

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

8(216)
2014

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

ОБЩИЕ ВОПРОСЫ

Нечаев В. В. Современные бионические информационные технологии 3

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

Кухаренко Б. Г., Солнцева М. О. Итеративная кластеризация траекторий управляемых объектов в многомерном пространстве характеристик 11
Соколов И. В., Агеев А. М., Волобуев М. Ф., Замыслов М. А., Мальцев А. М., Михайленко С. Б. Оценка статистических характеристик сигнала на выходе резервированной системы, управляемой нечетким контроллером 17

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

Имамвердиев Я. Н. Модель ситуационного управления информационной безопасностью электронного правительства 24
Куравский Л. С., Мармалюк П. А., Баранов С. Н., Алхимов В. И., Юрьев Г. А., Артюхина С. В. Марковские модели глазодвигательной активности и их применение для тестирования профессиональных навыков 34

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

Гаврилов С. В., Иванова Г. А., Манукян А. А. Новые проблемы логико-топологического синтеза заказных сложно-функциональных блоков и методы их решения 44
Ракитин В. В., Русаков С. Г. Электрические модели мемристорных элементов 50
Анисимов В. И., Гридин В. Н., Шабани М. А. Методы построения высокопроизводительных распределенных систем автоматизации схемотехнического проектирования 59

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

Авдошин С. М., Песоцкая Е. Ю. Экосистемы программного обеспечения: инновации в ИТ 64

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Амирасланова З. Н. Автоматизация формирования чертежей арматуры железобетонных конструкций 70
Новак Е. В. Информационно-коммуникационные технологии: оценка эффективности 74

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.
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.
ГАЛУШКИН А. И., д. т. н., проф.
ДИМИТРИЕНКО Ю. И., д. ф.-м. н., проф.
ДОМРАЧЕВ В. Г., д. т. н., проф.
ЗАГИДУЛЛИН Р. Ш., к. т. н., доц.
ЗАРУБИН В. С., д. т. н., проф.
ИСАЕНКО Р. О., к. т. н., с.н.с.
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.
КУХАРЕНКО Б. Г., к. ф.-м. н., доц.
ЛЬВОВИЧ Я. Е., д. т. н., проф.
МИХАЙЛОВ Б. М., д. т. н., проф.
НЕЧАЕВ В. В., к. т. н., проф.
РЯБОВ Г. Г., чл.-корр. РАН, д. т. н., проф.
СОКОЛОВ Б. В., д. т. н., проф.
УСКОВ В. Л., к. т. н. (США)
ФОМИЧЕВ В. А., д. т. н., проф.
ЧЕРМОШЕНЦЕВ С. Ф., 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

GENERAL QUESTIONS

Nechaev V. V. Modern Bionics Information Technology 3

INTELLIGENT SYSTEMS AND TECHNOLOGIES

Kukhareenko B. G., Solntseva M. O. Iterative Clustering Trajectories of Objects under Control in Multi-Dimensional Characteristic Space 11

Sokolov I. V., Agueyev A. M., Volobuev M. F., Zamyslov M. A., Maltsev A. M., Mikhaylenko S. B. Assessment of the Statistical Characteristics of the Signal at the Output of the Reserve System, Controlled by Fuzzy Controller 17

MODELING AND OPTIMIZATION

Imamverdiyev Y. N. A Model of Situational Management of E-Government Information Security 24

Kuravsky L. S., Marmalyuk P. A., Baranov S. N., Alkhimov V. I., Yuryev G. A., Artyukhina S. V. Markov Models of Oculomotor Activity and Their Application for Professional Skills Assessment 34

CAD-SYSTEMS

Gavrilov S. V., Ivanova G. A., Manukyan A. A. New Problems of Logic-Topological Synthesis of Custom IP-Blocks and Methods for Their Solution 44

Rakitin V. V., Rusakov S. G. Electrical Models of Memristor Elements. 50

Anisimov V. I., Gridin V. N., Shabani M. A. Methods of Construction of High-Performance Distributed Systems Automation Circuit Design. 59

SOFTWARE ENGINEERING

Avdoshin S. M., Pesotskaya E. Y. Software Ecosystem: Innovations in IT 64

APPLIED INFORMATION TECHNOLOGIES

Amiraslanova Z. N. Automation of Reinforcement Design in Ferroconcrete Construction 70

Novak E. V. Information and Communication Technologies: Measuring of the Effectiveness. 74

Editor-in-Chief:

Stempkovsky 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.
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.
Galushkin A. I., Dr. Sci. (Tech.), Prof.
Dimitrienko Yu. I., Dr. Sci. (Phys.-Math.), Prof.
Domrachev V. G., Dr. Sci. (Tech.), Prof.
Zagidullin R. Sh., Cand. Sci. (Tech.), Ass. Prof.
Zarubin V. S., Dr. Sci. (Tech.), Prof.
Isaenko R. O., Cand. Sci. (Tech.)
Karpenko A. P., Dr. Sci. (Phys.-Math.), Prof.
Kolin K. K., Dr. Sci. (Tech.)
Kulagin V. P., Dr. Sci. (Tech.), Prof.
Kureichik V. M., Dr. Sci. (Tech.), Prof.
Kukhareenko B. G., Cand. Sci. (Phys.-Math.)
Ljvovich Ya. E., Dr. Sci. (Tech.), Prof.
Mikhailov B. M., Dr. Sci. (Tech.), Prof.
Nechaev V. V., Cand. Sci. (Tech.), Ass. Prof.
Ryabov G. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Sokolov B. V., Dr. Sci. (Tech.)
Uskov V. L. (USA), Dr. Sci. (Tech.)
Fomichev V. A., Dr. Sci. (Tech.), Prof.
Chermoshentsev S. F., Dr. Sci. (Tech.), Prof.
Shilov 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"

ОБЩИЕ ВОПРОСЫ GENERAL QUESTIONS

УДК 004.8

В. В. Нечаев, канд. техн. наук., проф.,
Московский институт радиотехники и электроники,
e-mail: nechaev@mirea.ru

Современные бионические информационные технологии

Посвящается пятидесятилетию бионики

"Нет ничего более изобретательного, чем природа"
Марк Туллий Цицерон (106—43 гг. до н. э.)

"Ум человеческий ... никогда не найдет изобретения более прекрасного, более легкого, более быстрого, чем изобретение природы, ибо в ее изобретениях нет ничего недостаточного и ничего излишнего..."
Леонардо да Винчи (1452—1519 гг.)

Статья посвящена одному из актуальных и перспективных научно-практических направлений в области информационных технологий — информационной бионике (ИБ). Цель работы — привлечение внимания молодых специалистов и ученых к данной проблематике. ИБ рассматривается как комплексная технология, состоящая из трех фаз: определения биологического прототипа, его модельного представления и воплощения модели в техническом аналоге — устройстве или процессе. Предполагается, что содержание каждой фазы раскрывается посредством решения соответствующих совокупностей задач. Приводятся примеры современных наиболее актуальных направлений в области информационной бионики, а также библиография по этим направлениям.

Ключевые слова: бионика, информационные технологии, бионические информационные технологии, биологический прототип, информационная бионика, искусственный интеллект, инженерия знаний, когнитивное моделирование, экспертные системы, мультиагентные системы, нейробионика, нейронные сети, нейрокомпьютеры, эволюционное моделирование, генетические алгоритмы, генетическое программирование, гомеостатика, биологическая иммунная система, искусственные иммунные системы

Введение

Человек как наивысшее достижение в творчестве природы — неотъемлемая часть этой же самой природы. Вся жизнедеятельность человека, в том числе его интеллект и творчество, так или иначе связаны с природой. Для удовлетворения своих потребностей человечество использует знания о живой природе, по-видимому, с тех пор, как сформировались сознание и абстрактное мышление, т. е. возникли возможности осуществлять интеллектуальную деятельность по аналогии. Поэтому естественно, что творческий ум человека в своей деятельности многое заимствует из природы. Однако такое заимствование не копирует созданное природой, а на основе обобщения, абстрагирования, осознания и осмысления в процессе интеллектуальной деятельности порождает новые идеи, которые творчески реализует, создавая новое, не существующее в природе. Естественная природа и интеллект человека в своей

взаимосвязи и в своем взаимодействии, таким образом, образуют диалектическое единство, выступают источником новых идей, открытий, изобретений и одновременно служат инструментом реализации этих идей. Совместное творчество природы и человека — **заимствование патентов природы** — имеет длительную историю. Наиболее известные и научно обоснованные работы в этом направлении исторически, по всей видимости, принадлежат Леонардо да Винчи. Однако научное обоснование, оформление в виде самостоятельной науки, получившей название **бионика** и целенаправленное развитие, такой симбиоз получил только во второй половине XX века.

Особая роль в бионике принадлежит моделированию, благодаря которому осуществляется научно обоснованный переход от биологического прототипа к техническому аналогу. Однако следует отметить, что в смысле прямой аналогии технические решения далеко не всегда соответствуют своим биологическим прототипам. Традиционная биони-

ческая технология предполагает наличие определенной, необходимой и достаточной для построения адекватной модели и соответствующего этой модели технического аналога совокупности эмпирических, теоретических и экспериментальных знаний о биологическом объекте-оригинале. Однако зачастую такие условия трудно выполнимы. Неполнота, некорректность, неадекватность данных и знаний (информации) о биологическом прототипе приводят к потребности или необходимости расширения знаний о нем. В свою очередь, такая необходимость или потребность выступает в качестве фактора, активирующего творческое мышление специалиста. В процессе активной целенаправленной интеллектуальной деятельности возникают условия порождения таких ментальных инструментальных средств, как догадка, метафора, фантазия, гипотеза, интуиция, метод и т. п. Все названное выше в совокупности приводит к порождению и созданию новых — инновационных — теорий, методов, технологий и средств в научном, техническом творчестве, а как следствие, и в инновационном производстве.

Следует отметить и еще один важный аспект, возникающий в рамках бионического подхода, который условно назовем *метафорическим*. Идеи, гипотезы и методы, воплощаемые в моделях биологических прототипов, а также результаты соответствующих модельных экспериментов в силу неадекватности модели далеко не всегда соответствуют моделируемому оригиналу. Однако как сама модель, так и полученные с ее помощью результаты, могут иметь существенную новизну для технических приложений и быть весьма полезными применительно к решению технико-технологических задач и проблем. Таким образом, бионические информационные технологии могут порождать не только прямые технические аналоги биологических прототипов, но и спектр косвенных аналогов — метафорических моделей, а следовательно, и более широкий спектр полезных результатов.

В отличие от многих других наук бионика имеет официальную дату и место своего рождения. В мае 1960 года на XII ежегодной конференции по авионавигации и радиоэлектронике, проходившей в США, американский ученый Дэвид Стил в своем докладе впервые высказал мнение, что новая наука должна иметь свое имя и предложил термин *бионика*. Официально наименование *бионики* как науки, использующей в качестве прототипов или аналогов живые системы в целях создания новых технологий и технических систем, было закреплено на первом американском национальном симпозиуме по бионике, проходившем с 13 по 15 сентября 1960 года под девизом *"Живые прототипы — ключ к новым технологиям"*. Таким образом, бионика как наука имеет всего полувековую историю.

Прогресс современного информационного общества, и тем более будущего общества знаний, в существенной мере определяется уровнем порождения

и внедрения в жизнь инновационных технологий. Особое место в области инновационного социального и экономического прогресса по праву занимают информационные технологии, ресурсы и системы, определяющие уровень развития информационного общества, а также его высшей формы — *общества знаний* [1, 2]. Весьма актуальным и перспективным направлением в социально-экономическом, инновационном развитии как общества, так и государства на основе продуцирования новых знаний являются информационные технологии, аналогичные существующим в природе, т. е. *бионические информационные технологии (БИТ)*.

Настоящая работа посвящена краткому обзору наиболее актуальных современных информационных технологий, в основу создания которых положен бионический подход, т. е. информационные механизмы и процессы, реализуемые в высокоорганизованных живых системах. Изучением информационных процессов в живых организмах занимается *биоинформатика*, а разработкой технических приложений на основе биологических прототипов — *бионика*. Таким образом, информационные технологии, создаваемые по аналогии с биологическими прототипами, определяются как *бионические*.

Бионика — это комплексное научное направление, а также методология и технология создания инновационных технических систем на основе использования в качестве прототипов биологических систем.

Для удовлетворения своих потребностей человечество использует знания о живой природе с тех пор, как сформировались сознание и абстрактное мышление, т. е. возникли возможности осуществлять интеллектуальную деятельность по аналогии. Цели и задачи информационной бионики как самостоятельного научного направления ориентированы на создание не только *бионических информационных технологий*, но и средств реализации этих технологий. Предметом исследований в бионике выступают объекты, процессы и явления живой природы на различных уровнях ее организации — от молекулярных и субклеточных структур до целостных организмов, популяций и сообществ [3, 4]. В качестве теоретического базиса бионика использует широкий спектр наук о живом (по областям применений), естественных наук (в первую очередь математику и физику), а также адекватные решаемым задачам инженерно-технические дисциплины.

Развитие бионики в нашей стране связано с именами многих выдающихся ученых, в числе которых такие, как академики АН СССР А. И. Берг, П. К. Анохин, Б. Н. Петров, Г. С. Поспелов. Существенный вклад в становление и развитие бионики внесли профессор В. М. Ахутин и Д. С. Чернавский, возглавлявшие на протяжении ряда десятилетий соответственно секции "Бионика" и "Биокибернетика", входившие в Научный Совет по проблеме "Кибернетика" при Президиуме АН СССР, а позже РАН.

В 1992 году в Санкт-Петербурге была проведена Международная научная конференция по проблемам моделирования в бионике "БИОМОД-92" [5]. Под эгидой Международной академии информатизации (МАИ) в 1993 и 1994 годах проводились Международные научные конференции "Информационная бионика и моделирование" (ИНФОБИОМОД-93 и ИНФОБИОМОД-94) [6]. Определенную работу в этом направлении проводил и Международный научно-технологический центр (МНТЦ) "Проблемы информационной бионики и моделирования", созданный в 1998 году в рамках Международной академии информатизации [7].

Бионическая технология, в полном смысле этого понятия, не имеет единой, раз и навсегда определенной структуры. В одних случаях для реализации "патента природы" бывает достаточно просто идеи, в других — требуются фундаментальные исследования, натурные и модельные эксперименты; в-третьих — создаются чисто качественные — метафорические модели. Иными словами, в каждом конкретном случае имеет место своя ситуация. Несмотря на разнообразие возможных ситуаций традиционная бионическая научно обоснованная технология в общем случае предполагает реализацию последовательности определенных этапов, перечень которых приводится ниже:

- начало работ по реализации бионической технологии;
- определение предмета потребности, для создания которого используется бионическая технология;
- определение биологического объекта — прототипа предмета потребности;
- исследование и описание биологического прототипа;
- проверка полноты, целостности, необходимости и достаточности описания для создания модели;
- формализация описания биологического прототипа;
- разработка формальной (математической) модели биологического прототипа;
- определение предметной области (среды) реализации предмета потребности;
- создание формальной (математической) модели предмета потребности, аналогичной (подобной) изоморфной или изофункциональной модели биологического прототипа;
- техническая реализация объекта (предмета потребности) на основе формальной модели — аналога биологического прототипа;
- исследование технической системы — биологического аналога на предмет удовлетворения потребности по заданным критериям и показателям;
- документирование всех этапов работ по бионической технологии;
- окончание всех работ по созданию технической системы — биологического аналога.

Следует отметить, что перечисленный выше порядок этапов в зависимости от конкретной задачи мо-

жет изменяться, а реализация технической системы, как уже отмечалось, не обязательно во всем повторяет биологический прототип. В одних случаях техническая система (ТС) может быть аналогичной или подобной биологическому прототипу, в других — метафоричной.

Информационная бионика: современный этап

В настоящее время информационная бионика наиболее активно и успешно развивается по ряду направлений, имеющих не только научно-теоретическое, но и важное практическое значение. Именно некоторым из этих направлений и посвящено дальнейшее изложение.

Искусственный интеллект [8—15] — наиболее интригующее, привлекательное и достаточно широко разработанное как в теоретическом, так и в прикладном аспектах направление БИТ. Считается, что искусственный интеллект как самостоятельное научное направление "родился" и юридически оформился в 1956 году — в год проведения ставшей уже исторической Дортмундской научной конференции по моделированию на ЭВМ не вычислительных задач. Изначально развитие исследований по искусственному интеллекту основывалось, как отмечал академик Г. С. Поспелов, на метафорическом подходе. Основные теоретические и прикладные задачи решались на основе идеи функционально адекватного результата. Такой подход на начальном этапе сыграл весьма полезную роль в развитии проблемно-ориентированных математических методов, алгоритмических и программных средств, в частности таких, как:

- *нечеткая математика* и ее приложения — мягкие вычисления, нечеткий вывод, обработка нечеткой информации [8];
- *методы рассуждений*, включающие индуктивные и дедуктивные рассуждения, достоверные и правдоподобные выводы, а также ряд других [9, 10];
- *инженерия знаний* и рассуждения на знаниях, в основу которых положены результаты исследований в сфере порождения, накопления, представления и хранения знаний в памяти ЭВМ, а также обработка и использование знаний при решении задач [10, 11];
- *когнитивное моделирование* — область научных исследований и практических приложений, возникшая в конце семидесятых годов на стыке математических методов искусственного интеллекта и когнитивной психологии, т. е. исследование процессов познавательной и рассудочной деятельности человека: как он изобретает, создает новое — творит. Задача когнитивного моделирования — изучение, описание и моделирование когнитивных механизмов и процессов, создание моделей познавательных механизмов и, как следствие, автоматизация процессов накопления и формирования знаний, проблемно-ориентиро-

ванных на развитие интеллектуальных возможностей систем искусственного интеллекта;

- *экспертные системы* (ЭС) — одно из прикладных направлений искусственного интеллекта, ориентированное на решение практических проблемно-ориентированных задач на основе сформированных в базе знаний ЭС витальных знаний специалистов — экспертов по предметной области [12, 13];
- *интеллектуальные мультиагентные системы* — еще одно стремительно развивающееся современное направление искусственного интеллекта, важная составная часть стратегического комплекса "наук об искусственном". Подробное описание сущности данного направления изложено в прекрасной монографии В. Б. Тарасова [14].

В настоящее время спектр проблематики сферы искусственного интеллекта стал весьма обширен [15, 16]. В рамках ограниченности объема данной работы более подробный обзор не представляется возможным. Тем не менее, считаю необходимым отметить имена ведущих отечественных ученых, стоявших у истоков данного направления и внесших существенный вклад в его развитие. В первую очередь к ним следует отнести: академика АН СССР Г. С. Поспелова, профессоров Г. С. Осипова, Э. В. Попова, В. Н. Вагина, В. Б. Тарасова, Л. С. Болотову, Г. В. Рыбину, а также ряд других специалистов. Особо следует отметить выдающийся вклад в развитие искусственного интеллекта академика РАЕН, Президента отделения "Искусственный интеллект" МАИ профессора Д. А. Поспелова, многочисленные научные труды которого до настоящего времени не потеряли актуальности.

Нейробионика

Нейробионика — активно и успешно развивающееся научно-теоретическое и прикладное практическое направление в области БИТ. Согласно Е. Н. Соколову [17] "*Нейробионика*" — наука об организации технических систем из нейроподобных элементов. Основным этапом нейробиологических исследований является моделирование нейроподобных систем при помощи ЭВМ или на базе уже имеющихся функциональных элементов". Головной мозг и центральная нервная система человека, их нейронные структуры, в которых реализуются различные информационные процессы, давно привлекают внимание ученых и специалистов, разрабатывающих новые принципы, методы, способы, технологии и средства обработки информации. Нейробионика как раз и представляет широкий спектр как теоретических проблем, так и прикладных задач БИТ, для которых в качестве прототипа выступают нейрофизиологические элементы — нейроны, а также нейросистемы — ансамбли нейронов живого организма и в первую очередь головного мозга человека. Нейробионика включает *нейросетевые* информа-

ционные технологии и *нейрокомпьютерные* средства обработки информации, ориентированные на решение разнообразных задач [18, 19]. Формальные модели нейронов, нейронных механизмов и нейронных сетей, начало которых восходит к перцептронной модели Розенблата, реализуют принципы параллельной обработки информации, дают возможность создавать новые информационные технологии с широким спектром практических приложений. Необходимость теоретической базы нейробионики привела к созданию нейроматематики, ориентированной на решение разнообразных задач в нейросетевом базисе. В настоящее время нейробионика имеет широкий спектр практических приложений и продолжает активно развиваться. В России, а ранее в СССР, нейробионика успешно развивается на протяжении многих десятилетий благодаря усилиям известного талантливого ученого и организатора науки А. И. Галушкина — главного редактора многолетней серии научной литературы по теории и практике нейробионики, а также главного редактора научного журнала "Нейрокомпьютеры".

Эволюционное моделирование [20—25]

Естественная природа развивается и совершенствуется в результате процесса эволюции, в то время как антропогенная сфера — в результате интеллектуальной и материальной деятельности человека, т. е. посредством решения разнообразных задач и овеществления этих решений. Существует два несовпадающих подхода к теории эволюции: по Ж. Б. Ламарку и по Ч. Дарвину. Эволюционное моделирование основывается на традиционном учении Ч. Дарвина. В 1966 году Л. Дж. Фогель, А. Дж. Оуэнс и М. Дж. Уолш [20] опубликовали работу, в которой излагались результаты их исследований по эволюции простых автоматов, предсказывающих символы в цифровых последовательностях. В 1975 году Д. Х. Холланд предложил модель генетического алгоритма. Эти публикации и послужили толчком к широкому развитию работ по эволюционному моделированию и генетическим алгоритмам.

Замечание. Подробный анализ основных направлений и перспектив развития эволюционного моделирования в России изложен в обзоре И. Л. Букаевой [21—23].

Биологической основой теории эволюционного моделирования служат работы Ч. Дарвина по теории эволюции, а в качестве модели эволюции выступают специальные алгоритмы, названные эволюционными.

Замечание. В последние годы в научной литературе по эволюционному моделированию широкое распространение получил термин "*генетические алгоритмы*" [24].

Эволюционные алгоритмы несмотря на их самостоятельную роль относят к направлению "искусственный интеллект", определяемому как "кибер-

нетика черного ящика". Эволюционные алгоритмы принято различать по их функциональным особенностям, а также по входящим в их состав операторам. В настоящее время в рамках эволюционного моделирования выделяют следующие типы эволюционных алгоритмов: эволюционные стратегии (авторы — Рехтенберг и Швепфер); эволюционное программирование (Оуэн, 1968); генетические алгоритмы (Холланд Д., 1974); генетическое программирование (Коза И., 1984); адаптивный случайный поиск (Растрингин Л. А., 1970); классифицирующие системы (Поспелов Д. А., 1966).

Такие алгоритмы в общем случае имитируют различные аспекты развития популяции в условиях изменяющейся окружающей среды. Аналогом популяции могут выступать совокупности однородных объектов (например, варианты структур вычислительных систем, реализации решений системных задач и т. п.). Каждый объект в выделенной совокупности — это аналог индивида в популяции. Адаптация популяции к окружающей среде происходит за счет эволюции ее индивидов. Механизмы эволюции: селекция, скрещивание, мутация, отбор воспроизводятся эволюционным алгоритмом. Цикл эволюции представляется цепочкой *популяция* → *селекция* → *родители* → *скрещивание* → *потомки* → *отбор* → *замещение* → *обновленная популяция* и т. д. Примеры областей применения эволюционных алгоритмов:

- численная комбинаторная оптимизация;
- моделирование и идентификация систем;
- планирование и управление;
- инженерно-техническое проектирование;
- извлечение знаний и данных, приобретение знаний;
- автоматизация обучения;
- искусственная жизнь.

В России (СССР) эволюционное моделирование активно развивалось и продолжает развиваться благодаря трудам известных ученых: И. Л. Букатовой [21—23], С. Н. Гринченко [3, 4], В. М. Курейчика [24], Л. А. Растрингина [26], а также их учениками и последователями.

Гомеостатика

Гомеостатика [27—30] — наиболее молодое научно-теоретическое и прикладное направление БИТ. Физиология гомеостаза достаточно хорошо изучена [27]. Исторически учение о гомеостатах в его кибернетической трактовке восходит к работам незаслуженно забытого талантливого русского ученого Н. А. Белова. Им был открыт и сформулирован "*Закон параллельно-перекрестного взаимодействия*" или "*Закон замкнутых пространств*", основное содержание которого он опубликовал в 1912 году в журнале "Новое в медицине" [28]. Наряду с законом о параллельно-перекрестном взаимодействии Н. А. Белов фактически сформулировал принцип

отрицательной и положительной обратных связей, а также принцип гомеостаза, которые были изложены в докладе на Международном конгрессе в Париже осенью 1911 года [29], т. е. за 18 лет до известных работ В. Б. Кеннона. За рубежом, спустя два десятилетия, подобные исследования получили развитие в трудах К. Бернара, В. Б. Кэннона, У. Р. Эшби, а также ряда других ученых.

Понятие *гомеостатика* в кибернетическом аспекте, а одновременно и как самостоятельное научно-теоретическое и прикладное направление, родилось именно в России [30]. Основоположники и создатели научно-теоретических основ гомеостатики — известные талантливые ученые В. И. Астафьев, Ю. М. Горский, А. М. Степанов, а также их сотрудники — разработчики многочисленных приложений, определяют гомеостатику как раздел кибернетики, имеющий бионическую основу. Одна из последних совместных фундаментальных работ этих авторов — коллективная монография "Гомеостатика: гармония в игре противоречий", в которой подведены итоги двадцатилетней работы, опубликована в 2008 году [31].

В основе теории гомеостатики лежит раскрытие глубинной сущности закона единства и борьбы противоположностей, а также определение основных свойств систем, образуемых объединением противоположностей (Ю. М. Горский, 1994). Идея гомеостаза как динамического постоянства процессов за счет управления противоречием, стала фундаментальным базисом науки о гармонии в биологических, социальных, экономических, технических и других целостных системах. Согласно этой идее поддержание гомеостаза — устойчивого гармоничного функционирования целостной системы — обеспечивается через некоторый механизм, названный гомеостатом или гомеостатической системой. Устройство гомеостатических систем основывается на иерархически организованных противоречиях. Элементарное противоречие формируется триадой: противоречие между противоположностями и, как минимум, два противоречия внутри противоположностей.

Теоретическим ядром, определяющим фундаментальные основы гомеостатических систем, выступает совокупность следующих положений:

- структурным основанием гомеостаза является специфическая организация информационных потоков при управлении процессами, протекающими в объекте управления;
- наиболее простая гомеостатическая структура формируется из трех контуров управления, образующих такую целостность, в которой один контур управляет целями в двух других, ниже расположенных контурах управления, а последние имеют общий объект управления;
- цели управления, реализуемые в контурах нижнего уровня, соотносятся как противоположные,

порождающие в гомеостатических системах внутренние противоречия;

- поддержание гомеостаза динамических свойств объекта управления обеспечивается управлением противоречием как согласованием процессов в "треугольнике" управления, включающем управляющие механизмы U_1 и U_2 одного и того же управляемого процесса, а также вышестоящий механизм U_0 , управляющий первыми двумя.

Теоретическое развитие и обоснование приведенных выше положений определяют сущность БИТ при создании информационно-управляющих систем нового типа, а также при решении практических задач и проблем в различных сферах деятельности. В настоящее время и на ближайшую перспективу работы в области гомеостатики, вероятно всего, будут вестись по трем основным направлениям:

- изучение живых организмов в целях более полного раскрытия механизмов гомеостатического управления;
- построение моделей гомеостатов, исследование их свойств и условий возникновения различных патологий;
- приложение принципов и моделей гомеостатики к дальнейшему развитию различных естественных и технических наук, а также к решению практических задач и проблем.

Искусственные иммунные системы [32, 33]

Только два типа физиологических функциональных систем целостного высокоорганизованного живого организма (нервная и иммунная) обладают возможностями интеллектуальной обработки информации. Они реализуют такие функции, как запоминание и хранение информации, обучение и общение, распознавание и классификация, принятие решений в не определенных заранее, т. е. в незнакомых ситуациях, а также ряд других.

Биологическая иммунная система [32] — это совокупность средств и механизмов реализации процессов, направленных на защиту организма от чужеродного биологического материала — агентов и веществ. Биологическая иммунная система ориентирована на реализацию ряда сложных интеллектуальных функций. Особенный интерес для бионических исследований представляет способность биологических иммунных систем распознавать совершенно неизвестные для них чужеродные объекты, а также объекты, эволюционирующие, усложняющиеся и адаптирующиеся к организму хозяина. В силу ряда особых информационных свойств биологических иммунных систем, которые в прямом смысле можно рассматривать как БИТ, иммунная система живого организма и, в первую очередь человека, чрезвычайно интересна для исследований как прототип для создания новых информационных технологий на основе математического и компьютерного (имитационного) моделирования ее функций. Системы,

реализующие информационные технологии, прототипами которых выступает биологический иммунитет, принято называть **искусственными иммунными системами** (ИИС) [33]. Для такого рода систем характерны следующие основные свойства: ИИС является децентрализованной, локально-распределенной, самоорганизующейся. Она должна обладать способностью распознавать широкий спектр антигенов — чужеродных объектов и не быть слишком требовательной к вычислительным ресурсам. Благодаря подобному рода возможностям ИИС уже в настоящее время находят применение в задачах обработки и анализа информации, математического моделирования, распознавания, принятия решений, прогнозирования, определения уровней рисков, информационной безопасности. Дальнейшее развитие БИТ на основе биологических иммунных систем и сетей, безусловно, имеет самые широкие практические перспективы.

Заключение

Как уже отмечалось в начале статьи, перечисленные выше бионические информационные технологии отражают только небольшую часть тех возможностей в познавательной научно-теоретической, предметно-практической и инновационной деятельности, которые природа предоставляет человеку. В последние десятилетия бионика получила значительный импульс к новому развитию. Такая активность, в первую очередь, определяется тем, что современной науке доступны фундаментальные научные исследования и технологии на микрометровых и субмикрометровых — (нано)уровнях. Они дают возможность открывать, исследовать и заимствовать сверхминиатюрные природные механизмы и конструкции. Современная бионика позволяет разрабатывать новые инновационные материалы, технологии и устройства, копирующие природные аналоги на клеточном, субклеточном, молекулярном и внутримолекулярном уровнях. Результаты подобных бионических технологий наиболее перспективны. Таким образом, можно сделать заключение, что исследовать, осознать, осмысливать, формировать новые научные знания, реализовывать их для решения практических задач и использовать на благо человека и общества — задачи нашего научного сообщества.

Список литературы

1. **Нечаев В. В., Дарьин А. В.** Информационное моделирование процессов социальной эволюции: ресурсный подход. М.: МИРЭА (ТУ), 1999. 33 с.
2. **Нечаев В. В., Нечаева Н. В.** Когнитивно-ментальная информационная деятельность человека // Информационные и телекоммуникационные технологии. 2013. № 17. С. 61—74.
3. **Гринченко С. Н.** Системная память живого. М.: ИПИ РАН, 2004. 512 с.
4. **Гринченко С. Н.** Метаэволюция (систем неживой, живой и социально-технологической природы). М.: ИПИ РАН, 2007. 456 с.
5. **Нечаев В. В.** Интеллектуальные системы моделирования как методология и инструмент решения творческих задач. Про-

блемы моделирования в бионике // Тр. Международной конференции по проблемам моделирования в бионике — "Биомод-92". СПб.: "Информсервис". 1992. С. 359—360.

6. **Информационная** бионика и моделирование. Сб. науч. трудов (по материалам международной научной конференции "ИНФОБИОМОД-94") / Под ред. академика Л. Н. Лупичева. М.: Гос. ИФТП РАН, 1995. 184 с.

7. **Проблемы** информационной бионики и моделирования. Академический сборник научных трудов. М.: Информациология, 1998. 182 с.

8. **Пегат А.** Нечеткое моделирование и управление: Пер. с англ. / Под ред. Ю. В. Тюменцева. М. БИНОМ. Лаборатория знаний, 2009. 798 с.

9. **Вагин В. Н.** и др. Достоверный и правдоподобный вывод в интеллектуальных системах / Под ред. В. Н. Вагина и Д. А. Поспелова. Изд. 2-е, испр. и доп. М.: Физматлит, 2008. 318 с.

10. **Искусственный** интеллект. В 3-х книгах. Кн. 2. Модели и методы: Справочник / Под ред. Д. А. Поспелова. М.: Радио и связь, 1990. 304 с.

11. **Болотова Л. С.** Системы искусственного интеллекта: модели и технологии, основанные на знаниях: учебник. М.: Финансы и статистика, 2012. 664 с.

12. **Попов Э. В.** Экспертные системы: Решение неформализованных задач в диалоге с ЭВМ. М.: Наука. Гл. ред. Физ.-мат. лит., 1987. 288 с.

13. **Джарратано Дж., Райли Г.** Экспертные системы: принципы разработки и программирование. 4-е издание: Пер. с англ. М.: Вильямс, 2007. 1152 с.

14. **Тарасов В. Б.** От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. М.: Эдиториал УРСС, 2002. 352 с.

15. **Рассел С., Норинг П.** Искусственный интеллект: Современный подход. Второе издание. Москва, Санкт-Петербург, Киев: Вильямс, 2006. 1408 с.

16. **Рутковский Л.** Методы и технологии искусственного интеллекта / Пер. с польск. М.: Горячая линия—Телеком, 2010. 520 с.

17. **Соколов Е. Н., Шмелев Л. А.** Нейробионика. Организация нейророботных элементов и систем. М.: Наука, 1983. 280 с.

18. **Галушкин А. И.** Теория нейронных сетей. Кн. 1: учеб. пособие для вузов. М.: ИПРЖР, 2000. 416 с.

19. **Хайкин С.** Нейронные сети: полный курс. 2-е изд.: Пер. с англ. М.: Вильямс, 2006. 1104 с.

20. **Фогель Л., Оуэнс А., Уолш М.** Искусственный интеллект и эволюционное моделирование. М.: Мир, 1969. 230 с.

21. **Букатова И. Л.** Эволюционное моделирование и его приложения. М.: Наука, 1979. 232 с.

22. **Букатова И. Л., Михасев Ю. И., Шаров А. М.** Эвоинформатика: Теория и практика эволюционного моделирования. М.: Наука, 1991. 206 с.

23. **Букатова И. Л.** Развитие эволюционного моделирования в России: концепция, приложения, перспективы // Информационные технологии, 2013. № 9. С. 12—17.

24. **Курейчик В. М.** Генетические алгоритмы. Состояние. Проблемы. Перспективы // Известия РАН. ТиСУ. 1999. № 1. С. 144—160.

25. **Нечаев В. В., Дарьин А. В.** Информационное общество и человек. Вып. 1. Моделирование процессов эволюции: ресурсный подход. М.: Информациология, 1999. 68 с.

26. **Растринин Л. А.** Адаптация сложных систем. Рига: Зинатне, 1981. 375 с.

27. **Гомеостаз.** Под редакцией П. Д. Горизонтова. М.: Медицина, 1976. 464 с.

28. **Белов Н. А.** Закон параллельно-перекрестного взаимодействия // Новое в медицине. 1912, № 6 и № 7.

29. **Белов Н. А.** Закон параллельно-перекрестного взаимодействия // Тр. 1-го Всемирного конгресса по сравнительной патологии. Париж, 1911 (публикация 1912 г.).

30. **Горский Ю. М.** Основы гомеостатики (Гармония и дисгармония живых, природных, социальных и искусственных систем). Иркутск: Изд-во ИГЭА, 1998. 338 с.

31. **Горский Ю. М., Степанов А. М., Теслинов А. Г.** Гомеостатика: гармония в игре противоречий. Иркутск: Репроцентр А1, 2008. 634 с.

32. **Пол У., Сильверстайн А., Купер М.** и др. Иммунология. В 3-х т. М.: Мир, 1987—1988.

33. **Искусственные** иммунные системы и их применение / Под ред. Д. Дасгупты: Пер. с англ. под ред. А. А. Романюхи. М.: Физматлит, 2006. 344 с.

V. V. Nechaev, Professor, e-mail: nechaev@mirea.ru,

Moscow State Technical University of Radio Engineering, Electronics and Automation

Modern Bionics Information Technology

The article is devoted to one of the actual and prospective scientific-practical directions in information technology area — informational bionics (IB). IB is considered as a complex technology consisting of three phases: biological prototype, its model representation and model implementation in a technical analog device or in a process. The content of each phase is disclosed via a set of corresponding tasks. The article considers examples of modern and most actual directions in the informational bionics area and provides the bibliography list on these directions.

Keywords: bionics, information technology, bionics information technology, biological prototype, information bionics, artificial intelligence, knowledge engineering, cognitive modeling, expert systems, multy agents systems, neurobionics, neural networks, neurobionical computer, evolutionary modeling, genetics algorithm, genetics programming, intellectual technology, homeostatics, biological immune systems, artificial immune systems

References

1. **Nechaev V. V., Dar'in A. V.** Informacionnoe modelirovanie processov social'noj jevoljucii: resursnyi podhod. М.: MIRJeA (TU), 1999. 33 p.

2. **Nechaev V. V., Nechaeva N. V.** Kognitivno-mental'naja informacionnaya dejatel'nost' cheloveka. *Informacionnye i telekommunikacionnye tehnologii*. 2013. N. 17. P. 61—74.

3. **Grinchenko S. N.** Sistemnaja pamjat' zhivogo. М.: IPI RAN, Mir, 2004. 512 p.

4. **Grinchenko S. N.** Metajevojuciya (sistem nezhivoj, zhivoj i social'no-tehnologicheskoy prirody). М.: IPI RAN, 2007. 456 p.

5. **Nechaev V. V.** Intellektual'nye sistemy modelirovanija kak metodologija i instrument resheniya tvorcheskih zadach. *Problemy modelirovanija v bionike. Trudy Mezhdunarodnoj konferencii po problemam modelirovanija v bionike — "Biomod-92"*. SPb: Informservis, 1992. P. 359—360.

6. **Informacionnaja** bionika modelirovanie. Sbornik nauchnyh trudov (po materialam mezhdunarodnoj nauchnoj konferencii "INFOBIOMOD-94"). Pod red, akademika L. N. Lupichjeva. М.: Gos. IFTP RAN, 1995. 184 p.

7. **Problemy** informacionnoj bioniki i modelirovanija. Akademicheskij sbornik nauchnyh trudov. M.: Informaciologija, 1998. 182 p.
8. **Pegat A.** Nechjotkoe modelirovanie i upravlenie. Per s angl. Pod redakciej Ju. V. Tjumenceva. M.: BINOM. Laboratorija znaniy, 2009. 798 p.
9. **Vagin V. N.** i dr. Dostovernyj i pravdopodobnyj vyvod v intellektual'nyh sistemah. Pod red. V. N. Vagina i D. A. Pospelova. Vtoroe izd., ispr. i dop. M.: Fizmatlit, 2008. 318 p.
10. **Iskusstvennyj** intellekt. V 3-h knigah. Kn. 2. Modeli i metody: Spravochnik. Pod red. D. A. Pospelova. M.: Radio i svjaz', 1990. 304 p.
11. **Bolotova L. S.** Sistemy iskusstvennogo intellekta: modeli i tehnologii, osnovannye na znaniyah: uchebnik. M.: Finansy i statistika, 2012. 664 p.
12. **Popov Je. V.** Jekspertnye sistemy: Reshenie neformalizovannyh zadach v dialoge s JeVM. M.: Nauka: Gl. red. Fiz.-mat. lit., 1987. 288 p.
13. **Dzharratano Dz., Rajli G.** Jekspertnye sistemy: principy razrabotki i programmirovanie, 4-e izdanie: Per. s angl. M.: Vil'jams, 2007. 1152 p.
14. **Tarasov V. B.** Ot mnogoagentnyh sistem k intellektual'nym organizacijam: filosofija, psihologija, informatika. M.: Jeditorial URSS, 2002. 352 p.
15. **Rassel S., Norvig P.** Iskusstvennyj intellekt: Sovremennyj podhod. Vtoroe izdanie. Moskva, Sankt-Peterburg, Kiev, 2006. 1408 p.
16. **Rutkovskij L.** Metody i tehnologii iskusstvennogo intellekta. Per. s pol'sk. M.: Gorjachaja linija—Telekom, 2010. 520 p.
17. **Sokolov E. N., Shmel'ov L. A.** Nejobionika. Organizacija nejobodnyh jelementov i sistem. M.: Nauka, 1983. 280 p.
18. **Galushkin A. I.** Teorija neyronnyh setej. Kn. 1. Uchebn, posobie dlja Vuzov. M.: IPRZHR, 2000. 416 p.
19. **Hajkin S.** Neironnye seti: polnyj kurs, 2-e izdanie. Per s angl. M.: Vil'jams, 2006. 1104 p.
20. **Fogel' L., Oujens A., Uolsh M.** Iskusstvennyj intellekt i jevolucionnoe modelirovanie. M.: Mir, 1969. 230 p.
21. **Bukatova I. L.** Razvitie jevolucionnogo modelirovanija v Rossii: koncepcii, prilozhenija, perspektivy. *Informacionnye tehnologii*. 2013. N. 9. P. 12—17.
22. **Bukatova I. L.** Jevolucionnoe modelirovanie i ego prilozhenija. M.: Nauka, 1979. 233 p.
23. **Bukatova I. L., Mihas'ov Ju. I., Sharov A. M.** Jevoinformatika. Teorija i praktika jevolucionnogo modelirovanija. M.: Nauka, 1991. 206 p.
24. **Kurejchik V. M.** Geneticheskie algoritmy. Sostojanie. Problemy. Perspektivy. *Izvestija RAN. TiSU*. 1999. N. 1. P. 144—160.
25. **Nechaev V. V., Dar'in A. V.** Informacionnoe obshhestvo i chelovek. Vyp. 1. Modelirovanie processov jevolucii: resursnyj podhod. M.: Informaciologija, 1999. 68 p.
26. **Rastrigin L. A.** Adaptacija slozhnyh sistem. Riga: Zinatne, 1981. 375 p.
27. **Gomeostaz.** Pod redakciej P. D. Gorizontova. M.: Medicina, 1976. 464 p.
28. **Belov N. A.** Zakon parallel'no-perekrojstnogo vzaimodejstvija. *Novoe v medicine*. 1912. N. 8 i N. 7.
29. **Belov N. A.** Zakon parallel'no-perekrojstnogo vzaimodejstvija. *Trudy 1-go Vsemirnogo kongressa po sravnitel'noj patologii. Parizh, 1911 g.* (publikacija 1912 g.).
30. **Gorskij Ju. M.** Osnovy gomeostatiki. (Garmonija i disgarmonija zhivyh, prirodnyh, social'nyh i iskusstvennyh sistem). Irkutsk: Izd-vo IGJeA, 1998. 338 p.
31. **Gorskij Ju. M., Stepanov A. M., Teslinov A. G.** Gomeostatika: garmonija v igre protivorechij. Irkutsk: Reprocentr. A1. 2008. 634 p.
32. **Pol U., Sil'verstaj'n A., Kuper M.** i dr. Immunologija. V 3-h tomah. M.: Mir, 1987. 1988.
33. **Iskusstvennye** immunnnye sistemy i ih primenenie. Pod red. D. Dasgupty: Per s angl. pod red. A. A. Romaniuhi. M.: Fizmatlit, 2006. 344 p.

Поздравляем юбилера!



Известному специалисту в области вычислительной техники и информационных технологий, члену редакционной коллегии журнала «Информационные технологии»

Валерию Владимировичу ШИЛОВУ

исполнилось 60 лет.

В. В. Шилов — заведующий кафедрой «Проектирование вычислительных комплексов» Российского государственного технологического университета имени К. Э. Циолковского («МАТИ»), участник ряда проектов создания отечественных суперЭВМ, автор более 200 научных трудов и 6 монографий. Его работы не раз отмечались дипломами крупнейшего профессионального общества IEEE.

Высококвалифицированный специалист, энтузиаст, большой знаток и ценитель научно-исторической и биографической литературы, чуткий и отзывчивый товарищ, он пользуется глубоким уважением учеников и коллег.

Редколлегия и редакция журнала поздравляет с юбилеем В. В. Шилова, талантливого ученого и педагога, одного из ведущих российских историков в области информатики и вычислительной техники.

Редакционный совет, редакционная коллегия и редакция журнала желают Вам, Валерий Владимирович, крепкого здоровья, большого счастья и новых творческих успехов!

УДК 519.233

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

Итеративная кластеризация траекторий управляемых объектов в многомерном пространстве характеристик

Итеративное применение алгоритма кластеризации многомерных траекторий управляемых объектов позволяет выделить характерные паттерны движения и изолирует посторонние траектории. Первичная кластеризация траекторий выполняется последовательно в различных пространствах характеристик с помощью алгоритма сдвига среднего. На основании результатов кластеризации по всем пространствам характеристик выделяются обобщенные кластеры. Итеративная кластеризация демонстрируется на примере анализа траекторий движения самолетов в зоне аэропорта.

Ключевые слова: анализ данных, многомерные траектории, кластеризация

Введение

Траектории объектов видеонаблюдения являются важными элементами для представления и анализа движения объектов. Кластеризация является ключевым методом анализа траекторий, когда вместо моделирования или исследования движения индивидуального объекта множество траекторий обрабатывается одновременно, раскрывая структуру потоков движущихся объектов. Кластеризация позволяет разделить траектории на два основных типа: нормальные траектории, которые относятся к общим моделям движения, и посторонние траектории (outliers), которые отклоняются от общих моделей. В результате кластеризации в один кластер попадают траектории, имеющие сходные характеристики, а характеристики траекторий разных кластеров значительно различаются. В трехмерном декартовом пространстве траектории движущихся объектов описываются последовательностью трех пространственных координат, абсолютной скоростью и временем. Для классификации траекторий объектов применяется как одновременная кластеризация траекторий в многомерном пространстве [1, 2], так и кластеризация моделей траекторий в различных абстрактных пространствах характеристик с последующим объединением результатов [3].

В настоящей работе предлагается итеративный подход к кластеризации траекторий, отображенных в абстрактные пространства характеристик. Все абстрактные пространства характеристик траекторий используются одновременно для получения более высокой степени информативности, чем при исполь-

зовании единственного пространства характеристик. В каждом пространстве характеристик используется эмпирическое распределение вероятности векторов, представляющих траектории, для которого с помощью алгоритма сдвига среднего (mean shift) [4, 5] определяются моды — локальные максимумы этого распределения. Представляющие траектории вектора, которые относятся к конкретным модам, образуют кластеры. Далее результаты кластеризации в пространствах характеристик анализируются совместно для определения окончательных кластеров. Полученные на первой итерации кластеры с малым числом элементов и векторы, соответствующие траекториям, которые находятся далеко от центра кластера, считаются посторонними. Затем для множества всех посторонних траекторий процедура кластеризации повторяется до достижения стационарного результата, когда последующие итерации не меняют число посторонних траекторий. Результаты итеративной кластеризации траекторий в абстрактных пространствах характеристик демонстрируются на примере анализа данных радаров международного аэропорта г. Сан-Франциско, находящихся в свободном доступе на сайте <http://c3.nasa.gov/dashlink/resources/132>.

Представление траекторий в пространствах характеристик

Выбор модели представления траектории является основой алгоритма кластеризации [3]. Модели представления траекторий делятся на модели с обу-

чением и модели без обучения. Модели с обучением чувствительны к начальному выбору параметров, что может привести к ненадежным моделям кластеров. В отличие от моделей с обучением, модели без обучения не используют обучающую выборку траекторий и предварительное задание числа кластеров.

Данные анализируемых траекторий представляются в различных абстрактных пространствах характеристик, в каждом из которых выполняется непараметрическая кластеризация. Анализ множества траекторий с помощью алгоритма сдвига среднего [4, 5] выявляет моды и соответствующие им кластеры. Окончательные кластеры формируются в результате объединения результатов кластеризации по всем используемым пространствам характеристик. Кластеры, соответствующие похожим траекториям, рассматриваются как нормальные, в то время как отдельные, не вошедшие в кластеры, траектории считаются посторонними.

В трехмерном пространстве траектории объектов представляются в виде

$$T_j = \{(x_j[i], y_j[i], z_j[i]), i = \overline{1, N_j}\}, j = \overline{1, J},$$

где декартовы координаты $(x_j[i], y_j[i], z_j[i])$ соответствуют оцениваемому положению j -го объекта; N_j — число точек траектории (траектории объектов могут иметь различные длины).

Пусть функции $\Phi_m(\cdot)$, $m = \overline{1, M}$, переводят траекторию T_j в d_m -мерные пространства характеристик $\{\Psi_m^{d_m}, m = \overline{1, M}\}$, т. е.

$$\Phi_m(\cdot): T_j \rightarrow \Psi_m^{d_m}, m = \overline{1, M},$$

где M — число пространств характеристик.

Пространства характеристик $\{\Psi_m^{d_m}, m = \overline{1, M}\}$ считаются независимыми, во-первых, чтобы избежать нормирования, необходимого при совместном анализе характеристик, и, во-вторых, чтобы упростить совместный анализ неортогональных пространств характеристик с различными областями определения (пространственными и угловыми). Кроме того, этот подход допускает параллельную кластеризацию с использованием различных характеристик и дальнейшую интеграцию результатов, полученную в различных пространствах характеристик.

Выбор характеристик траектории определяется решаемой задачей, и каждое пространство характеристик способствует выделению кластеров из множества анализируемых траекторий. В качестве характеристик траектории рассматриваются пространственные и угловые представления траектории j -го объекта:

- средняя скорость объекта (average velocity) $\langle \mathbf{v}_j \rangle$;

- вектор перемещения объекта (directional distance) \mathbf{d}_j ;
- среднее траектории объекта (mean) \mathbf{m}_j ;
- модель формы траектории объекта в виде полиномиальной регрессии (shape);
- представление траектории объекта с помощью анализа главных компонент (Principal Component Analysis — PCA) [6, 7];
- гистограмма изгибов траектории объекта (directional histogram) [8].

Средняя скорость $\langle \mathbf{v}_j \rangle$ описывает изменение положения j -го объекта

$$\langle \mathbf{v}_j \rangle = \frac{1}{N_j - 1} \sum_{i=1}^{N_j-1} (x_j[i+1] - x_j[i], y_j[i+1] - y_j[i], z_j[i+1] - z_j[i]). \quad (1)$$

Средняя скорость помогает разделению траекторий объектов, движущихся с переменным шагом. Вектор перемещения \mathbf{d}_j j -го объекта представляет длину проекций траекторий на оси координат и описывает направление движения:

$$\mathbf{d}_j = (x_j[N_j] - x_j[1], y_j[N_j] - y_j[1], z_j[N_j] - z_j[1]). \quad (2)$$

Эта характеристика помогает разделить длинные и короткие траектории, а также противоположно направленные траектории. Пространственные координаты среднего положения \mathbf{m}_j j -й траектории

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{i=1}^{N_j} (x_j[i], y_j[i], z_j[i]). \quad (3)$$

Эта характеристика хорошо разделяет траектории, относящиеся к различным областям в трехмерном пространстве. Для моделирования формы траектории T_j независимо от ее длины и набора точек используется полиномиальная регрессия

$$\begin{cases} \mathbf{x}_j^* = (1 \mathbf{z}_j (\mathbf{z}_j)^2 \dots (\mathbf{z}_j)^q) \times (\boldsymbol{\beta}_0^{(1)} \boldsymbol{\beta}_1^{(1)} \dots \boldsymbol{\beta}_q^{(1)})^T + \boldsymbol{\varepsilon}^{(1)}; \\ \mathbf{y}_j^* = (1 \mathbf{x}_j (\mathbf{x}_j)^2 \dots (\mathbf{x}_j)^q) \times (\boldsymbol{\beta}_0^{(2)} \boldsymbol{\beta}_1^{(2)} \dots \boldsymbol{\beta}_q^{(2)})^T + \boldsymbol{\varepsilon}^{(2)}; \\ \mathbf{z}_j^* = (1 \mathbf{y}_j (\mathbf{y}_j)^2 \dots (\mathbf{y}_j)^q) \times (\boldsymbol{\beta}_0^{(3)} \boldsymbol{\beta}_1^{(3)} \dots \boldsymbol{\beta}_q^{(3)})^T + \boldsymbol{\varepsilon}^{(3)}, \end{cases} \quad (4)$$

где первые множители в правой части уравнений — стандартные матрицы Вандермонда размерности $N_j \times (q + 1)$ с компонентами $\mathbf{z}_j = \{z_j[i], i = \overline{1, N_j}\}$, $\mathbf{x}_j = \{x_j[i], i = \overline{1, N_j}\}$ и $\mathbf{y}_j = \{y_j[i], i = \overline{1, N_j}\}$ соответственно. Второй множитель в (4) — вектор размерности $(q + 1) \times 1$, и последний компонент — это вектор размерности $N_j \times 1$. Наилучшая модель определяется нахождением оптимальных значений коэффициентов $(\boldsymbol{\beta}_0^{(k)}, \boldsymbol{\beta}_1^{(k)}, \boldsymbol{\beta}_2^{(k)})$, $k = 1, 2, 3$, в (4), для которых $\boldsymbol{\varepsilon}^{(1)} = |\mathbf{x}^* - \mathbf{x}|$, $\boldsymbol{\varepsilon}^{(2)} = |\mathbf{y}^* - \mathbf{y}|$, $\boldsymbol{\varepsilon}^{(3)} = |\mathbf{z}^* - \mathbf{z}|$ минимальны. Этот поиск требует некоторого ком-

промисса между точностью и эффективностью. С увеличением степени полинома точность приближения растет, но только до некоторой величины. В настоящей работе фиксируется $q = 2$, так как рост значения q не влияет на общую точность. Коэффициенты $(\beta_0^{(k)}, \beta_1^{(k)}, \beta_2^{(k)})$, $k = 1, 2, 3$, определяют начальное положение, скорость и ускорение объекта.

Анализ главных компонент (РСА) используется для уменьшения размерности набора данных перед кластеризацией с минимальными потерями информативности [6, 7]. РСА-анализ хорошо работает в случае данных с единым гауссовым распределением. Этот метод требует точной оценки ковариационной матрицы шума на основе исходных данных. Пусть $\mathbf{r}_j[i] = (x_j[i], y_j[i], z_j[i])^T$, тогда траектория T_j может быть представлена в виде набора этих векторов $T_j = \{\mathbf{r}_j[i], i = \overline{1, N_j}\}$. Затем из каждой точки траектории T_j вычитаются координаты среднего положения \mathbf{m}_j этой траектории

$$\tilde{T}_j = \{\mathbf{r}_j[i] - \mathbf{m}_j, i = \overline{1, N_j}\}. \quad (5)$$

Матрица ковариации имеет вид

$$\Xi_j = \frac{1}{N_j} \tilde{T}_j \tilde{T}_j^T. \quad (6)$$

Разложение собственных чисел Ξ_j (6) дает собственные значения $\alpha = \{\alpha_i, i = \overline{1, N_j}\}$ и соответствующие собственные векторы $\Phi = \{\Phi_i, i = \overline{1, N_j}\}$. Упорядочив $\alpha_i, i = \overline{1, N_j}$, по убыванию, рассматривают первые два собственных вектора $\Phi_k, \Phi_l \in \Phi$, соответствующие двум наибольшим собственным значениям $\alpha_k, \alpha_l \in \alpha$, поскольку наибольшее изменение приходится как раз на эти две собственные компоненты [6, 7].

Для оценки резкости поворотов траектории T_j вычисляется гистограмма направленности (trajectory directional histogram) в каждой координатной плоскости. Эта характеристика описывает статистически направленное распределение траекторий. Гистограмма направленности вычисляется по методу, предложенному в работе [8]:

$$\mathbf{h}_j = H(\theta_j[i]), \quad (7)$$

где $H(\theta_j[i])$ — гистограмма углов направлений $\theta_j[i] = \arctg((y_j[i+1] - y_j[i]) / (x_j[i+1] - x_j[i]))$.

Аналогичные гистограммы вводятся для плоскостей (x, z) и (z, y) . Однако гистограмма направленности сама по себе недостаточна, так как не содержит пространственную информацию. Поэтому две траектории, которые разнесены в пространстве, попадут в один кластер, если их гистограммы на-

правленности подобны. Индексы трех самых больших значений \mathbf{h}_j соответствуют доминирующим углам в траектории.

Таким образом, данные анализируемых траекторий преобразуются в векторы пространства характеристик. Затем в этих пространствах выполняется поиск кластеров. Поскольку нет никаких предварительных знаний о типе анализируемых траекторий, все их характеристики считаются равнозначными и при кластеризации имеют равные веса. В следующем разделе описывается процедура кластеризации.

Кластеризация

В настоящей работе пространства характеристик считаются независимыми, что упрощает совместный анализ неортогональных пространств характеристик с различными областями определения. Каждому пространству характеристик ставится в соответствие эмпирическое распределение вероятности векторов, представляющих траектории в этом пространстве [4]. Метод сдвига среднего поднимается по градиенту распределения вероятности, достигая ближайший пик распределения, т. е. доминантную моду [4, 5, 9]. После этого каждый вектор, представляющий траекторию, приписывается к ближайшей моде, т. е. к соответствующему кластеру.

Пусть $\mathbf{x}_j \in \Psi_m^{d_m}, j = \overline{1, J}$, является набором векторов, представляющих траектории $T_j, j = \overline{1, J}$, в m -м пространстве характеристик. Эмпирическое распределение вероятности векторов имеет вид

$$p(\mathbf{x}, h_m^{d_m}) = \frac{1}{J h_m^{d_m}} \sum_{j=1}^J K\left(\frac{\mathbf{x} - \mathbf{x}_j}{h_m^{d_m}}\right), \quad (8)$$

где $h_m^{d_m}$ — ширина полосы ядра $K(\cdot)$. Правильный

выбор ширины полосы $h_m^{d_m}$ важен при кластеризации методом сдвига среднего. Для этого используется итеративная процедура. Первоначальная ширина полосы составляет 10 % от каждой размерности m -го пространства характеристик и итеративно возрастает до 80 %. Нижняя граница отсекает кластеры из одного вектора (для единственной траектории), а верхняя граница исключает кластер, объ-

единяющий векторы $\mathbf{x}_j \in \Psi_m^{d_m}, j = \overline{1, J}$, для всех траекторий. Малая величина $h_m^{d_m}$ снижает отклонение оценки распределения (8), но увеличивает дисперсию. В работе [10] показано, что оптимальным ядром в формуле (8), обеспечивающим минимум средней интегральной квадратичной ошибки

(mean integrated square error), является радиально-симметричное ядро Епанечникова

$$K(\mathbf{x}) = \begin{cases} \frac{1}{2V_m^{d_m}} (d_m + 2)(1 - \|\mathbf{x}\|^2), & \|\mathbf{x}\| < 1, \\ 0, & \|\mathbf{x}\| \geq 1, \end{cases} \quad (9)$$

где $V_m^{d_m}$ представляет объем d_m -мерной единичной сферы. Градиент эмпирического распределения (8) имеет вид

$$\nabla p(\mathbf{x}) = \frac{d+2}{h_m^{d_m} V_m^{d_m}} \left(\frac{1}{J_c} \sum_{\mathbf{x}_j \in S(\mathbf{x})} (\mathbf{x} - \mathbf{x}_j) \right), \quad (10)$$

где $S(\mathbf{x})$ — гиперсфера радиуса $h_m^{d_m}$ с объемом $h_m^{d_m} V_m^{d_m}$ и центром в точке \mathbf{x} , содержащая J_c векторов [5]. В результате вектор сдвига среднего

$$\mathbf{M}_h(\mathbf{x}) = \frac{1}{J_c} \sum_{\mathbf{x}_j \in S(\mathbf{x})} (\mathbf{x} - \mathbf{x}_j)$$

имеет вид

$$\mathbf{M}_h(\mathbf{x}) = \frac{h_m^{d_m} V_m^{d_m}}{(d_m + 2)} \frac{\nabla p(\mathbf{x})}{p(\mathbf{x})}. \quad (11)$$

Результатом выполнения процедуры сдвига среднего является множество векторов, связанных с конкретной модой [9]. После выбора начального вектора, представляющего некоторую траекторию, процедура сдвига среднего сходится к локальной моде, и все векторы, ограниченные шириной полосы $h_m^{d_m}$ ядра $K(\cdot)$ в формуле (8), приписываются к этой моде. Малая ширина полосы способствует росту числа мод и большой дисперсии, которая означает неустойчивость оценки локального распределения (8). Этот недостаток устраняется путем объединения близко расположенных мод [11]. В настоящей работе смежные кластеры объединяются, если соответствующие им моды распределения находятся на расстоянии, меньшем $h_m^{d_m} + 0,1\%(h_m^{d_m})$.

Окончательная кластеризация траекторий движения

После объединения смежных кластеров в каждом пространстве характеристик окончательная кластеризация исходных траекторий обеспечивается совместным анализом кластеров во всех пространствах характеристик. Эта кластеризация проходит в три этапа: сначала оценивается среднее число кластеров, затем устанавливается соответствие между кластерами в различных пространствах

характеристик и в результате каждая траектория приписывается к конкретному кластеру.

Пусть множество $\{N_m, m = \overline{1, M}\}$ содержит числа кластеров для каждого из пространства характеристик $\{\Psi_m^{d_m}, m = \overline{1, M}\}$. Окончательное число кластеров $N = \frac{1}{M} \sum_{m=1}^M N_m$. После определения числа кластеров N оптимизируется их структура. Оптимизация кластеризации начинается с выбора пространства характеристик $\Psi_l^{d_l} \in \{\Psi_m^{d_m}, m = \overline{1, M}\}$, для которого $N_l = N$. Исходными кластерами считаются кластеры $\{\mathbb{C}_i^l, i = \overline{1, N}\} \subset \Psi_l^{d_l}$. Затем ищется соответствие этих кластеров кластерам во всех остальных пространствах $\Psi_n^{d_n} \in \{\Psi_m^{d_m}, m = \overline{1, M}\}, n \neq l$. Если кластер $\mathbb{C}_j^n \subset \Psi_n^{d_n}$ имеет максимальное число перекрывающихся элементов с кластером $\mathbb{C}_i^l \subset \Psi_l^{d_l}$, т. е.

$$j = \arg \max_k |\mathbb{C}_i^l \cap \mathbb{C}_k^n|, \quad (12)$$

то кластер \mathbb{C}_i^l корректируется согласно $\mathbb{C}_i^l = \mathbb{C}_i^l \cap \mathbb{C}_j^n$. Такая корректировка выполняется для всех кластеров $\{\mathbb{C}_i^l, i = \overline{1, N}\} \subset \Psi_l^{d_l}$, и в результате формируется N согласованных кластеров траекторий $\{\mathbb{C}_i^{final}, i = \overline{1, N}\}$. После этого к кластерам $\{\mathbb{C}_i^{final}, i = \overline{1, N}\}$ присоединяется множество траекторий $\{\Gamma' \subseteq \{\Gamma_j, j = \overline{1, J}\}\}$, которые не вошли в кластеры. Каждый кластер моделируется собственным распределением Гаусса с шириной полосы, определяемой дисперсией кластера. Условная вероятность того, что траектория $T_0 \in \{\Gamma'\}$ генерируется в соответствии с окончательной моделью кластеров, имеет вид

$$p(T_0 | \mathbb{C}_k^{final}) = \frac{1}{\sqrt{2\pi}\sigma_k^{final}} \exp\left(-\left(\frac{\mathbf{m}_{T_0} - \boldsymbol{\mu}_k^{final}}{\sigma_k^{final}}\right)^2\right), \quad (13)$$

где \mathbf{m}_{T_0} — среднее траектории T_0 ; $\boldsymbol{\mu}_k^{final} = |\mathbb{C}_k^{final}|^{-1} \times \sum_{j=1}^{|\mathbb{C}_k^{final}|} \mathbf{m}_j$ (со средними траекторий (3)) — среднее

для кластера \mathbb{C}_k^{final} ; $\sigma_k^{final} = |\mathbb{C}_k^{final}|^{-1} \sum_{j=1}^{|\mathbb{C}_k^{final}|} \Xi_j$ (с мат-

рицами ковариации траекторий (6)) — стандартная дисперсия для кластера C_k^{final} [5]. Траектория T_0 приписывается к кластеру C_k^{final} , если вероятность

$$p(T_0|C_k^{final}) > p(T_0|C_l^{final}), l = \overline{1, N}, l \neq k. \quad (14)$$

Описанный подход к кластеризации траекторий, использующий выделение кластеров в пространствах характеристик, позволяет группировать даже траектории, удаленные в декартовом трехмерном пространстве, и выявляет кластеры траекторий, представляющих паттерны движения.

Выявление посторонних траекторий

Существует два типа посторонних траекторий. Посторонние траектории первого типа располагаются в областях с высокой плотностью траекторий, но демонстрируют поведение, отличное от общего паттерна движения. Если траектория $T_j \in C_k^{final}$ со средним m_{T_j} , лежит далеко от центра μ_k^{final} кластера, к которому она принадлежит, она считается посторонней при условии

$$\frac{m_{T_j} - \mu_k^{final}}{\sigma_k^{final}} > \tau, \quad (15)$$

где $\tau = 0,95$.

Второй тип посторонних траекторий располагается в районах с низкой плотностью траекторий. Если кластер содержит небольшое число траекторий и не может быть объединен с ближайшим кластером, то считается, что он состоит из посторонних траекторий. Пороговое значение для выделения кластера посторонних траекторий составляет 10 % от среднего числа элементов по всем кластерам.

Численный эксперимент

Описываемый в настоящей работе подход к кластеризации кривых тестируется на данных радар TRACON (Terminal Radar Approach Control), регистрирующего траектории полетов воздушных судов над заливом Сан-Франциско (данные находятся в открытом доступе на сайте <http://c3.nasa.gov/dash-link/resources/132/>). Подробное описание этих данных приводится в работах [9, 12]. В настоящей работе анализируются траектории 143 самолетов, совершивших посадку в течение одного дня наблюдения. Анализируются только траектории, содержащие от 150 до 200 точек наблюдения, регистрируемых через равные интервалы времени. Начало координат совпадает с положением радара.

На рис. 1 (см. вторую сторону обложки) показан исходный набор траекторий самолетов, обрабаты-

ваемых с помощью итеративного подхода к кластеризации траекторий.

После кластеризации траекторий (рис. 1, см. вторую сторону обложки) на первом этапе выделяются два кластера (показаны зеленым и розовым цветом на рис. 2, см. вторую сторону обложки) и посторонние траектории (показаны черным цветом на рис. 2). Как видно на рис. 2, в каждый из двух основных кластеров траектории объединяются не только по мере близости, но и по сходству формы. В работе [11] кластеризация аналогичных данных траекторий самолетов объединяет в кластеры пространственно близкие траектории. В настоящей работе объединение траекторий похожей формы в общие кластеры оказывается возможным, поскольку в двух из рассматриваемых пространств характеристик моделируется форма траектории (это является очевидным преимуществом применяемого подхода).

Траектории объектов, демонстрирующие движение, не характерное для первых двух кластеров на рис. 2 (см. вторую сторону обложки), ранее были отнесены к посторонним. Однако следующая итерация, состоящая в кластеризации посторонних траекторий на рис. 2, определяет два дополнительных кластера (рис. 3, см. вторую сторону обложки). Оставшиеся траектории (показаны черным цветом на рис. 3), которые не классифицированы при этой итерации, являются настоящими посторонними траекториями. На рис. 3 видны посадочные паттерны. Таким образом, применение итеративной процедуры кластеризации в пространствах характеристик позволяет выявить посадочные паттерны воздушных судов, соответствующие реальным посадочным полосам.

Заключение

Применение итеративного подхода к кластеризации траекторий управляемых объектов на основе представления траекторий в многомерных пространствах характеристик обеспечивает более полную кластеризацию этих траекторий, сокращает число посторонних траекторий и выявляет основные паттерны движения объектов.

Список литературы

1. Кухаренко Б. Г., Солнцева М. О. Кластеризация управляемых объектов на основе сходства их многомерных траекторий // Информационные технологии. 2014. № 5. С. 3—7.
2. Gaffney S., Smyth P. Joint probabilistic curve clustering and alignment / Saul L., Weiss Y., Bottou L., eds. // Proceedings of Neural Information Processing Systems (NIPS 2004). December 13—18, 2004, Vancouver, British Columbia, Canada. Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press. 2005. V 17. P. 473—480.
3. Anjum N., Cavallaro A. Trajectory clustering for scene context learning and outlier detection / Schonfeld D., Shan C., Tao D., Wang L., eds. // Video Search and Mining. Berlin. Studies in Computational Intelligence. Heidelberg: Springer. 2010. V. 287. P. 33—51.

4. **Fukunaga K., Hostetler L. D.** The estimation of the gradient of a density function, with applications in pattern recognition // *Transactions on Information Theory*. 1975. V. 21, N. 1. P. 32–40.
5. **Comaniciu D., Meer P.** Mean shift: A robust approach toward feature space analysis // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2002. V. 24, N. 5. P. 603–619.
6. **Jolliffe I. T.** *Principal Component Analysis*, 2nd ed. Springer Series in Statistics. New York: Springer. 2002.
7. **Yang K., Shahabi C.** A PCA-based similarity measure for multivariate time series // *Proceedings of the 2nd ACM International Workshop on Multimedia Databases*. New York: ACM Press. 2004. P. 65–74.
8. **Li X., Hu W.** A coarse-to-fine strategy for vehicle motion trajectory clustering // *Proceedings of 18th International Conference on Pattern Recognition (ICPR 2006)*. Washington, DC: IEEE Computer Society. 2006. V. 1. P. 591–594.
9. **Кухаренко Б. Г., Солнцева М. О.** Использование методов сокращения фона при сегментировании телеметрических изображений для идентификации групп объектов // *Информационные технологии*. 2014. N. 2. P. 3–8.
10. **Comaniciu D., Meer P.** Distribution free decomposition of multivariate data // *Pattern Analysis and Applications*. 1999. V. 2, N. 1. P. 22–30.
11. **Wang H., Suter D.** False-peaks-avoiding mean shift method for unsupervised peak-valley sliding image segmentation // *Proceedings of 7th International Conference on Digital Image Computing: Techniques and Applications (DICTA'03)*. 10–12 December 2003. Sydney, Australia: Macquarie University. 2003. P. 581–590.
12. **Gariel M., Srivastava A. N., Feron E.** Trajectory clustering and an application to airspace monitoring // *IEEE Transactions on Intelligent Transportation Systems*. 2011. V. 12. P. 1511–1524.

B. G. Kukhareno, Leader Research Scientist,
M. O. Solntseva, Graduate Student, e-mail: solntseva.chalei@gmail.com,
A. A. Blagonravova, Institute of Engineering, Russian Academy of Sciences

Iterative Clustering Trajectories of Objects under Control in Multi-Dimensional Characteristic Space

In present paper an iterative application of multi-dimensional trajectory clustering algorithm in abstract characteristic spaces is proposed for objects under control, which gives an opportunity to extract typical patterns of motion and isolate outliers. As trajectory features, spatial and angle characteristics are under consideration, such as average velocity, directional distance, trajectory mean, trajectory shape, principal components, and directional histogram. First vectors related to object trajectories are clustering sequentially in different characteristic spaces by means of mean-shift algorithm. Local maxima of each empirical density distribution are determined, which define the distribution modes related to trajectory clusters. Based on clustering results in all characteristic spaces, general trajectory clusters are extracted. An outlier set obtained at first iteration is an input to second iteration and so on. The iterations continue until the outlier number can't be changed. So, the stationary trajectory clustering result is obtained. The Iterative clustering trajectories in abstract characteristic spaces is demonstrated by analysis of airplane trajectories in an airport zone.

Keywords: data mining, multi-dimensional trajectories, clustering

References

1. **Kukhareno B. G., Solntseva M. O.** Klasterizaciya upravlyayemykh ob'ektov na osnove skhodstva ikh mnogomernykh trayektoriy. *Informacionnye tekhnologii*. 2014. N. 5. P. 3–7.
2. **Gaffney S., Smyth P.** Joint probabilistic curve clustering and alignment / Saul L., Weiss Y., Bottou L., eds. *Proceedings of Neural Information Processing Systems (NIPS 2004)*. December 13–18, 2004, Vancouver, British Columbia, Canada. *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press. 2005. V. 17. P. 473–480.
3. **Anjum N., Cavallaro A.** Trajectory clustering for scene context learning and outlier detection / Schonfeld D., Shan C., Tao D., Wang L., eds. *Video Search and Mining*. Berlin. *Studies in Computational Intelligence*. Heidelberg: Springer. 2010. V. 287. P. 33–51.
4. **Fukunaga K., Hostetler L. D.** The estimation of the gradient of a density function, with applications in pattern recognition. *Transactions on Information Theory*. 1975. V. 21, N. 1. P. 32–40.
5. **Comaniciu D., Meer P.** Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2002. V. 24, N. 5. P. 603–619.
6. **Jolliffe I. T.** *Principal Component Analysis*, 2nd ed. Springer Series in Statistics. New York: Springer. 2002.
7. **Yang K., Shahabi C.** A PCA-based similarity measure for multivariate time series. *Proceedings of the 2nd ACM International Workshop on Multimedia Databases*. New York: ACM Press. 2004. P. 65–74.
8. **Li X., Hu W.** A coarse-to-fine strategy for vehicle motion trajectory clustering. *Proceedings of 18th International Conference on Pattern Recognition (ICPR 2006)*. Washington, DC: IEEE Computer Society. 2006. V. 1. P. 591–594.
9. **Kukhareno B. G., Solntseva M. O.** Ispolzovanie metodov sokrashchenia fona pri segmentirovani teletricheskikh isobrazheniy dlya identifikatsii grupp ob'ektov. *Informacionnye tekhnologii*. 2014. N. 2. P. 3–8.
10. **Comaniciu D., Meer P.** Distribution free decomposition of multivariate data. *Pattern Analysis and Applications*. 1999. V. 2, N. 1. P. 22–30.
11. **Wang H., Suter D.** False-peaks-avoiding mean shift method for unsupervised peak-valley sliding image segmentation. *Proceedings of 7th International Conference on Digital Image Computing: Techniques and Applications (DICTA'03)*. 10–12 December 2003. Sydney, Australia: Macquarie University. 2003. P. 581–590.
12. **Gariel M., Srivastava A. N., Feron E.** Trajectory clustering and an application to airspace monitoring. *IEEE Transactions on Intelligent Transportation Systems*. 2011. V. 12. P. 1511–1524.

И. В. Соколов, преподаватель, **А. М. Агеев**, канд. техн. наук, зам. нач. отдела,
М. Ф. Волобуев, канд. техн. наук, доцент, нач. отдела,
М. А. Замыслов, канд. техн. наук, ст. науч. сотр.,
А. М. Мальцев, канд. техн. наук, доцент, ст. науч. сотр.,
С. Б. Михайленко, канд. техн. наук, ст. науч. сотр.

Военный учебно-научный центр ВВС "Военно-воздушная академия
имени проф. Н. Е. Жуковского и Ю. А. Гагарина" (г. Воронеж)
e-mail: volmf81@mail.ru

Оценка статистических характеристик сигнала на выходе резервированной системы, управляемой нечетким контроллером

Рассматривается имитационная модель для определения статистических характеристик сигнала на выходе резервированной системы, управляемой нечетким контроллером, при случайном характере входных сигналов. Приводятся результаты моделирования, подтверждающие целесообразность использования для повышения надежности систем авиационного оборудования резервированной системы, управляемой нечетким контроллером.

Ключевые слова: имитационная модель, нечеткий контроллер, надежность

Резервирование наиболее важных элементов систем (комплексов) авиационного оборудования является одним из перспективных путей повышения надежности пилотажно-навигационных комплексов летательных аппаратов. В работе [1] рассмотрено использование для управления резервированной системой (РС) методов нечеткой логики. Предложен подход к построению алгоритмов автоматизированного распознавания отказов резервированной с помощью мажоритарных элементов системы и управления ее реконфигурацией, направленной на восстановление работоспособности. Этот подход основан на представлении выходных параметров элементов резервированной системы в лингвистической форме и реализации стратегии управления с использованием нечеткой логики. Результаты работы алгоритмов представляются в виде вектора предпочтительности альтернативных решений, формируемых с помощью процедур нечеткого вывода, реализованных в программной среде fuzzyTECH. Там же проведено сравнение надежности резервированных систем, функционирующих на основе типового алгоритма мажоритарной логики и алгоритма нечеткого управления, построенного на основе методов нечеткой логики. Показано, что применение алгоритмов нечеткого управления РС позволяет в 1,73—1,14 раза повысить вероятность безотказной работы РС по сравнению с типовым алгоритмом мажоритарного резервирования при значениях вероятностей безотказной работы элементов, равных соответственно 0,5—0,8. Однако рассмотренные в работе [1] примеры и оценки приведены применительно к детерминированным сигналам в системе. На практике же представляет интерес ситуация, более адекватно соответствующая

реальности, когда сигналы резервируемых элементов изменяются случайным образом (флуктуируют). Применительно к такой ситуации оценка характеристик сигнала на выходе РС ранее не проводилась. Поэтому задача определения характеристик сигнала на выходе резервированной системы, управляемой нечетким контроллером, при случайном характере входных сигналов является актуальной.

Данную задачу предложено решать путем имитационного моделирования функционирования РС. Структурная схема математической имитационной модели РС, управляемой нечетким контроллером, приведена на рис. 1. На рис. 1 приняты следующие обозначения: РЭ, ЭС1, ЭС2 — рабочий элемент, первый и второй элементы сравнения соответственно; U_{pe} , U_{es1} , U_{es2} — напряжения на выходе РЭ, ЭС1 и ЭС2 соответственно; $R_{вых}^*$ — сигнал с выхода блока нечеткого контроллера, управляющий коммутацией выходных сигналов элементов РС.

Ниже рассматриваются основные элементы имитационной модели, приводится описание алгоритма функционирования нечеткого контроллера,



Рис. 1. Структурная схема имитационной модели резервированной системы, управляемой нечетким контроллером

управляющего резервированной системой, его структура и реализация модели в среде нечеткого моделирования fuzzyTECH и Fuzzy Logic Tools пакета прикладных программ MATLAB.

1. Структура и алгоритм функционирования нечеткого контроллера

Нечеткий контроллер предназначен для автоматического распознавания отказов резервированной с помощью мажоритарных элементов системы и управления ее реконфигурацией с учетом нечеткости информации о случайных возмущениях, влиянии ошибок измерения выходных параметров элементов на формирование команд управления работой РС, а также неполной достоверной информации о работоспособности отдельных элементов системы управления РС. Для построения такого контроллера использовался подход, основанный на методах теории нечетких множеств и нечеткой логики [1, 2]. При этом в качестве управляющих характеристик (УХ) рассматриваются отклонения основных выходных параметров (ОВП) резервируемых элементов от номинальных значений, характерных для соответствующих режимов работы элементов, т. е. отклонения этих параметров от значений, задаваемых соответствующими порогами по фактическим значениям $P(t) - \Delta$, где Δ — допустимый предел отклонения выходного параметра P элемента (РЭ или ЭС) от математического ожидания, при котором считается, что элемент работает безотказно.

Переход от числовых значений "четких" отклонений этих характеристик к их "нечеткой" интерпретации выполняется с помощью операции фаззификации [2]. Для этого возможные отклонения УХ от нормы рассматриваются как лингвистические переменные (ЛП), представляющие собой упорядоченную совокупность [1]:

$$\langle A, T = [\langle \beta, \gamma, M = \langle \mu_i(x)/x \rangle], X \rangle, \quad (1)$$

где A — имя ЛП; T — терм-множество (ТМ) (множество значений) ЛП; β — имя нечеткой переменной; γ — базовая шкала нечеткой переменной; M — функция принадлежности (ФП) нечеткой переменной; X — базовая шкала лингвистической переменной; μ_i — значение ФП.

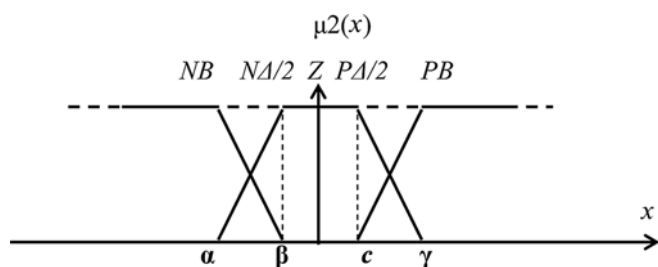


Рис. 2. Вид функции принадлежности качественных оценок отклонений значений характеристик ОВП от нормы для отдельной ЛП

Функция принадлежности в (1) задает степень соответствия текущего отклонения характеристик к каждому элементу терм-множества ЛП и представляет собой непрерывную функцию, принимающую множество значений в диапазоне от нуля до единицы ("0" — текущее значение полностью не соответствует данному терму, "1" — полностью соответствует).

Обычно ФП терма представляется в виде нечетких чисел $L-R$ -типа, задаваемых упорядоченными четверками [2]: $LR = (\beta, c, \alpha, \gamma)$ и $LR1 = (\alpha, \gamma, \beta, c)$.

Для четверки LR β, c — точки, в которых значение ФП равно единице; $\alpha(\gamma)$ — координаты ближайшей слева (справа) к $x = \beta(c)$ точки базовой шкалы, в которой ФП равна нулю. Четверка $LR1$ представляет собой перевернутое изображение LR .

В работе используется более сложная ФП, состоящая из совокупности перекрывающихся $L-R$ -чисел [2], рассмотренных выше. При этом каждая из лингвистических переменных может принимать одно из следующих значений множества:

$$L = \{NB, Z, PB\}. \quad (2)$$

В выражении (2) первая буква в обозначении указывает знак (сторону отклонения) числовой переменной (N — отрицательное, P — положительное), вторая буква говорит об абсолютном значении переменной (B (*big*) — большое), а символ Z соответствует попаданию переменной в допустимый предел Δ отклонения выходного параметра элемента от математического ожидания, при котором считается, что элемент работает безотказно.

С использованием выражения (2) определяются функции принадлежности для каждой из лингвистических переменных. Вид такой функции принадлежности качественных оценок отклонений значений ОВП от нормы показан на рис. 2.

Вычисление значений ФП терма, представленной в виде совокупности перекрывающихся $L-R$ -чисел в точке $x = \xi$ проводится по формуле

$$\mu_2(\xi) = \begin{cases} 1, & \xi \leq \alpha, \\ (\beta - \xi)/(\beta - \alpha), & \alpha \leq \xi \leq \beta, \text{ для отрезка } NB\beta \text{ ФП,} \\ (\xi - \alpha)/(\beta - \alpha), & \alpha \leq \xi \leq \beta, \text{ для отрезка } \alpha NA/2 \text{ ФП,} \\ 1, & \beta \leq \xi \leq c \\ (\gamma - \xi)/(\gamma - c), & c \leq \xi \leq \gamma, \text{ для отрезка } PN\Delta/2\gamma \text{ ФП,} \\ (\xi - c)/(\gamma - c), & c \leq \xi \leq \gamma, \text{ для отрезка } cPB \text{ ФП,} \\ 1, & \xi \geq \gamma. \end{cases} \quad (3)$$

Также вводятся качественные оценки и для принимаемых решений на выходе системы. Для этого формируется множество решений, которые должны приниматься в процессе функционирования РС. К такому множеству, соответствующему рассмотренным в работе [1] особенностям построения устройства управления резервированной с помощью мажоритарных элементов системы, относятся следующие восемь принимаемых решений: I — продол-

жение работы резервированной системы в исходном режиме (все элементы исправны, к нагрузке подключен РЭ); II — изменение конфигурации резервированной системы путем подключения к нагрузке ЭС1 при отказе РЭ; III — изменение конфигурации резервированной системы путем подключения к нагрузке РЭ при отказе ЭС1; IV — изменение конфигурации резервированной системы путем подключения к нагрузке РЭ при отказе ЭС2; V — изменение конфигурации резервированной системы при выходе из строя мажоритарного элемента, вызванного отказом РЭ и ЭС1, путем подключения к нагрузке ЭС2; VI — изменение конфигурации резервированной системы при выходе из строя мажоритарного элемента, вызванного отказом РЭ и ЭС2, путем подключения к нагрузке ЭС1; VII — изменение конфигурации резервированной системы при выходе из строя мажоритарного элемента, вызванного отказом ЭС1 и ЭС2, путем подключения к нагрузке РЭ; VIII — выход резервированной системы из строя вследствие отказа всех элементов (РЭ, ЭС1 и ЭС2).

Для того чтобы различать такие понятия в степени уверенности решений, как малая уверенность (S), достаточная уверенность (M) и сильная уверенность (B), представим для каждого из восьми принимаемых решений перечисленные понятия в виде трех термов, показанных на рис. 3.

При этом каждая из ЛП (I, II, III, IV, V, VI, VII, VIII) может принимать одно из значений множества $D = \{S, M, B\}$.

С учетом введенных выше понятий и терм-множеств под нечетким управлением будем понимать процедуры принятия решений, представленные в виде некоторой совокупности правил, основанных на логико-формальном представлении знаний о функционировании системы (в лингвистической форме) и преобразовании их с применением методов нечеткой логики. Эти правила описывают принятие решений по управлению РС и формируются экспертами в области логического резервирования систем. Правила, соответствующие принимаемым решениям I—VIII, полученные с использованием

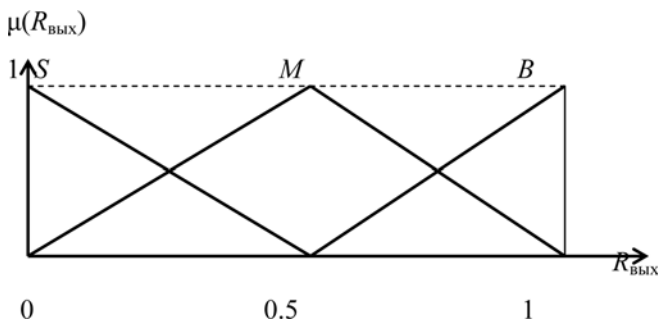


Рис. 3. Вид функций принадлежности качественных оценок уверенности в выбранных решениях по управлению резервированной системой

логико-формальных методов и введенных выше терм-множеств входных и выходных лингвистических переменных для рассматриваемой РС, в соответствии с упрощенным алгоритмом нечеткого вывода [2], записываются в следующем виде:

$$((P_{RE} = Z) \wedge (P_{ES1} = Z) \wedge (P_{ES2} = Z)) \mapsto \mapsto ((I = B) \wedge (II = S) \wedge (III = S) \wedge (IV = S) \wedge (V = S) \wedge (VI = S) \wedge (VII = S) \wedge (VIII = S)); (4)$$

$$((P_{RE} = PB \vee NB) \wedge (P_{ES1} = Z) \wedge (P_{ES2} = Z)) \mapsto \mapsto ((I = S) \wedge (II = B) \wedge (III = S) \wedge (IV = S) \wedge (V = S) \wedge (VI = M) \wedge (VII = S) \wedge (VIII = S)); (5)$$

$$((P_{RE} = Z) \wedge (P_{ES1} = PB \vee NB) \wedge (P_{ES2} = Z)) \mapsto \mapsto ((I = S) \wedge (II = S) \wedge (III = B) \wedge (IV = S) \wedge (V = S) \wedge (VI = S) \wedge (VII = M) \wedge (VIII = S)); (6)$$

$$((P_{RE} = Z) \wedge (P_{ES1} = Z) \wedge (P_{ES2} = PB \vee NB)) \mapsto \mapsto ((I = S) \wedge (II = S) \wedge (III = S) \wedge (IV = B) \wedge (V = S) \wedge (VI = S) \wedge (VII = M) \wedge (VIII = S)); (7)$$

$$((P_{RE} = PB \vee NB) \wedge (P_{ES1} = PB \vee NB) \wedge (P_{ES2} = Z)) \mapsto \mapsto ((I = S) \wedge (II = M) \wedge (III = M) \wedge (IV = S) \wedge (V = B) \wedge (VI = S) \wedge (VII = S) \wedge (VIII = S)); (8)$$

$$((P_{RE} = PB \vee NB) \wedge (P_{ES1} = Z) \wedge (P_{ES2} = PB \vee NB)) \mapsto \mapsto ((I = S) \wedge (II = M) \wedge (III = S) \wedge (IV = M) \wedge (V = S) \wedge (VI = B) \wedge (VII = S) \wedge (VIII = S)); (9)$$

$$((P_{RE} = Z) \wedge (P_{ES1} = PB \vee NB) \wedge (P_{ES2} = PB \vee NB)) \mapsto \mapsto ((I = S) \wedge (II = S) \wedge (III = M) \wedge (IV = M) \wedge (V = S) \wedge (VI = S) \wedge (VII = B) \wedge (VIII = S)); (10)$$

$$((P_{RE} = PB \vee NB) \wedge (P_{ES1} = PB \vee NB) \wedge (P_{ES2} = PB \vee NB)) \mapsto \mapsto ((I = S) \wedge (II = S) \wedge (III = S) \wedge (IV = S) \wedge (V = S) \wedge (VI = M) \wedge (VII = S) \wedge (VIII = B)); (11)$$

где \wedge — знак логической операции "и"; \mapsto — знак логической операции "секвенция", обозначающий, что высказывание, стоящее в правой части выражений (4)—(11) "истинно" тогда и только тогда, когда "истинно" высказывание в их левой части.

Для нечетких подмножеств в (4)—(11) степень выполнения условия "Если" определяет степень применения данного правила [1]. Поэтому степень уверенности в выборе того или иного решения определяется весом, соответствующим весу функции принадлежности пересечения нечетких подмножеств NB, Z, PB в условии "Если". При этом функция принадлежности пересечения нечетких подмножеств A и B определяется следующим образом [3]:

$$A, B \in E, \forall x \in E: \mu_{A \cap B} = \min(\mu_A(x), \mu_B(x)). (12)$$

Допустим, что функции принадлежности приняли следующие значения:

$$\mu_Z(P_{RE}) = A; \mu_Z(P_{ES1}) = B; \mu_{PB}(P_{ES2}) = C; C < A < B,$$

тогда в соответствии с (11) имеем

$$\beta_S(I) = \beta_S(II) = \beta_S(III) = \beta_B(IV) = \beta_S(V) = \\ = \beta_S(VI) = \beta_M(VII) = \beta_S(VIII) = \mu_Z \cap Z \cap PZB(x) = \\ = \min(\mu_Z(P_{RE}), \mu_Z(P_{ES1}), \mu_{PB}(P_{ES2})) = C,$$

где $\mu_A(x)$ — функция принадлежности лингвистической переменной x нечеткому подмножеству A ; $\beta_D(y)$ — весовой коэффициент, характеризующий степень принадлежности лингвистической переменной y (I, II, ..., VI, VII, VIII) нечеткому множеству $D = (S, M, B)$.

Приведенный выше алгоритм позволяет получить для каждого возможного решения восемь усеченных термов. Окончательное решение задачи находится путем их объединения и дефаззификации, т. е. переходом от нечеткого множества, полученного в результате объединения, к единственному четкому значению степени (часто трактуемой как вероятность) выполнения определенного решения на выходе системы. Для дефаззификации применяется метод определения взвешенного среднего значения [2], позволяющий вычислять выходное значение в реальном масштабе времени. Выходное значение в матричном виде записывается следующим образом [1]:

$$R_{\text{ВЫХ}} = \beta \cdot F^T \Delta \beta^{-1}, \quad (13)$$

где $\Delta \beta = \sum_{i=1}^n \beta_i F^*$, $F^* = \{\mu_S(R_{\text{ВЫХ}}), \mu_M(R_{\text{ВЫХ}}), \mu_B(R_{\text{ВЫХ}})\}$,

$$\beta = \begin{bmatrix} \beta_S(I) & \beta_M(I) & \beta_B(I) \\ \beta_S(II) & \beta_M(II) & \beta_B(II) \\ \beta_S(III) & \beta_M(III) & \beta_B(III) \\ \beta_S(IV) & \beta_M(IV) & \beta_B(IV) \\ \beta_S(V) & \beta_M(V) & \beta_B(V) \end{bmatrix},$$

F^* — вектор центральных значений нечетких подмножеств выходной лингвистической переменной (0, 0,5, 1); β_i — веса, учитывающие уровень выполнения условия "Если" i -го правила (уровни активности соответствующих правил); n — число правил вывода.

Полученные значения $R_{\text{ВЫХ}}$ представляют собой коэффициенты уверенности для каждого решения, с учетом которых строится ранжированный ряд состоятельных решений, имеющий, например, вид

$$R_{\text{ВЫХ}}^* = (R_{\text{ВЫХ}}^{\text{II}}, R_{\text{ВЫХ}}^{\text{VII}}, R_{\text{ВЫХ}}^{\text{IV}}). \quad (14)$$

Структурная схема нечеткого контроллера, формирующего в соответствии с рассмотренным выше алгоритмом вектор упорядоченных по предпочтительности решений по управлению РС, изображена на рис. 4. Подробное описание нечеткой модели управления резервированной системой в среде fuzzyTECH, разработанной с использованием нотации языка нечеткого управления FTL (Fuzzy Tech Language), приведено в работе [1], а описание имитационной модели, реализованной в среде Fuzzy Logic Tools приводится ниже.

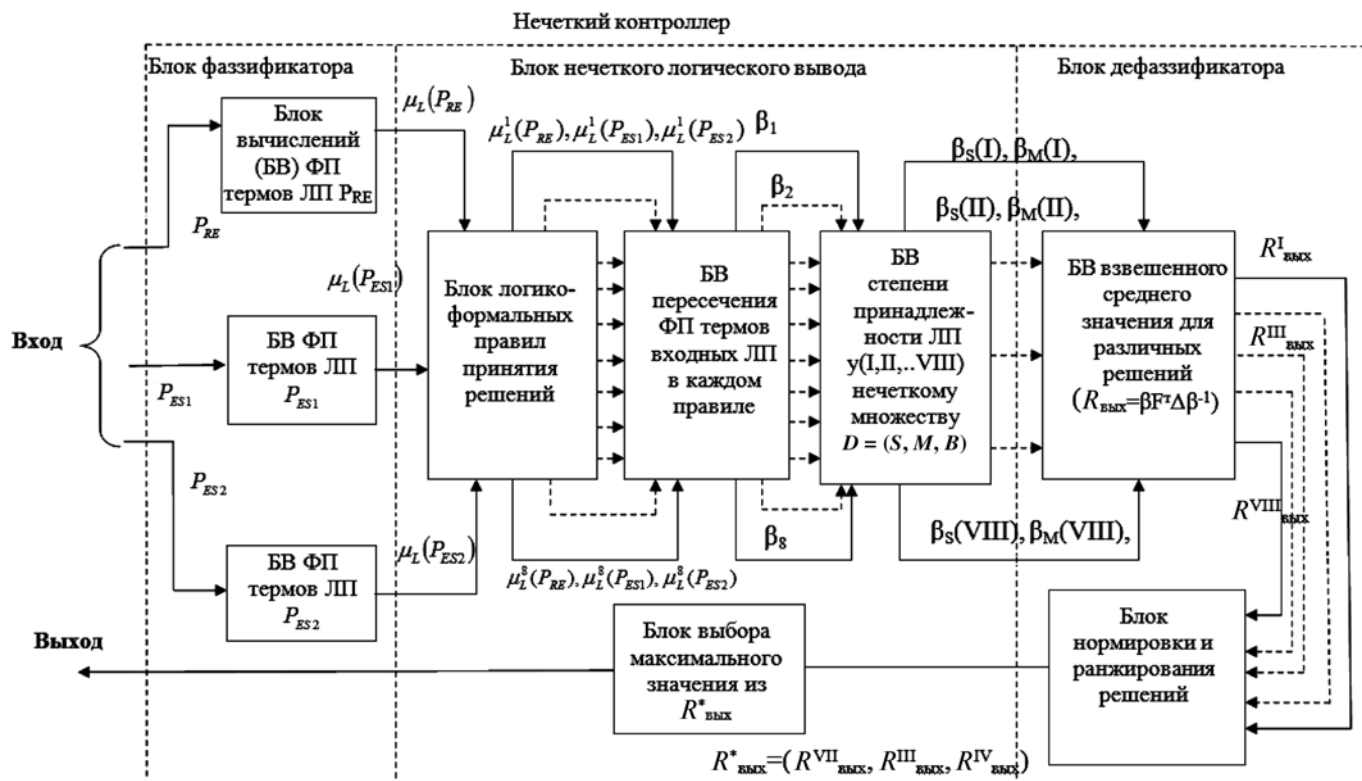


Рис. 4. Структурная схема нечеткого контроллера для управления резервированной системой

2. Реализация имитационной модели резервированной системы, управляемой нечетким контроллером, в среде нечеткого моделирования Fuzzy Logic Tools пакета прикладных программ MATLAB

Для получения более достоверных оценок имитационная модель разрабатывалась в двух вариантах — с использованием пакетов прикладных программ fuzzyTECH 6.00 Professional Demo [2] и Fuzzy Logic Tools из состава пакета прикладных программ MATLAB [4]. В первом случае с использованием программной среды fuzzyTECH разработаны блоки нечеткого контроллера и коммутатора элементов, а блоки формирования случайных сигналов элементов и вычисления среднего значения (математического ожидания) и среднего квадратического отклонения (СКО) сигналов создавали с помощью программы MathCad [5]. Подробное описание указанных элементов в среде fuzzyTECH приведено в работе [1]. При моделировании для нечеткого контроллера были приняты исходные данные, соответствующие иллюстративному примеру работы [1].

Во втором случае все элементы имитационной модели резервированной системы создавали с использованием только пакетов расширения *Simulink*, а также Fuzzy Logic Toolbox, входящих в состав программы MATLAB. Пример интерфейса имитационной модели, разработанной в среде Fuzzy Logic Toolbox программы MATLAB представлен на рис. 5 (см. третью сторону обложки).

Для создания имитационной модели использовали типовые модули среды Fuzzy Logic Toolbox и программы MATLAB, которые кратко рассмотрены ниже.

Сигналы с заданным средним квадратическим отклонением и математическим ожиданием формируются источниками случайных сигналов с нормальным распределением (*Random Number* № 1—3) и передаются на нечеткий контроллер (*Fuzzy Logic Controller*) и на выход резервированной системы через управляемый многопортовый переключатель сигналов (*Multiport Switch*). Нечеткий контроллер, реализующий описанный выше алгоритм, формирует вектор ранжированных решений. Из этого вектора с использованием встроенной функции MATLAB (*Embedded MATLAB Function*) выбирается решение с максимальным рангом. Далее, в соответствии с выбранным решением, определяется, сигнал какого канала пройдет на выход резервированной системы.

Сравнительная оценка статистических характеристик выходного и входных сигналов проводится с использованием блоков (*Running Var*) — расчет дисперсии сигнала, математической функции *Sqrt* — для определения среднего квадратического отклонения сигналов и виртуального генератора (*Scope*).

3. Результаты имитационного моделирования резервированной системы, управляемой нечетким контроллером

При моделировании с использованием программной среды fuzzyTECH рассматривались случаи, когда сигналы элементов РС имеют одинаковые средние значения математического ожидания (МО) и различные СКО, а также различные МО и СКО. В первом случае считалось, что значения основных выходных параметров элементов резервированной системы в исходном состоянии имеют следующие значения: $MO_{RE} = MO_{ES1} = MO_{ES2} = 1000$ мВ, $CKO_{RE} = 500$ мВ, $CKO_{ES1} = 600$ мВ, $CKO_{ES2} = 550$ мВ и гипотетическая резервированная система настроена на эти значения. При этом сигналы с указанными параметрами формировались в среде MathCAD в виде нормальных случайных процессов, задаваемых встроенными функциями *rnorm* (m , МО, СКО) векторами m случайных чисел с нормальным распределением.

Во втором случае формировались случайные процессы с различными математическими ожиданиями $MO_{RE} = 1000$ мВ, $MO_{ES1} = 1200$ мВ, $MO_{ES2} = 1600$ мВ и СКО, равными $CKO_{RE} = 400$ мВ, $CKO_{ES1} = 500$ мВ, $CKO_{ES2} = 400$ мВ. Причем для имитации постепенных отклонений рабочего элемента и второго элемента сравнения их сигналы формировались постепенно нарастающим и убывающим по времени соответственно. Формализовано, в терминах среды MathCad, это записывалось следующим образом:

$$\begin{aligned}j &= 1..30, U_{re}(j) = rnorm(1, 1000, 400) + 30j, \\U_{es1}(j) &= rnorm(1, 1200, 500), \\U_{es2}(j) &= rnorm(1, 1600, 400) - 25j.\end{aligned}\quad (15)$$

Сформированные указанным выше образом сигналы подаются на соответствующие входы БНК и БКЭ (см. рис. 1). С выхода БКЭ, являющегося выходом резервированной системы, сигнал поступает в блок вычисления среднего значения и СКО сигналов на входе и выходе БКЭ РС. В этом же блоке проводится сравнение статистических характеристик выходного сигнала РС с характеристиками входных сигналов БКЭ.

При моделировании с использованием программной среды Fuzzy Logic Toolbox параметры входных процессов задавались такими же, как и в среде fuzzyTECH. В процессе имитационного моделирования не было зафиксировано ни одного отказа РС, обусловленного отказом всех трех элементов РС. Это хорошо согласуется с выводами работы [1]. В результате имитационного моделирования функционирования резервированной системы, управляемой нечетким контроллером, в программных средах fuzzyTECH и Fuzzy Logic Toolbox получены похожие результаты, отличающиеся не более чем на 5...10 %. Поэтому ввиду ограниченного объема

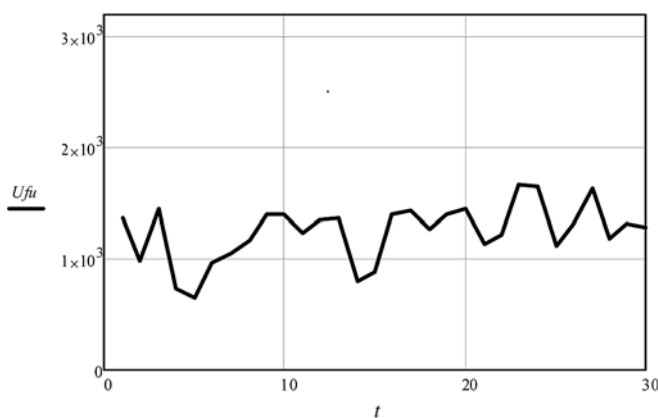


Рис. 7. График сигнала на выходе РС (U_{fi}), управляемой нечетким контроллером, в зависимости от времени t для случая, когда параметры сигналов элементов РС имеют различные МО и СКО

статьи, ниже приведены только результаты моделирования в среде FuzzyTech.

Результаты имитационного моделирования применительно к случаю, когда параметры сигналов элементов РС имеют различные МО и СКО, представлены на рис. 6, 7. На рис. 6 (см. третью сторону обложки) изображены графики выходных сигналов рабочего элемента (U_{re}) и первого и второго элементов сравнения (U_{es1} , U_{es2}) соответственно.

На рис. 7 представлен график сигнала на выходе РС, формируемого на выходе коммутатора элементов с помощью нечеткого процессора. С использованием полученных выше зависимостей были вычислены МО и СКО процесса на выходе коммутатора элементов РС. Их значения составили 1243 мВ и 253 мВ соответственно. При этом величины МО и СКО процессов, формируемых в соответствии с выражением (1) и описывающих сигналы РЭ и ЭС1, ЭС2, принимали по результатам вычислений следующие значения:

$$\begin{aligned} \text{МО}_{RE} &= 1467 \text{ мВ}, \text{МО}_{ES1} = 1090 \text{ мВ}, \\ \text{МО}_{ES2} &= 1101 \text{ мВ}, \end{aligned}$$

$$\begin{aligned} \text{СКО}_{RE} &= 569 \text{ мВ}, \text{СКО}_{ES1} = 502 \text{ мВ}, \\ \text{СКО}_{ES2} &= 485 \text{ мВ}. \end{aligned}$$

Анализ полученных результатов показывает, что при управлении резервированной системой с использованием нечеткого процессора СКО выходного сигнала снижается в 2,25, 1,98 и 1,92 раза по сравнению с СКО сигнала РЭ, ЭС1 и ЭС2 соответственно. При этом математическое ожидание выходного сигнала занимает положение, близкое к медианному значению среди сигналов элементов РС.

Результаты имитационного моделирования применительно к случаю, когда параметры сигналов элементов РС имеют одинаковые МО, но различные СКО, представлены на рис. 8 (см. третью сторону обложки), 9.

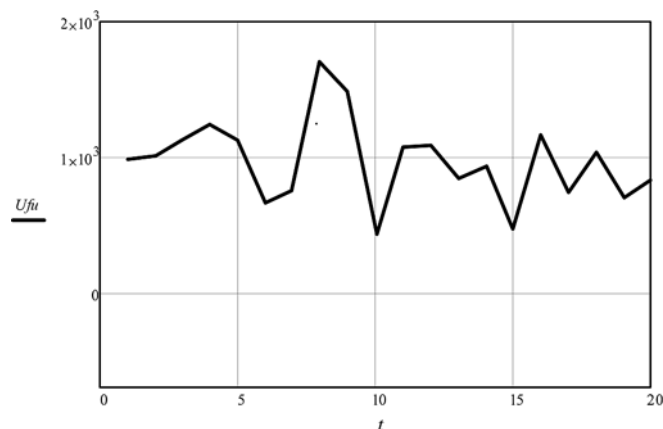


Рис. 9. График сигнала на выходе РС (U_{fi}), управляемой нечетким контроллером, в зависимости от времени t для случая, когда параметры сигналов элементов РС имеют одинаковые МО и различные СКО

На рис. 8 (см. третью сторону обложки) изображены выходные сигналы рабочего элемента U_{re} и первого и второго элементов сравнения U_{es1} , U_{es2} соответственно, а на рис. 9 — выходной сигнал РС (U_{f1}), управляемой нечетким контроллером. Результаты расчетов, проведенных в условиях, когда сигналы элементов имеют одинаковое МО, показали, что выигрыш в снижении СКО при использовании нечеткого управления составляет 1,6; 2 и 1,96 раза по сравнению с СКО сигналов РЭ, ЭС1 и ЭС2 соответственно.

* * *

Таким образом, имитационное моделирование процесса функционирования РС, управляемой нечетким контроллером, подтвердило целесообразность практического применения технологии интеллектуального управления на основе методов нечеткой логики при решении конкретной практической задачи — повышения надежности систем и комплексов авиационного оборудования. Дальнейшие усилия необходимо сосредоточить на технической реализации нечеткой системы управления при резервировании конкретных устройств (систем) авиационного оборудования.

Список литературы

1. Волобуев М. Ф., Демчук В. А., Замыслов М. А., Мальцев А. М., Михайленко С. Б. Управление резервированной с помощью мажоритарных элементов системой с применением методов нечеткой логики // Информационные технологии. 2012. № 12(196). С. 53—60.
2. Леоненков А. В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. СПб.: БХВ-Петербург, 2005. 736 с.
3. Кофман А. Введение в теорию нечетких множеств. М.: Радио и связь. 1982.
4. Дьяконов В. П. MATLAB 7*/R2006/R2007: Самоучитель. М.: ДМК Пресс, 2008.
5. Дьяконов В. П., Абраменкова И. В. MathCAD 7.0 в математике, физике и в Internet. М.: Нолидж, 1998. 352 с.

I. V. Sokolov, Teacher, A. M. Agueyev, Deputy Chief of Research Department, M. F. Volobuev, Chief of Research Department, M. A. Zamyslov, Senior Researcher, A. M. Maltsev, Senior Researcher, S. B. Mikhaylenko, Senior Researcher, Military Educational Scientific Center of the Air Force, "Air Force Academy named after prof. N. E. Zhukovsky, and Yu. Gagarin", Voronezh, e-mail: volmf81@mail.ru

Assessment of the Statistical Characteristics of the Signal at the Output of the Reserve System, Controlled by Fuzzy Controller

The article describes a simulation model to determine the statistical characteristics of the signal at the output of the redundant system controlled by fuzzy controller.

Questions of construction of a simulation model to determine the statistical characteristics of the signal at the output of the redundant system and the random nature of the input signals.

A description of the main elements of the simulation model, formal description of the algorithm functioning fuzzy controller, redundant control system, its structure and implementation of the model in an environment of fuzzy modeling fuzzyTECH and Fuzzy Logic Tools software package MATLAB.

To obtain more reliable estimates of the simulation model was developed in two versions — with the use of software packages FuzzyTech 6.00 Professional Demo and Fuzzy Logic Tools from the software package MATLAB. In the first case, using the software environment developed FuzzyTech blocks fuzzy controller and the switch elements, and blocks the formation of elements of random signals and calculating the mean value (expectation) and standard deviation (SD) signals generated using the program MathCAD. In the second case, all elements of a redundant system simulation model created using only the expansion packs, Simulink, and the Fuzzy Logic Toolbox, included in the program MATLAB.

Conducted simulation functioning redundant system controlled fuzzy controller, for the case when the signal parameters redundant system elements have different expectations and standard deviations.

Analysis of the results shows that for a redundant control system with a processor using fuzzy standard deviation of the output signal decreases to 2.25, 1.98 and 1.92 fold compared to the SD signal of the working element, the first and second comparison element, respectively. The mathematical expectation of the output signal of the system controlled by fuzzy controller, occupies a position close to the median value of the signal elements of a redundant system.

Also conducted simulations, for the case when the signal parameters redundant system elements have the same expectations, but different standard deviations.

The results of calculations carried out in conditions where the elements have the same signals expectations have shown that the gain reduction in standard deviations for applying the fuzzy control is 1.6, 1.96 and 2 times compared to the SD signal of the working element, the first and second comparison elements respectively.

The simulation of the functioning redundant system driven fuzzy controller confirmed the expediency of practical application of intelligent control technology on the basis of fuzzy logic methods in solving specific practical problems — improving the reliability of systems and avionics systems. Further efforts need to focus on the technical implementation of the fuzzy control system to reserve specific devices (systems) aircraft equipment.

Keywords: simulation model, fuzzy controller, the reliability

References

1. Volobuev M. F., Demchuk V. A., Zamyslov M. A., Maltsev A. M., Mikhaylenko S. B. Upravleniye rezervirovannoy s pomoshch'yu mazhoritarnyyh elementov sistemoy s primeneniye me-todov nechetkoy logiki. Informacionnye tehnologii. 2012. N 12(196).

2. Leonenkov A. V. Nechetkoye modelirovaniye v srede MATLAB i fuzzyTECH. SPb.: BHV-Peterburg, 2005. 736 p.

3. Kofman A. Vvedeniye v teoriyu nechetkikh mnozhestv. M.: Radio i svyaz', 1982.

4. D'yakonov V. P. MATLAB 7*/R2006/R2007: Samouchite l'. M.: DMK Press, 2008.

5. D'yakonov V. P., Abramenkova I. V. MathCAD 7.0 v matematike, fizike i v internet. M.: Nolidzh, 1998. 352 p.

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ MODELING AND OPTIMIZATION

УДК 004.056

Я. Н. Имамвердиев, канд. техн. наук, зав. отделом, e-mail: yadigar@lan.ab.az,
Институт информационных технологий Национальной академии наук Азербайджана, г. Баку

Модель ситуационного управления информационной безопасностью электронного правительства

Для надежного обеспечения информационной безопасности электронного правительства необходимо непрерывно и автоматически идентифицировать сложные динамические ситуации и реагировать на них в реальном времени. В результате требуются эффективные методы распознавания, прогнозирования, размышления и управления ситуациями — операции, в совокупности идентифицируемые как ситуационное управление. Разработана концептуальная модель ситуационного управления информационной безопасностью электронного правительства и предлагается ее реализация на основе теории прецедентов. Предложены подходы для представления знаний о ситуации информационной безопасности и выбора прецедентов, а также оптимизации весов признаков прецедентов на основе метода роя частиц. Предложенный подход можно обобщить для тактического и стратегического уровней управления.

Ключевые слова: информационная безопасность, электронное правительство, ситуационное управление, теория прецедентов, метод роя частиц

Введение

В результате широкого внедрения информационных и коммуникационных технологий (ИКТ) в процессы государственного управления основной акцент реформ государственного управления постепенно перемещается с совершенствования работы административных структур к более широкому привлечению граждан к формированию и реализации государственной политики, построению эффективной системы взаимодействия и сотрудничества между государством, частным сектором и гражданским обществом. Эти процессы приводят к фундаментальным преобразованиям в природе государства, и это явление обозначается термином *электронное правительство* [1].

В условиях всесторонней глобализации, возрастающих рисков и неопределенностей общественных процессов информационная безопасность становится одной из основных функций самосохранения электронного правительства [2]. Поэтому актуальным является управление системой обеспечения информационной безопасности электронного правительства.

Управление информационной безопасностью электронного правительства является слабоструктурированной задачей [3, 4]: объект управления является сложной социотехнической системой, состоящей из автономных компонентов, каждый из которых действует целенаправленно; в системе

происходят многочисленные процессы (социальные, политические, технологические), значительно взаимодействующие друг с другом. Характер этих процессов изменяется по времени, достаточная количественная информация о динамике процессов недоступна, а также присутствуют различные виды неопределенностей. Внешняя среда, окружающая электронное правительство, является потенциально "враждебной" средой. Как сами компоненты электронного правительства, так и внешняя среда являются источником многочисленных угроз, направленных на нарушение информационной безопасности электронного правительства.

Теория управления создала много эффективных механизмов быстрого и точного выбора управляющих воздействий на объект по данным о его состоянии. Один из них — метод ситуационного управления сложными объектами, основанный на идеях теории искусственного интеллекта [5, 6].

В данной работе разрабатывается концептуальная модель ситуационного управления информационной безопасностью электронного правительства, и для реализации концептуальной модели предлагается подход на основе теории прецедентов [7, 8]. Такой подход перспективен в контексте создания интеллектуальных систем поддержки принятия решений для управления информационной безопасностью электронного правительства.

О понятии "электронное правительство"

Для дальнейшего изложения важным является уточнение интерпретации понятия "электронное правительство". Отметим, что в области исследований электронного правительства выполнены многочисленные научно-практические работы, сюда можно добавить и разработку огромного числа официальных государственных и международных документов по электронному правительству [9]. Такая "критическая масса" позволяла даже около десяти лет назад ставить вопрос "превращаются ли исследования в области электронного правительства в новую самостоятельную научную дисциплину?" [10]. Но как показывает анализ, в научной литературе и в официальных документах международных организаций нет единого подхода при определении понятия "электронное правительство".

Кроме того, следует подчеркнуть неадекватность понятийно-терминологических систем в этой области в английском и русском языках. Термин "electronic government" (e-government) часто переводится как "электронное правительство". Такой перевод сужает понятие и сводит вопрос только к государственному управлению, осуществляемому органами исполнительной власти. Но термин e-government подразумевает поддержку с помощью ИКТ деятельности во всех трех ветвях власти — законодательной, исполнительной и судебной. Авторы работы [11] указывают, что более правильным является использование термина "электронное государство", при этом авторы допускают употребление двух этих терминов в качестве синонимов. Для наиболее точного понимания термина "электронное правительство" важно уяснить взаимосвязь и различия между такими понятиями, как "электронное государство", "информационное общество", "электронная администрация", "электронная организация" и т. д. [12]. Следуя устоявшейся терминологии в официальных документах, в этой статье используется термин "электронное правительство" [9].

Термин "электронное правительство" появился в середине 1990-х годов и с тех пор изучался с точки зрения различных дисциплинарных областей. Определения электронного правительства различными специалистами строятся по разным принципам [9, 13]. Некоторые определения описывают преобразования, происходящие в обществе и отдельных его структурах благодаря внедрению электронного правительства. Другие подчеркивают прикладной аспект электронного правительства, перечисляя разные применения отдельных его инструментов. Можно выделить следующие подходы, определяющие электронное правительство как [9, 12—14]:

- информационное взаимодействие органов власти с гражданами и организациями с использованием ИКТ, сетевых технологий и Интернета;

- процесс предоставления государственными органами управления услуг гражданам в электронной форме;
- использование ИКТ в государственном управлении в сочетании с организационными изменениями и новыми навыками для улучшения услуг и демократических процессов и укрепления поддержки для государственной политики;
- применение ИКТ для преобразования внутренних и внешних связей в целях оптимизации процессов исполнения функций правительства.

В работах [9, 13, 14] предпринята попытка классифицировать существующие понятия электронного правительства. Исследователи отмечают, что интерпретация термина "электронное правительство" связана с государственным политико-экономическим устройством и общественным менталитетом [9, 12], с восприятием правительства в общественном сознании [9] и во многом зависит от уровня внедрения информационных технологий в обществе и от понимания функций электронного правительства на различных этапах его становления [15].

В настоящей работе используется подход из работы [9], где понятие "электронное правительство" интерпретируется как "комплексная социально-экономическая и человеко-машинная система для реализации полномочий органов государственной власти, функций и государственных услуг для граждан и хозяйствующих субъектов с широким использованием информационно-коммуникационных технологий".

Концептуальная модель ситуационного управления ИБ электронного правительства

На всех этапах жизненной деятельности электронное правительство подвергается различным видам угроз информационной безопасности (ИБ). Интенсивность угроз возрастает экспоненциально, список возможных угроз довольно обширный, в ландшафте угроз ИБ электронного правительства происходят быстрые качественные и количественные изменения.

Следует отметить, что кибератаки становятся более сложными, целенаправленными, масштабными и часто угрожают критическим элементам национальной информационной инфраструктуры [16]. Успешное предотвращение таких кибератак не под силу отдельной организации и требует тесного сотрудничества между всеми заинтересованными сторонами, включая частный сектор и общество. Государство выполняет роль регулирующего органа при управлении ИБ, и во многих странах наблюдается централизованный подход к управлению ИБ на современном этапе развития технологии электронного правительства [17].

В предлагаемой концептуальной модели ситуационного управления ИБ информационная инфраструктура электронного правительства состоит из

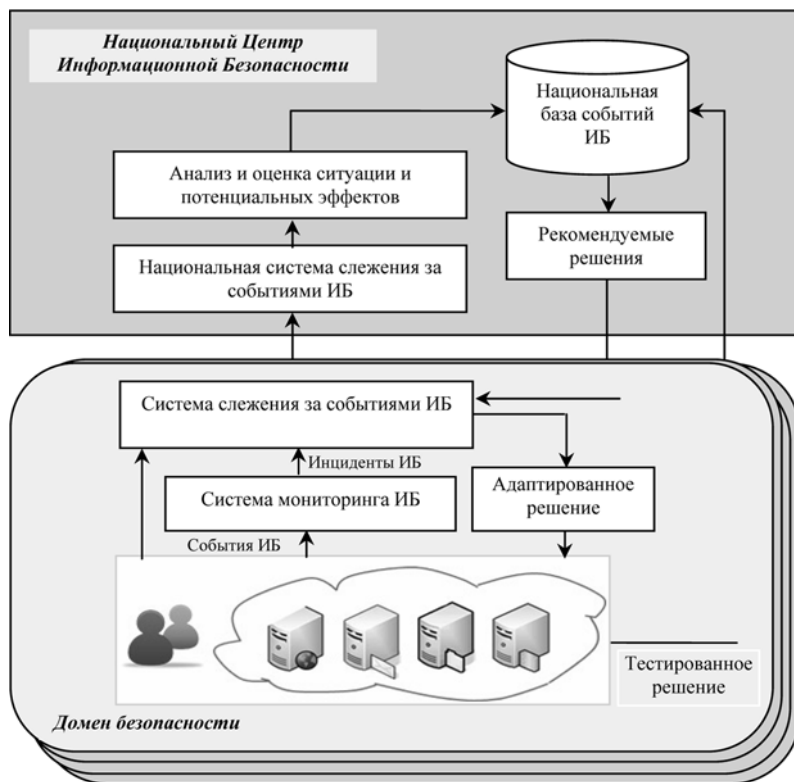


Рис. 1. Концептуальная модель ситуационного управления ИБ электронного правительства

отдельных доменов безопасности (рис. 1). В каждом домене безопасности на основе политики информационной безопасности домена осуществляется мониторинг ИБ. Под мониторингом ИБ понимается процесс постоянного наблюдения за событиями ИБ с целью своевременного выявления и реагирования на события, которые привели, либо могут привести к реализации угроз ИБ. Мониторинг ИБ включает наблюдение за сетевой и системной активностью, активностью пользователей, доступом к защищаемым ресурсам, сбоями в работе информационной системы (ИС) и мониторингом содержания потоков информации. Отметим, что согласно государственному стандарту ГОСТ Р ИСО/МЭК 27001—2006 под *событием* ИБ понимается "идентифицированное возникновение состояния системы, услуги или сети, указывающее на возможное нарушение политики информационной безопасности, отказ защитных мер, а также возникновение ранее неизвестной ситуации, которая может быть связана с безопасностью".

Система мониторинга домена безопасности является централизованной точкой сбора, обработки и анализа (консолидации, фильтрации, нормализации, агрегации, корреляции) данных о событиях ИБ, которые поступают с разных компонентов домена (межсетевые экраны, операционные системы (ОС), системы обнаружения вторжений, сетевое оборудование, антивирусные системы, приложения и т. д.) [18].

После обработки некоторые события ИБ идентифицируются как инциденты. Инцидент информационной безопасности — это событие или серия событий, указывающие на свершившуюся, предпринимаемую или вероятную реализацию угрозы ИБ [19]. Отметим, что некоторые события не могут обнаруживаться автоматически (например, события, связанные с нарушением должностных регламентов администратора). Также пользователи сами могут сообщить о событиях ИБ. После того, как событие было обнаружено, в соответствии с регламентом событие регистрируется в реестре (журнале) системы слежения за событиями, определяется категория и уровень приоритета события. Управление событиями (регистрация, анализ и обработка) осуществляется специальной группой (командой) реагирования (Computer Emergency Response Team, CERT) [19].

В существующих методиках управления событиями ИБ нет центрального органа управления, команды CERT, принадлежащие одному домену, при необходимости обмениваются информацией о событии с командами CERT из других доменов. Некоторым исключением является деятельность команды US-CERT (United States

Computer Emergency Readiness Team), которая осуществляет непрерывный мониторинг домена gov, используя систему EINSTEIN. EINSTEIN построен на основе системы обнаружения вторжений (Intrusion Detection System, IDS) на основе сигнатур и анализирует сетевые шлюзы государственных агентств и министерств [20].

В концептуальной модели, рассматриваемой в работе, данные о событиях ИБ регистрируются в локальном реестре и передаются в национальную систему слежения за событиями ИБ Национального Центра ИБ (НЦИБ). НЦИБ собирает данные о событиях со всех доменов безопасности, проводит их обработку, анализ (агрегация, корреляция) и визуализацию. Используя эти данные, НЦИБ строит целостную картину ситуации ИБ, анализирует ситуацию и оценивает потенциальные эффекты события ИБ не только на уровне отдельного домена, но и в масштабе нескольких взаимосвязанных и взаимозависимых доменов безопасности. НЦИБ осуществляет выбор подходящего способа обработки события из Национальной базы событий ИБ, и передает рекомендуемые решения соответствующим доменам безопасности. Национальная база событий ИБ является центральной базой знаний о событиях и способах их обработки, здесь хранятся также шаблоны скоординированных мер по обработке событий ИБ. В доменах безопасности рекомендуемое НЦИБ решение адаптируется к текущему событию, осуществляется обработка события (например,

вносятся необходимые изменения в конфигурации аппаратного и программного обеспечения). Для этой цели выделяются необходимые технические ресурсы, назначается группа обработки события, которая состоит из технических и других специалистов (экспертов), лица, принимающего решения (ЛПР). В ходе обработки события группа обработки события может внести необходимые изменения в адаптированное решение. После обработки события тестированный способ обработки события вносится в Национальную базу событий ИБ с указанием группы экспертов и ЛПР, которые обработали событие.

Предложенная архитектура дает возможность государственным органам узнать о событиях, происходящих в других частях информационной инфраструктуры электронного правительства, в частности, о событиях, которые непосредственно влияют на их домены. Домены безопасности имеют доступ к безопасному веб-порталу, где они могут познакомиться с информацией о событиях на своих системах.

Предложенная модель имеет несколько преимуществ по сравнению с существующими подходами. Корреляция данных о событиях и обмен информацией между несколькими доменами безопасности позволяет создать широкое представление о состоянии ИБ в информационной инфраструктуре электронного правительства у специалистов по безопасности и ЛПР. Также предложенная модель значительно сокращает (от нескольких дней до нескольких часов) время, потраченное государственными органами на сбор и обмен информацией о критических событиях ИБ. Это позволит государству укрепить потенциал анализа и предотвращения событий ИБ и улучшить реагирование на события коренным образом.

Основные элементы теории прецедентов

Ситуационное управление предусматривает принятие управленческих решений по регулированию проблемных (нештатных) и неproblemных (штатных) ситуаций, возникающих в сложной системе. В отличие от проблемных ситуаций штатные ситуации характеризуются тем, что механизмы их урегулирования априори известны.

Ситуационный подход в управленческой деятельности был предложен Р. Моклером в 1971 г. [21]. Несмотря на развитие в течение нескольких десятилетий методов ситуационного управления они не нашли должного применения на практике. Это связано с тем, что, с одной стороны, они разрабатывались на принципах классической научной рациональности, а с другой стороны, существует точка зрения, что управление ситуациями является скорее искусством, чем наукой [22, 23].

С развитием теории ситуационного управления практика показала, что при возникновении новой проблемной ситуации целесообразно использовать метод рассуждений на основе прецедентов (Case Based Reasoning, CBR). CBR возник из исследований

в когнитивной науке и, прежде всего, в исследованиях по динамической памяти Роджера Шенка и его учеников в Йельском университете [24]. Было установлено, что люди обычно решают проблемы, помня, как они решали подобные проблемы в прошлом. Методология CBR является подходом, позволяющим решить новую задачу, используя или адаптируя решение уже известной задачи. Понимание универсальности и широких возможностей этого подхода привело к созданию и быстрому развитию "теории прецедентов".

Сущность методологии CBR заключается в следующем. Прецедент является парой <ситуация, метод решения>. Возникающие с течением времени ситуации и методы их решения сохраняются в специальной базе — базе прецедентов. Когда возникает новая ситуация в базе прецедентов находят похожую ситуацию, и метод ее решения адаптируется к рассматриваемой ситуации.

Методология CBR осуществляется в виде циклической процедуры, состоящей из четырех процессов [8]:

- 1) поиск похожего прецедента для конкретного прецедента (Retrieve);
- 2) применение метода решения прецедента к конкретной ситуации (Reuse);
- 3) проверка полученного решения и коррекция (адаптация) решения при необходимости (Revise);
- 4) сохранение полученного решения в базе прецедентов для последующего использования (Retain).

Часто эти процессы называют 4Re-процессами. На рис. 2 показана схема цикла CBR.

Методология CBR применялась в самых разнообразных предметных областях для классификации, диагностики, прогнозирования, планирования и проектирования и показала очень хорошие результаты [7, 8, 25]. Также CBR применялась в различных аспектах ИБ — оценке рисков [26], обнаружении вторжений [27], анализе состояния сетевой безопасности [28].

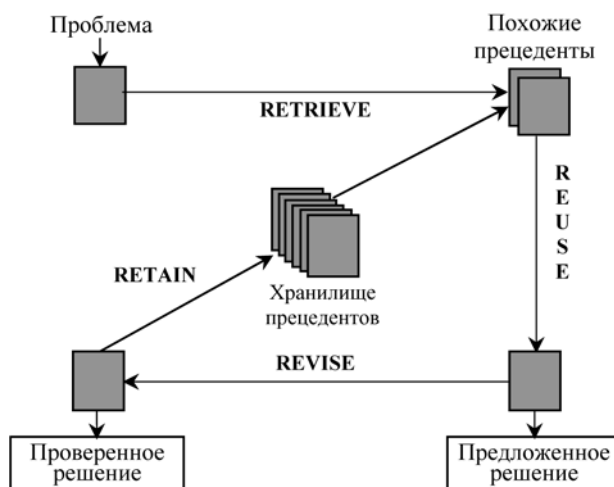


Рис. 2. Диаграмма CBR-цикла (Watson 95)

Методология CBR особенно важна для обеспечения принятия решений в экстремальных ситуациях, которые характеризуются острым дефицитом времени и быстро меняющейся обстановкой. Задачи, решаемые в экстремальных ситуациях, могут оказаться уже ранее решавшимися, и имеющиеся решения могут быть использованы для оперативного получения вариантов решений [29].

Построение CBR-системы предполагает решение следующего круга задач [7]:

- разработка способа представления знаний о ситуации и возможных решениях;
- разработка метода выбора прецедентов;
- разработка метода адаптации решений;
- разработка метода хранения и индексации прецедентов.

Можно выделить две основные проблемы, с которыми сталкиваются при построении подобных систем: поиск наиболее подходящих прецедентов и последующая адаптация найденного решения. Еще одна проблема применения CBR связана с представлением прецедентов и назначения подходящих весов их атрибутам, что уменьшает применимость таких систем в разных предметных областях.

Представление знаний о событии ИБ

Согласно [8, 25], прецедент включает:

- описание ситуации или проблемы;
- описание решения, принятого в данной ситуации или для данной проблемы;
- результат применения решения.

Учитывая это, формальную модель ситуации информационной безопасности электронного правительства можно задавать следующим образом.

Пусть информационная инфраструктура электронного правительства состоит из m доменов безопасности. Состояние ИБ электронного правительства в момент времени $t_i \in T$ описывается множеством зарегистрированных и не устраненных до этого момента времени событий ИБ в доменах безопасности $X = \{x_{i1}, \dots, x_{im}\}$, где x_{ij} — множество событий ИБ в домене безопасности j в момент времени t_i . Разумеется, это множество может быть пустым в некоторых моментах времени.

Отметим, что событие может представлять собой последовательность других событий. Назовем такое событие сложным. Кроме того, при регистрации и предварительном анализе событию присваиваются некоторые другие параметры (категория (тип), источник, возможное влияние, уровень критичности, приоритет и т. д.).

В данной работе для описания события ИБ предлагается следующая структура:

$$x = (ID, D, DT, e_1, e_2, \dots, e_k, CAT, Or, Im, Sev, Pr),$$

где

ID — уникальный идентификатор события;

D — идентификатор домена безопасности, где обнаружено событие;

DT — время и дата регистрации события;

e_1, e_2, \dots, e_k — последовательность событий ИБ, составляющих данное сложное событие (в частности, может быть $k = 1$);

CAT — категория (тип) события. Существуют различные методики определения категорий событий. Например, в методике ISF (Information Security Forum) [30] определяются следующие категории событий ИБ: внешняя атака, внутреннее неправильное использование и злоупотребление, кража, неисправность системы, прерывание сервиса, ошибки человека и непредвиденные последствия изменения. Отметим, что одному событию могут присваиваться несколько категорий;

Or — источник события (внутренний, междоменный, внешний, неизвестный);

Im — возможное влияние события (затронутые активы). Например, в [31] различаются функциональное влияние и информационное влияние;

Sev — серьезность события. Серьезность события является субъективной мерой его степени влияния на функционирование или целостность бизнес-процессов и информации. При определении серьезности события учитываются следующие факторы [32]: масштаб влияния, критичность системы или сервиса, чувствительность информации, обрабатываемой через систему или сервис, вероятность распространения на другие сегменты. Для серьезности события могут использоваться четыре уровня: высокий, средний, низкий, NA ("Not Applicable" не применим, используется для событий, принятых за инцидент, но после расследования не оказавшихся таковыми);

Pr — приоритет события. Приоритет устанавливает очередность обработки событий. По методике ITIL приоритет события определяется таблично на основе срочности устранения события и серьезности события (1 — критический, 2 — высокий, 3 — средний, 4 — низкий, 5 — планируемый) [33].

Для описания событий ИБ предложен ряд подходов [34, 35], существуют также попытки по определению единого формата данных, генерируемых различными источниками о событиях [36]. Заметим, что описание конкретного события определяется политикой ИБ, и сюда может включаться вся информация, интересная для области применения. В этой работе событие ИБ формально описывается следующим образом:

$$e = (Etype, DT, ID, rb, sev, v_1, \dots, v_n),$$

где e — событие ИБ; $Etype$ — описывает тип события; DT — указывает время и дату происхождения события; ID — идентификаторы источника, где обнаружено событие; rb — степень доверия источника; sev — степень риска события; v_1, \dots, v_n — другие атрибуты, требуемые для описания события в зависимости от области применения (например, IP-адрес, протокол, номер порта и т. д.). Атрибуты v_i могут

принимать одиночные значения или значения как набор элементов. События одинакового типа обладают одинаковым множеством атрибутов.

Кроме того, описание событий ИБ обычно включает и текстовую информацию, например, симптомы, проявления события, сообщения об ошибках и т. д. Для описания этих данных ниже предлагается использовать метод латентного семантического индексирования (Latent Semantic Indexing, LSI) [37].

LSI преобразует текстовый документ в семантический вектор. Для генерации этого вектора LSI использует матрицу, которая описывает частоту слов в наборе документов. Пусть имеется набор m документов, и в этих документах встречаются n терминов (слов). Пусть, для определенности, столбцы матрицы соответствуют документам, а строки — словам, встречающимся в документах. Каждый элемент a_{ij} матрицы $A_{m \times n}$ вычисляется следующим образом:

$$a_{ij} = \frac{n_{ij}}{t_j} \log \frac{N}{T_i},$$

где n_{ij} — число встречаемости i -го термина в j -м документе; t_j — число терминов в j -м документе; N — число документов; T_i — число документов, содержащих i -й термин.

Используя сингулярное разложение (SVD — singular value decomposition) [37], матрицу $A_{m \times n}$ разлагают на произведение трех матриц: $A = U \Sigma V^T$, где U — ортогональная матрица размером $m \times r$, V — ортогональная матрица размером $n \times r$; $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$ — диагональная матрица размера $r \times r$, $r = \text{rank}(A)$, σ_i — сингулярное значение A , $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$.

LSI уменьшает синонимии и полисемии взятием s наибольших сингулярных значений, в результате уменьшается ранг матрицы A ; $A_s = U_s V_s^T$. Оптимальное значение s выбирается в зависимости от r . Семантический вектор документов в A индексируется рядами V_s . А семантический вектор нового документа или запроса вычисляют, используя U_s, Σ_s [38].

Отдельным вопросом является представление решений. В общем случае под решением можно понимать:

- действия, выполнявшиеся ранее в аналогичных ситуациях, и их результаты;
- руководство к действию в данной ситуации, основанное на регламентах;
- экспертные рекомендации, как выполнить действия;
- справочники и иные документы, требуемые при принятии и реализации решения.

Описание результата применения решения — обработки события ИБ — может содержать:

- планируемое и фактическое время обработки события;
- действия, проведенные для устранения события и его последствий;
- результат обработки события;

- перечень того, что можно сделать, чтобы избежать события.

Описание результата может также включать ссылки на другие события (прецеденты).

После описания ситуации и решения мы можем формально определить прецедент следующим образом.

Прецедент p состоит из кортежа $\langle s, r, h, z \rangle$, где ситуация $s \in S$, связанное с ней решение $r \in R$ выполняется группой экспертов $h \in H$ под руководством ЛПП $z \in Z$.

Каждой поставленной ситуации s может соответствовать несколько решений, таким образом, можно допустить, что существуют прецеденты вида $\langle s, r \rangle$ и $\langle s, r' \rangle$, которые отличаются при условии, что $r \neq r'$.

Каждый прецедент $p_i \in \{p\}$ может рассматриваться как условная импликация $s_i \Rightarrow r_i$, $i = \overline{1, n}$. Таким образом, если задана некоторая ситуация $s_i \approx s_j$, $j = \overline{1, n}$, $i \neq j$, и существует прецедент $p_i = \langle s_i, r_i, h_i, z_i \rangle$, можно утверждать, что r_j является приближенным решением для ситуации s_i . Чем ближе ситуация s_i к ситуации s_j , тем правдоподобнее, что r_j является решением для s_i .

Выбор прецедентов

Проблема выбора самого "ближайшего" прецедента к рассматриваемому прецеденту из базы прецедентов является самой важной частью систем, основанных на CBR. Механизмы поиска в обычных базах данных и CBR-системах различны. Поиск в CBR-системах в основном основан на двух методах: индуктивный метод [39], который использует индуктивные алгоритмы для построения дерева решений, и метод ближайшего соседа, который классифицирует прецеденты на основе заранее определенных метрик подобия [40].

Метод "ближайшего соседа" является самым распространенным и часто используемым подходом. По методу " k ближайших соседей" (K -nearest neighbors, k -NN) сначала из базы прецедентов находят k схожих прецедентов (обычно $k = 3$ или $k = 5$), а затем среди них на основе выбранной метрики подобия определяется прецедент, ближайший текущему прецеденту.

Степень близости двух прецедентов определяется на основе степени близости соответствующих признаков прецедентов. Каждому признаку присваивается весовой коэффициент, отражающий относительную важность признака. Для определения степени близости прецеденту по всем признакам можно использовать следующую формулу:

$$\text{sim}(i, k) = \frac{\sum_{j=1}^n w_j \text{sim}(e_{ij}, e_{kj})}{\sum_{j=1}^n w_j},$$

где w_j — вес j -го признака; sim — метрика подобия; e_{ij} и e_{kj} — значения признака e_j для текущего события i и прецедента k соответственно.

Выбор метрики подобия sim является очень важной задачей, так как поиск похожих прецедентов значительно зависит от этого выбора. В качестве метрик подобия прецедентов традиционно используют Эвклидово расстояние, расстояние Махаланобиса, манхэттенскую метрику, меры сходства Хемминга, Роджерса—Танимото. Кроме того, могут использоваться и другие меры близости [40—42]: коэффициенты Рао, Жаккара, Гаусса, мера близости Миркина, метрики Брея—Кертиса, Канберровской и др. В каждом конкретном случае выбор метрики проводится с учетом физической и статистической природы используемой информации.

Как видно из вышеприведенного описания события, часть признаков прецедента принимает числовые и номинальные значения, а другая часть описывается текстовыми данными, которые кодируются семантическими векторами. Учитывая это, предлагается использовать два вида метрик подобия.

Подобие признаков с числовыми и номинальными значениями предлагается определить на основании гетерогенной функции расстояния (Heterogeneous Euclidean-Overlap Metric, НЕОМ) [43], которая использует различные функции расстояния для различных видов признаков: метрику перекрытия для номинальных признаков и нормированного Евклидова расстояния для числовых признаков. Используя НЕОМ, расстояние $sim_v(x, c)$ между двумя векторами признаков x и c определяется следующей формулой:

$$sim_v(x, c) = \sqrt{\sum_{i=1}^m d(x_i, c_i)^2},$$

где

$$d(x_i, c_i) = \begin{cases} 1, & \text{если тип } x_i \text{ или } c_i \text{ неизвестен,} \\ overlap(x_i, c_i), & \text{если } x_i \text{ и } c_i \text{ символьного типа,} \\ \frac{|x_i - c_i|}{\max_i - \min_i}, & \text{в противном случае;} \end{cases}$$

$$overlap(x_i, c_i) = \begin{cases} 0, & \text{если } x_i \neq c_i; \\ 1, & \text{если } x_i = c_i. \end{cases}$$

Подобие между двумя семантическими векторами $a = (a_1, a_2, \dots, a_n)$ и $b = (b_1, b_2, \dots, b_n)$ вычисляется косинусом угла между этими векторами:

$$sim_s(a, b) = \frac{ab^T}{\|a\| \cdot \|b\|}.$$

Таким образом, для заданного прецедента c и прецедента x и значений подобия $sim_v(x, c)$ и $sim_s(x, c)$

для соответствующих частей векторов x и c подобие S между x и c вычисляется формулой

$$S(x, c) = \alpha sim_v(x, c) + \beta sim_s(x, c).$$

Параметры α и β определяют значимость метрик близости, например, можно положить $\alpha = 0,6$ и $\beta = 0,4$.

Оптимизация весов признаков методом PSO

В методе CBR выбор подходящего веса для каждого признака имеет большое влияние на точность модели. Вес признака должен быть высоким, если он важен в процессе определения подобия между парой прецедентов. В последнее время такие эвристические подходы, как генетические алгоритмы (ГА), метод роя частиц (Particle Swarm Optimization, PSO) и эволюционные стратегии, используются для оптимизации весов, а также выбора признаков [44, 45].

Существующие исследования подтверждают, что метод CBR с весом, оптимизированным PSO или ГА, может значительно улучшить результаты оценки [44, 45]. Кроме того, PSO может превзойти ГА в поиске оптимизированных весов для CBR [45]. Поэтому в настоящей работе метод PSO был принят для определения подходящих весов признаков в методе CBR.

Метод PSO является одним из эффективных методов нахождения приближенных решений задач глобальной оптимизации. PSO впервые был предложен в 1995 г. Дж. Кеннеди и Р. Эберхартом [46] для оптимизации нелинейных функций, впоследствии применялся к множеству различных задач, были предложены многочисленные модификации [47]. PSO основан на моделировании группового поведения, например, стаи птиц или косяка рыб. Как и в генетических алгоритмах, PSO использует популяцию потенциальных решений для исследования пространства поиска. Начальную популяцию PSO можно сгенерировать случайно. В отличие от генетического алгоритма для генерации нового поколения потенциальных решений PSO не использует операторы, имитирующие естественную эволюцию. Взамен оператора мутации PSO опирается на обмен информацией между индивидуумами (частицами) популяции (роя). В сущности, каждая частица регулирует свою траекторию относительно своей наилучшей предыдущей позиции ($pbest$ — personal best) и наилучшей предыдущей позиции, достигнутой в ее локальном соседстве ($gbest$ — global best). В глобальной версии PSO весь рой рассматривается как локальное соседство, и, таким образом, происходит глобальный обмен информацией, и частицы используют открытия и опыт всех соседей при поиске областей, подающих надежды.

Каждое возможное решение соответствует позиции отдельной частицы. Текущая позиция (состояние) частицы характеризуется координатами в пространстве решений. Предположим, что про-

пространство решений d -мерное, i -ю частицу роя представим d -мерным вектором $X_i = (x_{i1}, \dots, x_{id})$, наилучшую частицу роя (имеющую максимальное значение функции) обозначим индексом g . Наилучшую позицию (которая соответствует наибольшему значению функции) i -й частицы обозначим $P_i = (p_{i1}, \dots, p_{id})$, а скорость i -й частицы (изменение позиции) обозначим $V_i = (v_{i1}, \dots, v_{id})$.

PSO является итеративным процессом. На каждой итерации сначала обновляется текущая скорость каждой частицы, при этом учитываются текущая скорость частицы, расстояния между текущей позицией и лучшим предыдущим персональным и глобальным опытом (позицией). Затем позиция каждой частицы обновляется с использованием значения новой скорости этой частицы. Обновление векторов скоростей и позиций частиц осуществляется по следующим формулам (верхний индекс показывает итерацию):

$$V_i^{k+1} = wV_i^k + c_1r_{i1}^k(P_i^k - X_i^k) + c_2r_{i2}^k(P_g^k - X_i^k),$$

$$X_i^{k+1} = X_i^k + V_i^k,$$

где $i = 1, \dots, N$, N — размер популяции; w — коэффициент инерции; c_1 и c_2 — положительные константы, называемые *когнитивным* и *социальным* параметрами; r_{i1} и r_{i2} — случайные числа, равномерно распределенные в интервале $[0, 1]$.

Константы c_1 и c_2 выбираются из интервала $[0, 2]$, малые значения c_1 и c_2 позволяют частице блуждать далеко от целевого пространства, тогда как большие значения способствуют достаточно быстрому движению к области пространства или за ее пределы.

Коэффициент инерции является важным для сходимости PSO, он используется для контроля влияния предыдущих скоростей на текущую скорость. Так, большое значение инерции способствует исследованию (поиску новых областей), а малое значение имеет склонность способствовать эксплуатации (подстраиванию текущей области поиска). Экспериментальные результаты показывают, что целесообразно выбирать вначале большое значение коэффициента инерции, стимулируя глобальное исследование пространства поиска, а затем, постепенно уменьшая коэффициент инерции, получить оптимальное решение за меньшее число итераций.

Основная цель метода PSO состоит в оптимизации функции пригодности. В этом исследовании основной целью является поиск оптимальных или почти оптимальных весов признаков, которые дают наиболее точное предсказание прецедента. Функция пригодности (f_T) для тестового набора данных T определена следующей формулой [48]:

$$\text{Maximize } f_T = \sum_{k=1}^n \text{hit}_k,$$

где n — размер тестового набора данных T ; hit_k — число совпадений между ожидаемым результатом (EO_k) и фактическим результатом (AO_k), т. е. если $EO_k = AO_k$ hit_k равен 1, иначе hit_k равен 0.

Заключение

В работе рассмотрены вопросы разработки модели ситуационного управления информационной безопасностью электронного правительства на оперативном уровне с использованием подхода рассуждения на основе прецедентов. Разработана концептуальная модель ситуационного управления информационной безопасностью электронного правительства и предложена ее реализация на основе теории прецедентов. Предложены подходы для представления знаний о ситуации информационной безопасности и выбора прецедентов, а также оптимизации весов признаков прецедентов на основе метода роя частиц. Предложенный подход можно обобщить для тактического и стратегического уровней управления. Полученные в работе результаты могут быть обобщены на тактические и стратегические уровни управления ИБ электронного правительства. Разработка подходов для адаптации решений является направлением дальнейших исследований. Полученные результаты можно внедрить в соответствующих ситуационных центрах по управлению ИБ, работающих в реальном режиме времени.

Список литературы

1. Grönlund A., Horan T. A. Introducing e-gov: history, definitions, and issues // Communications of the Association for Information Systems. 2004. V. 15. P. 713–729.
2. Wimmer M., von Bredow B. E-government: aspects of security on different layers // Proc. of the 12th International Workshop on Database and Expert Systems Applications. 2001. P. 350–355.
3. Chen Ye-Sho, Chong P. P., Zhang B. Cyber security management and e-government // Electronic Government. 2004. V. 1, N. 3. P. 316–327.
4. Агулиев Р. М., Имамвердиев Я. Н. Информационная безопасность э-государства: актуальные направления исследований // Проблемы информационного общества. 2010. № 1. С. 3–13.
5. Поспелов Д. А. Ситуационное управление. Теория и практика. М.: Наука, 1986.
6. Клык Ю. И. Ситуационное управление большими системами. М.: Энергия, 1981.
7. Aamodt A., Plaza E. Case-based reasoning: foundational issues, methodological variations, and system approaches // AI Communications. 1994. V. 7, N. 1. P. 39–59.
8. de Mántaras L. R., Bridge D., McSherry D. Case-based reasoning: an overview // AI Communications. 1997. V. 10, N. 1. P. 21–29.
9. Королев В. И., Новиков А. А., Шарков А. Е. Электронное правительство: анализ, концептуальное представление, функциональные приложения // Информационные технологии. Приложение. 2011. № 8. 32 с.
10. Delcambre L., Giuliano G. Digital government research in academia // IEEE Computer Society. 2005. V. 38, N 12. P. 33.
11. Федосеева Н. Н., Чайковская М. А. Понятие и сущность концепции электронного государства // Российская юстиция. 2011. № 11. С. 6–10.
12. Бачило И. Л. Правовая платформа построения электронного государства // Информационное право. 2008. № 4. P. 3–9.
13. Голобуцкий А. П., Шевчук О. Б. Электронное правительство. Киев: УМС-Атлант, 2002. 170 с.

14. Курнос И. Н. Реализация концепции электронного правительства: новый этап // Информационное общество. 2004. Вып. 6. С. 18–20.
15. Ланде Д. В. E-Government — лозунг или технология? // Телеком. 2003. № 5. С. 60–61.
16. Sood A. K., Enbody R. J. Targeted cyberattacks: a superset of advanced persistent threats // IEEE Security & Privacy. 2013. V. 11, N. 1. P. 54–61.
17. Li T., Zhong W. The Centralized or decentralized security analysis of e-government intercommunication // Proc. of International Conference on E-Business and E-Government (ICEE). 2010. P. 4760–4763.
18. Luckham D. The power of events. Boston: Addison-Wesley. 2002.
19. Killreese G., Kosakowsky K., Ruefle R., Zajicek M. State of the practice of computer security incident response team (CSIRT's) // Technical Report N IA-233, Carnegie Mellon Software Engineering Institute. 2003. 291 p.
20. Surhone L. M., Tennoe M. T., Henssonow S. F. EINSTELN (US-CERT program). Saarbrücken: Betascript Publishing. 2010.
21. Mockler R. I. Situational theory of management. Harvard Business Review. 1971. V. 49, N. 3. P. 151.
22. Савченко А. Б. Искусство управления ситуацией: опыт Востока и Запада. М.: Маркет ДС. 2006.
23. Витгих В. А. К определению понятия "ситуационное управление" // Материалы XIV Междунар. конф. "Проблемы управления и моделирования в сложных системах". г. Самара (Россия). 19–22 июня 2012 г. С. 112–115.
24. Schank R., Avelson R. Scripts, plans, goals and understanding. NJ: Erlbaum, 1977.
25. Watson I. Applying case-based reasoning. Morgan Kaufman, 1997.
26. Kim J., Hwang I. CBR evaluation modeling for security risk analysis in information security system // International Conference Security Technology. 2008. P. 66–70.
27. Balachandran B., Safavi-Naini R., Pieprzyk J. Case-based reasoning for intrusion detection // Computer Security Applications Conference. 1996. P. 214–223.
28. Zhu A., Zhang J. A case-based reasoning method for network security situation analysis // Control, Automation and Systems Engineering, 2011. P. 1–4.
29. Hernández J. Z., Serrano J. M. Knowledge-based models for emergency management systems // Expert Systems with Applications. 2001. V. 20, N. 2. P. 173–186.
30. Information security incident management — establishing an information security incident management capability // Information Security Forum (ISF). April 2006.
31. NIST Special Publication 800-61: Computer security incident handling guide. National Institute of Standards and Technology. January 2004.
32. West-Brown M. J., Stikvoort D., Kossakowski K. P. Handbook for computer security incident response teams (CSIRTs) // Report: CMU/SEI-98-HB-001. Carnegie Mellon University/Software Engineering Institute. December, 1998.
33. Foundations of ITIL V3 / Jan van Bon (Ed.). Zaltbommel (NL): Van Haren Publishing, 2007.
34. Debar H., Curry D., Feinstein B. The intrusion detection message exchange format (IDMEF). RFC 4765 (Experimental). 2007.
35. Kotenko I., Polubelova O., Chechulin A. and Saenko I. Data Repository for SIEM Systems // Future Internet. 2013. V. 5, N. 3. P. 355–375. doi: 10.3390/fi5030355.
36. Deerwester S., Dumais S., Landauer T., Furna G., Harshman R. Indexing by latent semantic analysis // Journal of the American Society for Information Science. 1990. V. 41, N. 6. P. 391–407.
37. Berry M. W., Drmac Z., Jessup E. R. Matrices, vector spaces, and information retrieval // SIAM Review. 1999. V. 41, N. 2. P. 335–362.
38. Wess S., Althoff K.-D., Derwand G. Using k-d trees to improve the retrieval step in case-based reasoning // Topics in Case-Based Reasoning, Lecture Notes in Computer Science. 1994. V. 837. P. 167–181.
39. Cunningham P. A taxonomy of similarity mechanisms for case-based reasoning // IEEE Transactions on Knowledge and Data Engineering. 2009. V. 21, N. 11. P. 1532–1543.
40. Núñez H. et al. A comparative study on the use of similarity measures in case-based reasoning to improve the classification of environmental system situations // Environmental Modelling & Software. 2004. V. 19, N. 9. P. 809–819.
41. Li H., Sun J. Gaussian case-based reasoning for business failure prediction with empirical data in China // Information Sciences. 2009. V. 179, N. 1–2. P. 89–108.
42. Wilson D. R., Martinez T. R. Improved heterogeneous distance functions // Journal of Artificial Intelligence Research. 1997. V. 6. P. 1–34.
43. Ahn H., Kim K. Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach // Applied Soft Computing. 2009. V. 9, N. 2. P. 599–607.
44. Wu D., Li J., Liang Y. Linear combination of multiple case-based reasoning with optimized weight for software effort estimation // Journal of Supercomputing. 2010. V. 64, N. 3. P. 898–918.
45. Kennedy J., Eberhart R. C. A new optimizer using particle swarm theory // Proc. IEEE Int. Conf. Neural Networks. 1995. P. 1941–1948.
46. Sousa T., Silva A., Neves A. Particle swarm based data mining algorithms for classification tasks // Parallel Computing. 2004. V. 30. P. 767–783.
47. Fu Y., Shen R. GA based CBR approach in Q&A system // Expert Systems with Applications. 2004. V. 26, N. 2. P. 167–170.

Ya. N. Imamverdiyev, Institute of Information Technology of ANAS, Baku, Azerbaijan
e-mail: yadigar@lan.ab.az

A Model of Situational Management of E-Government Information Security

E-government information security management is an ill structured problem: object of management is a complex socio-technical system, consisting of autonomous components, each of which acts purposefully; numerous processes (social, political, and technological), significantly interacting with each other occurs in the system. For reliable ensuring of e-government information security complex dynamic situations must be continuously and automatically identified and responded in real time. As a result, efficient methods of recognition, prediction, reasoning and handling of situations are required — operations collectively identified as situational management. In this paper we develop a conceptual model of situational management of e-government information security and propose its implementation using case-based reasoning. Approaches for representation of the knowledge about the information security situation and the choice of appropriate cases, as well as optimizing the weights of case features by particles swarm optimization are proposed. The proposed approach can be generalized for the tactical and strategic levels of management.

Keywords: information security, e-government, situational management, case based reasoning, particle swarm optimization

References

1. **Grönlund A., Horan T. A.** Introducing e-gov: history, definitions, and issues. *Communications of the Association for Information Systems*. 2004. V. 15. P. 713–729.
2. **Wimmer M., von Bredow B.** E-government: aspects of security on different layers. *Proc. of the 12th International Workshop on Database and Expert Systems Applications*. 2001. P. 350–355.
3. **Chen Ye-Sho, Chong P. P., Zhang B.** Cyber security management and e-government. *Electronic Government*. 2004. V. 1, N. 3. P. 316–327.
4. **Alguliyev R. M., Imamverdiyev Y. N.** Informatsionnaya bezopasnost e-gosudarstva: aktualnyye napravleniya issledovaniy. *Problemy informatsionnogo obshchestva*. 2010. N. 1. P. 3–13.
5. **Pospelov D. A.** Situatsionnoye upravlenie. Teoriya i praktika. M.: Nauka, 1986.
6. **Klykov Yu. I.** Situatsionnoye upravleniye bolshimi sistemami. M.: Energiya, 1981.
7. **Aamodt A., Plaza E.** Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Communications*. 1994. V. 7. N. 1. P. 39–59.
8. **de Mántaras L. R., Bridge D., McSherry D.** Case-based reasoning: an overview. *AI Communications*. 1997. V. 10, N. 1. P. 21–29.
9. **Kopolev V. I., Novikov A. A., Shapkov A. E.** Elektronnoye pravitelstvo: analiz, kontseptualnoye predstavleniye, funktsionalnyye prilozheniya. *Informatsionnyye tekhnologii. Prilozheniye*. 2011. N. 8.
10. **Delcambre L., Giuliano G.** Digital government research in academia. *IEEE Computer Society*. 2005. V. 38, N. 12. P. 33.
11. **Fedoseyeva N. N., Chaykovskaya M. A.** Ponyatiye i sushchnost kontseptsii elektronnoy gosudarstva. *Rossiyskaya yustitsiya*. 2011. N. 11. P. 6–10.
12. **Bacjilo I. L.** Pravovaya platforma postroyeniya elektronnoy gosudarstva. *Informatsionnoye pravo*. 2008. N. 4. P. 3–9.
13. **Golobutskiy A. P., Shevchuk O. B.** Elektronnoye pravitelstvo. Kiyev: UMC-Atlant, 2002. 170 p.
14. **Kurnosov I. N.** Realizatsiya kontseptsii elektronnoy pravitelstva: novyy etap. *Informatsionnoye obshchestvo*. 2004. Vyp. 6. P. 18–20.
15. **Lande D. V.** E-Government — lozung ili tekhnologiya? *Telekom*. 2003. N. 5. S. 60–61.
16. **Sood A. K., Enbody R. J.** Targeted cyberattacks: a superset of advanced persistent threats. *IEEE Security & Privacy*. 2013. V. 11, N. 1. P. 54–61.
17. **Li T., Zhong W.** The Centralized or decentralized security analysis of e-government intercommunication. *Proc. of International Conference on E-Business and E-Government (ICEE)*. 2010. P. 4760–4763.
18. **Luckham D.** The power of events. Boston: Addison-Wesley, 2002.
19. **Killrecre G., Kosakowsky K., Ruefle R., Zajicek M.** State of the practice of computer security incident response team (CSIRT's). *Technical Report N IA-233*, Carnegie Mellon Software Engineering Institute. 2003. 291 p.
20. **Surhone L. M., Tennoe M. T., Henssonow S. F.** EINSTEIN (US-CERT program). Saarbrücken: Betascript Publishing. 2010.
21. **Mockler R. I.** Situational theory of management. *Harvard Business Review*. 1971. V. 49, N. 3. P. 151.
22. **Savchenko A. B.** Iskustvo upravleniye situatsiyey: opyt Vostoka i Zapada. M.: Market DS. 2006.
23. **Vittikh V. A.** K opredeleniyu ponyatiya "situatsionnoye upravleniye". *Materialy XIV Mezhdunar. konf. "Problemy upravleniya i modelirovaniya v slozhnykh sistemakh"* Samara (Rossiya). 19–22 iyunya 2012 g. P. 112–115.
24. **Schank R., Avelson R.** Scripts, plans, goals and understanding. NJ: Erlbaum, 1977.
25. **Watson I.** Applying case-based reasoning. San Mateo, CA: Morgan Kaufman, 1997.
26. **Kim J., Hwang I.** CBR evaluation modeling for security risk analysis in information security system. *International Conference Security Technology*. 2008. P. 66–70.
27. **Balachandran B., Safavi-Naini R., Pieprzyk J.** Case-based reasoning for intrusion detection. *Computer Security Applications Conference*. 1996. P. 214–223.
28. **Zhu A., Zhang J.** A case-based reasoning method for network security situation analysis. *Control, Automation and Systems Engineering*, 2011. P. 1–4.
29. **Hernández J. Z., Serrano J. M.** Knowledge-based models for emergency management systems. *Expert Systems with Applications*. 2001. V. 20, N. 2. P. 173–186.
30. **Information security incident management — establishing an information security incident management capability.** *Information Security Forum (ISF)*. April 2006.
31. **NIST Special Publication 800-61: Computer security incident handling guide.** National Institute of Standards and Technology. January 2004.
32. **West-Brown M. J., Stikvoort D., Kossakowski K. P.** Handbook for computer security incident response teams (CSIRTs). *Report: CMU/SEI-98-HB-001*. Carnegie Mellon University/Software Engineering Institute. December, 1998.
33. **Foundations of ITIL V3 / Jan van Bon (Ed.).** Zaltbommel (NL): Van Haren Publishing, 2007.
34. **Debar H., Curry D., Feinstein B.** The intrusion detection message exchange format (IDMEF). RFC 4765 (Experimental). 2007.
35. **Kotenko I., Polubelova O., Chechulin A. and Saenko I.** Data Repository for SIEM Systems. *Future Internet*. 2013. V. 5, N. 3. P. 355–375. doi: 10.3390/fi5030355.
36. **Deerwester S., Dumais S., Landauer T., Furna G., Harshman R.** Indexing by latent semantic analysis. *Journal of the American Society for Information Science*. 1990. V. 41, N. 6. P. 391–407.
37. **Berry M. W., Drmac Z., Jessup E. R.** Matrices, vector spaces, and information retrieval. *SIAM Review*. 1999. V. 41, N. 2. P. 335–362.
38. **Wess S., Althoff K.-D., Derwand G.** Using k-d trees to improve the retrieval step in case-based reasoning. *Topics in Case-Based Reasoning, Lecture Notes in Computer Science*. 1994. V. 837. P. 167–181.
39. **Cunningham P.** A taxonomy of similarity mechanisms for case-based reasoning. *IEEE Transactions on Knowledge and Data Engineering*. 2009. V. 21, N. 11. P. 1532–1543.
40. **Núñez H.** et al. A comparative study on the use of similarity measures in case-based reasoning to improve the classification of environmental system situations. *Environmental Modelling & Software*. 2004. V. 19, N. 9. P. 809–819.
41. **Li H., Sun J.** Gaussian case-based reasoning for business failure prediction with empirical data in China. *Information Sciences*. 2009. V. 179, N. 1–2. P. 89–108.
42. **Wilson D. R., Martinez T. R.** Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*. 1997. V. 6. P. 1–34.
43. **Ahn H., Kim K.** Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach. *Applied Soft Computing*. 2009. V. 9, N. 2. P. 599–607.
44. **Wu D., Li J., Liang Y.** Linear combination of multiple case-based reasoning with optimized weight for software effort estimation. *Journal of Supercomputing*. 2010. V. 64, N. 3. P. 898–918.
45. **Kennedy J., Eberhart R. C.** A new optimizer using particle swarm theory. *Proc. IEEE Int. Conf. Neural Networks*. 1995. P. 1941–1948.
46. **Sousa T., Silva A., Neves A.** Particle swarm based data mining algorithms for classification tasks. *Parallel Computing*. 2004. V. 30. P. 767–783.
47. **Fu Y., Shen R.** GA based CBR approach in Q&A system. *Expert Systems with Applications*. 2004. V. 26, N. 2. P. 167–170.

Л. С. Куравский¹, д-р техн. наук, проф., декан, П. А. Мармалюк¹, канд. техн. наук, доц.,
С. Н. Баранов², канд. техн. наук, ген. директор, В. И. Алхимов¹, д-р физ.-мат. наук, проф.,

Г. А. Юрьев¹, канд. техн. наук, доц., С. В. Артюхина³, директор

¹Московский городской психолого-педагогический университет,

факультет информационных технологий, 127051 Москва, ул. Сретенка, 29

I.s.kuravsky@gmail.com

²ООО "Русское авиационное общество", 125130 Москва, ул. К. Цеткин, 33/50

³ГОУ СПО "Колледж метростроя № 53 им. Героя Советского Союза М. Ф. Панова"

Марковские модели глазодвигательной активности и их применение для тестирования профессиональных навыков

Предложен новый метод моделирования и анализа пространственно-динамических особенностей глазодвигательной активности испытуемых, основанный на представлении движения взгляда по поверхности стимула с помощью случайных марковских процессов с непрерывным временем. Рассмотрена процедура идентификации и оценки адекватности вероятностных моделей, а также способ построения классификатора, позволяющего оценить степень соответствия наблюдаемых распределений движения взгляда эталонным распределениям, полученным для различных диагностируемых групп. В качестве иллюстрации приведены примеры практического применения предложенного метода для оценки уровня математической подготовки испытуемых, степени дифференциации пространственно-временных распределений глазодвигательной активности пилотов летательных аппаратов разной квалификации и сформированности профессиональных навыков у диспетчеров автоматизированного комплекса управления инженерными системами здания.

Ключевые слова: диагностика, глазодвигательная активность, марковский процесс, уравнения Фоккера—Планка—Колмогорова

Введение

Методы регистрации движений глаз широко используются в современных фундаментальных исследованиях в области когнитивной психологии в целях выявления особенностей глазодвигательной активности (ГДА) испытуемых, например, в процессе чтения [4, 9, 13, 17], при решении задач на наглядно-образное мышление [10—12, 16, 18] или при изучении психологически значимых стимулов, например изображений лиц людей [2].

Результаты таких исследований часто свидетельствуют о наличии факторов (например, таких как возраст, пол, этническая или расовая принадлежность, уровень профессиональной подготовки, сформированность навыков), существенно влияющих на изменчивость показателей ГДА в рамках фиксированного стимульного материала и определяющих пространственно-временные особенности траекторий взгляда в гомогенных (по уровню выраженности фактора) группах испытуемых. К настоящему времени выявлен достаточно представительный список показателей ГДА, в которых проявляются значимые различия для диагностируемых категорий испытуемых [1].

Одной из наиболее актуальных практических задач, связанных с исследованием ГДА, является создание автоматизированных систем психолого-педагогической диагностики нового типа, основанных на регистрации ГДА испытуемых при изучении

ими стимульного материала (например при чтении текстов специальной тематики) и последующем построении и применении математических моделей, описывающих особенности ГДА в различных диагностируемых группах. Однако далеко не все показатели, представляющие интерес для фундаментальных исследований, пригодны для построения эффективных формализованных процедур диагностики, реализуемых в рамках таких систем. В то же время среди них можно выделить ряд характеристик, обеспечивающих решение востребованных частных задач, связанных с построением интеллектуальных и компетентностных тестов.

Одно из подобных решений представлено в этой работе, где рассмотрен новый метод моделирования и анализа пространственно-динамических особенностей глазодвигательной активности испытуемых, основанный на представлении движения взгляда по поверхности стимула с помощью случайного марковского процесса с непрерывным временем. Кроме того, здесь описаны процедура идентификации и оценки адекватности прогнозирующих моделей, а также способ построения вероятностного классификатора, позволяющего оценивать степень соответствия траекторий взгляда неидентифицированных испытуемых эталонам, полученным для различных диагностируемых групп. Подробное математическое обоснование разработанного метода дано в работе [9]. На основе полученных результатов соз-

дается соответствующее программно-аппаратное обеспечение для диагностики испытуемых.

В качестве иллюстрации приведены примеры практического применения предложенного подхода для оценки уровня математической подготовки испытуемых и степени дифференциации пространственно-временных распределений глазодвигательной активности пилотов летательных аппаратов разной квалификации, которые построены на результатах экспериментов, проведенных специалистами Факультета информационных технологий МГППУ, Центра экспериментальной психологии МГППУ и Русского авиационного общества [4, 9]. Полученные экспериментальные и расчетные результаты позволяют говорить об эффективности рассмотренного подхода и возможности построения на его основе автоматизированных диагностических средств для выявления степени сформированности навыков и компетенций.

1. Математические основы метода

Этапы решения задачи классификации. Будем полагать, что испытуемым в процессе регистрации глазодвигательной активности предъявляется плоский визуальный стимул прямоугольной формы, представляющий некоторую содержательную информацию (текст, формулы, изображения, пользовательский интерфейс, веб-сайт и т. п.). Испытуемые разделены на несколько заранее известных групп (категорий), отражающих уровень исследуемого фактора. Полагая, что для каждой из этих групп доступна соответствующая выборка наблюдаемых траекторий взора по изучаемому стимулу, поставим задачу распознавания группы, к которой принадлежит испытуемый, по данным о движении его

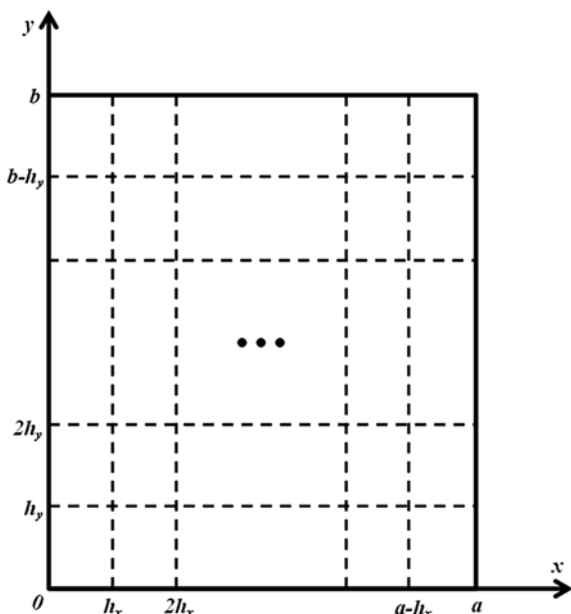


Рис. 1. Дискретизация замкнутой области стимула прямоугольной сеткой

взора по поверхности стимула. Классификация в указанной постановке обеспечивается решением рассмотренных далее следующих подзадач:

- разделения поверхности стимула на зоны с помощью прямоугольной сетки (рис. 1);
- расчета выборочных частот пребывания в зонах стимула по выборке траекторий в каждой диагностируемой группе испытуемых;
- идентификации (аппроксимации) распределений вероятности пребывания взора на поверхности стимула по полученным выборочным частотам для всех диагностируемых групп испытуемых;
- оценки и сравнения вероятностей наблюдаемой траектории движения взора неидентифицированного испытуемого по поверхности стимула для каждой из заданных диагностируемых групп.

Разделение стимула и построение модели. Пусть движение взора по поверхности стимула описывается векторной непрерывной случайной функцией времени $\mathbf{U}(t) = (U_1(t), U_2(t))$, где $t \in T$. Плотность вероятности $p(\mathbf{u}, t)$, характеризующая распределение значений функции $\mathbf{U}(t)$, определяется при этом как $p(\mathbf{u}, t)d\mathbf{u} = P\{\mathbf{u} \leq \mathbf{U}(t) < \mathbf{u} + d\mathbf{u}\}$, где $\mathbf{u} = (u_1, u_2) = (x, y)$, а $d\mathbf{u}$ — элемент объема.

Будем полагать, что рассматриваемая случайная функция принадлежит к достаточно общему классу *марковских процессов* или *процессов без последствия*, для которых в любые моменты времени $t_0 < t_1 < \dots < t_m$ выполняется условие $p(\mathbf{u}_m, t_m | \mathbf{u}_{m-1}, t_{m-1}; \dots; \mathbf{u}_0, t_0) = p(\mathbf{u}_m, t_m | \mathbf{u}_{m-1}, t_{m-1})$. Такие процессы полностью определяются начальным распределением $p(\mathbf{u}_0, t_0)$ и распределением *переходных вероятностей* $p(\mathbf{u}, t | \mathbf{u}_0, t_0)$. Это распределение в случае непрерывных процессов удовлетворяет *уравнению Фоккера—Планка—Колмогорова*:

$$\frac{\partial p}{\partial t} = - \sum_{j=1}^2 \frac{\partial}{\partial u_j} (a_j p) + \sum_{j=1}^2 \sum_{k=1}^2 \frac{\partial^2}{\partial u_j \partial u_k} (b_{jk} p)$$

с начальным условием $p(\mathbf{u}, t | \mathbf{u}_0, t_0) |_{t=t_0} = \delta(\mathbf{u} - \mathbf{u}_0)$.

Функции $a_j(\mathbf{u}, t)$ и $b_{jk}(\mathbf{u}, t)$ представляют интенсивности рассматриваемых марковских процессов.

В работе [9] показано, что поиск приближенного решения уравнения Фоккера—Планка—Колмогорова сводится к решению задачи Коши для системы обыкновенных дифференциальных уравнений относительно плотности вероятности пребывания взора по поверхности стимула, усредненной по площади ячеек. Если, умножив каждую усредненную плотность вероятности на площадь ячейки, рассматривать каждую ячейку прямоугольной сетки размерностью $l \times q$ как отдельное *дискретное состояние случайного процесса с непрерывным временем*, то данная система уравнений превратится в *систему уравнений Колмогорова*, описывающую динамику изменения вероятностей пребывания взора испытуемого в указанных состояниях. Каждому

состоянию с номером n при этом ставится в соответствие уравнение

$$\frac{d\tilde{p}^n(t)}{dt} = -\sum_j \eta_{nj}(t)\tilde{p}^n(t) + \sum_j \eta_{in}(t)\tilde{p}^i(t), \quad (1)$$

где $\tilde{p}^n(t) = \bar{p}^n(t)G$ и $\tilde{p}^i(t) = \bar{p}^i(t)G$ — вероятности пребывания в n -м и i -м состояниях; G — площадь ячейки прямоугольной сетки; $n, i = 1, \dots, M$; $M = lq$; η_{nj} — интенсивность переходов из состояния n в состояние j ; η_{in} — интенсивность переходов из состояния i в состояние n .

Следуя представленному в работе [9] методу решения диагностической задачи, необходимо, используя данные наблюдений, решить подзадачу идентификации распределений вероятности пребывания в состоянии на поверхности стимула для всех диагностируемых групп испытуемых.

Идентификация модели: критерий минимума χ^2 . Аппроксимация этих распределений с учетом проведенной дискретизации сводится к оценке свободных параметров модели — интенсивностей переходов между состояниями рассматриваемого случайного процесса $\{\eta_{ij}^i\}_{i,j=1,\dots,M}$. Для решения задачи идентификации выполняется численная процедура многомерной нелинейной оптимизации, обеспечивающая решение обратной задачи для системы дифференциальных уравнений Колмогорова [7, 8, 14, 15]. В результате ее решения находится набор свободных параметров, который определяет систему уравнений, решение этой системы имеет заданные характеристики (вероятностные функции времени, аппроксимирующие наблюдаемую динамику вероятностей пребывания в зонах стимула).

Оценки свободных параметров (интенсивностей переходов) могут быть определены, опираясь на *критерий соответствия наблюдаемых и прогнозируемых гистограмм*, описывающих распределения частот пребывания в состояниях процесса. В качестве такого критерия далее используется *статистика Пирсона*

$$\chi^2 = \sum_{i=1}^M \frac{(F_i - \tilde{p}^i(t)N)^2}{\tilde{p}^i(t)N},$$

где $\tilde{p}^i(t)N$ — прогнозируемая частота попадания в i -е состояние в момент времени t ; N — число элементов (испытуемых) в выборке, а вероятностные функции $\tilde{p}^i(t)$ получаются путем численного интегрирования систем уравнений Колмогорова. Соответствующие наблюдаемые частоты F_i определяются как число испытуемых, взор которых в заданные контрольные моменты времени пребывал в соответствующих зонах стимула.

Величина χ^2 является мерой соответствия в том смысле, что ее большие значения означают плохое согласование прогнозируемых и наблюдаемых ре-

зультатов, а малые значения — хорошее согласование. Таким образом, решение задачи идентификации сводится к нахождению таких интенсивностей переходов η_{ij} , которые обеспечивают минимальное значение суммы статистик Пирсона в те контрольные моменты времени, для которых имеются результаты наблюдений.

Такой способ идентификации свободных параметров называется *методом минимума χ^2* [6]. Согласно теореме Крамера [6, с. 462—470], при выполнении ряда общих условий, в случае адекватности модели случайного процесса наблюдениям, рассмотренная задача идентификации имеет единственное решение, которое сходится по вероятности к искомому решению, а значения статистики χ^2 асимптотически описываются распределением χ^2 с $M - \gamma - 1$ степенями свободы, где γ — число опережаемых значений свободных параметров. Значение распределения позволяет использовать приведенную выше статистику для проверки гипотезы о том, что полученные прогнозируемые частоты попадания в состояния согласуются с результатами наблюдений.

Очевидно, что использование этой статистики корректно только при выполнении условий указанной выше теоремы. Если условия теоремы не выполнены, то вычисление приближений к оценкам максимального правдоподобия с использованием критерия χ^2 остается возможным, однако полученное решение может быть не единственным, а значения соответствующей статистики не обязаны быть распределены как χ^2 . Следует отметить, что одним из признаков нарушения условий теоремы является невыполнение неравенства $M > \gamma - 1$.

Критерий, применяемый в случае нарушения условий теоремы Крамера. В случае нарушения условий теоремы для оценки степени соответствия модели наблюдениям следует подобрать критерий, отличный от указанного выше. Такой критерий строится на основе анализа регрессионной зависимости между наблюдаемыми и прогнозируемыми частотами пребывания в состояниях исследуемого случайного процесса в контрольные моменты времени [9]. Для этого предположим, что наблюдаемые частоты выражаются через соответствующие прогнозируемые вероятности по формуле гауссовской простой линейной регрессии

$$F_k^i = a^i \tilde{p}_k^i + \varepsilon_k^i,$$

где F_k^i — наблюдаемая частота пребывания в i -м состоянии в k -й контрольный момент времени ($k = 1, \dots, K$), \tilde{p}_k^i и ε_k^i — соответствующие прогнозируемые вероятности и случайные ошибки; a^i — неизвестные параметры регрессии. Полагается, что случайные ошибки ε_k^i независимы и распре-

лены по нормальному закону с нулевым средним значением и одинаковой для всех моментов времени дисперсией.

Оценки степени соответствия результатов решения наблюдениям обеспечиваются с помощью стандартных критериев, которые используются для проверки статистической значимости используемой регрессионной модели, а именно: критерия Стьюдента для проверки гипотез $a^i = 0$ и F -критерия для проверки значимости уравнения регрессии.

Кроме того, в ряде случаев можно дополнительно проверить, входит ли единица в 95 %-ный доверительный интервал для величины a^i/N . Если рассмотренная регрессионная модель статистически значима при заданном уровне α , а единица входит в указанный доверительный интервал, то можно говорить о приемлемой степени соответствия результатов решения наблюдениям.

Понижение размерности задачи идентификации. Как правило, идентификация свободных параметров случайных процессов со структурой связей между состояниями, представленной на рис. 2, является сложной вычислительной задачей. Однако имеется эффективный прием, позволяющий упростить эту работу.

Для этого строки и столбцы состояний исходной сети могут быть объединены в сгруппированные состояния, а именно: нахождение в состоянии V_j ($j = 1, 2, \dots, q$) равносильно нахождению в одном из состояний $S_j, S_{j+q}, \dots, S_{j+(l-1)q}$, а нахождение в состоянии H_i ($i = 1, 2, \dots, l$) равносильно нахождению в одном из состояний $S_{(i-1)q+1}, S_{(i-1)q+2}, \dots, S_{iq}$ (рис. 3).

Получаемые при этом графы состояний представлены на рис. 4 и 5.

Поскольку вероятность пребывания в каждом из сгруппированных состояний равна сумме вероятностей пребывания в составляющих его состояниях исходного случайного процесса, уравнения Колмогорова для вероятностей нахождения взора в сгруппированных состояниях получаются путем согласованного суммирования левых и правых частей всех уравнений (1), которые соответствуют ячейкам, формирующим данные состояния:

$$\begin{cases} \frac{dp_V^1}{dt} = v_{1,-}p_V^2 - v_{1,+}p_V^1; \\ \dots \\ \frac{dp_V^i}{dt} = v_{i-1,+}p_V^{i-1} + v_{i,-}p_V^{i+1} - (v_{i,+} + v_{i-1,-})p_V^i \quad (2) \\ (i = 2, 3, \dots, q-1); \\ \dots \\ \frac{dp_V^q}{dt} = v_{q-1,+}p_V^{q-1} - v_{q-1,-}p_V^q, \end{cases}$$

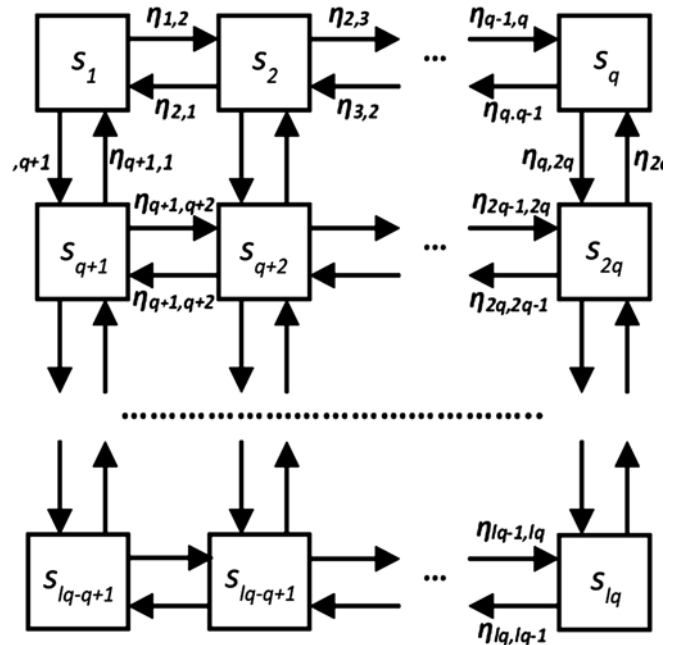


Рис. 2. Граф дискретных состояний, аппроксимирующий случайный процесс перемещения взора по поверхности стимула, соответствующий проведенной дискретизации стимула

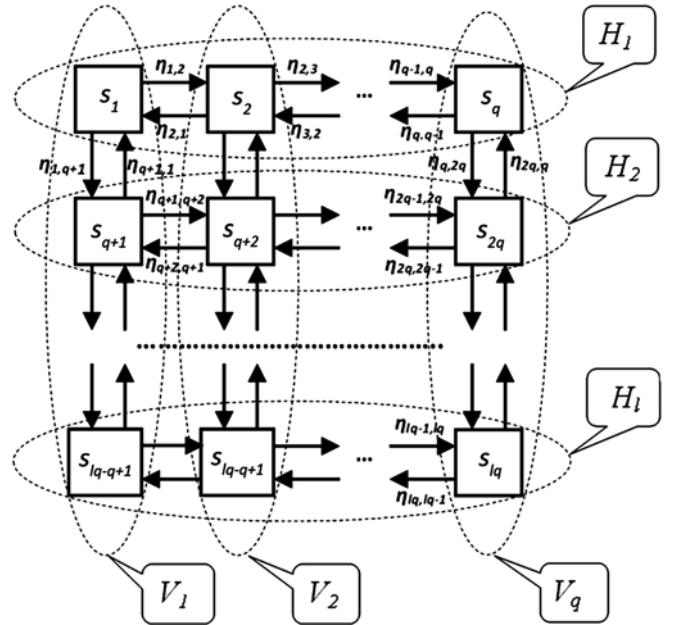


Рис. 3. Объединение состояний в группы

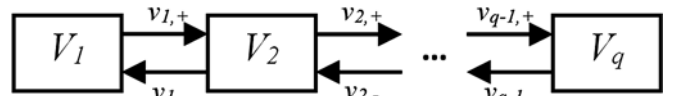


Рис. 4. Граф сгруппированных состояний (результат группировки столбцов)

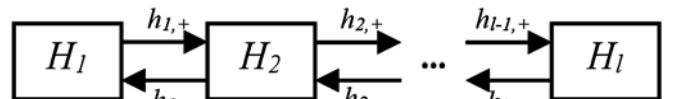


Рис. 5. Граф сгруппированных состояний (результат группировки строк)

	V1	V2	V3	V4
H1	<p>Корректны ли следующие рассуждения?</p> <p>Число λ называется <i>собственным значением</i> матрицы A, если существует такой ненулевой вектор x, что</p>			
H2	<p>$Ax = \lambda x$.</p> <p>Любой ненулевой вектор x, удовлетворяющий данному уравнению, называется <i>собственным вектором</i> матрицы A, соответствующим собственному значению λ.</p> <p>Представим указанное уравнение в виде $Ax = \lambda Ex$ где E – единичная матрица.</p> <p>Запишем его в следующей форме:</p>			
H3	<p>$(Ax - \lambda E)x = 0$.</p> <p>Если матрица $Ax - \lambda E$ не является вырожденной, то полученная система линейных алгебраических уравнений имеет единственное решение. Поэтому собственные значения определяются из следующего характеристического соотношения:</p>			
H4	<p>$\det(Ax - \lambda E) \neq 0$.</p>			

Рис. 7. Дискретизация области стимула: в левых верхних и правых нижних углах указаны их координаты; строки и столбцы обозначены, соответственно, как H1–H4 и V1–V4

пояснив их мотивы. Ответ испытуемого фиксируется с помощью средств аудиозаписи.

После проведения эксперимента аудиозаписи ответов были проанализированы экспертами в целях определения корректности ответа, данного испытуемым. В соответствии с оценками экспертов выборка испытуемых была разделена на группы правильно и неправильно выполнивших задание.

Запись движений глаз проводили с помощью стационарной системы регистрации движений глаз *SMI High Speed* [5] (рис. 6) в монокулярном режиме с частотой 500 Гц. Стимульный монитор работал с разрешением 1280×1024 точек. Калибровку проводили стандартным способом с использованием девяти опорных точек.

Применение нового диагностического метода. Стимульный материал задания связан с решением

алгебраической проблемы собственных значений. Он содержит две контрольные ошибки: во-первых, вектор x некорректно вынесен за скобки, и, во-вторых, собственные значения матрицы некорректно определяются из неравенства $\det(Ax - \lambda E) \neq 0$. Область, на которой располагается стимульный материал, была без предварительных преобразований разбита прямоугольной сеткой размерностью 4×4 на 16 ячеек, что обеспечило высоту строк и ширину столбцов равными, соответственно, 256 и 320 точкам (рис. 7).

Выборка испытуемых ($N = 24$) после исключения испытуемого с экстремальными результатами (слишком долго читавшего) была разделена на две диагностируемые группы: неправильно ($N = 11$) и правильно выполнивших задание ($N = 12$) испытуемых. Выборки траекторий, соответствующие этим группам, использовались в данном случае только в качестве обучающих множеств (используемых для идентификации моделей и "внутренней" оценки качества классификации) ввиду малого числа наблюдений.

Аппроксимация решения и результаты диагностики. Уравнения для графов сгруппированных состояний, представленных на рис. 4 и 5, принимают вид, аналогичный виду уравнений (2) и (3). Интенсивности переходов между состояниями исследуемого случайного процесса идентифицировались по отдельности для следующих друг за другом временных интервалов длительностью 10 с.

Численное интегрирование систем уравнений Колмогорова с идентифицированными значениями указанных параметров позволило вычислить прогнозируемые вероятности пребывания в сгруппированных состояниях.

Пример зависимости вероятностей от времени с соответствующими оценками, полученными в результате наблюдений, в качестве иллюстрации приведен на рис. 8.

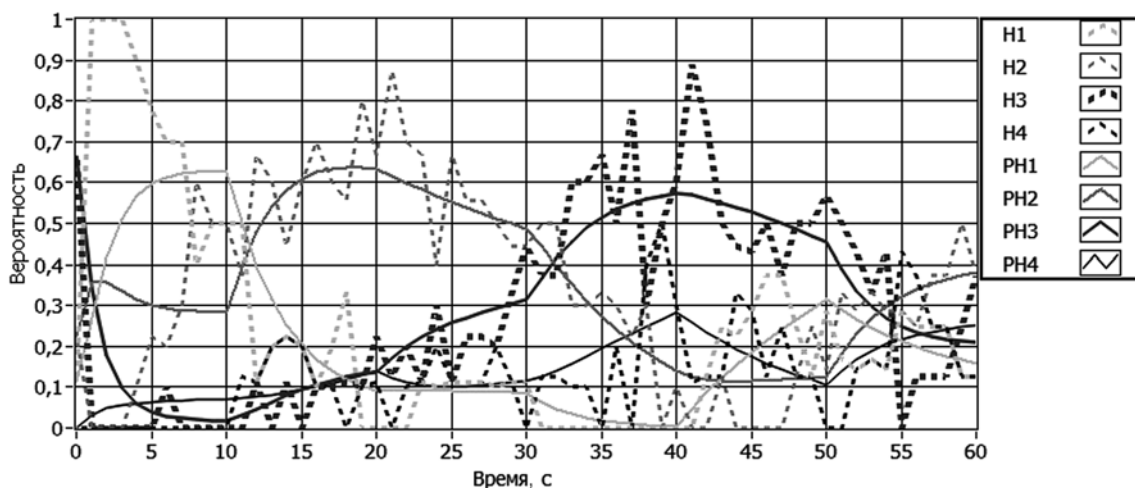


Рис. 8. Зависимости от времени (в секундах) прогнозируемых вероятностей пребывания в состояниях, сгруппированных по строкам, и соответствующих им оценок, полученных в результате наблюдений, для испытуемых, не выполнивших задание: H1–H4 — наблюдаемые оценки (показаны штриховой линией); PH1–PH4 — прогнозируемые оценки

Полученные модели были применены для построения классификатора, который позволил отнести испытуемых обучающей выборки к группам верно и неверно выполнивших задание с приемлемой точностью. Вероятностные оценки, характеризующие прохождение наблюдаемых траекторий, обеспечили правильный выбор категории испытуемых в 87 % экспериментов (имеет место статистически значимое отличие от равномерного распределения по критерию Пирсона, $p < 0,00001$), что в данном примере также позволяет говорить о высокой эффективности предложенного подхода к построению процедуры тестирования.

3. Второй пример практического применения

Экспериментальная процедура и оборудование.

Целью второго исследования была экспериментальная проверка гипотезы о наличии значимых различий в пространственно-временных распределениях глазодвигательной активности пилотов летательных аппаратов разной квалификации в целях последующего использования этих данных для оценки уровня летной подготовки. Актуальность работы обусловлена необходимостью создания систем компьютерной диагностики нового типа, основанных на анализе движений взора испытуемых. Это, в свою очередь, определяется тем, что видеоокуло-



Рис. 9. Полет самолета Ф-15 "Сапсан" в процессе эксперимента



Рис. 10. Фиксация мобильной системы видеорегистрации движений глаз SMI iView X HED на голове пилота

графия является фактически единственным средством получения информации о корректности считывания данных с авиационных приборов, которая недоступна пилотам-инструкторам и является важной составляющей летной квалификации. В рамках летной программы были проведены полеты экспериментаторов с двумя разными уровнями навыков пилотирования на самолете Ф-15 "Сапсан" (рис. 9).

Запись движений глаз проводилась с помощью мобильной системы регистрации движений глаз SMI iView X HED (рис. 10) в монокулярном режиме с частотой 50 Гц.

Калибровку проводили стандартным способом с использованием четырех опорных точек на приборной доске. Наблюдаемая сцена фиксировалась с помощью фронтальной видеокамеры.

Результаты исследований. В результате обработки экспериментальных данных были построены приведенные на рис. 11–13 (см. четвертую сторону обложки) оценки усредненных вероятностных распределений координат взора на поверхности приборной доски во время взлета, полета по кругу и посадки для неопытного пилота и пилота средней квалификации. Поверхность приборной доски разбита на прямоугольные зоны группировки с помощью регулярной сетки размерностью 10×10 .

Для количественной оценки степени дифференциации полученных вероятностных распределений предлагается *метод правдоподобных траекторий*, включающий следующие шаги:

1) генерация случайных последовательностей моментов времени и нахождение соответствующих им T наиболее вероятных последовательностей точек на поверхности стимула в случае каждого из полученных распределений (формирование наиболее правдоподобных траекторий);

2) вычисление для каждой из T последовательностей точек, полученных на предыдущем шаге, показателя правдоподобия, представляющего вероятность ее наблюдения в случае каждого из рассматриваемых распределений (эти величины характеризуют степень соответствия наиболее правдоподобных траекторий всем исследуемым уровням квалификации);

3) сравнение по критерию знаковых рангов Уилкоксона на имеющейся выборке наиболее вероятных последовательностей точек, полученных на шаге 1,

распределений показателей
$$\left\{ \ln \left(\frac{L_{ii}}{\sum_{j=1}^Q L_{ij}} \right) \right\}_{i=1}^Q$$
, где

L_{ij} — показатель правдоподобия для наиболее вероятной последовательности точек, вычисленной для i -го распределения, в случае j -го распределения, а Q — число рассматриваемых уровней квалификации, с равномерным усредненным уровнем $1/Q$.

В случае статистически значимого отличия от равномерного усредненного уровня $1/Q$ делается вывод о значимой дифференциации рассматриваемых вероятностных распределений.

Представленные на рис. 11–13 (см. четвертую сторону обложки) пары вероятностных распределений имеют высокосignификантные отличия в смысле приведенного выше критерия ($p < 0,0001$; $T = 20$), что свидетельствует о возможности использования рассмотренного подхода для оценки уровня летной подготовки пилотов и реальной перспективе построения соответствующих автоматизированных диагностических средств.

4. Третий пример практического применения

Экспериментальная процедура и оборудование. Целью третьего исследования была экспериментальная проверка гипотезы о наличии значимых различий пространственно-временных распределений глазодвигательной активности у диспетчеров с разными уровнями сформированности навыков работы с автоматизированным комплексом управления инженерными системами современного здания "Умный дом" в целях последующего использования этих результатов для оценки результатов учебной подготовки. В рамках эксперимента проводилась видеорегистрация движений глаз испытуемых

при прослушивании постановки задачи, связанной с возникшей на удаленном контролируемом объекте экстренной ситуацией, при изучении стимула (рис. 14) и при ответе на целевой вопрос.

Запись движений глаз проводили с помощью системы видеорегистрации SMI HED в монокулярном стационарном режиме с частотой 50 Гц. Калибровку проводили по девяти точкам экрана стимульного монитора стандартным способом. Наблюдаемую сцену фиксировала фронтальная видеочка айтрекера.

Результаты исследований. Для каждой измеренной траектории движения взгляда оценивали "персональную" динамику вероятностного распределения координат взгляда. Для этого задавали определенное число точек траектории, определяющих ширину "скользящего" по траектории "окна", после чего траекторию разбивали на максимально возможное число участков без перекрытия. Затем для каждого из полученных участков траектории оценивали выборочные частоты пребывания взгляда испытуемого в пространственных состояниях $V1$ – $V4$, $H1$ – $H4$ наблюдаемого стимула (рис. 14).

Полученные гистограммы выборочных частот пребывания в этих состояниях использовали при идентификации марковских моделей. Каждой диагностируемой группе испытуемых соответствовала своя идентифицированная модель.

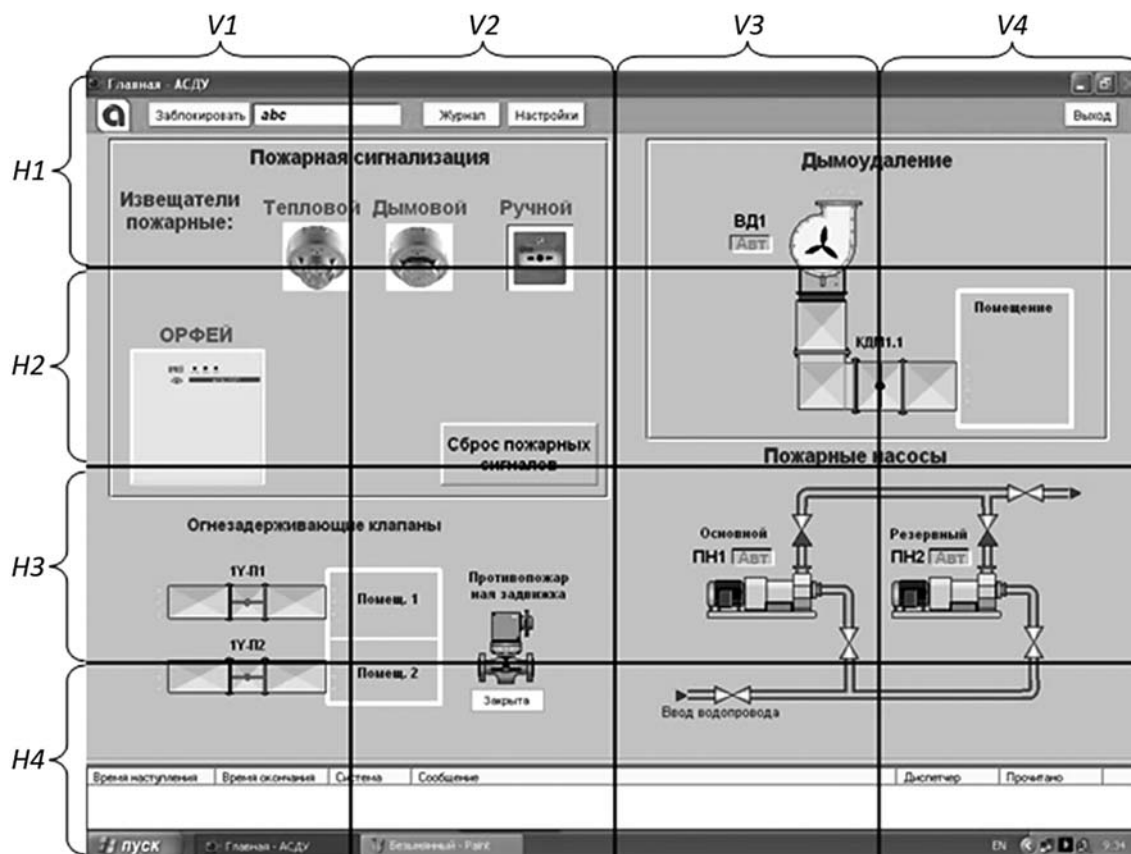


Рис. 14. Разделение стимула задачи на горизонтальные и вертикальные зоны (пространственные состояния)

На рис. 15 (см. четвертую сторону обложки) приведены оценки усредненных вероятностных распределений точек расположения взора на поверхности стимула для неопытных и опытных диспетчеров.

Предложенный способ классификации обеспечил правильный выбор категории испытуемых в 82 % проведенных экспериментов (имеет место статистически значимое отличие от равномерного распределения по критерию Пирсона, $p < 0,05$), что свидетельствует о возможности использования рассмотренного подхода для оценки уровня сформированности профессиональных навыков у диспетчеров автоматизированного комплекса управления инженерными системами здания.

Заключение

Разработан новый метод моделирования и анализа пространственно-динамических особенностей глазодвигательной активности испытуемых, основанный на аппроксимации динамики распределения вероятностей пребывания взора в различных зонах визуального стимула с помощью случайных марковских процессов с непрерывным временем и применении средств идентификации прогнозирующих моделей,

Разработана формализованная процедура, позволяющая определять наиболее вероятные диагнозы путем оценки степени соответствия траектории взора тестируемого эталонным вероятностным распределениям, идентифицированным по результатам наблюдений для каждой из диагностируемых групп испытуемых. На основе этой процедуры создано программно-аппаратное обеспечение, позволившее провести ряд экспериментальных исследований.

Предложен метод правдоподобных траекторий, позволяющий количественно оценивать степень дифференциации вероятностных распределений пребывания взора на поверхности стимула в целях оценки возможности их использования для диагностики.

Представленные примеры практического применения предложенного метода для оценки уровня математической подготовки испытуемых, степени дифференциации пространственно-временных распределений глазодвигательной активности пилотов летательных аппаратов разной квалификации и сформированности профессиональных навыков у диспетчеров автоматизированного комплекса управления инженерными системами здания позволяют говорить об эффективности рассмотренного подхода и возможности построения на его основе автоматизированных диагностических средств для выявления степени сформированности навыков и компетенций.

Работа выполнена при поддержке РФФИ, проект № 14-06-00191.

1. **Барабанщиков В. А.** Окуломоторные структуры восприятия. М.: Изд-во ИП РАН, 1997. 384 с.
2. **Барабанщиков В. А., Ананьева К. И., Харитонов В. Н.** Организация движений глаз при восприятии изображений лица // Экспериментальная психология. 2009. № 2. С. 31–60.
3. **Барабанщиков В. А.** Экспрессии лица и их восприятие. М.: Изд-во ИП РАН, 2012. 341 с.
4. **Безруких М. М., Иванов В. В.** Движения глаз в процессе чтения как показатель сформированности навыка // Физиология человека. 2013. Т. 39. № 1. С. 83–93.
5. **Демидов А. А., Жегалло А. В.** Оборудование СМІ для регистрации движений глаз: тест-драйв // Экспериментальная психология. 2008. № 1. С. 149–159.
6. **Крамер Г.** Математические методы статистики. М.: Мир, 1976. 648 с.
7. **Куравский Л. С., Баранов С. Н., Корниенко П. А.** Обучаемые многофакторные сети Маркова и их применение для исследования психологических характеристик // Нейрокомпьютеры: разработка и применение. 2005. № 12. С. 65–76.
8. **Куравский Л. С., Баранов С. Н., Юрьев Г. А.** Синтез и идентификация скрытых марковских моделей для диагностики усталостного разрушения // Нейрокомпьютеры: разработка и применение. 2010. № 12. С. 20–36.
9. **Куравский Л. С., Мармалюк П. А., Алхимов В. И., Юрьев Г. А.** Математические основы нового подхода к построению процедур тестирования // Экспериментальная психология. 2012. Т. 5, № 4. С. 75–98.
10. **Куравский Л. С., Мармалюк П. А., Алхимов В. И., Юрьев Г. А.** Применение обучаемых структур для анализа результатов компьютерного тестирования // Нейрокомпьютеры: разработка и применение. 2013. № 4. С. 18–27.
11. **Куравский Л. С., Мармалюк П. А., Барабанщиков В. А., Безруких М. М., Демидов А. А., Иванов В. В., Юрьев Г. А.** Обучаемые структуры как основа программного обеспечения для диагностики навыков и компетенции // Нейрокомпьютеры: разработка и применение. 2013. № 5. С. 46–60.
12. **Мармалюк П. А., Звонкина О. М.** Опорные показатели глазодвигательной активности при прохождении теста Равена и автоматизация их расчета // Молодые ученые — нашей новой школе: Матер. XI Межвуз. науч.-практ. конф. с межд. участием. — М.: Изд-во МГППУ, 2012. С. 350–352.
13. **Хохлова А. А.** Исследование глазодвигательной активности при прохождении матричного теста интеллекта Равена // Молодые ученые — нашей новой школе: Матер. X науч.-практ. межвуз. конф. М.: Изд. МГППУ, 2011. С. 343–345.
14. **Hayes T. R., Petrov A. A., Sederberg P. B.** A novel method for analyzing sequential eye movements reveals strategic influence on Raven's Advanced Progressive Matrices // Journal of Vision. 2011. V. 10. P. 1–11.
15. **Jansen A. R., Marriott K., Yelland G. W.** Parsing of algebraic expressions by experienced users of mathematics // European Journal of Cognitive Psychology. 2007. V. 19. P. 286–320.
16. **Kuravsky L. S., Baranov S. N.** The concept of multifactor Markov networks and its application to forecasting and diagnostics of technical systems // In: Proc. Condition Monitoring 2005, Cambridge, United Kingdom. July 2005. P. 111–117.
17. **Kuravsky L. S., Baranov S. N., Yuryev G. A.** Synthesis and identification of hidden Markov models based on a novel statistical technique in condition monitoring // In: Proc. 7th International Conference on Condition Monitoring & Machinery Failure Prevention Technologies, Stratford-upon-Avon, England, June 2010.
18. **Kuravsky L. S., Marmalyuk P. A., Alkhimov V. I., Yuryev G. A.** Mathematical backgrounds of a new technique for testing condition monitoring personnel professional skills // In: Proc. 10th International Conference on Condition Monitoring & Machinery Failure Prevention Technologies, Krakow, Poland, June 2013.
19. **Patalano A. L., Juhasz B. K., Dicke O.** The Relationship Between Indecisiveness and Eye Movement Patterns in a Decision Making Informational Search Task // Journal of Behavioral Decision Making. 2010. V. 23. P. 353–368.
20. **Rayner K.** Eye movements in reading and information processing: 20 years of research // Psychological Bulletin. 1998. V. 124. P. 372–422.
21. **Vigneau F., Caissie A., Bors D.** Eye-movement analysis demonstrates strategic influences on intelligence // Intelligence. 2006. V. 34. P. 261–272.

L. S. Kuravsky¹, Professor, P. A. Marmalyuk¹, Assistant Professor, S. N. Baranov², Assistant Professor, V. I. Alkhimov¹, Professor, G. A. Uryev¹, Assistant Professor, S. V. Artyukhina³, Director

¹ Moscow State University of Psychology and Education, Computer Science faculty, l.s.kuravsky@gmail.com

² JSC Russian Aviation Co. Ltd, Moscow,

³ Metrostroi college N 53

Markov Models of Oculomotor Activity and Their Application for Professional Skills Assessment

Suggested is a new method for modeling and analyzing spatio-temporal peculiarities of subjects oculomotor activity, which is based on representation of gaze movement on a stimulus surface with the aid of continuous-time Markov processes with discrete states. Procedure of model identification provided with goodness-of-fit tests and classifier-building technique, which makes it possible to quantitatively assess differences between observed and reference gaze-movement distributions estimated for different diagnosed groups, is described. Examples of practical application of the suggested approach for assessing subjects' mathematical background level, degree of differentiation of gaze movement spatio-temporal distributions of aircraft pilots with different proficiency levels as well as professional skills of automated supervisory utility systems control software users are given as illustrations.

Keywords: diagnostics, oculomotor activity, Markov process, Fokker—Planck—Kolmogorov equations

References

1. Barabanshchikov V. A. Okulomotornye struktury vospriyatiya. M.: IP RAS, 1997. 384 p.
2. Barabanshchikov V. A., Ananyeva K. I., Kharitonov V. N. Organizatsiya dvizheniy glaz pri vospriyatii izobrazheniy litsa. *Ekspierimentalnaya psikhologiya*. 2009. N 2. P. 31—60.
3. Barabanshchikov V. A. Ekspressii litsa i ih vospriyatie. M.: IP RAS, 2012. 341 p.
4. Bezrukikh M. M., Ivanov V. V. Dvizheniya glaz v processe chteniya kak pokazatel sformirovannosti navyka. *Fiziologiya cheloveka*. 2013. Vol. 39, N 1. P. 83—93.
5. Demidov A. A., Zhegallo A. V. Oborudovanie SMI dlya registratsii dvizheniy glaz: test-drayv. *Ekspierimentalnaya psikhologiya*. 2008. N. 1. P. 149—159.
6. Cramer G. Matematicheskie metody statistiki. M.: Mir, 1976. 648 p.
7. Kuravsky L. S., Baranov S. N., Kornienko P. A. Obuchaemye mnogofaktornye seti Markova i ikh primenenie dlya issledovaniya psikhologicheskikh kharakteristik. *Neyrokomp'yutery: razrabotka i primenenie*. 2005. N. 12. P. 65—76.
8. Kuravsky L. S., Baranov S. N., Yuryev G. A. Sintez i identifikatsiya skrytykh markovskikh modeley dlya diagnostiki ustalostnogo razrusheniya. *Neyrokomp'yutery: razrabotka i primenenie*. 2010. N. 12. P. 20—36.
9. Kuravsky L. S., Marmalyuk P. A., Alkhimov V. I., Yuryev G. A. Matematicheskie osnovy novogo podkhoda k postroenlyu protsedur testirovaniya. *Ekspierimentalnaya psikhologiya*. 2012. Vol. 5, N. 4. P. 75—98.
10. Kuravsky L. S., Marmalyuk P. A., Alkhimov V. I., Yuryev G. A. Primenenie obuchaemikh struktur dlya analiza rezultatov komp'yuternogo testirovaniya. *Neyrokomp'yutery: razrabotka i primenenie*. 2013. N. 4. P. 18—27.
11. Kuravsky L. S., Marmalyuk P. A., Barabanshchikov V. A., Bezrukikh M. M., Demidov A. A., Ivanov V. V., Yuryev G. A. Obuchaemie struktury kak osnova programmnoy obespecheniya dlya diagnostiki navykov i kompetentsii. *Neyrokomp'yutery: razrabotka i primenenie*. 2013. N. 5. P. 46—60.
12. Marmalyuk P. A., Zvonkina O. M. Opornye pokazateli glazodvigatelnoy aktivnosti pri prokhozhenii testa Ravena i avtomatizatsiya ikh rascheta. — Molodye uchenye — nashey novoy shkole: mater. XI Mezhevuz. nauch.-prakt. konf. s mezhd. uchastiem. M.: MSUPE, 2012. P. 350—352.
13. Khokhlova A. A. Issledovanie glazodvigatelnoy aktivnosti pri prokhozhenii matrichnogo testa intellekta Ravena. — Molodye uchenye — nashey novoy shkole: mater. XI Mezhevuz. nauch.-prakt. konf. s mezhd. uchastiem. M.: MSUPE, 2011. P. 343—345.
14. Hayes T. R., Petrov A. A., Sederberg P. B. A novel method for analyzing sequential eye movements reveals strategic influence on Raven's Advanced Progressive Matrices. *Journal of Vision*. 2011. V. 10. P. 1—11.
15. Jansen A. R., Marriott K., Yelland G. W. Parsing of algebraic expressions by experienced users of mathematics. *European Journal of Cognitive Psychology*. 2007. V. 19. P. 286—320.
16. Kuravsky L. S., Baranov S. N. The concept of multifactor Markov networks and its application to forecasting and diagnostics of technical systems. In: *Proc. Condition Monitoring 2005, Cambridge, United Kingdom*. July 2005. P. 111—117.
17. Kuravsky L. S., Baranov S. N., Yuryev G. A. Synthesis and identification of hidden Markov models based on a novel statistical technique in condition monitoring. In: *Proc. 7th International Conference on Condition Monitoring & Machinery Failure Prevention Technologies*, Stratford-upon-Avon, England. June 2010.
18. Kuravsky L. S., Marmalyuk P. A., Alkhimov V. I., Yuryev G. A. Mathematical backgrounds of a new technique for testing condition monitoring personnel professional skills. In: *Proc. 10th International Conference on Condition Monitoring & Machinery Failure Prevention Technologies*. Krakow, Poland, June 2013.
19. Patalano A. L., Juhasz B. J., Dicke O. The Relationship Between Indecisiveness and Eye Movement Patterns in a Decision Making Informational Search Task. *Journal of Behavioral Decision Making*. 2010. V. 23. P. 353—368.
20. Rayner K. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*. 1998. V. 124. P. 372—422.
21. Vigneau F., Caissie A., Bors D. Eye-movement analysis demonstrates strategic influences on intelligence. *Intelligence*. 2006. V. 34. P. 261—272.

УДК 621.3.049.771.14

С. В. Гаврилов, д-р техн. наук, зав. отделом, e-mail: sergey.v.gavrilov@ippm.ru

Г. А. Иванова, мл. науч. сотр.,

Институт проблем проектирования в микроэлектронике РАН (ИППМ РАН), Москва,

А. А. Манукян, аспирант, Национальный исследовательский университет "МИЭТ"

Новые проблемы логико-топологического синтеза заказных сложно-функциональных блоков и методы их решения

Работа посвящена решению проблем автоматизации проектирования заказных интегральных схем с технологическими нормами 22 нм и ниже. Особое внимание уделяется новой технологии, получившей название FinFET (технология с трехмерной структурой транзистора в форме плавника (fin)). Известно, что FinFET-технология обеспечивает улучшение быстродействия и мощности по сравнению со стандартной КМОП-технологией. Вместе с тем широкое распространение технологии FinFET затруднено ввиду проблем автоматизации проектирования схем и топологии: развитие средств автоматизированного проектирования интегральных схем традиционно отстает от развития технологии. Для решения проблем автоматизации проектирования в данной работе проведен сравнительный анализ существующих подходов разработки топологии FinFET-структур, а также сравнительный анализ этих структур со стандартной КМОП-структурой. Разработаны методы формирования топологической конструкции различных FinFET-структур для синтеза топологии элементов с регулярной топологической структурой в слоях поликремния и диффузии.

Ключевые слова: САПР (системы автоматизированного проектирования), SP-граф, сложно-функциональный блок (СФ-блок), FinFET-резисторы (транзисторы с трехмерной структурой в форме плавника), КМОП-технология

Введение

С переходом на новые технологические нормы проектирования микроэлектронных систем увеличивается состав и число правил проектирования, так, например, в правила проектирования для технологий с размерами транзистора 32 нм и ниже входит несколько тысяч ограничений. Соблюдение полного набора правил в автоматическом режиме становится невозможным при использовании существующих подходов к решению задач логического и топологического синтеза.

Переход к использованию регулярных структур в нижних слоях топологии позволяет решить проблемы возрастающего числа проектных норм для технологий 22 нм и ниже. Исторически регулярность топологии в нижних слоях связывают с так называемой технологией базовых матричных кристаллов (БМК) с предопределенным набором библиотечных элементов. Применение аналогичного подхода при проектировании заказных сложно-функциональных блоков позволяет значительно упростить проверку правил проектирования при разработке общего вида топологии.

В работах компании Intel для соблюдения требований регулярности в системах автоматизированного проектирования заказных микросхем предлагается подход к проектированию регулярных топо-

логических структур с использованием конструкции специального типа — так называемого транзисторного шаблона [1, 2]. Транзисторный шаблон определяется как промежуточная конструкция между транзистором и стандартной ячейкой [1, 2].

Данная работа представляет собой дальнейшее развитие идеи использования регулярности топологических структур в системах автоматизированного проектирования заказных микросхем по следующим основным направлениям:

1) согласованное решение проблем синтеза схемы и топологии на основе модели компонент в форме графа вложенности последовательно-параллельных структур (SP-граф) с независимым анализом цепей земли (*pull-down*) и питания (*pull-up*);

2) анализ и методы формирования регулярных шаблонов для новой технологии FinFET.

1. Существующие модели FinFET-транзисторов

FinFET-транзисторы — это новое направление в проектировании микросхем, где за счет использования трехмерного затвора транзистора в форме плавника повышается эффективная ширина затвора при сходной площади логической ячейки. В последнее время появился ряд публикаций, связанных с проектированием на основе FinFET-транзисторов.

Известны три основные конструкции FinFET-структур [3]:

1) SG (Shorted-Gate) или TG (Tri-Gate) FinFET, в которых затворы транзистора соединены друг с другом;

2) IG (Independent-Gate) или DG (Double-Gate) FinFET, где затворы транзистора независимы (на каждый затвор подается свой сигнал);

3) LP (Low Power) FinFET, где второй затвор (*back-gate*) транзистора подключен к обратному смещенному напряжению для уменьшения статической мощности.

В работе [3] проведен сравнительный анализ по потребляемой мощности библиотек с различными конструкциями транзисторов. Показано, что в случае TG-транзисторов статическая мощность уменьшается на 31,3 %, для комбинации LP- и TG-транзисторов статическая мощность уменьшается на 64,74 %, а полная мощность — в среднем на 14,51 %. При этом площадь в LP-режиме увеличивается в среднем на 122,6 %. Комбинация IG/LP- и TG-режима дает снижение полной мощности на 6,91 %, статической мощности — на 20,66 %, увеличение площади — на 62,2 %.

Использование FinFET-транзисторов при разработке набора базовых блоков для создания регулярной топологии приведено в работе [4], где рассмотрены два типа конструкций FinFET: SG-FinFET и IG-FinFET. Предлагается создавать геометрически регулярную топологию с помощью повтора однотипных блоков (элементы NAND2, NOR2 и NOT). Показано, что в плане уменьшения статической мощности и занимаемой площади для элементов NAND2, NOR2 и NOT эффективнее IG-FinFET, чем SG-FinFET. Так как не приведен эскиз топологии, сложно оценить правдоподобность полученных результатов.

В работах [5] и [6] можно найти возможные эскизы топологии FinFET-транзисторов и базовых логических элементов, таких как NOT, NOR2 и 2xNOR2. Вместе с тем на сегодняшний день проектирование на основе FinFET-транзисторов осуществляется на основе ограниченного набора библиотечных элементов, отсутствуют средства синтеза сложносоставных элементов, поэтому возникает необходимость самостоятельной разработки принципов создания топологии FinFET-структур для маршрута заказного проектирования с регулярными структурами.

На рис. 1 приведена простая структура SG (Shorted-Gate) FinFET транзистора. Структура состоит из "плавника" (*fin*) — области диффузии для стока и истока, которая окружена затвором. Физические размеры транзистора описываются высотой плавника h_{FIN} , толщиной плавника (или кремния) T_{SI} и длиной канала L_{FIN} [7].

Так как верхняя часть затвора отделена от плавника сравнительно толстым слоем оксида, вклад

этой части затвора в работу транзистора можно не учитывать [8]. У такого транзистора длина канала L_{FIN} , а ширина

$$W_{min} \approx 2h_{FIN} \quad (1)$$

На рис. 2 приведена структура IG (Independent-Gate) FinFET-транзистора, которую можно получить путем удаления верхней части затвора обычного SG-FinFET-транзистора [9].

Затворы IG-FinFET-транзистора независимы, это позволяет использовать один из затворов (*back gate*) для обратного смещения транзистора, что дает возможность управлять пороговым напряжением транзистора, тем самым его утечками и задержками [10]. Часто второй затвор подключают к самому низкому (или высокому) потенциалу для получения маломощных схем — LP (Low Power) FinFET. Независимость затворов IG-FinFET-структур также дает возможность образовать параллельно подключенные транзисторы, ширина каждого из которых $W_{min} = h_{FIN}$ [11].

Если соединить затворы IG-FinFET-транзистора, можно получить транзистор с шириной W_{min} согласно формуле (1), но это приведет к увеличению площади транзистора, так как контакт ко второму затвору требует определенного места и соблюдения правил проектирования [12].

Минимальная ширина FinFET-транзисторов равна W_{min} , из чего следует, что для получения ширины W , которая больше чем W_{min} , нужно подключить N_{FIT} транзисторов параллельно [13]:

$$N_{FIT} = W/W_{min} \quad (2)$$

Так как N_{FIT} — целое число, то получается, что ширина транзистора кратна W_{min} .

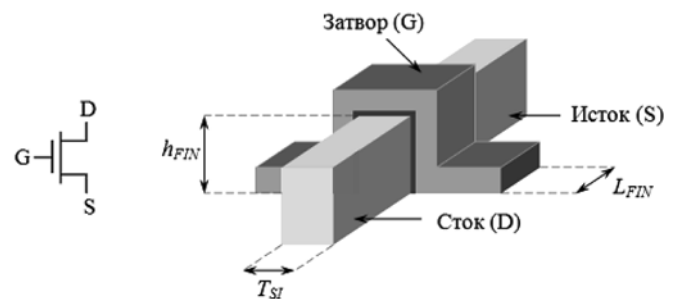


Рис. 1. Структура транзистора SG-FinFET

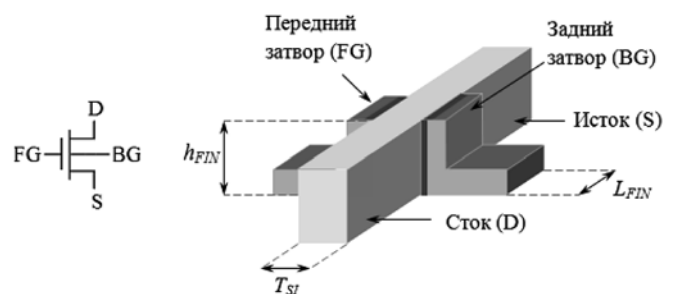


Рис. 2. Структура транзистора IG-FinFET

Таблица 1

Технологические нормы для разных технологий

Размеры	Технология		
	32 нм	45 нм	65 нм
L_{FIN}	35	45	55
T_{SI}	23	30	36
$h_{FIN} = 2T_{SI}$	46	60	72

Таблица 2

Сравнение топологий ячеек двухвходового ИЛИ-НЕ (NOR2)

Параметр	Bulk CMOS	SG FinFET	IG FinFET
Шаг плавников (<i>fin pitch</i>)	—	60 нм	174 нм
H_p	W_p	$0,5W_p$	$1,3W_p$
H_n	W_n	$0,5W_n$	$1,3W_n$
Площадь	S_b	$0,7S_b$	$1,79S_b$

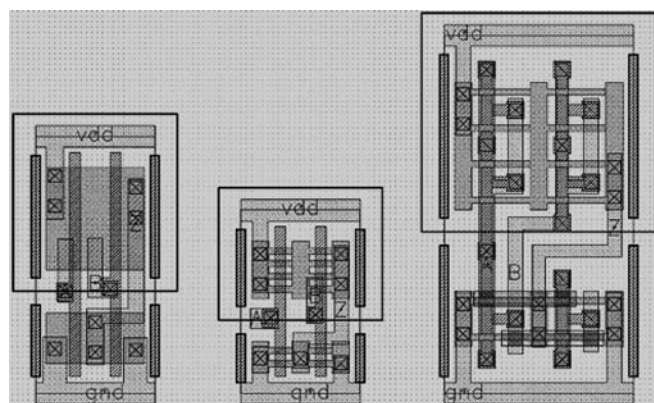


Рис. 3. Ячейка двухвходового ИЛИ-НЕ (слева направо): bulk CMOS-структура, SG-FinFET-структура, IG-FinFET-структура

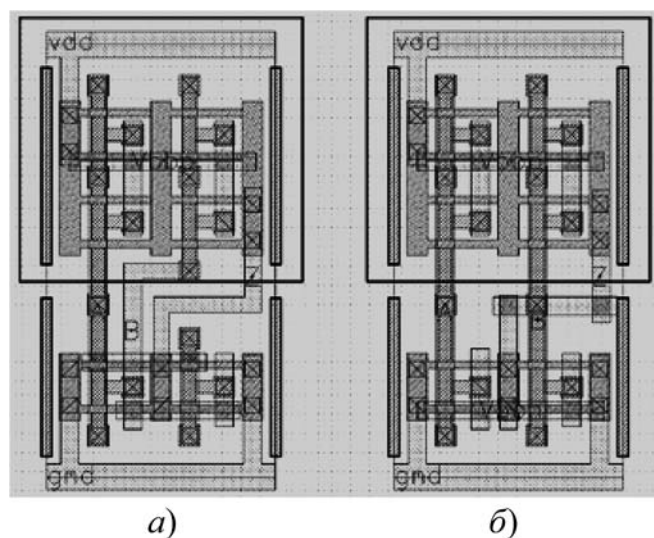


Рис. 4. Ячейка двухвходового ИЛИ-НЕ: IG-FinFET-структура с параллельно подключенными NFET-транзисторами (а), LP-FinFET-структура (б)

2. Автоматизация проектирования топологии FinFET-структур

В рамках данной работы был проведен сравнительный анализ топологий разных ячеек с технологическими нормами 32, 45 и 65 нм. На основе имеющихся данных делается прогноз о проектировании топологии для перспективных технологий с технологическими размерами 22 и 16 нм. В табл. 1 приведено сравнение размеров L_{FIN} , T_{SI} и h_{FIN} для технологий 32, 45 и 65 нм [8].

В табл. 2 приведены результаты сравнения топологий ячеек двухвходового ИЛИ-НЕ (NOR2), разработанного в рамках данной работы на основе технологии 45 нм.

На высоту ячейки влияют параметры H_p и H_n , которыми обозначены максимальные высоты области диффузии соответственно p - и n -канальных транзисторов (рис. 3, мелко заштрихованные области). Параметром S_b обозначена площадь ячейки по технологии bulk CMOS.

Для bulk CMOS-транзисторов H_p равна ширине W_p p -канального транзистора, а H_n — ширине W_n n -канального транзистора. Параметры W_p и W_n были выбраны соответственно 480 нм и 240 нм, для простоты расчетов. Высота плавника h_{FIN} выбрана в 2 раза больше толщины плавника T_{SI} [9].

На рис. 3 приведены двухвходовые ячейки ИЛИ-НЕ с одинаковой нагрузочной способностью. Можно заметить, что по сравнению со структурой bulk CMOS, SG-FinFET-структура меньше, за счет высоты ячейки.

В случае IG-FinFET-структуры контакты ко вторым затворам требуют определенного места. Как можно заметить из рис. 3, IG-FinFET-структура больше не только по высоте, но и по ширине, к тому же ввиду сложности трассировки первым металлом, приходится использовать второй металл, что усложняет трассировку на более высоком уровне металлизации.

Свойства IG-FinFET-структур довольно интересны и многообещающи, особенно с точки зрения управления утечками и уменьшения задержек транзистора, но нужно учитывать, что при той же площади нагрузочная способность ячеек падает в 2 раза. На рис. 4 приведены наиболее часто используемые IG-FinFET-структуры топологии [10, 11] ячейки ИЛИ-НЕ.

Благодаря независимости затворов IG-FinFET-структуры на рис. 4 n -канальные транзисторы подключены параллельно, а затворы *back-gate* p -канальных транзисторов подключены к напряжению обратного смещения, которое используется для управления пороговым напряжением транзисторов, что позволяет ощутимо сократить затраты по мощности. При этом нагрузочная способность падает частично вследствие того, что ширина p -канального транзистора в 2 раза меньше.

3. Разработка регулярных FinFET-структур в слоях поликремния и диффузии

В данном разделе описана попытка разработки регулярного шаблона в слоях поликремния и диффузии для FinFET-технологии.

3.1. Метод проектирования регулярных структур на основе выбора шаблона

Использование шаблонов при построении логических элементов является одним из подходов к проектированию схем с регулярной топологией.

На рис. 5 представлено сравнение обычной SG-FinFET-структуры стандартной ячейки ИЛИ-НЕ (рис. 5, а) с предлагаемой регулярной SG-FinFET-структурой на основе шаблона (рис. 5, б).

Особенность шаблона в том, что объединив множество таких же шаблонов, можно получить регулярность по слоям поликремния и диффузии (рис. 6 и см. ниже рис. 8).

В регулярной структуре присутствуют изолирующие затворы, которые подключены к шинам питания, следовательно, всегда закрыты. Между изолирующими затворами два функциональных затвора, как и в стандартной ячейке. По сравнению со структурой стандартной ячейки, в регулярной структуре ничего не меняется, кроме того, что слои диффузии выходят за пределы ячеек и появляются изолирующие затворы.

Однако, если стандартные ячейки позволяют менять размеры ячейки на кратные шаги сетки, а также менять число подключений к затворам внутри ячейки, то в регулярной структуре число подключений к затворам фиксировано, и ячейки должны быть кратны размерам самой маленькой ячейки. В нашем случае самые маленькие блоки — двухвходовые, в которых можно реализовать такие элементы, как ИЛИ-НЕ, И-НЕ, буфер, инвертор.

Итак, для реализации трехвходовых регулярных структур таких, как И-ИЛИ-НЕ21 (AOI21), понадобится ячейка вдвое больше (рис. 7).

Из рис. 7 можно заметить, что один из входов будет подключен к транзистору, размеры которого в 2 раза больше, чем размеры остальных, или один затвор останется в плавающем (*floating*) состоянии, что не рекомендуется. Получается, что в таких регулярных структурах сложнее получить нужную нагрузочную способность, чем в стандартных ячейках, но как было сказано выше, автоматизировать проектирование такой структуры гораздо легче.

Стоит отметить, что если уйти от вентиляльной структуры к заказной, то можно обеспечить нужную нагрузочную способность, сэкономив площадь и улучшив быстродействие, однако при этом усложняется задача трассировки.

Итак, имея схематическое представление схемы на транзисторном уровне, с помощью программного обеспечения можно на регулярной структуре получить топологию нужной схемы по определенным шаблонам и алгоритму синтеза.

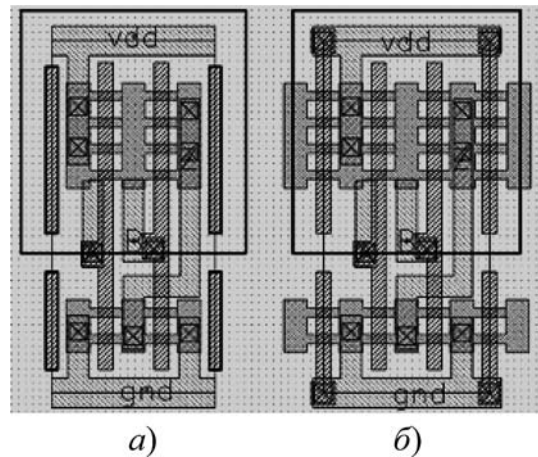


Рис. 5. Ячейка двухвходового ИЛИ-НЕ: SG-FinFET-структура стандартной ячейки (а), регулярная SG-FinFET-структура (б)

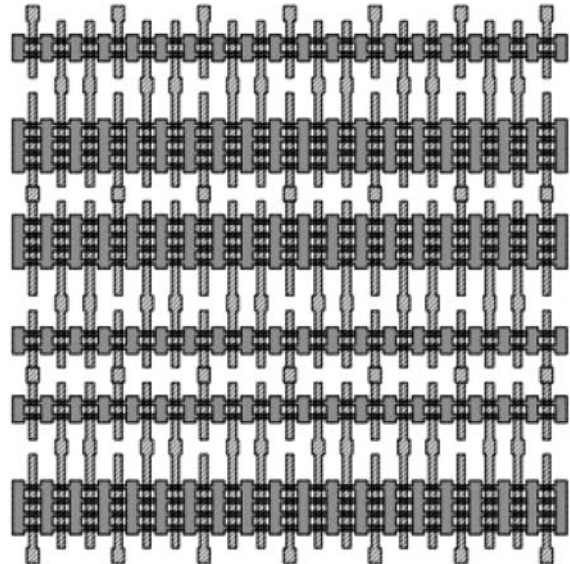


Рис. 6. Регулярная структура рядом стоящих ячеек в слоях диффузии и поликремния

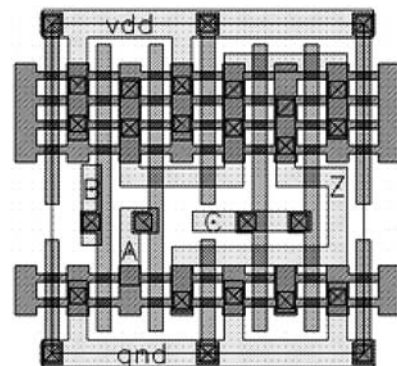


Рис. 7. Регулярная структура трехвходового И-ИЛИ-НЕ21

3.2. Метод проектирования регулярных структур на основе технологии режущих (cut) слоев

Для технологических процессов 22 нм и ниже все труднее становится получать топологические элементы заданной формы. Для решения этой проблемы увеличивают количество норм проектирования, особенно нормы проектирования усложняются для слоев поликремния и диффузии, а также для слоев металлизации.

Процесс упрощается, как уже было сказано выше, при введении регулярной топологической структуры в слоях поликремния и диффузии. Кроме этого существует подход, при котором в процессе проектирования используют особые режущие слои (*cut layers*), из которых изготавливают маски (*cut masks*). Эти маски в процессе производства "режут" слои топологии. Такой метод широко используется, начиная с технологии 20 нм и ниже.

В регулярных структурах метод режущих слоев также дает возможность регулировать синтез сложно-функциональных блоков в слоях поликремния и диффузии. Отрезав поликремний в тех местах, где это нужно, можно получить почти любую схему и без выбора шаблона.

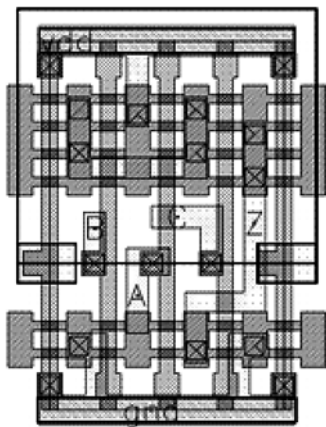


Рис. 8. Регулярная структура трехвходового И-ИЛИ-НЕ21 с применением PCUT-слоя

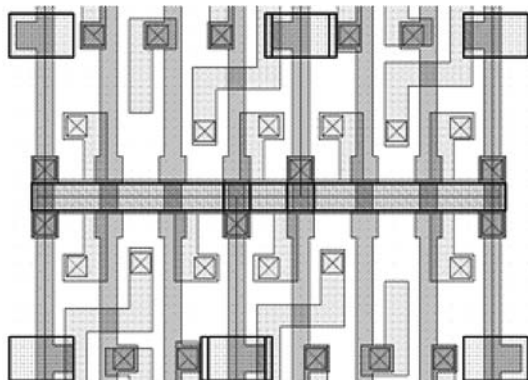


Рис. 9. Фрагмент топологии регулярных структур с применением PCUT-слоя

В данной работе разработан метод синтеза регулярной топологии с внедрением особого слоя. Этот слой назван РСИТ и применяется как аналог режущего слоя, который используется лидирующими производителями уже давно.

На рис. 8 приведена структура трехвходового И-ИЛИ-НЕ21 с применением PCUT-слоя. Можно заметить, что этот метод позволяет синтезировать любые схемы на матрице регулярных слоев поликремния и диффузии, удаляя слой поликремния там, где необходимо, тем самым изолируя функциональные затворы друг от друга (рис. 9).

По сравнению с использованием регулярных шаблонов без PCUT этот метод позволяет эффективнее размещать функциональные блоки, сокращать площадь за счет общих областей диффузии и уменьшать число изолирующих затворов. Однако алгоритм и программное обеспечение для такого синтеза топологии усложняются, а также усложняется трассировка.

4. Математические модели логических элементов

Отличительная особенность предлагаемого подхода состоит в независимом синтезе *pull-up*- и *pull-down*-цепей с использованием графа вложенности последовательно-параллельных структур (SP-граф) [14]. Это обеспечивает дополнительные степени свободы для достижения требуемых значений характеристик схемы по площади, быстродействию и потребляемой мощности по сравнению со стандартными подходами на основе predetermined библиотек элементов. На рис. 10 пример элемента XOR в двух вариантах, с комплементарными *pull-up*- и *pull-down*-цепями (рис. 10, а) и некомплементарными (рис. 10, б).

Граф вложенности последовательно-параллельных структур определяется как бинарный ориентированный упорядоченный ациклический граф $G = (V, E)$, где каждая вершина из множества V описывает логическую функцию, а каждая дуга в множестве E , исходящая из вершины, описывает ссылки на аргументы соответствующей функции. Рассматриваются функции пяти типов, а именно,

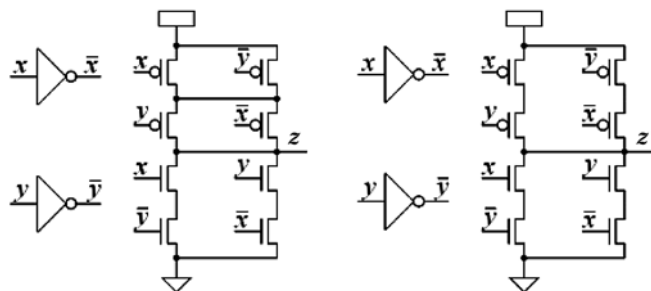


Рис. 10. Схема XOR: а — комплементарные *pull-up*- и *pull-down*-цепи; б — некомплементарные *pull-up*- и *pull-down*-цепи

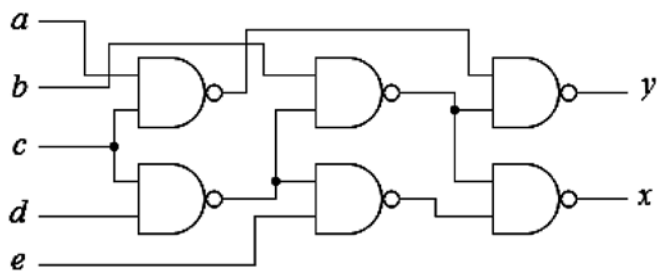


Рис. 11. Схема с17

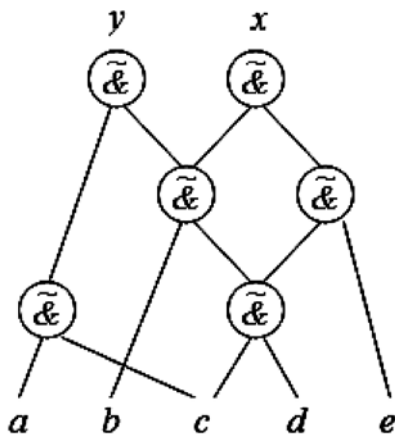


Рис. 12. SP-граф схемы с17

конъюнкция ($f(a, b) = a \& b$) для последовательного соединения внутри вентиля, дизъюнкция ($f(a, b) = a + b$) для параллельного соединения внутри вентиля, отрицание ($f(a) = \bar{a}$), функция конъюнкции с отрицанием (NAND) ($f(a, b) = \overline{a \& b}$) для последовательного соединения на выходе вентиля, функция дизъюнкции с отрицанием (NOR) ($f(a, b) = \overline{a + b}$) для параллельного соединения на выходе вентиля.

Таким образом, граф описывает композицию логических функций. Вершины соответствуют логическим функциям. Дуги описывают подстановку аргументов для функции, соответствующей вершине, из которой эти дуги исходят. Пример SP-графа схемы с17 (рис. 11) представлен на рис. 12.

Благодаря обобщенной графовой модели появляется возможность разработки эффективного алгоритма структурной оптимизации. При этом, с одной стороны, сохраняется большее число степеней свободы за счет логического аспекта (т. е. булевой формы графа), а с другой стороны, появляется нужная точность в анализе задержек, мощности и площади за счет структурной интерпретации на транзисторном уровне.

Заключение

В данной работе предложено согласованное решение проблем автоматизации проектирования схемы и топологии на основе модели компонент в форме графа вложенности последовательно-параллельных структур (SP-граф) с независимым анализом цепей земли (*pull-down*) и питания (*pull-up*). Исследованы варианты топологического представления логических элементов на FinFET-транзисторах. Разработан регулярный шаблон для отображения SP-структуры на FinFET-транзисторах.

Работа выполнена при поддержке РФФИ (проект № 13-07-00148).

Список литературы

1. Талалай М. С., Трушин К. В., Венгер О. В. Логический синтез комбинационных схем на основе транзисторных шаблонов с регулярной топологией // Информационные технологии. 2011. № 4 (176). С. 2–7.
2. Талалай М. С., Трушин К. В., Венгер О. В. Логический синтез булевой функции для проектирования интегральных схем на транзисторных шаблонах // Информационные технологии. 2012. № 6(190). С. 2–11.
3. Muttreja A., Agarwal N., Jha N. K. CMOS Logic Design with Independent-gate FinFETs // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2007. P. 560–567.
4. Meinhardt C., Reis R. FinFET Basic Cells Evaluation for Regular Layouts // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2013. P. 1–4.
5. Mishra P., Muttreja A., Jha N. K. FinFET Circuit Design // Springer Science+Business Media LLC. 2011. P. 23–54.
6. Datta A., Goel A., Cakici R. T. Modeling and Circuit Synthesis for Independently Controlled Double Gate FinFET Devices // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2007. P. 1957–1966.
7. Huang X., Lee W.-C., Kuo C., Hisamoto D. et al. Sub 50-nm FinFET: PFET // In Tech. Dig. IEDM. Washington: DC, 1999. P. 67–70.
8. Collinge J. P. (Ed.) FinFETs and Other Multi-Gate Transistors. New York: Springer. 2008.
9. Liu Y. X., Matsukawa T., Endo K., Masahara M. et al. Cointegration of high-performance tied-gate three-terminal FinFETs and variable threshold-voltage independent-gate four-terminal FinFETs with asymmetric gate-oxide thicknesses // IEEE Electron Device Lett. Jun. 2007. Vol. 28, N 6. P. 517–519.
10. Agostinelli M., Alioto M., Esseni D., Selmi L. Leakage-delay tradeoff in FinFET logic circuits: A comparative analysis with bulk technology // IEEE Trans. Very Large Scale Integr. (VLSI) Syst. Feb. 2010. Vol. 18, N 2. P. 232–245.
11. Tawfik S. A., Kursun V. Low-power and compact sequential circuits with independent-gate FinFETs // IEEE Trans. Electron Devices. Jan. 2008. Vol. 55, N 1. P. 60–70.
12. Cakici R. T., Roy K. Analysis of options in double-gate MOS technology: A circuit perspective // IEEE Trans. Electron Devices. Dec. 2007. Vol. 54, N 12. P. 3361–3368.
13. Anil K. G., Henson K., Biesemans S., Collaert N. Layout density analysis of FinFET // Proc. ESSDERC, 2003. P. 139–142.
14. Гаврилов С. В., Гудкова О. Н., Щелоков А. Н. Логико-временной анализ нанометровых схем на основе интервального подхода // Известия ЮФУ. Технические науки. 2012. № 7(132). С. 85–91.

S. V. Gavrilov, Head of Department, G. A. Ivanova, Junior Researcher,
Institute for Design Problem in Microelectronics of Russian Academy of Science,
A. A. Manukyan, Graduate Student, National Research University of Electronic Technology, Moscow

New Problems of Logic-Topological Synthesis of Custom IP-Blocks and Methods for Their Solution

This article is devoted to solving the problems of computer aided design of custom IC with technological norms 22 nm and below. Particular attention is paid to a new technology called FinFET. It is known that FinFET technology provides improving performance and power compared to a standard CMOS technology. However, wide dissemination FinFET technology is difficult because of problem of automation design circuit and topology, development of computer-aided design of integrated circuits lags behind the development of technology. To solve the problems computer aided design, this article provides a comparative analysis of existing approaches topology development FinFET structures, as well as a comparative analysis of these structures with conventional CMOS structure. Developed methods of formation of topological structures of various FinFET structures for topology synthesis of elements with a regular topological structure in layers of polysilicon and diffusion.

Keywords: CAD (computer-aided design), SP-DAG, Intellective property IP-block, finFET transistor, CMOS technology

References

1. Talalay M. S., Trushin K. V., Venger O. V. Logicheskiy sintez kombinatsionnykh skhem na osnove tranzistornykh shablonov s regulirnoy topologiyey. *Informatsionnye tehnologii*. 2011. N 4(176). P. 2–7.
2. Talalay M. S., Trushin K. V., Venger O. V. Logicheskiy sintez bulevoy funktsii dlya proektirovaniya integralnykh shem na tranzistornykh shablonah. *Informatsionnye tehnologii*. 2012. N 6(190). P. 2–11.
3. Muttreja A., Agarwal N., Jha N. K. CMOS Logic Design with Independent-gate FinFETs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2007. P. 560–567.
4. Meinhardt C., Reis R. FinFET Basic Cells Evaluation for Regular Layouts. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2013. P. 1–4.
5. Mishra P., Muttreja A., Jha N. K. FinFET Circuit Design. *Springer Science+Business Media LLC*. 2011. P. 23–54.
6. Datta A., Goel A., Cakici R. T. Modeling and Circuit Synthesis for Independently Controlled Double Gate FinFET Devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2007. P. 1957–1966.
7. Huang X., Lee W.-C., Kuo C., Hisamoto D. et al. Sub 50-nm FinFET: PFET. *In Tech. Dig. IEDM*. Washington: DC, 1999. P. 67–70.
8. Collinge J. P. (Ed.) FinFETs and Other Multi-Gate Transistors. New York: Springer. 2008.
9. Liu Y. X., Matsukawa T., Endo K., Masahara M. et al. Cointegration of high-performance tied-gate three-terminal FinFETs and variable threshold-voltage independent-gate four-terminal FinFETs with asymmetric gate-oxide thicknesses. *IEEE Electron Device Lett.* Jun. 2007. Vol. 28, N 6. P. 517–519.
10. Agostinelli M., Alioto M., Esseni D., Selmi L. Leakage-delay tradeoff in FinFET logic circuits: A comparative analysis with bulk technology. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* Feb. 2010. Vol. 18, N 2. P. 232–245.
11. Tawfik S. A., Kursun V. Low-power and compact sequential circuits with independent-gate FinFETs. *IEEE Trans. Electron Devices*. Jan. 2008. Vol. 55, N 1. P. 60–70.
12. Cakici R. T., Roy K. Analysis of options in double-gate MOS technology: A circuit perspective. *IEEE Trans. Electron Devices*. Dec. 2007. Vol. 54, N 12. P. 3361–3368.
13. Anil K. G., Henson K., Biesemans S., Collaert N. Layout density analysis of FinFET. *In Proc. ESSDERC*, 2003. P. 139–142.
14. Gavrilov S. V., Gudkova O. N., Schelokov A. N. Logiko-vremennyy analiz nanometrovykh shem na osnove intervalnogo podhoda. *Izvestiya UFU. Tehnicheskie nauki*. 2012. N 7(132). P. 85–91.

УДК 621.372.061

В. В. Ракитин^{1, 2}, канд. техн. наук, вед. науч. сотр., e-mail: rakitin@niifp,
С. Г. Русаков, д-р техн. наук, чл.-корр. РАН, гл. науч. сотр., e-mail: rusakov@ippm.ru

¹ Институт проблем проектирования в микроэлектронике РАН

² ФГУП "НИИ Физических Проблем им. Ф. В. Лукина"

Электрические модели мемристормых элементов

Обсуждаются принципы создания моделей мемристоров для схемотехнического моделирования. Мемристор — новый схемный двухполюсный элемент, который можно рассматривать как резистор, обладающий памятью. Для описания этого свойства в модель мемристора необходимо ввести дополнительное дифференциальное уравнение. Наличие внутреннего дифференциального уравнения является особенностью мемристормой модели с точки зрения схемного моделирования. Представлены основные известные подходы к созданию моделей мемристоров. Рассмотрены линейно-дрейфовая и нелинейные модели. Уравнения моделей могут рассматриваться как основа библиотечных моделей для стандартных программ схемотехнического моделирования. Применение предлагаемых моделей иллюстрируется примерами моделирования различных режимов электрических схем с использованием мемристормых элементов.

Ключевые слова: схемотехническое моделирование, мемристор, электрические модели, моделирование мемристора, оконная функция

Введение

Мемристором называют пассивный двухполюсник, сопротивление которого зависит от протекающего через него заряда. Теория мемристора была создана в 1971 г. профессором Л. Чуа [1]. Он рассматривал мемристор как новый элемент, дополняющий традиционные элементы: резистор, конденсатор и индуктивность [1]. Теория предполагает возможность существования элемента, в котором взаимосвязаны магнитный поток и накопленный заряд. Одним из следствий такой взаимосвязи является то, что мемристор обладает сопротивлением, значение которого определяется интегралом во времени протекающего через него тока, иначе говоря, имеет свойство памяти.

Состояние мемристора w изменяется под действием тока i или приложенного напряжения v . В первом случае имеем мемристор, управляемый током, во втором — мемристор, управляемый напряжением. По определению мемристор, управляемый током, описывается следующими соотношениями:

$$v = R(w, i)i; \quad (1)$$

$$\frac{dw}{dt} = F(w, i). \quad (2)$$

Мемристор, управляемый напряжением, описывается системой

$$i = G(w, v)v; \quad (3)$$

$$\frac{dw}{dt} = F(w, v). \quad (4)$$

Функция $R(w, i)$ имеет размерность сопротивления и называется *мемристаносом*. Функция $G(w, v)$ имеет размерность проводимости и называется *мемкондактансом*. Функция $F(w, v)$ при безразмерной переменной w имеет размерность, обратную временной переменной, и определяет скорость изменения состояния. Отметим, что переменная состояния w ограничена ($0 \leq w \leq 1$), что всегда можно обеспечить за счет нормировки.

Формулы (1) и (3) отражают мгновенную вольт-амперную характеристику (ВАХ) мемристора, а уравнения (2) и (4) — управляющую характеристику (УХ) мемристора. Следует отметить, что ВАХ мемристора всегда проходит через начало координат. Состояние мемристора $w(t)$ является решением уравнений (1)—(2) или (3)—(4) при начальном состоянии мемристора $w(0)$.

Приборы со свойствами резистивной памяти известны уже давно. К ним относятся, например, приборы на базе электрохимических элементов. На протяжении ряда лет ведутся работы по созданию резистивных элементов памяти (RRAM), которые также можно отнести к мемристорам. Исторически первые приборы с изменяемым под действием тока сопротивлением применялись еще в середине

прошлого столетия. Так, в 1960 г. был предложен трехэлектродный электрохимический прибор — мемристор, в котором сопротивление изменялось при протекании тока через третий электрод [2]. Аналогичные работы по созданию адаптивных систем на мемристорах проводились в России [3]. Предпринимались попытки создания твердотельных приборов с изменяемым сопротивлением [4]. Было показано наличие гистерезиса в структурах металл—изолятор—металл [5].

Внимание к мемристорам значительно выросло в 2008 г. после появления публикаций группы сотрудников (Д. Струков и др.) из лаборатории HP Labs, руководимой доктором S. Williams [6]. Им удалось создать мемристоры на основе тонких пленок оксида титана, содержащих кислородные вакансии. Экспериментальные образцы приборов были выполнены по современной полупроводниковой технологии и имели нанометровые размеры. На основе полученных результатов было предсказано появление в ближайшем времени коммерческих приборов на мемристорах. Хотя этот прогноз пока не подтвердился, число публикаций по мемристорам стало быстро возрастать. Мемристоры еще находятся на стадии экспериментальных исследований, но успехи в технологии их изготовления и понимании физических процессов дают основания надеяться на появление на рынке изделий на их основе.

Областями применения мемристоров могут быть как различные системы хранения информации, так и специальные схемы, обладающие возможностями удобной регулировки, самонастройки, оптимизации параметров и т. п. [7]. Важным положительным обстоятельством является совместимость технологии мемристоров с кремниевой КМОП-технологией. Причем сами мемристоры могут изготавливаться на завершающей стадии изготовления КМОП ИС в виде специальных контактов между слоями металлических проводников (crossbar-матрицы). Подобная гибридная КМОП-технология с мемристивными 3D-матрицами по оценкам способна обеспечить терабитную плотность хранения информации, малое потребление энергии, высокое быстродействие, возможность динамического конфигурирования архитектуры систем хранения и обработки информации.

Очень перспективным является потенциальное применение мемристоров в нейроморфных системах типа нейронных сетей, в которых они обеспечивают возможность простой перестройки синаптических (весовых) коэффициентов [8]. Интересным является уже продемонстрированная возможность создания мемристорных систем на органических и полимерных материалах [9].

К настоящему времени разработано достаточно много моделей мемристоров, сравнение ряда из них приведено в обзорах [10, 11]. Тип модели определяется способом конкретизации функций $R(w, i)$

или $G(w, v)$ и $F(w, v)$ с требованиями соответствия реальным экспериментальным данным. К моделям предъявляются противоречивые требования: точность и вычислительная эффективность, простота и физическая обоснованность, наглядность. Желательно также, чтобы они подходили для широкого класса мемристоров. Особенности функций $R(w, i)$, $G(w, v)$ и $F(w, v)$ определяют следующие свойства моделей мемристоров: нелинейность, асимметрия, пороговые свойства.

Ниже рассмотрены основные принципы построения электрических моделей мемристорных элементов для систем схемотехнического моделирования. Показана возможность применения таких моделей для моделирования схем, содержащих мемристоры.

Линейно-дрейфовая модель мемристора

Прибор, изготовленный сотрудниками HP Labs [6], представлял собой тонкую пленку оксида титана (толщиной D примерно 5 нм) между двумя платиновыми электродами (рис. 1, а). Он содержит два слоя, один из них обогащен вакансиями кислорода (легированный вакансиями слой TiO_x), что увеличивает его проводимость, а другой слой (TiO_2) имеет низкую проводимость. Общее сопротивление прибора зависит от суммы сопротивлений слоев и в первом приближении определяется положением

границы между ними. При протекании тока от легированного слоя к нелегированному возрастает толщина легированного слоя W , и общее сопротивление уменьшается. При обратном направлении тока толщина легированного слоя и общая проводимость уменьшаются. Далее в качестве безразмерной переменной состояния мемристора выбирается нормализованная толщина легированного слоя $w = W/D$ (см. рис. 1, а).

Была предложена простая линейно-дрейфовая модель поведения мемристора, управляемого током:

$$v = (R_{ON}w + R_{OFF}(1 - w))i; \quad (5)$$

$$\frac{dw}{dt} = \alpha i, \quad (6)$$

где R_{ON} , R_{OFF} , α — минимальное сопротивление, максимальное сопротивление и скорость изменения состояния мемристора соответственно.

В такой модели при постоянном управляющем сигнале временное изменение состояния (дрейф границы раздела) имеет линейный характер. При протекании постоянного положительного (отрицательного) тока сопротивление мемристора R_{ON} (R_{OFF}) будет изменяться, пока не достигнет предельного значения.

Обычно при измерениях характеристик мемристоров на них подается периодическое напряжение (синусоидальное или треугольное) с нулевой постоянной составляющей. В этом случае решением системы уравнений в координатах v и i (на фазовой плоскости) является некоторая фигура (типа фигуры Лиссажу), периодически проходящая через начало координат. Через некоторое время, определяемое соотношением частоты ω внешнего сигнала и параметра α , процесс на фазовой плоскости придет к стационарной замкнутой кривой. Эта кривая может быть названа гистерезисной ВАХ мемристора или просто петлей гистерезиса мемристора (рис. 1, б). Петля гистерезиса сужается при возрастании частоты входного периодического сигнала, т. е. на высокой частоте мемристор ведет себя как сопротивление. На малой частоте петля гистерезиса вырождается в ломаную линию, соответствующую R_{ON} при положительном входном напряжении и R_{OFF} при отрицательном (прямые линии на рис. 1, б). При подаче на мемристор однополярного сигнала его состояние и сопротивление изменяются монотонно. Принято процесс увеличения проводимости мемристора называть *записью*, уменьшения проводимости — *стиранием*, а определение его состояния (желательно без разрушения) — *считыванием*.

Нелинейные модели мемристоров

Отметим, что обычно удобно рассматривать функции $R(w, i)$ и $F(w, i)$ как функции одной переменной, а сами функции выбирать из числа стандартных элементарных функций: кусочно-линей-

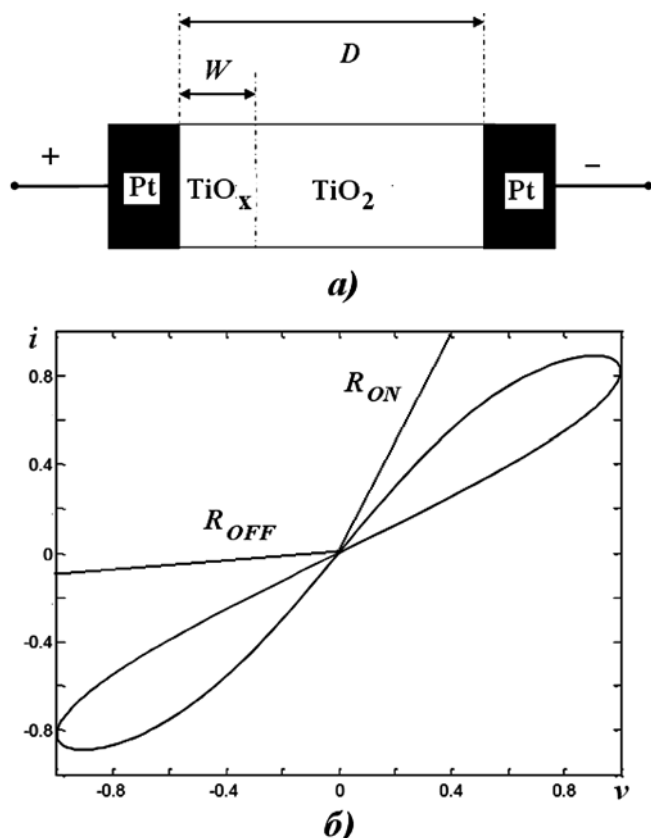


Рис. 1. Структура (а) и гистерезисная ВАХ мемристора (б)

ных, степенных и гиперболических. Совпадение с экспериментальными данными достигается подгонкой соответствующих параметров.

В большинстве случаев внутреннее состояние описывается одной переменной w , которая часто интерпретируется как относительная координата границы раздела областей с различными проводимостями. Реальные приборные ситуации гораздо сложнее, но описание их моделей выходит за рамки настоящего рассмотрения.

Линейно-дрейфовая модель допускает неограниченное изменение состояния w , которое в действительности должно быть ограничено: $0 \leq w \leq 1$. Это достигается введением оконной функции $f(w)$, которая обнуляет скорость изменения состояния на краях диапазона.

В результате уравнение (6) преобразуется к виду

$$\frac{dw}{dt} = \alpha f(w)i. \quad (7)$$

Оконная функция может симметрично ограничивать скорость изменения состояния на краях диапазона [12] (рис. 2, а):

$$f(w) = 1 - (2w - 1)^{2p}, \quad (8)$$

где p — подгоночный параметр.

Ограничение может зависеть от направления изменения состояния [13] (рис. 2, б):

$$f(w) = 1 - [(w - \text{stp}(-i))^{2p}], \quad (9)$$

где $\text{stp}(i)$ — ступенчатая функция (0, если $i \leq 0$, и 1, если $i \geq 0$).

На рис. 3 изображено влияние оконной функции на гистерезисную ВАХ линейно-дрейфового мемристора [11].

С помощью оконной функции можно существенно модифицировать вид гистерезисной ВАХ. Вместе с тем, этот подход не вполне отражает такие нелинейные свойства УХ и самой ВАХ, как их пороговые характеристики.

Отдельный класс моделей построен непосредственно для описания конкретных приборов и конкретных режимов работы. Такие модели позволяют получать высокую точность, но имеют узкое применение. Так, для мемристоров из халькогенидов серебра, имеющих гистерезисную ВАХ (рис. 4, а) предложена модель [14]

$$i = \alpha wv;$$

$$\frac{dw}{dt} = K^* \exp(k^*(v - v_{th}^*(f(w))),$$

где K^* , k^* , v_{th}^* — коэффициенты, а (*) указывает, что значение коэффициента зависит от полярности v . Подбором этих коэффициентов можно получить гистерезисную ВАХ (рис. 4, б), близкую к экспериментальной.

В качестве другого примера сложной модели конкретного прибора можно привести уточненную модель вышеописанного мемристора из оксида титана. Модель дает описание проводимости прибора за счет наличия высокопроводящих каналов, отделенных от электрода туннельно-тонким слоем. В этом случае зависимость тока от напряжения имеет экспоненциальный характер, а изменение проводимости происходит при сравнительно небольших изменениях толщины туннельного слоя. Формулы ВАХ и УХ имеют следующий вид [15]:

$$i = \frac{l}{w^2} [\varphi_i \exp(-m\sqrt{\varphi_i}) - (\varphi_i + |v|) \exp(-m\sqrt{(\varphi_i + |v|)})];$$

$$\frac{dw}{dt} = n^* \sinh(k^*|i/i_0^*|) f^*(w),$$

где $\varphi_i = \varphi_0 - |v|k(w)$; l, m, n, i_0, φ_0 — подгоночные параметры.

Сравнение экспериментальных и расчетных значений приведено на рис. 5 [11].

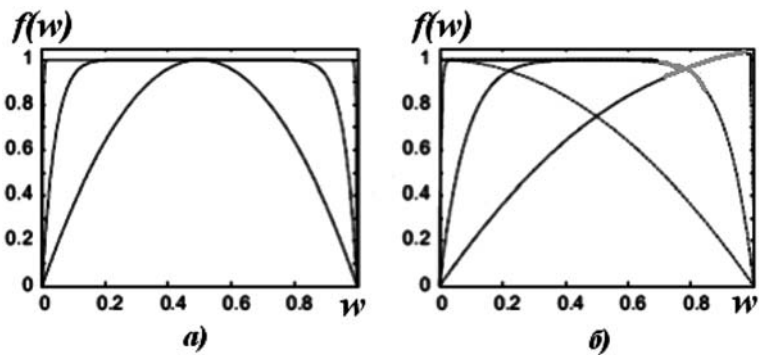


Рис. 2. Графики оконных функций [11]: а — формула (8); б — формула (9)

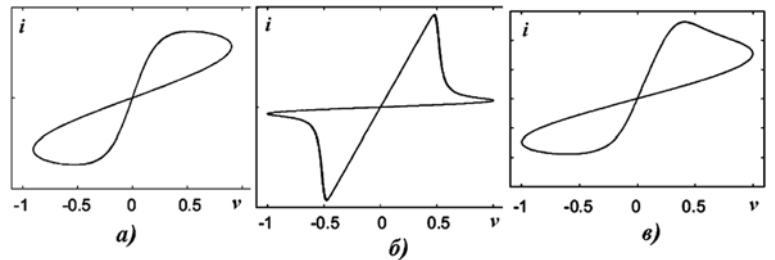


Рис. 3. Возможности вариации гистерезисной ВАХ за счет оконной функции [11]

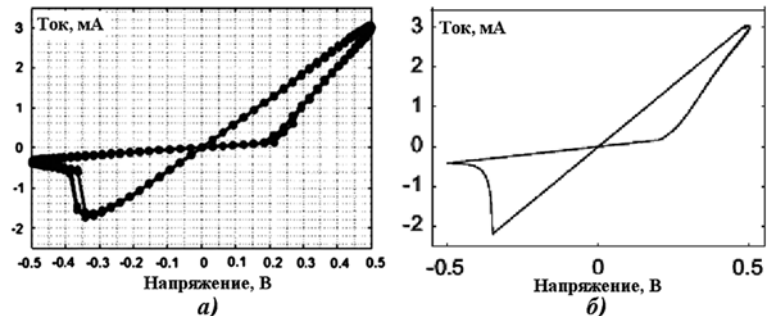


Рис. 4. Экспериментальная (а) и расчетная (б) ВАХ мемристора [11]

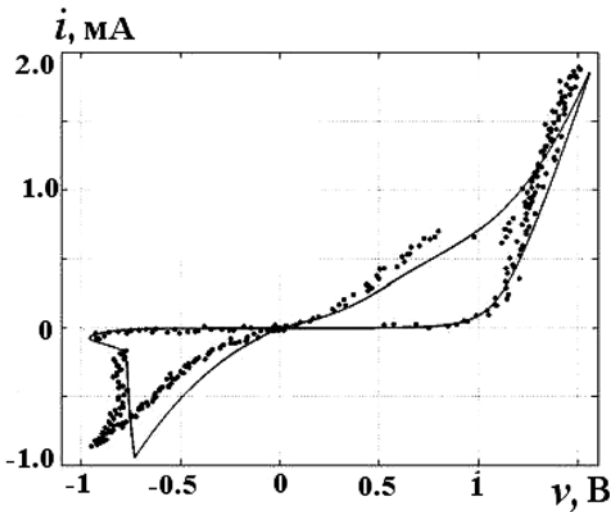


Рис. 5. Сравнение экспериментальных данных (точки) с результатами расчетов [11]

Использование подобных моделей позволяет подгонять результаты моделирования к экспериментальным данным с большой точностью. Однако их применение в качестве универсальных для различных типов мемристоров ограничено в связи с высокой специализацией. Вместе с тем, использование экспоненциальных или гиперболических функций в универсальных моделях является оправданным [16–19]. Во-первых, они соответствуют физическим процессам в приборах, описывая туннельную составляющую тока. Во-вторых, такими средствами удобно вводить нелинейность ВАХ и УХ и обеспечивать им пороговый характер.

Достаточно общая система уравнений для мемристоров, управляемых напряжением (sinh-модель), имеет вид

$$i = aw \sinh(bv) \theta(v); \quad (10)$$

$$\frac{dw}{dt} = c \sinh(dv) f(w), \quad (11)$$

где $\theta(i)$ — ограничивающая функция; $f(w)$ — оконная функция; a, b, c, d — параметры настройки модели. Набор параметров (a, b, c, d) может быть дополнен зависимостью от знака напряжения v . Конкретизация значений набора параметров позволяет получать достаточно широкий спектр гистерезисных кривых, как это показано на рис. 6 [18].

Как следует из формул (10), (11) и рис. 6, в данной модели в качестве входной переменной выбрано напряжение. Зависимость тока от напряжения в модели нелинейна, а от внутренней переменной w — линейна. В свою очередь, изменение w зависит только от напряжения на мемристоре. Конкретная гистерезисная ВАХ мемристора, описываемого sinh-моделью (10)–(11), полученная при синусоидальном сигнале с частотой 2 МГц и амплитудой 2 В, приведена на рис. 7 [18]. Развитие этой модели, отражающее вклад нескольких процессов, приведено в работе [19]:

$$i = aw \sinh(bw) + \gamma(1 - w)(1 - \exp(\beta v));$$

$$\frac{dw}{dt} = c \sinh(dv) - w/\tau,$$

где γ, β, τ — подгоночные коэффициенты.

Второе слагаемое в формуле для ВАХ наряду с вкладом туннельного тока учитывает вклад тока через барьер Шоттки, образованный металлическим электродом и мемристивным слоем. В формуле для УХ, наряду с диффузионно-полевой составляющей, учтена деградиционная составляющая (второе слагаемое с отрицательным знаком).

Более сложные модели, описанные, например, в работе [14], вводят асимметрию в формулу ВАХ типа (9) и пороговую зависимость в управляющую характеристику типа (6) с добавлением асимметричной экспоненциальной зависимости от w . Они позволяют подгонять экспериментальные данные под результаты моделирования, но содержат очень много параметров и не обладают наглядностью.

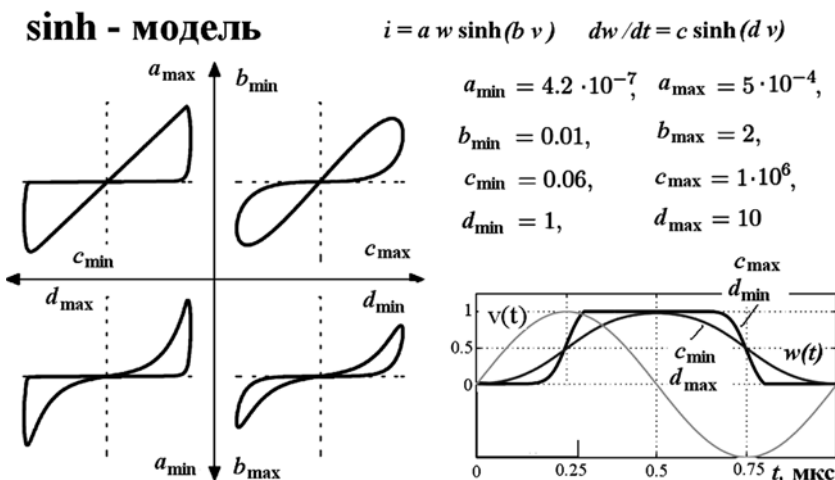


Рис. 6. Модель мемристора на базе описания функцией гиперболического синуса [18]

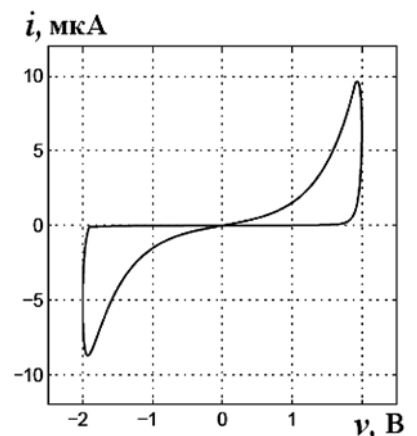


Рис. 7. Гистерезис характеристики мемристора в sinh-модели [18]

В ряде работ, например [20], используются упрощенные модели. Для мемристора, управляемого током, в этом случае введены следующие эмпирические формулы:

$$v = R_{ON} \exp(\lambda w) i;$$

$$\frac{dw}{dt} = k^* (|i|/i_0^* - 1) \gamma^*,$$

где константы k^* , i_0^* , γ^* принимают различные значения в зависимости от полярности тока.

Такие модели позволяют добиться приемлемой точности при небольшом числе введенных параметров.

В целом можно констатировать, что предложенные модели мемристоров обеспечивают широкий диапазон требований по полноте описания их поведения и возможностям согласования результатов расчетов с экспериментальными данными. Их многообразие соответствует и обусловлено разнообразием физических механизмов и технологических вариантов нового перспективного класса приборов.

Все вышеописанные модели легко вводятся в системы схемотехнического моделирования как подсхемы или формируются из стандартных элементов по имеющимся правилам. Кривые, приведенные на рис. 3–5, получены с помощью SPICE-подобного симулятора. Именно на эту систему ориентируются разработчики моделей [21–27]. Имеются отдельные публикации по моделированию мемристоров в системе Verilog-A [28] и в системе MATLAB + SIMULINK [29].

Примеры моделирования мемристоров

Система MATLAB+SIMULINK является удобным средством проверки моделей и инструментом анализа характеристик мемристоров и систем на их основе. Ниже приведем примеры применения моделей мемристоров.

Модель мемристора, реализующая формулы (3) и (4), может быть представлена интегратором с ограниченным диапазоном изменения величины w ($0 \leq w \leq 1$) и умножителем (рис. 8). Амплитуда входного сигнала мемристора умножается на постоянный коэффициент $\alpha = 0,5$, задающий скорость изменения состояния, и интегрируется, а выходной сигнал интегратора умножается на входную амплитуду. В данной работе величина α порядка единицы, что соответствует диапазону частот в герцах. У реальных мемристоров этот диапазон лежит в интервале

до сотен мегагерц, а величина α в соответствующее число раз больше. Интегратор с ограниченным диапазоном выполняет роль оконной функции (формулы (7) или (8) при $p \rightarrow \infty$).

При синусоидальном входном сигнале и линейности интегратора выходной сигнал мемристора содержит синусоиду исходной и удвоенной частот, что проявляется в виде гистерезисной кривой вида классической фигуры Лиссажу (рис. 8, з).

При воздействии на мемристор гармонического сигнала возрастающей частоты (рис. 9) петля гистерезиса сужается (рис. 9, з).

На рис. 10 показана структурная схема модели мемристора с гиперболическими операторами, в частности sinh-модели мемристора с характеристиками типа гиперболического синуса. Поведение sinh-модели при коэффициентах порядка единицы качественно не отличается от линейно-дрейфовой модели (рис. 11). В данном случае использовалась последовательность треугольных импульсов с одинаковым нарастанием и спадом. При медленном

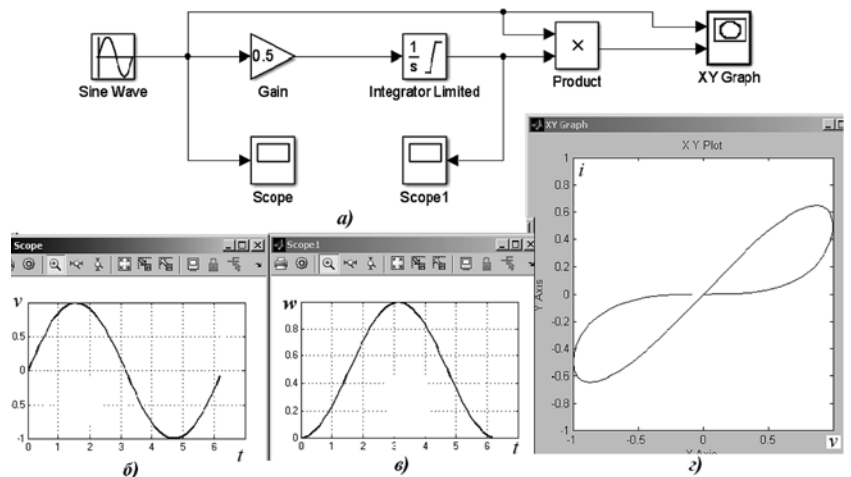


Рис. 8. Структура модели мемристора и результат моделирования гистерезиса

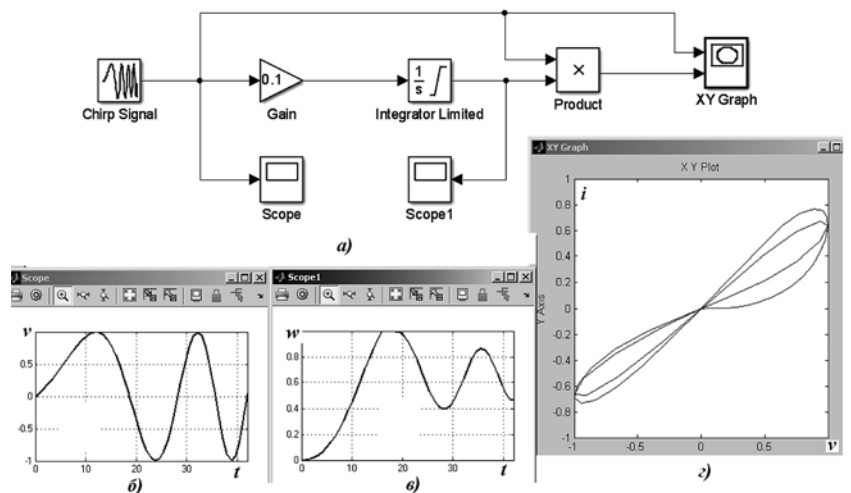


Рис. 9. Моделирование мемристора при частотно-модулированном сигнале: а — общая схема; б — входной сигнал; в — сигнал состояния мемристора; г — петля гистерезиса

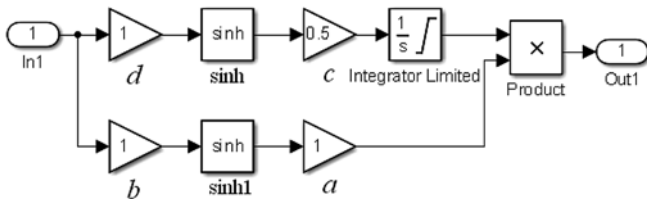


Рис. 10. Sinh-модель мемристора в SIMULINK

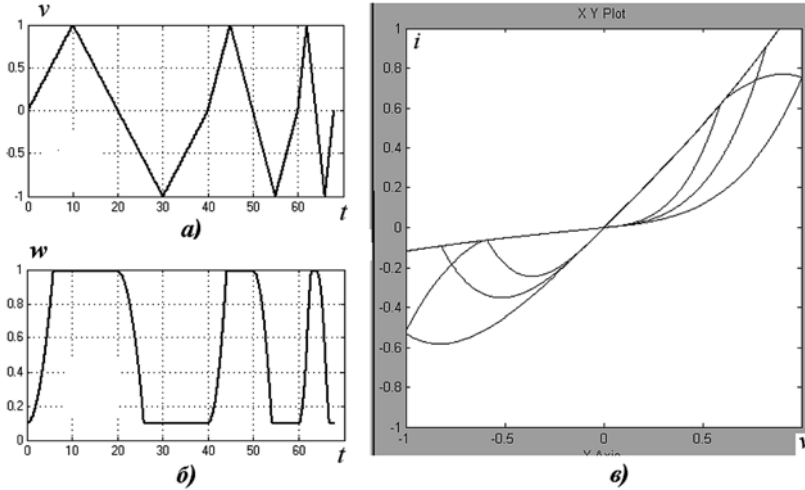


Рис. 11. Результаты моделирования при частотно-модулированном сигнале: а — форма сигнала; б — изменение проводимости; в — гистерезисная ВАХ

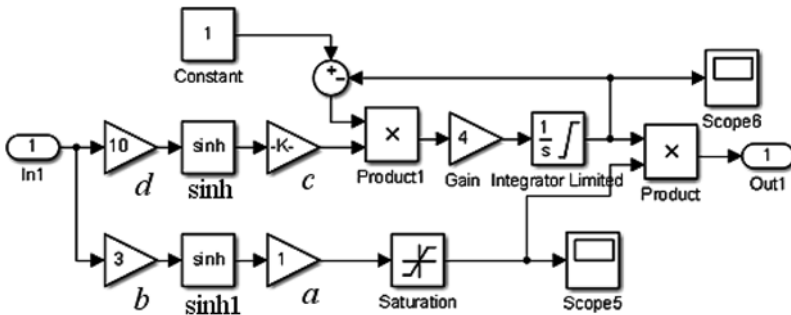


Рис. 12. Sinh-модель мемристора с ограничителями в SIMULINK

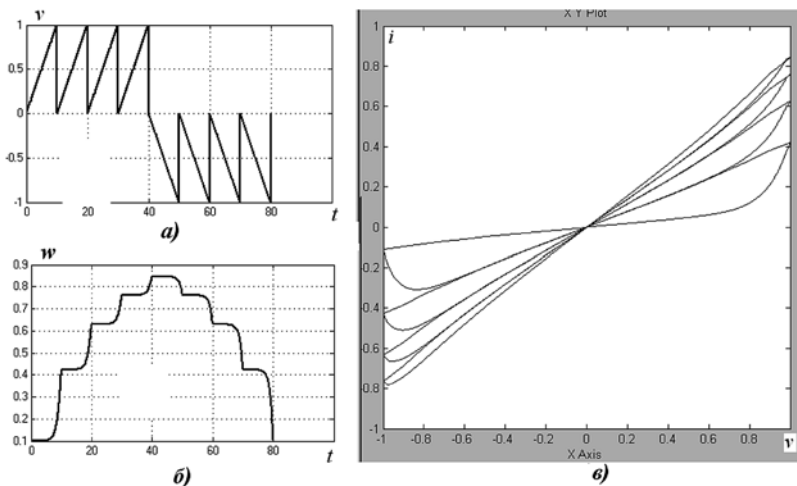


Рис. 13. Результаты моделирования при нестационарном сигнале модели (рис. 12) ($b = 0,9$, $c = 10^{-5}$): а — форма сигнала; б — изменение проводимости; в — гистерезисная ВАХ

нарастании состояние мемристора достигает критических значений (диапазон w задан от 0,1 до 1) и гистерезисная кривая вырождается в прямую линию. Повышение частоты сохраняет состояние мемристора в линейной области и увеличивает область гистерезиса с последующим ее сужением на высоких частотах.

Использование нелинейных свойств гиперболических функций при больших значениях коэффициентов b и d в системе (10)—(11) требует введения оконной функции в УХ и ограничений в ВАХ (рис. 12). Как следует из результатов моделирования (рис. 13), поведение мемристора становится более сложным. В данном случае на мемристор подается последовательность несимметричных треугольных однополярных импульсов. Под их воздействием при положительных сигналах идет постепенное нарастание тока при росте проводимости, а при отрицательных сигналах наблюдается противоположный эффект. Существенно влияние нелинейностей ВАХ и нелинейностей УХ. Ток на ВАХ изменяется линейно, пока напряжение не достигает пороговой величины (порядка 0,7 В). Аналогично для УХ — проводимость не изменяется, пока напряжение не достигает некоторого порога.

Пороговые свойства ВАХ и УХ мемристора, описываемые sinh-моделью, позволяют упростить конструкцию матрицы памяти на их основе. В таком случае элемент памяти из одного мемристора (рис. 14) обеспечивает возможность простого управления матрицей, а также изоляцию ее элементов. Мемристорный элемент памяти работает следующим образом. В режиме хранения на шинах матрицы удерживаются нулевые потенциалы, и информация, накопленная в мемристоре, сохраняется. Для осуществления неразрушающего считывания на словарной шине необходимо сформировать импульс считывания, меньший по ам-



Рис. 14. Конструкция мемристорного элемента матрицы памяти

плитуде порога УХ. Для записи/стирания информации на словарной шине формируется положительный/отрицательный импульс, превышающий порог (записи/стирания). Для блокирования записи в выбранный элемент нужно подать на его разрядную шину положительный импульс, снижающий суммарный управляющий сигнал на мемристорном элементе.

При моделировании описываемого элемента (рис. 15, а) на мемристор подается последовательность импульсов управления (рис. 15, б) с амплитудами: считывание 0,5 В, запись 1 В, считывание 0,5 В, стирание — 1 В, считывание — 0,5 В. Как показано на рис. 15, в, под действием импульса записи состояние мемристора w (и его проводимость) становится максимальным (примерно 0,9), а под действием импульса стирания достигает минимума (0,1). При этом ток, протекающий через мемристор, во время импульсов считывания (рис. 15, г) соответствует состоянию мемристора (максимален после записи и минимален после стирания). Таким образом, из результатов моделирования, представленных на рис. 15, следует, что состояние мемристора и ток считывания соответствуют алгоритму работы элемента памяти. Вместе с тем, можно отметить небольшой дрейф состояния мемристора (рис. 15, д) под действием сигналов считывания, что свидетельствует о неидеальности порога УХ рассматриваемого элемента.

Заключение

Отличительной особенностью моделей мемристорных элементов в качестве компонентов систем схемотехнического моделирования является собственное дифференциальное уравнение, вызванное необходимостью введения в модель свойств памяти. Приведенные выше модельные соотношения могут рассматриваться как базовые для эквивалентных схем, включаемых в библиотеки моделей традиционных систем схемотехнического моделирования с дальнейшим применением в инженерных экспериментальных исследованиях и разработках нового класса схем.

Авторы выражают благодарность профессору Иванникову Александру Дмитриевичу за полезные замечания.

Список литературы

1. Chua L. (1971) Memristor — The missing circuit element // IEEE Trans. Circuit Theory. 1971. V. 18, N 5. P. 507—519.
2. Widrow B. Adaptive ADALINE neuron using chemical memristors. Stanford Electron. Lab., Stanford, CA, Tech. Rep. 1553-2, 1960.

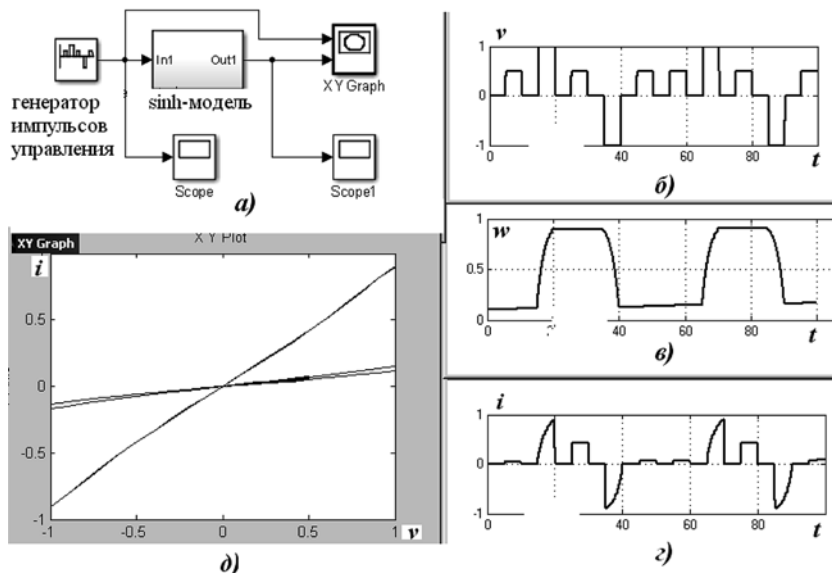


Рис. 15. Моделирование работы мемристора в режиме элемента памяти: а — схема; б — входной сигнал; в — состояние мемристора; г — выходной ток; д — фазовый портрет

3. Айрапетянц С., Ракитин В. и др. Бионические аспекты микроэлектроники // Электронная техника. Сер. "Микроэлектроника". 1967. Вып. 5. С. 15.
4. Gibbons J. et al. Switching properties of thin NiO films // Solid-State Electron. 1964. V. 7. P. 785—790.
5. Simmons J. Generalized formula for the electric tunnel effect between similar electrodes separated by a thin insulating film // J. Appl. Phys. 1963. V. 34, N. 6. P. 1793—1803.
6. Strukov D. et al. The missing memristor found // Nature. 2008. V. 453. P. 80—83.
7. Mazumder P. et al. Memristors: Devices, Models and Applications // Proc. IEEE 2012. V. 100, N. 6. P. 1911.
8. Likharev K. CrossNets: Neuromorphic Hybrid CMOS/Nano-electronic Networks // Science of Advanced Materials. 2011. V. 3. P. 322.
9. Kim S. et al. Flexible Memristive Memory Array on Plastic Substrates // Nano Lett. 2011. V. 11. P. 5438—5442.
10. Eshraghian K. et al. Memristive Device Fundamentals and Modeling: Applications to Circuits and Systems Simulation // Proceedings of the IEEE. 2012. V. 100, N. 6. P. 1911—1916.
11. Yakopcic C. et al. Memristor SPICE Modeling // Advanced in Neuromorphic Memristor Science and Application. 2012. P. 211—244.
12. Joglekar Y. et al. The elusive memristor: properties of basic electrical circuits // Eur. J. Phys. 2009. 30(661).
13. Bielek Z. et al. Spice model of memristor with nonlinear dopant drift // Radioengineering. 2009. V. 18(2). P. 210—214.
14. Pino et al. A Compact method for modeling and simulation of memristor devices: ion conductor chalcogenide-based memristor devices // IEEE/ACM international symposium on nanoscale architectures. 2010. P. 1—4.
15. Abdalla H. et al. SPICE Modeling of Memristors // ISCAS. 2011. P. 1832—1835.
16. Yakopcic C. et al. A memristor device model // IEEE Electron Dev Lett. 2011. V. 32(10). P. 1436—1438.
17. Laiho M. et al. Cellular nanoscale network cell with memristors for local implication logic and synapses // Proc. of ISCAS. 2010. P. 2051—2054.
18. Lehtonen E. Memristive Computing. dissertation UNIVERSITY OF TURKU. 2012. URL: <http://www.doria.fi/bitstream/handle/10024/79925/AnnalesA14461LehtonenDISS.pdf?sequence=4>
19. Chang T. et al. Synaptic behaviors and modeling of a metal oxide memristor device // Appl. Phys. A. 2011. V. 102. P. 857—863.
20. Kvatinisky S. et al. TEAM: ThrEshold Adaptive Memristor Model // IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS. 2012. V. 60, N. 1. P. 211—221.

21. **Benderli S.** et al. On SPICE macromodelling of TiO₂ memristor // *Electron. Lett.* 2009. V. 45, N 7. P. 377–379.

22. **Biolek D.** et al. PSPICE modeling of meminductor // *Analog Integrated Circuits and Signal Processing.* January 2011. V. 66, N. 1. P. 129–137.

23. **Biolek D.** et al. SPICE Modeling of Memcapacitor // *Electronics Letters.* 2010. V. 46, N. 7. P. 520–522.

24. **Bohl N.** et al. Compact Method for Modeling and Simulation of Memristor Devices: Ion Conductor Chalcogenide-based Memristor Devices // *IEEE/ACM International Symposium on Nanoscale Architectures.* June 2010. P. 1–4.

25. **Mahvash M.** et al. A memristor SPICE model for designing memristor circuits (MWSCAS-2010), 2010. P. 989–992.

26. **Rak A.** et al. Macromodelling of the memristor in SPICE // *IEEE Trans. Computer.-Aided Design Integr. Circuits Syst.* 2010. V. 29, N. 4. P. 632–636.

27. **Zhang Y.** et al. Approximated SPICE model for memristor // *ICCCAS.* 2009. P. 928–931.

28. **Kvatinsky S.** et al. Verilog-A for Memristor models // *CCIT Technical Report.* 2011.

29. **Zaplatilek K.** Memristor modeling in MATLAB & Simulink // *Proceedings of the European Computing Conference.* 2011. P. 62–67.

V. V. Rakitin^{1,2}, Leading Researcher, e-mail: rusakov@ippm.ru, **S. G. Rusakov**¹, Chief Researcher
¹ Institute for Design Problems in Microelectronics of Russian Academy of Sciences
² Scientific Research Physical Problems Institute named after F. V. Lukin

Electrical Models of Memristor Elements

The principles to construct memristor models for circuit simulation are discussed. The memristor is a new two-terminal circuit element. The memristor can be considered as resistor with memory. It is necessary to exploit the additional differential equation for incorporation the memory property into memristor model. The presence of the own internal differential equation is a peculiarity of the memristor models from circuit simulation point of view. The main known approaches to generate the memristor models for circuit simulation are presented in the paper. The linear drift and nonlinear models are given. The model equations can be considered as basic relations for model libraries in standard circuit simulators. The application of the models is illustrated by simulation examples of the different modes of electrical behavior using memristor elements.

Keywords: circuit simulation, memristor, electrical models, memristor modeling, window function

References

1. **Chua L.** (1971) Memristor — The missing circuit element. *IEEE Trans. Circuit Theory.* 1971. V. 18, N. 5. P. 507–519.

2. **Widrow B.** Adaptive ADALINE neuron using chemical memristors. Stanford Electron. Lab., Stanford, CA, Tech. Rep. 1553-2, 1960.

3. **Ayrapetyants S., Rakitin V.** et al. Bionic aspects of microelectronics. *Electronic Engineering. Ser. Microelectronics.* 1967. N. 5. P. 15.

4. **Gibbons J.** et al. Switching properties of thin NiO films. *Solid-State Electron.* 1964. V. 7. P. 785–790.

5. **Simmons J.** Generalized formula for the electric tunnel effect between similar electrodes separated by a thin insulating film. *J. Appl. Phys.* 1963. V. 34, N. 6. P. 1793–1803.

6. **Strukov D.** et al. The missing memristor found. *Nature.* 2008. V. 453. P. 80–83.

7. **Mazumder P.** et al. Memristors: Devices, Models and Applications. *Proc. IEEE.* 2012. V. 100, N. 6. P. 1911.

8. **Likharev K.** CrossNets: Neuromorphic Hybrid CMOS/Nano-electronic Networks. *Science of Advanced Materials.* 2011. V. 3. P. 322.

9. **Kim S.** et al. Flexible Memristive Memory Array on Plastic Substrates. *Nano Lett.* 2011. V. 11. P. 5438–5442.

10. **Eshraghian K.** et al. Memristive Device Fundamentals and Modeling: Applications to Circuits and Systems Simulation. *Proceedings of the IEEE.* 2012. V. 100, N. 6. P. 1911–1916.

11. **Yakopcic C.** et al. Memristor SPICE Modeling. *Advanced in Neuromorphic Memristor Science and Application.* 2012. P. 211–244.

12. **Joglekar Y.** et al. The elusive memristor: properties of basic electrical circuits. *Eur. J. Phys.* 2009. 30(661).

13. **Biolek Z.** et al. Spice model of memristor with nonlinear dopant drift. *Radioengineering.* 2009. V. 18(2). P. 210–214.

14. **Pino** et al. A Compact method for modeling and simulation of memristor devices: ion conductor chalcogenide-based memristor devices. *IEEE/ACM international symposium on nanoscale architectures.* 2010. P. 1–4.

15. **Abdalla H.** et al. SPICE Modeling of Memristors. *ISCAS.* 2011. P. 1832–1835.

16. **Yakopcic C.** et al. A memristor device model. *IEEE Electron Dev Lett.* 2011. V. 32(10). P. 1436–1438.

17. **Laiho M.** et al. Cellular nanoscale network cell with memristors for local implication logic and synapses. *Proc. of ISCAS.* 2010. P. 2051–2054.

18. **Lehtonen E.** Memristive Computing, dissertation UNIVERSITY OF TURKU. 2012. URL: <http://www.doria.fi/bitstream/handle/10024/79925/AnnalesA14461LehtonenDISS.pdf?sequence=4>

19. **Chang T.** et al. Synaptic behaviors and modeling of a metal oxide memristor device. *Appl. Phys. A.* 2011. V. 102. P. 857–863.

20. **Kvatinsky S.** et al. TEAM: Threshold Adaptive Memristor Model. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS.* 2012. V. 60, N. 1. P. 211–221.

21. **Benderli S.** et al. On SPICE macromodelling of TiO₂ memristor. *Electron. Lett.* 2009. V. 45, N. 7. P. 377–379.

22. **Biolek D.** et al. PSPICE modeling of meminductor. *Analog Integrated Circuits and Signal Processing.* January 2011. V. 66, N. 1. P. 129–137.

23. **Biolek D.** et al. SPICE Modeling of Memcapacitor. *Electronics Letters.* 2010. V. 46, N. 7. P. 520–522.

24. **Bohl N.** et al. Compact Method for Modeling and Simulation of Memristor Devices: Ion Conductor Chalcogenide-based Memristor Devices. *IEEE/ACM International Symposium on Nanoscale Architectures.* June 2010. P. 1–4.

25. **Mahvash M.** et al. A memristor SPICE model for designing memristor circuits (MWSCAS-2010), 2010. P. 989–992.

26. **Rak A.** et al. Macromodelling of the memristor in SPICE. *IEEE Trans. Computer.-Aided Design Integr. Circuits Syst.* 2010. V. 29, N. 4. P. 632–636.

27. **Zhang Y.** et al. Approximated SPICE model for memristor. *ICCCAS.* 2009. P. 928–931.

28. **Kvatinsky S.** et al. Verilog-A for Memristor models. *CCIT Technical Report.* 2011.

29. **Zaplatilek K.** Memristor modeling in MATLAB & Simulink. *Proceedings of the European Computing Conference.* 2011. P. 62–67.

В. И. Анисимов¹, д-р техн. наук, проф., e-mail: vianisimov@inbox.ru,
В. Н. Гридин¹, д-р техн. наук, проф., директор, e-mail: info@ditc.ras.ru,
М. А. Шабани², аспирант, e-mail: mabruk_1975_07@mail.ru,

¹ Центр информационных технологий в проектировании РАН

² Санкт-Петербургский государственный электротехнический университет

Методы построения высокопроизводительных распределенных систем автоматизации схмотехнического проектирования

Рассматриваются способы повышения надежности функционирования и увеличения производительности распределенных систем автоматизированного проектирования за счет минимизации времени взаимодействия веб-сервисов с сетью Интернет на основе перехода к компактной форме хранения и обработки разреженных матриц. Дается описание основных методов компактного хранения и обработки разреженных матриц, списковых схем хранения, метода строчного фиксированного формата, метода строчно-столбцового формата. Показывается, что отличительной особенностью методов фиксированного формата является невозможность включения в описание дополнительных ненулевых элементов, что исключает возможность непосредственного применения этих методов для обработки информации при решении систем уравнений любым численным методом вследствие неизбежного появления новых ненулевых элементов в процессе этого решения. Для возможности учета новых ненулевых элементов в компактном описании моделируемой системы предлагается использовать двухэтапную процедуру формирования такого описания и дается описание реализации двухэтапной процедуры обработки разреженных матриц на основе разделения ее на две независимые части символьного и численного анализа.

Ключевые слова: системы автоматизированного проектирования, веб-сервисы, моделирование систем, компактная обработка, разреженные матрицы, распределенные системы, Интернет-технологии

Введение

При организации работы распределенных систем автоматизированного проектирования на основе веб-сервисов особенно актуальными являются вопросы увеличения их производительности, поскольку для повышения надежности функционирования распределенной системы необходимо минимизировать время взаимодействия веб-сервисов с сетью Интернет [1–4]. Такая задача может быть решена путем перехода к компактной форме хранения и обработки разреженных матриц на основе тех или иных методов сжатия данных.

Наличие в математическом описании моделируемых систем разреженных матриц ставит задачу изменения стандартных подходов к формированию и решению систем уравнений. Это объясняется, с одной стороны, требованием экономии памяти, которую нежелательно использовать для хранения нулевых элементов, а с другой стороны, требованием повышения быстродействия за счет устранения выполнения арифметических операций с нулевыми элементами. Применяемый обычно для этого способ логической проверки элементов в целях устранения арифметических операций с нулевыми элементами не приводит к ожидаемому эффекту увеличения быстродействия программного обеспечения, поскольку для выполнения логических операций проверки элементов также необходимо затратить определенное время.

Степень разреженности матриц оценивается коэффициентом разреженности

$$\alpha = \frac{m}{m_{\max}} = \frac{m}{n^2},$$

где m — число ненулевых элементов матрицы; n — порядок матрицы.

Известные методы компактной обработки разреженных матриц существенно различаются по своим характеристикам и эффективности, поэтому для правильного выбора соответствующего метода необходимо провести их сравнительную оценку, которая приводится ниже для всех возможных методов.

Метод индексно-адресных матриц

Метод основан на том, что вводится некоторая целочисленная матрица, которая в точности повторяет структуру исходной матрицы W , входящей в уравнение системы моделирования. В качестве элементов индексно-адресная матрица содержит порядковый номер a ненулевых элементов матрицы W , которые перечисляются в некотором массиве WZ . При этом если порядковый номер ненулевого элемента исходной матрицы равен a , то в индексно-адресную матрицу вводится значение $A(i, j) = a$.

При практической реализации метода индексно-адресных матриц достаточно провести сканирование индексно-адресной матрицы и выбрать из нее очередной ненулевой элемент. Затем в соответ-

ствии с ее числовым значением необходимо выбрать соответствующий порядковый элемент из массива WZ. Эффективность метода индексно-адресных матриц в случае, когда для хранения индексно-адресной матрицы A используется тип данных длиной 2 байт, а для хранения каждого значимого элемента исходной матрицы, используются данные длиной 8 байт, может быть определена выражением

$$\beta = \frac{8n^2}{n^2(\alpha \cdot 8 + 2)} = \frac{1}{\alpha + 0,25}.$$

Из приведенного соотношения видно, что коэффициент экономии памяти для данного метода всегда меньше 4. Следовательно, метод индексно-адресных матриц характеризуется незначительной эффективностью при компактной записи разреженных матриц.

Связные схемы хранения

К связным схемам относится классический метод Кнута [5, 6], а также сокращенные варианты основного метода.

При использовании основного метода Кнута необходимо создать следующие массивы:

WZ — для значений ненулевых элементов w_{ij} исходной матрицы;

WI — для номеров строк ненулевых элементов;

WJ — для номеров столбцов ненулевых элементов;

NR — для хранения относительного адреса a следующего ненулевого элемента строки (a — порядковый номер элемента в массиве WZ);

NC — для хранения относительного адреса a следующего ненулевого элемента столбца;

ER — для относительного адреса a входа в очередную строку;

ES — для относительного адреса a входа в очередную столбец.

Весьма важной особенностью метода Кнута является возможность записи элементов массива WZ в любом порядке и, как следствие, возможность введения в описание дополнительных ненулевых элементов. Достоинством метода Кнута является возможность сканирования исходной матрицы как по строкам, так и по столбцам, некоторым недостатком метода является значительное число массивов, необходимых для хранения информации о ненулевых элементах исходной матрицы.

Если учесть, что массивы WZ, WI, WJ, NR, NC имеют длину m , определяемую числом ненулевых элементов, а массивы ER, ES имеют длину n , определяемую порядком исходной матрицы, то эффективность использования памяти может быть определена для метода Кнута выражением

$$\beta = \frac{8n^2}{16n^2 \cdot \alpha + 4n} \approx \frac{1}{2\alpha}.$$

Из полученного выражения видно, что с уменьшением α эффективность метода неограниченно возрастает.

Сокращенный метод Кнута позволяет уменьшить число используемых массивов путем исключения из полного описания массивов, которые реализуют сканирование по столбцам. При этом для компактного описания исходной матрицы необходимо ввести только массивы WZ, WJ, NR, ER. Аналогично можно составить вариант сокращенной схемы Кнута, позволяющий осуществить сканирование только по столбцам. При этом для компактного описания необходимо использовать только массивы WZ, ES, NC, WI.

Эффективность любой из схем сокращенного метода Кнута может быть определена выражением

$$\beta = \frac{8n^2}{n^2 \cdot \alpha \cdot 12 + n \cdot 2} \approx \frac{1}{1,5\alpha}.$$

Очевидно, что эффективность сокращенной схемы Кнута выше, чем эффективность полной схемы Кнута, однако это достигается за счет ограничения возможности выбора направления сканирования, что является некоторым недостатком сокращенного метода Кнута. Однако возможность введения в описание дополнительных ненулевых элементов, а также произвольный порядок их записи в массиве остаются в силе. Это обстоятельство является существенным достоинством сокращенного метода Кнута.

Методы фиксированного формата

К методам фиксированного формата относятся метод строчного формата и метод строчно-столбцового формата.

Для использования метода строчного фиксированного формата [6] требуются следующие массивы:

WZ — для хранения значения ненулевых элементов w_{ij} исходной матрицы;

WJ — для хранения индексов столбцов ненулевых элементов исходной матрицы W ;

ER — массив, содержащий указатели точек входа в очередную строку.

Длина массивов WZ, WJ составит m элементов, а длина массива ER составит $n + 1$ элементов, при этом в $n + 1$ заносится значение $m + 1$.

Одной из отличительных особенностей метода является невозможность произвольного выбора порядка записи ненулевых элементов в массиве WZ, и, как следствие, невозможность включения в описание дополнительных ненулевых элементов. Это объясняется тем, что формат всех массивов жестко зафиксирован и не может меняться произвольным образом в процессе расчета.

Эффективность метода строчного фиксированного формата может быть определена выражением

$$\beta = \frac{8n^2}{n^2 \cdot \alpha \cdot 10(n+1) \cdot 2} \approx \frac{1}{1,25\alpha}.$$

Очевидно, что эффективность метода выше, чем эффективность рассмотренных ранее методов. Существенным недостатком метода является невозможность включения в описание дополнительных элементов, что объясняется наличием фиксированного формата.

Метод строчно-столбцового формата [6, 7] также основан на использовании фиксированного формата. Однако, в отличие от предыдущего метода, предполагается, что исходная матрица является структурно-симметричной, так что для каждого ненулевого элемента w_{ij} можно поставить в соответствие элемент w_{ji} . В случае если такой элемент в исходной матрице отсутствует, то его необходимо создать искусственно путем включения в компактное описание элемента $w_{ji} = 0$.

Для компактного хранения исходных элементов матрицы требуется создать три массива:

WD — для хранения диагональных элементов;

WL — для хранения ненулевых элементов, расположенных ниже диагонали (поддиагональных элементов);

WU — для хранения ненулевых элементов, расположенных выше диагонали (наддиагональных элементов).

В соответствии с методом при формировании массива WU наддиагональные элементы записываются по строкам, а при формировании массива WL поддиагональные элементы записываются по столбцам. Согласно принятому порядку формирования массивов WU, WL, относительный адрес некоторого элемента w_{ij} , расположенного в массиве WU, совпадает с относительным адресом элемента w_{ji} , расположенным в массиве WL. Отмеченное свойство существенно упрощает процесс программирования для организации сканирования элемента.

Для хранения индексов строк и столбцов ненулевых элементов в методе строчно-столбцового фиксированного формата используется массив WJI, при этом данный массив содержит номера столбцов ненулевых элементов, расположенных выше диагонали, которые совпадают с номерами строк транспонированных ненулевых элементов, расположенных ниже диагонали. Для определения точки входа в строку выше диагонали (точка входа в столбец ниже диагонали) используется массив ERC. В последний n -й элемент этого массива заносится значение $\frac{m-n}{2} + 1$.

Если учесть, что длина массива WD составляет n элементов, длина массивов WU, WL, WJI состав-

ляет $\frac{m-n}{2}$ элементов, а длина массива ERC — n элементов, то эффективность метода строчно-столбцового фиксированного формата может быть определена выражением

$$\beta = \frac{8n^2}{n^2 \cdot \alpha \cdot 9 + n} \approx \frac{8}{9} = \frac{1}{1,1\alpha}.$$

Отсюда следует, что эффективность метода выше, чем всех рассмотренных выше методов. Однако так же, как для метода строчного фиксированного формата, метод строчно-столбцового фиксированного формата не позволяет перечислять ненулевые элементы в произвольном порядке, следовательно, не имеется возможности включения дополнительных элементов в компактное описание.

Сравнительная оценка эффективности и реализация методов сжатия данных

В таблице приводятся сравнительные оценки эффективности и данные о накладных расходах на оперативную память для различных методов сжатия данных на основе компактного хранения разреженных матриц. В этой же таблице приведена числовая оценка эффективности для коэффициента разреженности $\alpha = 0,1$.

На основании сравнительной оценки методов следует, что наибольший интерес для практической реализации высокопроизводительного программного обеспечения распределенных систем автоматизации схемотехнического проектирования представляют метод индексно-адресных матриц и метод строчно-столбцового фиксированного формата. Первый метод, хотя и имеет сравнительно низкую эффективность, но характеризуется чрезвычайно простой технологией реализации программного обеспечения, для построения которого достаточно выполнить формирование индексно-адресной матрицы A и одномерного массива параметров моделируемой схемы WZ . Массив WZ целесообразно создавать как объект класса коллекций `ArrayList`, что позволяет использовать для формирования и

Схема	Эффективность	Накладные расходы, %	Оценка эффективности
Индексно-адресные матрицы	$\frac{1}{\alpha + 0,25}$	$\frac{25}{\alpha}$	2,86
Схема Кнута	$\frac{1}{2\alpha}$	100	5
Сокращенные схемы Кнута	$\frac{1}{1,5\alpha}$	50	6,7
Строчный фиксированный формат	$\frac{1}{1,25\alpha}$	25	8
Строчно-столбцовый фиксированный формат	$\frac{1}{1,1\alpha}$	10	9

обработки этого массива методы класса коллекций. Обработка данных компактного массива WZ осуществляется на основании результатов простого сканирования индексно-адресной матрицы A , при этом относительный адрес элемента w_{ij} в массиве WZ определяется значением элемента $A(i, j)$. Использование этого метода особенно целесообразно в случаях, когда имеется программное обеспечение системы моделирования на основе полного описания разреженных матриц, так как его переработка к форме с компактной обработкой данных требует минимальных затрат трудовых ресурсов.

Метод строчно-столбцового фиксированного формата обладает наибольшей эффективностью по сравнению со всеми остальными рассмотренными методами. Однако, поскольку формат всех массивов жестко зафиксирован, и не может меняться произвольным образом в процессе расчета, отличительной особенностью метода является невозможность включения в описание дополнительных ненулевых элементов. Это обстоятельство исключает возможность непосредственного применения строчно-столбцового фиксированного формата для обработки информации при решении систем уравнений любым численным методом вследствие неизбежного появления новых ненулевых элементов в процессе этого решения. Поэтому для возможности учета новых ненулевых элементов в компактном описании моделируемой системы необходимо использовать двухэтапную процедуру формирования такого описания [8]. Двухэтапная процедура характеризуется относительно сложной логикой построения программного обеспечения, и переход к компактному описанию является весьма трудоемким процессом. Построение веб-сервиса на основе компактной обработки данных существенно упрощается при наличии его прототипа на основе полного математического описания задачи.

При реализации двухэтапной процедуры на первом, символьном этапе решается задача определения размеров всех используемых массивов, при этом не ставится задача численного формирования этих массивов. Поскольку задача формирования численных массивов на символьном этапе не ставится, то выполнение этого этапа может быть осуществлено путем заполнения всех массивов произвольными числовыми константами, которые должны отобразить наличие или отсутствие соответствующих элементов матрицы некоторой информацией. Иначе говоря, на первом этапе работа над числовой матрицей может быть заменена обработкой некоторой индексной матрицы C , элементы которой имеют только два произвольных значения (например, 0 и 1). При этом, если в исходной матрице некоторый элемент $w_{ij} \neq 0$, то для введенной индексной матрицы C соответствующий элемент $C_{ij} = 1$, а все остальные элементы равны 0. Таким образом, вместо рассмотрения исходной схемы будет рассмат-

риваться некоторый "портрет" этой схемы, в точности отображающий ее структуру, но не содержащий информации о числовых значениях параметров. Проведение процедуры LU-факторизации над индексной матрицей C позволяет выявить все появляющиеся при LU-факторизации новые ненулевые элементы и установить тем самым фактический формат всех массивов. Все параметры этого формата на заключительном шаге символьного этапа заносятся в координатные массивы WJI и ECR, что позволяет после выполнения символьного этапа удалить индексную матрицу C .

Формирование числовых массивов WD, WU, WL осуществляется на втором этапе, при выполнении которого используются определенные на первом этапе форматы массивов, учитывающие зарезервированные места для всех новых ненулевых элементов. Для формирования компактных числовых массивов, а также для решения систем уравнений на основе LU-факторизации используются виртуальные алгоритмы формирования и обработки компактных массивов, не требующие построения полного математического описания задачи.

Существенным достоинством такой двухэтапной процедуры является разделение ее на две независимые части символьного и численного анализа. Так как практически все реальные задачи проектирования электронных схем связаны с многовариантным расчетом схемы одной и той же структуры, то символьный этап выполняется для каждой структуры единственный раз, в то время как численный этап реализуется десятки, сотни, а часто и тысячи раз. Поэтому накладные расходы от введения символьного этапа в реальных задачах моделирования систем с разреженными матрицами оказываются весьма незначительными, так как все объекты, связанные с обработкой индексной матрицы C , не используются при выполнении численного этапа. Поскольку программное обеспечение современных распределенных систем автоматизированного проектирования реализуется на платформенно-независимых языках Java или C#, имеющих встроенные средства распределения и освобождения неиспользуемой динамической памяти, то все объекты, созданные на символьном этапе, автоматически удаляются системой при завершении этого этапа.

Заключение

Сравнительная оценка методов сжатия данных на основе компактной обработки разреженных матриц позволяет сделать вывод, что наибольший интерес для практической реализации высокопроизводительного программного обеспечения распределенных систем автоматизации схмотехнического проектирования представляют методы индексно-адресных матриц и метод строчно-столбцового фиксированного формата. Метод индексно-адресных матриц позволяет наиболее просто привести про-

граммное обеспечение к требуемой компактной форме, а метод строчно-столбцового фиксированного формата обеспечивает наибольшую эффективность при функционировании распределенных систем автоматизированного проектирования. Для возможности учета новых ненулевых элементов в компактном описании моделируемой системы при использовании метода строчно-столбцового фиксированного формата необходимо использовать двухэтапную процедуру формирования такого описания, при этом на первом, символьном этапе решается задача определения размеров всех используемых массивов, а на втором этапе ставится задача численного формирования рабочих массивов.

Практическая реализация рассмотренных методов компактной обработки данных позволяет существенно повысить производительность работы веб-сервисов распределенных систем автоматизированного проектирования и обеспечивает повышение надежности работы системы вследствие уменьшения времени взаимодействия с сервером в процессе эксплуатации САПР.

Список литературы

1. **Гридин В. Н., Анисимов В. И.** Методы построения систем автоматизированного проектирования на основе Интернет-технологий и компактной обработки разреженных матриц // Информационные технологии в проектировании и производстве. 2009. № 1. С. 3—7.
2. **Гридин В. Н., Анисимов В. И., Ларистов А. И., Аль-Шахи Моххамед.** Web-ориентированная база данных сеанса проектирования // Информационные технологии и вычислительные системы. 2013. № 3. С. 40—45.
3. **Гридин В. Н., Дмитриевич Г. Д., Анисимов Д. А.** Построение систем автоматизированного проектирования на основе Web-технологий // Информационные технологии. 2011. № 5. С. 23—27.
4. **Гридин В. Н., Дмитриевич Г. Д., Анисимов Д. А.** Построение систем автоматизированного проектирования на основе Web-сервисов // Автоматизация в промышленности. 2011. № 1. С. 9—12.
5. **Кнут Д.** Искусство программирования для ЭВМ. Т. 1 / Пер. с англ. М.: Мир, 1976. 734 с.
6. **Писсанецки С.** Технология разреженных матриц / Пер. с англ. М.: Мир, 1988. 406 с.
7. **Влах И., Сингхал К.** Машинные методы анализа и проектирования электронных схем / Пер. с англ. М.: Радио и связь, 1988. 560 с.
8. **Гридин В. Н., Дмитриевич Г. Д., Анисимов Д. А.** Построение веб-сервисов систем автоматизации схемотехнического проектирования // Информационные технологии и вычислительные системы. 2012. № 4. С. 79—84.

V. I. Anisimov¹, Sciences, prof., e-mail: vianisimov@inbox.ru

V. N. Gridin¹, director CITP RAS, e-mail: info@ditc.ras.ru,

M. A. Shabani², Postgraduate, e-mail: mabruk_1975_07@mail.ru,

¹Center information technology in the design of the RAS

²Sankt Petersburg State Electrotechnical University

Methods of Construction of High-Performance Distributed Systems Automation Circuit Design

Discusses ways to improve the reliability and the increase of distributed computer-aided design by minimizing the interaction time web services to the Internet on the basis of the transition to a compact form of data storage and processing of sparse matrices. Describes the main methods of compact storage and processing of sparse matrix and the comparative evaluation of the effectiveness of the method index-addressable matrix of list storage schemes, fixed-format string method, and the method of row-column format. Shows that feature fixed-format methods is not possible to include a description of additional non-zero elements, which precludes the direct application of these techniques for information processing in solving systems of equations by any numerical method, because of the inevitable emergence of new non-zero elements in the process of this decision. To be able to incorporate new non-zero elements in the compact description of the simulated system is proposed to use a two-stage procedure for the formation of such a description and a description of the implementation of a two-stage procedure for handling sparse matrix on the basis of its division into two independent parts symbolic and numerical analysis.

Keywords: computer-aided design, web services, system modeling, compact processing, sparse matrices, distributed systems, Internet-based technologies

References

1. **Gridin V. N., Anisimov V. I.** Metody postroeniya sistem avtomatizirovannogo projektirovaniya na osnove Internet-tehnologij i kompaktnoj obrabotki razrezhennyh matric. *Informacionnye tehnologii v projektirovanii i proizvodstve*. 2009. N. 1, P. 3—7.
2. **Gridin V. N., Anisimov V. I., Laristov A. L., Al'-Shami M. Hhamed.** Web-orientirovannaja baza dannyh seansa projektirovaniya. *Informacionnye tehnologii i vychislitel'nye sistemy*. 2013. N. 3. P. 40—45.
3. **Gridin V. N., Dmitrevich G. D., Anisimov D. A.** Postroenie sistem avtomatizirovannogo projektirovaniya na osnove Web-tehnologij. *Informacionnye tehnologii*. 2011. N. 5. P. 23—27.

4. **Gridin V. N., Dmitrevich G. D., Anisimov D. A.** Postroenie sistem avtomatizirovannogo projektirovaniya na osnove Web-servisov. *Avtomatizacija v promyshlennosti*. 2011. N. 1. P. 9—12.
5. **Knut D.** Iskusstvo programirovaniya dlja JeVM. T. 1: Per. s angl. M.: Mir, 1976. 734 p.
6. **Pissanecki C.** Tehnologija razrezhennyh matric: Per. s angl. M.: Mir, 1988. 406 p.
7. **Vlah I., Singhal K.** Mashinnye metody analiza i projektirovaniya jelektronnyh shem, per. s angl. M.: Radio i svjaz', 1988. 560 p.
8. **Gridin V. N., Dmitrevich G. D., Anisimov D. A.** Postroenie veb-servisov sistem avtomatizacii shemotekhnicheskogo projektirovaniya. *Informacionnye tehnologii i vychislitel'nye sistemy*. 2012. N. 4. P. 79—84.

С. М. Авдошин, зав отделением программной инженерии, зав. каф., e-mail: savdoshin@hse.ru

Е. Ю. Песоцкая, доцент каф., e-mail: epesotskaya@hse.eu

Национальный исследовательский университет "Высшая школа экономики", Москва

Экосистемы программного обеспечения: инновации в ИТ

Перед владельцами и разработчиками платформ встает немало возможностей, но также и вызовов в силу высокой конкуренции и желания пользователей иметь доступ к разнообразнейшим продуктам и приложениям. В статье приведен обзор существующих определений и подходов к созданию экосистем, рассматриваются ключевые участники и модель, которая может использоваться для создания экосистемы. Анализируются преимущества, которые дает использование экосистем и внедрение инноваций для компаний-разработчиков.

Ключевые слова: экосистема, программное обеспечение, разработчики, архитектура, информационные технологии

Введение

Возникновение экосистем — одно из самых значительных событий для софтверных компаний. Производители программного обеспечения (ПО) все чаще стремятся сосредоточить свои усилия на том, что отличает их от конкурентов, пытаются развивать свои сильные стороны и при этом диверсифицировать свою деятельность, что сегодня невозможно без экосистемы. Кроме этого, на развитие экосистем сильно повлияло ужесточение требований к программному обеспечению, поскольку область ИТ значительно отличается от других отраслей экономики. Программное обеспечение компании часто выступает частью одной или нескольких программных экосистем.

Концепция экосистемы может применяться для описания региональных программ экосистемы, где различные производители программного обеспечения путем сотрудничества на конкретном рынке или в рамках совместной работы с открытым кодом производят новые программные продукты на основе набора конкретных технологий и с использованием набора возможностей.

В основном экосистема программного обеспечения взаимосвязана с продуктами или сообществом разработчиков вокруг продукта. Некоторые западные авторы определяют экосистему как "...совокупность предприятий, функционирующих как единое целое и взаимодействующих на общем рынке программного обеспечения и услуг, а также отношений между ними" [3]. Кроме того, экосистема определяется как экономическое сообщество при взаимной поддержке взаимодействующих органи-

заций и частных лиц — участников делового мира [6]. В любом случае экосистема — это синтез:

- симбиоза отношений, где выживание всех членов подразумевает выживание экосистемы;
- совместной эволюции, в рамках которой партнеры совместно развиваются, а также формируют возможности вокруг инноваций;
- фокусирования экосистемы на конкретной платформе или продукте, который может быть использован и расширен всеми партнерами экосистемы.

Компании строят свой бизнес вокруг инноваций: они совместно трудятся над разработкой новых продуктов и поддержанием конкурентоспособности, а также над удовлетворением потребностей клиентов. Подобная концепция экосистемы программного обеспечения существует в Силиконовой долине, в Бангалоре и Ченнаи в Индии [8]. Программная экосистема вокруг устройства постепенно становится важнее самого устройства.

Экосистемы программного обеспечения в России только зарождаются, хотя некоторые рынки существуют только в формате экосистем, например рынок мобильных приложений. Программные экосистемы, как правило, присутствуют в самых динамичных областях, таких как встроенные системы, интернет-бизнес, игры.

1. Понятие экосистемы программного обеспечения

Экосистема ПО включает программное обеспечение, среду его разработки, эксплуатации, сопровождения и утилизации, связанные между собой процессами обмена программными продуктами и "интеллектом" [2]. Кроме того, экосистема определяется как экономическое сообщество при взаим-

ной поддержке взаимодействующих организаций и частных лиц — участников делового мира [6]. Команды программных инженеров переходят от неудобных и закрытых групп к открытой и сетевой организации. Также можно утверждать, что экосистема образуется из связанных проектов и технологий, многие из которых сначала развивались в рамках проекта, а впоследствии стали самостоятельными приложениями или продуктами [1].

Изначально принцип экосистем использовали несколько компаний, которые совмещали разработку программного и аппаратного обеспечения. Первопроходцами выступали такие компании, как Nokia, Motorola, прочие компании, ориентированные на телеком-сервисы. Новое поколение решений представили такие компании, как Google, Samsung, Oracle, SAP, Microsoft, Apple. Эти компании взялись за разработку программной платформы, подходящей для любых устройств, и заявили о своих продуктах как о части экосистемы. Принципиально экосистемы отличаются возможностью изменения платформ, что, в свою очередь, влияет на оптимальный масштаб и технологии проектирования. Например, Microsoft определяет экосистему программного обеспечения как "совокупность взаимодействий и взаимных влияний организаций (государственных, учебных и коммерческих) и индивидуумов, работающих с программным обеспечением" [10]. При этом Windows является частью экосистемы, в которую вовлечены сотни тысяч людей, чье предназначение состоит в том, чтобы у пользователей всегда был выбор, будь то конфигурация компьютера, настройка ОС или приложения/службы, которые они создают, используют или предлагают. Разработчики Windows постарались предоставить пользователю максимальный выбор по использованию продуктов и услуг всей экосистемы. "Вещи работают лучше, когда аппаратное и программное обеспечение обдумывается вместе", — заявляет генеральный директор Microsoft Стив Балмер.

Одна из важных целей, которую преследует корпорация, взявшись за выпуск планшетов самостоятельно, — сыграть роль катализатора для привлечения как можно большего числа производителей и разработчиков к Windows 8.

Экосистема объединяет в себе усилия различных участников: владельцев, разработчиков, пользователей, инвесторов, содействует развитию механизмов инвестирования, разрабатывает специальные программы и реализует программные проекты. Экосистема позволяет распределять усилия и создавать среду, в которой некоторые из участников сосредоточены на аппаратной платформе, в то время как другие игроки вовлечены в процесс интеграции своего программного обеспечения поверх существующих платформ. Многие компании стремятся сфокусироваться на собственных преимуще-

ствах и сильных сторонах, отдавая на аутсорсинг непрофильные направления деятельности.

В рамках стратегии развития бизнеса экосистема включает в себя поставщиков услуг, дистрибьюторов, изготовителей комплектного оборудования, системных интеграторов и разработчиков программных решений, которые в совокупности образуют высокоинтегрированное сообщество профессионалов в области промышленной автоматизации.

Экосистема имеет достаточно широкую классификацию, поскольку поставщики услуг могут делиться на разработчиков, архитекторов, бизнес-аналитиков, системных интеграторов и прочих участников процесса разработки ПО. Владельцы, в свою очередь, могут подразделяться на владельцев платформ, продуктов, коробочных решений, SaaS-решений и пр.

Каждый участник экосистемы играет ключевую роль в создании среды, в которой любой пользователь может взять в руки компьютер или мобильное устройство, а различные компании могут предлагать пользователю на выбор уникальные продукты или услуги. Такой подход позволяет повышать эффективность самых разных аспектов производственной деятельности, таких как обеспечение качества, производительность, трудовые ресурсы, соблюдение экологических норм и стандартов во многих отраслях промышленности. По данным маркетинговых исследований, проводимых компанией IDC (International Data Corporation), развитие экосистемы и установление новых партнерств станет ключом к росту рынка ПО [7].

2. Классификация экосистем программного обеспечения

Наиболее ярко классификация экосистем прослеживается на мобильных платформах, отражающих принцип совместной работы над аппаратной и программной частью решения с участием множества партнеров. На мобильном рынке стоит отметить большое число конкурентов, таких как WebOS, Android, LiMo, Symbian, Windows Mobile, MeeGo и т. д. До 2009 г. большинство этих платформ не получили широкого промышленного внедрения. Однако, начиная с 2010 года, многие из вертикально интегрированных компаний потеряли долю рынка и Android с открытым исходным кодом платформы достаточно быстро восстановил свои позиции и начал завоевывать долю рынка. Кроме того, Microsoft — доминирующий игрок на рынке персональных компьютеров, активизировал свои усилия в области мобильных платформ и приложений. В результате последние несколько лет прошли в ожесточенной борьбе между различными экосистемами (рис. 1).

В представленную классификацию входят четыре возможных типа экосистем.

1. Вертикально-интегрированные компании, занимающиеся разработкой платформ аппаратного и

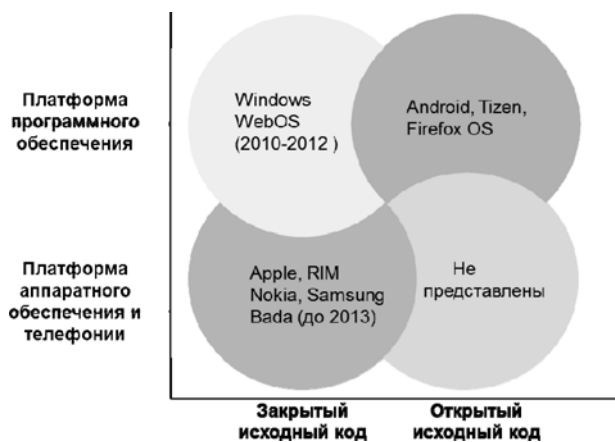


Рис. 1. Классификация экосистем программного обеспечения

программного обеспечения, в том числе производящие устройства (компьютер/телефон/планшет) и снабжающие его аппаратным обеспечением с закрытым исходным кодом. Примеры таких компаний — Apple, RIM, ранее Nokia и Samsung с Bada (платформа, прекратившая существование в 2013 г.).

2. Производители платформ, предлагающие программное обеспечение с закрытым исходным кодом и платформу для нескольких производителей телефонов, например Windows Phone и WebOS (2010—2012).

3. Производители программного обеспечения с открытым исходным кодом платформы. Программная платформа с открытым исходным кодом производится на основе следующей концепции: несколько участников (партнеров) могут объединить усилия в области развития программных продуктов, и с учетом того, что источник открыт, производители телефонов могут изменять, добавлять или удалять функции по своему усмотрению. Примерами являются Android, Tizen и Firefox OS.

Программное обеспечение с открытым исходным кодом — это на сегодняшний день уже хорошо сложившаяся часть экосистемы программного обеспечения, которая предоставляет разработчикам и пользователям важную альтернативную форму разработки и распределения программного обеспечения, что крайне необходимо.

4. Производители открытого программного обеспечения и аппаратных платформ. Пользователи такой открытой платформы или устройства имеют возможность самостоятельно изменить код программной платформы и добавить свои собственные функции. В настоящий момент на мировом рынке не представлена аппаратная платформа для телефонии с открытым кодом.

В высококонкурентной среде поведение различных экосистем может различаться, например, компания Apple для производства оборудования привлекает партнеров, фокусируясь на ограниченном наборе продуктов [9], а компания Microsoft, не имеющая собственного аппаратного подразделения,

сосредоточилась на создании вертикально-интегрированной компании, предлагающей как программное, так и аппаратное обеспечение силами партнеров. С появлением Microsoft Surface корпорация решила перенять схему работы Apple, взяв под свой контроль разработку не только программной платформы Windows 8, но и портативных устройств на ее базе. Компания Google предлагает совместимые решения и заключает множество соглашений с производителями мобильных устройств по использованию приложений Google и их пользовательских интерфейсов [4]. Так, например, продажа компании Motorola Mobility (стоимость примерно составляет половину от суммы, уплаченной за компанию три года назад) — это очередной шаг Google по укреплению своей экосистемы, которая сейчас находится в несколько хаотичном состоянии. Такой шаг снимет крупнейшее противоречие, связанное с Android, так как Google перестает конкурировать с другими производителями смартфонов и становится еще более выгодным партнером. Недавнее соглашение с Samsung об интеллектуальной собственности вместе с этим шагом говорит о том, что экосистема Android может больше не опасаться конкуренции со стороны Google и сосредоточиться на создании качественных смартфонов, что позволит компаниям объединиться в одну команду и еще больше укрепить существующую экосистему. По прогнозам [11], перемещение пользователей из экосистемы Google (на устройствах Samsung) в экосистему Samsung (на устройствах Samsung) увеличит лояльность пользователей и позволит удержать цены на более высоком уровне, сохранив высокие прибыли. В случае развития экосистемы согласно плану Google главными выгодополучателями окажутся Samsung, LG Electronics, Sony Mobile и Huawei. Это плохие новости для Microsoft и ее союзников, надежды которых были связаны с хаосом и противоречиями внутри лагеря Android.

Если же темп появления инноваций в компаниях замедлится, то можно ожидать, что большинство вертикально интегрированных компаний не смогут окупить свои инвестиции в развитие собственных решений, и преимущество получат экосистемы с открытым исходным кодом, которые более гибко и эффективно используют свои преимущества.

Рассмотрим компоненты одной из трех экосистем Apple — экосистемы iPhone (iPhone Software Ecosystem):

- разработчики ПО, проектировщики, системные архитекторы;
- дистрибьюторы и магазины, предоставляющие сервисное обслуживание;
- производители устройств (телефонов);
- провайдеры сотовой связи (сетей передачи данных);
- маркетологи и бренд-менеджеры;
- пользователи устройства.

В данной экосистеме отсутствуют системные интеграторы, либо поставщики лицензий. Основное внимание Apple уделяет разработке платформы и дизайну сложных программных интерфейсов на ограниченном аппаратном комплексе с использованием стандартных языков C++ и Objective-C.

В отличие от экосистемы iPhone, где фактически один производитель создает монополию на распространение приложений, экосистема Android взаимодействует с множеством магазинов, дистрибьюторов и рынков. Каждый рынок имеет определенные правила в отношении поведения, распределения выручки и пр. Таким образом, Android имеет гораздо больше свободного пространства на рынке и потенциально больше возможностей. Его экосистема существенно отличается от iPhone, поскольку содержит гораздо больший набор дополнительных игроков и партнеров — провайдеров библиотек, ядра системы, Java-средств разработки приложений (анимация, камера, телефония) и других подсистем.

3. Модель экосистем программного обеспечения

Принятие модели экосистемы программного обеспечения бросает компаниям вызов с целью открыть его успешные продукты и производственные линии для функциональных расширений, создаваемых сторонними разработчиками. Вместо того чтобы делать эксклюзивные и закрытые продукты для внешних разработчиков, организации изучают новые способы предоставления доступа к программным продуктам и услугам внешним заинтересованным сторонам.

Экосистемы могут принимать различные формы. В литературе различают коммерческие и социальные экосистемы. В коммерческой экосистеме участниками являются предприятия, поставщики и клиенты, объектами деятельности — программные продукты, услуги и операции, а также информация и обмен знаниями, запросы, пред- и послепродажные контакты.

В коммерческой экосистеме программного обеспечения компании, как правило, развивают программную платформу и сообщество сторонних разработчиков, обеспечивая функциональность, расширяя базовую платформу и привлекая новых пользователей, чтобы получать от них новые запросы и потребности для экосистемы.

Понятие экосистемы программного обеспечения часто дополняется социальной экосистемой, состоящей из пользователей, их социальных связей и потоков информации между ними.

Смартфоны и планшеты имеют характеристики встраиваемых устройств (*embedded systems*), а также вычислительных устройств широкого (общего) назначения. В качестве встроенного устройства они служат для выполнения специализированных функций, таких как телефонные звонки, записи видео и воспроизведения музыки, которые имеют требова-

ния к производительности в режиме реального времени.

Мобильные устройства работают при ограничении ресурсов, например, должны расходовать как можно меньше энергии, работая на аккумуляторах. Мобильный телефон также имеет характеристики вычислительных устройств широкого назначения, поскольку мобильные телефоны и планшеты предназначены для различных задач, таких как выход в Интернет через интернет-браузер, просмотр и редактирование документов, доступ к социальным и медиаресурсам, игры.

Учитывая высокую скорость инноваций, можно определить коммерческий жизненный цикл продукта равным около двух лет. Компания, которая отстает и не производит продукт, отвечающий или предупреждающий потребности клиента, может потерять долю рынка очень быстро.

Грамотно выбранная модель экосистемы обуславливает ее масштаб, распределение функций внутри всех участников, требования к качеству, способы работы внутри экосистемы.

Углубленно в модели экосистемы программного обеспечения сосредоточено основное внимание на трех блоках (рис. 2): стратегии, архитектуре и организации [3].

1. **Стратегия компании.** Стратегия предусматривает развитие бизнеса компании и увеличение доходов, она имеет большое значение для внутренней последовательности решений, принятых в рамках организации, и тесно связана со стратегией развития программного обеспечения и архитектуры. Бизнес-модель определяет, как компания создает ценность для клиентов, а затем преобразует полученные платежи в прибыль. Стратегия может также заключаться в поиске ниши, чтобы стать в ней краеугольным партнером и расширить силы в этой позиции.

2. **Архитектура.** Архитектура включает в себя техническую структуру по построению системы программного обеспечения, а также технологии выбора. Стратегия компании определяет, какие аспекты бизнеса будут развиваться в будущем и должны



Рис. 2. Модель экосистемы программного обеспечения

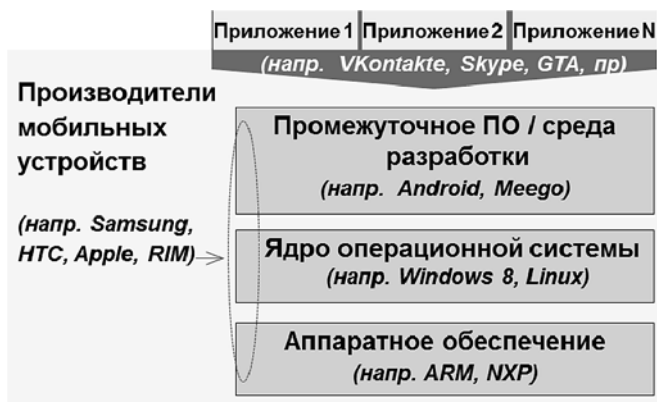


Рис. 3. Высокоуровневая архитектура экосистемы программного обеспечения

быть приоритетными. Это важные входные данные для архитектуры решения, которые позволяют эффективно управлять эволюцией будущей стоимости.

Существует множество интерфейсов, обеспечивающих интеграцию внутри экосистемы: интерфейс между подрядчиками внутри компании; интерфейс между компаниями, которые строят программы поверх существующего продукта или платформы; интерфейс между компаниями, которые работают в той же экосистеме, но обеспечивают другие виды функций или деятельности, интеграцию экосистемных решений; интерфейс между компанией и интеграторами. Изменения внешней среды требуют, чтобы участники экосистемы активно внедряли инновации, связанные с выпуском новой функциональности или модернизацией.

Высокоуровневая архитектура экосистемы представлена на рис. 3 и состоит из аппаратной платформы, ядра системы и межплатформенного ПО, которое предлагает универсальную среду разработки для производителей программных продуктов и приложений [5].

3. Организация. Организация определяет порядок работы компании с клиентами, поставщиками и партнерами по экосистеме, способ организации труда (роли и ответственность, вовлеченность в процессы, регламенты и инструменты по разработке программного обеспечения), которые тесно связаны с архитектурой и стратегией компании.

Организация изначально ориентирована на возникающие сегодня требования отрасли информационных технологий и запросы клиентов и способна предложить новые решения для уже реализованных платформ. Такой подход позволяет традиционным приложениям и приложениям, разработанным под конкретные запросы, работать вместе на одной платформе, сетевой инфраструктуре, способной обеспечить их динамичную работу и масштабируемость. Мобильные устройства и аппаратные платформы, раньше диктовавшие способ работы

приложений, теперь применяют исходя из требований приложений, которые зачастую должны работать на любом устройстве в любом месте.

Связь участников модели экосистемы осуществляется между несколькими организациями и на нескольких уровнях, как правило, по принципу лицензирования, реже — на условиях договора. Считается, что у iOS самая платежеспособная аудитория, у Android больше всего пользователей, Symbian вместе с компанией Nokia, которая производит хорошие аппараты, исторически сильны в СНГ.

Заключение

В последнее десятилетие в результате распространения мобильных технологий и развития ИТ в индустрии потребительской электроники произошла настоящая революция. В связи с этим можно утверждать, что эволюция программного обеспечения неизбежна. В настоящее время масштабное программное обеспечение все реже создает одна команда разработчиков. В процессе работы намечаются тенденции по созданию альянсов, появляются экосистемы программного обеспечения или программные экосистемы. Компаниям, которые хотят укрепить (или хотя бы не потерять) свои позиции на новом рынке программного обеспечения, требуется занять новую нишу и иначе взаимодействовать с другими заинтересованными лицами на уровне экосистемы, в то же время обеспечивая максимальную производительность и надежность внутри самой компании. При этом остро стоит необходимость поступательного совершенствования других элементов экосистемы — квалификации и опыта партнеров и разработчиков, умений и потребностей пользователей. Это вызов для многих организаций, но компании, принявшие вызов, могут рассчитывать на более успешную и продуктивную деятельность в будущем. Это и новые возможности собирать обратную связь относительно использования продукции, удовлетворенности клиентов, и многие другие аспекты, которые обычно мы упускаем из виду.

Список литературы

1. Bosch J., Hartmann H., Trew T. The changing industry structure of software development for consumer electronics and its consequences for software architectures // *Journal of Systems and Software*. 2012. V. 85. P. 178—192.
2. Jansen S., Brinkkemper S., Cusumano M. A., eds. *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry* // Cheltenham, UK: Edward Elgar Publishing, 2013. P. 85—102.
3. Bosch J., Bosch-Sijtsema P. From integration to composition: On the impact of software product lines, global development and ecosystems // *Journal of Systems and Software*. 2010. N 83(1). P. 67—76.
4. Van Angeren J., Blijleven V., Jansen S., Brinkkemper S. Complementor Embeddedness in Platform Ecosystems: The Case of Google Apps // *Proc. of the Seventh International Conference on Digital Ecosystems and Technologies*. 2013. P. 37—42.

5. **Hanssen G. K.** A longitudinal case study of an emerging software ecosystem: Implications for practice and theory // *Journal of Systems and Software*. 2012. N. 85(7). P. 1455–1466.
6. **Moore J. F.** Business ecosystems and the view from the firm // *The Antitrust Bulletin*. 2006. N. 51(1). P. 31–75.
7. **Worldwide** Software on 2007–2011 Forecast: A Preliminary Look at Delivery Model Performance // *Analytical research of IDC #206240*. March 2007.
8. **Bosch J.** From Software Product Lines to Software Ecosystems // *Proc. of 13th International Software Product Line Conference*. San Francisco, CA, 2009.

9. **Jobs S.** Thought on Flash. URL: <http://www.apple.com/hot-news/thoughts-on-flash/>. April 2010 (Accessed November 24 2011).
10. **Gantz J. F., Bibby D.** Partner Opportunity in the Microsoft Ecosystem. White paper, 2011. URL: http://www.microsoft.com/en-us/news/presskits/partnernetwork/docs/id_wp_0211.pdf (accessed May, 2014).
11. **Иванов М.** Сделка Google и Lenovo укрепляет экосистему Android. URL: <http://bezoom.uz/novosti/1247-sdelka-google-i-lenovo-ukrepyaet-ekosistemu-android> (от 30.01.2014).

S. M. Avdoshin, Head of Department, Head of the Chair, e-mail: Savdoshin@hse.ru
E. Y. Pesotskaya, Assistant Professor, e-mail: Epesotskaya@hse.ru
 National Research University Higher School of Economics Moscow

Software Ecosystem: Innovations in IT

Today customer expectations on services continue to increase. The owners of platforms face many challenges as they need to survive in a very competitive environment and provide full variety of products and applications to the end consumer. Many providers start thinking on how to define profitable customers and their needs? How to balance cost to serve and customer value? The paper provides the overview of the existing approaches to build software ecosystems. Some examples of the successful ecosystems and its evolution are explained. The paper also provides a step towards better understanding of the participants and the model that can be applicable to the software ecosystem and aims to give a better understanding of why software ecosystems and open innovation are increasingly important for software development companies.

Keywords: ecosystem, application framework, strategy, software developers, information technologies (IT)

References

1. **Bosch J., Hartmann H., Trew T.** The changing industry structure of software development for consumer electronics and its consequences for software architectures. *Journal of Systems and Software*. 2012. 85(1). P. 178–192.
2. **Jansen S., Brinkkemper S., Cusumano M. A.** Software Ecosystems: analyzing and managing business networks in the software industry. Cелtenham, UK: Edward Elgar Publishing, 2013. P. 85–102.
3. **Bosch J., Bosch-Sijtsema P.** From integration to composition: On the impact of software product lines, global development and ecosystems. *Journal of Systems and Software*. 2010. 83(1). P. 67–76.
4. **Van Angeren J., Blijleven V., Jansen S., Brinkkemper S.** Complementor Embeddedness in Platform Ecosystems: The Case of Google Apps. *Proceedings of the Seventh International Conference on Digital Ecosystems and Technologies*. 2013. P. 37–42.
5. **Hanssen G. K.** A longitudinal case study of an emerging software ecosystem: Implications for practice and theory. *Journal of Systems and Software*. 2012. N. 85(7). P. 1455–1466.

6. **Moore J. F.** Business ecosystems and the view from the firm. *The Antitrust Bulletin*. 2006. N. 51(1). P. 31–75.
7. **Worldwide** Software on 2007.2011 Forecast: A Preliminary Look at Delivery Model Performance. *Analytical research of IDC #206240*, March 2007.
8. **Bosch J.** From Software Product Lines to Software Ecosystems. *Proceeding of 13th International Software Product Line Conference*. San Francisco, CA, 2009.
9. **Jobs S.** Thought on Flash, <http://www.apple.com/hotnews/thoughts-on-flash/>, April 2010 (accessed May, 2014).
10. **Gantz J. F., Bibby D.** Partner Opportunity in the Microsoft Ecosystem. White paper, 2011. URL: http://www.microsoft.com/en-us/news/presskits/partnernetwork/docs/id_wp_0211.pdf (accessed May, 2014).
11. **Ivanov M.** Sdelka Google i Lenovo ukrepyaet ekosistemy Android. URL: <http://bezoom.uz/novosti/1247-sdelka-google-i-lenovo-ukrepyaet-ekosistemu-android>, ot Jan., 2014 (accessed May, 2014).

УДК 004.9

З. Н. Амирасланова, науч. сотрудник, e-mail: zarbaf2003@yahoo.com,
Институт кибернетики Национальной академии наук Азербайджана, г. Баку

Автоматизация формирования чертежей арматуры железобетонных конструкций

Рассмотрены вопросы автоматизации формирования чертежей арматуры железобетонных конструкций в строительном секторе. Предложено эффективное решение поставленной задачи с использованием базы данных.

Ключевые слова: AutoLISP, AutoCAD, база данных

Введение

Как известно, после разработки архитектурных чертежей строений с железобетонными конструкциями выполняются прочностные (статические) расчеты [1]. Затем по этим прочностным расчетам для каждого элемента железобетонной конструкции строения определяются диаметры арматуры, ее количество, шаг распределения, защитный бетонный слой, расстояние до соединения арматуры и т. д.

Далее формируются чертежи арматуры железобетонной конструкции. При этом проводится ряд простых расчетов. Например, длина арматуры колонн рассчитывается с учетом расстояния между этажами, арматура в фундаменте рассчитывается с учетом бетонного слоя и высоты фундамента и т. д. По этим расчетам составляются чертежи арматуры (рабочие чертежи). При этом необходимая исходная информация для проведения расчетов и составления чертежей находится в различных документах. Процесс сбора показателей элементов конструкций (колонн, балок, стен, межэтажных перекрытий и др.), вычисления длины арматуры и параметров для черчения многократно повторяется, поэтому имеет смысл автоматизировать данный процесс. В этой работе предлагается система автоматизации указанных расчетов и составления соответствующих чертежей.

Постановка задачи

Как было описано выше, после проведенных инженерами-конструкторами прочностных (статических) расчетов, для монтажа элементов¹ зданий на строительном участке должны быть подготовлены

¹ Впредь под элементами здания будут подразумеваться колонны (бетонные стойки), балки, стены, межэтажные перекрытия.

детализированные чертежи, т. е. чертежи и спецификации арматуры. В зависимости от проекта чертежи арматуры разрабатывают в соответствии с различными стандартами. Эти стандарты схожи между собой, но между ними, тем не менее, также есть некоторые различия. Рассмотрим эти различия, сравнив Еврокод-2 и российские стандарты.

Важным является то обстоятельство, что основные принципы назначения характеристик бетона и арматуры в Еврокод-2 и в российских нормах близки (по классу прочности, нормативным значениям сопротивлений, коэффициентам безопасности для материалов, диаграммам деформирования, деформационным характеристикам). Однако численные значения характеристик бетона и арматуры отличаются друг от друга. Различаются и методы оценки соответствия [2].

В Еврокод-2 включены расчетные характеристики для бетона по цилиндрической прочности, а в российских нормах — по призменной прочности. Также в российских нормативах имеются показатели качества бетона в виде марок по морозостойкости и водонепроницаемости, чего нет в европейских нормах.

Методы расчета прочности железобетонных элементов на действие изгибающих моментов и продольных сил — главного расчета железобетонных элементов, определяющего количество продольной арматуры, в Еврокоде-2 и в российских нормах основаны на единых предпосылках использования уравнений равновесия (гипотезы плоских сечений и диаграмм деформирования бетона и арматуры) и различаются только некоторыми деталями.

Конструктивные требования, приведенные в российских нормах, в целом приближены к Еврокоду-2. Это относится, в первую очередь, к анкерровке арматуры, толщине защитного слоя бетона, расстоянию между арматурными стержнями [2].

Отметим, что подобные сходства и различия проявляются и в требованиях на разработку чертежей арматуры. Вне зависимости от стандартов, при создании рабочих чертежей арматуры многоэтажных зданий затрачивается много времени на повторные действия. При этом потеря времени происходит за счет, например, поиска показателей колонн из чертежей элементов строений, выполнения простых вычислений для нахождения длины арматуры в соответствии с нормами и последующего создания рабочих чертежей в программе AutoCAD с использованием этих показателей. Среди показателей конструкции строения можно перечислить следующие: название, высота, длина, ширина, спецификация (диаметр арматуры, их количество) арматурных стержней железобетонных конструкций (колонн, балок, стен, межэтажных перекрытий) и др. Ставится задача разработки автоматизированной системы создания рабочих чертежей арматуры. Цель поставленной задачи заключается в минимизации времени и труда, затраченных на формирование чертежей арматуры, а также в минимизации механических ошибок в рабочих чертежах. В данной статье исследуется задача автоматизации формирования чертежей арматуры. Решение описывается на примере создания чертежей арматур колонн по стандарту Еврокод.

Решение поставленной задачи

Автоматизация создания чертежей арматуры возможна в среде AutoCAD с использованием языка программирования AutoLISP² [3]. При этом показатели элементов могут вводиться с клавиатуры в диалоговом режиме. В этом случае для показателей каждого этажа необходимо обратиться к нескольким документам, а также в диалоговом режиме ответить на запросы по каждому этажу. Данная процедура сильно осложняет работу оператора или инженера, выполняющего эту работу. Допущение ошибок при заполнении запросов приводит к тому, что приходится работу начинать сначала. Необходимо отметить, что заполнение такого количества запросов является очень утомительной задачей. Вместо исполнения данной процедуры было бы проще создать чертеж в AutoCAD. В связи с этим предлагается автоматизация передачи данных для генерации чертежа. С этой целью создана база данных показателей в Microsoft Access, интерфейс для ее заполнения и разработан программный модуль для генерации чертежей. Остановимся более подробно на каждой из этих задач.

² Являясь частью AutoCAD, AutoLISP позволяет определять переменные различных типов и передавать их значения командам AutoCAD при вводе данных. При ответах на запросы команд AutoCAD можно также использовать выражения AutoLISP, в которых могут выполняться различные арифметические и условные операции над числовыми значениями и значениями определенных переменных.

Создание базы данных. Перечисленные выше параметры сгруппированы, и для их сохранения предложены следующие таблицы:

1. COLUMN — перечень колонн (ID, название колонн);
2. LEVEL — показатели уровня (ID, название, показатели уровня);
3. BEAM — параметры балок (ID балки, название балки, ее ширина, высота);
4. G_COLUMN — показатели колонн.

В качестве примера приведем структуру таблицы G_COLUMN.

Field Name (Имя поля)	Data Type (Тип данных)	Description (Описание)
ID_G_C ID_Column	AutoNumber Number	Автоматизированный код Код колонн из таблицы "Column"
Level	Number	Код уровня из таблицы "Level" (уровень этажа, на котором колонна располагается)
Width_C	Number	Ширина колонны
Length_C	Number	Длина колонны
Quantity_16	Number	Количество арматуры с диаметром 16 мм
Quantity_20	Number	Количество арматуры с диаметром 20 мм
Quantity_25	Number	Количество арматуры с диаметром 25 мм
Quantity_32	Number	Количество арматуры с диаметром 32 мм
Right_Beam	Number	Код балок из таблицы "Beam"
Left_Beam	Number	Код балок из таблицы "Beam"
Front_Beam	Number	Код балок из таблицы "Beam"
Back_Beam	Number	Код балок из таблицы "Beam"
Note	Short Text	Примечание

Данные таблицы содержат только те показатели, которые необходимы для решения поставленной задачи.

Программный интерфейс. Для составления чертежей необходимо выбирать нужный элемент и при необходимости корректировать его показатели, а также загружать их в AutoLISP файл. С этой целью в программе Visual Basic Editor создана форма (сохраняющаяся как файл "COLONLAR.dvb"), которую можно вызвать из AutoLISP [4].

Процесс составления чертежей осуществляется следующим образом. Сначала в системе AutoCAD загружается файл "Columns.lsp". Активизируется функция "draw". Эта функция, в свою очередь, активизирует программу "COLONLAR.dvb". Появляется диалоговая форма (рис. 1). Эта форма создается для того, чтобы иметь возможность извлечь из базы данных Microsoft Access информацию, касающуюся колонн, балок, а также других показателей. Далее с помощью этой программы генерируются файлы AutoLISP с расширением "lsp". В этих файлах содержатся необходимые данные для генерации чертежей. Затем из диалогового окна запускается блок генерации чертежей. Во время работы этого блока также происходит обработка данных.

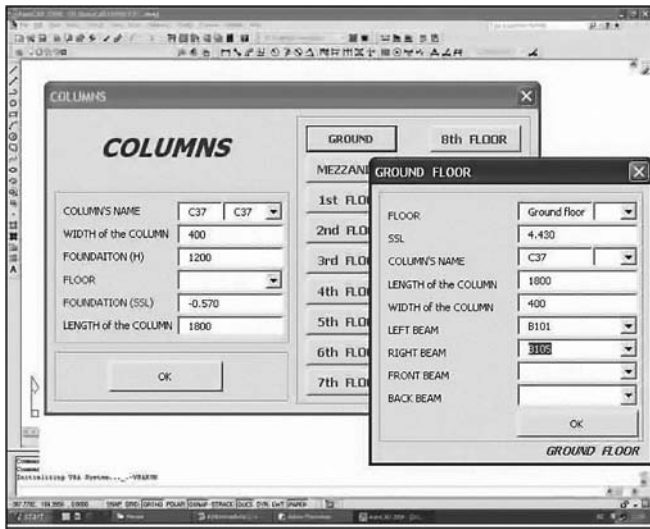


Рис. 1. Основная форма и форма "GROUND FLOOR", которая открывается из основной формы при нажатии кнопки "GROUND"

Под обработкой данных подразумевается определение, например, расстояния между этажами, расстояния от этажей до балок и т. д. [5]. Помимо этого вычисляются также такие показатели, как длина соединения арматуры, количество арматуры, расположенной по ширине колонн, их шаг и др. После этого в AutoCAD автоматически создаются рабочие чертежи арматуры. Таким образом, данные обрабатываются и генерируются по схеме, изображенной на рис. 2.

Из опубликованных в последнее время научных статей можно сделать вывод, что многие задачи решаются с использованием баз данных в системе CAD/CAE/CAM [6]. В этих системах исходные параметры для создания чертежей считываются из двух двумерных таблиц.

В рассмотренной нами задаче исходные параметры исчисляются десятками. Следует отметить, что решение поставленной задачи в AutoCAD только в диалоговом режиме AutoLISP, минуя процесс диалогового интерфейса в Visual Basic, сопровождается большими затратами времени и утомительным трудом.

Рассмотрим пример формирования чертежей арматуры в фундаменте строения. По стандартам Eurocode-2 [7] арматуру размещают у верхнего уровня фундамента колонны вертикально на глубине фундамента $40D$ (D — диаметр арматуры; если $D = 16$ мм, $40D = 40 \times 16 = 640$ мм; значение может меняться в зависимости от назначения) с минимальным наклоном в 450 мм (рис. 3). Тогда длина арматуры до наклона, с учетом защитного слоя, не должна превышать высоту фундамента H : ($40D \leq H - P$). В противном случае $40D - (H - P) = L1$, т. е. арматура расположена на глубине фундамента $(H - P)$ мм и должна быть наклонена на $(L1 + 450)$ мм. По стандарту защитный слой бетона у фундамента составляет 120 мм ($P = 120$ мм). Длина арматуры по высоте колонн, расположенных между этажами,

в зависимости от расположения арматуры слева или справа от колонны, рассчитывается различным образом: с одной стороны колонны длина арматуры, согласно стандарту [7], берется как $2/3$ межэтажной высоты (МЕН), т. е. $2/3$ МЕН, а с другой стороны, колонны как $1/3$ МЕН + $50D$. Здесь МЕН — эффективная длина, получаемая из разности расстояния между этажами и максимальной толщины балки, или, в случае отсутствия балки, толщиной пола (рис. 3). Арматура с обеих сторон распределена одинаково. Если длина арматуры с учетом части, расположенной у фундамента, составляет ($40D \leq H - P$), тогда $L = (40D + 450 + 2/3$ МЕН) мм; если ($40D > H - P$), тогда $L = (L1 + 450 + 2/3$ МЕН) мм. Отметим, что данные (P, D, H, \dots) и методика расчета берутся из результатов прочностных расчетов и в зависимости от проекта могут изменяться. Такие данные сохраняются, как глобальные переменные [8].

Согласно российским стандартам, основные требования, предъявляемые к формированию чертежей арматуры в колоннах, следующие: толщина бетона в защитном слое должна быть не меньше диаметра арматуры и не менее 10 мм ($P \geq 10$ мм, $P \geq D$) [9]. Продольные соединения арматур между собой зависят от типа арматуры, марки используемого бетона, шовных или бесшовных соединений. Длина нахлестки (перепуска) арматуры и длина анкеровки

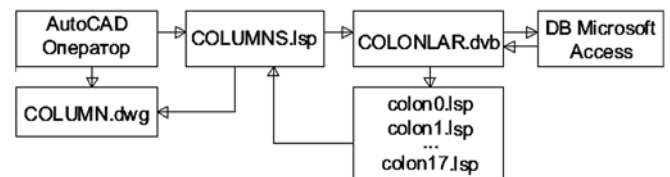


Рис. 2. Схема работы

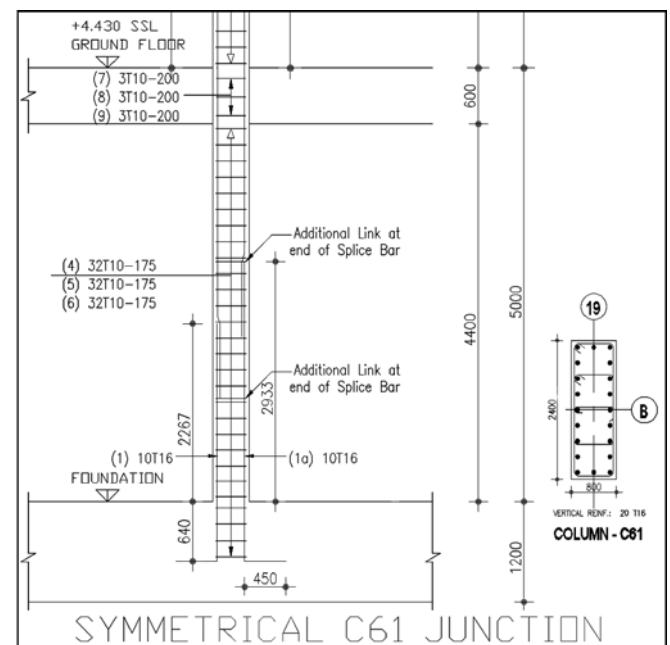


Рис. 3. Автоматически сгенерированная развертка колонны С61 в среде AutoCAD, после нажатия кнопки "OK" в основной форме

определяются из таблицы (исходя из относительной (в диаметрах) и абсолютной (в мм) длины анкеровки и нахлестки для различных случаев) [10]. Согласно таблице, арматура размещается вертикально в верхней части фундамента на глубину $34D$ ($34 \times 16 = 544 \approx 550$ мм) и анкеруется (наклоняется) на $28D$ ($28 \times 16 = 448 \approx 450$ мм). Таким образом, для арматуры диаметром 16 мм анкеровка по обоим стандартам составляет 450 мм; глубина вертикальной арматуры фундамента по российским стандартам составляет 550 мм, в то время как по европейским стандартам этот показатель равен 640 мм.

Расчеты, проводимые по правилам, определяемым стандартами, предназначены для детализовки арматуры каждого этажа в отдельности. Следует отметить, что в многоэтажном здании подобные расчеты для каждой колонны повторяются для каждого этажа.

В постсоветских странах для прочностных (статических) расчетов считается целесообразным использовать систему проектирования "ПК ЛИРА®-САПР" и другие им подобные. С их помощью подготавливают эскизы, которые затем должны быть доработаны в виде рабочих чертежей, пригодных для использования на строительной площадке [11]. В отличие от таких систем, представляемая нами система автоматически генерирует рабочие чертежи на основе выполненных прочностных расчетов.

Функциональное описание созданного программного модуля

В основной форме после выбора или введения названия колонны из базы данных Microsoft Access извлекаются такие ее характеристики, как ширина, длина, показатели этажности и высота фундамента. В правой части формы в зависимости от этажности для ввода балок, присоединенных к колонне, представлен список вспомогательных форм по числу этажей. При выборе нужного уровня раскрывается соответствующая этому уровню форма (см. рис. 1). В ней отражены показатель уровня, ширина и длина колонны, соответствующей данному уровню (длина и ширина колонны может меняться в зависимости от этажности). Характеристики арматуры колонны по умолчанию временно хранятся в переменных, и в форме их не видно.

Контроль над вводимыми переменными осуществляет программный модуль. При раскрытии вспомогательных форм, в случае, если будет изменен этаж, т. е. высота этажа будет определена не по умолчанию, а введена с клавиатуры, высота между этажами не должна превышать минимальный показатель [5].

Обычно в строительных чертежах строительные элементы индексируются в соответствии с каждым этажом. Если это верно и для текущего проекта, то сделав активной индексацию глобальных переменных в главной форме, можно осуществлять контроль элементов для каждого этажа. В частности, при формировании чертежей арматуры колонн в

соответствии с этой индексацией осуществляется контроль балок, присоединенных к колоннам. Если название балки указано неверно, на экране появляется информация об ошибочном вводе. В этом случае требуется заново выбрать балку или ввести ее название с помощью клавиатуры. Эта операция продолжается вплоть до выбора правильного варианта. Число заполняемых вспомогательных форм соответствует числу этажей строения. После ввода во вспомогательные формы информации о балках, присоединенных к колоннам, осуществляется возврат в главную форму и подтверждается операция ввода с помощью кнопки "ОК" [5].

Заключение

С помощью созданной системы в среде AutoCAD автоматически генерируются чертежи арматуры колонн и балок в двухмерном пространстве. Показатели колонн, балок и этажности, извлекаемые из базы данных, обрабатываются программой AutoLISP. Для каждого этажа автоматически проектируется чертеж колонны вместе с названиями балок, присоединенных к колоннам, длинами арматуры на этажах, показателями уровня и другими данными об арматуре (количество, диаметр, шаг распределения, длина и др.). В результате получается готовый чертеж.

Предложенную систему можно применить к другим элементам железобетонных конструкций. Таким образом, разработано программное обеспечение, с помощью которого можно в системе AutoCAD, используя базу данных, автоматически создать развертку колонн, балок, а также строить чертежи арматуры. Созданный программный модуль может быть использован вне зависимости от назначения в любых проектах, где применяются железобетонные конструкции.

Список литературы

1. Байков В. Н., Сигалов Э. Е. Железобетонные конструкции. М.: Стройиздат, 1991. 728 с.
2. <http://old.stroi.mos.ru/nauka/d27dr9329m128rr6099.html>
3. Гладков С. А. Программирование на языке АВТОЛИСП в системе САПР АВТОКАД. М.: Диалог-МИФИ, 1991. 93 с.
4. AutoCAD 2002 для конструктора. Искусство проектирования. Москва—Санкт-Петербург—Киев: DiaSoft ТИД, 2002. 944 с.
5. Əmiraslanova Z. N. Azərbaycan Milli Elmlər Akademiyasının xəbərləri // XXVIII cild. № 6. Bakı: Elm, 2008. S. 138—142.
6. Вяткин С. И., Долговесов Б. С. Создание баз данных на основе скалярных функций возмущения. Новосибирск: Институт автоматизации и электротехники СО РАН. URL: http://www.graphicon.ru/2006/wr52_64_Vyatkin.pdf
7. http://eurocodes.jrc.ec.europa.eu/doc/1110_WS_EC2/Presentations/05_EC2WS_Arrieta_Detailing
8. Əmiraslanova Z. N. Azərbaycan Milli Elmlər Akademiyasının xəbərləri // XXXI cild. № 3. Bakı: Elm, 2011. S. 164—168.
9. <http://www.google.az?url?sa=f&rct=j&q=&esrc=s&source=web&cd=1&cad=ria&ved=0CCKQFjAA&url=http%3A%2F%2Fwww.aliantechnologies.ru%2Fdocs%2Fsnip-52-01-2003.doc&ei=SecuUy2VAsm2hQei24DACA&usg=AFQjCNHnLjRljZ2oLsht1t2OCMpqCnNbfAhttp://podomostroim.ru/soedinenie-armatury-bez-svarkichast-pervaya/>
10. <http://fordewind.org/wiki/doku.php?id=%D0%B0%D0%BD%D0%BA%D0%B5%D1%80%D0%BE%D0%B2%D0%BA%D0%B0>
11. <http://www.liraland.ru/>

Automation of Reinforcement Design in Ferroconcrete Constructions

Reinforcement design is basically performed in AutoCAD based on static calculations of concrete structures. 'Reinforcement design' here refers to creating of reinforcement drawing with regard to its diameter, placement, step of distribution, protective coating etc. in compliance with corresponding standards. Characteristics of elements forming the structure (columns, beams, intermediate floors etc.) are referred to repeatedly by design engineer creating reinforcement design. Characteristics are classification of reinforcement forming the element, i. e. diameter, step of distribution at certain intervals from one another, crossing spans, tilting spans and other such characteristics. They are searched in several drawings depending on the project development method, which leads to waste of extra time. As is obvious, a database of characteristics needs to be created to minimize time and labour consumption. At the same time, solution to the problem provides the foundation for reduction of project cost due to economical use of labour force.

Automation of the design of reinforcement in ferroconcrete constructions in the construction sector is considered. An efficient solution to the problem using databases is proposed.

Keywords: AutoLISP, AutoCAD, database

References

1. Baykov V. N., Sigalov E. E. Zhelezobetonnye konstruktsiya [Reinforced concrete structure]. Moscow: Stroyizdat, 1991. 728 p.
2. <http://old.stroi.mos.ru/nauka/d27dr9329m128rr6099.html>
3. Gladkov S. A. Programmirovaniye na yazyke AVTOLISP v sisteme SAPR AVTOKAD [Programming AUTOLISP in system CAD AutoCAD]. Moscow: Dialog-MIFI. 1991. 93 p.
4. AutoCAD 2002 dlya konstruktora. Iskustvo proektirovaniya [AutoCAD 2002 for the designer. Art design.]. Moscow—St. Peterburg—Kiev: DiaSoft TID, 2002. 944 p.
5. Amiraslanova Z. N. Proceedings of the Academy of Sciences of Azerbaijan. v. XXVIII, no. 6 (2008). Baku. P. 138—142.
6. Vyatkin S. I., Dolgovosov B. S. Sozdanie baz dannykh na osnove skalyarnykh funktsii vozmushcheniya [Database Creation Based on Scalar Perturbation Functions]. Novosibirsk: Institute Automati-
- tion and Electrometry SB RAS. Russia. URL: http://www.graphicon.ru/2006/wr52_64_Vyatkin.pdf
7. http://eurocodes.jrc.ec.europa.eu/doc/1110_WS_EC2/Presentations/05_EC2WS_Arrieta_Detailing
8. Amiraslanova Z. N. Proceedings of the Academy of Sciences of Azerbaijan. V. XXXI, no. 3 (2011). Baku. P. 164—168.
9. <http://www.google.az/url?sa=f&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CCkQFjAA&url=http%3A%2F%2Fwww.aliantechnologies.ru%2Fdocs%2Fsnip-52-01-2003.doc&ei=SecuUv2VAsm2hQei24DACA&usq=AFQjCNHnLjRlJZ2oLsht1t2OCMPQcnNbfAhttp://podomostroim.ru/soedinenie-armatury-bez-svarki-chast-pervaya/>
10. <http://fordewind.org/wiki/doku.php?id=%D0%B0%D0%BD%D0%BA%D0%B5%D1%80%D0%BE%D0%B2%D0%BA%D0%B0>
11. <http://www.liraland.ru/>

УДК 338.3

Е. В. Новак, канд. экон. наук, председатель Совета директоров,
ГК "ИМПУЛЬС-ИВЦ", e-mail: nove@impuls-ivc.ru

Информационно-коммуникационные технологии: оценка эффективности

Информационно-коммуникационные технологии (ИКТ) используют в бизнесе уже давно и активно. Но каков эффект их применения для конкретной организации, не говоря уже об отраслях, городах, регионах, стране? Понятно не всегда. Сейчас очень нелегко найти методики, позволяющие качественно и точно провести расчет полезности использования оборудования, каналов связи, программного обеспечения, автоматизированных систем и комплексов.

В предлагаемой статье автор старается обозначить практическое значение оценки эффективности (или ее отсутствия) ИКТ и привести некоторые примеры приемов и методов, приглашая к дискуссии экономистов, специалистов отрасли, инвесторов.

Ключевые слова: эффективность информационно-коммуникационных технологий (ИКТ), технико-экономическое обоснование (ТЭО), инвестиции в ИКТ

"Если вы будете продолжать работать так, как работали, вы будете получать то, что получали".

Когда нужна автоматизация?

Как рассматривать ИКТ — как инвестиции или расходы?

Помогает ли автоматизация в работе с конкурентами?

Как определить эффект от автоматизированной системы?

1. С позиций процессного подхода

Часто представители IT-компаний, особенно западных, говорят о преимуществах, которые дает автоматизация управления, стратегической значимости ИКТ для бизнеса. Очевидно, что стратегия — это концентрированный и относительно долгосрочный план действия. Проект создания полноценной ERP-системы в стратегической карте действий может являться инициативой на уровне деловых процессов для достижения цели "Качество выходной продукции на отличном уровне". Конечно, автоматизированная система управления сама по себе преимуществом не является. Только в комплексе с другими инициативами она способна влиять на ключевые факторы успеха и создавать, тем самым, действительные конкурентные преимущества. Но это справедливо не для всех. В своей книге "Караоке-капитализм. Менеджмент для человечества" [1] Нордстрем К. и Риддерстралле Й. выражаются более жестко: "Множество корпораций не в состоянии осознать, что не относится к разряду ключевых способностей и никогда не будут. Сущность любой компании состоит в уникальности ее предложения потребителям. IT — хороший помощник, не более и не менее. Эта инфраструктура, электронная канализация в корпоративном туалете. Копирование других не приводит к устойчивой конкурентоспособности".

Процессный подход предполагает классификацию деловых процессов по значению для клиентов — контрагентов, дающих доход предпринятию либо клиентов по закону (например, "Об основах охраны здоровья граждан в Российской Федерации" [2]). Из всех процессов компании те из них, которые добавляют стоимость исходному продукту и имеют ценность для внешнего потребителя, называют основными. Конечно, вспомогательные процессы тоже в нормальной ситуации должны рождать ценность, но для своих же основных процессов, обеспечивая их маркетингом, продажами, ресурсами, учетом и т. д. Поэтому значение ИКТ для конкретной организации напрямую зависит от того, каким является процесс информатизации (автоматизации) — основным или вспомогательным.

Так как деятельность компаний одного рынка принципиально похожа и отличия проявляются только в деталях, совершенно конкретный набор и логика процессов и функций присущи каждой от-

дельно взятой отрасли. Безусловно, участники одной индустрии, конкурируя между собой, ведут поиск преимуществ, но в целом логика примерно одна. Поэтому говорить о роли ИКТ нужно в отношении следующих определенных видов экономической деятельности:

1) финансовая деятельность, торговля (особенно электронная), некоторые виды госуслуг и т. п. — в них ИКТ больше инструмент основных процессов;

2) строительство, сельское хозяйство, рыболовство и т. д., где ИКТ в большей степени составная часть вспомогательных процессов.

Но пойти можно и дальше, оценивая роль ИКТ по их конкретным системам и средствам, применяемым в организациях. Например, если бухгалтерская программа Интернет-магазина, как средство вспомогательного процесса, даст сэкономить на управленческих расходах, то получить дополнительную выручку она вряд ли поможет. Но более быстрая и более удобная работа сайта этого магазина точно приведет новых покупателей, улучшит имидж поставщика и приведет к росту доходов.

В связи с этим наилучший вариант — оценивать важность и результативность конкретной автоматизированной системы (АС или АСУ) в рамках конкретной отрасли (см. таблицу).

Условная классификация роли ИКТ в деятельности организаций

Вид технологии	Стройкомплекс	Госсектор	Торговля
Интернет-сайт	Вспомогательная	Основная	Основная
АСУ технологическим (основным) процессом	Основная	Основная	Основная
АСУ персоналом	Вспомогательная	Вспомогательная	Вспомогательная
АСУ финансами	Вспомогательная	Вспомогательная	Вспомогательная
АСУ снабжением	Основная	Вспомогательная	Основная
Локальная или распределенная вычислительная сеть	Вспомогательная	Основная	Основная

Систематизация отраслей, технологий, процессов автоматизации имеет несколько прикладных значений. Одно из них — определение целей и средств технико-экономического обоснования автоматизации.

2. От теории к практике

Наверняка многим известны методики инвестиционного анализа в различных вариациях, сводящиеся к сопоставлению полученного финансового результата с объемом вложенных инвестиций или активов (ROI, англ. *Return on Investment*). Похо-

жим способом подсчитывается чистый дисконтированный доход (NPV, англ. *net present value*) как сравнение дисконтированных поступлений и вложений (расходов) денежных средств.

Вместе с тем, неплохо описан в деловых и научных трудах подход, оценивающий совокупную стоимость владения (ТСО, англ. *total cost of ownership*), основная идея которого состоит в подсчете не только первоначальных затрат на приобретение какого-либо актива, но и расходов на его эксплуатацию и развитие в течение жизненного цикла. Этот метод, например, хорош в случае сравнения двух автоматизированных систем разных производителей.

Приверженцы методики сбалансированной системы показателей, к коим автор относит и себя, справедливо применили аналогичный подход и для ИКТ. Современные технологии, как правило, влияют на предприятие в целом: финансы, взаимоотношения с контрагентами, внутренние процессы, персонал, его развитие и т. д.

Но, хотя приведенные методики и хороши, они разработаны для инвестирования вообще. Основная же трудность анализа инвестиций в ИКТ состоит в расчете экономического эффекта — дополнительного финансового результата (размера прибыли). Это достигается либо увеличением объема выручки при неизменном уровне расходов или снижением себестоимости, управленческих и коммерческих затрат. Как мы уже сказали, ИКТ можно использовать в основной и вспомогательной деятельности компании. Совершенствование основных процессов ведет к росту прибыльности, эффективности (выручка растет и/или себестоимость единицы продукта снижается). Развитие вспомогательных процессов снижает лишь удельные управленческие или коммерческие затраты на единицу оказываемых внутренних услуг (обслуживание одного сотрудника, бухгалтерскую операцию, тонну материала и т. д.).

Однако сегодня чаще всего решения о сделках в области ИКТ не имеют четкого определения конкретных целей, хорошо просчитанного экономического обоснования, порой, выполняются исходя из принципа "чтобы было". Максимум, чем обеспокоен заказчик, так это проведением сравнительного анализа различных ИКТ и их поставщиков, да и то не всегда. Главные причины такого положения вещей следующие:

- реальная сложность темы, предопределенная технической стороной ИКТ, часто вспомогательным характером процесса автоматизации (влияет на деятельность в основном косвенно), специфичным анализом;
- сравнительно низкий уровень экономической или как модно говорить бизнес-подготовки ИКТ-специалистов как главных представителей заказчика;
- недоступность профессиональных инструментов анализа как в свободном распространении, так и у поставщиков ИКТ-решений, особенно для малых и средних предприятий;

Несистемный взгляд заказчика в ИКТ-сделках ведет к увеличению времени принятия решения, недостаточной обоснованности представлений о возможных расходах, а отсюда, как следствие, и к разочарованию инвесторов-заказчиков в ИКТ и поставщиках, что приводит к поверхностным выводам "с вами всегда так, информатизация ничего не дает" и т. п.

3. С чего же начинать?

Начинать следует с общего управленческого аудита деятельности предприятия, со сбора и систематизации всех требуемых первичных данных. К сожалению, сегодня в большинстве компаний отсутствуют собственные реальные и надежные технологии планирования, учета и анализа ИКТ-вложений. Неточно, а порой и просто неверно оценивая эффективность ИКТ, заказчики принимают ошибочные решения, ведущие фактически к потере денежных средств, многократному повышению рисков, связанных с внедрением ИКТ. Выход из такой ситуации лежит только в системном технико-экономическом обосновании (ТЭО), пусть и от потенциально заинтересованной стороны (исполнителя). Но в процессе коммерческой предпроектной работы он — первый, кто реально может помочь инвестору в принятии рационального, осознанного решения о проекте разработки, внедрения или обслуживания автоматизированной системы.

При этом целесообразно проводить ТЭО автоматизации, уже имея в своем опыте определенный портфель успешных организационных проектов по "наведению порядка" в управлении. Предшествующий опыт по реорганизации, реструктуризации, бизнес-моделированию, разработке бизнес-стратегий и т. д. станет надежным фундаментом для принятия необходимых решений.

Разработка технико-экономического обоснования может идти по разным альтернативным вариантам:

А. Во главу угла ставится первичность средств автоматизации. В этом случае заказчик ставит задачу технически и экономически обосновать правильность предварительного выбора средства автоматизации для достижения определенных целей. Его интересует: "Будет ли эффективной такая автоматизация?"

В. Главным являются "слабые места" (скрытые резервы) эффективности. В этом случае заказчик ставит задачу подобрать, а также технически и экономически обосновать конкретное средство автоматизации для достижения определенных целей. Его интересует: "Какая автоматизация необходима для достижения нужного эффекта?"

Рассмотрим несколько примеров. Внедрение АСУ в строительных компаниях, например, может приводить к снижению злоупотреблений — снижению материалоемкости, фондоемкости, зарплатоемкости и т. д. (бригады, материалы, строительные машины и механизмы будут использоваться только на объектах компании). В госсекторе необходимо ориентироваться прежде всего на операционную эффективность,

политическую отдачу (удовлетворение социальных ожиданий населения, исполнение решений и рекомендаций вышестоящего руководства и т. п.).

Логика оценки эффективности: если ИКТ будут, то это приведет к..., если ИКТ не будет, то ...

Разрабатывая ТЭО, необходимо исходить из поставленных заказчиком конкретных, реальных, не только хорошо фиксируемых, но и измеримых целей автоматизации. Часто от поставщика ИКТ требуется помощь в определении этих целей заказчика, формулировании четких критериев оценки. Автоматизация — это ведь элемент совершенствования системы управления. Источник ее эффективности — исключение потерь, задействование скрытых резервов. Поэтому экономическое обоснование может строиться исходя из проекта, программы сопровождения, которую предварительно заказывает организация. В нем нужно уделять сбалансированное внимание и результатам, и затратам. Только тогда заказчик сможет принять объективное взвешенное решение. Принцип один — чем выше затраты на неавтоматизированные процессы сегодня, тем выше потенциальный возврат на инвестиции завтра. Таким образом, при разработке проекта автоматизации необходимо рассматривать самые ресурсоемкие, самые дорогие рабочие места (должности), где велика доля и стоимость ручных операций и, как следствие, высок потенциал повышения производительности труда, либо снижение злоупотреблений или их вероятности. Исходя из такого взгляда на ситуацию, по каждой отрасли можно определить примерный список должностей, деловых процессов, функций, процедур и тому подобного, где автоматизация позволит принести наибольшую выгоду, ведь возврат на инвестиции на каждом предприятии индивидуален. То есть минимальная адаптация методики потребует всегда.

Общая характеристика в ТЭО дается как организации инвестора в целом, так и автоматизируемым деловым процессам, функциям, подразделениям и другим объектам. При этом особое внимание уделяется показателям, меняющимся в результате автоматизации (трудоемкость управленческих операций, оборачиваемость активов, финансовые потери от злоупотреблений и т. п.). Источником информации для создания такой характеристики могут быть в том числе актуальные отчеты консультационных и аудиторских фирм, работавших с данным заказчиком.

Характеризуя систему управления, необходимо определить:

- ее название;
- стиль управления;
- основные элементы корпоративной культуры;
- указать фактическую органиграмму с функциями;
- выделить наиболее актуальные для автоматизации деловые процессы: основные и вспомогательные;
- функции, их процедуры;
- подразделения;
- рабочие места и другие объекты.

Помимо этого важно выявить и указать фактические методы управления, средства управления. В характеристике недостатков необходимо оперировать фактической информацией заказчика, осматривательно и обоснованно описывать недостатки, прежде всего, устранимые за счет автоматизации. Оценка потерь проводится по показателям данного объекта ТЭО (автоматизации), выбираемым из приведенного исполнителем анализа списка. При отборе конкретных показателей целесообразно исходить из принципа Парето: 20 % важнейших показателей дают 80 % данных.

Готовность объекта автоматизации определяется:

- а) наличием, хотя бы предварительных, целей и задач автоматизации объекта;
- б) стабильным функционированием объекта (объект не находится в состоянии кардинального реформирования или создания);
- в) наличием подсистем нижнего уровня (компьютеры, сети);
- г) достаточным описанием объекта (внешние, внутренние документы и регламенты, бизнес-модели и т. п.). Например, для регулируемых предметных областей (бухгалтерия, кадры) делового программного обеспечения внешними документами являются многочисленные нормативные акты;
- д) пониманием персонала объекта потребности и актуальности автоматизации своего объекта;
- е) наличием достаточных финансовых и трудовых ресурсов для автоматизации;
- ж) другими обстоятельствами на усмотрение исполнителя и заказчика.

Варианты готовности объекта автоматизации: "готов"; "условно готов"; "не готов". При выборе варианта "условно готов" необходимо четко по элементам объекта описать условия обретения полной готовности и по возможности мероприятия, при выполнении которых, готовность повысится. При указании варианта "не готов" важно обосновать причины такого вывода, а также отметить вероятные негативные последствия автоматизации при этом состоянии объекта.

Возможные цели (результаты) автоматизации:

1. Необходимость повышения отказоустойчивости АС в связи с моральным и физическим устареванием оборудования, моральным устареванием ПО.
2. Достижение соответствия АС (подсистемы АС) новым требованиям объекта, достижение соответствия какой-либо подсистемы АС новым требованиям других подсистем АС (например, предстоящее внедрение коллективной САПР потребовало новой скорости передачи данных, а значит изменения компьютерных сетей).
3. Появление новых объектов автоматизации — подразделений, процессов, функций, сотрудников.
4. Повышение производительности труда (снижение трудоемкости) управленческих работников (ИТР), производственного персонала.
5. Снижение (исключение) потерь от упущений и злоупотреблений в профессиональной деятельности

сотрудников заказчика, связанных с недостатком и(или) неоперативностью информации и контроля.

6. Повышение эффективности основных и вспомогательных процессов (проектирование, календарное планирование, материально-техническое обеспечение, управление финансами, персоналом и т. д.).

7. Увеличение оборачиваемости запасов (отношение выручки либо себестоимости к средним остаткам запасов) за счет их снижения.

8. Увеличение оборачиваемости незавершенного производства (отношение выручки либо себестоимости к средним остаткам незавершенного производства) за счет его снижения.

9. Увеличение оборачиваемости дебиторской задолженности (отношение выручки либо оборота по задолженности к средним остаткам задолженности) за счет снижения задолженности.

10. Повышение качества создания, обработки, передачи и хранения, других действий над производственной и управленческой информацией. Другие цели, определяемые заказчиком, предлагаемые исполнителем.

Ограничения, задаваемые заказчиком, по созданию АС могут носить финансовый, временной, технический и иной характер.

Автоматизировать в первую очередь можно те деловые процессы, функции и подразделения, улучшение которых принесет наибольший эффект.

При описании требований к характеристикам подробно описывается состояние автоматизируемых деловых процессов, функций, подразделений и других объектов на настоящий момент (как есть) и предполагаемое состояние в результате автоматизации (как будет).

Среди дополнительных требований к АС могут быть вспомогательные и незначительные характеристики. Например, если необходима комплексная автоматизация и основное требование — улучшение ликвидности баланса за счет повышения оборачиваемости активов, требование соответствия серверного оборудования новому деловому программному обеспечению будет носить дополнительный (вспомогательный) характер.

Источники эффективности должны быть привязаны к реальной жизни заказчика и его индивидуальным особенностям. Показатели для конкретного ТЭО подбирают в зависимости от отраслевой принадлежности, целей и вида автоматизации (стадийная, элементная). Важно помнить, что между вложениями в ИКТ и рядом экономических показателей, как правило, имеется корреляционная связь, но не прямая зависимость. Желательно, чтобы в ТЭО прямых количественных экономических показателей было не менее 60 %, а косвенных качественных — не более 40 %. К экономически измеряемым косвенным выгодам для уточнения их вклада стоит применять понижающие коэффициенты 0,3...0,7. Ведь сэкономленный рабочий час сотрудника не весь пойдет на продуктивную деятельность.

Оценка инвестиций может проводиться в различных разрезах:

- очереди (приоритетность этапов работ при необходимости определяет заказчик);
- периоды (по годам, кварталам, месяцам);
- цели, результаты автоматизации (десять и более);
- стадии (проектирование АС, проектирование и монтаж инженерных систем, проектирование и установка компьютерных систем, внедрение делового ПО, сопровождение АС и т. д.);
- подсистемы, элементы АС (инженерные подсистемы, компьютерные подсистемы, деловое ПО, с большей либо меньшей детализацией);
- объекты автоматизации (офисные и производственные помещения, деловые процессы, функции, подразделения, должности и другие объекты);
- направление затрат (оборудование, программное обеспечение, расходы на сотрудников заказчика в связи с автоматизацией, расходы на внешних подрядчиков и т. д.);
- характер расходов средств заказчика (инвестиции, текущие расходы, расходы будущих периодов);
- в других разрезах.

Обобщающие показатели сводят данные о результатах и затратах по различным направлениям автоматизации. Например, эффект от автоматизации нескольких деловых процессов суммируется, и рассчитывается совокупный эффект. Такие сведения должны позволять делать общие выводы о целесообразности, обоснованности автоматизации.

Выводы об автоматизации должны находиться в пределах от необходимого (ликвидация "слабых мест") до достаточного (целесообразность). Необходимость соответствует минимальным целям автоматизации, целесообразность — максимальным. Необходимость обосновывается некими критическими параметрами ситуации заказчика, например, учащение отказов в работе серверов в связи с увеличением информации потребовало модернизации всего центра обработки данных (ЦОД), а целесообразность должна апеллировать, в первую очередь, к позитивному изменению технико-экономических показателей.

Важно, чтобы предложения по организации и управлению касались только достаточно изученных объектов автоматизации. Здесь нелишне вспомнить Гиппократу с его "не навреди". Авторы ТЭО должны учитывать и соизмерять свою квалификацию по этим ответственным вопросам. Рекомендации по автоматизации должны содержать конкретные мероприятия (кому, что, для чего нужно сделать), учитывать фактическую ситуацию, условия заказчика.

Сравнение плановых целей и итоговых результатов (по расчетам исполнителя ТЭО) наиболее реально только в том случае, если последние были четко заданы заказчиком перед ТЭО. В противном случае объективность сопоставления резко снижается. При генерации исполнителем и результатов, и целей будет лучше, если это сопоставление проводили разные специалисты исполнителя.

В итоге исполнитель ТЭО должен четко сообщить решение по автоматизации данного объекта, группы объектов: автоматизация требуется, автоматизация не требуется, и почему.

Так как конкретные проектные решения являются результатом последующих этапов, на этом уровне работы с заказчиком могут определяться только предварительные рекомендации.

Исходя из целей и задач, текущего состояния средств автоматизации управления необходимо предложить комплексную (в целом АС), элементную (подсистемы, элементы) либо стадийную автоматизацию (создание, сопровождение). Необходимо определить и рекомендовать условия работы подсистем нижнего уровня, даже если заказчик ориентирует исполнителя только на некоторые подсистемы. Например, если требуется установить более современное сетевое оборудование, способное передавать информацию на скорости 1 Гбит/с и выше, необходимо определение адекватных требований и для кабельной сети вне зависимости от того, кто будет выполнять работы: исполнитель, заказчик ТЭО или третье лицо. То есть проект установки компьютерных систем может потребовать предварительного (параллельного) выполнения проекта монтажа инженерных систем.

Под организационной и функциональной структурой АСУ подразумеваются изменения структуры и процедур ИКТ-отдела заказчика, а также подразделений, затрагиваемых автоматизацией. То есть для предполагаемого проекта по внедрению делового ПО может стать необходимым ввести дополнительную должность сотрудника управленческой системы и новую функцию ИКТ-отдела (организация развития, модернизации и сопровождения информационной системы управления).

При подготовке рекомендаций по составу и характеристикам подсистем желательно придерживаться принятой заказчиком терминологии и ориентироваться на исходные ИКТ-продукты, поставляемые на рынке. Важно определить место и роль существующих подсистем и элементов. Все они, по возможности, должны эффективно использоваться в новой автоматизированной системе. А при невозможности использования желательно выдать заказчику рекомендации по их реализации на сторону либо обосновать утилизацию. К основным видам обеспечения АС относятся: информационное, программное, техническое, организационное.

Максимальная объективность разработчиков — главное условие успешности проекта автоматизации. Авторы ТЭО должны исходить прежде всего из интересов заказчика. Особое внимание нужно обращать на подготовительные к автоматизации этапы — управленческого (по различным предметным областям) консультирования, бизнес-моделирования, в целом "наведения порядка" на объектах автоматизации. Нужно помнить, что "автоматизация хаоса дает только автоматизированный хаос".

В проекте автоматизации могут участвовать следующие организации:

- организация-заказчик (инвестор, пользователь), для которой создается АС и которая обеспечивает финансирование, приемку работ и эксплуатацию, а также выполнение отдельных работ по созданию системы;
- организация-генпроектировщик объекта автоматизации;
- организации-проектировщики различных частей проекта объекта автоматизации для проведения строительных, электротехнических и других подготовительных работ, связанных с созданием АС;
- организация-разработчик, которая осуществляет работы по созданию АС, предоставляет заказчику совокупность научно-технических услуг на разных стадиях и этапах создания, а также разрабатывает и поставляет различные программные и технические средства;
- организация-поставщик, которая изготавливает и поставляет программные и технические средства по заказу разработчика или заказчика;
- строительные, монтажные, наладочные организации;
- обслуживающие организации, отвечающие за стабильную нормативную эксплуатацию и поддержку систем.

В зависимости от условий создания АСУ возможны различные варианты совмещения функций заказчика, разработчика, поставщика и других организаций, участвующих в работах по созданию или обслуживанию системы.

Организация автоматизации является рациональной тогда, когда поставленные цели достигаются в кратчайшие сроки с максимальным качеством и минимальным для таких сроков и качества бюджетом. Конечно, при этом должны учитываться общие закономерности, технологии, риски и другие факторы.

Источниками финансирования автоматизации могут быть:

- собственные средства заказчика (прибыль, уставный капитал и т. д.);
- заемные средства заказчика;
- долгосрочные кредиты, займы;
- краткосрочные кредиты, займы, кредиторская задолженность, резервы предстоящих расходов и другие краткосрочные обязательства.

Если направляемые на автоматизацию средства представляют собой инвестиции (долгосрочные вложения во внеоборотные активы), например, внедрение новой системы управления проектами и календарного планирования, желательно задействовать на финансирование собственные средства либо долгосрочные кредиты и займы. При нормативной деловой активности, ликвидности баланса и состоянии активов можно рекомендовать банковское целевое кредитование. Если же затраты на автоматизацию являются текущими расходами либо расхо-

дами будущих периодов, финансировать их можно и за счет краткосрочных обязательств. В таком случае более внимательно нужно рассматривать изменение себестоимости, рентабельности, выполнение бюджетов затрат сейчас и в будущем в связи с дополнительными расходами на автоматизацию.

Во сколько же обойдется технико-экономическое обоснование применения информационно-коммуникационных технологий? В зависимости от поставленных задач и объема инвестиций планка затрат на анализ составит 0,2...2 % от предполагаемых инвестиций. И лучше обойтись малыми потерями, поняв ошибку и скорректировав план, чем непродуманно рисковать многими миллионами и десятками миллионов рублей или долларов.

Сейчас в экономике масса возможностей для выгодных вложений, так давайте инвестировать денежные средства в наиболее эффективные проекты!

1. **Риддерстрале Й., Нордстрем К.** Караоке-капитализм. Менеджмент для человечества. СПб.: Стокгольмская школа экономики в Санкт-Петербурге, 2004.
2. **Об основах** охраны здоровья граждан в Российской Федерации. Федеральный закон от 21 ноября 2011 г. № 323-ФЗ.
3. **Бузырев В. В., Нужина И. П.** Анализ и диагностика финансово-хозяйственной деятельности строительного предприятия: учебник. М.: Проспект, 2010.
4. **ГОСТ 24.202—80** СТДАСУ. Требования к содержанию документа "Технико-экономическое обоснование создания АСУ".
5. **Ильин В. Н., Плотников А. Н.** Сметное ценообразование в строительстве. Ростов н/Д: Феникс, 2011.
6. **Карр Н.** Блеск и нищета информационных технологий. Почему ИТ не являются конкурентным преимуществом. М.: Секрет фирмы, 2005.
7. **Николаева О. Е., Шишкова Т. В.** Классический управленческий учет. М.: КомКнига, 2008, 2010.
8. **Новак Е. В.** Преимущество использования специализированных программных средств при управлении строительными проектами // Управление проектами и программами. 2010. № 1(21).

E. V. Novak, Chairman of the Board of Directors, GC "IMPULS-IVC"
E-mail: nove@impuls-ivc.ru

Information and Communication Technologies: Measuring of the Effectiveness

Information and communication technologies (ICT) has been actively used in the business. However, whether became easier to understand the effect of their use for a particular organization, not to mention industries, cities, regions, country? Today, with great difficulty, you can find techniques to enable reasonably accurate count benefit from the use of equipment, communication channels, software, automated complexes and systems.

In this article the author tries to identify practical value assessment of the effectiveness of ICT (or lack of it), and give some examples of techniques and methods inviting to the discussion of economists, industry experts, investors.

Keywords: *efficiency of information and communication technologies (ICT), a feasibility study (FS), investments in ICTs*

References

1. **Ridderstrale J., Nordstrom K.** Karaoke-kapitalizm. Menedzhment dlja chelovechestva. SPb.: Stokgol'mskaja shkola jekonomiki v Sankt-Peterburge, 2004.
2. **Ob osnovah** ohrany zdorov'ja grazhdan v Rossijskoj Federacii. Federal'nyj zakon ot 21 nojabrja 2011 g. N 323-FZ.
3. **Buzyrev V. V., Nuzhina I. P.** Analiz i diagnostika finansovohozjajstvennoj dejatel'nosti stroitel'nogo predpriyatija: uchebnik. M.: Prospekt, 2010.
4. **GOST 24.202—80** STDASU. Trebovanija k soderzhaniju dokumenta "Tehniko-jekonomicheskoe obosnovanie sozdanija ASU".
5. **Il'in V. N., Plotnikov A. N.** Smetnoe cenoobrazovanie v stroitel'stve. Rostov n/D: Feniks, 2011.
6. **Karr N.** Blesk i nisheta informacionnyh tehnologij. Pochemu IT ne javljajutsja konkurentnym preimushhestvom. M.: Sekret firmy, 2005.
7. **Nikolaeva O. E., Shishkova T. V.** Klassicheskij upravlencheskij uchet. M.: KomKniga, 2008; Izd-vo LKI, 2010.
8. **Novak E. V.** Preimushhestvo ispol'zovanija specializirovannyh programnyh sredstv pri upravlenii stroitel'nymi proektami. *Upravlenie proektami i programmami*. 2010. № 1(21).

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *Е.В. Конова*.

Корректор *Е.В. Комиссарова*.

Сдано в набор 06.06.2014. Подписано в печать 14.07.2014. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ ИТ814. 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 "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".

119071, г. Москва, Ленинский пр-т, д. 19, стр. 1.