Решение сложных задач в нейросетевом логическом базисе зачастую требует аппаратной поддержки, которая в настоящее время реализуется на базе GPU, FPGA и во многих случаях — на базе цифровых, аналогово-цифровых и аналоговых нейрочипов.

В течение многих лет кроме стандартной КМОП-технологии для разработки и изготовления нейрочипов имеют место активные попытки применения для этих целей других технологий:

- мемристорной в сочетании с КМОП-технологией;
 - оптической;
 - квантовой;
 - молекулярной;
 - технологии вакуумных схем;
- и др.

Каждый тип технологий накладывает существенные ограничения на структуру аппаратно-реализуемого фрагмента нейронной сети и, как следствие, на алгоритмы настройки коэффициентов и нейросетевые алгоритмы решения задач. Примером могут являться российские разработки аналогово-цифровых нейрокомпьютеров 60-х, 70-х годов прошлого века, когда весовые коэффициенты реализовывались на потенциометрах, а сумматоры с нелинейным преобразованием — на операционных усилителях. При этом после аппаратной реализации достаточно большой нейронной сети с полными последовательными связями была сделана попытка построения нейрокомпьютера с уменьшением числа операционных усилителей нейронов с некоторым увеличением числа весовых коэффициентов (потенциометров), с реализацией нейронной сети с перекрестными связями, настраиваемыми связями с предыдущего слоя на последующий слой нейронной сети. В то время значительная разница в стоимости и энергопотреблении потенциометров и операционных усилителей являлась определяющей.

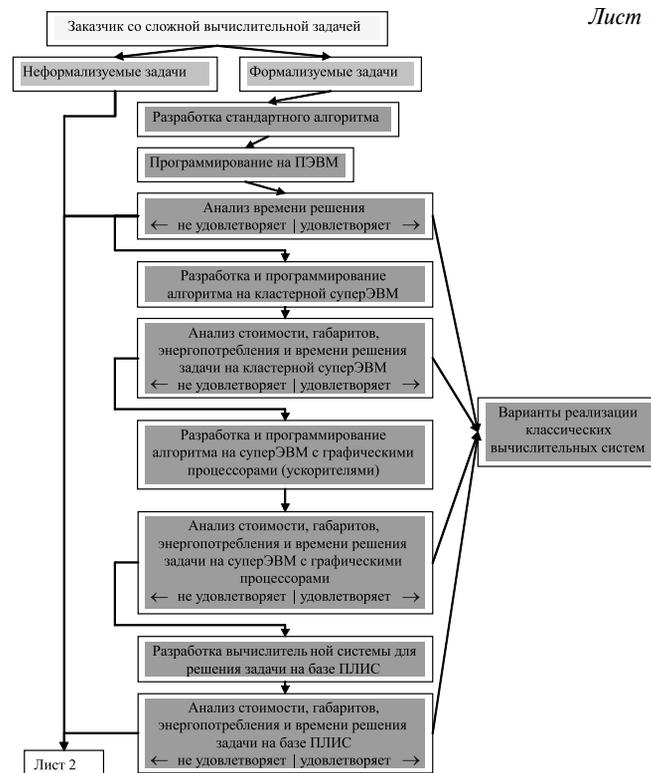
Данная статья, посвященная в основном проблемам создания нейрокомпьютеров с применением мемристоров, обращает внимание на то, что при построении ана-

логово-цифровых нейрочипов с применением мемристоров, число мемристоров в СБИС-нейрочипе может быть достаточно велико, что в свою очередь приведет к специализации структуры фрагмента аппаратно-реализуемой нейронной сети.

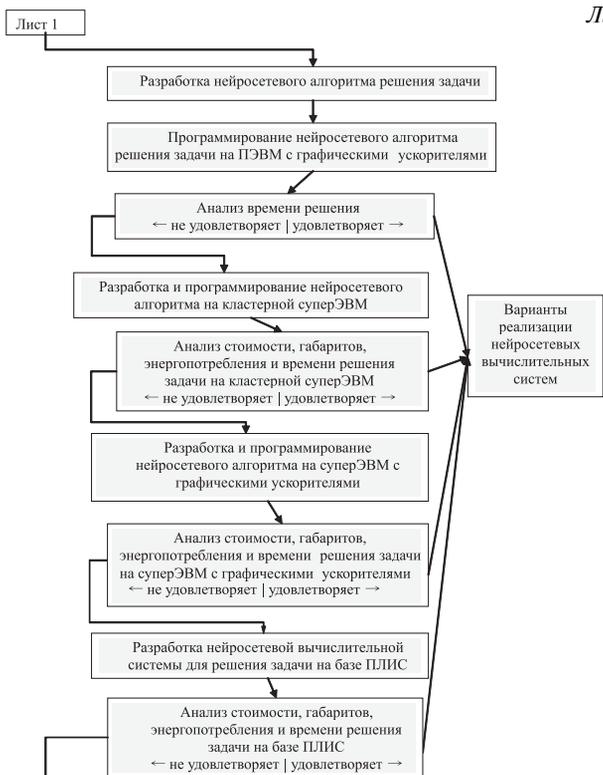
На рисунке представлена структура работ по внедрению новой технологии микроэлектроники (на примере мемристорной) в перспективные нейрокомпьютеры.

Данная структура представляет собой предлагаемые автором порядок и состав проведения работ по внедрению новой технологии микроэлектроники в высокопроизводительные нейрокомпьютеры. И порядок, и состав работ могут меняться в зависимости от конкретного вида технологии с обязательным или, по крайней мере, желательным обоснованием исключения отдельных этапов работ, представленных ниже в данной статье.

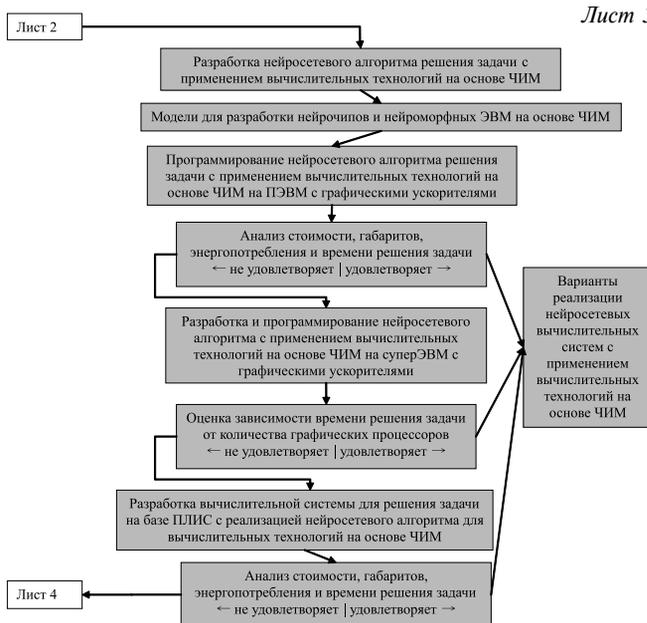
Лист 1



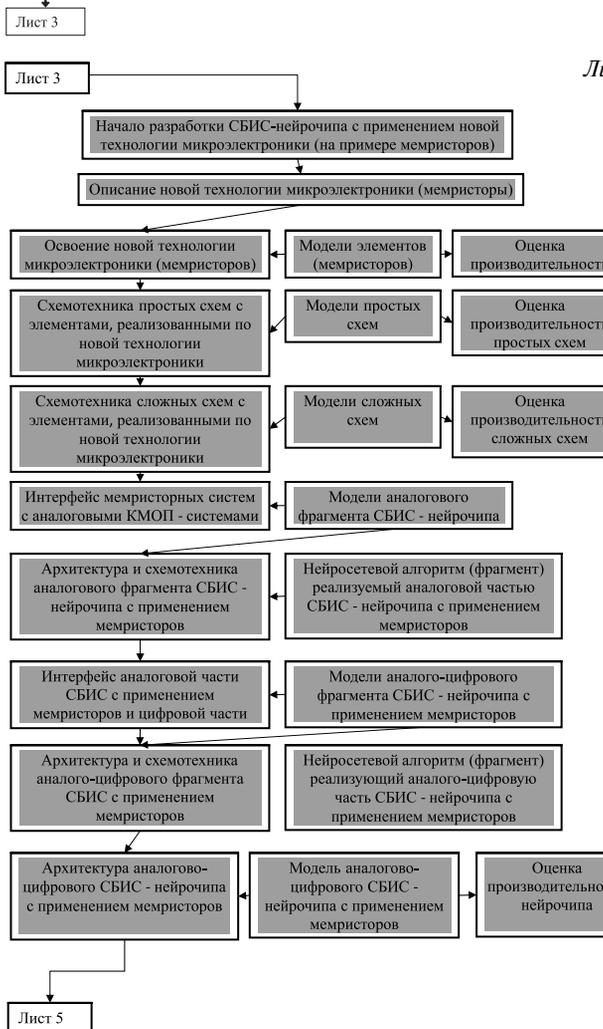
Структура работ по внедрению новой технологии микроэлектроники в высокопроизводительные нейрокомпьютеры (продолжение см. на следующей странице)



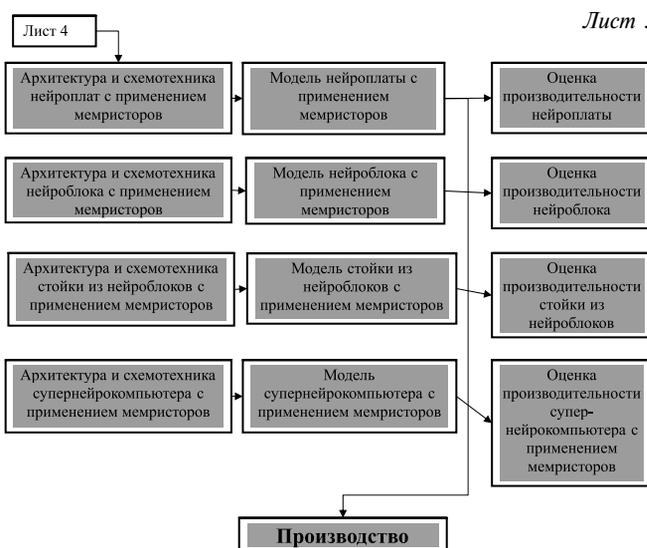
Лист 2



Лист 3



Лист 4



Лист 5

Структура работ по внедрению новой технологии микроэлектроники в высокопроизводительные нейрокомпьютеры (окончание)

1. О классических подходах к решению задач

В основе работ по внедрению новых технологий микроэлектроники в высокопроизводительные нейрокомпьютеры лежит базовая задача заказчика или группа задач группы заказчиков. Указанные задачи можно подразделить на два класса:

- формализуемые;
- неформализуемые.

Для формализуемых задач совместно с разработкой стандартного алгоритма и его программированием на современной персональной ЭВМ анализируется время решения задачи.

Если время решения задачи удовлетворяет заказчика (или заказчиков), то на этом ее решение заканчивается и потребности во внедрении новой технологии микроэлектроники нет. Если время решения задачи не удов-

летворяет заказчика, то, как показано на рис. 1, у разработчика два пути:

- переход к разработке и программированию алгоритма для кластерной суперЭВМ;
- разработка нейросетевого алгоритма решения задачи.

На первом пути анализируются стоимость, габаритные размеры, энергопотребление и время решения задачи на кластерной суперЭВМ для различного числа узлов. Если эти характеристики удовлетворяют заказчика, то на этом решение задачи заканчивается, и потребности во внедрении новой технологии микроэлектроники нет.

Если стоимость, габаритные размеры, энергопотребление или время решения задачи на кластерной суперЭВМ не удовлетворяют заказчика, необходимо разработать и запрограммировать алгоритм решения задачи на суперЭВМ с графическими процессорами.

В этом случае, как и в предыдущем, необходимо провести анализ стоимости, габаритных размеров, энергопотребления и времени решения задачи на суперЭВМ с графическими процессорами. Если эти требования удовлетворяют заказчика, то на этом этапе решение задачи заканчивается, и потребности во внедрении новой технологии микроэлектроники опять отсутствуют.

Если эти требования не удовлетворяются, то необходимо провести программирование и реализацию классических алгоритмов решения задачи на базе программируемых логических интегральных схем (ПЛИС).

Если требования заказчика при этой реализации удовлетворяются, то данная реализация не требует в дальнейшем применения новой технологии микроэлектроники и является самостоятельным решением.

Указанные выше реализации алгоритмов решения задачи на ПЭВМ, на кластерной суперЭВМ, на суперЭВМ с графическими процессорами, на базе ПЛИС, удовлетворяющие требованиям заказчика, не требуют внедрения новых технологий микроэлектроники.

В случае когда время решения задачи на персональной ЭВМ, стоимость, габаритные размеры, энергопотребление и время решения задачи на кластерных суперЭВМ или суперЭВМ на базе графических процессоров, а также на базе ПЛИС, не удовлетворяют заказчика, задача является неформализуемой, и необходимо переходить к разработке реализации нейросетевых алгоритмов решения задач.

2. Структура работ по созданию и реализации нейросетевых алгоритмов решения задач

Разработка нейросетевого алгоритма решения задачи проводится в соответствии с соответствующей методикой [1], апробированной на многих задачах.

После окончания разработки нейросетевого алгоритма решения задачи необходимо провести его программирование на ПЭВМ с графическими ускорителями и анализ времени решения задачи.

Если время решения задачи удовлетворяет заказчика, то на этом разработка данного варианта нейромкомпьютера (ПЭВМ с графическими ускорителями и нейросетевым программным обеспечением) заканчивается, и потребность во внедрении новой технологии микроэлектроники отсутствует.

Если время решения задачи не удовлетворяет заказчика, то необходимо провести разработку и программирование нейросетевого алгоритма на кластерной суперЭВМ и также, как и ранее, провести анализ стоимости, габаритных

размеров, энергопотребления и времени решения задачи в зависимости от числа узлов кластерной суперЭВМ.

Если эти требования заказчика удовлетворяются, то на этом решение задачи заканчивается разработкой нейромкомпьютера в виде варианта кластерной суперЭВМ с нейросетевым программным обеспечением, и потребность во внедрении новой технологии микроэлектроники отсутствует.

Если время решения задачи, стоимость, габаритные размеры и энергопотребление не удовлетворяют заказчика, то необходимо провести разработку и программирование нейросетевого алгоритма для суперЭВМ на базе графических процессоров и также, как и ранее, провести анализ стоимости, габаритных размеров, энергопотребления и времени решения задачи в зависимости от числа графических процессоров.

Если эти требования заказчика удовлетворяются, то на этом решение задачи заканчивается разработкой нейромкомпьютера в виде варианта суперЭВМ на базе определенного числа графических процессоров с нейросетевым программным обеспечением, и потребность во внедрении новой технологии микроэлектроники отсутствует.

Если время решения задачи, стоимость, габаритные размеры и энергопотребление при реализации нейросетевого алгоритма на суперЭВМ на базе графических процессоров не удовлетворяют заказчика, то необходимо провести разработку, программирование и реализацию нейросетевого алгоритма на вычислительной системе на базе ПЛИС и также, как и ранее, провести анализ стоимости, габаритных размеров, энергопотребления и времени решения задачи в зависимости от числа ПЛИС в вычислительной системе.

Если эти требования заказчика удовлетворяются, то на этом решение задачи заканчивается разработкой нейромкомпьютера в виде вычислительной системы на базе ПЛИС с нейросетевыми алгоритмами функционирования, и потребность во внедрении новой технологии микроэлектроники отсутствует.

Указанные выше реализации нейросетевых алгоритмов решения задачи на ПЭВМ, на кластерной суперЭВМ, на суперЭВМ с графическими процессорами, на базе ПЛИС, удовлетворяющие требованиям заказчика, не требуют внедрения новой технологии микроэлектроники.

В случае когда время решения задачи в нейросетевом логическом базисе на персональной ЭВМ, стоимость, габаритные размеры, энергопотребление и время решения задачи на кластерных суперЭВМ или суперЭВМ на базе графических процессоров, а также на базе ПЛИС не удовлетворяют заказчика, необходимо переходить к разработке и реализации нейросетевых алгоритмов решения задач с применением частотно-импульсной модуляции сигналов (ЧИМ) [2] (*spiking neural network*).

3. Структура работ в случае нейросетевых алгоритмов решения задач с применением нейронных сетей с частотно-импульсной модуляцией сигналов

Переход в разработке нейромкомпьютеров к представлению уровня сигналов в виде частоты последовательности узких импульсов является принципиальным, в первую очередь, с точки зрения энергопотребления. Ярким примером этого является разработка нейромчипа TrueNorth фирмы IBM [3]. Это делает данный результат нейросетевых технологий (*spiking neural network*) достаточно самостоятельным и перспективным.

Естественно, начальным этапом этих работ является попытка реализации этой нейросетевой парадигмы на ПЭВМ, суперЭВМ с графическими ускорителями и вычислительными системами на базе ПЛИС. Вариант реализации на кластерной суперЭВМ здесь пропускается вследствие его неперспективности.

Необходимо отметить, что для варианта НСЧИМ (нейронных сетей с частотно-импульсной модуляцией сигналов) необходима разработка специфических разделов:

- теории нейронных сетей;
- нейроматематики;
- нейроуправления,

как алгоритмической основы реализации нейрокомпьютеров с применением НСЧИМ.

Первым этапом работ в этом направлении является разработка нейросетевого алгоритма решения задачи с применением НСЧИМ.

В рамках решения задач по данному первому этапу работ необходимо создание комплекса моделей нейрончиков и нейроморфных ЭВМ на основе НСЧИМ [4], а именно моделей (см. рисунок):

- входного сигнала нейронных сетей;
- функционалов первичной оптимизации;
- нейронных сетей;
- процессов настройки коэффициентов нейронных сетей в нейроморфных ЭВМ;
- процессов настройки коэффициентов нейронной сети с переменной структурой;
- процессов настройки коэффициентов нейронных сетей в нейроморфных ЭВМ с ограничениями на настраиваемые коэффициенты;
- типовых входных сигналов нейронных сетей;
- планов экспериментов и визуализации результатов для проверки качества работы нейронных сетей;
- решения формализуемых и неформализуемых задач с применением больших нейронных сетей для перспективных нейроморфных ЭВМ;
- распараллеливания нейросетевых алгоритмов в соответствии с реальной структурой физических моделей перспективных нейроморфных ЭВМ.

Этот комплекс моделей должен быть реализован на суперЭВМ с применением графических процессоров.

Разработанные нейросетевые алгоритмы должны быть запрограммированы и исследованы на ПЭВМ с графическими ускорителями с анализом стоимости, габаритных размеров, энергопотребления и времени решения задачи для различного числа графических модулей. В случае удовлетворения требований заказчика разработан вариант нейрокомпьютера (ПЭВМ с несколькими графическими платами и нейросетевое (НСЧИМ) программное обеспечение).

Если требования заказчика не удовлетворяются вариантом ПЭВМ с несколькими графическими ускорителями, необходимо перейти к разработке и программированию нейросетевого алгоритма (НСЧИМ) решения задачи на суперЭВМ с применением графических процессоров с соответствующей данному варианту времени решения задачи в зависимости от числа графических процессоров в используемой суперЭВМ. В случае удовлетворения требований заказчика по стоимости, габаритным размерам, энергопотреблению и времени решения задачи разработан вариант нейрокомпьютера (суперЭВМ на базе графических процессоров с нейросетевым (НСЧИМ) программным обеспечением).

Если требования заказчика не удовлетворяются вариантом суперЭВМ на базе графических процессоров, необходимо перейти к разработке и реализации вычислительной системы на базе ПЛИС с реализацией нейросетевых (НСЧИМ) алгоритмов решения задачи. Если данный вариант реализации удовлетворяет требованиям заказчика, то разработан нейрокомпьютер на базе некоторого числа ПЛИС с нейросетевой (НСЧИМ) структурой реализации алгоритма решения задачи.

Отметим, что и эти варианты реализации нейрокомпьютеров с ЧИМ (ПЭВМ с несколькими графическими модулями, суперЭВМ на базе графических процессоров, системы на базе ПЛИС) не требуют развития и внедрения новых технологий микроэлектроники.

Выводы по п. 1—3

Все вышерассмотренные варианты реализации нейрокомпьютеров практически не требуют развития и внедрения новых технологий микроэлектроники. Потребность в них появляется из-за невозможности удовлетворить требования заказчика предыдущими вариантами. Однако необходимо отметить следующее:

- отсутствие реализации нейрокомпьютера по вышеперечисленным вариантам не может служить основанием для попытки внедрения новых технологий микроэлектроники в перспективные нейрокомпьютеры;
- само по себе развитие и внедрение новых технологий микроэлектроники требует проведения значительной части работ из п. 1—3;
- наиболее эффективным вариантом реализации нейрокомпьютера, ориентированном на решение задач в нейросетевом логическом базисе (НСЧИМ) в виде заказного цифрового нейрончика, в настоящее время является разработка нейрончика TrueNorth [3] фирмы IBM. Для реализации нейрокомпьютеров с применением новых технологий микроэлектроники (ниже на примере мемристоров) требуется проведение комплекса работ, описанного ниже, включая практически все работы из п. 3.

4. О разработках нейрончиков и нейроморфных ЭВМ с применением мемристоров

В настоящее время мемристоров являются основной перспективной технологией для будущих нейрокомпьютеров. Развитие непосредственно самой технологии изготовления мемристормых матриц является важным самостоятельным вопросом и не рассматривается в данной работе. Здесь анализируются проблемы, связанные с разработкой нейрончиков и нейроморфных ЭВМ с применением мемристоров, относящиеся конкретно к разработке нейрончиков и нейроморфных ЭВМ с применением мемристоров.

Необходимо отметить, наряду с п. 3, что для вариантов нейрончиков и нейроморфных ЭВМ с применением мемристоров необходима разработка специфических разделов, а именно:

- теории нейронных сетей, алгоритмы адаптации которых ориентированы на специфические мемристормые структуры нейронных сетей и на учет динамических свойств мемристормых как элементов, реализующих весовые коэффициенты;
- нейроматематики, как раздела вычислительной математики, связанного с решением задач с учетом указанной модификации теории нейронных сетей, используемой для решения задач в нейросетевом логическом базисе;

- методов распараллеливания нейросетевых алгоритмов решения задач на специфические "мемристорные", аппаратно реализованные нейросетевые структуры;
- нейроруправления, где для работы в реальном времени в обязательном порядке необходим учет динамических характеристик мемристоров.

В рамках освоения новой технологии микроэлектроники — мемристоров, разработчикам нейрочипов и нейроморфных ЭВМ совместно с разработчиками технологии мемристоров необходимо создать модели элементов (мемристоров и мемристорных матриц) с оценкой производительности в операциях будущих нейрочипов и нейроморфных ЭВМ.

Разработчиками нейросетевых (НСЧИМ) алгоритмов решения задач, архитектуры нейрочипов и нейроморфных ЭВМ, а также специалистами по распараллеливанию нейросетевых алгоритмов и схемотехниками должна быть разработана архитектура, функциональная структура и схемотехника *аналоговой* части нейрочипа с применением мемристоров, а совместно с технологами — технология стыковки мемристорных матриц большого размера с аналоговой частью нейрочипа.

Параллельно с этим должны быть созданы модели указанных аналоговых схем в средах Cadence или Pspice, а также соответствующие модели аналоговых схем с мемристорными блоками и оценка производительности данной части нейрочипа в терминах операций нейроморфных ЭВМ.

Разработчиками нейросетевых (НСЧИМ) алгоритмов решения задач, архитектуры нейрочипов и нейроморфных ЭВМ, специалистами по распараллеливанию нейросетевых алгоритмов, схемотехниками должна быть разработана архитектура, функциональная структура и схемотехника *цифровой* части нейрочипа с применением мемристоров, а отдельно схемотехниками — интерфейс между аналоговой и цифровой частями СБИС с применением мемристоров.

При этом должна быть проведена оценка производительности цифровой части нейрочипа в операциях нейроморфной ЭВМ.

В результате решения задачи распараллеливания нейросетевого алгоритма решения задачи должен быть разработан фрагмент нейросетевого алгоритма, реализуемого цифровой частью нейрочипа. Параллельно с этим должны быть созданы модели цифровой части в среде Cadence.

Интеграция указанных выше работ с интерфейсом нейрочипа с внешней средой позволит провести разработку архитектуры, схемотехники и моделей нейрочипа с применением мемристоров в среде Cadence, а также оценку его производительности в операциях нейроморфной ЭВМ.

Заключительными этапами работ по созданию нейроморфных ЭВМ будут разработка архитектуры и схемотехники нейрочипа, нейроблоков, нейростоек, супернейрокомпьютера, а также работы по оценке их производительности и созданию моделей в среде Cadence.

Результаты последних этапов работ будут переданы на производство с учетом библиотек элементов и технологических норм конкретного производства.

Заключение

Представленная структура позволит, с одной стороны, сильно не рисковать работами по освоению новой технологии микроэлектроники при разработке перспективных нейрокомпьютеров и не тормозить излишними проблема-

ми работы, которые инициируются, а с другой стороны — оптимально выбирать состав работ по освоению новой технологии микроэлектроники с обоснованием необходимости этих работ, кроме горячего энтузиазма разработчиков и ссылок на многочисленные зарубежные работы.

Данная структура может быть основой для выбора состава работ по внедрению новых технологий микроэлектроники в разработки перспективных нейрокомпьютеров. При этом возможны две ошибки:

первая — когда работа по созданию и освоению новых технологий микроэлектроники ставится в некотором значительном отрыве от разработки будущего изделия, где эта технология должна применяться. Соответственно, ошибка здесь заключается в риске самой разработки;

вторая — когда финансирующая организация требует от разработчика функциональных элементов обоснования эффективности применения этих функциональных элементов в количественных характеристиках изделия, в котором они применяются. При этом такое обоснование требует проведения значительного объема работ до начала освоения новой технологии микроэлектроники. Иногда эти требования дополняются требованием подписания у конечного заказчика изделия подтверждения эффективности применения новой технологии микроэлектроники.

Обе указанные ошибки заводят проблему разработки в тупик. В первом случае — из-за рисков отрицательного результата разработки, во втором случае — из-за необходимости проведения дополнительной работы, иногда многократно превышающей объема самой разработки.

На втором пути заказчик своими требованиями доказательства эффективности применения новой технологии микроэлектроники на уровне конечного продукта заведомо завышает объем предварительных работ. Причинами этого, как правило, являются либо профессиональная неподготовленность заказчика, либо опасение наказания за риски при следовании по первому пути разработки.

Естественно, любая предлагаемая технология микроэлектроники может иметь более широкую сферу применения, нежели нейрокомпьютеры. Поскольку данная статья посвящена технологии мемристоров и их внедрению в нейрокомпьютеры, то необходимо отметить, что в части массового производства рынок мемристорных систем будет формироваться не за счет нейрокомпьютеров, а за счет энергонезависимой памяти, коммутационной среды ПЛИС, других разработок.

Это позволяет в значительной степени увеличить допустимые риски разработки нейрокомпьютеров с применением мемристоров.

Список литературы

1. **Галушкин А. И.** О методике решения задач в нейросетевом логическом базисе. Приложение к журналу "Информационные технологии". 2006. № 9. 24 с.
2. **Spiking neural network.** Wikipedia.
3. **Akopyan F., Sawada J., Cassidy A., Alvarez-Icaza R., Arthur J., Merolla P., Imam N., Nakamura Y., Datta P., Nam Gi-Joon, Taba B., Beakes M., Brezzo B., Kuang J. B., Manohar R., Risk W. P., Jackson B., Modha D. S.** TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. Oct. 2015. Vol. 34. N. 10. P. 1537—1557.
4. **Галушкин А. И.** Модели для разработки нейрочипов и нейроморфных ЭВМ // Информатизация и связь. 2015. № 4. С. 35—42.

New Microelectronics Technology and the Development of Advanced Neurocomputers

The article describes main problems and difficulties that are currently available and are likely to be present in the future in the field of development and introduction of new microelectronic technologies in the advanced neuro-computers.

This structure of works represents the operation procedure and composition of the work on introduction of new microelectronics technology in high-performance neurocomputers proposed by the author. Both the operation procedure and scope of work may vary depending on the type of technology, with required, or at least desirable, justification for the exclusion of certain stages of work presented below in this article:

- about classical possible approaches to task solutions;
- work structure for the design and implementation of neural network based algorithms for problem solving;
- the description of spiking neural network algorithms;
- development of neurochips and neuromorphic computers with use of memristor.

Keywords: neural network technology, new technology of microelectronics, spiking neural networks and neurochips, memristors

References

1. Galuskin A. I. O metodike peshenul zadach w neurosetevom logicheskom bazise. *Prilogenie k zhurnaln "Informazionnie nechnologii*, 2006, no. 9, 24 p.
2. Spiking neural network. Wikipedia.
3. Akopyan F., Sawada J., Cassidy A., Alvarez-Icaza R., Arthur J., Merolla P., Imam N., Nakamura Y., Datta P., Nam Gi-Joon, Taba B.,

Beakes M., Brezzo B., Kuang J. B., Manohar R., Risk W. P., Jackson B., Modha D. S. TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Oct. 2015, vol. 34, no. 10, pp. 1537–1557.

4. Galushkin A. I. Modely dlja razrabotki neurochipo v neuro-morfnyh computers, *Informatizacija u svjaz*, 2015, no. 4, pp. 35–42.

УДК 621.391.037.372

В. Н. Дам, аспирант, e-mail: damvan.nhich@gmail.com,
Московский физико-технический институт

Автоматическое распознавание цифровой модуляции радиосигналов с помощью многослойной нейронной сети по кумулянтным признакам

Решается задача распознавания видов цифровой модуляции радиосигналов с помощью многослойной нейронной сети. Распознавание проводится с основными видами цифровой модуляции: 2-PSK, 4-PSK, 8-PSK, 2-FSK, 8-QAM, 16-QAM, 64-QAM. Лучший результат был показан при использовании девяти признаков и 40 нейронов в скрытом слое.

Ключевые слова: модуляция, распознавание вида модуляции, IQ-сигналы, моменты, кумулянты, гауссовый шум, нейронная сеть, Neural Network Toolbox

Введение

В коммуникационной системе важной задачей является распознавание заданных видов модуляции радиосигналов, т. е. радиосигналов с заданными видами модуляции. Эта задача сложная, сложность обусловлена тем, что здесь распознаванию подлежат целые классы сигналов — радиосигналы с заданными видами модуляции и разными значениями параметров модуляции. Распознавание видов модуляции радиосигналов в реальных условиях затрудняется появлением радиосигналов с новыми неизвестными ранее видами модуляции.

1. Модуляция и задача распознавания видов цифровой модуляции

В практике чаще всего используют фазовую (M-PSK), частотную (M-FSK) и квадратурную амплитудную модуляцию (M-QAM), где M — уровень квантования сигнала. Существуют несколько подходов к решению поставленной задачи, но чаще всего применяются три подхода: нейронные сети [1–4], кумулянты высокого порядка [5, 6] и спектральные признаки [7, 8]. В настоящей работе исследуется возможность повышения эффективности распознавания видов цифровой модуляции с помощью ней-

ронной сети. Нейронные сети использовались для распознавания видов модуляции радиосигналов во многих работах, в которых рассматривались сигналы с гауссовым шумом.

В работе [5] использован набор кумулянтов $C_{2,0}, C_{4,0}, C_{8,0}$ для распознавания семи видов модуляции (2-PSK, 4-PSK, 8-PSK, 2-FSK, 8-QAM, 16-QAM, 64-QAM) в качестве признаков для дерева логических решений. В результате при отношении сигнала к шуму (ОСШ) ≥ 5 дБ — практически 100 %-ное распознавание вида модуляции, при ОСШ = 0 дБ результат для 64-QAM слишком низкий (57,8 %). Весь процесс выбора параметров дерева логических решений выполнялся вручную. Этот подход требует больших временных ресурсов, и может возникнуть погрешность в выборе предельных значений для построения дерева логических решений.

В работе [1] использована нейронная сеть для распознавания видов модуляции (2-ASK, 4-ASK, 2-PSK, 4-PSK, 2-FSK, 4-FSK, 16-QAM, V29, V32, 64-QAM). Признаками для входа нейронной сети являются кумулянты $C_{2,0}, C_{1,1}, C_{0,2}, C_{3,0}, C_{2,1}, C_{1,2}, C_{0,3}, C_{4,0}, C_{3,1}, C_{2,2}, C_{1,3}, C_{0,4}$. Табл. 1 показывает, как число нейронов в скрытом слое влияет на результаты распознавания.

Таблица 1

Результаты распознавания при различных количествах нейронов в скрытом слое (10 и 40 нейронов)

Число нейронов в скрытом слое	Отношение сигнала к шуму (ОСШ)				
	-5 дБ	0 дБ	5 дБ	10 дБ	20 дБ
10	73,62 %	87,02 %	99,26 %	89,86 %	99,93 %
40	89,64 %	98,01 %	99,33 %	99,86 %	99,99 %

Таблица 2

Результаты распознавания при использовании 10 нейронов в скрытом слое

ОСШ	Виды модуляции						
	2-ASK	2-PSK	2-FSK	4-ASK	4-PSK	4-FSK	16-QAM
5 дБ	98 %	99,1 %	85,8 %	88,7 %	97 %	91 %	76,7 %
10 дБ	98,3 %	99,5 %	99 %	99,1 %	99 %	99 %	93,3 %
15 дБ	99 %	99,8 %	100 %	99,8 %	99,6 %	99,4 %	99,5 %

Таблица 3

Результаты распознавания в работе [4]

Число нейронов в скрытом слое	ОСШ	
	10 дБ	20 дБ
5	81,5 %	83,1 %
10	97,4 %	99,6 %
15	93,7 %	99,6 %

Таблица 4

Исходные параметры

Параметры	Значение
Частота дискретизации, Гц	500 000
Символьная скорость, Гц	5000

В работе [2] описана нейронная сеть для распознавания видов модуляции: 2-ASK, 2-PSK, 2-FSK, 4-ASK, 4-PSK, 4-FSK, 16-QAM. В качестве входа использованы только спектральные признаки:

- максимальное значение спектральной плотности мощности нормированной центрированной мгновенной амплитуды γ_{\max} ;
- стандартное отклонение абсолютного значения центрированных нелинейных компонент мгновенной фазы σ_{ap} ;
- стандартное отклонение значения центрированных нелинейных компонент мгновенной фазы σ_{dp} ;
- стандартное отклонение абсолютного значения нормированной центрированной мгновенной амплитуды σ_{aa} ;
- стандартное отклонение абсолютного значения нормированной центрированной мгновенной частоты σ_{af} .

Авторы экспериментировали с различным числом нейронов в скрытом слое (1, 5, 10, 15 нейронов). В табл. 2 представлены результаты при использовании 10 нейронов в скрытом слое.

В статье [3] использованы два спектральных признака (максимальное значение спектральной плотности мощности нормированной центрированной мгновенной амплитуды γ_{\max} , среднее значение центрированного нормированного абсолютного мгновенного фазового отклонения m_{pd}) в качестве входа нейронной сети для распознавания восьми видов модуляции 2-ASK, 2-FSK, 4-ASK, 4-FSK, 2-PSK, 4-PSK, DPSK и 16-QAM. Для каждого вида модуляции использовано 200 сигналов. Средняя вероятность распознавания составила 80 % при ОСШ = 10 дБ и 85 % при ОСШ = 20 дБ.

В работе [4] исследованы шесть видов цифровой модуляции: 2-ASK, 4-ASK, 2-PSK, 4-PSK, 2-FSK, 4-FSK с помощью шести спектральных признаков (пять признаков из [2] и коэффициент P). Результаты их работы внесены в табл. 3.

Из изложенного выше следует, что применение различных признаков и структур нейронной сети в задаче распознавания вида модуляции приводит к различной точности решения данной задачи. Эффективное применение кумулянтных признаков и правильный выбор структуры нейронной сети для решения задачи распознавания вида цифровой модуляции и является целью настоящей работы.

2. Постановка задачи

Отправитель передает сигнал с одним из следующих видов цифровой модуляции: 2-PSK, 4-PSK, 8-PSK, 2-FSK, 8-QAM, 16-QAM, 64-QAM. Получатель принимает искаженный аддитивными комплексными гауссовыми белыми шумами $\xi(t)$ переданный сигнал $z(t), t = 0, T$,

$$z(t) = a(t) + \xi(t), \quad (1)$$

где $a(t)$ — переданный сигнал без шума.

Принимаемый сигнал проходит через цифровой понижающий преобразователь (Digital Down Converter), и на выходе этого преобразователя получаем I (синфазную составляющую) и Q (квадратурную составляющую) сигналы, которые являются исходными данными для поставленной задачи. Получателю известны следующие параметры переданного сигнала: частота дискретизации и символьная скорость (табл. 4).

Получателю требуется распознавать используемый вид цифровой модуляции.

3. Выбор признаков для входа нейронной сети

Пусть получаемый модулированный сигнал имеет вид

$$a(t) = A(t)\cos[\omega(t)t + \phi(t)]$$

или

$$a(t) = A(t)\cos[\omega(t)t]\cos\phi(t) - A(t)\sin[\omega(t)t]\sin\phi(t).$$

Если частота не меняется, тогда модулированный сигнал можно записать в виде

$$a(t) = A(t)\cos\omega_c t \cos\phi(t) - A(t)\sin\omega_c t \sin\phi(t),$$

где ω_c — несущая частота.

Следовательно,

$$a(t) = I(t)\cos\phi(t) - Q(t)\sin\phi(t),$$

где $I(t) = A(t)\cos\omega_c t$ и $Q(t) = A(t)\sin\omega_c t$.

Модулирующий сигнал $s(t)$ представлен синфазной составляющей $I(t)$ и квадратурной составляющей $Q(t)$:

$$s(t) = I(t) + jQ(t). \quad (2)$$

Тогда $a(t)$ можно записать в виде

$$a(t) = \text{Re}[s(t)\exp(j\omega_c t)]. \quad (3)$$

Рассмотрим случайный комплексный модулирующий сигнал $s_i(t) = I(t) + jQ(t)$ [12] и комплексно сопряженный к нему сигнал $\bar{s}_i(t) = I(t) - jQ(t)$. Такому сигналу соответствует двумерная плотность распределения вероятностей $h(x, y)$ и соответствующая двумерная характеристическая функция $\theta(u, v)$, являющаяся Фурье-преобразованием двумерной плотности распределения вероятностей $h(x, y)$.

Моменты случайного сигнала представляют собой множество параметров, которое характеризует распределение плотности вероятностей случайного процесса. Однако это множество не является единственным и тем более наилучшим для решения ряда практических задач.

Иное множество параметров — так называемых кумулянтов, обладает свойствами, более полезными как с теоретической, так и с практической точки зрения.

Коэффициенты $C_{n, m}$ разложения логарифма характеристической функции $\theta(u, v)$ в ряд Тейлора

$$\ln\theta(u, v) = \sum_{n, m=0}^{\infty} C_{n, m} (iu)^n (iv)^m / n!m! \quad (4)$$

являются кумулянтами данного распределения [11]. Порядком кумулянта r называется сумма $r = n + m$.

К важнейшим свойствам кумулянтов относится тот факт, что для двумерной нормальной плотности распределения только кумулянты первого и второго порядков отличны от нуля. Таким образом, кумулянты характеризуют негауссовость случайного процесса, а поскольку модулированные процессы существенно негауссовы, то кумулянты служат индикаторами соответствующих видов модуляции.

Для комплексного цифрового модулированного сигнала $s_i(t)$ моменты двумерного распределения $E_{n, m}$ имеют вид $E_{n, m} = \text{Ex}(s_i^n \bar{s}_i^m)$ [6], где Ex — символ математического ожидания, \bar{s}_i — комплексно-сопряженный сигнал. Порядком момента, как и для кумулянта, является сумма $r = n + m$. Кумулянты r -го порядка выражаются через моменты до r -го порядка включительно. Предполагаем, что сигнал $s(t)$ — центрированный.

В табл. 5 представлены формулы для вычисления кумулянтов второго, четвертого, шестого и восьмого порядков. Значения кумулянтов являются комплексными числами. Основные отличия значений кумулянтов для различных видов манипуляции проявляются в значениях их действительных частей. Поэтому в качестве распознающего признака здесь приняты значения действительных частей кумулянтов (далее будем называть значением кумулянтов) [5].

Таблица 5

Формулы для вычисления кумулянтов

Порядок	Кумулянт	Выражение через моменты
2	$C_{2,0}$	$C_{2,0} = E_{2,0} = E[s^2(\bar{s})^0]$
2	$C_{1,1}$	$C_{1,1} = E_{1,1} = E[s(\bar{s})]$
4	$C_{4,0}$	$E_{4,0} - 3(E_{2,0})^2$
4	$C_{3,1}$	$E_{3,1} - 3E_{2,0}E_{1,1}$
4	$C_{2,2}$	$E_{2,2} - (E_{2,0})^2 - 2(E_{1,1})^2$
6	$C_{6,0}$	$E_{6,0} - 15E_{2,0}E_{4,0} + 30(E_{2,0})^3$
6	$C_{5,1}$	$E_{5,1} - 10E_{2,0}E_{3,1} - 5E_{1,1}E_{4,0} + 30(E_{2,0})^2E_{1,1}$
6	$C_{4,2}$	$E_{4,2} - E_{2,0}E_{4,0} - 8E_{1,1}E_{3,1} - 6E_{2,0}E_{2,2} + 6(E_{2,0})^3 + 24(E_{1,1})^2E_{2,0}$
6	$C_{3,3}$	$E_{3,3} - 6E_{2,0}E_{3,1} - 9E_{1,1}E_{2,2} + 18(E_{2,0})^2E_{1,1} + 12(E_{1,1})^3$
8	$C_{8,0}$	$E_{8,0} - 35(E_{4,0})^2 - 630(E_{2,0})^4 + 420(E_{2,0})^2E_{4,0}$
8	$C_{7,1}$	$E_{7,1} - 35E_{4,0}E_{3,1} - 630(E_{2,0})^3E_{1,1} + 210E_{4,0}E_{2,0}E_{1,1} + 210(E_{2,0})^2E_{3,1}$
8	$C_{6,2}$	$E_{6,2} - 15E_{4,0}E_{2,2} - 20(E_{3,1})^2 + 30E_{4,0}(E_{2,0})^2 + 60E_{4,0}(E_{1,1})^2 + 240E_{3,1}E_{1,1}E_{2,0} + 90E_{2,2}(E_{2,0})^2 - 90(E_{2,0})^4 - 540(E_{2,0})^2(E_{1,1})^2$
8	$C_{5,3}$	$E_{5,3} - 5E_{4,0}E_{3,1} - 30E_{3,1}E_{2,2} + 90E_{3,1}(E_{2,0})^2 + 120E_{3,1}(E_{1,1})^2 + 180E_{2,2}E_{1,1}E_{2,0} + 30E_{4,0}E_{2,0}E_{1,1} - 270(E_{2,0})^3E_{1,1} + 360(E_{1,1})^3E_{2,0}$
8	$C_{4,4}$	$E_{4,4} - (E_{4,0})^2 - 18(E_{2,2})^2 - 16(E_{3,1})^2 - 54(E_{2,0})^4 - 144(E_{1,1})^4 - 432(E_{2,0})^2(E_{1,1})^2 + 12E_{4,0}(E_{2,0})^2 + 96E_{3,1}E_{1,1}E_{2,0} + 144E_{2,2}(E_{1,1})^2 + 72E_{2,2}(E_{2,0})^2 + 96E_{3,1}E_{2,0}E_{1,1}$

Правилами, по которым надо выбрать кумулянты для входа нейронной сети, являются следующие:

- значения кумулянтов не должны зависеть от уровня (ОСШ), т. е. значения кумулянтов не должны значительно изменяться, когда изменяются значения ОСШ;
- не должно быть похожих кумулянтов, значения действительных частей которых одинаковые.

Рассмотрим сигнал $s(t) = a + bj$. Для такого сигнала кумулянт

$$C_{2,0} = E_{2,0} = (a + bj)^2 = a^2 - b^2 + 2abj.$$

Если $a = b$, то $C_{2,0}$ не зависит от величины шума. Далее кумулянт $C_{1,1}$ вычисляется по формуле

$$C_{1,1} = E_{1,1} = (a + bj)(a - bj) = a^2 + b^2.$$

Таблица 6

Значения кумулянтов при различных значениях ОСШ для 2-PSK

Кумулянт \ ОСШ	20 дБ	10 дБ	5 дБ	0 дБ
C20	-0,0387	-0,0345	-0,0248	-0,0018
C11	1,0090	1,0919	1,3213	1,9256
C40	-0,0095	0,0169	-0,1152	-0,2994
C31	0,0786	0,0402	0,0203	-0,0857
C22	-0,9976	-0,9818	-1,0387	-0,8104
C60	0,0122	0,0751	0,3629	2,5532
C51	0,0518	-0,0643	0,4749	0,1877
C42	-0,4330	-0,2126	-0,1819	0,0769
C33	3,9949	3,8893	4,1683	2,5281
C80	-0,9334	-0,7491	-2,1701	5,9027
C71	0,0799	-0,2008	2,2319	35,4588
C62	-0,5843	1,0589	-8,6745	-64,5972
C53	3,4696	-1,0747	-5,8719	-36,8123
C44	-15,3429	4,1966	84,7813	944,1348

Таблица 7

Значения кумулянтов при различных значениях ОСШ для 16-QAM

Кумулянт \ ОСШ	20 дБ	10 дБ	5 дБ	0 дБ
C20	-0,0152	-0,0043	-0,0137	-0,0552
C11	1,0092	1,1023	1,3261	1,9717
C40	-0,6600	-0,7131	-0,6053	-0,6686
C31	0,0348	0,0011	0,0131	-0,0657
C22	-0,6815	-0,6889	-0,7359	-0,5849
C60	-0,1782	-0,0133	-0,1394	0,2244
C51	2,0093	2,2433	1,8996	2,1324
C42	-0,1728	-0,0078	-0,1234	-0,1282
C33	2,0929	2,08168	2,3395	1,4557
C80	-13,0162	-15,8061	-12,422	-56,3791
C71	1,7873	0,5779	0,6476	15,4644
C62	-29,5409	-37,526	-47,6135	-126,581
C53	1,2313	-0,7299	-3,9638	-119,967
C44	19,3195	43,9592	142,9622	1157,298

Таблица 8

Основные параметры нейронной сети

Число слоев	3
Число входов	3, 9
Число нейронов в скрытом слое	40, 80, 120
Число нейронов в выходном слое	7
Максимальное число эпох	1000
Функция активации на скрытом слое	Сигмоидальная
Функция активации на выходном слое	Softmax

Следовательно, значение $C_{1,1}$ зависит от величины шума. Если уровень шума повышается, то $C_{1,1}$ тоже повышается, и наоборот.

В табл. 6, 7 представлены значения кумулянтов при различных значениях ОСШ для 2-PSK и 16-QAM. Значения кумулянтов $C_{1,1}$, $C_{7,1}$, $C_{6,2}$, $C_{5,3}$, $C_{4,4}$ (выделено жирным шрифтом) зависят от уровня шума, поэтому в качестве признаков нельзя их выбирать.

Кроме того, значения кумулянтов могут зависеть от шума при низких ОСШ из-за больших значений коэффициентов в формулах вычисления кумулянтов в табл. 5, т. е. при небольших изменениях значений моментов от шума следует значительное изменение кумулянтов.

Далее рассмотрим $C_{3,1}$ и $C_{1,3}$:

$$C_{3,1} = (a + bj)^3(a - bj);$$

$$C_{1,3} = (a + bj)(a - bj)^3.$$

$C_{1,3}$ является комплексным сопряженным $C_{3,1}$, поэтому значения действительных частей $C_{3,1}$ и $C_{1,3}$ равны.

Аналогично, из табл. 5 в качестве признаков выбираем девять кумулянтов: $C_{2,0}$, $C_{4,0}$, $C_{3,1}$, $C_{2,2}$, $C_{6,0}$, $C_{5,1}$, $C_{4,2}$, $C_{3,3}$, $C_{8,0}$.

4. Применение нейронной сети в задаче распознавания видов модуляции

Моделирование нейронной сети проводится в MATLAB с помощью Neural Network Toolbox. Входом нейронной сети является набор значений кумулянтов.

Структура нейронной сети. Основные параметры нейронной сети проведены в табл. 8.

Для составления базы данных для обучения нейронной сети были сгенерированы 7000 сигналов в MATLAB (1000 сигналов по каждому виду модуляции). Из них 5600 сигналов для обучения, 700 — для проверки и 700 — для тестирования. Входом является матрица размером 3×7000 или 9×7000 , а выходом является матрица размером 7×7000 .

Процесс обучения проводится в режиме offline, максимальное число эпох равно 1000 и алгоритм обучения является Levenberg — Marquardt [13].

Результаты тестирования. Результаты тестирования выполнены для двух вариантов: использование трех и девяти признаков.

Использование трех признаков. В табл. 9 приведены результаты распознавания в % при использовании трех признаков [5]: $C_{2,0}$, $C_{4,0}$, $C_{8,0}$; число нейронов в скрытом слое 40.

Процент распознавания при небольшом ОСШ (0,5 дБ) низкий, особенно для видов модуляции 16-QAM, 64-QAM (47,1 % и 57,1 %). Отсюда следует, что необходимо корректировать предельные значения в дереве логических решений в работе [5]. Этот результат показывает преимущество применения нейронной сети, поскольку здесь отсутствует необходимость выбора диапазона признаков, которая существует в классическом методе.

Использование девяти признаков. Набор состоит из следующих признаков $C_{2,0}$, $C_{4,0}$, $C_{3,1}$, $C_{2,2}$, $C_{6,0}$, $C_{5,1}$, $C_{4,2}$, $C_{3,3}$, $C_{8,0}$. Эти признаки выбраны согласно правилам в п. 3. Результаты распознавания, которые получаются при числе нейронов в скрытом слое, равном 40, внесены в табл. 10.

Полученные результаты распознавания для большинства видов модуляции достаточно хорошие, однако результаты по распознаванию 16-QAM и 64-QAM еще не очень высокие, особенно при низких ОСШ.

В табл. 11 приведены результаты при использовании 80 нейронов в скрытом слое. Результаты стали хуже для всех видов модуляции.

Результаты, которые получаются при числе нейронов в скрытом слое, равном 120, внесены в табл. 12. Невысокие результаты только для видов модуляции 16-QAM и 64-QAM, при ОСШ = 0 дБ процент распознавания стал лучше, но при ОСШ = 5, 10, 20 дБ стал хуже, чем процент распознавания при использовании 40 нейронов в скрытом слое.

Из анализа приведенных выше таблиц следует вывод, что вероятность распознавания не прямо пропорциональна числу нейронов в скрытом слое. Единственным методом выбора правильного числа нейронов в скрытом слое является проведение экспериментов. При исполь-

зовании 40 нейронов в скрытом слое показан лучший результат. Хорошие результаты получены для видов модуляции 2-PSK, 4-PSK, 8-PSK, 2-FSK, 8-QAM, а для видов модуляции 16-QAM, 64-QAM результаты хуже.

Заключение

В данной работе использована нейронная сеть, структура которой подходит для распознавания видов цифровой модуляции: 2-PSK, 4-PSK, 8-PSK, 2-FSK, 8-QAM, 16-QAM, 64-QAM. Лучший результат показан при использовании девяти признаков и 40 нейронов в скрытом слое. Для видов модуляции 2-PSK, 4-PSK, 8-PSK, 2-FSK, 8-QAM результаты достаточно хорошие при любом значении ОСШ (0, 5, 10, 20 дБ), а для видов модуляции 16-QAM, 64-QAM результаты ухудшаются при ОСШ (0,5 дБ). Новизна статьи заключается в том, что найдены правила выбора признаков и подходящая структура нейронной сети, которые обеспечивают хорошие результаты распознавания. Дальнейшие исследования будут направлены на повышения точности распознавания видов модуляции 16-QAM, 64-QAM, а также на исследовании нейросетевого подхода для распознавания видов цифровой модуляции реальных модулированных сигналов.

Список литературы

1. **Wong M. L. D., Nandi A. K.** Automatic digital modulation recognition using artificial neural network and genetic algorithm // *Signal Processing*. 2004. Vol. 84. P. 351–365.
2. **Khan M. A., Bangash Y. A.** Automatic Modulation Recognition of Communication Signals // *Master Thesis Electrical Engineering*. 04-2012. P. 34–58.
3. **El-Khoribi R. A., Shoman M. A. I., Mohammed A. G. A.** Automatic Digital Modulation Recognition using Artificial Neural Network in Cognitive radio // *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*. May – June, 2014. Vol. 3. Is. 3. P. 132–136.
4. **Nandi A. K., Azzouz E. E.** Modulation recognition using artificial neural networks // *Signal Processing*. 1997. Vol. 56. P. 165–175.
5. **Авельян Э. Д., Дам В. Н.** К выбору кумулянтных признаков в задаче распознавания видов цифровой модуляции радиосигналов // *Информатизация и связь*. 2015. № 4. С. 11–15.
6. **Young A. F.** Classification of digital modulation types in multipath environments // *Master's Thesis. Naval Postgraduate School, Monterey, CA 93943-5000, California*. June 2008. P. 1–65.
7. **Li J., Wang J., Fan X., Zhang Y.** Automatic Digital Modulation Recognition Using Feature Subset Selection // *Progress In Electro magnetics Research Symposium, Hangzhou, China*. March 24–28, 2008. P. 351–354.
8. **Kubankova A., Kubanek B.** Digital Modulation Recognition Based on Feature, Spectrum and Phase Analysis and its Testing with Disturbed Signals // *TELE-INFO 11 – Recent Researches in Telecommunications, Informatics, Electronics and Signal Processing*. 01. 2011. P. 162–166.
9. **Azarbad M., Hakimi S., Ebrahimzadeh A.** Automatic Recognition of Digital Communication Signal // *International Journal of Energy, Information and Communications*. November, 2012. Vol. 3, Is. 4. P. 21–33.
10. **Bagga J., Tripathi N.** Automatic modulation classification using statistical features in fading environment // *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*. August 2013. Vol. 2. Iss. 8. P. 3701–3709.
11. **Малахов А. Н.** Кумулянтный анализ случайных негауссовских процессов и их преобразований. М.: Советское радио. 1978. 376 с.
12. **Valkama M.** Complex-valued signals and systems — Basic principles and applications to radio communications and radio signal processing // *Dept. of Communications Engineering Tampere University of Technology*. 2008, pp. 1–62. <http://www.cs.tut.fi/TLT-9707/presentation/complex-Signal-and-Radios-short-2pp.pdf>.
13. **Suratgar A. A., Tavakoli M. B., Hoseinabadi A.** Modified Levenberg — Marquardt method for neural networks training // *International Journal of Computer, Electrical, Automation, Control and Information Engineering*. 2007. Vol. 1, N. 6. P. 1745–1747.

Таблица 9

Результаты при использовании трех признаков

Вид модуляции \ ОСШ	ОСШ			
	0 дБ	5 дБ	10 дБ	20 дБ
2-PSK	100	100	100	100
4-PSK	67,6	80,6	100	100
8-PSK	80,0	90,9	98,9	100
2-FSK	98,2	98,2	100	100
8-QAM	100	100	100	100
16-QAM	47,1	51,3	61,9	66,4
64-QAM	69,0	57,1	75,3	82,6

Таблица 10

Результаты распознавания при использовании 40 нейронов в скрытом слое

Вид модуляции \ ОСШ	ОСШ			
	0 дБ	5 дБ	10 дБ	20 дБ
2-PSK	100	100	100	100
4-PSK	94,2	100	100	100
8-PSK	95,0	98,9	100	100
2-FSK	97,2	97,9	100	100
8-QAM	100	100	100	100
16-QAM	55,6	66,4	87,4	88,0
64-QAM	56,7	72,9	81,1	87,5

Таблица 11

Результаты при использовании 80 нейронов в скрытом слое

Вид модуляции \ ОСШ	ОСШ			
	0 дБ	5 дБ	10 дБ	20 дБ
2-PSK	100	100	100	100
4-PSK	81,9	100	100	100
8-PSK	84,5	100	100	100
2-FSK	94,4	92,6	89,8	99,0
8-QAM	100	100	100	100
16-QAM	50,5	63,2	68,0	82,5
64-QAM	63,2	72,7	69,9	80,0

Таблица 12

Результаты при использовании 120 нейронов в скрытом слое

Вид модуляции \ ОСШ	ОСШ			
	0 дБ	5 дБ	10 дБ	20 дБ
2-PSK	100	100	100	100
4-PSK	90,5	97,2	100	100
8-PSK	87,4	100	100	100
2-FSK	96,9	100	100	100
8-QAM	100	100	100	100
16-QAM	59,4	74,1	75,9	81,9
64-QAM	63,7	70,9	71,9	77,2

Automatic Digital Modulation Recognition of Radio Signals Using Multilayer Neural Network with Cumulant Features

The article solves the problem of digital modulation recognition of radio signals based on the using multilayer neural network with cumulant features. Recognition is performed with the basic types of digital modulation 2-PSK, 4-PSK, 8-PSK, 2-FSK, 8-QAM, 16-QAM, 64-QAM. Features for the entrance of the neural network are cumulants $C_{2,0}$, $C_{4,0}$, $C_{3,1}$, $C_{2,2}$, $C_{6,0}$, $C_{5,1}$, $C_{4,2}$, $C_{3,3}$, $C_{8,0}$. MATLAB R2013b used to generate the data and perform the simulation, the article studied the Gaussian noise. The best result was shown using 9 features and 40 neurons in the hidden layer. These results are good enough, but recognition result of 16-QAM and 64-QAM is not very high, especially at low SNR. Increase of recognition results of 16-QAM, 64-QAM is a work in further studies.

Keywords: modulation, modulation recognition, IQ signals, moments, cumulants, Gaussian noise, signal to noise ratio (SNR), neural network, learning algorithm, Neural Network Toolbox

References

1. Wong M. L. D., Nandi A. K. Automatic digital modulation recognition using artificial neural network and genetic algorithm, *Signal Processing* 84, 2004, pp. 351–365.
2. Khan M. A., Bangash Y. A. Automatic Modulation Recognition of Communication Signals, *Master Thesis Electrical Engineering*, 04-2012, pp. 34–58.
3. El-Khoribi R. A., Shoman M. A. I., Mohammed A. G. A. Automatic Digital Modulation Recognition using Artificial Neural Network in Cognitive radio, *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*. May – June, 2014, vol. 3, Is. 3, pp. 132–136.
4. Nandi A. K., Azzouz E. E. Modulation recognition using artificial neural networks, *Signal Processing* 56, 1997, pp. 165–175.
5. Aved'yan E. D., Dam V. N. K vyboru kumuljantnyh priznakov v zadache raspoznavanija vidov cifrovoj moduljacji radiosignalov, *Informatizacija i syjaz'*, 2015, no. 4, pp. 11–15. (Aved'yan E. D., Dam V. N. To choosing the cumulant features in the problem of recognition of types of digital modulation of radio signals. *Informatization and Communication*. 2015. № 4. P. 11–15.)
6. Young A. F. Classification of digital modulation types in multipath environments, *Master's Thesis. Naval Postgraduate School, Monterey*, CA 93943-5000, California, June 2008, pp. 1–65.
7. Li J., Wang J., Fan X., Zhang Y. Automatic Digital Modulation Recognition Using Feature Subset Selection, *Progress In Electro magnetic Research Symposium*, Hangzhou, China, March 24–28, 2008, pp. 351–354.
8. Kubankova A., Kubanek B. Digital Modulation Recognition Based on Feature, Spectrum and Phase Analysis and its Testing with Disturbed Signals, *TELE-INFO 11 – Recent Researches in Telecommunications, Informatics, Electronics and Signal Processing*, 01/2011, pp. 162–166.
9. Azarbad M., Hakimi S., Ebrahimzadeh A. Automatic Recognition of Digital Communication Signal, *International Journal of Energy, Information and Communications*. November, 2012, vol. 3, Is. 4, pp. 21–33.
10. Bagga J., Tripathi N. Automatic modulation classification using statistical features in fading environment, *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, August 2013, vol. 2, Is. 8, pp. 3701–3709.
11. Malahov A. N. *Kumuljantnyj analiz sluchajnyh negaussovyh processov i ih preobrazovanij*. Moscow: Sovetskoe radio, 1978, 376 p. (Malakhov A. N. Cumulant analysis of non-Gaussian random processes and their transformations. M.: Soviet Radio. 1978. 376 p.)
12. Valkama M. Complex-valued signals and systems — Basic principles and applications to radio communications and radio signal processing, *Dept. of Communications Engineering Tampere University of Technology*, 2008, pp. 1–62.
13. Suratgar A. A., Tavakoli M. B., Hoseinabadi A. Modified Levenberg — Marquardt method for neural networks training, *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 2007, vol. 1, no. 6, pp. 1745–1747.

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Технический редактор Е. В. Конова.

Корректор Е. В. Комиссарова.

Сдано в набор 29.04.2016. Подписано в печать 22.06.2016. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT716. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Авансд солюшнз". Отпечатано в ООО "Авансд солюшнз".

119071, г. Москва, Ленинский пр-т, д. 19, стр. 1.