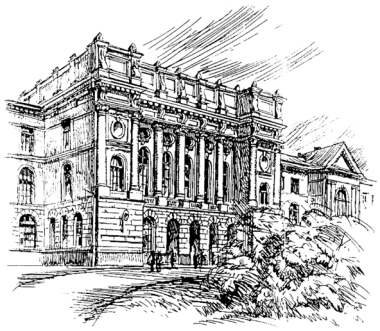
5(65)/2008



Научно-технические ведомости СПбГПУ

Информатика. Телекоммуникации. Управление.

Санкт-Петербург. Издательство Политехнического университета

Федеральное агентство по образованию
Санкт-Петербургский государственный политехнический университет

Редакционная коллегия

Главный редактор
Васильев Юрий Сергеевич
академик РАН, доктор технических наук, профессор, президент СПбГПУ, заслуженный деятель науки и техники РФ

Зам. гл. редактора
Рудской Андрей Иванович
член-корреспондент РАН, доктор технических наук, профессор, проректор по научной работе СПбГПУ

Зам. гл. редактора
Бабкин Александр Васильевич
доктор экономических наук, профессор, директор научно-исследовательского комплекса СПбГПУ

Ответственный секретарь
Екимова Маргарита Матвеевна
кандидат технических наук, доцент

Члены редколлегии

Арсеньев Дмитрий Германович
доктор технических наук, профессор, проректор по академической и международной деятельности СПбГПУ

Башкарев Альберт Яковлевич
доктор технических наук, профессор, директор инновационно-инвестиционного комплекса СПбГПУ

Боронин Виталий Николаевич
доктор технических наук, профессор, заслуженный деятель науки РФ

Глухов Владимир Викторович
доктор экономических наук, профессор, проректор по учебной работе СПбГПУ, заслуженный деятель науки РФ

Дегтярева Раиса Васильевна
доктор исторических наук, профессор, ученый секретарь Ученого совета СПбГПУ, заслуженный работник высшей школы РФ

Иванов Александр Васильевич
доктор технических наук, начальник Управления информационных ресурсов СПбГПУ

Ильин Владимир Иванович
доктор физико-математических наук, профессор, заслуженный работник высшей школы РФ

Козлов Владимир Николаевич
доктор технических наук, профессор, проректор по УМО СПбГПУ, заслуженный работник высшей школы РФ

Кузнецов Дмитрий Иванович
кандидат технических наук, доктор философских наук, член Союза писателей России

Лопота Виталий Александрович
чл.-кор. РАН, доктор технических наук, профессор, президент и генеральный конструктор РКК "Энергия", директор – главный конструктор ЦНИИ РТК, заслуженный деятель науки РФ

Федотов Александр Васильевич
доктор экономических наук, профессор, научный руководитель Института государственного управления и информатизации

Перепечатка материалов, опубликованных в нашем журнале, приветствуется. Ссылка на авторов цитируемых и перепечатываемых статей и на журнал "Научно-технические ведомости СПбГПУ" обязательна.

Журнал включен в "Перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученой степени доктора и кандидата наук".

Подписной индекс **18390** в каталоге "Газеты. Журналы" ОАО Агентства "Роспечать".

© Санкт-Петербургский государственный политехнический университет, 2008

Содержание

Телекоммуникационные системы и компьютерные сети

Лукьянова Л.М. *Методика структурно-целевого анализа и синтеза организационно-технических комплексов* 7

Савельев М.В., Аликов А.Ю., Федосеев С.В. *Вероятность связности сети с аддитивной структурой* 14

Тихомиров В.А., Котляров В.П. *Верификация программного обеспечения при интеграции новой функциональности в существующую систему* 20

Мезенцева О.С., Алексеев А.И. *Применение аппарата пороговых схем и модулярной арифметики для повышения производительности распределенных вычислительных систем* 26

Дронь В.О., Марков А.М., Наумов А.С. *Сокращение объема вычислений при случайном поиске в задачах пеленгования* 31

Березкин А.А. *Построение оптимальных нейронных декодеров блоковых кодов* 34

Радиотехника и связь

Кудряшов Д.А., Цикин И.А. *Вероятностно-временные характеристики двунаправленной радиосистемы передачи тревожных извещений* 41

Сидоров Ю.Е., Наспер М. Л. *Оценка эффективности обнаружителя радиосигналов в каналах с замираниями при разнесенном приеме* 46

Былин А.Р., Былина М.С., Глаголев С.Ф. *Пути повышения разрешающей способности импульсных измерений* 49

Сидоров Ю.Е., Лаврентьев Н.В. *Обнаружитель радиосигналов в каналах связи с неизвестными характеристиками замираний* 54

Рыжов В.А., Кипоть В.Л., Биряльцев Е.В. *Параметризация спектров в технологии низкочастотного сейсмического зондирования на основе вейвлет анализа* 58

Данилович Д.А. О*птимизация характеристик антенн и высот их подвеса на многоинтервальных цифровых радиорелейных линиях* 62

Бузюкова И.Л., Гайдамака Ю.В. *Исследование сигнальной нагрузки интеллектуальных сетей связи с узлами в разных часовых поясах* 67

Проблемы передачи и обработки информации

Максимов М.И., Сидорова Н.А. *Принципы работы и практическая реализация речевого кодека на основе метода линейного предсказания с возбуждением от кода* 75

Кукунин Д.С. *Модель системы с адаптивным выбором кода* 81

Кукунин Д.С., Когновицкий О.С. *Методика оценки качества канала в процессе передачи данных* 86

Вычислительные машины и программное обеспечение

Новиков Ф.А., Тихонова У.Н. *Определение проблемно-ориентированных языков интерпретируемыми автоматами* 93

Дунаев А.В., Ларченко А.В., Бухановский А.В. *Инструментальная оболочка поддержки принятия решений разработчика высокопроизводительных приложений в Грид-среде* 98

Рачинский Е.С. *Программная среда для оценки производительности клиент-серверных систем при нагрузочном окружении* 105

Старостенко А.С., Самочадин А.В. *Программные средства дистанционного управления очередями потребителей* 111

Куракин А.С. *Использование мультиагентного подхода при разработке комплексов программ систем корпоративного дистанционного образования* 114

Ицыксон В.М., Моисеев М.Ю., Цесько В.А., Карпенко А.В. *Исследование систем автоматизации обнаружения дефектов в исходном коде программ* 119

Ицыксон В.М., Захаров А.В., Ахин М.Х., Мяснов А.В. *Автоматическое обнаружение дефектов программных систем на основе метода проверки модели* 127

Системный анализ и управление

Савельев М.В., Аликов А.Ю., Федосеев С.В. *Синтез простых структур и наборов импульсных управляющих воздействий* 134

Чебыкин С.В. *Методика производственного планирования выпуска листового стекла* 139

Приборы. Материалы электронной техники

Лавров А.П., Молодяков С.А. *Возможности построения процессоров обработки сигналов в виде гибридных оптоэлектронных микросхем* 144

Зайцев Э.Ф., Гузенко К.В., Молдаванова Е.Н. *Анализ волн в ферритодиэлектрическом волноводе вариационным методом* 152

Моделирование: методы, алгоритмы, технологии

Макатов М.Р., Вердиев М.М. *Информационная модель описания объектов и ситуаций проблемной среды* 156

Исмаилов Ш.А. Магомедов Ш.Г. *Алгоритмы и структуры преобразования числовых данных из позиционной системы счисления в систему остаточных классов* 159

Новиков К.В. *Быстродействующий алгоритм сегментации изображений на основе спектральных свойств анизотропных областей* 169

Суров А.Н., Игизьянова Н.А., Потапов В.И. *О сходимости решения разностной схемы в математической модели генерации тепловой энергии* 174

Павлов Д.А. *Алгоритм построения расширенного универсального базиса Гребнера* 179

Манцеров Д.И. *Система верификации для параметрических классов задач по математике* 183

Канунникова Е.А., Колтунов Л.И., Потапенко А.Н. *Численное моделирование распределенных электротехнических систем в полубезграничных областях на основе метода инверсии* 189

Ушмоткин А.С., Коржик В.И. *Оценка эффективности использования системы цифровых "водяных" знаков для изображений в условиях коалиционных атак* 198

Суслов Н.В. *Принципы построения виртуального обучающего пространства* 204

Contens

Telecommunication systems and computer networks

Lukianova L.M. *Technique of the structurally-target analysis and synthesis of organizational-technical complexes* ... 7

Saveljev M.V., Alikov A.J., Fedoseev S.V. *Probability of connectivity of the network with additive structure* ... 14

Tikhomirov V.A, Kotljarov V.P. *Verification of the software at integration of new functionality into existing system* ... 20

Mezentseva O.S., Alekseev A.I. *Improving distributed computing systems performance using threshold schemes and modular arithmetic's algorithms* ... 26

Dron V.O., Markov A.M., Naumov A.S. *Reduction of calculations volume by random search in problems of direction finding* ... 31

Beryozkin A.A. *Construction of optimal neuronal decoders of block codes* ... 34

Radio engineering and link

Kudryashov D.A., Tsikin I.A. *Probability-time characteristics of duplex alarm radio systems* ... 41

Sidorov Yu.E., Nasper M.L. *Evaluation of the effectiveness of a radio signals detecting device in channels with the fade during diversity reception* ... 46

Bylin A.R., Bylina M.S., Glagolev S.F. *Ways of increase of resolution of pulse measurements* ... 49

Sidorov Yu.E., Lavrentiev N.V. *The detector radio signals in the channels of communication with unknown character of fading* ... 54

Rizhov V.A., Kipot V.L.,Birialtsev E.V. *The estimation of parameters of spectrum for low-frequency seismic sensing technology on basis of wavelet analysis* ... 58

Danilovitch D.A. *The optimization of antenna characteristics and heights on multihop digital microwave links* ... 62

Buzyukova I.L., Gaidamaka Y.V. *Analysis of signaling load in national intelligent network with different time zones* ... 67

Problems of transfer and processing of the information

Maximov M.I., Sidorova N.A. *Principles of operation and practical implementation of code excited linear predictive speech codec* ... 75

Kukunin D.S. *Model of system with an adaptive choice of a code* ... 81

Kukunin D.S., Kognovitskiy O.S. *Technique of the estimation of quality of the channel in the course of data transmission* ... 86

Computers and the software

Novikov F.A., Tikhonova U.N. *Definition of domain-specific languages by interpreted automata* ... 93

Dunaev A.V., Larchenko A.V, Boukhanovsky A.V. *Integrated development environment of decision support for high-performance applications developer in Grid* 98

Rachinskiy E.S. *Software framework for performance analysis of client-server systems under load* . 105

Starostenko A.S., Samochadin A.V. *Software of remote control by queues of consumers* 111

Kurakin A.S. *Usage multiagental the approach by development of complexes of programs of systems of corporate remote derivation* 114

Itsykson V.M., Moiseev M.J., Tsesko V.A., Karpenko A.V. *Research on tools for automation of defects detection in program source code* 119

Itsykson V.M., Zakharov A.V., Akhin M.K., Myasnov A.V. *Automatic model checking based defect detection.* 127

The system analysis and management

Saveljev M.V., Alikov A.J., Fedoseev S.V. *Synthesis of simple structures and sets of pulse managing influences* 134

Chebykin S.V. *Technique of production planning of float glass manufacturing* 139

Devices. Materials of electronic technics

Lavrov A.P., Molodyakov S.A. *Possibilities of construction of signal processors in the form hybrid optoelectronic microcircuits* 144

Zaitsev E.F., Guzenko K.V., Moldavanova E. N. *Analysis of waves in a ferrite-dielectric waveguide by variation method* 152

Modelling: methods, algorithms, technologies

Makatov M.R. Verdiev M.M. *Information model of the description of objects and situations of the problem environment* 156

Ismailov SH.A., Magomedov SH.G. *Algorithms and structures of the transformation numeric given from positional system in residue number system* 159

Novikov K.V. *Fast image segmentation algorithm based on spectral characteristics of anisotropic areas* 169

Surov A.N., Igizjanova N.A., Potapov V.I. *About convergence of decision difference scheme in mathematical model of generation thermal energy* 174

Pavlov D.A. *Algorithm for building the extended universal Grцbner Base* 179

Mantserov D.I. *System of verification for parametric classes of tasks in mathematics* 183

Kanunnikova E.A., Koltunov L.I., Potapenko A.N. *Numerical modeling of distributed electrotechnical systems in semi-infinitive regions based on the inversion method* 189

Ushmotkin A.S., Korzhik V.I. *The estimation of an efficiency of digital watermarking system for images under conditions of a coalition attack* 198

Suslov N.V. *Principles of construction of virtual training space* 204

Лукьянова Л.М.

Методика структурно-целевого анализа и синтеза организационно-технических комплексов[1]

Развитие системных исследований привело к выделению системного анализа как методологии, с которой связывалось решение сложных проблем высокой степени неопределенности. Главным достижением системного анализа провозглашались научные методы, обеспечивающие учет субъективных факторов. Способствуя выработке и обоснованию решений, проведение системного анализа, однако, не гарантирует их непротиворечивость: решения нередко оказываются логически некорректными, что выражается в наличии различного рода логических ошибок в программах планах создания и развития систем. Выявляемые при исполнении решений, они снижают эффективность управления соответствующими объектами.

В последнее время при выработке решений в проблемных ситуациях наряду с традиционными методиками и программными средствами системного анализа [1] используются системы поддержки принятия решений, стратегического планирования, целевого управления, управления проектами. Тем самым расширяется состав автоматизированных функций и увеличивается глубина их реализации, что обеспечивает более эффективное выполнение системного анализа, но все же не исключает возможную некорректность его результатов.

Один из источников некорректности результатов системного анализа — недостаточная определенность логического компонента данной методологии. Другие — сложность объектов анализа и неопределенность проблемных ситуаций, требующих разрешения. В совокупности с субъективными факторами и логикой здравого смысла это может повлечь неправильное выполнение системного анализа и неправильное оценивание корректности его результатов. К логически некорректным результатам отдельных этапов системного анализа и несогласованности результатов, разных его этапов приводит неопределенность логических значений проблем, целей, критериев и систем целедостижения. В производственной сфере это в полной мере относится к организационно-техническим комплексам (далее комплексы) — сложным и фондоемким объектам, цена ошибок в результатах системного анализа которых часто оказывается недопустимо высокой.

В методике структурно-целевого анализа и синтеза комплексов реализованы методы [2], обеспечивающие логически правильное выполнение опорных процедур системного анализа этих объектов и логическую корректность их результатов.

Общая методика структурно-целевого анализа и синтеза комплесов

Методику составляют три группы обобщенных процедур, реализующие взаимоувязанные методы: 1) процедуры анализа проблем, целей и оценивания структур целей (СЦ); 2) процедуры анализа и синтеза систем целедостижения (СЦД); 3) процедуры синтеза структурных схем целедостижения (ЦД).

Процедуры анализа проблем, целей и оценивания СЦ комплекса

Схема анализа проблем, включающая процедуры П1–П5, расширена по сравнению с традиционной схемой процедурами, выделенными курсивом:

выявление и предварительный анализ проблем → формирование каталога проблем и анализ "причинно-следственных" связей проблем → формирование и анализ каузативной СП → определение главной проблемы → анализ проблем и формирование СП.

[1] Работа выполнена при финансовой поддержке РФФИ, проект № 06-07-89242.

П1. Выявление и предварительный анализ проблем. Арсенал методов выявления проблем достаточно широк (от методов выработки коллективных решений и морфологического анализа до кластер- и контент-анализа). Выбор метода зависит от степени неопределенности проблемной ситуации, форм взаимодействия с экспертным коллективом (ЭК) и некоторых других факторов. Предварительный анализ проблем выполняется с целью повышения надежности информации о проблемах, выявленных ЭК. При этом руководствуются общими принципами систематики. Метод выполнения процедуры является неформальным.

П2. Формирование каталога проблем и анализ "причинно-следственных" связей проблем. Сформировать каталог проблем, используя их список, полученный в процедуре П1, и руководствуясь методом логико-лингвистического моделирования и анализа проблем (ЛЛМАП), аналогичным по аксиоматике методу логико-лингвистического моделирования и анализа целей (ЛЛМАЦ) [2]. Определить проблемы-"следствия" каждой из проблем каталога, рассматривая ее как частичную "причину" проблем-"следствий". Используемые методы — неформальные.

П3. Формирование и анализ каузативной СП. Использовать "причинно-следственные" связи проблем, выявленные в процедуре П2, для формирования каузативной СП. Метод построения каузативной СП формален. Проанализировать каузативную СП, руководствуясь методом ЛЛМАП. Данный метод неформален.

Результатом процедуры, как показывает опыт проблемного анализа комплексов, в общем случае является "лес" каузативных СП. При получении сетевых СП целесообразно стратифицировать сложные проблемы по аспектам, например управленческому, экономическому, производственному. Уменьшая сложность проблем, этот прием позволяет выделить древовидные остовы в сетевых СП.

П4. Определение главной проблемы (n_0). Построение в процедуре П3 "причинно-следственной" СП может привести к единственной проблеме-"следствию", которую выбирают в качестве главной проблемы. В случае незавершенной иерархии или "леса" "причинно-следственных" СП необходимо проанализировать полученную структуру и сформировать список проблем-претендентов на роль главной, а для ее выбора осуществить дополнительный анализ проблемной ситуации. Метод выполнения процедуры является неформальным.

П5. Анализ проблем и формирование СП. При выполнении процедуры посредством метода ЛЛМАП, строится логически корректная и полная СП. Формирование иерархической СП начинается с анализа главной проблемы n_0 и заканчивается такими проблемами $n_{(m-1)j_{m-1}}^{(m-2),j_{m-2}}$, ($m$-1)-го уровня, сложность и неопределенность которых достаточно мала для того, чтобы не вызывать у ЛПР затруднений при выборе средств их решения.

Примечание. Исследование проблемных ситуаций, помимо выявления, анализа и синтеза проблем, осуществляемых на логической страте, должно включать анализ динамики проблем в контексте внешних для исследуемой системы и внутрисистемных условий посредством их количественного анализа на математической страте. Это позволяет выявить принципиальную разрешимость проблем, требует использования экономико-математических методов [1] и реализуется после выполнения процедур П1–П5.

Процедуры и методы анализа целей

Метод ЛЛМАЦ, регламентирующий лингвистическое описание целей (структуру предложения-цели и семантических компонентов его фраз), реализуется в процедурах анализа и полагания целей комплексов, а также в процедурах выбора состава и содержания словарной и тезаурусной баз прикладной системы анализа и синтеза целей [2]. Высокая степень формализации метода придает аналогичные свойства процедурам анализа и полагания целей, обеспечивает возможность настройки языка описания целей и стратегий их анализа на предметную область и перестройки семиотической системы рассуждений о целях в процессе ее функционирования.

Предлагаются три схемы формирования СЦ: СЦ-i, i = 1(1)3. Самая простая — СЦ-1 (процедуры анализа и полагания целей в СЦ-1 в явном виде не отражены): *формулирование целей и формирование СЦ по СП.*

В соответствии с данной схемой проводится "переформулирование" проблем n_0, …, $n_{i,j_i}^{(i-1),j_{i-1}}$, …, $n_{(m-1)j_{m-1}}^{(m-2),j_{m-2}}$, увязанных после выполнении процедур П1–П5 в СП, в соответствующие цели u_0,

$\ldots, u_{i,j_i}^{(i-1),j_{i-1}}, \ldots, u_{(m-1)j_{m-1}}^{(m-2),j_{m-2}}$ методом "отрицания проблем".

Преобразование формулировки проблемы в формулировку цели осуществляется следующим образом. Вначале формируется предварительный вариант предложения-цели. Для этого маркер H проблемных фраз предложения-проблемы заменяется маркером G целевых фраз. После того, как все цели сформулированы, имеем СЦ, изоморфную соответствующей СП.

В соответствии со схемой СЦ-2, имеющей вид: *формулирование главной цели → анализ и полагание целей и формирование СЦ*, — главная проблема n_0 переформулируется в главную цель u_0, содержание которой раскрывается в результате анализа и полагания целей с помощью метода ЛЛМАЦ. При этом формируется СЦ, которой присущи свойства логической корректности и полноты в том точном смысле, который определен для данных структур в [2].

В соответствии со схемой СЦ-3, имеющей вид: *выявление и предварительный анализ целей → формирование каталога целей и анализ "причинно-следственных" связей целей → формирование и анализ каузативной СЦ →определение главной цели → анализ и полагание целей и формирование СЦ*, — проводится полномасштабный анализ целей, при котором состав и содержание процедур аналогичны процедурам, определенным для схемы анализа проблем.

Независимо от используемого ЛПР варианта схемы СЦ-*i*, i = 1(1)3 конечным результатом регламентируемых им процедур анализа целей Ц8, Ц9,..., Ц12 являются СЦ, логически корректные с позиций логических законов и закономерностей полагания и анализа целей, полные с позиций тезаурусов предметной области и представляемые в виде семантических графов ${}^{C}G^{a}$ [2]. Указанным основным процедурам предшествуют дополнительные процедуры анализа целей Ц1, Ц2,..., Ц7.

Ц1. Выбор варианта схемы анализа целей СЦ-*i*, i = 1(1)3 в соответствии с проблемной ситуацией, сложившейся в комплексе и окружающей среде.

Ц2. Построение канона СЦ, представляющего собой обобщенную стратегическую структуру целей деятельности систем конкретного сегмента производственной сферы. Для этого провести неформальный анализ стратегических целей, декларируемых в руководящих материалах, директивных, нормативных и других документах, руководствуясь опытом управления системами в этом сегменте.

Ц3. Определение макро- и микроописателя предложения-цели. В зависимости от требуемой глубины функционального описания структуры деятельности в анализируемом сегменте производственной сферы использовать макроописатель предложения-цели по умолчанию либо его измененный (усеченный или расширенный) вариант. В зависимости от требуемой в анализируемом сегменте производственной сферы глубины качественного и количественного описания результатов и средств деятельности использовать микроописатель фразы предложения-цели, принятый по умолчанию либо его измененный вариант. Методы выполнения процедуры являются частично формальными.

Ц4. Создание словарной базы *BD* ролевых базовых предметов и их видовых базовых свойств по результатам извлечения знаний о результатах деятельности в сегменте производственной сферы и используемых для их осуществления средств, представленных в отраслевых тезаурусах, нормативно-справочных и руководящих документах. Метод выполнения процедуры частично формальный.

Ц5. Определение структуры тезаурусной базы *TZ*. В зависимости от требуемой глубины описания стратегий управления в анализируемом сегменте производственной сферы использовать принятый по умолчанию либо измененный набор тезаурусов, имена которых выражают иерархичность базовых свойств и предметов, составляющих организационно-технический комплекс. Метод выполнения процедуры является частично формальным.

Ц6. Заполнение тезаурусной базы *TZ* путем анализа пар элементов каждого словаря в базе *BD* и выявления на них, руководствуясь отраслевыми тезаурусами, нормативно-справочными и другими документами, регламентирующими деятельность комплекса, отношений включения или подчинения, сопоставленности по включению или подчинению, соответствующих именам конкретных тезаурусов. Метод выполнения процедуры является частично формальным.

Ц7. Определение стратегий анализа целей. В зависимости от структуры *TZ* использовать базовый либо измененный набор закономерностей анализа и полагания целей (выражающий закономерности анализа и полагания целей в конкретном комплексе). Метод выполнения процедуры частично формальный.

Описываемые далее процедуры Ц8–Ц12 полиморфны и зависят от выбранной схемы формирования структуры целей СЦ-i, где $i \in 1(1)3$. Поэтому выполняются те из них, которые обозначены номером i, соответствующим номеру СЦ- i.

Ц8-1. Формирование СЦ по СП, являющейся результатом процедуры П5.

Ц8-2. Формулирование главной цели u_0 в ходе неформального анализа главной проблемы n_0, определенной в процедуре П4.

Ц9-2. Формирование СЦ аналогично процедуре П5, исходя из цели u_0.

(Ц8-3)–(Ц12-3). Формирование СЦ аналогично процедурам П1–П5.

Методика оценивания СЦ

Анализ целей на логической страте завершается оцениванием степени логической корректности и полноты СЦ, сформированной любым способом, среди которых предпочтителен способ, описанный в процедурах Ц1–Ц12. В случае "ручного" формирования, исходная СЦ помечается одним штрихом, а результат анализа и синтеза целей — двумя штрихами.

Для СЦ′ рассчитываются коэффициенты общей некорректности ($k_п$), изолированности ($k_и$), равнозначности ($k_д$), обратного подчинения ($k_о$), нарушения непрерывности ($k_н$), полноты ($k_л$), а также число компонентов связности (d). Для СЦ″ вычисляются последние три показателя. Для СЦ, представленной в виде семантического графа целей ${}^cG^a$, использованы следующие обозначения: Ц — множество вершин (целей); U — множество дуг (семантических отношений подчинения целей ${}^cR^п$, определенных на Ц); p, c, d — число элементов множества Ц, отношений ${}^cR^п$, компонентов связности соответственно.

Ц13. Оценивание показателей логической корректности и полноты СЦ. В процедуре оценивания СЦ′ используются значения переменных Ц′, U′, ${}^cR^{п'}$, p′, c′ и d′, а СЦ″ оценивается с использованием Ц″, U″, ${}^cR^{п''}$, p″, c″ и d″.

Обозначим общее число логических ошибок анализа целей через N, а индекс типа ошибки — через i. Тогда: $N = \sum_{i=1}^{n} N_i$, где N_i — число ошибок i-го типа.

В процедуре оценивания показателей логической корректности и полноты СЦ используются следующие расчетные формулы: $k_п = N/(s' + c')$, где s' — минимально возможное число петель в СЦ′, равное числу целей во множестве Ц; $k_и = N_1/p'$; $k_д = N_2/c'$; $k_о = c_o'/c'$, где c_o' — число дуг графа ${}^cG^a$ с ориентацией, обратной по отношению к СЦ′;

$k_Н \in \{\min k_{неп}, \max k_{неп}\}$, где $k_{неп}$ — коэффициент непрерывности в е-й ветви СЦ′; $k_{не} = t_р / t_{неп}$, где $t_р$ — реальное число уровней в е-й ветви СЦ′; $t_{неп}$ — идеальное число уровней в е-й ветви СЦ′ в соответствии с базовыми знаниями первого и второго уровней системы поддержки анализа и полагания целей;

$k_Л \in \{\min k_{лк}, \max k_{лк}\}$, где $k_{лк}$ — коэффициент полноты подцелей в кусте целей; $K_{лк} = P_р/P_и$, где $P_р$ — реальное число подцелей, $P_и$ — идеальное число подцелей в соответствии с базовыми знаниями первого и второго уровней прикладной системы анализа и синтеза целей [2].

Процедуры анализа и синтеза СЦД

Анализ и синтез комплексов как систем СЦД включает: 1) анализ входов и выходов СЦД (построение модели СЦД типа "черный ящик"); 2) анализ функций СЦД (построение модели состава функций СЦД); 3) синтез компонентов СЦД (построение модели состава компонентов СЦД); 4) анализ и синтез структур СЦД (построение моделей структур СЦД); 5) синтез СЦД (построение модели СЦД типа "белый ящик").

СЦД1. Анализ входов-выходов СЦД. Осуществить с помощью экспертных методов анализ выходов СЦД, под которыми понимаются результаты функционирования комплекса, соответствующие его целям, и входов, под которыми понимаются элементы окружающей среды, способствующие достижению его целей.

Для выделении комплекса из среды и представления его в виде модели "черного ящика" $М^{чя}$ осуществляется анализ выходов (Y) и входов (X) СЦД: $М^{чя} = <A, Θ^{чя}>$, где A — множество выходов и входов, A = {Y, X}; $Θ^{чя}$ — свойство целостности множества A, соотносимое с главной целью комплекса u_0.

СЦД2. Анализ функций СЦД. При построении модели состава функций СЦД ($М^Ф$) определение функций активных средств (например, "техники"), в описаниях которых на языке описания целей L_1 эксплицированы функциональные свойства (СФ), является тривиальной задачей, решаемой формальным образом (например, СФ-свойству "копчение" соответствует функция "копчение").

Для целей, в описаниях которых отсутствуют СФ-свойства, но эксплицированы характеристические свойства (СХ1), например, "конечного объекта", функции соответствующих активных "средств" определены имплицитно, но "вычисляются" просто. Данный метод легко формализуем, а процедура, в которой он использован, может выполняться без участия ЛПР (например, СХ1-свойству "консервированная", определяющему продукцию комплекса, соответствует функция "консервирование", определяющая "технику"). Метод определения функций активных "средств" достижения целей, в описаниях которых отсутствуют СФ- и СХ1-свойства, неформален.

Моделирование состава функций проиллюстрируем на примере формирования СЦД, для которой определена трехуровневая СЦ (i = 0(1)2, где i — уровень структуры целей). Свяжем с первым, высшим, уровнем СЦД функцию $ф_1$, которую определим в соответствии с описанием главной цели (цель c_0), и сформируем модель состава функций первого уровня системы достижения целей $М_1^{фс}$: $М_1^{фс} = < Ф_1, Θ_1^{фс} >$, где $Ф_1$ — единичное множество функций 1-го уровня СЦД, $Ф_1 = \{Ф_1\}$; $Θ_1^{фс}$ — свойство целостности функций $Ф_1$ по отношению к цели c_0.

Затем, исходя из множества целей C_1 первого уровня СЦ, $C_1 = \{c_{1,ji}^0\}$, формируется модель состава функций второго уровня СЦД $М_2^{фс}$: $М_2^{фс} = < Ф_2, Θ_2^{фс} >$, где $Ф_2$ — множество функций 2-го уровня СЦД, $Ф_2 = \{ф_{2i}\}$ $Θ_2^{фс}$ — свойство целостности множества функций $Ф_2$ по отношению к множеству целей C_1.

Далее, исходя из целей C_2 второго уровня СЦ, $C_2 = \{c_{2,ji}^{1,a}\}$, формируется модель состава функций третьего уровня СЦД $М_3^{фс}$: $М_3^{фс} = < < Ф_3, Θ_3^{фс} >$, где $Ф_3$ — множество функций 3-го уровня СЦД; $Ф_3 = \{ф_{3i}\}$. $Θ_3^{фс}$ — свойство целостности множества функций $Ф_3$ по отношению к множеству целей C_2.

На основе моделей $М_1^{фс}$, $М_2^{фс}$, $М_3^{фс}$ синтезируется модель $М^{фс}$ состава функций СЦД: $М^{фс} = < Ф, Θ^{фс} >$, где $Ф$ — множество функций СЦД, $Ф = \{Ф_1, Ф_2, Ф_3\}$; $Θ^{фс}$ — свойство целостности $Ф$ по отношению к множеству целей C.

СЦД3. Синтез компонентов СЦД (построение модели состава компонентов). Для синтеза рационального состава компонентов СЦД анализируется множество функций $Ф_2 \cup Ф_3$ из моделей $М_1^2$ и $М_2^2$, и за каждой функцией $ф \in Ф_2 \cup Ф_3$ закрепляется исполнитель – компонент $к$ (из рассмотрения исключается функция $ф_1 \in Ф_1$ СЦД). Сгруппируем функции $ф_{2i}$, $ф_{2i} \in Ф_2$, $ф_{3j}$, $ф_{3j} \in Ф_2$, а тем самым и соответствующих исполнителей с помощью экспертных методов, руководствуясь теми или иными принципами, например субъектно-объектным, "жизненным циклом" (ЖЦ) продукции или управленческой информации [1,2].

Анализ функций производства продукции и управления этим производством позволяет выделить функции управляющей (субъект управления) и управляемой (объект управления) подсистем СЦД, сгруппировать функции внутри этих подсистем по общепринятым нормам и правилам. Для группирования функций управляемой подсистемы СЦД используются стадии ЖЦ ее продукции, для группирования функций управляющей подсистемы — фазы ЖЦ управления.

Сформируем вначале модель состава компонентов СЦД 1-го уровня $М_1^{кс}$: $М_1^{кс} = < К_1, Θ_1^{кс}$, где $К_1$ — единичное множество компонентов 1-го уровня, $К_1 = \{к_1\}$ $к_1$ — СЦД в целом; $Θ_2^{кс}$ — свойство целостности множества $К_1$ по отношению к множеству функций $Ф_1$ затем модель состава компонентов СЦД 2-го уровня $М_2^{кс}$: $М_2^{кс} = < К_2, Θ_2^{кс} >$, где $К_2$ — множество компонентов (подсистем и/или элементов) 2-го уровня СЦД, $К_2 = \{к_{2i}\}$ $i = 1(1)n$; $Θ_2^{кс}$ — свойство целостности $К_2$ по отношению к множеству функций $Ф_2$, и при необходимости модель состава компонентов СЦД 3-го уровня $М_3^{кс}$: $М_3^{кс} = < К_3, Θ_3^{кс} >$, где $К_3$ — множество компонентов 3-го уровня СЦД, $К_3 = \{к_{3j}\}$, $j = 1(1)m$; $Θ_3^{кс}$ — свойство целостности $К_3$ по отношению к множеству функций $Ф_3$.

Основываясь на моделях $М_1^{кс}$, $М_2^{кс}$, $М_3^{кс}$, синтезируем модель $М^{кс}$ состава компонентов СЦД: $М^{кс} = < К, Θ^{кс} >$, где $К$ — множество компонентов СЦД, $К = \{К_1, К_2, К_3\}$; $Θ^{кс}$ — свойство целостности $К$ по отношению к множеству $Ф$.

СЦД4. Анализ и синтез структуры СЦД. Для синтеза рациональной структуры СЦД вначале анализируется модель $М^{кс}$, построенная в процедуре СЦД3. Если модель $М_2^{кс}$ покрывает функции $ф \in Ф_2 \cup Ф_3$, синтезируется *двухэшелонная* структура системы достижения целей $М_2^{стр}$: $М_2^{стр} = (K; R)$, где K — множество компонентов СЦД, $К = К_1 \cup К_2$, $К_1$ — СЦД, $К_2$ — множество элементов СЦД, $К_2 = \{к_{2i}\}$; R — отношение включения компонентов СЦД в другие компоненты данной системы, в противном случае синтези-

руется *трехэшелонная* структура $М_3^{стр}$: $М_3^{стр}$ = = (K; R), где K — множество компонентов СЦД, К = $К_1 \cup К_2 \cup К_3$, $К_1$ — СЦД, $К_2$ — множество подсистем СЦД, $К_2 = \{к_{2i}\}$, $К_3$ — множество элементов СЦД, $К_3 = \{к_{3j_{2i}}\}$.

Таким образом, модель $М_2^{стр}$ представляет собой двухуровневое дерево с корнем $к_1$ (СЦД) и листьями $К_{2i}$, образуемыми компонентами 2-го уровня (элементы СЦД). Модель $М_3^{стр}$ представляет собой трехуровневое дерево с корнем $к_1$ (СЦД), с которым связано n кустов, которые образованы компонентами 2-го и 3-го уровней данной структуры. "Корнями" указанных кустов являются компоненты $К_{2i}$ (подсистемы СЦД), а листьями – компоненты $к_{3j_{2i}}$ (элементы СЦД).

СЦД5. Синтез СЦД. Осуществляется с помощью экспертных методов. При этом, основываясь на моделях $М^{чя}$, $М^{фс}$, $М^{кс}$ и $М^{стр}$ системы достижения целей, синтезируют модель СЦД типа "белый ящик" $М^{бя}$.

Процедуры синтеза структурных схем ЦД

Формирование структурной схемы ЦД обычно проводится в два этапа. На первом формируется начальная, а на втором конечная структурная схема ЦД. Цели в начальной структурной схеме ЦД, обозначаемой как $CTPSN_{СЦС^{внут}}$, моделью которой является семантический граф ${}^cGN^c$, обозначим через $цц_{i,j_i}^{(i-1),j_{i-1}}$ в отличие от соответствующих целей в $СТРЦ_{СЦА^{внут}}$ как результате анализа целей (моделью которой является семантический граф ${}^cG^a$), обозначаемых через $ц_{i,j_i}^{(i-1),j_{i-1}}$ (таким образом, цели в $CTPSN_{СЦС^{внут}}$ и $СТРЦ_{СЦА^{внут}}$ имеют одинаковые индексы, а цели нулевого уровня обозначаются через $цц_0$ и $ц_0$ соответственно). В начальной структурной схеме ЦД имеется дополнительный по сравнению с СЦ, m-й, уровень с единственной целью $цц_m$. Ее формулировка — "начать целедостижение", а роль аналогична роли начальной вершины в сетевом плане-графике. Наконец, в отличие от $СТРЦ_{СЦА^{внут}}$, в которой j_i-е цели i-го уровня, подчинены j_{i-1}-м целям (i – 1)-го уровня, в $CTPSN_{СЦС^{внут}}$, напротив, j_{i-1}-е цели (i – 1)-го уровня подчинены j_i-м целям i-го уровня.

Цели в конечной структурной схеме ЦД, обозначаемой через $CTPSK_{СЦС^{внут}}$, моделью которой является семантический граф ${}^cGK^c$, будем обозначать в отличие от целей в $CTPSN_{СЦС^{внут}}$ через $цд$ с теми же правыми нижними, но другими правыми верхними индексами. При формировании конечной структурной схемы ЦД j_i-е цели i-го уровня синтезируются из совокупностей следующих целей (i + 1)-го уровня: (j_{i+1}= d)-й, (d + 1)-й,…,($d + n_d$)-й. Именно это и отражено в верхних индексах синтезируемых целей, обозначаемых через $цд_{i,j_i}^{((i+1),d),((i+1),(d+1)),\ldots,((i+1),(d+n_d))}$.

ССЦД1. Осуществить преобразование структуры целей $СТРЦ_{СЦА^{внут}}$ в (m + 1)-уровневую структурную схему ЦД путем изменения исходных обозначений целей ($ц_0$ на $цц_0$ и $ц_{i,j_i}^{(i-1),j_{i-1}}$ на $цц_{i,j_i}^{(i-1),j_{i-1}}$), введения дополнительного m-го уровня с единственной целью $цц_m$ и формирования связей с ней всех целей низшего уровня $СТРЦ_{СЦА^{внут}}$ (дуги между $цц_m$ и листовыми вершинами $ц_{(m-1),j_{m-1}}^{(m-2),j_{m-2}}$ графа ${}^cG^a$) или всеми листовыми целями структуры $СТРЦ_{СЦА^{внут}}$.

ССЦД2. Заменить отношения подчинения между целями полученной в процедуре ССЦД1 структурной схемы ЦД на обратные отношения между (мыслимыми) средствами и результатами деятельности комплекса.

ССЦД3. Уточнить с помощью экспертных методов формулировки неопределенных целей структурной схемы $CTPSN_{СЦС^{внут}}$, сформированной в ССЦД2.

Логическая корректность получаемой таким образом начальной структурной схемы ЦД обеспечивается логической корректностью соответствующей структуры целей. Если все цели в $CTPSN_{СЦС^{внут}}$ являются определенными, она используется в качестве логического основания формирования программы (плана) деятельности. Если же не все цели $CTPSN_{СЦС^{внут}}$ отвечают требованию определенности, возникает необходимость во втором этапе формирования структурной схемы ЦД — в этапе синтеза конечной структурной схемы ЦД

($CTPSK_{\text{СЦС}^{\text{внут}}}$). Конечная структурная схема ЦД формируется в ходе преобразования $CTPSN_{\text{СЦС}^{\text{внут}}}$, выполняемого процедурами ССЦД4 и ССЦД5. Процедура ССЦД4 является неформальной и полиморфной, включающей две разновидности: ССЦД4-1 и ССЦД4-2.

ССЦД4-1. Синтез целей

$$цд_{i,j_i}^{((i+1),d),((i+1),(d+1)),\ldots,((i+1),(d+n_d))} \in ЦК$$

(всех или некоторых), где ЦК — множество целей конечной структурной схемы ЦД $CTPSK_{\text{СЦС}^{\text{внут}}}$, начиная с ее низшего уровня, за исключением $цд_m$, и формирование конечной структурной схемы ЦД. Процедура осуществляется за счет конкретизации понятий, из которых составлены формулировки целей $цц_{i,j_i}^{(i-1),j_{i-1}}$ $\in$ ЦН, где ЦН — множество целей, включенных в начальную структурную схему ЦД. Указанная конкретизация становится возможной на основе информации, полученной ЛПР во время выполнения этапов структурно-целевого анализа и синтеза комплексов, связывающих этапы формирования СЦ и структурных схем ЦД. В результате конкретизации число вершин

$$цд_{i,j_i}^{((i+1),d),((i+1),(d+1)),\ldots,((i+1),(d+n_d))} \in ЦК$$

семантического графа ${}^cGK^c$ и связей между ними не изменяется по отношению к графу ${}^cGN^c$, но претерпевает изменение содержание вершин (целей) и семантика связей между ними.

ССЦД4-2. Синтез целей

$$цд_{i,j_i}^{((i+1),d),((i+1),(d+1)),\ldots,((i+1),(d+n_d))} \in ЦК$$

и формирование конечной структурной схемы ЦД (путем замены всех, за исключением $цд_m$, или некоторых целей соответствующими структурными подсхемами ЦД, определяющими, например, этапы технологического процесса достижения указанных целей в подразделениях комплекса. К такому же виду конкретизации целей $цд_{i,j_i}^{((i+1),d),((i+1),(d+1)),\ldots,((i+1),(d+n_d))}$ может привести оценивание и выбор вариантов достижения целей на математической страте их анализа. В результате указанной конкретизации изменяется число и семантика целей и связей между ними.

ССЦД5. Преобразовать структуру $CTPSK_{\text{СЦС}^{\text{внут}}}$ в структуру целей $СТРЦ_{\text{СЦА}^{\text{внут}}}^{обр1}$, выполнив для этого процедуры $\overline{\text{ССЦД2}}$, $\overline{\text{ССЦД1}}$ обратного (по отношению к процедурам ССЦД2, ССЦД1) преобразования. Выполнить анализ целей структуры $СТРЦ_{\text{СЦА}^{\text{внут}}}^{обр1}$ с помощью процедур Ц8–Ц12 и оценить степень ее логической корректности, а также корректности структуры $СТРЦ_{\text{СЦА}^{\text{внут}}}^{обр2}$, полученной в результате ее изменения после выполнения указанных процедур. Логическая корректность структуры $СТРЦ_{\text{СЦА}^{\text{внут}}}^{обр1}$ подтверждает логическую корректность соответствующей ей конечной структурной схемы ЦД $CTPSK_{\text{СЦС}^{\text{внут}}}$. В случае обнаружения в $СТРЦ_{\text{СЦА}^{\text{внут}}}^{обр1}$ логических ошибок и формирования в результате их исправления структуры целей $СТРЦ_{\text{СЦА}^{\text{внут}}}^{обр2}$ для последней необходимо вновь выполнить процедуры ССЦД1–ССЦД5.

Заключение

Методика структурно-целевого анализа и синтеза комплексов используется при разработке частных методик, обеспечивающих логически правильное выполнение опорных процедур системного анализа конкретных объектов данного класса и логическую корректность результатов этих процедур. Но и с их использованием в силу психических свойств человека в неопределенных ситуациях не всегда удается сформировать логически корректные результаты опорных процедур системного анализа комплексов. Для уменьшения влияния человеческого фактора в дополнение к методике разработано программное средство [3], обеспечивающее автоматизацию ее обобщенных процедур [4].

СПИСОК ЛИТЕРАТУРЫ

1. Системный анализ и принятие решений: Словарь-справочник: Учеб. пособие для вузов. / Под ред. В. Н. Волковой, В. Н. Козлова. М.: Высшая школа, 2004. 616 с.

2. **Лукьянова Л. М.** Теоретико-методологические основы структурно-целевого анализа и синтеза организационно-технических комплексов. СПб.: Наука, 2006. 276 с.

3. **Лукьянова Л. М.** Программное средство "Анализ и синтез систем целедостижения". Свидетельство о государственной регистрации программы для ЭВМ № 2008614503 от 19.09.2008. М.: ФГУ ФИПС, 2008.

4. **Лукьянова Л. М.** Автоматизация опорных логических процедур системного анализа организационно-технических комплексов // Науч.-техн. ведомости СПбГПУ. 2008. №3. С. 86–91.

Савельев М.В., Аликов А.Ю., Федосеев С.В.

Вероятность связности сети с аддитивной структурой

Современные сети включают тысячи элементов – узлов и соединений. Подобные сети избыточны, связь между узлами может осуществляться по нескольким различным путям. Отказ узла или соединения не позволяет их использовать. Множество возможных трасс связи между узлами, а следовательно, и надежность сети зависят от конфигурации (структуры) сети. Структурная надежность сети определяется вероятностями существования хотя бы одного пути из исправных каналов между узлами из заданного подмножества (вероятность связности подмножества узлов).

Так как один и тот же элемент сети может входить в состав нескольких путей, то существование различных путей есть взаимозависимые события. Взаимозависимость делает крайне сложным решение задачи определение вероятности связности заданной пары узлов. Трудности многократно возрастают при нахождении вероятности связности всего множества узлов сети, когда необходимо отслеживать одновременное наличие путей между возможными парами узлов сети.

В общем случае решение возможно только численными методами. Точное вычисление характеристик связности имеет экспоненциальную зависимость трудоемкости от размера, и для сетей, включающих несколько десятков узлов, требуемый объем вычислений превосходит ресурсы современных ЭВМ. Поэтому определяются оценки, базирующиеся на имитационном моделировании либо на эвристических методах по формированию путей и сечений сети [1].

Для сокращения трудоемкости накладываются ограничения на точность результатов, на параметры надежности элементов, на структуру сети. В современной теории надежности наименее исследованной является направление, где эффективность структурных ограничений может быть очень высокой. Машинная реализация структурной аппроксимации имеет экспоненциальную зависимость трудоемкости от размера сети. Это расчет характеристик и зависимостей структурной надежности тестовой сети, адаптация результатов с учетом особенностей структурной аппроксимации.

Основное препятствие – узкий исходный набор классов тестовых сетей. При нетривиальной структуре сети и произвольных надежностях элементов вероятностный анализ крайне сложен. Для рекуррентных структур сеть образуется последовательным присоединением подсетей (рекуррентности) с одинаковой структурой и произвольными надежностями элементов. Более общим является класс аддитивных сетей (с аддитивной структурой), когда структура присоединяемых подсетей не является постоянной. Количество комплектующих подсетей определяет ранг сети.

Изучение рекуррентных сетей (радиально-кольцевой, лестничного типа и др.) [2–4] привело к разработке конструктивных точных итеративных методов расчета характеристик связностей с линейной зависимостью трудоемкости от размера (ранга) сети. Методы основаны на специальном вероятностном анализе каждой из присоединяемых подсетей. Однако трудоемкость быстро увеличивается с ростом числа m узлов подсети, и прикладное использование результатов крайне затруднено.

Неформально аддитивная структура означает, что каждый последующий кусок сети присоединяется к введенному на предыдущем

этапе. Аддитивная сеть n-го ранга (n-размер сети) $G_{1,n}$ определяется посредством итеративного процесса. Задается множество $\{g_k\}$ подсетей k-го дополнения, $k = \overline{2,n}$, с безотказными узлами и произвольными надежностями каналов связи. Отказы элементов взаимно независимы. Топология подсетей дополнения удовлетворяет следующим ограничениям.

1. В каждой подсети g_k выделены подмножества: $\{V_{k-1}\}$ — входных узлов, $\{U_1\}$ — выходных узлов, причем $|\{U_k\}| = |\{V_k\}|$.

2. Любые два входных узла не являются смежными, то есть между ними связь отсутствует.

3. Между выходными узлами $\{U_k\}$ подсети g_k и входными узлами $\{V_k\}$ подсети g_{k+1} установлено взаимно однозначное соответствие.

Начальная подсеть g_1 включает только подмножество выходных узлов $\{U_1\}$ и соединяющие их связи и в дальнейшем рассматривается как сеть первого ранга $G_{1,1}$. Для построения сети k-го ранга $G_{1,k}$ входные узлы $\{V_{k-1}\}$ подсети g_k совмещаются с выходными узлами $\{U_{k-1}\}$ сети $(k-1)$-го ранга $G_{1,k-1}$. Совмещение производится согласно соответствию между подмножествами $\{V_{k-1}\}$ и $\{U_{k-1}\}$. Выходные узлы $\{U_k\}$, присоединенные к подсети, становятся выходными узлами сети $G_{1,k}$ и идентифицируются как узлы k-го ранга. Надежности элементов сети $G_{1,k}$ такие же, как у соответствующих элементов сети $G_{1,k-1}$ или подсети g_k.

Сеть $G_{1,n}$ рекуррентна, если структура всех подсетей дополнения, кроме начальной, постоянна. Подсеть (k, m) сети $G_{1,n}$ n-го ранга, $1 \le k < m \le n$ аддитивной сети $G_{k,m}$ включает все элементы подсетей $g_{k+1}, g_{k+2}, \ldots, g_m$ и связи, соединяющие выходные узлы $\{U_k\}$ подсети g_k. В соответствии с построением входными узлами подсети $G_{k,m}$ будут узлы $\{V_k\}$ (или $\{U_k\}$), выходными – узлы $\{U_m\}$. Фактически из сети $G_{1,n}$ вырезается участок, начинающийся узлами k-го и заканчивающийся узлами m-го ранга.

Сеть или подсеть деформирована по входу (выходу), если она отличается от исходной номенклатурой или параметрами надежности соединений, связывающих входные (выходные) узлы. Деформированная подсеть $G_{k,m}$ будет замкнутой по входу (выходу) и обозначается $G_{k,m}(s_k = 0)$ ($G_{k,m}(s_m = 0)$), если любая пара входных (выходных) узлов соединяется безотказным каналом связи. Иначе исходная сеть $G_{k,m}$ разомкнута. Вероятность связности двух узлов сети есть вероятность события, то есть между этими узлами существует хотя бы один путь из исправных соединений связи. Вероятность связности заданного подмножества узлов сети будет вероятность события, при котором любая пара узлов из подмножества связна. Вероятность безотказной работы $P_{1,n}$ сети $G_{1,n}$ — это вероятность связности всех узлов сети.

Необходимо определить $P_{1,n}$ — вероятности безотказной работы (связности) аддитивной сети произвольного ранга $G_{1,n}$.

Для рекуррентных одномерных сетей разработан метод [2–4] для точного вычисления характеристик связности с линейной зависимостью трудоемкости от ранга n. Идея метода — замена анализа всей сети на последовательный анализ каждой из подсетей рекуррентности. Поскольку рекуррентные сети есть частные случаи сетей с аддитивной структурой, то можно сравнительно просто применить метод к аддитивным сетям.

Метод иллюстрируется на примере нахождения вероятности связности P_n рекуррентной сети $G_{1,n}$ n–го ранга второй степени со структурой лестничного типа. Построение сети достаточно очевидно: две расположенные одна под другой “стоки” узлов, нумеруемых от 1 до n и от $\tilde{1}$ до $\tilde{n}$ соответственно, соседние узлы соединяются каналами. Обозначим: $s_k = 1 - r_k$ — вероятность отказа “вертикального” $(k, \tilde{k})$ между выходными узлами k и $\tilde{k}$ подсети рекуррентности g_k; $q_k = 1 - p_k$, $\tilde{q}_k = 1 - \tilde{p}_k$ — вероятность отказа соответственно “горизонтальных” связей $(k-1,k)$, $(\tilde{k}-1,\tilde{k})$, соединяющих входные и выходные узлы подсети g_k, $k = \overline{2,n}$.

Этап 1. Множество состояний сети $G_{1,n}$ разбивается на подмножества в соответствии с разложением искомой характеристики по формуле полной вероятности по параметрам надежности каналов подсети рекуррентности g_n. Подмножества отличаются друг от друга фиксированной связью узлов n-го и $(n-1)$-го ранга между собой.

Подмножества, в которых хотя бы один из узлов n-го ранга не связан с хотя бы одним узлом $(n-1)$-го ранга, априорно не удовлетворяют требованиям связности сети, и для них вероятность безотказной работы сети равна нулю. В дальнейшем такие подмножества состояний сети $G_{1,n}$ можно не рассматривать.

В силу транзитивности понятия пути каждое из остальных подмножеств эквивалентно множеству состояний некоторой сети $(n-1)$-го ранга $G_{1,n-1}(*)$, в которой между определенными выходными узлами $(n-1)$-го ранга обеспечивается дополнительная безотказная связь по пути из безотказных, согласно разложению элементов подсети g_n. При этом связность сети $G_{1,n-1}(*)$ автоматически влечет связность исходной сети $G_{1,n}$. Обозначив через Π_{n-1} вероятность связности сети $(n-1)$-го ранга, в которой между выходными узлами существует безотказный путь, получим

$$P_n = (s_n p_n \tilde{p}_n + r_n q_n \tilde{p}_n + r_n p_n \tilde{q}_n) P_{n-1} + r_n p_n \tilde{p}_n \Pi_{n-1}. \quad (1)$$

Этап 2. Рассматривается каждая из построенных выше сетей $(n-1)$-го ранга, между отдельными выходными узлами которых существует дополнительный безотказный путь, включающий узлы $\{U_n\}$. Такой путь заменяется безотказным соединением, связывающим соответствующий выходные узлы сетей $(n-1)$-го ранга. Введенное соединение рассматривается в дальнейшем как элемент сети $(n-1)$-го ранга, и если он оказывается параллелен ранее существовавшему соединению, то оба соединения заменяются на безотказный.

Полученные сети $(n-1)$-го ранга являются деформированными и обозначаются $G_{1,n-1}(*)$, где в скобках указаны отличающиеся от исходных параметров. Искомая характеристика записывается как соответствующая линейная комбинация характеристик деформированных сетей $G_{n-1}(*)$:

$$P_n = (s_n p_n \tilde{p}_n + r_n q_n \tilde{p}_n + r_n p_n \tilde{q}_n) P_{n-1} + r_n p_n \tilde{p}_n P_{n-1}(s_{n-1} = 0). \quad (2)$$

Этап 3. Существенно, что деформация затрагивает только элементы подсети g_n. Поэтому для рекуррентной структуры каждой из деформированных подсетей $(n-1)$-го дополнения можно поставить в соответствие таким же образом деформированную подсеть, но уже n-го дополнения $g_n(*)$. После этого строится деформированная сеть $G_{1,n}(*)$, то есть полученная присоединением деформированной подсети $g_n(*)$ к сети $G_{1,n-1}$. Над сетями $G_{1,n}(*)$ выполняются процедуры, описанные на этапах 1, 2, причем разложение осуществляется по параметрам надежности подсетей $g_n(*)$. Процесс заканчивается, когда новые типы деформированных сетей перестают появляться:

$$P_n(s_n = 0) = (q_n \tilde{p}_n + p_n \tilde{q}_n) P_{n-1} + p_n \tilde{p}_n P_{n-1}(s_{n-1} = 0). \quad (3)$$

Этап 4. В итоге устанавливается совокупность линейных соотношений (2) и (3) между характеристиками связности одинаковым образом организованных множеств деформированных сетей соседних рангов, n-го и $(n-1)$-го рангов.

Математически полученные соотношения записываются в матричной форме. Вводится вектор-столбец P_n, координатами которого являются характеристики связности деформированных сетей n-го ранга, $P_n = |\ldots, P_n(*), \ldots|^T$, где T — индекс транспонирования. Коэффициенты установленных на этапах 2 и 3 линейных соотношений в соответствии с упорядочностью координат вектора P_n записываются в виде матрицы перехода A_n:

$$P_n = A_n + P_{n-1}, \quad (4)$$

где $P_n = |P_n, P_n(s_n = 0)|^T$,

$$A_n = \begin{vmatrix} s_n p_n \tilde{p}_n + r_n q_n \tilde{p}_n + r_n p_n \tilde{q}_n & r_n p_n \tilde{p}_n \\ q_n \tilde{p}_n + p_n \tilde{q}_n & p_n \tilde{p}_n \end{vmatrix}.$$

Этап 5. Соотношение (4) справедливо для любого n. В силу рекуррентности исходной сети элементы матриц A_k, $k = 2, \ldots, n-1$, определяются на основе матриц A_n простой заменой индексов при параметрах надежности каналов связи. Тогда

$$P_n = AnA_{n-1}P_{n-2} = \ldots = AnA_{n-1}\ldots A_2 P_1. \quad (5)$$

Этап 6. Координаты начального вектора P_1 есть характеристики деформированных начальных сетей 1-го ранга, то есть подсети g_1. Их вычисления проще, чем для исходной сети $G_{1,n}$, и возможно непосредственно. В большом числе случаев удается еще более упростить задачу, вводя в структуру соответствующие фиктивные элементы нулевого ранга и рассматривая в качестве начальной сеть G_0. Достаточно ввести в подсеть g_1 связи, вероятности отказа которых будут: $s_0 = 1, q_1, \tilde{q}_1$ — произвольные. Итак,

$$P_1 = |\, r_1,\ 1\,|^T, \quad P_0 = \begin{vmatrix} 0, & \dfrac{1}{p_1 \tilde{p}_1} \end{vmatrix}^T.$$

Обобщение метода на общий случай аддитивных сетей достаточно очевидно.

Этапы 1а, 2а. Дословно совпадают с соответствующими этапами анализа рекуррентных

сетей. В результате анализ исходной сети сводится к исследованию некоторого множества деформированных сетей $(n-1)$-го ранга.

Этап 3а. Для каждой из полученных деформированных сетей $G_{1,n-1}(*)$ проводится анализ в соответствии с этапами 1а, 2а, что приводит к изучению совокупности множеств деформированных сетей уже $(n-2)$-го ранга. Пересечение этих множеств непусто, на практике они почти совпадают, поэтому мощность объединения полученных множеств деформированных сетей $(n-2)$-го ранга не столь велика, как можно было ожидать.

Максимальное количество различных деформаций равно числу $C(m)$ различных структур, которые можно построить на множестве из m входных узлов подсети дополнения. Число m есть степень подсети дополнения. Для рекуррентной структуры m постоянно и будет степенью сети. При увеличении m мощность $C(m)$ стремительно растет и оценивается сверху как число различных частичных графов полного графа с m окрашенными вершинами.

Можно показать, что $C(m)$ определяется из мнемонической формулы, в которой C^k заменяется на $C(k)$, и $C^1 = C(1) = 1$:

$$C(m) = (C+1)m,\ C^k = C(k),\ C^1 = C(1) = 1. \quad (6)$$

При m =2, 3, 4, 5, 6 мощность $C(m)$ множества всех различных деформированных сетей будет равна соответственно 2, 5, 15, 52, 203. Значения являются оценкой сверху, так как для конкретных структур сетей некоторые из типов деформаций не возникают либо не участвуют в дальнейшем анализе.

Этап 4а. Аналогично этапу 4 установленные соотношения между характеристиками деформированных сетей соседних рангов записываются в матричной форме. В отличие от рекуррентных сетей матрицы будут не квадратными, а прямоугольными, а размерность векторов характеристик – переменна.

Этап 5а. В силу аддитивности сети операции согласно, этапам 3а, 4а, проводятся итеративно для сетей $(n-2)$-го, $(n-3)$-го,..., 2-го рангов, пока не приходим к исследованию начальной сети первого ранга.

Трудоемкость метода определяется следующим образом. Пусть $\bar{k}$ — среднее число соединений в подсетях дополнения, $\bar{C}(m)$ — средняя мощность множеств деформированных сетей. Тогда трудоемкость точного вычисления характеристик аддитивной сети n-го ранга на основе предлагаемого метода пропорциональна произведению $n^*\bar{C}(m)\times 2^{\bar{k}}$. Последний сомножитель отражает трудоемкость анализа подсети дополнения.

Если не учитывать структурных особенностей сети, то точное вычисление характеристик потребовало бы порядка 2^{nk} операций. Использование предлагаемого метода для точного вычисления характеристик связности сетей высокого ранга позволяет уменьшить трудоемкость в $\frac{2^{(n-1)\bar{k}}}{n\bar{C}(m)}$ раз. Так как деформированные сети являются подграфами одного и того же графа, то на практике трудоемкость счета еще более сокращается.

Вычисления осуществляются известными [5] методами. Для матрицы A определяются собственные значения, $\lambda_1, \lambda_2, \ldots, \lambda_r$, корни характеристического уравнения $A(\lambda) = |\lambda E - A| = 0$. Здесь $|A|$ — определитель матрицы A, E – единичная матрица. Из построения следует, что матрица перехода $A = |a_{i,j}|$ неотрицательная, $a_{i,j} \geq 0$ является минором стохастической матрицы. Для неотрицательных матриц, $A \geq 0$, справедливо [5] следующее утверждение: существует число $\lambda \geq 0$, которое является простым корнем неотрицательной матрицы, большее всех ее других корней. Это число является максимальным корнем матрицы и лежит в интервале, ограниченном минимальной и максимальной суммой элементов строк матрицы A.

Любая координата вектора характеристик P_n представляется в виде линейной комбинации n-х степеней корней (собственных значений) матрицы A. Коэффициенты разложения определяются из начальных условий либо по формуле Перрона [5].

Сети второй степени со структурой лестничного типа содержат условие изотропности: $s_k = s = 1 - r$, $q_k = q = 1 - p$, $\tilde{q}_k = \tilde{q} = 1 - \tilde{p}$. Характеристические числа матрицы A есть корни, λ_1, λ_2 квадратного уравнения:

$$\lambda^2 - (sp\tilde{p} + p\tilde{p} + rp\tilde{q} + rq\tilde{p})\lambda + sp^2\tilde{p}^2 = 0. \quad (7)$$

Вероятность связности сети представляется в конечном виде:

$$P_n = \frac{r(\lambda_1^n - \lambda_2^n)}{\lambda_1 - \lambda_2},$$

$$\Pi_n = P_n(s_n = 0) = \frac{[(p\tilde{p} - \lambda_2)\lambda_1^n - (p\tilde{p} - \lambda_1)\lambda_2^n]}{p\tilde{p}(\lambda_1 - \lambda_2)}.$$

Таким образом, при увеличении размера (ранга n) сети $G_{1,n}$ вероятность связности $P_{1,n}$ стремится к экспоненциальной зависимости от ранга с показателем, равным максимальному корню. Отсюда вытекает, что приближенные методы для оценки вероятности связности многомерных аддитивных сетей целесообразно искать в мультипликативной (произведение, отношение) форме.

В случае надежных каналов, $q << 1$, $s << 1$, максимальный корень близок к единице, тогда как остальные собственные числа близки к нулю. Для нашего примера с точностью порядка $(s+q+\tilde{q})^4$

$$\lambda_1 \cong 1 - q\tilde{q} - 2sq\tilde{q} - sq^2 - s\tilde{q}^2, \qquad (8)$$

$$\lambda_2 \cong s(1 - 2q - 2\tilde{q} + q^2 + \tilde{q}^2 + 5q\tilde{q}).$$

При надежных соединениях значимость максимального собственного значения в соотношениях для вычисления надежности изотропной сети быстро возрастает с увеличением ранга, тогда как остальными корнями можно пренебречь уже при ранге сети $n > (5–6)$. Слагаемые в приближенном выражении (8) для максимального корня по сути являются вероятностями простейших сечений, нарушающих связность сети. При этом изолируются отдельные узлы или группы узлов.

Локальные окрестности узлов изотропной сети одинаковы. Поэтому можно определить "удельную прочность" сети, отображаемую максимальным корнем. Это обеспечивает перспективность построения оценочных методов на основе последовательного анализа взаимно пересекающихся фрагментов аддитивной структуры окрестностей соседних узлов или соседних групп узлов.

Множественность возможных путей между узлами в избыточной сети порождает шунтирующее свойство соединений относительно локального возмущения — изменения надежности некоторого соединения. Для однородной сети можно определить вероятность связности в первом приближении. Таким образом, оценку целесообразно искать в форме, позволяющей компенсировать влияние возмущений. Предпочтительнее выбирать отношение вероятностей связности одинаковых подсетей исходной сети, когда одна является расширением другой. Но в этом случае необходима высокая точность вычисления компонент оценки, поскольку характеристики делимого и делителя будут очень близкими. Небольшая неточность может привести к большой ошибке.

Рекуррентная изотропная сеть n-го ранга строится из подсетей дополнения, имеющих одинаковую структуру и параметры отказа идентичных по назначению каналов связи при $n > k > 1$. В этом случае существенно лишь значение ранга, поэтому используются более компактные обозначения: $G_{1,n} = G_n, g_k = g, P_{i,i+k} = P_{k+1}$, $\Pi_{i,i+k} = \Pi_{k+1}$. Примером рекуррентной сети n-го ранга m-й степени является сеть со структурой в виде правильной решетки с квадратными ячейками размером $m \times n$. Для такой сети подсеть g_1 представляет собой последовательное соединение m узлов.

Для рекуррентной изотропной сети неравенства записываются в виде:

$$P_k\left(\frac{\Pi_k}{\Pi_{k-1}}\right)^{n-k} \le P_n \le P_k\left(\frac{P_k}{P_{k-1}}\right)^{n-k}. \qquad (9)$$

Относительная погрешность δ есть отношение максимальной оценки к минимальной минус единица. Для надежных сетей, $1 - P_n << 1$, относительная погрешность близка к абсолютной

$$\Delta \cong \delta = \left(\frac{\Pi_{k-1}P_k}{P_{k-1}\Pi_k}\right)^{n-k} - 1 = (n-k)\frac{\Pi_{k-1}P_k - P_{k-1}\Pi_k}{P_{k-1}\Pi_k}. \qquad (10)$$

Упорядочим корни (собственные числа) матрицы перехода изотропной рекуррентной сети так, чтобы их величина убывала с ростом номера (λ_1 максимальное, λ_2 второе по величине и так далее). Так как P_n, Π_n есть различные координаты одного и того же вектора характеристик P_n, то они представляются в виде:

$$P_n = a_1\lambda_1^n - a_2\lambda_2^n + O(\lambda_3^n), \quad \Pi_n = b_1\lambda_1^n + b_2\lambda_2^n + O(\lambda_3^n).$$

Коэффициенты a_i, b_i неотрицательны. Подстановкой (11) в выражении (10) найдем, что погрешность оценок (10) будет порядка k-й степени второго корня:

$$\Delta \cong \delta = (n-k)\left(\frac{\lambda_2}{\lambda_1}\right)^k (\lambda_1 - \lambda_2)\cdot (a_1b_2 + a_2b_1) - O\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}. \qquad (11)$$

Из соотношений (11) и (12) следует, что для изотропных рекуррентных сетей найденные оценки (10) вероятности связности фактически эквивалентны оценке максимального, первого, корня матрицы перехода с погрешностью $O(\lambda_2^{k-1})$. Так как анализируются сети, максимальный ранг которых равен k, то получим оценки первого уровня k-го порядка.

Коэффициенты a_i, b_i и корни λ_i в формуле типа (11) не зависят от ранга сети, их значения можно оценить, исходя из вероятности связности сетей меньшего ранга. Поэтому для изотропных сетей можно повысить уровень оценки, включив в рассмотрение несколько наибольших корней матрицы перехода. Погрешность оценки i-го уровня k-го порядка в этом случае будет $O(\lambda_{i+1}^{k-1})$, при малых k рост уровня оценки не всегда ведет к увеличению точности.

Ограничимся вторым уровнем. С погрешностью $O(\lambda_3^{k-1})$ соотношения (11), записанные для характеристик сетей $(k-2)$-го, $(k-1)$-го, k-го рангов дают шесть уровней, позволяющих однозначно определить необходимые компоненты формул (11) через значения вероятностей связности сетей. Для этого достаточно перейти к эквивалентной системе соответствующих конечно-разностных уравнений второго порядка, после чего нахождение оценочной матрицы перехода $\tilde{A}$ представляет чисто техническую задачу. Сохраняя принятые обозначения, получим

$$\tilde{A} = \frac{1}{\Pi_{k-2}P_{k-1} - P_{k-2}\Pi_{k-1}} \cdot \begin{vmatrix} \Pi_{k-2}P_k - \Pi_{k-1}P_{k-1} & P_{k-1}P_{k-1} - P_{k-2}P_k \\ \Pi_{k-2}\Pi_k - \Pi_{k-1}\Pi_{k-1} & \Pi_{k-1}P_{k-1} - P_{k-2}\Pi_k \end{vmatrix}. \quad (12)$$

Собственные значения λ_1, λ_2 матрицы A есть корни уравнения:

$$\lambda^2 - \frac{\Pi_{k-2}P_k - P_{k-2}\Pi_k}{\Pi_{k-2}P_{k-1} - P_{k-2}\Pi_{k-1}}\lambda + \frac{\Pi_{k-1}P_k - P_{k-1}\Pi_k}{\Pi_{k-2}P_{k-1} - P_{k-2}\Pi_{k-1}} = 0. \quad (18)$$

Вероятность связности сети оценивается в конечном виде:

$$P_n = \frac{P_k - \lambda_2 P_{k-1}}{\lambda_1 - \lambda_2}\lambda_1^{n-k-1} - \frac{P_k - \lambda_1 P_{k-1}}{\lambda_1 - \lambda_2}\lambda_2^{n-k-1} + O[(n-k)\lambda_3^{k-2}]. \quad (19)$$

Погрешность приложения второго уровня определяется третьим по величине корнем матрицы перехода, но в явном виде установить оценку не удалось. Имеются сильные основания ожидать, что по формулам (12)–(14) для вероятности P_n находится верхняя оценка. Существуют также определенные предпосылки для обобщения матрицы (12) на случай оценки характеристик неизотропных (произвольные надежности элементов) рекуррентных сетей, но и здесь строгого доказательства нет.

Численный эксперимент был проведен в широком спектре значений s и q, изменяющихся от нуля до 0,8–0,9. Ранг n и степень m сети варьировались от 5 до 100. Так как во всем диапазоне надежностей элементов зависимость $P_{n,m}$ от n и m при $m, n > 20$ вырождается в чистую экспоненту с показателем, пропорциональным произведению $m \times n$, то в качестве базового значения принято $n = m = 20$.

При $s << q$ точность оценок очень хорошая, при $s > q$ точность плохая. Многое, конечно, зависит от конкретных значений параметров надежности каналов, ранга и степени сети. Это согласуется с физическими соображениями: каждый столбец узлов играет роль фильтра, блокирующего влияние удаленных каналов на связь между данной парой узлов. В предельных случаях: когда $s = 0$ в установлении связи участвуют только узлы соседних рангов и оценка уже второго порядка дает точное значение; когда s близко к единице, в связи участвуют узлы удаленных рангов и необходим высокий порядок оценки.

Целесообразно проводить двойной цикл оценок для различных ориентаций сети, из полученных значений выбирать наилучшее.

В данной работе удалось обобщить точные методы анализа рекуррентных сетей на случай одномерных аддитивных сетей, найти конструктивные методы вычисления двухсторонних оценок вероятности связности. Идеи, положенные в основу методов, перспективны и для более широкого класса задач.

СПИСОК ЛИТЕРАТУРЫ

1. **Филин Б.П.** О методе экспресс-оценки надежности и коэффициенте потенциальной структурной неуязвимости связей в сложных системах // АиТ. № 5. С.158–182. 1994.

2. **Гадасин В.А. Ушаков И.А.** Надежность сложных информационно-управляющих систем. М.: Сов.радио, 1975.

3. **Гадасин В.А.** Интерполяционный метод оценки характеристик надежности ретрансляционных сетей с однородной структурой // АиТ. 1979.№ 8. С.172–179.

4. **Гадасин В.А.** Методы расчета структурной надежности сетей сложной конфигурации. М.: "Знание", 1984. С. 3–53.

5. **Романовский В.И.** Дискретные цепи Маркова. М.-Л.: Гостехиздат, 1949.

Тихомиров В.А., Котляров В.П.

Верификация программного обеспечения при интеграции новой функциональности в существующую систему

Введение

На сегодняшний день, в связи с постоянным увеличением объема и сложности разрабатываемых систем, процесс интеграции новых компонент в уже существующую версию программного продукта становится ключевым. Это связано с тем, что процесс разработки компоненты проводится часто независимо от других компонент, и от того, как будет проводиться интеграция компонентов, зависит работоспособность всей системы. Целью интеграционного тестирования системы, расширенной новой функциональностью, является проверка того, как новый модуль согласуется с остальной частью программного продукта (с остальными модулями продукта).

Использование при интеграционном тестировании проверки на моделях (*model checking*) — известного подхода к верификации и тестированию программных систем — затруднено по причине отсутствия средств настройки на локальную задачу — тестирование "стыка" система / новая компонента. Если формальной модели системы нет, то ради тестирования стыка нужно потратить много усилий для их создания. Если модели системы и новой компоненты есть, то из них нужно выделить часть, которая необходима в рамках данной задачи. При этом предполагается, что система и компонента протестированы по отдельности. Это можно делать путем изменения процесса верификации, а можно создать новую модель.

Рассмотрим, каким образом можно изменить процесс верификации. При верификации компоненты отдельно от системы вызов системных функций (то есть функций, внешних для компоненты) не может быть проверен. Но если системе верификации предоставить дополнительную информацию о системе (как реализованы соответствующие системные функции), то процесс интеграционной верификации будет проведен при верификации компоненты автоматически. Такой подход требует наличия полной спецификации компоненты, что в случае большого объема функциональности требует значительный объем ресурсов для полной верификации. Если компонента уже была протестирована, часть работы по верификации дублируется.

Рассмотрим другой подход к тестированию "стыка" — создание модели. В нее необходимо включить спецификации только тех участков кода системы и интегрируемой компоненты, которые отвечают за взаимодействие и создавая таким образом частичные модели (*partial specifications*) системы и компоненты. Последний вариант более выигрышный, потому что не требует наличия полных моделей системы и компоненты. Именно такой подход предлагается в данной статье. Несмотря на большое количество исследований в области частичных спецификаций (обзоры можно найти в работах [1, 2, 3]), до сих пор эта идея не использовалась для интеграционного тестирования ПО.

Наш подход использует в качестве языка построения моделей базовые протоколы Летичевского [4]. Этот формализм создан в Институте Кибернетики НАН Украины и предназначен для формализации поведенческих аспектов программных систем. В качестве формы записи базовых протоколов мы вслед за [5] используем *MSC*-диаграммы [4]. Эти диаграммы являются языком описания поведения системы в виде последовательности событий и хорошо согласуются с парадигмой базовых

протоколов. В качестве инструментальных технологий предложено использовать систему верификации *VRS* [5] и систему тестирования *TAT* [6], которые разработаны специально для базовых протоколов Летичевского.

В статье представлена схема и описание самого подхода, а также приводится пример, в котором в телекоммуникационную систему добавляется компонента, отвечающая за реализацию сервиса коротких сообщений (*Short Message Service — SMS*).

1. Частичные спецификации

При создании и использовании полной модели программного продукта возникают следующие трудности:

— создание такой модели требует больших трудозатрат;

— полные модели очень объемны;

— анализ и обработка модели требуют больших трудозатрат.

Построение и анализ моделей только для фрагментов программного продукта во многом решает описанные трудности. Такой подход носит название частичных спецификаций (partial specifications). Этот подход используется многими исследователями в различных областях программной инженерии (подробные обзоры содержатся в работах [1, 2, 3]). В статье приводится краткий анализ основных работ последних лет.

В [7] рассматривается ситуация, когда необходимо построить модель по требованиям к программной системе для последующего проведения тестирования. Для каждого требования строится своя спецификация на основе логики первого порядка. На выходе создается набор частичных спецификаций, каждая из которых соответствует определенному требованию. Разработка таких частичных спецификаций требует меньших трудозатрат по сравнению с полной спецификацией системы, поскольку не тратятся ресурсы на детальное описание связей между требованиями. Такая же идея лежит в основе использования базовых протоколов, используемых в данной статье как средство создания моделей ПО.

В [8] рассматривается способ задания конечного автомата с помощью частичных спецификаций. Описывается метод, позволяющий на основе частичных спецификаций обнаружить недостижимые или дублированные состояния автомата. После применения этого метода становится возможным упростить конечный автомат и обнаружить возможные ошибки в его описании.

В [9] с помощью частичных спецификаций, описывающих конечный автомат, решается задача получения тестов, проверяющих интересующую область функциональности автомата. Традиционно сложно достижимые состояния автомата являются проблемным участком тестирования. В статье показано, как с помощью частичных спецификаций протестировать сложнодостижимые состояния с заданным функциональным покрытием.

В работе норвежских ученых [2] для объектно-ориентированной системы предлагается создавать формальные спецификации для каждого из объектов в отдельности. Построение таких спецификаций для любого объекта аналогично процессу создания.частичных спецификаций всей системы, то есть может проводиться независимо от других объектов. Такой подход является актуальным в современном процессе разработки распределенных систем, когда разработкой многочисленных модулей и компонент системы занимаются разные команды инженеров. В работе приводится методика объединения частичных спецификаций для получения полной спецификации системы, которая впоследствии может быть подвержена верификации и тестированию. Похожие принципы заложены в объектно-ориентированный язык программирования *Eiffel*[1], где при задании методов классов описываются пред- и постусловия, задаются инварианты класса.

Тестирование взаимодействия системы с другими модулями на основе частичных спецификаций рассматривается в статье [10]. Для исследуемой системы строится диаграмма внешних вызовов. Затем среди различных цепочек вызовов выделяются наиболее частые и для них строятся спецификации, основанные на исходном коде системы. При верификации и тестировании построенные частичные спецификации позволяют сфокусироваться на наиболее вероятных сценариях поведения системы. Однако при таком подходе цепочки вызовов, которые не были включены в выборку, вообще не подвергаются верификации, что на практике отрицательно сказывается на качестве продукта.

[1] http://www.Eiffel.com

Выделяется работа японских ученых [11], в которой описывается методика построения частичных спецификаций программных компонентов, образующих целостную систему. По результатам исследования кода функциональность компоненты разделяется на два типа: с локальной и внешней областью действия. Фрагмент функциональности считается внешним, если ответственен за взаимодействие с другими компонентами: содержит вызовы их функций, использует глобальные переменные, доступные другим модулям. В противном случае он является локальным. После построения набора спецификаций становится возможным проводить тестирование внешнего взаимодействия отдельно от тестирования локальной функциональности компонента. Однако в статье основной акцент сделан на теоретические аспекты разделения функциональности, при этом практическое применение для реальных проектов не рассматривается.

2. Описание метода

Предложенный в данной статье метод представляет собой следующую последовательность основных шагов.

Шаг №1. Идентификация интерфейса взаимодействия.

Шаг №2. Идентификация регионов взаимодействующего кода.

Шаг №3. Построение модели.

Шаг №4. Верификация и тестирование.

Общая схема метода представлена на рис. 1.

Рассмотрим пошаговый процесс реализации метода.

Шаг №1. Идентификация интерфейса взаимодействия.

Все начинается с определения интерфейса взаимодействия. Артефакты компоненты, используемые системой, и артефакты системы, используемые компонентой, по сути, составляют интерфейс взаимодействия. Под артефактом понимается любая программная сущность, например функция, переменная или класс. Если доступен исходный код компоненты и системы, то артефакты, как правило, объявлены соответствующим образом (в отдельной секции программы или с использованием ключевого слова языка программирования). В случае, если исходный код недоступен, для некоторых языков программирования, возможно использование декомпиляторов. Формат записи интерфейса может, любым, при этом, наряду с именем артефакта и его описанием полезно описывать ограничения, накладываемые на артефакт.Например, с помощью формального языка зафиксировать тот факт, что функция вызывается в компоненте только с отрицательными параметрами. Добавление такой информации позволяет уже на данном шаге

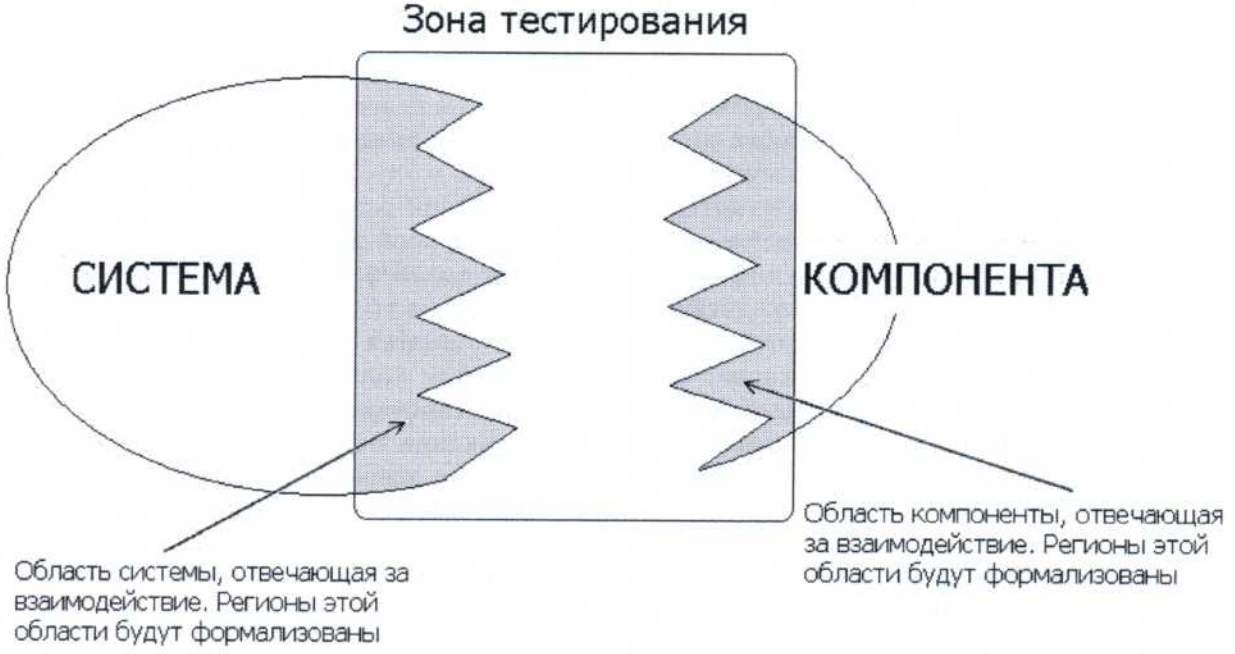


Рис.1. Схема метода

обнаружить ошибки. Результатом шага является построенный интерфейс взаимодействия компоненты и системы.

Шаг №2. Идентификация регионов взаимодействующего кода.

Для каждого артефакта интерфейса взаимодействия определяется область кода, в которой он реализован (для системных артефактов — в системе, для компонентных артефактов — в компоненте). Такие области назовем внутренними регионами. Поиск внутреннего региона сводится либо к поиску реализации (в случае функции или класса), либо к поиску инициализации и объявления (в случае переменной).

Под внешним регионом будем понимать область кода компоненты или системы, которая определяет параметры вызова внешней функции (изменения переменной или использования класса) и параметры, которые изменяются после внешнего вызова. Например, компонента в своем коде вызывает функцию системы с некоторым параметром. Внешним регионом вызова функции системы будет считаться то место в коде компоненты, которое определяет значение параметра перед вызовом функции системы, и те артефакты компоненты, которые изменяются после ее вызова.

Определение регионов осуществляется с помощью построенного на предыдущем шаге интерфейса взаимодействия. Поиск внешнего региона осуществляется следующим образом: если исходный код доступен, то производится поиск точек вызова функциональности системы из компоненты (или вызов функциональности компоненты из системы) в соответствии с построенным интерфейсом. Затем определяется окружение обнаруженной точки вызова. Чем более детально будет описаны внешних регионов и реализация для внутренних регионов, тем более будет точна спецификация интерфейса взаимодействия и результаты верификации.

На данном шаге удобно использовать автоматизированные кодоанализаторы (например, *KlocWork*[2]). Рассмотрим применение такого подхода на примере *Klocwork*. При создании нового проекта в *Klocwork*, содержащего компонент и систему, становятся видны все связи между ними. Каждой связи соответствует использование внешней функциональности с некоторыми параметрами. Отслеживание этих параметров в области вызова и реализации позволяет построить соответствующие регионы.

В случае, если код недоступен, в ряде случаев возможно использовать декомпиляцию кода.

Шаг №3. Построение модели.

На основе полученных регионов взаимодействующего кода системы и компоненты строится модель взаимодействия. Каждый регион подвергается формализации отдельно от других регионов в терминах базовых протоколов. В результате получаем набор спецификаций, соответствующий взаимодействующим регионам системы и компоненты. В случае внешнего региона спецификация описывает состояние компоненты или системы до вызова внешней функциональности, ее непосредственный вызов и состояние после внешнего вызова. Спецификация внутреннего региона описывает реализацию системной или компонентной функции. Таким образом, модель состоит из двух наборов спецификаций — формализованные регионы взаимодействующего кода системы и компоненты. Результат шага — построенная модель взаимодействия в терминах базовых протоколов.

Шаг №4. Верификация и тестирование.

На данном шаге построенная модель подвергается верификации и тестированию (например, используется система *VRS/TAT*). Другими словами, проверяется соответствие спецификаций окружения внешних вызовов (внешние регионы) и реализации внешних артефактов (внутренние регионы). На данном шаге осуществляется контроль синтаксиса базовых протоколов, контроль связанности базовых протоколов друг с другом и генерация тестовых трасс. Полученные тестовые трассы являются готовыми сценариями, обеспечивающими полное функциональное покрытие выделенных регионов. Трассы можно использовать в качестве тестовых процедур в ручном тестировании или для генерации автоматических тестов для конкретной платформы, например используя технологию *TAT*. Если в процессе верификации и тестирования были найдены критические ошибки, то исходная система подвергается переработке и процесс повторяется, начиная с первого шага. Критерий критичности ошибок в каждом проекте свой. Например, в случае прототипа допускаются ошибки, не мешающие демонстрации основной функциональности. После данного шага принимается решение об исправлении ошибок или готовности продукта.

[2] http://www.klocwork.com

3. Пример

Данный метод апробирован на индустриальном проекте, в котором в качестве системы выступало ПО мобильного устройства, а в качестве компоненты диспетчер сообщений, обеспечивающий отправку, прием и хранение различных типов сообщений. Итак, рассмотрим применение предлагаемого метода на примере отправки *SMS*-сообщений. В качестве новой компоненты выступает набор классов, отвечающих за *SMS*-сервис. Методы этих классов используют функции, которые реализованы в системных модулях.

Для отправки *SMS*-сообщения необходимо инициализировать адрес, по которому будет, осуществляется отправка — открыть соединение. Рассмотрим пример реализации открытия соединения, описанный в компоненте.

```
Class Connection {
    open();
    close();
}
function open(port) {
    if (port >=0) {
            SYS_OpenLowLevel(port); //вызов функции системы
            status = OPENED; //соединение открыто
    } else throw DataException;
}
```

Если значение порта (port) отрицательное, то функция *open* завершается (бросается исключение *DataException*). Системная функция *SYS_OpenLowLevel* обеспечивает установление соединения на низком уровне. После вызова метода *SYS_OpenLowLevel* соединение считается открытым (*status = OPENED*).

Рассмотрим применение метода по шагам.

Шаг №1. Идентификация интерфейса взаимодействия.

Компонента использует функцию системы *SYS_OpenLowLevel*. При этом система не использует функции компоненты, поэтому интерфейс взаимодействия ограничивается одной функцией.

Шаг №2. Идентификация регионов взаимодействующего кода.

Внешним регионом для *SYS_OpenLowLevel* является область кода функции *open*, где фильтруются значения номера порта. Внутренним регионом является область кода в системе, где эта функция реализована. Ниже представлена реализация данной функции:

```
function SYS_OpenLowLevel(port) {if (port <= 0) TERMINATE_PROGRAM; else SYS_HWOPEN; }
```

Шаг №3. Построение модели.

Для формализации внешнего региона необходимо рассмотреть только случай неотрицательного значения порта (в случае отрицательного значения вызов внешней функции не производится).

На рис. 2 представлена спецификация на языке *MSC*, описывающая внешний регион компоненты.

Рассмотрим спецификацию детально. На рис 2. показано взаимодействие двух сущностей — *Connection* (соответствует классу *Connection*) и *System*(Система). Предусловие *PRE* требует, чтобы соединение было в состоянии "закрыто". Постусловие *POST* показывает состояние соединения после вызова *SYS_OpenLowLevel*. В соответствии с реализацией соединение должно открыться (*STATUS = = OPENED*). Событие *SYS_OpenLowLevel* соответствует вызову функции *SYS_OpenLowLevel*. В качестве параметра события выступает макрос *#nonnegative_ports*, который принимает множество неотрицательных значений.

При формализации внутреннего региона системной функции *SYS_OpenLowLevel* рассмотрим случай, когда порт равен нулю. MSC-диаграмма для этого случая представлена на рис. 3.

При реализации шага 3 система верификации для построенной модели обнаружила ошибку (см. рис. 2 и рис. 3). Когда значение порта нулевое, компонента вызывает системную функцию *SYS_OpenLowLevel*, которая терминирует выполнение программы при нулевом значении порта согласно ее спецификации. Ошибки можно было бы избежать, добавив проверку на ноль в компоненту (вместо "*if (port >= 0)*" написать "*if (port > 0)*").

Шаг №4 после построения спецификаций взаимодействия относится к технической части метода и не представляет практического интереса после того, как спецификации построены.

Заключение

В рамках данной статьи предложена методика тестирования взаимодействия компоненты и системы. Кроме того, методика также позволяет обеспечить тестирование взаимодействия нескольких компонент или нескольких систем.

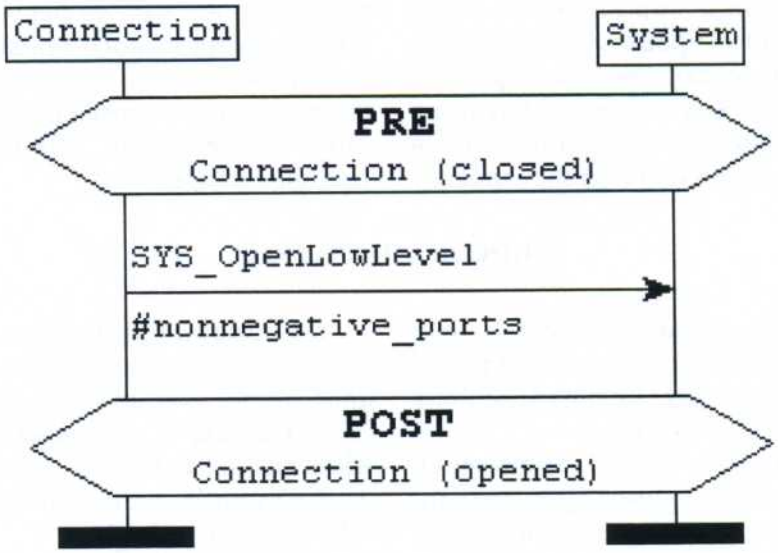


Рис.2. Внешний регион

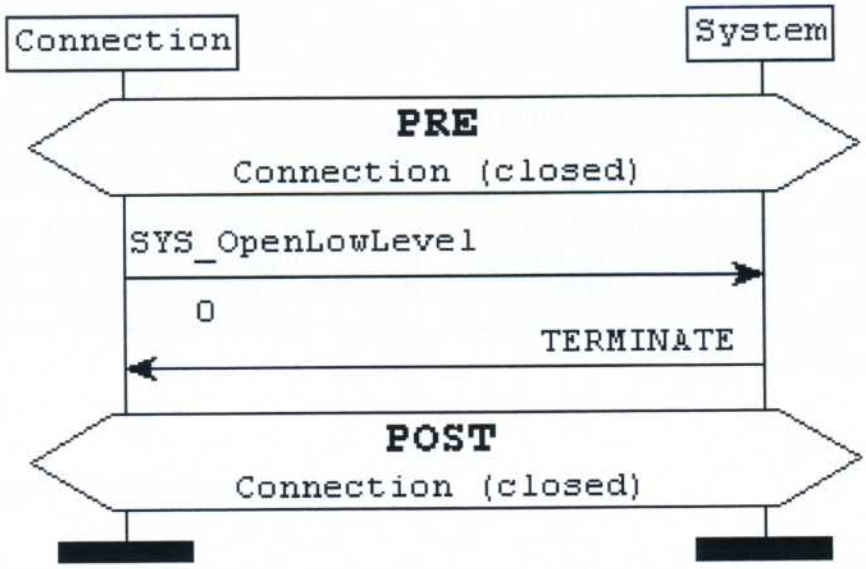


Рис. 3. Внутренний регион

Предлагаемый подход автоматизирует тестовый процесс и гарантирует полное покрытие функциональности при генерировании тестов. Данная методика апробирована на нескольких индустриальных проектах по разработке встроенного программного обеспечения. Предварительные оценки трудоемкости на примерах проектов среднего размера (около 100 требований) подтвердили его эффективность: цикл тестирования был сокращен на 3 человека-месяца по сравнению с обычным тестированием. Кроме того, были найдены дополнительные ошибки. В настоящее время метод применяется в больших индустриальных проектах.

В дальнейшем планируется расширить функциональность системы верификации для проведения интеграционных проверок на основе предложенного метода. Теорети-

ческая составляющая основа верификатора построена и проходит апробацию в больших индустриальных проектах разработки ПО. Также планируется определить формат и способы составления словарей внешних вызовов функциональности компоненты и системы. Формализация словарей упростит многие шаги метода и позволит проводить верификацию уже на первых этапах разработки программного продукта. Разработка указанных улучшений предоставит возможность полной автоматизации интеграционного тестирования, обеспечивая заданное тестовое покрытие.

СПИСОК ЛИТЕРАТУРЫ

1. **Easterbrook, S., Callahan, J.** 1997. Formal Methods for Verification and Validation of partial specifications: A Case Study. Virginia University symposium, vol. 1, 1997, 26–37.

2. **Johnsen, E. B., Owe, O.** 2002. Composition and Refinement for Partial Object Specifications. Parallel and Distributed Processing Symposium, 2002, 210–217.

3. **Hendrix, J., Clavel, M., Meseguer, J.** 2005. A Sufficient Completeness Reasoning Tool for Partial Specifications. Proceedings of the 16h International Conference on Rewriting. LNCS, vol. 3467. 2005. May 31. Springer, 165–174.

4. **Letichevsky, A.A., Kapitonova, J.K.** 2004 Basic Protocols, Message Sequence Charts, and the Verification of Requirements Specifications. Proceedings of International Workshop, WITUL. 2004, 30–38.

5. **Baranov, S.N., Kotlyarov, V.P., Letichevsky, A.A.** 2005. The technology of automation verification and testing in industrial projects. IEEE Russia Northwest Section, The International Scientific Conference "110 Anniversary of Radio Invention" Saint Petersburg, 2005, 12–34.

6. **Kotlyarov, V.P., Golubev, A.A., Karpov, A.N.** 2006. Testing Automation for system core kJava applications. Consumer Electronics, 2006, 1–4.

7. **Falcone, Y., Fernandez, J.C., Mounier, L., Richier ea**. 2007. A Compositional Testing Framework Driven by Partial Specifications. TestCom FATES, 2007, 107–122.

8. **Petrenko, A., Yevtushenko, N.** 2005. Testing from Partial Deterministic FSM Specifications. IEEE Trans. Computers, vol.54, No. 9, 2005, 1154–1165.

9. **Petrenko, A., Yevtushenko, N.** 2000. On Test Derivation from Partial Specifications. Procedings of IFIP Joint International Conference Formal Description Techniques, 2000, 85–102.

10. **Acharya, M., Xie, T., Pei, J. ea.** 2007. Mining API patterns as partial orders from source code: from usage scenarios to specifications. Proceedings of SIGSOFT Seminar, 2007, 25–34.

11. **Aoshima, T., Yonezaki, N.** 2000. An Efficient Tableau-Based Verification Method with Partial Evaluation for Reactive System Specifications. Proceedings of EJC Conference, 2000, 363–374.

Мезенцева О. С., Алексеев А. И.

Применение аппарата пороговых схем и модулярной арифметики для повышения производительности распределенных вычислительных систем

Распределенные вычислительные системы, основанные на широко используемом в задачах криптографии аппарате пороговых схем, имеют ряд отличительных особенностей. Основными преимуществами таких систем являются высокая устойчивость к потере связи между участниками и возможность проверки информации, поступающей от каждого узла. Малая чувствительность к выходу одного или нескольких участников из процесса вычислений следует из избыточности хранимых данных, однако для распределенных систем, подверженных частой потере связи с вычислительными узлами, такой подход представляется оправданным. Так, в [1] освещается

применение пороговых схем для создания системы распределенного хранения данных с регулируемой избыточностью. В данной работе предлагаются методы повышения производительности распределенных вычислительных систем, основанных на аппарате пороговых схем.

Наиболее часто используемым вариантом пороговой схемы является схема разделения секрета Шамира. Метод Шамира позволяет создать по блоку D информации n теней таким образом, что D может быть восстановлено из любых $k < n$ теней, но по $k-1$ тени D невозможно получить никакой информации о D.

Будем рассматривать элементы $P = GF(2^q)$ как последовательности длиной q бит. Существует k-мерное пространство L векторов над P. Любая исходная последовательность элементов P может быть представлена в виде набора векторов из L. Пусть $f = \{l_1, l_2, \ldots, l_m\}$ – некоторая исходная последовательность векторов из L, и пусть $E = \{e_1, e_2, \ldots, e_n\}$ – набор n векторов из L, такой, что любые k из них образуют базис в L. Тогда каждому вектору e_i из E можно поставить в соответствие последовательность скалярных произведений $\{(l_1, e_i), (l_2, e_i), \ldots, (l_m, e_i\}$. Пусть имеются произвольные k частей построенного набора. Обозначим соответствующие им вектора через $e_1, e_2, \ldots, e_k$. По условиям выбора E имеющиеся k векторов образуют базис в L, следовательно, матрица $A_{k\times k}$, строки которой – вектора $e_1, e_2, \ldots, e_k$, имеет обратную матрицу A^{-1}. Тогда $S_i = A \times l_i$ – столбец, состоящий из i-ых элементов всех частей. Таким образом, процедура сборки исходной последовательности имеет вид $l_i = A^{-1}S_i$ [1].

Принципиальная схема алгоритма, предложенного в [1], показана на рисунке 1.

Помимо упомянутых проверяемости вычислений и малой чувствительности к потере участников, одной из отличительных особенностей такого подхода является то, что хранимые данные могут собираться не полностью, а лишь в том объеме, который необходим на данный момент [1], причем сборка каждой части может производиться независимо от других.

Приведенные характеристики позволяют предположить, что подобная схема хранения данных позволит существенно повысить надежность и производительность вычислений, проводимых в распределенных грид-системах. Одним из возможных объектов внедрения является крупнейшая научная грид-сеть СНГ "СКИФ Полигон" (инициатива суперкомпьютерной программы "СКИФ-ГРИД" Союзного государства), участником которой является Северо-Кавказский государственный технический университет. Однако высокие требования к интенсивности обмена данными между участниками вычислений и высокая пропускная способность элементов сети предъявляют столь же высокие требования к быстродействию системы хранения данных. Таким образом, чтобы избежать ситуации, в которой система хранения становится узким местом всей сети, необходимо добиться максимально возможной ее производительности.

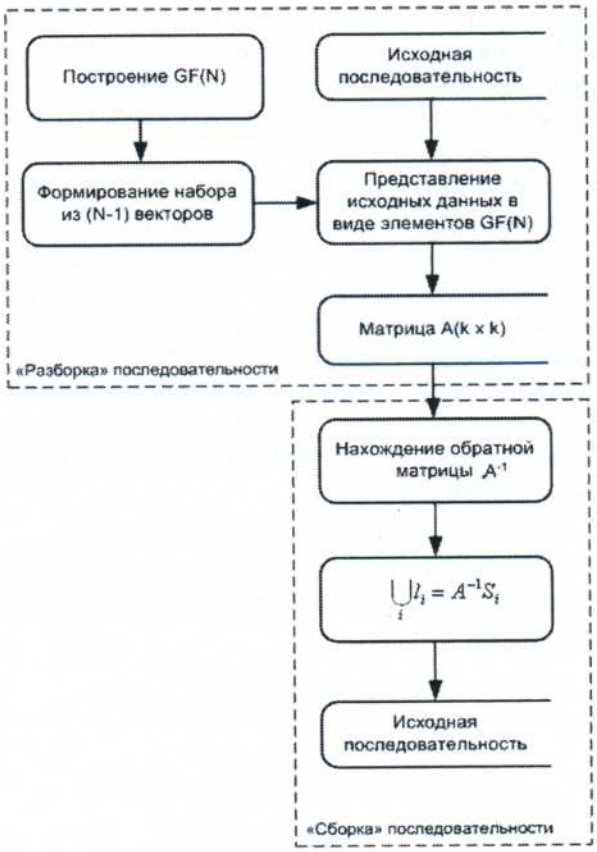


Рис. 1. Схема алгоритма "сборки-разборки"

Наиболее трудоемкими частями алгоритма являются нахождение матрицы, обратной A, и операция сборки $A^{-1}S_i$.

Согласно [1] $A^{-1}S_i$ состоит из k^2 операций умножения и сложения в поле Галуа, что в свою очередь представляет собой k^2 операций обычного сложения и обычного умножения и $2k^2$ операций % (вычисление остатка от деления на N). Учитывая, что число тактов на выполнение операции % для современных SISD-процессоров превышает число тактов на выполнение операции умножения в среднем в четыре раза,

становится ясно, что значительное увеличение общей трудоемкости алгоритма связано именно с этим участком вычислений.

Для решения проблемы оптимизации проблемного участка предлагается использовать математический аппарат системы остаточных классов. Выбор СОК диктуется малой разрядностью остатков, используемых для представления чисел, и возможностью параллельной обработки, следующей из независимости разрядов друг от друга [2]. Наиболее выигрышным вариантом представляется переход в СОК на этапе построения $GF(N)$ и формирования матрицы A. Однако повышенная сложность реализации и наличие эффективных параллельных алгоритмов, оптимизирующих выполнение стадии разборки, приводят к решению выполнять перевод в СОК после обращения матрицы A. Такой подход позволит выполнять трудоемкую операцию перехода ПСС → СОК лишь единожды.

Рассмотрим процедуру сборки-разборки с использованием СОК при $n = 3, k = 2, m = 2$. Для элементов поля будем использовать десятичное представление. Пусть исходная последовательность $f = \{l_1, l_2\} = \{(1, 2), (3,4)\}$. Набор векторов E построим таким образом, чтобы матрица, составленная из их компонентов, являлась матрицей Вандермонда:

$$E = \{e_1, e_2, e_3\} = \{(1, 1), (1, 2), (1, 3)\}.$$

Поставим в соответствие векторам из E последовательность скалярных произведений вида (l_i, e_j):

$$e_1 : \{(l_1, e_1), (l_2, e_2)\} = \\ = \{(1,2) \cdot (1,1), (3,4) \cdot (1,1)\} = \{3,7\},$$

$$e_2 : \{(l_1, e_2), (l_2, e_2)\} = \\ = \{(1,2) \cdot (1,2), (3,4) \cdot (1,2)\} = \{5,3\},$$

$$e_3 : \{(l_1, e_3), (l_2, e_3)\} = \\ = \{(1,2) \cdot (1,3), (3,4) \cdot (1,3)\} = \{7,15\}.$$

Пусть

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}.$$

Тогда:

$$A^{-1} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix},$$

$$S_1 = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \times \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix},$$

$$S_2 = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \times \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 7 \\ 11 \end{pmatrix}.$$

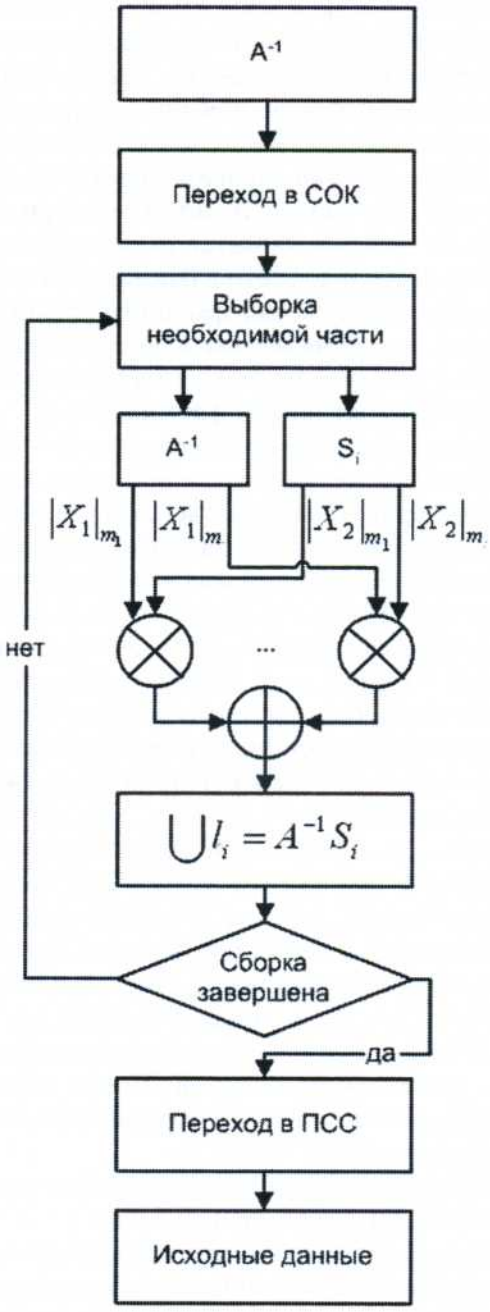


Рис. 2. Схема сборки исходной последовательности

Осуществим перевод коэффициентов A^{-1} и S_1 в СОК по основаниям (2,3,5), для представления отрицательных чисел воспользуемся способом введения знака в неявном виде [2]:

$$A^{-1} = \begin{pmatrix} (0,2,2) & (1,2,4) \\ (1,2,4) & (1,1,1) \end{pmatrix},$$

$$S_1 = \begin{pmatrix} (1,0,3) \\ (1,2,0) \end{pmatrix}.$$

Произведем сборку первой части исходной последовательности:

$$l_1 = A^{-1}S^1 = ((0,2,2)\cdot(1,0,3) + (1,2,4)\cdot(1,2,0), (1,2,4)\cdot(1,0,3) + (1,1,1)\cdot(1,2,0)),$$

$$l_1 = ((1,1,1), (0,2,2)).$$

Выполнив перевод в ПСС, получим $l_1 = (1,2)$. Сборка второй части исходной последовательности производится аналогично.

Таким образом, модифицированная схема алгоритма "сборки" может быть представлена в следующем виде (рисунок 2), где в качестве сумматора и умножителя предлагается использовать сумматор без использования LUT-таблиц и умножитель, основанный на исчислении степеней (умножитель Галуа) [3].

Рассмотрим возможные пути повышения быстродействия на стадии нахождения матрицы, обратной A. При малых значениях k предлагается использовать метод вычисления обратной матрицы, основанный на рекурсивной блочной схеме умножения матриц в QT-формате [4]. Указанная схема особенно эффективна в случае, когда коэффициенты матрицы являются элементами конечного поля, характеристика которого не превосходит $2\cdot10^8$ [4].

Для значений $k > 1286$ повышение производительности может быть достигнуто применением алгоритма Штрассена для нахождения обратной матрицы [5]. Трудоемкость алгоритма умножения Штрассена, на котором базируется одноименный метод отыскания обратной матрицы, составляет $O(n^{\log_2 7})$ против $O(n^3)$ для традиционного метода. Используя модификации метода Штрассена для параллельных вычислений [6], можно добиться существенного снижения порога значений k, при котором становится целесообразным применение этого алгоритма (до $k > 64$). В этом случае схема вычисления обратной матрицы примет вид, показанный на рисунке 3.

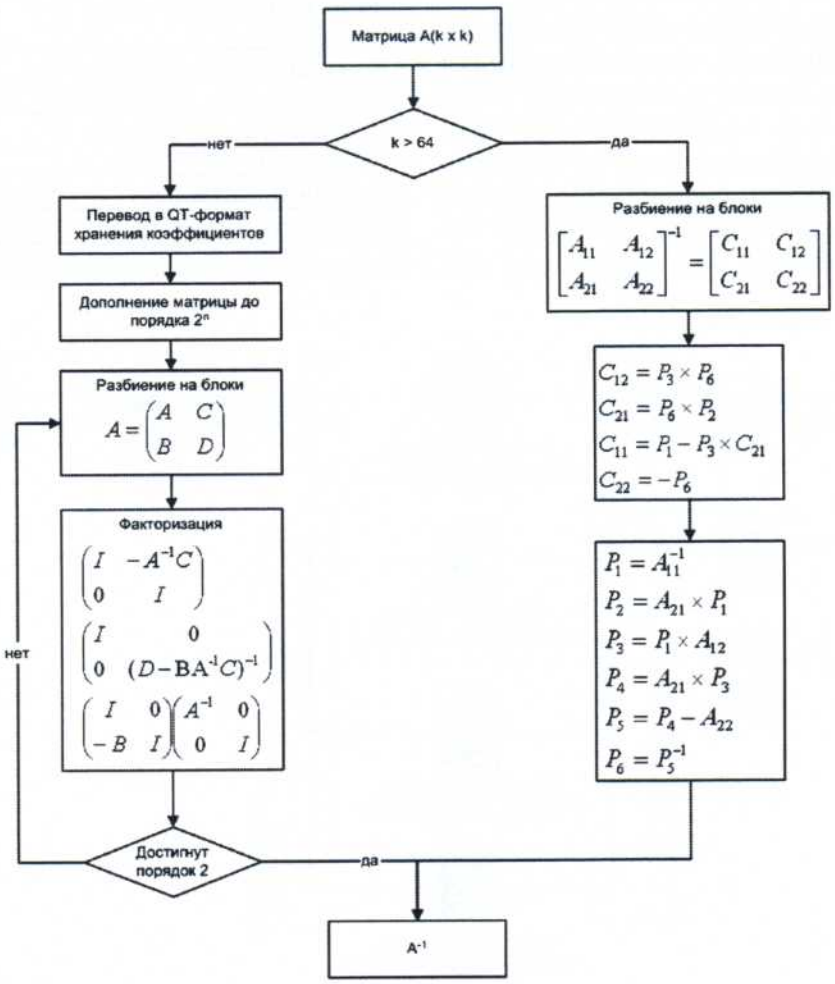


Рис. 3. Схема вычисления обратной матрицы

Естественная параллельность вычислений в СОК говорит в пользу реализации их на аппаратной базе векторных процессоров или вычислительных кластеров, составленных из процессоров традиционной SISD-архитектуры [7].

Одной из наиболее высокопроизводительных моделей, реализующих векторную архитектуру, является разработка российской компании НТЦ "Модуль" – процессор NM6403, важнейшей особенностью которого является работа с операндами произвольной длины (даже не кратной степени двойки) в диапазоне 1–64 бит. Умение динамично в процессе вычислений изменять разрядность операндов позволяет значительно повысить производительность [8].

В качестве аппаратной платформы для реализации модифицированного алгоритма используется вычислительный кластер университета, включающий как векторные, так и SISD-процессоры. Характеристики кластера приведены в таблице 1, схема – на рисунке 4.

В работе были проанализированы факторы, влияющие на быстродействие систем, функционирующих на основе аппарата пороговых схем, были предложены пути устранения узких мест алгоритма за счет применения параллельных методов вычислений, в частности, с использованием модулярной арифметики. Разработана и построена аппаратная платформа для проведения вычислений в системе остаточных классов, на базе которой были реализованы отдельные части предложенного алгоритма. Среди основных проблем, подлежащих решению в дальнейшем, первоочередными являются задача перехода к модулярной арифметике на самой ранней стадии – формировании $GF(N)$, повышение степени параллелизма всех частей алгоритма, тщательная оценка производительности и ликвидация выявленных узких мест.

Таблица 1

Технические характеристики кластера

Число узлов	4
Число процессоров Intel Xeon 2.4 GHz на узел	2
Объем оперативной памяти на узел (Гб)	1
Объем дисковых накопителей на узел (Гб)	160
Число векторных процессоров NM6403	2
Расчетная пиковая производительность R_{peak} (GFlops)	19,5
Максимальная производительность R_{max} (GFlops)	14,2
Операционная система	Gentoo Linux 2007.0
Коммуникационная среда	LAM-MPI 7.1.2
Коммутатор	Gigabit Ethernet

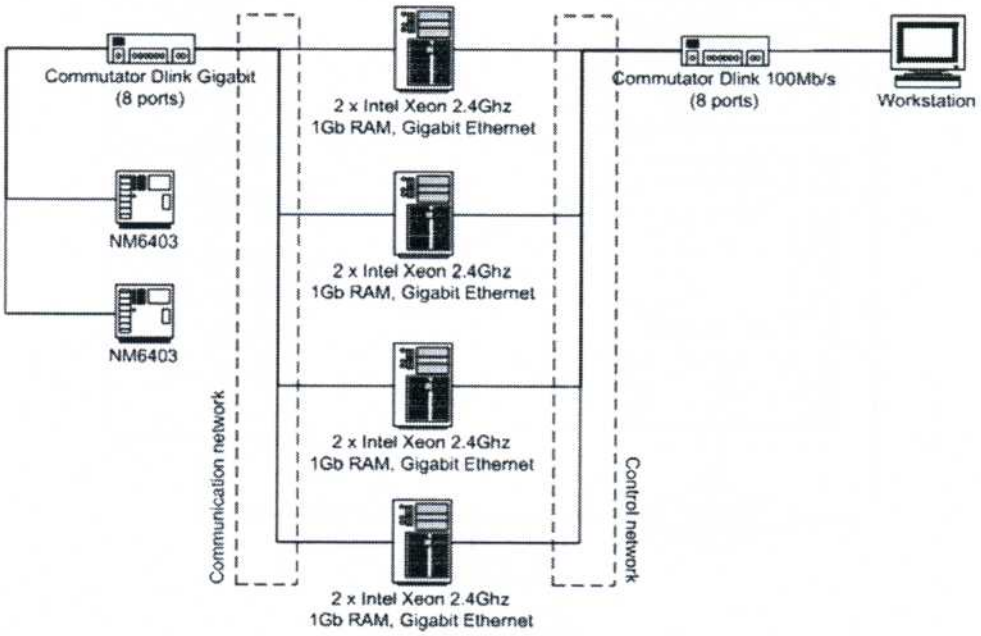


Рис. 4. Схема вычислительного кластера

СПИСОК ЛИТЕРАТУРЫ

1. **Тормасов А. Г.** Модель распределенного хранения данных с регулируемой избыточностью [Текст] / А. Г. Тормасов, М. А. Хасин, Ю.И. Пахомов // Электронный журнал "Исследовано в России", 35 – 2001. – С. 355–364. http://zhurnal.ape.relarn.ru/articles/2001/035.pdf

2. **Червяков Н. И.** Модулярные параллельные вычислительные структуры нейропроцессорных систем [Текст] / Н. И. Червяков, П. А. Сахнюк, А. В. Шапошников, С. А. Рядков. Физматлит – 2003. – 288 с.

3. **Червяков, Н. И.** Принципы построения модулярных сумматоров и умножителей [Текст] / Н. И. Червяков, И. В. Дьяченко // Сборник научных трудов М.– Зеленоград. – 2006. – С. 447–510.

4. **Малашонок Г. И.** Параллельные алгоритмы компьютерной алгебры [Текст] / Г. И. Малашонок, А. И. Аветисян, Ю. Д. Валеев, М. С. Зуев. Труды Института Системного Программирования РАН, т. 8, ч. 2 – 2004. С. 169-180.

5. **Malaschonok G. I.** Complexity Considerations in Computer Algebra [Текст] / G. I. Malaschonok // Computer Algebra in Scientific Computing, CASC. Techn. Univ. Munchen, Garching, Germany – 2004. – С. 325–332.

6. **Bailey D. H. A** Strassen-Newton algorithm for high-speed parallelizable matrix inversion [Текст] / D. H. Bailey // Conference on High Performance Networking and Computing – 1988. – С. 419–424.

7. **Мезенцева О. С.** О возможностях реализации модулярной арифметики на процессоре Л1879ВМ1 (NM6403) [Текст] / О. С. Мезенцева, А. И. Алексеев // Вестник Северо-Кавказского государственного технического университета №2 (11) – 2007. – С. 29–35.

8. НТЦ "Модуль", Архитектура процессора цифровой обработки сигналов Л1879ВМ1 (NM6403) – 2006. http://www.module.ru/files/nm6403arch-r.pdf

Дронь В. О., Марков А. М., Наумов А. С.

Сокращение объема вычислений при случайном поиске в задачах пеленгования

Введение

В современных комплексах радиопеленгования решается задача определения направления на источник радиоизлучения (ИРИ) путем измерения разностей пространственно-информативных параметров радиоизлучений в различных элементах распределенной антенной системы. Из совокупности значений измеряемых параметров, взаимных удалений между антенными элементами и номинала частоты радиоизлучения формируется некоторая многопараметрическая решающая функция, аргументом которой является орт направления на источник радиоизлучения. Наиболее вероятное направление на источник определяется ортом, при котором данная функция принимает экстремальное значение. Поиск глобального экстремума функции данного вида не имеет аналитического решения и осуществляется численным методом. Очевидно, что наиболее точное решение может быть получено при полном переборе бесконечной области допустимых значений направлений на источник.

Применяемый метод случайного поиска [1] дает удовлетворительные по точности решения, однако объем производимых при этом вычислений настолько велик, что по временным затратам соизмерим с длительностью сигналов современных видов передач. Актуальность сокращения объема необходимых вычислений определяется возникшими ограничениями на применение метода пеленгования.

1. Описание задачи

Целью данной работы является существенное сокращение вычислительного ресурса при незначительном увеличении ошибки определения искомого направления в решающей функции.

В данной работе в качестве пространственно-информативного параметра использовалась фаза радиоизлучения [2].

В качестве входных данных задачи были определены:

- N – количество антенных элементов;
- (x_i, y_i, z_i) – координаты i-го антенного элемента (АЭ) в системе координат антенны;

для удобства, но не умаляя общности рассматривались плоские антенные системы с расположением антенных элементов в плоскости xOz;

- λ – длина волны;
- $\Delta\phi_j$ – измеренная разность фаз между j-ым и нулевым АЭ.

Выходными данными является орт направления на источник.

2. Предварительные построения

2.1. Определение направления на источник по двум антенным элементам

Рассмотрим пару антенных элементов. Если нам известна разность расстояний от ИРИ до заданных антенных элементов, то источник необходимо лежит на двуполостном гиперболоиде вращения с фокусами в антенных элементах. Поскольку расстояние до ИРИ много больше размеров антенны, заменим гиперболоид конусом. Отметим, что конус определяется осью (линия расположения антенных элементов) и образующим углом (половина угла раствора).

2.2. Построение конусов

Рассмотрим два антенных элемента с индексами i и j и полученную на них разность разностей фаз $\Delta\phi(i,j) = \Delta\phi_i - \Delta\phi_j$. Рассмотрим также величину $d\phi(i,j) = 2\pi/\lambda \times \Delta r(i,j)$. Поскольку мы получаем величину $\Delta\phi(i,j)$ с точность до 2π (а именно $\Delta\phi(i,j) = d\phi(i,j) \ mod \ 2\pi$, то есть $d\phi(i,j) = \Delta\phi(i,j) + 2\pi k$, $k \in \mathbb{Z}$), то мы не можем однозначно вычислить $\Delta r(i,j)$ — разность расстояний от источника излучения до выбранных АЭ.

Однако, очевидно, разность расстояний $\Delta r(i,j)$ не может превышать по модулю $d(i,j)$ — расстояния между элементами, то есть $|\Delta r(i,j)| \le d(i,j)$, равенство достигается, когда источник лежит на прямой (E_iE_j). Из этого следуют следующие ограничения:

$$-2\pi/\lambda \times d(i,j) \le d\phi(i,j) \le 2\pi/\lambda \times d(i,j),$$

то есть

$$-2\pi/\lambda \times d(i,j) \le \Delta\phi(i,j) + 2\pi k \le 2\pi/\lambda \times d(i,j). \quad (1)$$

Множество значений k, удовлетворяющих соотношению (1) для пары (i, j), обозначим через K_{ij}. Число таких k, $|K_{ij}|$, очевидно, конечно. Следовательно, для каждой пары элементов можно получить конечное семейство конусов возможных направлений на источник. Примеры различных конусов на два источника приведены на рис. 1.

Обозначим через $\mu(i,j)$ угол поворота против часовой стрелки луча $[E_iE_j)$ относительно прямой, параллельной оси Ox и проходящей через точку E_i. Очевидно, что $\mu(i,j) = \mu(j,i) + \pi$. Через $\rho(i,j)_k$ обозначим образующий угол k-того конуса для пары элементов (i,j).

Углы $\rho(i,j)_k$ находятся следующим образом:

$$\rho(i,j)_k = \pi - \arccos\{(\Delta\phi(i,j) + 2\pi k) / /(2\pi/\lambda \times d(i,j))\} \ \forall k \in K_{ij}. \quad (2)$$

Поскольку источник удален от антенны на расстояние, которое много больше размеров антенны, будем считать, что все конусы имеют общую вершину — точку О. Тогда для каждой пары (i, j) центры всех конусов задаются величиной $\mu(i,j)$ (это угол поворота оси симметрии конуса относительно Ox), а образующие углы — $\rho(i,j)_k$. Среди $|K_{ij}|$ конусов для каждой пары (i, j) есть только один истинный (т.е. тот, на котором действительно расположен источник). Пересечением истинных конусов для всех пар АЭ являются две прямые, симметричные относительно плоскости xOz (в которой расположена антенна). Причем только одна из этих прямых лежит в допустимом полупространстве и представляет собой истинное направление на источник.

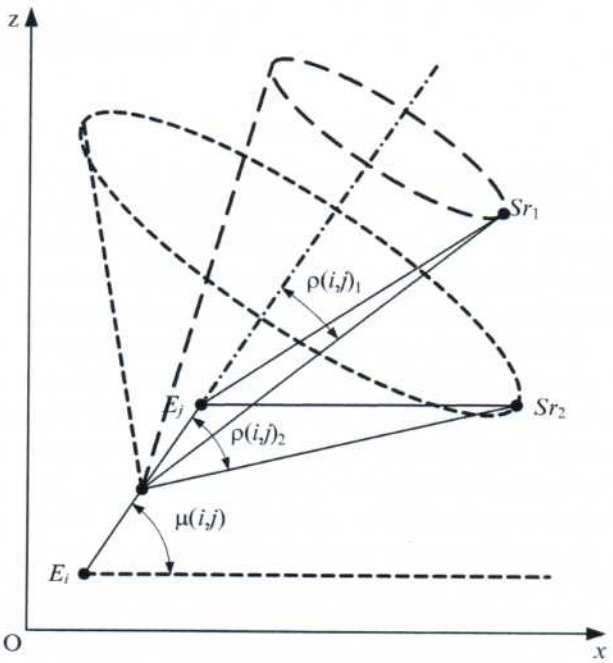


Рис. 1. Конусы возможных направлений

Предположим, источник находится от антенны на некотором (большом) расстоянии R, то есть лежит на сфере радиуса R с центром в точке O. Для каждой пары АЭ пересечение

семейства конусов и этой сферы дает набор окружностей. "Центром семейства" всех окружностей этого набора будем называть радиус-вектор точки пересечения общей оси семейства конусов со сферой, а радиусом для каждой окружности набора назовем угловое расстояние на сфере между центром ее семейства и любой точкой окружности. Ниже для каждой пары АЭ будем говорить именно о наборе окружностей на некоторой сфере радиуса R.

Описанные выше конусы и построенные с их помощью окружности можно использовать при решении задачи определения направления на источник.

3. Методы определения направления на источник

3.1. Метод наибольшего количества пересечений

Количество всех возможных пар АЭ равно $N(N-1)/2$ (пары (i, j) и (j, i) считаются одинаковыми). Для каждой из этих пар строим, как было изложено выше, все возможные окружности, среди которых только одна (для каждой пары АЭ) является истинной, т.е. той, на которой действительно располагается источник. Отметим, что окружности, соответствующие одной паре АЭ, не пересекаются между собой. Таким образом, если нет ошибок измерения и вычислительных погрешностей, в истинной точке должно пересекаться ровно $N(N-1)/2$ окружностей. Поскольку во входные данные внесены ошибки, то будем считать истинной точкой ту, в которой пересекается наибольшее число окружностей.

3.2. Метод построения окружностей

Этот метод использует следующий факт: искомая точка (при отсутствии ошибок измерения фаз) лежит на одной из окружностей, построенных для некоторой пары.

Таким образом, можно ограничить область поиска истинной точки, то есть перебирать не все точки решетки, а только точки, лежащие на окружностях.

Для более эффективного определения истинной точки имеет смысл рассматривать несколько пар, причем таких, чтобы использовались все антенные элементы.

3.3. Метод построения точек пересечения окружностей

Любые две пары АЭ дают набор окружностей. Если рассмотреть точки пересечения всех окружностей из первого набора со всеми окружностями из второго, то одна из них будет являться истинной при отсутствии ошибок измерения и вычислений. Таким образом, если бы в исходных данных не было ошибок, то достаточно было бы для каждой из точек пересечений посчитать значение функции корреляции и выбрать точку с максимальным значением. Поскольку ошибки в исходных данных есть, значения функции корреляции во всех точках будут меньше теоретического максимума. Однако точка пересечения с максимальным значением функции корреляции часто находится достаточно близко к истинной. Чтобы уменьшить влияние ошибок, можно перебирать также точки пересечения для других пар АЭ. Для повышения точности необходимо использовать большее количество пар АЭ, вплоть до всех возможных паросочетаний.

4. Результаты моделирования

Метод наибольшего количества пересечений использует свою собственную целевую функцию, отличную от фазово-корреляционной. В результате для получения тех же точностных характеристик затрачивается в среднем сопоставимое и даже большее время, что и при случайном поиске.

Метод построения окружностей и метод пересечения окружностей были реализованы и сравнивались со случайным поиском по решетке. Во всех случаях моделировались комплексные напряжения для всех точек координатной решетки в области видимости антенны с шагом один градус. При этом в фазу вносилась случайная ошибка с нулевым смещением и средним квадратичным отклонением 15°.

Для определения промаха между истинным направлением и полученным в результате работы конкретного алгоритма использовалась обычная евклидова норма $d = \|x\|_2$. Ситуацию, когда $d > 2$, назовем выбросом. Основной оцениваемый параметр рассматриваемых алгоритмов – процент выбросов.

Параметры методов для заданной длины волны радиосигнала (порядка 30% от характерного размера антенной системы) подбирались таким образом, чтобы процент выбросов был порядка единицы. Также методы сравнивались при тех же параметрах и в два раза меньшей длине волны. В табл. 1 приведены результаты моделирования.

Таблица 1

Сравнение различных методов.

	Процент выбросов	Количество вычислений корреляционной функции	Время работы, нс
$\lambda \sim 30\%$			
Случайный поиск	1.09%	443	398 284
Поиск по окружностям	0.98%	239	347 208
Поиск по пересечениям окружностей	0.85%	56	79 271
$\lambda \sim 15\%$			
Случайный поиск	2.11%	1 448	592 314
Поиск по окружностям	1.71%	366	290 040
Поиск по пересечениям окружностей	0.01%	130	172 121

Видно, что предлагаемые методы при тех же точностных характеристиках дают выигрыш по скорости работы в несколько раз.

Заключение

При равных точностях определения направления сокращение объема вычислений в первом приближении можно оценить отношением площади области поиска решающего орта к суммарной площади апертуры антенной системы. Эта величина может достигать десятков децибел и позволит расширить применимость метода пеленгования для короткоживущих сигналов в реальном масштабе времени.

Отметим, что остались непроработанными вопросы о выборе пар антенных элементов, в результате которого обеспечивалась бы задаваемая точность пеленгования при минимальном времени работы алгоритма.

СПИСОК ЛИТЕРАТУРЫ

1. **Бейко И. В., Бублик Б. Н., Зинько П. Н.** Методы и алгоритмы решения задач оптимизации. — Киев : Вища шк., 1983. — 512 с.

2. **Гришин Ю.П., Ипатов В.П., Казаринов Ю.М.** / Под ред. Ю.М.Казаринова. Радиотехнические системы. — М.: Высшая школа, 1990. — 496 с.

Березкин А.А.

Построение оптимальных нейронных декодеров блоковых кодов

Актуальность проблемы. Динамичный переход нашей технологической цивилизации на цифровые системы обработки и передачи информации создает много проблем при проектировании современных систем телекоммуникаций. Одной из важнейших задач, которые при этом необходимо решать во всех подобных системах, является обеспечение высокой достоверности передачи данных. Очевидно, что только при безусловно правильно переданных цифровых данных, поступивших на дальнейшую обработку, имеет смысл развивать весьма капиталоемкие технологии информатизации мирового сообщества.

К наиболее эффективным методам обеспечения высокого качества цифровой передачи в условиях высокого уровня шума в канале относятся уже весьма мощные в настоящее время корректирующие коды и алгоритмы их декодирования. Вместе с тем на практике не используются множество различных кодов из-за отсутствия алгоритмов их обработки с низкими вычислительными затратами. Таким образом, весьма важным становится вопрос о поиске эффективных, легко реализуемых, быстрых схем кодирования и декодирования, сложность реализации которых с ростом длины кода увеличивалась бы как можно медленнее.

В настоящее время поиск подобных альтернативных методов декодирования все дальше уходит в смежные с теорией кодирования области знаний, основные достижения которых можно использовать для построения эффективных схем кодирования и декодирования. Разработанные таким образом методы могут являться альтернативой существующим и зарекомендовавшим себя методам коррекции ошибок и наделять существующую систему новыми свойствами.

Декодирование блоковых кодов на основе нейросетевого классификатора. Оптимальный декодер, минимизирующий вероятность ошибки (когда все переданные последовательности равновероятны), известен как *декодер максимального правдоподобия* (MLD – maximum likelihood detector). Задачу MLD можно сформулировать следующим образом [1]: имея принятый вектор r из всех возможных слов d, принадлежащих коду, выбрать такое $d^{(j)}$, для которого

$$P\left(r \mid d^{(j)}\right) = \max_{d \in \text{коду}} P(r \mid d). \qquad (1)$$

Функция правдоподобия (решение задачи декодирования по MLD) задается или вычисляется, исходя из спецификации канала. В случае использования канала АБГШ, при котором шум влияет на каждый символ кода независимо от остальных символов, вычисление $P(r|d)$в (1) сводится к произведению условных вероятностей отдельных бит [1, 2]:

$$L_j = P\left(r \mid d\right) = \prod_{i=1}^{N} P\left(r_i \mid d_i^{(j)}\right) =$$

$$= \frac{1}{\delta\sqrt{2\pi}} \exp\left[-\sum_{i=1}^{N} \frac{\left(r_i - d_i^{(j)}\right)^2}{2\delta^2}\right]. \qquad (2)$$

Как показано в [3], квадрат Евклидова расстояния, $(r_i - d_i^{(j)})^2$, является метрикой для декодирования по максимуму правдоподобия. При использовании "жесткого" декодирования максимизация L_j эквивалентна максимизации:

$$L_j = P\left(r \mid d\right) = \sum_{i=1}^{N} r_i d_i^{(j)}. \qquad (3)$$

Как правило, увеличение помехоустойчивости системы передачи можно добиться лишь путем повышения вычислительной сложности реализации процедур кодирования и декодирования данных. Для решения задачи декодирования по максимуму правдоподобия по "жесткой" (3) и "мягкой" (2) схеме и уменьшения вычислительной сложности разрабатываемых декодеров могут применяться *нейросетевые классификаторы*, где (3) соответствует вычислению индуцированного локального поля соответствующих нейронов слоев нейронной сети, а (2) – использованию радиальных базисных элементов в составе сети.

В большинстве опубликованных работ по построению нейронного декодера MLD для блоковых кодов используется нейронная сеть Хэмминга [4, 5, 6]. Для простоты назовем подобную структуру *рекуррентным нейронным декодером 1-й модификации (РНД1)* [6].

В данной статье предложено упростить структуру РНД1 в сторону уменьшения вычислительной сложности реализации путем удаления конкурирующего слоя нейронов, предназначенного для нахождения ближайшей к принятому вектору разрешенной кодовой комбинации. Полученная таким образом структура названа РНД2 (рис. 1).

Удаление конкурирующего слоя стало возможным после введения в слой образцов РНД1 пороговых активационных функций с порогами

$$b_i = -(N - d_{\min} - 1). \qquad (4)$$

Пороги нейронов слоя образцов, определенные выражением (4), "подавляют" сигналы малоактивных нейронов, переводя их в отрицательные значения. При возникновении ошибок кратности $\le t$ в слое образцов РНД2 активизируется только один нейрон, имеющий максимальное значение, который через синаптические веса пороговых нейронов выходного слоя представляет информационную часть ближайшей к входному вектору разрешенной кодовой комбинации. Такая структура не требует обучения и полностью зависит от характеристик используемого кода. Это позволяет строить подобные декодеры для любых (не обязательно линейных) блоковых кодов. Далее описана структура и алгоритм работы такого декодера.

В качестве весовых коэффициентов синапсов первого слоя РНД2 $w_{ij}^{(1)}$ используются элементы разрешенных кодовых комбинаций c^j, записанных в биполярной форме: $w_{ij}^{(1)} = 2c_i^j - 1$, $1 \le j \le 2^K$, $1 \le i \le N$, где $w_{ij}^{(1)}$ – весовой коэффици-

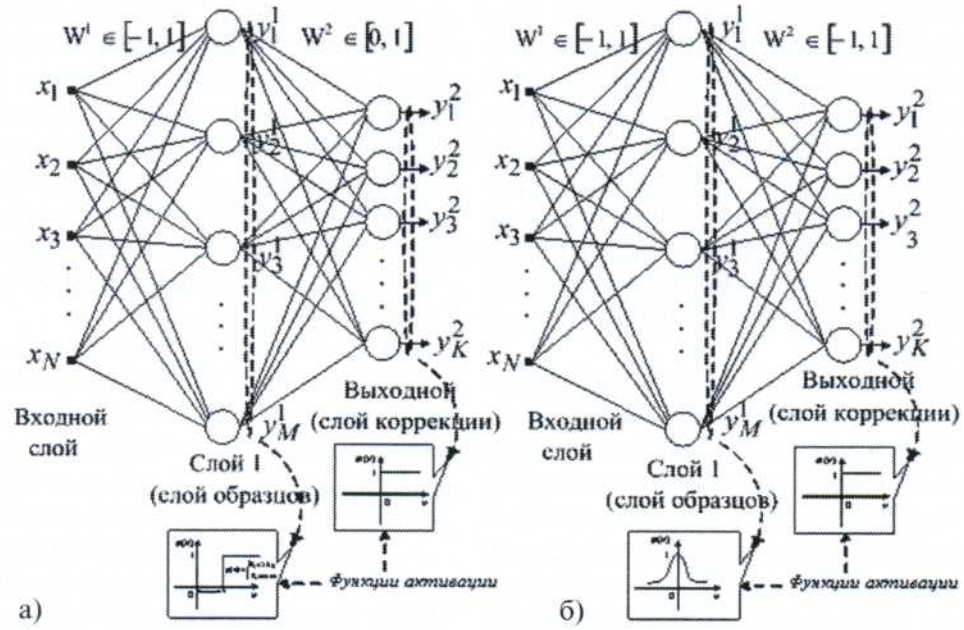


Рис.1. Структурная схема рекуррентного нейронного декодера блокового кода:
а) "жесткое" декодирование (РНД2); б) "мягкое" декодирование (РНД2м)

ент синапса i-го элемента входного вектора (слой 0) и j-го нейрона слоя образцов. Весовые коэффициенты синапсов выходного слоя эквивалентны информационным элементам разрешенных кодовых комбинаций, представленных в бинарной форме: $w_{ij}^{(2)} = c_i^j$, $1 \le i \le 2^K$, $1 \le j \le K$, где $w_{ij}^{(2)}$ – весовой коэффициент синапса i-го нейрона слоя образцов и j-го нейрона выходного слоя. Индуцированные локальные поля нейронов слоя образцов и выходного слоя равны:

$$v_i^1 = \sum_{j=1}^{N} w_{ji}^{(1)} x_j + b_i, \quad 1 \le i \le 2^K,$$

$$v_i^2 = \sum_{j=1}^{M} w_{ji}^{(2)} y_j^1, \quad 1 \le i \le K, \; M = 2^K.$$

Величины порогов нейронов слоя образцов b_i вычисляются в соответствии с выражением (4).

Выходные сигналы всех нейронов РНД2 определяются пороговой функцией активации j (индекс сверху указывает номер слоя):

$$y_i = \phi^1(v_i^1) = \phi^2(v_i^2) = \begin{cases} 1, & \text{если } v_i \ge 0 \\ 0, & \text{иначе} \end{cases}.$$

В результате работы на выходе РНД2 декодируется информационная часть разрешенной кодовой комбинации, связанной с искаженным входным вектором.

В случае использования канала АБГШ структура нейросетевого классификатора РНД2 может быть модифицирована. Для этого в качестве подсети нейронов слоя образцов в данной статье предложено использовать *вероятностную нейронную сеть* [7, 8], которая представляет собой параллельную реализацию известных статистических методов ядерной аппроксимации, оформленных в виде нейронной сети. Подобная модификация структуры РНД2 названа РНД2м (рис. 1,б).

Различие РНД2м от РНД2 заключается в том, что нейроны слоя образцов РНД2м представляют собой *радиальные базисные элементы*, активность которых определяется функцией Гаусса с нормальным законом распределения. В соответствии с этим выходные сигналы нейронов слоя образцов РНД2м можно получить в соответствии с выражением:

$$y_k^1 = \exp\left(-\frac{\sum_{i=1}^{N} \left(w_{ik}^{(1)} - x_i^k \right)^2}{2\delta^2} \right), \quad k = 1, \ldots, M, \quad (5)$$

где y_k^1 – выходное значение нейрона k слоя образцов, $w_{ik}^{(1)}$ – весовые коэффициенты синаптической связи, соединяющей вход i нейронной сети и нейрон k первого слоя, x_i^k – i-й элемент

входной кодовой комбинации, δ^2 – дисперсия, характеризующая ширину радиально-базисной функции.

Количество радиальных элементов РНД2м также определяется разрешенными кодовыми комбинациями, которые образуют классы.

Алгоритм работы РНД2м полностью соответствует алгоритму работы РНД2 с учетом того, что выходные значения нейронов слоя образцов y_k^1 вычисляются в соответствии с (5), а весовые коэффициенты синапсов выходного слоя эквивалентны информационным элементам разрешенных кодовых комбинаций, представленных в биполярной форме $w_{ij}^{(2)} = 2c_i^j - 1$.

Декодирование блоковых кодов на основе нейросетевой обучаемой машины. С целью уменьшения размера РНД2 можно использовать механизм обучения нейронных декодеров на примерах и тем самым адаптации своей структуры для решения конкретной задачи декодирования. В данном случае все нейроны сети являются нелинейными. Эта нелинейность достигается за счет использования в передаточных функциях всех нейронов сети антисимметричной функции активации гиперболического тангенса. Суть заключается в том, что каждый нейрон обучаемого нейронного декодера отвечает за создание гиперплоскости

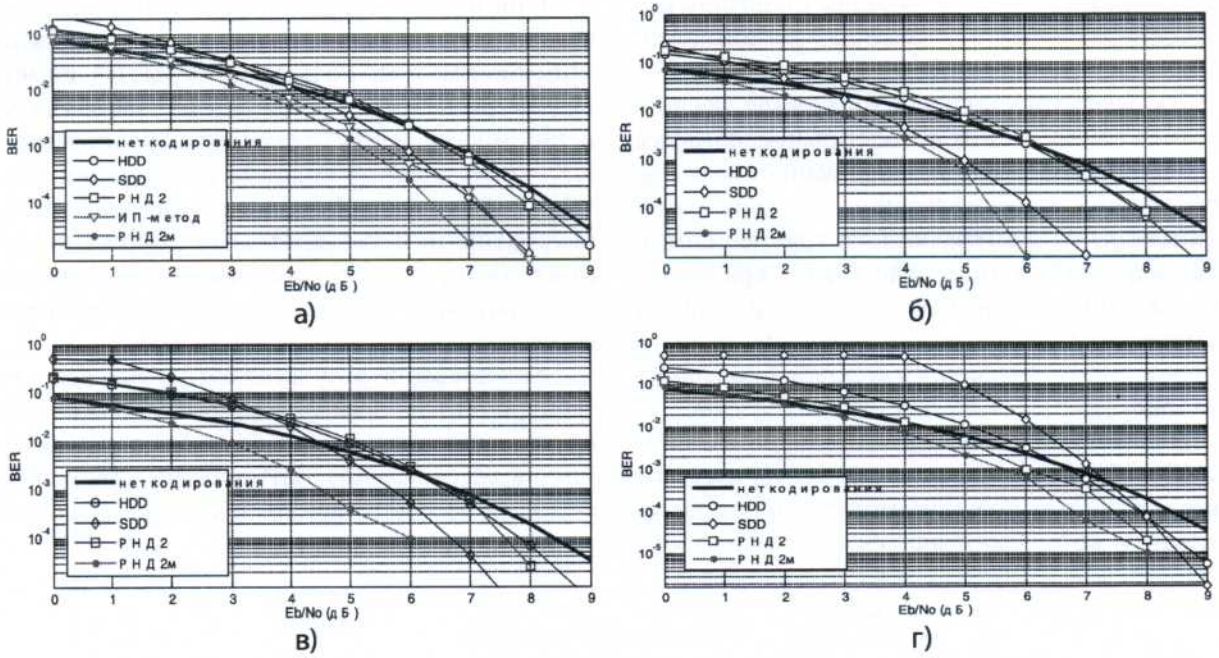


Рис. 2. Оценка помехоустойчивости РНД2/РНД2м при работе в канале АБГШ:
а) для кода БЧХ (7,4); б) для кода БЧХ (15,5); в) для кода БЧХ (15,7); г) для кода БЧХ (15,11)

Как показали результаты моделирования, помехоустойчивость РНД2м выше в среднем на ~1,5 дБ по сравнению с оптимальным "мягким" декодером (SDD – soft decision decoder) (рис. 2). Помехоустойчивость РНД2 сравнима с оптимальным "жестким" декодером (HDD – hard decision decoder). РНД2 способен исправлять ошибки, стирания и искажения без изменения структуры и логики работы. Более того, позиции стираний не имеют никакого значения, то есть с помощью РНД2 возможно декодирование без приведения кодового слова к двоичному виду.

(гиперповерхности) в пространстве решений. В процессе обучения с учителем комбинация гиперплоскостей, сформированных всеми нейронами сети, итеративно трансформируется для разделения примеров, преднамеренно взятых из разных классов, а также ранее не встречавшихся, с наименьшей средней ошибкой классификации (обобщения). Выбор данных обучения и их размещение относительно конкретных гиперплоскостей или гиперповерхностей становится очень важным для минимизации структуры разрабатываемых нейронных кодеров и декодеров. В связи с этим

размер скрытого слоя полностью определяет сложность реализации нейронных кодеров и декодеров.

В соответствии со статистической теорией восстановления зависимостей по эмпирическим данным [9, 10] (теорией статистического обучения В.Н. Вапника и А.Я. Червоненкиса) для минимизации ошибки обобщения размер обучающей выборки должен быть бесконечно большим. Увеличение размера обучающей выборки влечет за собой увеличение числа нейронов в скрытом слое разрабатываемых нейронных декодеров. Наилучшие результаты достигаются в случае, когда обучающие примеры располагаются с разных сторон границ гиперплоскостей, разделяющих пространство классов разрешенных кодовых комбинаций кода.

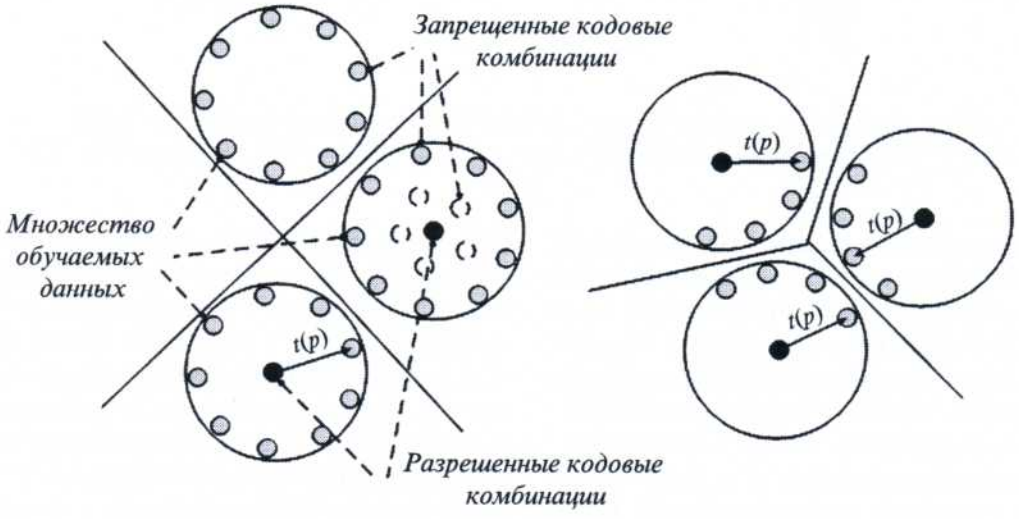
В качестве обучающего множества для построения нейросетевых декодеров предлагается использовать множество запрещенных кодовых комбинаций кода, отстоящих на расстоянии t (и/или p) от центра класса (разрешенной кодовой комбинации), где t (p) – кратность исправляемых кодом ошибок (стираний), и находящихся на границах сегментов гиперповерхностей, разделяющих кодовое пространство (рис. 3).

В качестве минимально необходимой обучающей выборки можно использовать множество запрещенных кодовых комбинаций соответствующего кода, находящихся на границах сегментов гиперплоскостей, разделяющих кодовое пространство (рис. 3б). При таком выборе обучающих данных достигается минимальная структурная и вычислительная сложность обученных нейронных декодеров, а также существенное уменьшение размера обучающей выборки (рис. 4). Для точного определения таких данных разработчику необходимо знать характеристики пространственного распределения кодовых слов соответствующего кода и весовую структуру кода, что является достаточно трудной задачей.

Подобный нелинейный декодер назван *нейронным декодером прямого распространения (НДПР)* в соответствии со структурой сети, используемой в его составе.

При использование предложенных обучающих данных были получены практические аппроксимации размеров скрытых слоев НДПР для некоторых кодов БЧХ (табл. 1), где в таблице введены следующие обозначения: КК – разрешенные кодовые комбинации; Е1-Е3 – кодовые комбинации с однократными, двукратными и трехкратными ошибками соответственно; S1–S6 – кодовые комбинации с однократными – шестикратными стираниями соответственно.

Как показали результаты многочисленных экспериментов, использование предложенных обучающих данных (рис. 3а) ведет к уменьшению скрытого слоя НДПР линейных блоковых кодов, а соответственно, и вычислительной сложности реализации.



а) б)

Рис. 3. Выбор обучающих данных для нейронного декодера блокового кода:
а) необходимое множество; б) минимально необходимое множество

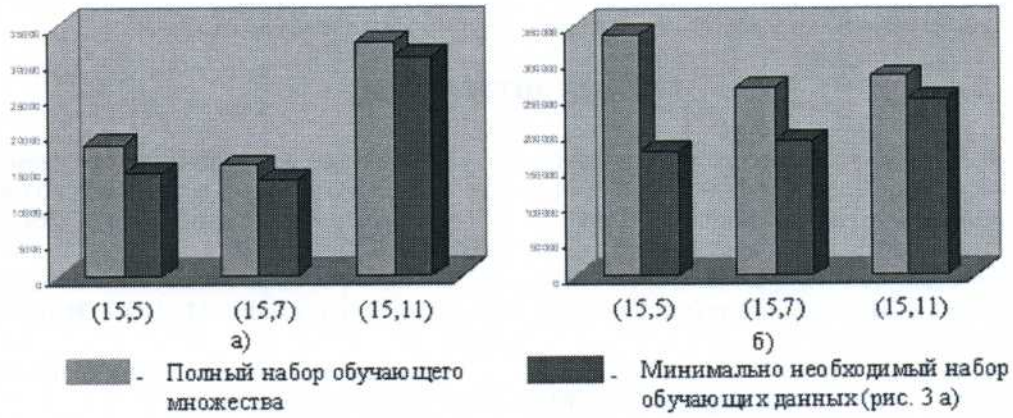


Рис. 4. Изменение размера обучающего множества некоторых (N, K) кодов БЧХ с увеличением K: а) при исправлении ошибок; б) при исправлении ошибок и стираний

Таблица 1

Размер скрытого слоя НДПР некоторых кодов БЧХ

№ п/п	Код	Размер скрытого слоя, L	Обучающее множество		Размер обучающего множества
			ошибки	стирания	
Исправление ошибок					
1	(7,4)	7	E1	–	112
		9	KK,E1	–	128
2	(15,5)	38	E3	–	14 560
		40	KK,E1,E2,E3	–	18 432
3	(15,7)	37	E2	–	13 440
		38	KK,E1, E2	–	15 488
Исправление ошибок и стираний					
1	(7,4)	9	E1	S2	448
		10	KK, E1	S1, S2	576
2	(15,5)	58	E3	S6	174 720
		69	KK, E1, E2, E3	S1, S2, S3, S4, S5, S6	336 768
3	(15,7)	39	E2	S4	188 160
		40	KK, E1, E2	S1, S2, S3, S4	263 680

Заключение.

Предложенные модифицированные структуры нейронных декодеров могут являться основой для построения быстрых схем кодирования. При их аппаратной реализации на специализированных нейронных процессорах декодирование любого блокового кода может происходить в реальном масштабе времени. Более того, использование нейронных сетей в качестве составных частей декодеров наделяет систему кодирования важными дополнительными свойствами. Данные свойства, безусловно, являются преимуществом нейросетевых систем кодирования по сравнению с классическими методами защиты от ошибок. К таким свойствам, например, относятся *простота реализации, высокая скорость функционирования при постоянной сложности реализации, большая отказоустойчивость, эффективная реализуемость, единообразие анализа и проектирования.*

Нейронные сети, входящие в состав нейронных декодеров, являются универсальным механизмом обработки информации. Нейроны в той или иной форме являются стандартными составными частями любого нейронного кодера и декодера. Эта общность позволяет использовать одни и те же подходы и алгоритмы для построения нейросетевых кодеров и декодеров для кодов различных длин и классов.

Таким образом, выполненные исследования показали широкие возможности для построения быстрых кодирующих и декодирующих

устройств. Разработанные модели являются открытыми системами и могут быть дополнены различными компонентами, расширяющими их функциональные возможности.

СПИСОК ЛИТЕРАТУРЫ

1. **Скляр Б.** Цифровая связь. Теоретические основы и практическое применение. М.: Вильямс, 2004.

2. **Вернер М.** Основы кодирования. М.: Техносфера, 2004.

3. **Морелос-Сарагоса Р.** Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. М.: Техносфера, 2005.

4. **Zeng G., Hush D., Ahmed N.** An application of neural net in decoding error-correcting codes // IEEE International Symposium on Circuits and Systems. – 1989.

5. **Stefano A.D, Mirabella 0., Cataldo G.D., Palumbo G.** On the use of neural networks for hamming coding // IEEE International Symposium on Circuits and Systems. – 1991. – vol. 3. – Pp. 1601-1604.

6. **Березкин А.А.** Рекуррентный нейронный декодер двоичных блоковых кодов и его модификации // Труды учебных заведений связи / СПбГУТ. СПб, 2007. №176. С. 135 - 146.

7. **Хайкин С.** Нейронные сети: полный курс, 2-е изд., испр. / Пер. с анл. – М.: ООО "И.Д. Вильямс", 2006.

8. **Каллан Р.** Основные концепции нейронных сетей. – М.: Вильямс, 2003.

9. **Вапник В.Н.** Восстановление зависимостей по эмпирическим данным.– М.: Наука, 1979.

10. **Вапник В.Н., Червоненкис А.Я.** О равномерной сходимости частот появления событий к их вероятностям // ДАН СССР.– Т. 181, № 4.– 1968.

Кудряшов Д.А., Цикин И.А.

Вероятностно-временные характеристики двунаправленной радиосистемы передачи тревожных извещений

Введение

В связи с широким распространением двунаправленных радиосистем передачи тревожных извещений (РСПТИ) актуальной является задача увеличения числа обслуживаемых объектовых станций (ОС) при сохранении достоверности передачи не хуже заданной. Как было показано в [1], данная задача может быть решена путем параметрической оптимизации алгоритма функционирования. При этом для проведения оптимизации необходимым является получение вероятностно-временных характеристик, связывающих вероятность недоставки сообщения с требуемым допустимым временем его доставки, числом объектовых станций и параметрами алгоритма функционирования.

Постановка задачи

Рассмотрим двунаправленную РСПТИ, состоящую из одной центральной станции (ЦС) и N объектовых станций. Обмен сообщениями между ЦС и любой из ОС происходит следующим образом.

1. Система функционирует в асинхронном режиме, т.е. как только на ОС возникает какое-либо событие или подходит время автотеста, объектовая станция отправляет центральной извещение на общей для всех ОС частоте $f_{ОС}$. Так как все ОС отправляют извещения независимо друг от друга, существует отличная от нуля вероятность перекрытия во времени сигналов извещений (коллизии) на входе ЦС.

2. Получив извещение от ОС, центральная станция оправляет данной ОС квитанцию на частоте $f_{ЦС}$, отличной от $f_{ОС}$. Таким образом, коллизии между квитанциями на входе любой из ОС исключаются.

3. Отправив извещение, ОС в течение определенного времени $t_{ожид}$ ожидает квитанцию от ЦС. В случае неполучения такой квитанции ОС повторно отправляет извещение через случайный интервал времени $t_{повт}$ и вновь ожидает квитанцию. Извещение может повторяться многократно при условии, что от момента времени первой попытки передачи до момента времени начала очередной попытки прошло не больше, чем $T_{дост}$. Если за время $T_{дост}$ извещение не было доставлено на ЦС, попытки передачи его на данной частоте прекращаются. Данное событие считается недоставкой извещения. Возможная ситуация изображена на рис 1.

При этом

$$t_{повт} = 2 \cdot t_c + t_0 + \Delta t,$$

где t_c – длительность извещения, t_0 – средний интервал повторения, а Δt – случайная величина, равномерно распределенная на интервале $(-t_\Delta, t_\Delta)$.

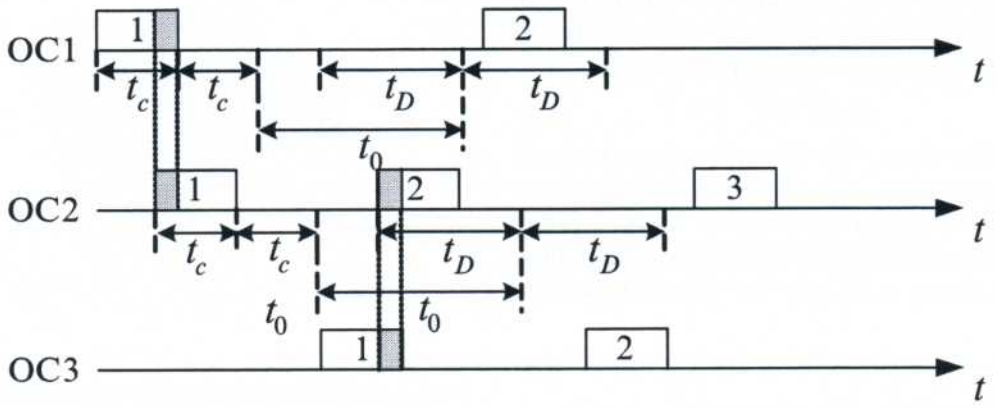


Рис. 1. Передача извещения в условиях коллизий

Выберем некоторый интервал наблюдения T. Будем считать, что в течение данного интервала времени каждая из ОС с вероятностью 1 предпринимает первую попытку отправки извещения. Такая ситуация имеет место, например, при передаче служебных сообщений в режиме автотестирования. Для удобства оценки будем считать, что длительность всех извещений одинакова и равна t_c.

Оценим далее величину вероятности недоставки извещений.

Вероятность недоставки извещений

Рассмотрим прежде всего простейшую ситуацию, когда каждая ОС имеет ровно одну попытку доставить сигнал на ЦС. Рассмотрим сигнал от i-й ОС, где $i = 1...N$ (далее – сигнал "i").

Вероятность того, что данный сигнал окажется в коллизии с сигналом "k" ($k = 1...N$, $k \neq j$), очевидно, равна:

$$p_0 = \frac{2 \cdot t_c}{T}. \qquad (1)$$

Вероятность того, что данный сигнал окажется в коллизии с хотя бы одним сигналом от любой другой ОС в условиях $p_0 << 1$ может быть представлена:

$$P = N \cdot p_0.$$

Вернемся к рассмотрению реального случая, когда число попыток передачи сигнала от любой из ОС может превышать 1 и определяется временем $T_{дост}$.

Введем в рассмотрение величину K – общее число сигналов, передаваемых от объектовых станций на ЦС за время T. Очевидно, что $K \geq N$.

Рассмотрим некий j-й сигнал из переданных K ($j = 1..K$). Вероятность того, что данный сигнал окажется в коллизии хотя бы с одним из $K - 1$ других сигналов примерно равна:

$$P = K \cdot p_0. \qquad (2)$$

Предположим, что вероятность коллизии трех и более сигналов много меньше вероятности коллизии двух сигналов. Тогда каждая коллизия увеличивает число сигналов в общем канале доступа к ЦС на 2 сигнала. Следовательно,

$$K = N + N \cdot 2 \cdot P = N \cdot (1 + 2 \cdot P). \qquad (3)$$

Объединяя (1)...(3), получаем:

$$P = \frac{N \cdot \frac{2 \cdot t_c}{T}}{1 - N \cdot \frac{4 \cdot t_c}{T}}. \qquad (4)$$

Введем в рассмотрение величину M – среднее число сигналов, отправленных одной ОС на ЦС при условии, что сообщение данной ОС не было доставлено за время $T_{дост}$.

Для алгоритма функционирования с постоянным средним интервалом повторения имеем:

$$M = \frac{T_{дост}}{t_0 + 2 \cdot t_c} + 1. \qquad (5)$$

Если пренебречь вероятностью Q повторной коллизии сигналов, то вероятность недоставки извещения одной станции можно было бы вычислить как

$$P_M = P^M.$$

Учет вероятности Q приводит к иному соотношению:

$$P_M = P \cdot (P + Q)^{M-1}, \qquad (6)$$

где

$$Q = \frac{t_c}{t_\Delta - t_c}. \qquad (7)$$

Вероятность недоставки извещения от хотя бы одной из N объектовых станций может быть оценена:

$$P_{нед} = 1 - (1 - P_M)^N. \qquad (8)$$

Ниже (рис. 2....рис. 4.) в качестве примера представлены зависимости вероятности недоставки от параметров алгоритма функционирования. Для удобства далее будем задавать величину $t_\Delta = t_0 \cdot k_\Delta$, где $0 < k_\Delta < 1$.

На рис 2. представлена зависимость вероятности $P_{нед}$ от величины среднего интервала повторения и величины.

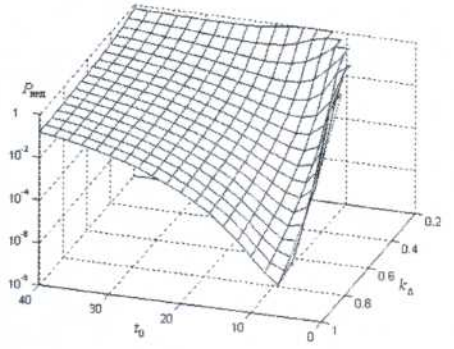


Рис. 2. Зависимости вероятности недоставки извещений от среднего интервала повторения и параметра k_Δ при $N = 500$, $T = 30$ мин., $T_{дост} = 1$ мин.

Отдельно на рис. 3 приведена зависимость вероятности $P_{нед}$ от величины среднего интервала повторения при фиксированном значении величины k_{Δ}.

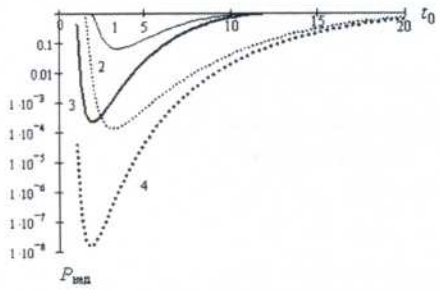


Рис. 3. Зависимости вероятности недоставки извещений от среднего периода повторений при N = 750 (кривые *1,2*), 500 (*3,4*); k_{Δ} = 0.25 (*1,3*), 0.75(*2,4*); T = 30 мин., $T_{дост}$ = 1 мин.

На рис 4. приведена зависимость вероятности недоставки от величины k_{Δ}.

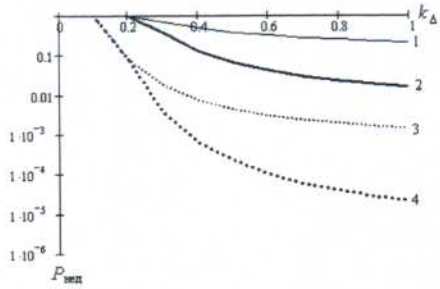


Рис. 4. Зависимости вероятности недоставки извещений от параметра k_{Δ}: N = 750 (кривые *1,2*), 450 (*3,4*); t_0 = 7 с. (*1,3*), 5 с. (*2,4*); T = 30 мин., $T_{дост}$ = 1 мин.

Как видно из полученных зависимостей, для каждых значений N и T существуют такие значения параметров t_0 и k_{Δ}, при которых достигается минимум вероятности недоставки. Таким образом, полученные аналитические зависимости позволяют решить проблему параметрической оптимизации алгоритма функционирования системы по различным критериям, таким как, например, минимум вероятности недоставки или максимум количества ОС при заданных вероятностно-временных характеристиках.

Однако точность полученных выше аналитических оценок может оказаться недостаточной в области больших значений вероятности недоставки извещений, так как ограничение на время доставки учитывается лишь в выражении (5) и далее, причем допустимое количество повторений попыток передачи определялось, исходя из среднего значения интервала повторений. По существу в этих условиях полученные результаты соответствуют несколько завышенным значениям вероятности недоставки извещения, т.е. "худшему" случаю.

Тем не менее, с целью получения точных результатов в широкой области значений параметров была разработана имитационная модель работы системы.

Имитационная модель двунаправленной РСПТИ

Рссмотрим имитационную модель двунаправленной РСПТИ. Зададим случайным образом моменты времени первой попытки передачи извещения для каждой из ОС. Расставим эти времена в порядке возрастания. Будем последовательно перебирать элементы данного массива, проверяя, не перекрывается ли данное извещение с предыдущим. В случае перекрытия добавляем в массив новые моменты повторной передачи, но так, чтобы массив остался ранжированным. Если разница между новым моментом времени и моментом времени первой попытки будет превышать величину $T_{дост}$, данное событие будем считать недоставкой извещения хотя бы от одной из N станций. Выполняя данную процедуру многократно, можно получить оценку вероятности недоставки извещения от любой из ОС, поделив число недоставок на общее число проведенных экспериментов. Алгоритм модели представлен на рис 5:

1. сформируем вектор начальных моментов выход в эфир для каждой из N станций t_0 = $= (t_{0,1} \ldots t_{0,j} \ldots t_{0,N})$, где $t_{0,j} = rand(0 \ldots T$;
2. проведем сортировку вектора t_0так, что бы $t_{0,j-1} < t_{0,j} < t_{0,j+1}$, где j $\in (0 \ldots N)$;
3. введем вектор $t = t_0$;
4. введем K – общее число пакетов в эфире, $K = N$;
5. введем флаг наличия коллизии a = 0;
6. j =2;
7. если $t_j - t_{j-1} > t_c$, то a = 0, переходим к шагу 13 ;
8. добавляем в вектор $\bar{t}$ элемент $t_{K+1} = t_j +$ $+ 2 \cdot t_c + rand(-t_{\Delta} \ldots t_{\Delta})$, в вектор $\overline{t_0}$ элемент $t_{0,K+1} = t_{0,j}$, $K = K+1$;
9. если $t_{K+1} = t_{0,j} > T_{дост}$, фиксируем недоставку и выходим из процедуры;
10. проведем сортировку векторов t и t_0 так, чтобы $t_{j-1} < t_j < t_{j+1}$, где $j \in (0 \ldots K)$;

11. если $a = 0$, проводим шаги 8...10 для элемента $j - 1$;

12. фиксируем коллизию $a = 1$;

13. $j = j+1$, если $j < K$, переходим к шагу 7;

14. выходим из процедуры.

Относительную частоту v возникновения недоставки извещения хотя бы от одной и N объектовых станций можно найти следующим образом:

1. зададимся величиной E – числом повторений эксперимента;,

2. счетчик повторений эксперимента $I = 0$, счетчик недоставок $c = 0$;

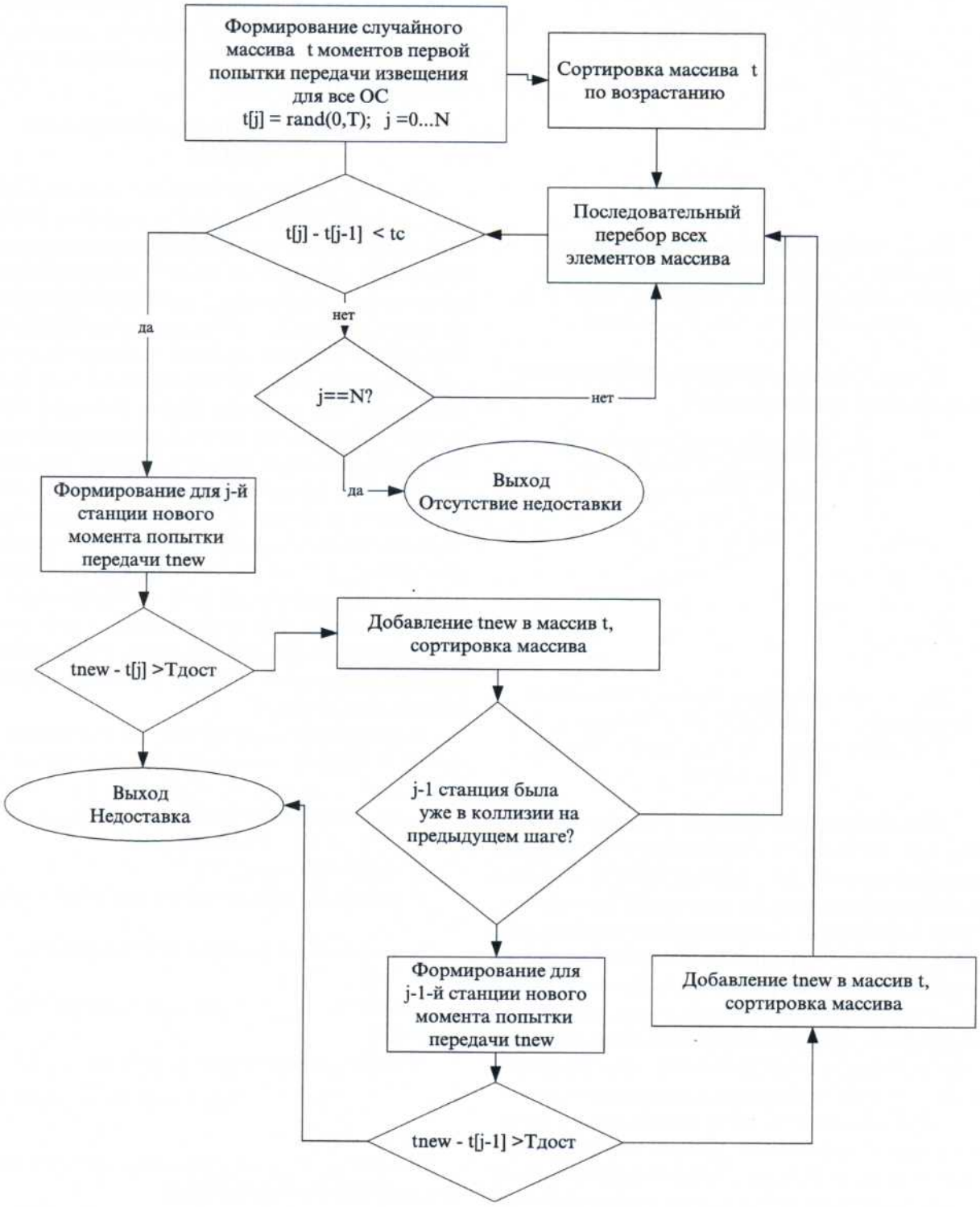


Рис. 5. Алгоритм модели

3. выполняем процедуру $П_1$, если фиксируется событие "недоставка" $c = c + 1$;

4. $I = I + 1$, если $I \neq E$, переходим к шагу 3;

5. если $c < d$ (где d – минимальное число событий "недоставка", необходимых для оценки вероятности недоставки с заданным доверительным интервалом), переходим к шагу 2 при $E = E \cdot 10$;

6. $\nu = \frac{c}{E}$, выход.

Оценка вероятности недоставки $P_{нед}$ может быть получена по значению ν с доверительным интервалом 2δ при заданной доверительной вероятности $1 - \beta$ [2].

В таблице приведены величины доверительных интервалов для различных величин ν и p_0 при доверительной вероятности $1 - \beta = 0{,}99$.

Таблица 4.1

Значения доверительного интервала при заданных значениях ν и n

$n = 10^7$, $\nu = 10^{-6}$	$2\delta = 1{,}6 \cdot 10^{-6}$
$n = 10^7$, $\nu = 10^{-5}$	$2\delta = 2{,}5 \cdot 10^{-6}$
$n = 10^6$, $\nu = 10^{-4}$	$2\delta = 2{,}5 \cdot 10^{-5}$
$n = 10^5$, $\nu = 10^{-3}$	$2\delta = 2{,}5 \cdot 10^{-4}$

Величину d можно задать как $d = \nu \cdot n$.

С целью оценки точности приведенных выше аналитических выражений сравним результаты аналитического расчета с результатами имитационного моделирования, полученными на основе разработанной модели.

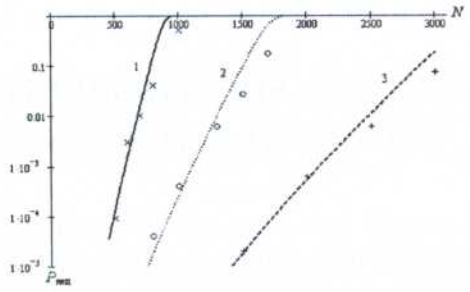


Рис. 6. Зависимости вероятности недоставки извещений от числа объектовых станций N: T = 30 мин. (кривая *1*), 60 мин. (*2*), 120 мин. (*3*); t_0 = 5 с., k_Δ = 0,1, $T_{дост}$ = 1 мин.

На рис. 6 приведены зависимости вероятности недоставки от числа объектовых станций, причем сплошными линиями изображены аналитические зависимости, а точками – результаты моделирования. Как видно из рисунка, в представляющей практический интерес области значений вероятности недоставки $P_{нед} = 10^{-5} \ldots 10^{-2}$ наблюдается удовлетворительное совпадение расчетных значений с результатами моделирования. В то же время в области значений вероятности недоставки $P_{нед} > 10^{-2}$ результаты моделирования, как и предполагалось выше, несколько расходятся с расчетными значениями.

Таким образом, полученные в работе результаты позволяют получить вероятностно-временные характеристики в любой интересующей области значений параметров алгоритма функционирования РСПТИ.

СПИСОК ЛИТЕРАТУРЫ

1. **Варгаузин В.А., Кудряшов Д.А., Цикин И.А., Червинский М.В.** Охранные радиосистемы со случайным множественным доступом к общему частотному каналу// Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. – 2008. – № 2. – С. 39–45.

2. **Вентцель Е.С., Овчаров Л.А.** Теория случайных процессов и ее инженерные приложения. – М.: Издательский центр "Академия", 2003. – 432 с.

Сидоров Ю.Е., Наспер М.Л.

Оценка эффективности обнаружителя радиосигналов в каналах с замираниями при разнесенном приеме

Основной задачей радиотехнических и телекоммуникационных систем является обнаружение и выделение сигналов, несущих информацию, на фоне помех. Как сигналы, так и помехи представляют собой случайные процессы, поэтому решение задач обнаружения и выделения сигналов опирается на теорию статистических решений [1].

Воздействие помехи на сигнал обычно бывает необратимым и вызывает уменьшение вероятности обнаружения сигнала (правильного приема). В реальных радиоканалах действуют разнообразные помехи, порождаемые внешними источниками, а также случайные искажения. Прежде чем сигнал появится на входе приемника, на него накладывается целый ряд искажений, вносимых средой распространения сигнала. К ним относятся затухание, замирания, доплеровский сдвиг несущего колебания, нелинейные искажения, межстанционные помехи и аддитивный шум [2].

Применение классических методов оптимального синтеза решающих правил обнаружения и выделения сигналов на фоне помех требует обширных априорных сведений о вероятностных моделях и статистических характеристиках сигналов и шумов [3,4]. Эти методы характеризуют потенциальную эффективность систем обработки радиотехнической информации. В реальных условиях функционирования радиосистем эти сведения часто отсутствуют. Поэтому большую актуальность приобретают методы преодоления априорной неопределенности относительно сигнально-помеховых ситуаций. Существует две разновидности априорной неопределенности: параметрическая и непараметрическая. При параметрической неопределенности функциональный вид закона распределения входных данных известен, но неизвестны параметры этого закона. При непараметрической априорной неопределенности неизвестен сам функциональный вид закона распределения наблюдаемых процессов, которые нельзя описать конечным числом параметров. При этом класс допустимых распределений может включать все соответствующим образом нормированные функции, которые нельзя поставить в однозначное соответствие с какой-либо областью евклидова пространства.

Целью данной статьи является представление результатов исследования обнаружителя сигналов в канале радиосвязи с замираниями при разнесенном приеме, который описан в монографии [5], и оценка его характеристик методом статистического имитационного моделирования. На вход обнаружителя поступает сигнал с релеевскими замираниями на фоне аддитивного гауссова белого шума. Отметим, что разнесенный прием часто используется для борьбы с общими замираниями сигнала связи.

Рассмотрим задачу обнаружения сигналов при разнесенном многоканальном приеме, полагая, что сигналы в каналах приема подвержены общим статистически независимым замираниям по закону Релея и шумы в каналах являются независимыми гауссовскими процессами с постоянной спектральной плотностью в пределах полосы пропускания линейного тракта приемника (ЛТП). Амплитудно-частотная характеристика ЛТП считается прямоугольной с полосой пропускания Δf. При данных предпосылках наблюдаемый процесс $x_i(t), i=\overline{1,N}$, в каждом из N неидентичных каналов приема характеризуется достаточными статистиками:

$$T_i(\dot{x}_i)=\int_0^T|\dot{x}_i(t)|^2\,dt;\ V_i(\dot{x}_i)=\left|\int_0^T\dot{x}_i(t)s_i^*(t)dt\right|^2,$$

где $\dot{x}_i(t)$ — комплексная огибающая процесса $x_i(t)$; $\dot{s}_i(t)$ — комплексная огибающая сигнала в i-ом канале. Эти статистики имеют плотность вероятности:

$$W(\vec{V};\vec{T}\,|\,\vec{\gamma};\vec{\pi})=$$

$$k\prod_{i=1}^{N}\frac{\pi_i^{\nu+1}}{1+\gamma_i}h(V_i;T_i)\times\prod_{i=1}^{N}\exp(-\pi_iT_i+\frac{\gamma_i\pi_iV_i}{1+\gamma_i}),$$

$$\vec{V}=(V_1,.....,V_N);\vec{T}=(T_1,...,T_N);$$
$$\vec{\gamma}=(\gamma_1,...,\gamma_n);\vec{\pi}=(\pi_1,....,\pi_N)$$

где γ_i — среднее отношение сигнал/шум в i-ом канале; $\pi_i = 1/(4N_{0i})$; N_{0i} — спектральная плотность шума в i-м канале; k — нормирующий множитель; $\nu = |\Delta fT|$ — целая часть полосы пропускания ЛТП на время наблюдения; функции $h(V_i; T_i)$ при любом $\delta > 0$ удовлетворяют равенству $h(\delta V_i; \delta T_i) = \delta^{\nu-1} h(V_i; T_i)$. Параметры γ_i и π_i полагаются априорно неопределенными и независящими от $i = \overline{1,N}$, если каналы приема идентичны по коэффициенту передачи и уровню шума, и различными при разных i в случае неидентичности каналов.

При идентичных каналах вычисляются достаточные статистики: $T = \sum_{i=1}^{N} T_i$, $V = \sum_{i=1}^{N} V_i$.

Семейство распределений этих статистик симметрично относительно группы $G = \{g : T \to \to \delta T;\ V \to \delta V;\ \delta \in (0; \infty)\}$ масштабных преобразований, которая представляет дружные изменения коэффициентов передачи каналов и уровней шумов в них. Поэтому применим принцип инвариантности [6], который приводит к построению РНМ инвариантного правила с решающей функцией [5]:

$$\varphi(V;T) = \begin{cases} 1 \text{ при } V \geq C(\alpha)T; \\ 0 \text{ при } V \leq C(\alpha)T. \end{cases} \qquad (1)$$

Для оценки эффективности правила (1) целесообразно использовать метод статистического имитационного моделирования (метод Монте-Карло), который уже успешно применялся для моделирования радиотехнических систем [7, 8].

Статистическое моделирование состоит из следующих частей.

1. Моделирование сигналов.
2. Моделирование шума.
3. Моделирование замираний.
4. Моделирование обнаружителя.

Рассмотрим компоненты моделирования подробнее.

1. Обнаруживаемый сигнал представляет собой радиоимпульс с прямоугольной огибающей. База сигнала $\nu = |\Delta fT| = 2$.

2. Нормальное распределение, которому подчиняется шум, моделируется соотношением [9]:

$$y_i = \sigma \sqrt{2Ln(\frac{1}{x_i})} * \cos(2\pi x_{i-1}) + a,$$

где x_i и x_{i-1} – случайные равномерно распределенные на интервале [0, 1] величины.

3. Замирания сигнала в каждом канале обнаружителя независимы и подчиняются закону Релея.

Сигнал, приходящий на каждый канал обнаружителя подвергнут независимым релеевским замираниям. Эффект замираний будет реализован случайным изменением коэффициента передачи среды k для каждого канала по закону Релея [9]:

$$W(k) = \frac{k}{\sigma^2} \exp[-\frac{k^2}{2\sigma^2}].$$

Математическое ожидание $m = \sqrt{\frac{\pi}{2}}\sigma \approx 1{,}25\sigma$.

Выберем σ такой, что $m = 1$. Получим $\sigma = 0.8$

Таким образом, распределение коэффициентов передачи для каждого из каналов будет определяться следующим соотношением [9]:

$$W(k) = \frac{k}{0{,}16} \exp[-\frac{k^2}{0{,}32}].$$

4. Моделирование решающего правила (1) с помощью метода Монте-Карло в среде C++ Builder производилось для вероятностей ложной тревоги $\alpha_{ЛТ} = 10^{-2}, 10^{-3}, 5 \times 10^{-4}$, количество каналов N выбрано равным 2, 4, 8. Количество испытаний для каждой точки $m = 30000$, что соответствует максимальному относительному отклонению в 10% при доверительной вероятности $P = 0{,}9$.

В результате моделирования были получены зависимости вероятности правильного обнаружения D от соотношения сигнал/шум q при различных вероятностях ложной тревоги α и количества каналов приема N (рис. 1). Так как сигнал в каждом из каналов подвержен независимым релеевским замираниям, то мощность сигнала в каждом из каналов различна, поэтому под отношением сигнал/шум q понимается усредненное значение отношения мощности сигнала к мощности шума по всем каналам приема.

На представленных выше кривых видно, что при уменьшении уровня ложной тревоги падает вероятность правильного обнаружения. При переходе от $\alpha_{ЛТ} = 10^{-2}$ к $\alpha_{ЛТ} = 5 \times 10^{-4}$ для обеспечения того же значения вероятности правильного обнаружения D нужно увеличить соотношение сигнал/шум на 2,2 дБ. Также заметим, что при увеличении числа каналов приема N увеличивается и вероятность правильного

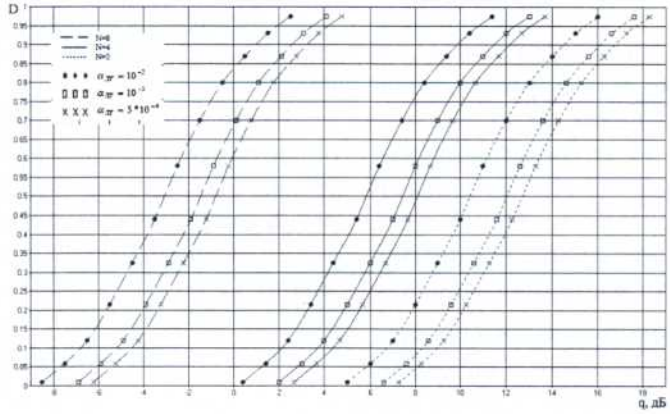


Рис. 1. Характеристики обнаружителя при разнесенном приеме

обнаружения. При переходе от $N = 2$ к $N = 4$ энергетический выигрыш составляет 8,9 дБ, а при переходе от $N = 4$ к $N = 8$ – 4,6 дБ. Отсюда можно сделать вывод: при дальнейшем увеличении N энергетический выигрыш системы будет увеличиваться, но не беспредельно, скорость увеличения его будет падать по мере возрастания N. Предположительно, зависимости будут стремиться к некоему пределу, который определяет характеристики при обнаружении сигнала без замираний.

Заключение

В данной статье рассмотрен обнаружитель сигналов в каналах связи с постоянным уровнем ложной тревоги, предназначенный для использования в условиях априорно неизвестной помеховой обстановки.

Методом статистического имитационного моделирования сделан расчет и проведен анализ характеристик обнаружителя при различных уровнях ложных тревог и каналов приема при подаче на вход радиоимпульса, амплитуда которого флюктуирует по закону Релея.

Анализ характеристик обнаружения позволяет сделать вывод о целесообразности использования данного обнаружителя в системах передачи информации в условиях априорной неопределенности.

СПИСОК ЛИТЕРАТУРЫ.

1. **Вальд А.** Основные идеи общей теории статистических решений // Последовательный анализ. М., Физматгиз, 1960 С. 308–325.

2. **Садомовский А. С.** Приемо-передающие радиоустройства и системы связи // Учебное пособие. Ульяновск :УлГТУ, 2007. 243 с.

3. **Гуткин Л. С.** Теория оптимальных методов приема при флуктуационных помехах. М.: Сов. Радио 1972. 447 с.

4. **Цикин И. А.** Оптимальная обработка сигналов в радиотехнических системах/ Учебное пособие. Л.: ЛПИ им. М.И. Калинина, 1986. 77с.

5. Теория обнаружения сигналов /П.С. Акимов, П.А. Бакут, В.А. Богданович и др./Под ред. П.А. Бакута. М.: Радио и связь, 1984. 440с.

6. **Леман Э.** Проверка статистических гипотез. М.: Наука 1964. 498 с.

7. **Сидоров Ю. Е.** Статистический синтез автоматизированных решающих систем при априорной неопределенности. М.: Воениздат, 1993. 227 с.

8. **Сидоров Ю. Е., Лаврентьев Н.В.** Оптимальный обнаружитель радиосигналов: решающее правило статистическое имитационное моделирование. Труды СПБГТУ, 2008, №507, с. 118–124.

9. **Быков В.В.** Цифровое моделирование в статистической радиотехнике. М.: Советское радио, 1971. 326 с.

Былин А.Р., Былина М.С., Глаголев С.Ф.

Пути повышения разрешающей способности импульсных измерений

Для локализации неоднородностей и повреждений широко используют импульсный метод измерения [1, 2], основанный на зондировании кабельных цепей короткими импульсами. Импульсные приборы, реализующие этот метод измерения, называют рефлектометрами во временной области.

При исследовании кабельных цепей импульсным методом обычно наблюдается много неоднородностей, импульсы, отраженные от которых, могут накладываться друг на друга, затрудняя локализацию отдельных неоднородностей. В данной работе рассматриваются два альтернативных пути повышения разрешающей способности импульсного метода: использование биимпульсного зондирующего сигнала и специальной обработки зарегистрированной рефлектограммы.

Различают стыковые, концевые и внутренние (в пределах строительной длины) неоднородности. По характеру неоднородности можно разделить на перепады волнового сопротивления, резистивные, емкостные, индуктивные и комплексные [1, 2], которые могут включаться в линию параллельно и последовательно. На практике чаще всего встречаются перепады волнового сопротивления и резистивные неоднородности. Их особенностью является то, что их импульсная характеристика по обратному потоку соответствует по форме импульсной характеристике линии $g(t)$ с длиной, равной удвоенному расстоянию до неоднородности $2l_n$:

$$g(t)=\frac{2l_n\sqrt{\tau_0}}{\sqrt{\pi}\cdot(t-t_3)^{\frac{3}{2}}}\cdot\exp\left(-\frac{4\tau_0 l_n^2}{t-t_3}\right),\ \text{при } t\geq t_3,\ (1)$$

где $t_3 = 2\tau_3 \cdot l_n$ – время задержки выходного импульса относительно зондирующего (с), τ_0 – удельная конструктивная постоянная кабельной цепи (с/км^2). Значения параметров τ_0 и τ_3 для различных кабельных цепей приведены в [3].

Для расчета формы отраженного импульса на входе рефлектометра $u_2(t)$ без учета шумов и помех можно использовать выражение:

$$u_2(t)=r\cdot\int_0^t u_1(t-\tau)g(\tau-t_3)d\tau,\qquad (2)$$

где $u_1(t)$ – форма зондирующего импульса, r – коэффициент отражения от неоднородности.

Известно [1, 2], что искажения отраженных от неоднородностей импульсов возрастают с увеличением расстояния до неоднородностей пропорционально квадрату этого расстояния. Отраженные импульсы расширяются, а их амплитуда уменьшается. Для уменьшения длительности отраженных импульсов можно использовать биимпульсный зондирующий сигнал или биимпульс (рис. 1), который представляет собой два прямоугольных импульса противоположной полярности с одинаковыми площадями $S_{im}/2$ и различными в общем случае длительностями t_+ и t_-, сумма которых равна t_{im} [4]. Для биимпульса справедливо:

$$u_1(t)=\begin{cases}\dfrac{0{,}5S_{im}}{t_+}, & 0\leq t<t_+\\ \dfrac{0{,}5S_{im}}{t_-}, & t_+\leq t<t_{im}\\ 0, & t<0, t>t_{im}\end{cases}.\qquad (3)$$

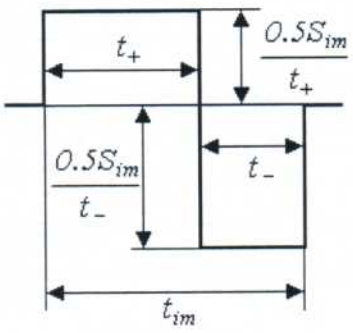


Рис. 1. Биимпульс

Особенностью такого сигнала является отсутствие у него постоянной составляющей.

Были проведены расчеты формы сигналов, отраженных от обрывов, для кабельных цепей ТПП-0.4 различной длины при прямоугольном и биимпульсном зондирующих сигналах (рис. 2). Из рис. 2 видно, что при биимпульсном зондирующем сигнале резко уменьшается длительность отраженного сигнала, на который

практически не влияет отношение t_+ / t_-. Введем коэффициент для оценки эффективности использования биимпульсного зондирующего сигнала по сравнению с прямоугольным:

$$K_{eff} = \frac{U_r \cdot t_{u2}}{t_{ur} \cdot U_2}, \qquad (4)$$

где U_2, U_r – амплитуды, t_{u2}, t_{ur} – длительности на уровнях $m \cdot U_2$ и $m \cdot U_r$ при использовании прямоугольного и биимпульсного зондирующего сигнала.

отношении $t_+ / t_- = 1$ и $S_{im} = 100$ В·нс. Видно, что существует оптимальная длительность биимпульса, обеспечивающая максимальный коэффициент эффективности. Оптимальная длительность биимпульса увеличивается при увеличении расстояния до неоднородности примерно пропорционально квадрату этого расстояния.

При использовании зондирующего биимпульса отраженный от неоднородности сигнал имеет нежелательный отрицательный выброс.

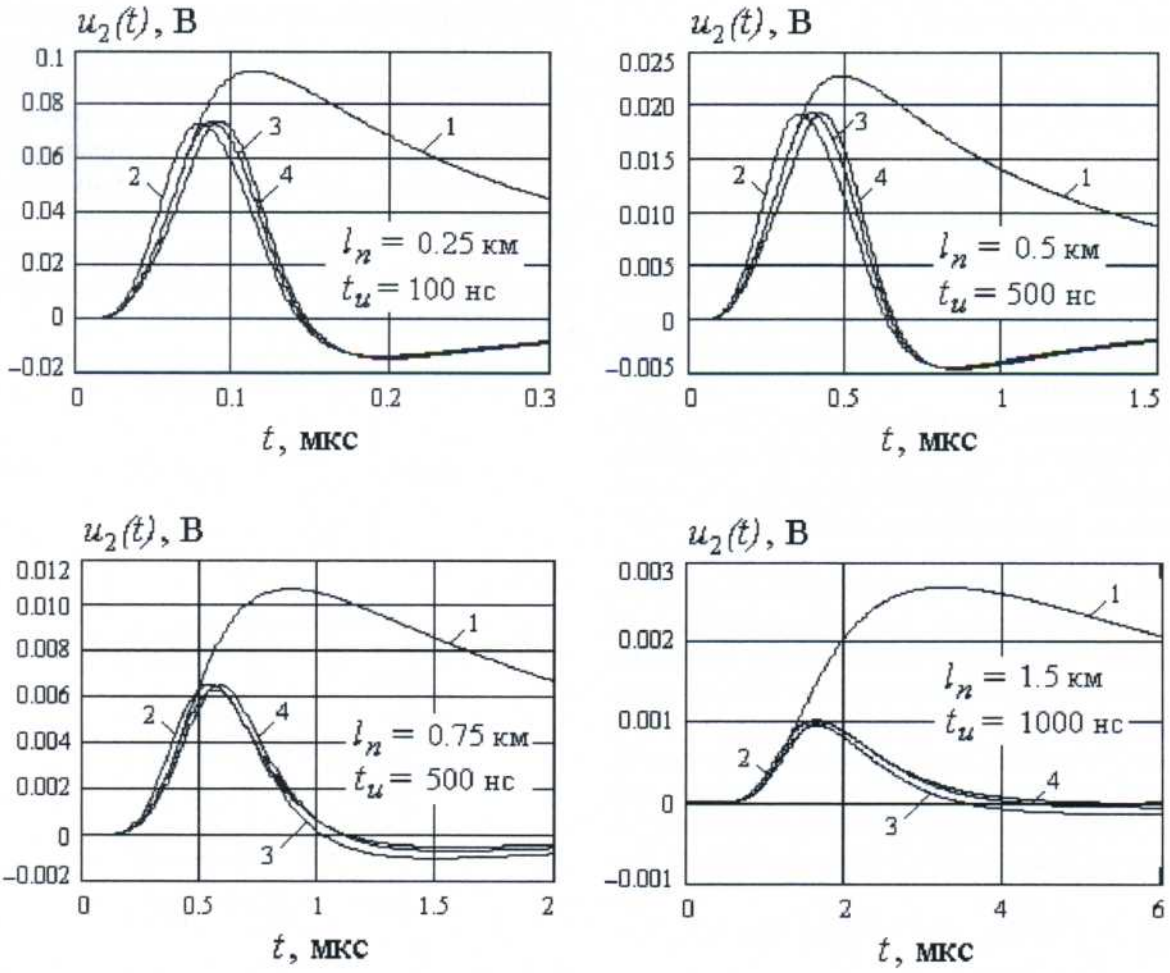


Рис. 2. Отраженные от обрыва сигналы, рассчитанные для прямоугольного (1) и биимпульсных (2–4) зондирующих импульсах при S_{im} = 100 В·нс: 2 – $t_+ / t_- = 0.6$, 3 - $t_+ / t_- = 1$, 4 – $t_+ / t_- = 1.4$

На рис. 3 показаны зависимости коэффициента эффективности от длительности биимпульса для различных расстояний до обрыва кабельной цепи ТПП-0.4. Расчет проведен при

Для его уменьшения можно модифицировать биимпульс, изменив соотношение площадей положительной S_{im+} и отрицательной S_{im-} его частей в сторону увеличения площади поло-

жительной части. Были проведены расчеты коэффициента эффективности и отношения U_-/U_r для различных значений отношения S_{im+}/S_{im-}, которые показали что, увеличивая отношение S_{im+}/S_{im-}, можно добиться существенного уменьшения отрицательного выброса ценой некоторого снижения коэффициента эффективности.

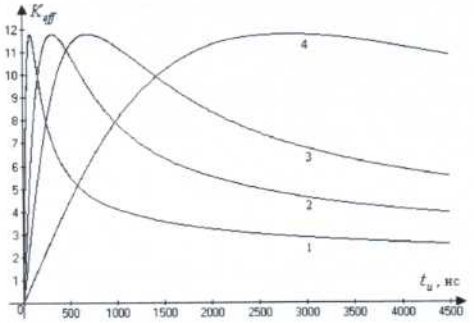


Рис. 3. Зависимости коэффициента эффективности от длительности биимпульса для $t_+/t_- = 1$ при следующих расстояниях до обрыва: 1 – l_n = 0.25 км, 2 – l_n = 0.5 км, 3 – l_n = 0.75 км, 4 – l_n = 1.5 км

Для различных расстояний до неоднородности в кабеле ТПП-0.4 были проведены расчеты зависимостей K_{eff} и U_-/U_r от отношения S_{im+}/S_{im-} при длительностях биимпульса, оптимальных для каждого из этих расстояний. Было обнаружено, что эти зависимости для различных расстояний совпадают (рис. 4). Из рис. 4 видно, что с увеличением отношения S_{im+}/S_{im-} быстро уменьшается амплитуда отрицательного выброса и при $S_{im+}/S_{im-} = 1.19$ он исчезает. При этом коэффициент эффективности сохраняет достаточно большое значение – около 9.

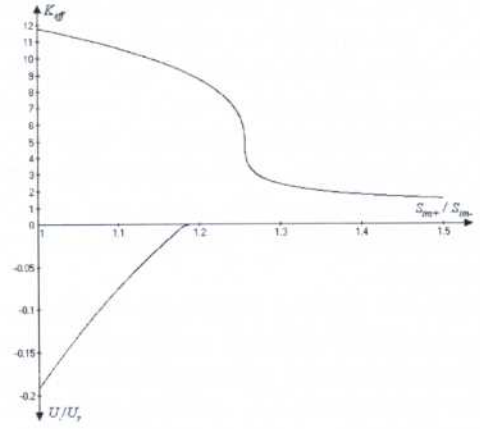


Рис. 4. Зависимость коэффициента эффективности и отношения U_-/U_r от отношения S_{im+}/S_{im-} при оптимальной длительности зондирующего импульса

Реальная рефлектограмма регистрируется при определенной, выбранной оператором, длительности зондирующего биимпульса. Эта длительность является оптимальной для конкретного расстояния. Для неоднородностей, расположенных ближе этого расстояния, на отраженном сигнале будет наблюдаться отрицательный выброс. Для оптимального и больших расстояний отрицательного выброса не будет.

Использование биимпульсного зондирующего сигнала позволяет существенно повысить разрешающую способность импульсного метода. Под разрешающей способностью понимается минимальное расстояние между двумя неоднородностями, которое дает возможность различить их на рефлектограмме [5].

Для объективной оценки разрешающей способности был введен специальный коэффициент разрешения (рис. 5):

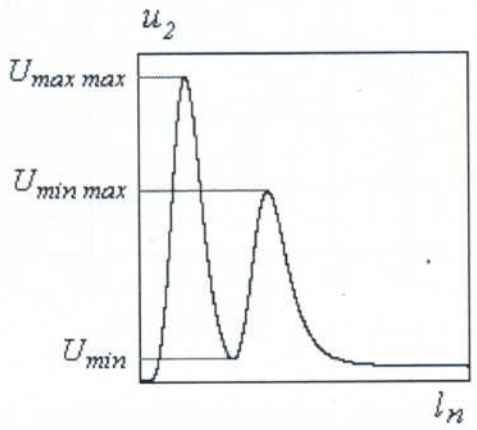


Рис. 5. Напряжения сигнала обратного потока, входящие в выражение (5)

$$K_{razr2} = \frac{2\cdot\left(U_{2\min\max} - U_{2\min}\right)}{U_{2\max\max} + U_{2\min\max}}. \qquad (5)$$

Будем считать, что неоднородности различимы, если коэффициент разрешения превышает 0.13.

Были проведены расчеты сигналов, отраженных от двух близко расположенных неоднородностей в кабельной цепи ТПП-0.4, имеющих один и тот же знак коэффициента отражения, при зондировании биимпульсом и прямоугольным импульсом. Расчеты проведены для оптимальных длительностей биимпульса, то есть для l_n = 0.25 км – t_u = 75 нс; для l_n = 0.5 км – t_u = 300 нс. Отношение площадей положительной и отрицательной частей биимпульса S_{im+}/S_{im-} выбрано равным 1.19. Некоторые результаты расчетов приведены на рис. 6.

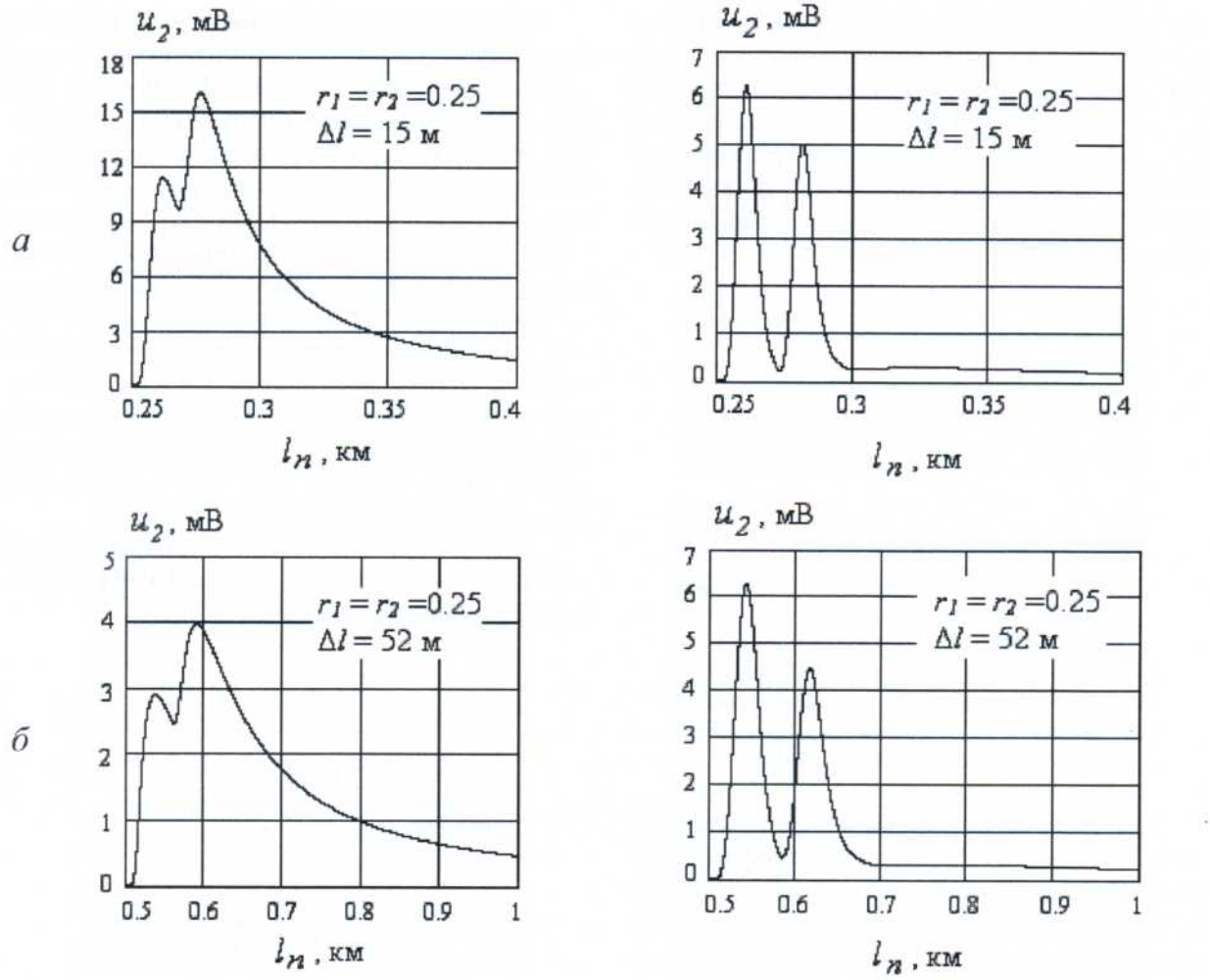


Рис. 6. Сигнал обратного потока кабельной цепи с двумя неоднородностями, расположенными на расстоянии Δl друг от друга при зондировании ее прямоугольным импульсом (слева) и биимпульсом (справа). Расчет произведен для кабельной цепи ТПП-0.4 при следующих расстояниях до первой неоднородности: а – 0.25 км; б – 0.5 км, в – 0.75 км и значениях S_{im+} зондирующего биимпульса: а – S_{im+} =37.5 нс·В; б – S_{im+} =150 нс·В (длительность прямоугольного импульса выбиралась равной t_+)

По результатам расчетов получена зависимость разрешающей способности от расстояния до первой неоднородности при использовании зондирующего биимпульса оптимальной длительности (рис. 7). Для сравнения на этом же рисунке приведена аналогичная зависимость, рассчитанная для прямоугольных зондирующих импульсов с длительностями, равными половине длительности соответствующих биимпульсов. Из рис. 7 видно, что использование биимпульсов позволяет существенно улучшить разрешающую способность.

Основным достоинством использования биимпульсного зондирующего сигнала является возможность улучшения разрешающей способности в реальном масштабе времени путем регистрации нескольких рефлектограмм неоднородного кабельного участка биимпульсами с оптимальными для данного расстояния длительностями. К сожалению, существующие импульсные приборы используют, как правило, только прямоугольный зондирующий импульс. Поэтому рассмотренные преимущества не могут на них быть реализованы.

В данной работе предлагается методика последующей обработки цифровых отсчетов зарегистрированной рефлектограммы, позволяющая преобразовать ее в "биимпульсную" рефлектограмму — рефлектограмму, которая могла бы быть получена при использовании зондирующего биимпульса.

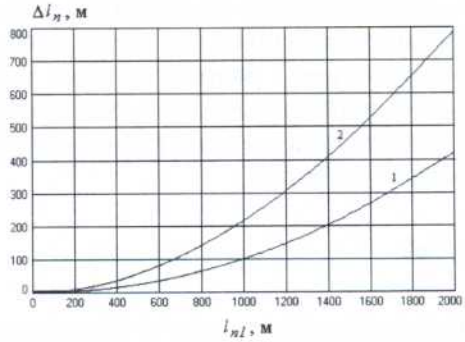


Рис. 7. Зависимости разрешающей способности от расстояния до первой неоднородности при использовании в качестве зондирующего сигнала биимпульса оптимальной длительности (1) и прямоугольного импульса (2)

Общая идея обработки основана на том, что неоднородная кабельная цепь является линейной и рефлектограммы, полученные при ее зондировании прямоугольными импульсами одинаковой длительности и амплитуды, но различной полярности, отличаются только знаком. Задержка зондирующего импульса на определенное время приводит только к смещению рефлектограммы на то же самое время. Поэтому рефлектограмма, полученная при использовании биимпульса в качестве зондирующего сигнала, может рассматриваться как сумма двух рефлектограмм, полученных при зондировании кабельной цепи прямоугольными импульсами одинаковой длительности, но разной полярности и смещенных друг относительно друга на время, равное их длительности.

На основании этого представления предложен алгоритм обработки рефлектограммы, зарегистрированной с помощью прямоугольного зондирующего импульса. В соответствии с алгоритмом исходная рефлектограмма инвертируется, сдвигается на время, равное длительности зондирующего прямоугольного импульса, и домножается на коэффициент, который соответствует оптимальному соотношению площадей положительной и отрицательной составляющих эквивалентного биимпульса. Полученная таким образом рефлектограмма суммируется с исходной (рис. 8).

Рассчитанные ранее сигналы обратного потока при прямоугольном зондирующем импульсе были обработаны по предлагаемому алгоритму. Обработанные сигналы сравнивались с сигналами, рассчитанными при тех же условиях для зондирующего биимпульса удвоенной длительности. Было обнаружено, что эти сигналы полностью идентичны.

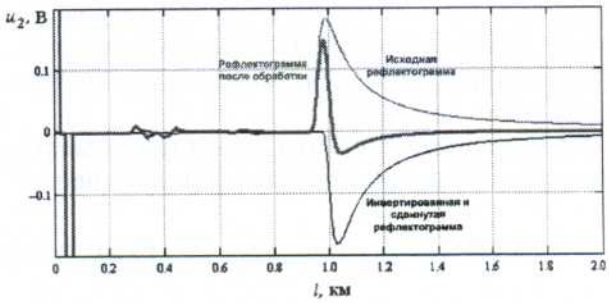


Рис. 8. Предлагаемая методика обработки зарегистрированной рефлектограммы

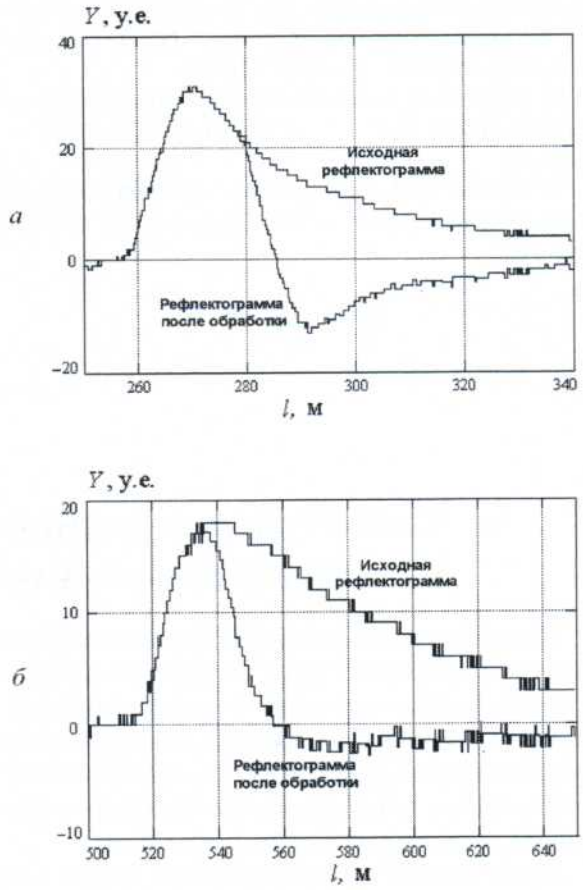


Рис. 9. Применение предлагаемой обработки к экспериментально зарегистрированным рефлектограммам кабельных цепей ТПП-0.4, оборванных на расстоянии 254 м (а) и 502 м (б)

Эффективность предложенной методики была проверена экспериментально. С помощью цифрового импульсного прибора РИ-20м регистрировались сигналы, отраженные от неоднородностей, расположенных на различном расстоянии в кабельной цепи ТПП-0.4. Зарегистрированные рефлектограммы обрабатывались по предложенному алгоритму. Некоторые

из полученных результатов представлены на рис. 9. Видно, что предлагаемая обработка значительно улучшает разрешающую способность импульсного метода.

Авторы надеются, что представленные результаты исследования будут полезны специалистам, занимающимся профилактическими и аварийными измерениями на медножильных кабельных линиях, разработчикам новых импульсных приборов и создателям программного обеспечения к существующим цифровым импульсным приборам.

СПИСОК ЛИТЕРАТУРЫ

1. **Воронцов А.С., Фролов П.А.** Импульсные измерения коаксиальных кабелей связи. — М.: Радио и связь, 1985. — 96 с., ил.

2. **Былина М.С., Глаголев С.Ф.** Определение характера повреждения или неоднородности по рефлектограмме кабельной цепи. — Труды учебных заведений связи № 168, СПб, 2002.

3. **Андреев В.А.** Временные характеристики кабельных линий связи. — М.: Радио и связь, 1986. — 104 с., ил.

4. **Шалыт Г.М.** Определение мест повреждения в электрических сетях. — М.: Энергоиздат, 1982. — 312 с., ил.

5. **Тарасов Н.А.** Использование метода импульсной рефлектометрии для определения повреждений кабельных линий. [Электронный ресурс] / "СТЭЛЛ". — Электрон. дан. (1 файл) — [199-]. — Режим доступа http://reis.narod.ru/metod.htm, свободный. — Яз. рус.

6. **Былина М.С., Глаголев С.Ф.** Повышение точности и информативности импульсного метода измерений. — Сборник трудов 4-й Всероссийской конференции "Современные технологии проектирования, строительства и эксплуатации линейно-кабельных сооружений — СТЛКС", СПб, 2005. — 6 с.

Сидоров Ю.Е., Лаврентьев Н.В.

Обнаружитель радиосигналов в каналах связи с неизвестными характеристиками замираний

Одними из наиболее быстро развивающихся областей информационных технологий являются беспроводные сети и системы сотовой связи. Их работа неразрывно связана с обнаружением и различением в шумах сигналов, подверженных искажениям и замираниям, что характерно для функционирования каналов радиосвязи.

Существенной особенностью задач обработки сигналов в радиотехнических системах является использование для их решения вероятностно-статистического подхода, что обусловлено случайным характером процесса взаимодействия электромагнитных волн с атмосферой и подстилающей поверхностью, наличием ошибок в коэффициентах и переменных, описывающих процесс распространения радиосигналов, и различного рода помех. В этих условиях важно найти такой алгоритм обработки, который позволил бы решить задачу наилучшим в некотором смысле (или оптимальным) образом.

Теоретической базой для синтеза систем обработки радиосигналов должна быть теория статистических решений, дающая единый подход к выбору оптимальных правил обработки случайных сигналов. Эта теория впервые сформулирована в конце 30-х и в 40-е годы XX века в работах А. Вальда [1]. В последующие годы усилиями советских и зарубежных ученых была в основном закончена теория принятия решений, основанная на использовании различных критериев качества решающих процедур (Байеса, минимакса, максимальной апостериорной вероятности, максимального правдоподобия и др.) и ставшая классической [2]. Однако для ее применения требуются обширные априорные сведения о моделях и статистических характеристиках сигналов и помех, что на практике далеко не всегда бывает.

Таким образом, классические методы синтеза характеризуют потенциальную эффективность систем обработки радиосигналов

и приводят к реализации устройств, очень чувствительных к любому отклонению входных характеристик от заранее рассчитанных (исследуемых).

В связи с этим большую актуальность приобретает разработка и применение методов преодоления возникающей априорной неопределенности, являющейся специфическим видом помех, эффективная борьба с которыми должна осуществляться специальными средствами [3].

В данной статье рассмотрим задачу обнаружения сигнала, подверженного замираниям, в канале передачи информации на фоне белого гауссовского шума с неизвестной спектральной плотностью [4]. Для решения этой задачи необходимо синтезировать правило, которое должно отвечать следующим практически важным требованиям: а) обладать стабильной вероятностью ложной тревоги при любых изменениях спектральной мощности шума; б) обеспечивать инвариантность вероятности ложной тревоги относительно начальной фазы сигнала; в) максимизировать вероятность правильного обнаружения при всех отношениях сигнал/шум.

Исходные данные для синтеза правила обнаружения сигнала с релеевскими замираниями в шумах неизвестной мощности, удовлетворяющего представленным выше требованиям, таковы [4].

Осуществляется обнаружение сигнала $\sqrt{Es}(t,\phi_0)$ с произвольной начальной фазой ϕ_0 на фоне гауссовского шума. Шум на входе линейного тракта приемника (ЛТП) полагаем белым, сигнал $s(t, \phi_0)$ при фиксированной фазе ϕ_0 — известным и имеющим единичную энергию. Наблюдаемым процессом $x(t)$, $t \in [0, T]$ считаем колебание на выходе ЛТП с П-образной амплитудно-частотной характеристикой, единичным коэффициентом передачи и полосой пропускания Δf. К априорно неопределенным параметрам принятого сигнала относим энергию $E \in (0, \infty)$ и начальную фазу $\phi_0 \in (-\pi, \pi]$, к априорно неопределенному параметру шума — спектральную плотность $N \in (0, \infty)$.

Перейдем к рассмотрению правила обнаружения, которое можно получить с помощью теории комбинированного многокомпонентного статистического синтеза [3], базирующегося на результатах фундаментальных работ А. Вальда [1], Э. Лемана [5] и Ю.В. Линника [6].

Образуем выборку $\dot{x} = (\dot{x}_1, ..., \dot{x}_\nu)$ объема $\nu = \Delta f T$ из отсчетов $\dot{x}_i$ комплексной огибающей $\dot{x}(t)$ наблюдаемого процесса в моменты времени $t_i = i/\Delta f$. Плотность вероятности этой выборки

$$W(\dot{x}|E;\ \varphi_0;\ N_0) = \\ = k\exp\left\{-\frac{1}{4N_0\Delta f}\left\|\dot{x} - \sqrt{E}\exp(-i\varphi_0)\dot{s}\right\|^2\right\}, \quad (1)$$

где $k = 1/(4\pi N_0\Delta f)^\nu$; $\dot{s} = (\dot{s}_1, ..., \dot{s}_\nu)$ — сигнальный вектор, образованный из отсчетов комплексной огибающей $\dot{s}(t)$ сигнала $s(t;\ \phi_0)$ при $\phi_0 = 0$; $\|\cdot\|$ — норма в комплексном евклидовом пространстве.

Введем полезный параметр $\vartheta = (\vartheta_1, \vartheta_2)$, где $\vartheta_1 = \sqrt{E}\cos\varphi_0/(2N_0)$, $\vartheta_2 = \sqrt{E}\sin\varphi_0/(2N_0)$; мешающий параметр $\pi = 1/(4N_0)$ и статистики

$$U_1(\dot{x}) = \frac{1}{\Delta f}\mathrm{Re}\langle\dot{x};\ \dot{s}\rangle,\ U_2(\dot{x}) = \frac{1}{\Delta f}\mathrm{Im}\langle\dot{x};\ \dot{s}\rangle,$$

$T(\dot{x}) = \frac{1}{\Delta f}\|\dot{x}\|^2$, $\langle .., .. \rangle$ — скалярное произведение в комплексном евклидовом пространстве. При большом произведении $\Delta f T$ (большой базе сигнала) статистики U_1, U_2 и T можно приближенно представить в интегральной форме:

$$U_1(\dot{x}) = \mathrm{Re}\int_0^T \dot{x}(t)\dot{s}^*(t)dt;\ U_2(\dot{x}) = \mathrm{Im}\int_0^T \dot{x}(t)\dot{s}^*(t)dt;$$

$$T(\dot{x}) = \int_0^T |\dot{x}(t)|^2 dt. \quad (2)$$

Тогда плотность вероятности (1) можно записать в следующем виде:

$$W(\dot{x}|\vartheta) = k\exp\left(-\frac{\vartheta_1^2 + \vartheta_2^2}{4\pi}\right) \cdot \\ \cdot\exp\left[-\pi T(\dot{x}) + \vartheta_1 U_1(\dot{x}) + \vartheta_2 U_2(\dot{x})\right], \quad (3)$$

где $\vartheta = (\vartheta_1, \vartheta_2, \pi)$.

Через параметр ϑ задача обнаружения формулируется как задача проверки сложных гипотез:

$$H_0 : \vartheta \in \Theta_0 = \{\vartheta : \vartheta_1^2 + \vartheta_2^2 = 0;\ 0 < \pi < \infty\};$$

$$H_0 : \vartheta \in \Theta_1 = \{\vartheta : 0 < \vartheta_1^2 + \vartheta_2^2 < \infty;\ 0 < \pi < \infty\}.$$

Для решения этой задачи не удается воспользоваться принципом несмещенности [3,4,5], так как полезный параметр $(\vartheta_1, \vartheta_2)$ не одноме-

рен [3,4]. В монографии [4] была рассмотрена и доказана возможность применения принципа инвариантности. Его применение позволяет построить правило проверки гипотез H_0 и H_1 с решающей функцией [4]:

$$\varphi(x)=\begin{cases}1 \text{ при } U_1^2(\dot{x})+U_2^2(\dot{x})\geq C(\alpha)T(\dot{x}).\\ 0 \text{ в остальных случаях.}\end{cases} \quad (4)$$

Правило (4) обладает стабильной вероятностью ложной тревоги при любых изменениях спектральной плотности шума, обеспечивает инвариантность вероятности правильного обнаружения относительно начальной фазы сигнала, максимизирует эту вероятность для всех отношений сигнал/шум. Поэтому оно применимо в случае общих замираний, которые сопровождаются изменениями амплитуды и начальной фазы сигнала.

Моделирование обнаружителя, реализующего правило (4), сводится к проверке неравенства в его решающей функции. Были составлены операторная и блок-схема моделирующего алгоритма, а для получения случайных величин с различными распределениями использовались результаты работы [7]. Моделирование проводилось в среде C++ Builder 6. Оценка характеристик проводилась методом Монте-Карло, с использованием которого было успешно проведено имитационное моделирование обнаружителя радиосигналов [8]. Было проведено 150000 испытаний. Погрешность моделирования при этом числе испытаний составляет 0,026%.

Наиболее интересной для рассмотрения является зависимость вероятности правильного обнаружения от параметра $\nu = \Delta f T$ — произведения полосы пропускания ЛТП на время наблюдений. Моделирование проводилось для значений вероятности ложной тревоги $\alpha = 10^4$ и $\nu = 5, 10, 50$ (на рис.1 нижняя кривая соответствует $\nu = 5$, средняя — $\nu = 10$, верхняя — $\nu = 50$) .

Осуществим проверку полученных результатов. Вероятность правильного обнаружения может быть также получена усреднением вероятности правильного обнаружения правила (4) по распределению амплитудных флуктуаций сигнала [4]. В частности, при рэлеевском распределении амплитудных флуктуаций эта вероятность определяется как

$$D=\alpha\left(\frac{1+q^2}{1+q^2\alpha^{1/\nu}}\right)^{\nu}. \quad (5)$$

Из данного выражения следует, что вероятность правильного обнаружения растет с увеличением параметра ν и в пределе становится равной вероятности $D=\alpha^{1/(1+q_{cp}^2)}$ правильного обнаружения флюктуирующего сигнала в шумах известного уровня.

Моделирование процесса в соответствии с формулой (5) проводилось в среде C++ Builder 6, при выбранном значении вероятности ложной тревоги $\alpha = 10^{-4}$ и $\nu = 10$.

На рис.2 представлены две кривые, соответствующие зависимостям вероятности правильного обнаружения от отношения сигнал/шум при $\alpha = 10^{-4}$ и $\nu = 10$ для имитационного статистического моделирования правила (4)

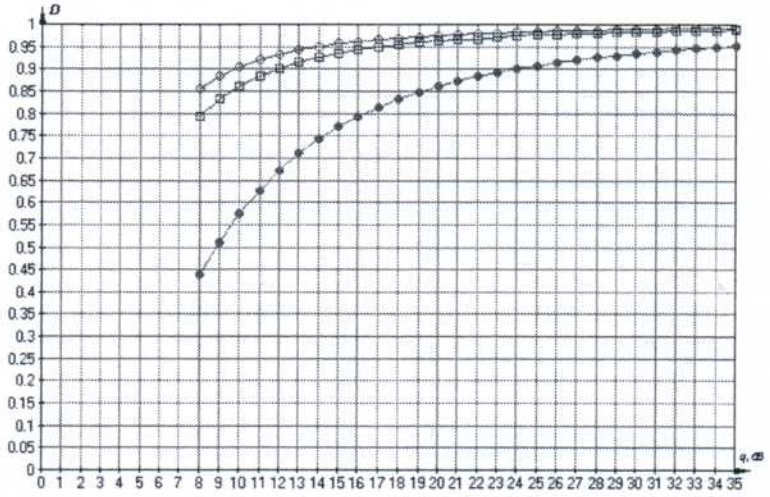


Рис. 1. Зависимость вероятности правильного обнаружения от отношения сигнал/шум q при $\nu = 5, 10, 50$; $\alpha = 10^{-4}$

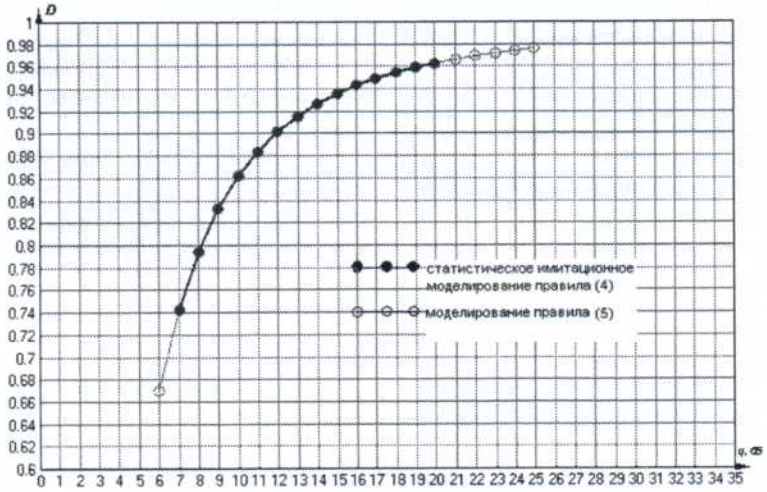


Рис. 2. Зависимость вероятности правильного обнаружения от отношения сигнал/шум q при $\nu = 10$; $\alpha = 10^{-4}$ для статистического имитационного моделирования правила (4) и моделирования правила (5)

и моделирования правила (5). Как видно из графика, визуально кривые полностью совпадают. Это объясняется тем, что, как было отмечено выше, погрешность моделирования при проведении 150000 испытаний составляет 0,026%.

Проанализируем полученные результаты. Как видно из представленных на рис.1 характеристик, вероятность правильного обнаружения с увеличением параметра ν растет. Возникает вопрос, какой величины достаточно выбрать ν, чтобы считать обнаружитель эффективным. Рассмотрим выигрыши по отношению сигнал/шум при увеличении ν при выбранном значении вероятности ложной тревоги $\alpha = 10^{-5}$. Переход от $\nu = 5$ к $\nu = 10$ дает выигрыш по уровню 0,9 12 дБ. Дальнейший переход от $\nu = 10$ к $\nu = 50$ дает выигрыш уже только 2 дБ. Очевидно, что с ростом ν выигрыш от этого роста падает. При этом для увеличения ν необходимо увеличивать либо полосу пропускания ЛТП, либо время наблюдения. И тот и другой путь решения проблемы могут быть недопустимыми в конкретной ситуации. Например, мы не можем увеличивать время наблюдения, когда система обнаружения принимает решение о наличии или отсутствии цели, здесь дорога каждая сотая доля секунды. А при расширении полосы пропускания ЛТП мы можем столкнуться с проблемой сосредоточенных по частоте помех, а также с трудностями реализации широкополосных антенн. Поэтому выбор ν целиком и полностью зависит от выбранного критерия эффективности.

Заключение

Получено решающее правило обнаружения радиосигналов в каналах связи с неизвестными характеристиками замираний, удовлетворяющее сформулированным выше требованиям, определяющим его работоспособность в реальных условиях функционирования радиотехнических систем и позволяющим автоматизировать процесс обнаружения. Показано, что эффективность обнаружения возрастает с увеличением произведения полосы пропускания линейного тракта приемника на время наблюдений.

Синтезированное правило (с учетом оценки его эффективности) можно рекомендовать для использования в автоматизированных системах радиосвязи, функционирующих в условиях априорно неизвестной сигнально-помеховой обстановки.

СПИСОК ЛИТЕРАТУРЫ

1. **Вальд А.** Основные идеи общей теории статистических решений// Последовательный анализ. М., Физматгиз, 1960. С. 308–325.

2. **Цикин И.А.** Оптимальная обработка сигналов в радиотехнических системах/ Учебное пособие. Л.: ЛПИ им. М.И. Калинина, 1986. 77 с.

3. **Сидоров Ю.Е.** Статистический синтез автоматизированных решающих систем при априорной неопределенности. М.: Воениздат, 1993. 227 с.

4. **Теория обнаружения сигналов** /П.С. Акимов, П.А. Бакут, В.А. Богданович и др. /Под ред. П.А. Бакута. — М.: Радио и связь, 1984.

5. **Леман Э.** Проверка статистических гипотез. М.: Наука, 1964. 498 с.

6. **Линник Ю.В.** Статистические задачи с мешающими параметрами. М.: Наука, 1966. 252 с.

7. **Быков В.В.** Цифровое моделирование в статистической радиотехнике. М.: Сов. Радио, 1971. 326 с.

8. **Сидоров Ю.Е., Лаврентьев Н.В.** Оптимальный обнаружитель радиосигналов: решающее правило, статистическое имитационное моделирование.- Труды СпбГТУ, 2008, №507, с. 118–124.

Рыжов В.А., Кипоть В.Л., Биряльцев Е.В.

Параметризация спектров в технологии низкочастотного сейсмического зондирования на основе вейвлет-анализа

Введение

В пассивной технологии низкочастотного сейсмического зондирования[1] (НСЗ) [1], направленной на поиск нефти и газа, регистрируются фоновые микросейсмические колебания поверхности Земли. Известно [2], что в низкочастотном диапазоне, от 1 Гц до 10 Гц, наблюдается корреляция интенсивности спектральных максимумов (рис. 1) с наличием нефтегазовой залежи в разрезе. Вместе с тем наши исследования физической природы эффекта [3], а также исследования ряда зарубежных компаний [4] указывают на явление резонанса продольных сейсмических волн между акустически контрастными плоско-параллельными границами природных резонаторов. Следовательно, увеличение спектрального максимума связано с увеличением добротности колебательной системы. Исходя из теоретических представлений о резонансных явлениях следует, что добротность можно оценить двумя способами: 1) по отношению частоты собственных колебаний к полосе пропускания системы; 2) по отношению амплитуды гармонического сигнала на выходе системы к амплитуде на входе при частоте, соответствующей собственным колебаниям системы. Чтобы использовать данные способы оценки добротности, необходимо знать основные параметры спектральных максимумов: частота, ширина полосы, сигнал/шум. В данной статье решается задача разработки алгоритма параметризации спектральных кривых с целью оценки информативных параметров.

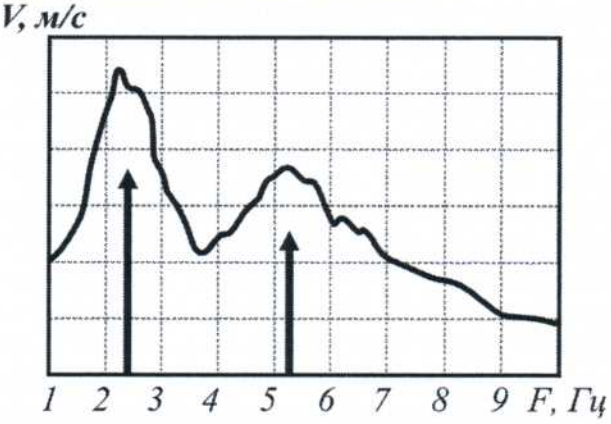


Рис.1. Пример спектра НСЗ

Основная часть

На рис. 2 представлена модель спектрального максимума и его параметры. В аналитической форме она определяется как

$$sp(f, f_0, \sigma_0, S, N) = \begin{vmatrix} S \cdot \left(1 - \frac{(f-f_0)^2}{\sigma_0^2}\right) \cdot e^{\frac{(f-f_0)^2}{2\sigma_0^2}} + \\ +N \left| x \in (-\sigma_0, \sigma_0) \right| x \in (-\infty; -\sigma_0] \cap [\sigma_0; +\infty) \end{vmatrix}, \quad (1)$$

[1] Работа выполнена по материалам компании ЗАО "Градиент".

где f_0 — частота, S — амплитуда максимума над фоном, N — амплитуда фона, σ_0 — полуширина основания максимума, SNR — соотношение "сигнал/шум", Δf — ширина.

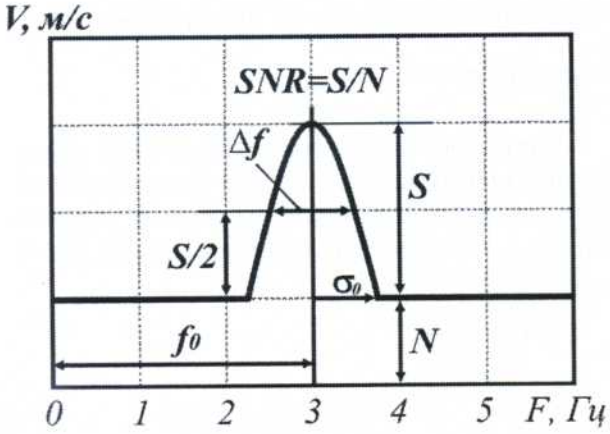


Рис. 2. Модельный максимум на фоне модельного шума

Для решения поставленной задачи был разработан алгоритм, основанный на вейвлет-преобразовании. В данном случае вейвлет-преобразование применяется к амплитудным спектрам микросейсмических сигналов, а не самим сигналам. Идея применения вейвлетов заключается в том, чтобы оценить параметры частота f_0 [Гц] и ширина базисного вейвлета σ_0 [Гц] (масштаб вейвлета), при которых вейвлет образ обретает локальный максимум. Применение вейвлетов здесь вполне обосновано, так как в рассматриваемых спектрах содержится многообразие спектральных максимумов во всем частотном диапазоне с различной шириной и амплитудой. В качестве базисного вейвлета выбрана вторая производная функции Гаусса (мексиканская шляпа) [5]:

$$\varphi(f,\sigma)=\frac{\sqrt{e}}{2\sigma}\cdot\left(1-\frac{f^2}{\sigma^2}\right)\cdot e^{-\frac{f^2}{2\sigma^2}}. \qquad (2)$$

Такой выбор сделан для соответствия положительной части этого вейвлета форме модельного спектрального максимума (1).

Алгоритм определения координат локальных максимумов вейвлет-образа реализован с учетом особенностей неравномерного распределения плотности локальных максимумов. В классическом ортогональном вейвлет-разложении для того, чтобы полностью описать сигнал, достаточно только масштабов $\sigma(b) = 2^b$, где $b = 0,1,2\ldots$. Плотность максимумов вейвлет-образа увеличивается с понижением масштаба σ. Во избежание избыточности вычислений с учетом этих закономерностей генерируется массив стартовых точек в плоскости (f, σ). Для каждой стартовой точки выполняется уточнение координат f и σ ближайшего локального максимума с применением метода прямого поиска Хука-Дживса [6]. Несколько стартовых точек могут собраться возле одного локального максимума. Это приводит к избыточности информации в выходной таблице, поэтому применяются методы кластерного анализа для объединения их в одну группу, а затем в одну точку через усреднение координат группы. Кластерный анализ позволяет выделить группу аномалий с близкими параметрами, представить наблюдаемые данные в наглядные структуры. Наиболее популярные методы кластерного анализа – объединение (древовидная кластеризация) и метод К-средних. В данном случае использовался метод объединения. Важную роль в кластерном анализе играет понятие мера расстояния. Наиболее прямой путь вычисления расстояний между объектами в многомерном пространстве состоит в вычислении евклидовых расстояний. На расстояния сильно влияют различия между осями, по координатам которых вычисляются эти расстояния. С учетом выше приведенных закономерностей вейвлет-разложения мера близости между двумя точками на вейвлет-образе определяется как

$$l(f_1,\sigma_1,f_2,\sigma_2)=\sqrt{\left(\frac{f_2-f_1}{(\sigma_2+\sigma_1)/2}\right)^2+\left(\log_2\sigma_2-\log_2\sigma_1\right)^2}.$$

В результате работы алгоритма формируется сводная таблица, каждая строка которой содержит оценку параметров f_0 и σ_0 одного из спектральных максимумов.

Параметр сигнал/шум SNR определятся через введение дополнительной функции, которая в положительной области значений полностью совпадает с (2):

$$\chi(f,\sigma)=\begin{cases}\dfrac{\sqrt{e}}{2\sigma}\cdot\left(1-\dfrac{f^2}{\sigma^2}\right)\cdot e^{-\frac{f^2}{2\sigma^2}} & \left|x\in(-\sigma,\sigma)\right.\\ 0 & \left|x\in(-\infty;-\sigma]\cap[\sigma;+\infty)\right.\end{cases} \qquad (3)$$

Обе функции (2) и (3) нормированы из условий:

$$\int_{-\sigma}^{\sigma}\chi(f,\sigma)df\equiv 1 \qquad \int_{-\sigma}^{\sigma}\varphi(f,\sigma)df\equiv 1.$$

Свертка модельного спектрального максимума (1) с функциями (2) и (3) вычисляется:

$$\Phi_{sp}^{\varphi}(f,\sigma)=\frac{1}{\sqrt{|\sigma|}}\int sp(f')\varphi^{*}(f'-f,\sigma)df'$$

$$X_{sp}^{\chi}(f,\sigma)=\frac{1}{\sqrt{|\sigma|}}\int sp(f')\chi^{*}(f'-f,\sigma)df'.$$

В грубом приближении параметр *SNR* для модельного спектрального максимума определяется как

$$snr=\frac{\Phi_{sp}^{\varphi}(f_0,\sigma_0)}{X_{sp}^{\chi}(f_0,\sigma_0)-\Phi_{sp}^{\varphi}(f_0,\sigma_0)}. \qquad (6)$$

$\Phi^{\varphi}_{sp}(f_0, \sigma_0)$даст величину, пропорциональную S. $X^{\chi}_{sp}(f_0, \sigma_0)$ даст величину, пропорциональную $S + N$. Поэтому

$$snr\sim\frac{S}{(S+N)-(S)}=\frac{S}{N}.$$

Чтобы получить коэффициент пропорциональности α, вычислим snr/SNR, где $SNR = S/N$, по определению

$$\frac{snr}{SNR}=\frac{\dfrac{\Phi_{sp}^{\varphi}(f_0,\sigma_0)}{X_{sp}^{\chi}(f_0,\sigma_0)-\Phi_{sp}^{\varphi}(f_0,\sigma_0)}}{\dfrac{S}{N}}=$$

$$=\int_{-\sigma_0}^{\sigma_0}\varphi(f_0,\sigma_0)^2df.$$

Таким образом, α есть функция масштаба вейвлета σ и вычисляется:

$$\alpha(\sigma_0)=\int_{-\sigma_0}^{\sigma_0}\varphi(f_0,\sigma_0)^2df. \qquad (7)$$

Подставив (2) в (7), получаем: $\alpha(\sigma_0) = 0{,}3863/\sigma_0$. В результате

$$SNR=\frac{\sigma_0}{0.3863}\frac{\Phi_{sp}^{\varphi}(f_0,\sigma_0)}{X_{sp}^{\chi}(f_0,\sigma_0)-\Phi_{sp}^{\varphi}(f_0,\sigma_0)}.$$

Ширина полосы модельного максимума Δf (рис. 2) по уровню $\frac{1}{2}S+N$ аналитически соотносится с σ_0 как

$$\Delta f = 1{,}252\cdot\sigma_0.$$

Таким образом, для каждого спектрального максимума мы оценили его параметры: f_0, σ_0, Δf, SNR и имеем возможность оценить добротность колебательной системы двумя способами:

1) $Q^{\langle 1\rangle}=\dfrac{f_0}{\Delta f}$;

2) $Q^{\langle 2\rangle}=\dfrac{A}{N}=\dfrac{S+N}{N}=\dfrac{S}{N}+1=SNR+1,$

где A — амплитуда спектрального максимума, которая характеризует результат усиления природным резонатором амплитуды входного фонового шума N.

В технологии низкочастотного сейсмического зондирования исследуются спектры с небольшими добротностями $1 < Q < 5$, поэтому часто их морфология осложняется влиянием шума. Это приводит к флуктуациям искомых параметров. На рис. 3 представлен набор спектров, для которых была выполнена оценка параметров (см. табл. 1) наиболее выраженного спектрального максимума в диапазоне от 4 до 7 Гц.

При вычислении добротности первым способом $Q^{\langle 1\rangle}$ в знаменателе стоит $2\Delta f$ вместо Δf, т.к. оцениваемый максимум является первой, а не нулевой модой явления резонанса. $Q^{\langle 1\rangle}$ зависит только от оцененной ширины и частоты спектрального максимума и не учитывает распределение энергии фона и самого максимума, как

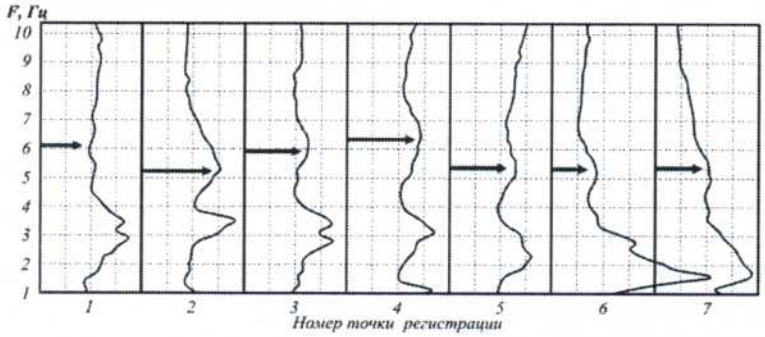


Рис. 3. Пример спектров микросейсмического поля

Таблица 1

Параметры спектральных максимумов, полученных предложенным алгоритмом в условиях их слабой выраженности.

№	Δf, Гц	σ_0, Гц	Δf, Гц	snr+1	$Q^{(1)} = f_0/(2 \cdot \Delta f)$	$Q^{(2)} = SNR + 1$
1	6,16	0,94	1,17	1,12	2,63	1,28
2	5,22	0,66	0,83	1,23	3,15	1,39
3	5,93	0,98	1,23	1,25	2,42	1,64
4	6,30	1,16	1,46	1,22	2,16	1,66
5	5,33	0,86	1,08	1,20	2,47	1,46
6	5,30	0,63	0,79	1,22	3,33	1,36
7	5,33	0,75	0,95	1,09	2,82	1,17

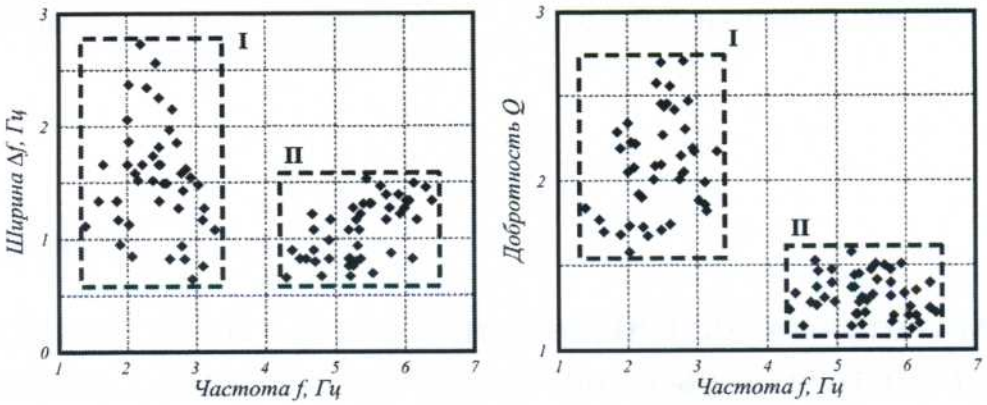


Рис.4. Результаты статистического анализа

это имеет место во втором случае $Q^{(2)}$. Поэтому при оценке добротности природного резонатора в НСЗ применяется $Q^{(2)}$.

Сопоставление полученных параметров по всей площади наблюдения позволяет выявить области их группировки, а также среднее и дисперсию. На рис. 4 представлена диаграмма распределения параметров $\Delta f(f)$ и $Q^{(2)}(f)$. Каждая точка на диаграммах занимает свое положение в соответствии с параметрами своего спектрального максимума. Точки на диаграммах группируются в кластеры. Для данной площади выделяется два кластера, соответствующие двум основным спектральным максимумам на средних частотах 2.42Гц и 5.38Гц. Параметр "добротность" характеризует выраженность аномалии, следовательно, согласно [2] пропорционален вероятности обнаружения нефтегазовой залежи. Для каждой исследуемой площади критерий принятия решений выбирается индивидуально в зависимости от геологической информации и сейсмической обстановки.

Заключение

Таким образом, для технологии НСЗ был разработан алгоритм оценки параметров спектральных максимумов, который позволяет: численно оценить информативные параметры; автоматизировать процесс параметризации. Дополнительное применение методов кластерного анализа при площадном анализе дает возможность статистически оценить характер распределения; разделить устойчивые спектральные максимумы от случайных; автоматизировать построение физических полей; численно сравнить информативные параметры между разными площадями работ.

Предложенный алгоритм внедрен в стандартный обрабатывающий комплекс НСЗ и опробован уже более чем на 5500 спектрах (с 01.2007 по 07.2008).

СПИСОК ЛИТЕРАТУРЫ

1. **Birialtsev, E. V. and I. N. Plotnikova, I. R. Khabibulin, N. Y. Shabalin,** "The analysis of microseisms spectrum at prospecting of oil reservoir on Republic Tatarstan," EAGE Conference, Saint Petersburg, Russia, 2006. http://www.earthdoc.org/detail.php?paperid=B016&edition=20&PHPSESSID=7b705646e1f439f737518cd21e91d5d8

2. АНЧАР [Электронный ресурс] Группа коммерческих компаний; — Режим доступа: http://www.anchar.ru/text.phtml?m=165, свободный. — "О технологии АНЧАР " . — Яз. рус..

3. **Rizhov, V; Birialtsev, E.** The microseism spectral analysis at the range from 1 to 20 Hz for the geology prospecting GI1-1MO5O-004; Oral: 14.4. 18:15 - 18:30 Lecture Room 2; 1607-7962/gra/EGU2008-A-05718 http://www.cosis.net/abstracts/EGU2008/05718/EGU2008-A-05718-2.pdf?PHPSESSID

4. SPECTRASEIS [Электронный ресурс] коммерческая компания / Zurich, Switzerland — Режим доступа: http://www.spectraseis.com, свободный. — "Welcome to Spectraseis". — Яз. англ.

5. **Малла С.** Вэйвлеты в обработке сигналов: Пер. с англ. — М.: Мир, 2005. — 671 с., ил. ISBN 5-03-003691-1.

6. **Медынский М.М., Антоний Е.В.,** Численные методы нелинейной оптимизации: алгоритмы и программы. Учебное пособие - Москва: МАИ, 2003. — 192 с. ISBN 5-7035-1278-6.

7. Кластерный анализ. Электронный учебник "StatSoft". 1984–2001г. http://www.statsoft.ru/home/textbook/modules/stcluan.html

Данилович Д. А.

Оптимизация характеристик антенн и высот их подвеса на многоинтервальных цифровых радиорелейных линиях

При строительстве РРЛ в большинстве случаев основные затраты относятся к антенно-мачтовым сооружениям (АМС), которые включают в себя затраты на сооружение антенных опор и стоимость самих антенн. В этой связи минимизация затрат на АМС является одной из основных задач проектирования многоинтервальных РРЛ. При этом наибольшие трудности при решении указанной задачи возникают при проектировании многоинтервальных РРЛ с двухчастотными планами в условиях наличия дополнительных неформальных ограничений.

Решение частной задачи оптимизации высот подвеса антенн на многоинтервальных ЦРРЛ с двухчастотными планами при наличии дополнительных ограничений и заданных характеристиках направленности антенн содержится в [1]. Решение другой частной задачи нахождения нескольких ранжированных лучших решений при оптимизации выбора антенн на многоинтервальных ЦРРЛ с двухчастотными планами представлено в [2]. В [3] содержится решение упрощенной задачи совместной оптимизации выбора антенн и высот их подвеса без учета дополнительных неформальных ограничений, т.е. задачи нахождения единственного – оптимального решения.

В данной статье рассматривается решение общей задачи многовариантной минимизации суммарных затрат на АМС на основе совместной оптимизации выбора антенн и высот их подвеса на многоинтервальных ЦРРЛ с двухчастотными планами с учетом дополнительных неформальных ограничений. Подобная постановка задачи предусматривает нахождение заданного числа ранжированных лучших решений и обеспечивает оптимизацию проектного решения.

Рассматриваемая задача решается на основе принципа минимума затрат, при котором суммарные затраты на все антенны, антенные опоры и фидерные (волноводные) тракты минимальны при условии выполнения существующих критериев допустимости пар высот антенн и требований к показателям качества передачи на отдельных интервалах с учетом влияния внутрисистемных помех от сигналов обратного направления (СОН).

При решении данной задачи будем считать, что заданы места расположения радиорелейных станций и характеристики радиорелейного оборудования, а высоты антенных опор определяются высотами подвеса на них верхних антенн.

Введем следующие обозначения:

K – заданное число отыскиваемых лучших решений задачи;

N – число интервалов на ЦРРЛ;

a_{nl} и a_{nr} – соответственно возможные варианты антенн на левом и правом концах n-го интервала, $n=\overline{1,N}$, определяющие типы и диаметры антенн, их коэффициенты усиления и защитного действия;

$a_n = \{a_{nl}; a_{nr}\}$ – возможная пара антенн на n-ом интервале;

h_{nl} и h_{nr} – соответственно возможные высоты подвеса левой и правой антенн на интервале n, $n=\overline{1,N}$;

$q_n = \{h_{nl}; h_{nr}\}$ – возможная пара высот антенн (ПВА) на интервале n;

$u_n = \{a_n; q_n\}$ – элемент множества возможных вариантов пар антенн и высот их подвеса (ПАВП) n-ом интервале;

G_{un} – множество возможных ПАВП n-ом интервале, $u_n \in G_{un}$;

$U = \{u_1; u_2; \ldots; u_N\}$ – упорядоченная последовательность ПАВП на всей ЦРРЛ (УП-ПАВП);

$\Omega(U)$ – дискретное множество возможных УП-ПАВП, $U \in \Omega(U)$;

$U^{(k)} = \{u_1^{(k)}, u_2^{(k)}, \ldots, u_N^{(k)}\}$ – k-я лучшая ранжированная УП-ПАВП, $k=\overline{1,K}$;

$u_n^{(k)}$ – ПАВП на интервале n, входящая в состав k-й лучшей УП-ПАВП;

$f_n(u_{n-1}; u_n)$ – затраты на n-ю опору и расположенные на ней антенны $a_{n-1,r}$ и a_{nl}, зависящие от ПАВП на $(n-1)$-м и n-м интервалах, $n=\overline{2,N}$;

$f_1(u_1)$ и $fN+1(u_N)$ – соответственно затраты на 1-ю и $(N+1)$-ю опоры и расположенные на них антенны a_{1l} и a_{Nr}, зависящие от ПАВП на 1-м и N-м интервалах;

$F(U^{(k)})$ – суммарные затраты на все опоры, соответствующие УП-ПАВП $U^{(k)}$.

Будем считать, что при $k_1 < k_2$ $U^{(k_1)}$ не хуже $U^{(k_2)}$, т. е. $F(U^{(k_1)}) \le F(U^{(k_2)})$.

С учетом введенных обозначений математическая модель рассматриваемой задачи совместной оптимизации антенн и высот их подвеса может быть записана следующим образом: найти такие $U^{(k)} = \{u_1^{(k)}, u_2^{(k)}, \ldots, u_N^{(k)}\}$, $k=\overline{1,K}$, для которых

$$F(U^{(k)}) = \min_{\Omega^{(k)}(U)} [f_1(u_1) + f_{N+1}(u_N) + \sum_{n=2}^{N} f_n(u_{n-1}; u_n)] \quad (1)$$

при условиях:

$$u_n^{(k)} \in G_{un}, \quad n=\overline{1,N}; \quad (2)$$

$$p(u_n^{(k)} / g(20\%)) \ge 1, \quad n=\overline{1,N}; \quad (3)$$

$$M_1(u_1^{(k)}) \ge M_{1.\min}(u_1^{(k)}) + \Delta M_1^{(k)}(u_1^{(k)}; u_2^{(k)}); \quad (4)$$

$$M_n(u_n^{(k)}) \ge M_{n.\min}(u_n^{(k)}) + \Delta M_n^{(k)}(u_{n-1}^{(k)}; u_n^{(k)}; u_{n+1}^{(k)}), \quad n=\overline{2,(N-1)}; \quad (5)$$

$$M_N(u_N^{(k)}) \ge M_{N.\min}(u_N^{(k)}) + \Delta M_N^{(k)}(u_{N-1}^{(k)}; u_N^{(k)}); \quad (6)$$

$$\Omega^{(1)}(U) = \Omega(U); \quad \Omega^{(k)}(U) = \Omega(U) / \Omega^{(k-1)*}(U). \quad k = 2, K. \quad (7)$$

Входящие в целевую функцию (1) стоимости промежуточных антенных опор вместе с находящимися на них антеннами $f_n(u_{n-1}; u_n) = f_a[a_{n-1,r}(u_{n-1})] + f_a\{[a_{n,l}(u_n)] + f_{on}(\max[h_{n-1,r}(u_{n-1}); h_{n,l}(u_n)])$, $n=\overline{2,N}$, где

$a_{n-1,r}(u_{n-1})$ и $a_{n,l}(u_n)$ – соответственно правая антенна на $(n-1)$-м интервале, входящая в состав ПАВП u_{n-1}, и левая антенна на n-м интервале, входящая в состав ПАВП u_n;

$f_a[a_{n-1,r}(u_{n-1})]$ и $f_a\{[a_{n,l}(u_n)]$ – соответственно стоимости антенн $a_{n-1,r}(u_{n-1})$ и $a_{n,l}(u_n)$;

$h_{n-1,r}(u_{n-1})$ и $h_{n,l}(u_n)$ – соответственно высоты подвеса правой антенны на $(n-1)$-м интервале, входящей в состав ПАВП u_{n-1}, и левой антенны на n-м интервале, входящей в состав ПАВП u_n;

$f_{on}(\max[h_{n-1,r}(u_{n-1}); h_{n,l}(u_n)])$ – стоимость n-й опоры, высота которой равна большей из высот $h_{n-1,r}(u_{n-1})$ и $h_{n,l}(u_n)$.

Подобным образом определяются стоимости крайних опор (1-й и $(N+1)$-й) и расположенных на них антенн:

$$f_1(u_1) = f_a[a_{1,l}(u_1)] + f_{o1}[h_{1,l}(u_1)];$$

$$f_{N+1}(u_N) = f_a[a_{N,r}(u_N)] + f_{oN+1}[h_{N,r}(u_N)].$$

Входящие в целевую функцию (1) величины суммарных стоимостей опор и находящихся на них антенн определяются типом используемых опор, зависимостью стоимости опор от их высоты, высотой подвеса находящихся на этих опорах антенн, а также стоимостью самих антенн, зависящей от их типа и диаметра.

В ограничениях (2)–(7):

$p(u_n^{(k)} / g(20\%))$ – относительный просвет на интервале n при 20%-й рефракции для ПАВП $u_n^{(k)}$;

$M_n(u_n^{(k)})$ – запас на замирания без учета СОН на интервале n для ПАВП $u_n^{(k)}$, дБ;

$M_{n.\min}(u_n^{(k)})$ – минимально допустимое значение запаса на замирания на интервале n для ПАВП $u_n^{(k)}$, дБ;

$\Delta M_n^{(k)}(u_{n-1}^{(k)};u_n^{(k)};u_{n+1}^{(k)})$ – деградация запаса на замирания на n-м интервале из-за влияния СОН от $(n-1)$-го и $(n+1)$-го интервалов [4];

$\Delta M_1^{(k)}(u_1^{(k)};u_2^{(k)})$ и $\Delta M_N^{(k)}(u_{N-1}^{(k)};u_N^{(k)})$ – соответственно деградация запаса на замирания на двух крайних интервалах: 1-м и N-м;

$\Omega^{(k)}(U)$ – подмножество множества $\Omega(U)$, на котором отыскивается k-е лучшее решение;

$\Omega^{(k-1)^*}(U)$ – множество, элементами которого являются $(k-1)$ ранжированных лучших УП-ПАВП $U^{(1)}, U^{(2)}, \ldots, U^{(k-1)}$.

Минимальный запас на замирания на интервале n для ПАВП $u_n^{(k)}$, дБ,

$$M_{n.\min}(u_n^{(k)}) = \max\left[\begin{matrix} M_{n.\min}(u_n^{(k)};UAR_S); \\ M_{n.\min}(SESR); M_{n.\min}(UAR_R) \end{matrix}\right], \quad (8)$$

где $M_{n.\min}(u_n^{(k)};UAR_S)$ – минимально допустимое значение запаса на замирания на n-м интервале с точки зрения критерия допустимости пары высот антенн при субрефракции, равное величине дифракционных потерь при субрефракции $LSD(u_n^{(k)})$ и зависящее от ПАВП $u_n^{(k)}$ и требований к субрефракционной составляющей показателя неустойчивости [4], дБ;

$M_{n.\min}(SESR)$ – минимально допустимое значение запаса на замирания на n-м интервале с точки зрения требований к показателю качества по ошибкам (показателю SESR) [5,6], дБ;

$M_{n.\min}(UAR_R)$ – минимально допустимое значение запаса на замирания на n-м интервале с точки зрения требований к показателю неготовности, учитывающему ослабление в дождях (показателю UAR_R) [7], дБ.

Решение задачи (1)–(7) целесообразно осуществлять в 3 этапа.

С целью уменьшения объема вычислений на первом этапе выполняются все вычисления, относящиеся к отдельным интервалам и не зависящие от других интервалов. При этом формируются множества $w_n(U)$ ПАВП u_n, $n = \overline{1,N}$, удовлетворяющих условиям (2), (3) и

$$M_n(u_n) \geq M_{n.\min}(u_n). \quad (9)$$

На втором этапе для всех пар соседних интервалов вычисляются дополнительные дифракционные потери на трассах распространения СОН для правых и левых антенн.

На третьем этапе решается собственно задача многовариантной оптимизации выбора ПАВП на многоинтервальной ЦРРЛ. С учетом сепарабельности ограничений (2)–(6) для решения рассматриваемой задачи дискретной оптимизации целесообразно использовать метод многовариантного динамического программирования (МВДП) [8].

Процедура дискретной оптимизации на основе МВДП имеет многошаговый характер, причем процесс условной оптимизации участков ЦРРЛ на отдельных шагах удобно рассматривать в обратном порядке, т.е. справа–налево.

Рассмотрим произвольный шаг процесса пошаговой оптимизации выбора ПАВП, на котором анализируется участок ЦРРЛ, содержащий интервалы: n, $(n+1)$, …, N.

Пусть в результате предшествующего шага условной оптимизации, на котором рассматривался участок, содержащий интервалы $(n+1)$, $(n+2)$, …, N, имеется множество $\Omega_{n+1,2}(U)$ УП-ПАВП $U_{n+1,2} = \{u_{n+1}; u_{n+2}\}$, каждой из которых соответствуют $K_{n+3} \leq K$ допустимых условно лучших УП-ПАВП:

$$U_{n+1}^{(k_{n+3})^*}(U_{n+1,2}) = \{U_{n+1,2}; U_{n+3}^{(k_{n+3})^*}(U_{n+1,2})\},$$

$$k_{n+3} = \overline{1,K_{n+3}}, \text{ где}$$

$$U_{n+3}^{(k_{n+3})^*}(U_{n+1,2}) = \{u_{n+3}^{(k_{n+3})^*}(U_{n+1,2});$$

$$u_{n+4}^{(k_{n+3})^*}(U_{n+1,2}); \ldots; u_N^{(k_{n+3})^*}(U_{n+1,2})\}.$$

Каждому $U_{n+3}^{(k_{n+3})^*}(U_{n+1,2})$ соответствуют условно минимальные суммарные затраты $F_{n+3}^{(k_{n+3})^*}(U_{n+1,2})$ на опоры O_{n+3}; O_{n+4}; …; O_{N+1} и антенны на интервалах $(n+3)$, $(n+4)$, …, N.

Выберем одну из УП-ПАВП $U_{n+1,2} \in \Omega_{n+1,2}(U)$ и рассмотрим ее продолжения $U_{n,3} = \{u_n; U_{n+1,2}\}$ для всех $u_n \in w_n(U)$. Для всех полученных таким образом УП $U_{n,3}$ вычисляются деградация запаса на замирания на $(n+1)$-м интервале $\Delta M_{n+1}(u_n, u_{n+1}, u_{n+2})$ и проверяется условие $M_{n+1}(u_{n+1}) \geq M_{n+1.\min}(u_{n+1}) + \Delta M_{n+1}(u_n; u_{n+1}; u_{n+2})$.

УП-ПАВП $U_{n,3}$, для которых указанное условие выполняется, образуют множество $\Omega_{n,3}(U)$, остальные $U_{n,3}$ исключаются из дальнейшего рассмотрения.

Аналогичные вычисления выполняются для всех других УП-ПАВП $U_{n+1,2} \in \Omega_{n+1,2}(U)$. В результате формируется множество допустимых УП-ПАВП $U_{n,3}$, каждой из которых соответствуют не более K условно лучших УП-ПАВП $U_n^{(k_{n+3})}(U_{n,3}) = \{u_n(U_{n,3}); U_{n+1}^{(k_{n+3})^*}(U_{n+1,2})\}$, $k_{n+3} \leq K$.

Далее фиксируется $U_{n,2}$ и для каждого u_{n+2} вычисляются не более K значений частной целевой функции, относящейся к участку ЦРРЛ, содержащему интервалы $(n+2)$, $(n+3)$, … , N:

$$F_{n+2}^{(k_{n+3})}[U_{n,2} / u_{n+2}(U_{n,2})] = f_{n+2}[u_{n+1}(U_{n,2}); u_{n+2}(U_{n,2})] + F_{n+3}^{(k_{n+3})^*}[u_{n+1}(U_{n,2}), u_{n+2}(U_{n,2})]. \quad (10)$$

Осуществляя в (10) минимизацию по всем парам $[k_{n+3}, u_{n+2}(U_{n,2})]$, запишем следующее функциональное уравнение МВДП, которое позволяет найти не более K условно минимальных значений суммарной стоимости антенн и антенных опор на рассматриваемом участке ЦРРЛ:

$$F_{n+2}^{(k_{n+2})^*}(U_{n,2}) = \min_{\Gamma^{(k_{n+2})}(U_{n,2})} \{f_{n+2}[u_{n+1}(U_{n,2}); u_{n+2}(U_{n,2})] + F_{n+3}^{(k_{n+3})^*}[u_{n+1}(U_{n,2}); u_{n+2}(u_{n,2})]\}, k_{n+2} \leq K, \quad (11)$$

где $\Gamma^{(k_{n+2})}(U_{n,2})$ – множество, на котором определяется k_{n+2}-я ранжированная условно лучшая УП-ПАВП:

$$U_{n+2}^{(k_{n+2})^*}(U_{n,2}) = \{u_{n+2}^{(k_{n+2})^*}(U_{n,2}); u_{n+3}^{(k_{n+2})^*}(U_{n,2}); \ldots; u_N^{(k_{n+2})^*}(U_{n,2})\}.$$

Элементами множества $\Gamma^{(k_{n+2})}(U_{n,2})$ являются пары $[k_{n+3}, u_{n+2}(U_{n,2})]$, при этом

$$\Gamma^{(1)}(U_{n,2}) = \Gamma(U_{n,2}); \quad (12)$$

$$\Gamma^{(k_{n+2})}(U_{n,2}) = \Gamma(U_{n,2}) / \Gamma^{(k_{n+2-1})^*}(U_{n,2}), k_{n+1} > 1. \quad (13)$$

В (12) и (13) $\Gamma(U_{n,2})$ – множество всех возможных пар $[k_{n+3}, u_{n+2}(U_{n,2})]$; $\Gamma^{(k_{n+2-1})^*}(U_{n,2})$ – множество $(k_{n+1}-1)$ ранжированных условно лучшых УП-ПАВП.

Аналогичным образом осуществляется условная оптимизация для других $U_{n,2}$, т.е. выполняется цикл по всем допустимым $U_{n,2}$.

Таким образом, в результате выполнения рассмотренного произвольного шага условной многовариантной оптимизации имеется множество $\Omega_{n,2}(U)$ пар $U_{n,2} = \{u_n, u_{n+1}\}$, каждой из которых соответствуют не более K допустимых ранжированных условно лучших УП-ПАВП $U_n^{(k_{n+2})^*}(U_{n,2}) = \{U_{n,2}; U_{n+2}^{(k_{n+2})^*}(U_{n,2})\}$, $k_{n+2} \leq K$. При этом каждой УП-ПАВП $U_n^{(k_{n+2})^*}(U_{n,2})$ соответствует свое условно минимальное значение суммарной стоимости антенных опор $O_{n+2}; O_{n+3}; \ldots; O_{N+1}$ и антенн на $(n+2)$-м, $(n+3)$-м, …, N-м интервалах $F_{n+2}^{(k_{n+2})^*}(U_{n,2})$.

Аналогично выполняются другие шаги условной оптимизации ПАВП.

На последнем шаге оптимизации вначале для каждой рассматриваемой $U_{1,3} = \{u_1; u_2; u_3\}$ проверяются условия $M_1(u_1) \geq M_{1.\min}(u_1) + M_1(u_1; u_2)$ и $M_2(u_2) \geq M_{2.\min}(u_2) + M_2(u_1; u_2; u_3)$.

Далее для каждого $U_{1,2} = \{u_1, u_2\}$, входящего в состав допустимых $U_{1,3}$, вычисляются ранжированные условно минимальные значения $F_1^{(k_1)^*}(U_{1,2}) = [u_1(U_{1,2}) + u_2(U_{1,2}) + F_3^{(k_1)^*}(U_{1,2})]$, $k_1 = \overline{1,K}$. После этого в результате минимизации $F_1^{(k_1)^*}(U_{1,2})$ по $U_{1,2}$ определяются заданное число безусловно лучших ранжированных УП-ПАВП для всей рассматриваемой ЦРРЛ $U^{(k)} = \{u_1^{(k)}, u_2^{(k)}, \ldots, u_N^{(k)}\}$, $k = \overline{1,K}$ и соответствующие им значения суммарной стоимости всех антенн и антенных опор $F(U^{(k)})$.

Объем вычислений при многовариантной оптимизации многоинтервальных ЦРРЛ с использованием метода МВДП определяется числом проверок допустимости ПАВП на интервалах, числом вычислений текущих значений целевой функции и числом сравнений этих значений при их ранжировании и отборе заданного числа наименьших значений.

Верхняя граница общего числа проверок допустимости ПАВП

$$\sup V_{\text{пд}} = (N-2)J^3 + 2J^2, \quad (14)$$

где J – максимальное число рассматриваемых вариантов ПАВП на интервале, $J = \max_n J_n$,

J_n – число вариантов ПАВП на интервале n, удовлетворяющих условию $M_n(u_n) \geq M_{n.\min}(u_n)$, $n = \overline{1,(N)}$.

Следует отметить независимость числа проверок допустимости ПАВП от числа отыскиваемых лучших решений K.

Верхняя граница общего числа вычислений текущих значений целевой функции определяется выражением:

$$\sup V_{\text{ВЦФ}} = J^3[K(N-3)+1]. \qquad (15)$$

На каждом шаге процесса условной оптимизации при фиксированном $U_{n.2}$ верхняя граница числа сравнений текущих значений целевой функции $\sup V_{\text{СЦФ.1}}$ определяется суммой первых K членов арифметической прогрессии, первый член которой равен $(J-1)$ на первом шаге и $(JK-1)$ на всех остальных шагах, а разность на всех шагах равна минус 1. При этом

$$\sup V_{\text{СЦФ.1}} = \frac{K(2J-K-1)}{2}$$

для первого шага оптимизации и

$$\sup V_{\text{СЦФ.1}} = \frac{K(2JK-K-1)}{2}$$

для всех остальных шагов.

Верхняя граница общего числа сравнений текущих значений целевой функции

$$\sup V_{\text{СЦФ}} = 1/2[K(2J-K-1)J^2 + K(2JK-K-1)J^2(N-3)].$$

При $K > 1$ и $J > 1$ можно полагать

$$\sup V_{\text{СЦФ}} \approx (N-3)K^2J^3. \qquad (16)$$

Сравнение (14), (15) и (16) позволяет утверждать, что при $K >> 1$ наибольший объем вычислений связан с ранжированием и отбором заданного числа ранжированных лучших решений. При этом увеличение объема вычислений примерно пропорционально квадрату числа отыскиваемых решений.

При решении рассматриваемой задачи многовариантной оптимизации методом простого перебора наибольший объем вычислений также связан с ранжированием и отбором заданного числа ранжированных лучших решений. При этом верхняя граница общего числа сравнений значений целевой функции

$$\sup V_{\text{СЦФ}} = \frac{1}{2}K(2J^N - K - 1) \approx KJ^N. \qquad (17)$$

Сравнение выражений (16) и (17) позволяет сделать вывод о том, что при решении рассматриваемой задачи совместной оптимизации на многоинтервальных ЦРРЛ метод МВДП обеспечивает большой выигрыш в объеме вычислений в сравнении с методом простого перебора, причем указанный выигрыш быстро возрастает с увеличением числа интервалов на ЦРРЛ.

Применение МВДП позволяет находить достаточно большое число ранжированных лучших решений рассмотренных задач многовариантной оптимизации в условиях большой протяженности ЦРРЛ и большой вариантности изменяемых характеристик интервалов.

СПИСОК ЛИТЕРАТУРЫ

1. **Данилович Д.А., Сиверс М.А., Зайцев С.П.** Оптимизация высот подвеса антенн на многоинтервальных цифровых РРЛ с учетом дополнительных неформальных ограничений // Электросвязь. – 2008. – № 7. – С. 36 – 38.

2. **Сартбаев Д.А., Данилович О.С., Шалгимбаев М.Ж.** Нахождение ранжированных лучших решений задачи оптимизации характеристик антенн на цифровых РРЛ с двухчастотными планам // Труды учебных заведений связи / СПбГУТ. – СПб, 2003. - № 169. – С. 225 – 234.

3. **Данилович О.С., Сартбаев Д.А., Гумбинас А.Ю.** Комплексная оптимизация выбора антенн и высот их подвеса на многоинтервальных цифровых радиорелейных линиях // Электросвязь. – 2003. – № 6. – С. 35 – 37.

4. Методика расчета трасс цифровых РРЛ прямой видимости в диапазоне частот 2 – 20 ГГц. – ЗАО "Инженерный центр". – М., 1998.

5. ITU-R Recommendation F.1668. Error performance objectives for real digital fixed wireless links used in 27 500 km hypothetical reference paths and connections. 2003.

6. ITU-R Recommendation P.530-11. Propagation data and prediction methods required for the design of terrestrial line-of-sight systems. 2005.

7. ITU-R Recommendation F.1703. Availability objectives for real digital fixed wireless links used in 27 500 km hypothetical reference paths and connections. 2005.

8. **Данилович О.С.** Оптимизация радиотехнических систем на основе многовариантного динамического программирования // Радиотехника. 1990. № 4. С. 13 – 17.

Бузюкова И.Л., Гайдамака Ю.В.

Исследование сигнальной нагрузки интеллектуальных сетей связи с узлами в разных часовых поясах

В процессе проектирования интеллектуальных сетей связи (ИСС) операторы связи сталкиваются с необходимостью расчета сетей общеканальной сигнализации ОКС №7 и определения объемов сигнального оборудования, требуемого для нормального функционирования сети. В России, где операторы постоянно расширяют свои сети и предоставляют услуги абонентам, находящимся в разных часовых поясах, решение этой задачи приобретает все большее значение.

Для расчета сетей ОКС№7 в первую очередь требуется оценить сигнальную нагрузку, возникающую в ходе предоставления интеллектуальных услуг. С точки зрения оборудования для обслуживания сетей ИСС федерального уровня требуется один или несколько узлов управления услугами SCP (Service Control Point) и узлы коммутации услуг SSP (Service Switching Point), расположенные в разных часовых поясах.

Целью данной статьи является анализ влияния конфигурации сети на требуемые объемы сигнального оборудования. Задача состоит в расчете сигнальной нагрузки на узлы ИСС, а также на звенья сигнализации (ЗС) между узлами ИСС при различных конфигурациях сети. Для анализа выбран фрагмент сети междугородного оператора связи, предоставляющего услуги ИСС абонентам различных регионов России. Вычисления проведены с помощью программного средства, позволяющего рассчитывать параметры сигнального трафика в ИСС, разработанного на кафедре систем телекоммуникаций Российского университета дружбы народов.

При обслуживании вызовов каждой услуги ИСС по сети ОКС №7 передаются потоки сигнальной информации, требующие значительных сетевых ресурсов и высокого качества обслуживания. В процессе обработки вызова интеллектуальной услуги, осуществляемой в режиме реального масштаба времени, участвуют следующие узлы:

– узел коммутации услуг SSP с интегрированной интеллектуальной периферией — выполняет функции управления вызовом, коммутации услуги, взаимодействия с пользователем;

– узел управления услугами SCP с интегрированным узлом базы данных услуг (SDP, Service Data Point) — выполняет функции управления услугами.

Взаимодействие между интеллектуальной надстройкой и базовой сетью связи осуществляется через ОКС №7 посредством стандартизованного протокола INAP (Intelligent Network Application Protocol) [1,2]. Сигнальные отношения протокола INAP устанавливаются между каждой парой узлов SSP и SCP/SDP.

Алгоритм расчета сигнальной нагрузки ИСС основан на методе расчета параметров сигнального трафика ИСС, предложенном в работах [3,4]. Этот метод был модифицирован для случая наличия в интеллектуальной сети нескольких узлов SCP. Ниже алгоритм расчета сигнальной нагрузки на узлы и ЗС ИСС при различных конфигурациях сети ИСС изложен на примере фрагмента междугородной сети связи, предоставляющей услуги интеллектуальной сети абонентам различных регионов (рис.1). В рассматриваемом примере узлы ИСС расположены в четырех часовых поясах — от GMT+03:00 до GMT+06:00 (GMT — Greenwich Mean Time). Вызовы интеллектуальных услуг поступают от восьми транзитных междугородных узлов связи (ТмгУС), пять из которых реализуют функции узла SSP. Сигнальная нагрузка от различных узлов SSP обслуживается одним узлом SCP, расположенном в часовом поясе GMT+03:00, сигнальные отношения между узлами ИСС реализованы в связанном режиме сигнализации.

В соответствии с модифицированным методом вычисления нагрузки ИСС расчет величины сигнальной нагрузки подсистемы INAP производится на основании исходных данных о нагрузочных параметрах сигнальной нагрузки между узлами SSP и SCP, объемах сигнальной информации, пересылаемых между узлами при предоставлении различных услуг, а также сведений о часах наивысшей нагрузки (ЧНН) для каждой услуги. Так как вызовы услуг от ТмгУС

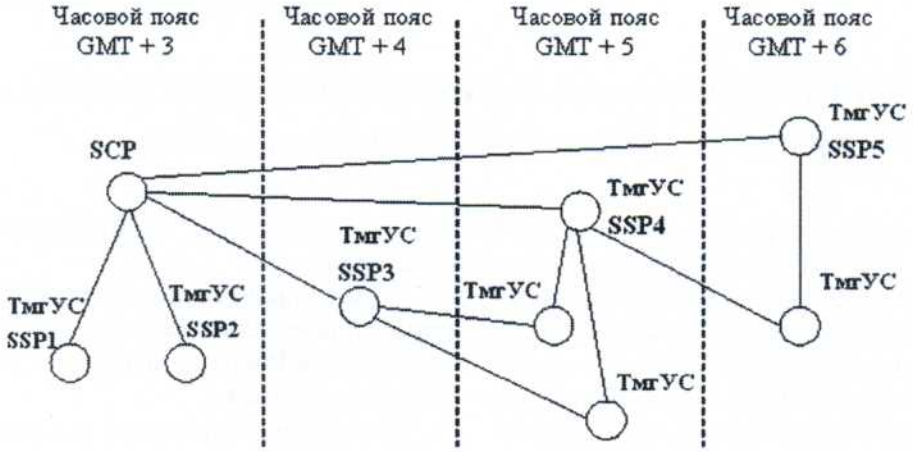


Рис. 1. Фрагмент интеллектуальной сети связи

могут быть обслужены различными узлами SSP, при расчетах учитывается распределение сигнальной нагрузки между сигнальными отношениями.

Приведем обозначения для основных параметров, используемых для расчета сигнальной нагрузки ИСС:

v_i — вершина графа сети, соответствующая узлу ИСС;

V_1 — множество вершин графа, соответствующих узлам, от которых поступают вызовы услуг интеллектуальной сети (узлам ТмгУС),

V_2 — множество вершин графа, соответствующих узлам SSP,

V_3 — множество вершин графа, соответствующих узлам SCP;

(v_j, v_l) — пара, соответствующая сигнальному отношению межу узлом SSP и узлом SCP, $v_j \in V_2$, $v_l \in V_3$;

M — количество услуг интеллектуальной сети;

$\lambda_m(v_i)$ — интенсивность вызовов услуги m в ЧНН этой услуги от узла сети, соответствующего вершине v_i;

L'_m, L''_m — объем сигнальной информации, передаваемой от узла SSP к узлу SCP и от узла SCP к узлу SSP при обслуживании одного вызова услуги m;

— доля вызовов услуги m, поступающих от узла сети, соответствующего вершине $v_i \in V_1$, на сигнальное отношение (v_j, v_l), $v_j \in V_2$, $v_l \in V_3$.

На первом шаге определяется сигнальная нагрузка, создаваемая вызовами каждой услуги m от каждого пункта сигнализации в ЧНН (в направлении от узла SSP к узлу SCP и наоборот). Данная сигнальная нагрузка для каждого узла сети v_i и каждой услуги m может быть вычислена в прямом направлении по формуле:

$$a'_m(v_i) = \frac{\lambda_m(v_i) \cdot L'_m}{8000 \cdot 3600}, \quad v_i \in V_1, \qquad (1)$$

в обратном направлении (от узла SCP к узлу SSP) по формуле:

$$a''_m(v_i) = \frac{\lambda_m(v_i) \cdot L''_m}{8000 \cdot 3600}, \quad v_i \in V_1. \qquad (2)$$

На втором шаге вычисляется сигнальная нагрузка, создаваемая вызовами каждой услуги от каждого пункта сигнализации для каждого часа по московскому времени в прямом и обратном направлениях ($a'_m(h, v_i)$ и ($a''_m(h, v_i)$ соответственно), при этом учитывается разница в часовых поясах. Кроме того, коэффициент $k_m(h, v_i)$ задает соотношение между интенсивностью вызовов услуги в ЧНН и интенсивностью вызовов этой услуги не в ЧНН.

На третьем шаге полученная сигнальная нагрузка, создаваемая вызовами всех услуг ИСС, суммируется отдельно по каждому пункту сигнализации для каждого часа по московскому времени. Таким образом, суммарная нагрузка, создаваемая всеми услугами от узла сети, соответствующего вершине $v_i (v_i \in V_1)$ графа G, в каждый час суток h в прямом и в обратном направлениях может быть представлена в следующем виде:

$$a'(h, v_i) = \sum_{m=1}^{M} a'_m(h, v_i), \quad v_i \in V_1, \qquad (3)$$

$$a''(h, v_i) = \sum_{m=1}^{M} a''_m(h, v_i), \quad v_i \in V_1. \qquad (4)$$

На четвертом шаге величина сигнальной нагрузки должна быть определена для каждого сигнального отношения интеллектуальной сети, т.е. для каждой пары узлов SSP и SCP. С учетом коэффициента $K_m(v_i, (v_j, v_l))$ распределения нагрузки между сигнальными отношениями сигнальная нагрузка на пучки ЗС между узлами SSP и SCP в каждый час суток h в прямом и обратном направлениях определяется выражениями:

$$a'(h,(v_j,v_l)) = \sum_{m=1}^{M} \sum_{v_i \in V_1} a'_m(h,v_i) \cdot K_m(v_i,(v_j,v_l)),\ v_j \in V_2,\ v_l \in V_3, \quad (5)$$

$$a''(h,(v_j,v_l)) = \sum_{m=1}^{M} \sum_{v_i \in V_1} a''_m(h,v_i) \cdot K_m(v_i,(v_j,v_l)),\ v_j \in V_2,\ v_l \in V_3. \quad (6)$$

На пятом шаге необходимо определить максимальную сигнальную нагрузку на каждое сигнальное отношение в прямом и в обратном направлениях, создаваемую в ЧНН вызовами всех услуг с учетом разницы в часовых поясах:

$$a'(v_j,v_l) = \max(h = \overline{0,23}) a'(h,(v_j,v_l)),\quad v_j \in V_2,\ v_l \in V_3, \quad (7)$$

$$a''(v_j,v_l) = \max(h = \overline{0,23}) a''(h,(v_j,v_l)),\quad v_j \in V_2,\ v_l \in V_3. \quad (8)$$

На последнем, шестом, шаге для каждого сигнального отношения рассчитывается минимальное число $N(v_j, v_l)$ звеньев сигнализации, необходимых для обслуживания максимальной сигнальной нагрузки, создаваемой на соответствующее сигнальное отношение в ЧНН вызовами всех услуг с учетом разницы в часовых поясах. Для этого на основе полученных данных для каждого сигнального отношения вычисляется значение целочисленной величины $N(v_j, v_l)$ по формуле:

$$N(v_j,v_l) = \left\lceil \frac{\max(a'(v_j,v_l), a''(v_j,v_l))}{0{,}2} \right\rceil,\quad v_j \in V_2,\ v_l \in V_3, \quad (9)$$

где $\lceil x \rceil = \begin{cases} x, x \in N, \\ [x]+1, x \notin N. \end{cases}$

Здесь $[x]$ — целая часть числа x, N — множество натуральных чисел. В формуле (9) учитывается, что ЗС функционирует в дуплексном режиме, а также требование ITU-T о том, что сигнальная нагрузка на каждое звено в условиях нормального функционирования сети не должна превосходить значения 0,2 Эрл [5].

Расчеты в соответствии с изложенным алгоритмом проведены с помощью программного средства, разработанного на кафедре систем телекоммуникаций Российского университета дружбы народов. Для анализа были выбраны данные, приближенные к реальным. На рассматриваемой сети предоставляется шесть услуг ИСС, относящихся к набору возможностей CS-1 (Capability Set), а именно — бесплатный вызов FPH (Freephone), вызов с дополнительной платой PRM (Premium Rate), универсальный номер доступа UAN (Universal Access Number), телеголосование VOT (Televoting), закрытая группа пользователей CUG (Closed User Group) и виртуальная карта связи VCC (Virtual Card Calling) [6]. Узлы сети, от которых поступают вызовы услуг ИСС, расположены в городах с населением более одного миллиона человек. В таблице 1 для каждого узла сети приведена интенсивность нагрузки, создаваемой вызовами услуг ИСС в ЧНН этой услуги. Кроме того, в таблице представлено множество часов наивысшей нагрузки для каждой услуги и объемы сигнальной информации, пересылаемой в прямом и обратном направлениях. Из исходных данных видно, что наибольшая интенсивность вызовов у услуги телеголосование (около 63% от суммарной нагрузки всеми услугами ИСС), а наименьшая интенсивность услуги виртуальная карта связи (менее 1% от суммарной нагрузки всеми услугами ИСС).

Задача исследования заключается в расчете сигнальной нагрузки на узлы ИСС, а также на ЗС между узлами ИСС при различных конфигурациях сети. Были рассмотрены несколько конфигураций сети, отличающиеся количеством и расположением узлов управления услугами, а также распределением нагрузки по сигнальным отношениям. Результаты расчетов позволяют оценить влияние изменения конфигурации интеллектуальной сети на объемы сигнального оборудования, требуемого для нормального функционирования сети. Ниже приведены результаты для трех наиболее интересных конфигураций.

В качестве конфигурации 1 был выбран фрагмент интеллектуальной сети связи феде-

Таблица 1

Исходные данные для услуг ИСС

Узел ИСС	FPH	PRM	UAN	CUG	VCC	VOT
$\lambda_m(v_1)$	22050	2520	2394	31500	50	83160
$\lambda_m(v_2)$	9618	1099	1044	13740	22	36274
$\lambda_m(v_3)$	2400	274	261	3429	5	9053
$\lambda_m(v_4)$	2747	314	298	3924	6	10359
$\lambda_m(v_5)$	2295	262	249	3279	5	8657
$\lambda_m(v_6)$	2163	247	235	3090	5	8158
$\lambda_m(v_7)$	2934	335	319	4191	7	11064
$\lambda_m(v_8)$	2392	273	260	3417	5	9021
ЧНН	11–16; 19; 20	19–23	9–18	11–16; 18–20	10–12; 20–22	19–23
L', байт	460	445	460	625	623	160
L'', байт	575	825	575	625	1215	310

рального уровня с одним узлом управления услугами, представленный на рис.1. В рассматриваемом примере узлы ИСС расположены в четырех часовых поясах, узел SCP находится в зоне GMT + 03:00, сигнальные отношения между узлами ИСС реализованы в связанном режиме сигнализации.

Результаты расчетов представлены на рис. 2. Тонкие марктрованные линии показывают распределение сигнальной нагрузки на пучки ЗС между узлами SSP и узлом SCP в течение суток (по московскому времени). Жирная линия на рис. 2 соответствует суммарной сигнальной нагрузке, представляющей собой нагрузку, которую должен обслужить узел SCP.

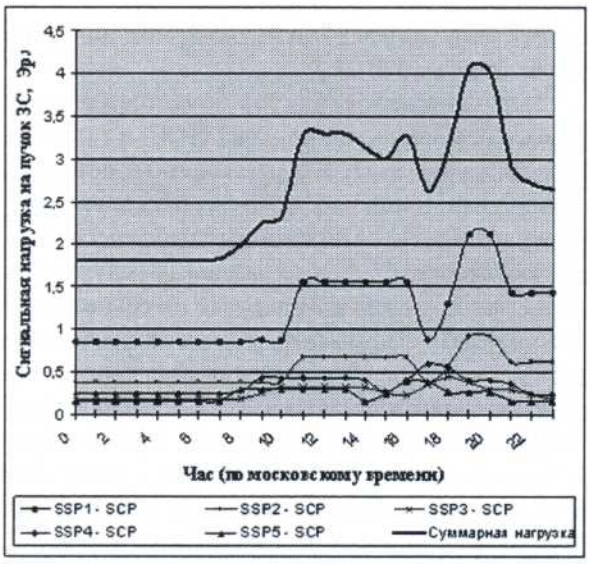


Рис. 2. Суточный профиль сигнальной нагрузки для конфигурации 1

На графиках видны всплески сигнальной нагрузки в периоды времени, соответствующие ЧНН. Двухпиковые профили нагрузки объясняются наличием дневного и вечернего периодов времени наивысшей нагрузки (ПНН) для различных услуг ИСС. Заметное увеличение сигнальной нагрузки в вечернее время обусловлено высокой интенсивностью потока вызовов на предоставление услуги телеголосование, доля которой в суммарной нагрузке, создаваемой вызовами на предоставление услуг ИСС, составляет примерно 2/3. Кроме того, на графике суммарной нагрузки заметна размытость ЧНН, которая вызвана разницей в часовых поясах.

Результаты по расчету максимального значения сигнальной нагрузки и минимального числа ЗС, необходимого для обслуживания сигнальной нагрузки на каждом сигнальном отношении, представлены в таблице 3. Полученные данные приведены для двух ярко выраженных ЧНН — 19:00 и 20:00.

Как видно из таблицы 2 максимальное значение нагрузки, которую должен обслужить узел SCP, соответствует 19:00 московского времени. Поскольку емкость одного пучка ЗС между каждой парой узлов ОКС №7 составляет не более 16 звеньев [5], нетрудно заметить, что для нормального функционирования сети при использовании конфигурации 1 для обслуживания нагрузки на каждом сигнальном отношении достаточно одного пучка ЗС.

В процессе исследований было рассмотрено несколько вариантов возможных конфигураций интеллектуальной сети и оценено влияние конфигурации сети на параметры сигнального

Таблица 2

Параметры сигнального трафика для конфигурации 1 (по московскому времени)

	Max значение нагрузки на 19:00, Эрл	Max значение нагрузки на 20:00, Эрл	Емкость пучка ЗС
Звено SSP1–SCP	2,111	2,112	11
Звено SSP2-SCP	0,920	0,921	5
Звено SSP3-SCP	0,375	0,302	3
Звено SSP4-SCP	0,403	0,402	3
Звено SSP5-SCP	0,269	0,269	2
SCP	4,078	4,006	

трафика ИСС. Вариант перемещения узла SCP из часового пояса GMT+03:00 в соседние часовые пояса не приводит к уменьшению объема сигнального оборудования, поскольку максимальная сигнальная нагрузка останется неизменной, смещается лишь суточный профиль нагрузки. Одним из вариантов оптимизации сети является введение дополнительного узла SCP как с целью возможного уменьшения объема необходимого сигнального оборудования, так и с целью повышения надежности сети. Для представленного на рис. 1 фрагмента сети ИСС дополнительный узел SCP можно разместить в часовых поясах GMT+05:00 или GMT+06:00, поскольку в этих поясах суммарная интенсивность поступающей сигнальной нагрузки выше, чем в часовом поясе GMT+04:00. В процессе исследований были рассмотрены следующие варианты внедрения дополнительного узла SCP, на который должна быть полностью перенаправлена нагрузка от некоторых узлов коммутации услуг SSP, и тем самым снижена суммарная нагрузка на первый узел SCP:

– размещение второго узла SCP в часовом поясе GMT+05:00 или GMT+06:00;

– обслуживание вторым узлом SCP нагрузки от узлов SSP, находящихся в часовых поясах GMT+04:00 - GMT+06:00 или GMT+05:00 и GMT+06:00;

– изменение распределения сигнальной нагрузки между сигнальными отношениями, а именно изменение доли вызовов, поступающих от ТмгУС на определенные узлы SSP.

В результате анализа параметров сигнального трафика ИСС для перечисленных выше конфигураций были получены близкие результаты. В статье они приведены на примере конфигурации 2, изображенной на рис. 3.

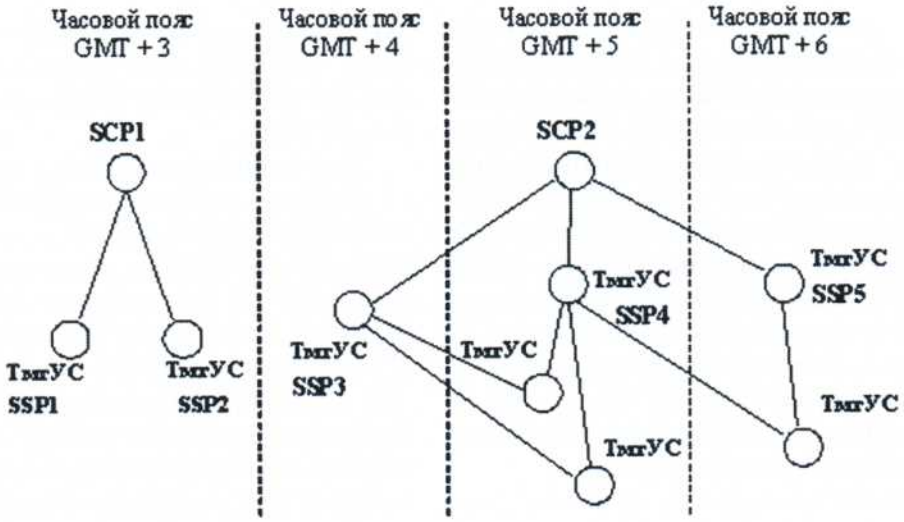


Рис. 3. Конфигурация 2 ИСС

По результатам расчета по-прежнему выявлены всплески сигнальной нагрузки в ЧНН, однако ПНН для SCP2 смещен на несколько часов. Профиль суммарной нагрузки на оба SCP по московскому времени совпадает с суточным профилем суммарной нагрузки (жирная линия на рис.2).

Результаты расчета максимального значения сигнальной нагрузки на каждое сигнальное отношение и на узлы SCP представлены в таблице 3. Результаты приведены для ЧНН SCP1 (20:00) и ЧНН SCP2 (17:00). Также в таблице представлены результаты по расчету емкости пучков ЗС между узлами SSP и SCP. Сравнение результатов на 20:00 по московскому времени таблиц 2 и 3 подтверждает вывод о том, что изменения общей нагрузки, возникающей в ходе обслуживания вызовов интеллектуальных услуг, не происходит, а происходит лишь ее перераспределение.

Таблица 3

Параметры сигнального трафика для конфигурации 2 (по московскому времени)

	Max значение нагрузки на 17:00, Эрл	Max значение нагрузки на 20:00, Эрл	Емкость пучка ЗС
Звено SSP1-SCP	0,885	2,112	11
Звено SSP2-SCP	0,386	0,921	5
SCP1	**1,271**	**3,033**	
Звено SSP3-SCP	0,354	0,302	3
Звено SSP4-SCP	0,591	0,402	3
Звено SSP5-SCP	0,396	0,269	2
SCP2	**1,341**	**0,973**	

С учетом потенциального увеличения уровня спроса на услуги ИСС для всех рассмотренных выше конфигураций были проведены дополнительные расчеты на основе измененных исходных данных. Интенсивность вызовов каждой интеллектуальной услуги для каждого из пунктов сигнализации была сначала увеличена на порядок, а затем на два порядка. По результатам вычислений были подтверждены результаты, полученные на первоначальных исходных данных: в результате введения дополнительного узла SCP уменьшение числа ЗС между узлами SSP и SCP не происходит даже при повышенном спросе на интеллектуальные услуги.

В заключение была рассмотрена конфигурация 3, схема которой приведена на рис. 4. Кроме SCP1, расположенного в часовом поясе GMT+03:00, в сети предусмотрен дополнительный SCP2, находящийся в часовом поясе GMT+06:00. Каждый из узлов SCP опирается на связанную пару узлов STP (Signalling Transfer Point) — транзитных узлов сигнализации. Несмотря на то, что конфигурация с двумя STP требует большего количества сигнального оборудования (узлов и звеньев сигнализации), именно эта конфигурация является оптимальной с точки зрения надежности сети в целом. Действительно, при отказе одного из узлов SCP или STP сигнальная нагрузка не блокируется, а будет перенаправлена по альтернативному маршруту на соответствующий узел. Кроме того, конфигурация 3 обладает свойством масштабируемости. Если сегодня для обслуживания нагрузок, возникающих при предоставлении услуг ИСС, достаточно одного узла SCP, который должен быть связан с обоими парными узлами STP, то при увеличении спроса на интеллектуальные услуги рассмотренная конфигурация позволяет легко добавлять дополнительные узлы SCP, каждый из которых следует связать с обоими узлами STP.

Выравнивание нагрузки по сигнальным отношениям может быть достигнуто за счет подбора коэффициентов $K_m(v_i, (v_j, v_l))$ для каждой услуги. Например, для случая сети с двумя узлами SCP и равномерным распределением нагрузки между ними все отличные от нуля $K_m(v_i, (v_j, v_l))$ будут равны 0,5. При этом в каждый час суток на каждый узел SCP поступает сигнальная нагрузка, равная половине суммарной сигнальной нагрузки (жирная линия на рис.2). Результаты расчетов для указанного случая равномерного распределения нагрузки между двумя узлами SCP при исходных данных таблицы 1 приведены в таблице 4.

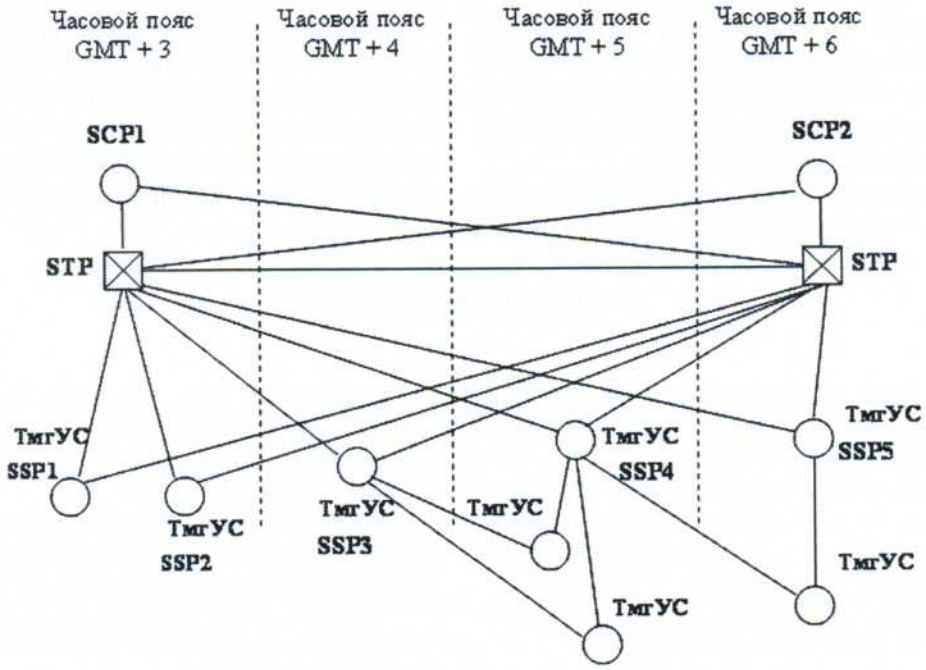


Рис. 4. Конфигурация 3 ИСС

В результате анализа влияния конфигурации сети на количество требуемых звеньев сигнализации выявлено, что перемещение узлов SCP в соседние часовые пояса или введение дополнительных узлов SCP не приводит к уменьшению емкости пучков ЗС на сигнальных отношениях. Это объясняется тем, что сигнальная нагрузка, обслуживаемая сетью, не уменьшается, а перераспределяется по сигнальным отношениям. Таким образом, можно сделать вывод о том, что введение дополнительного узла SCP с целью уменьшения числа звеньев сигнализации в сети нецелесообразно.

Отметим, что на сегодняшний день российские сети, обслуживающие абонентов в нескольких часовых поясах, построены на основе конфигурации с одним узлом SCP, что снижает надежность функционирования ИСС в целом. Оптимальным по всем критериям является решение с двумя узлами SCP, разнесенным по разным часовым поясам и связанным с узлами SSP посредством по крайней мере двух STP.

Таблица 4

Параметры сигнального трафика для конфигурации 3 (по московскому времени)

	Max значение нагрузки на 20:00, Эрл	Емкость пучка ЗС
Звено SSP1-STP1	1,115	6
Звено SSP2-STP1	0,486	3
Звено SSP3-STP1	0,224	2
Звено SSP4-STP1	0,313	2
Звено SSP5-STP1	0,115	1
SCP1	**2,253**	
Звено SSP1-STP2	1,115	6
Звено SSP2-STP2	0,486	3
Звено SSP3-STP2	0,224	2
Звено SSP4-STP2	0,313	2
Звено SSP5-STP2	0,115	1
SCP2	**2,253**	

Наличие дополнительного узла SCP позволит также решить проблемы перегрузок и резервирования в сети сигнализации. Эти и некоторые другие задачи, актуальные для российских операторов в настоящее время, являются предметом дальнейших исследований.

В заключение авторы выражают благодарность зав. кафедрой сетей связи СПбГУТ им. Бонч-Бруевича проф. Г.Г. Яновскому за плодотворные дискуссии, зав. кафедрой систем телекоммуникаций РУДН проф. К.Е. Самуйлову за полезные советы при определении задачи исследований и выпускнице кафедры систем телекоммуникаций РУДН О.А. Волковой, принимавшей участие в разработке программного средства.

СПИСОК ЛИТЕРАТУРЫ

1. ITU-T: White Book, Recommendation Q.1600. Signalling System No. 7 — Interaction between ISUP and INAP. Geneva, September, 1997.

2. **Бузюкова И.Л., Яновский Г.Г.** Методика расчета вероятностно-временных характеристик интеллектуальных сетей связи. Информационно-управляющие системы, №1 (2008). С. 17–23.

3. **Самуйлов К.Е.** Методы анализа и расчета сетей ОКС 7. Изд-во РУДН. Москва, 2002.

4. **Полищук В.П., Самуйлов К.Е., Чукарин А.В.** Применение инструментальных программных средств для планирования сетей ОКС 7, Электросвязь №12 (2005). С. 19–23.

5. ITU-T: White Book, Recommendation Q.706: Signalling System No.7 — Message Transfer Part Signalling Performance. Geneva, March, 1993.

6. ITU-T: White Book, Recommendation Q.1211: Introduction to intelligent network capability set 1. Geneva, March, 1993.

Максимов М.И., Сидорова Н.А.

Принципы работы и практическая реализация речевого кодека на основе метода линейного предсказания с возбуждением от кода

В настоящее время широкое распространение получили системы связи с кодовым разделением каналов, базовым элементом которых являются речевые кодеры, работающие на низких скоростях [1-5 и др.]. Одним из распространенных стандартов данных систем является Американский стандарт IS-95 [4], разработанный фирмой Qualcomm. IS-95 обеспечивает высокое качество кодирования речевых сигналов, используя стандарты IS-96-A [6] и IS-733 [7]. В этой связи одной из актуальных, является задача разработки и практической реализации алгоритмов работы вокодеров, совместимых с данными стандартами. Ниже на основе стандартов IS-96-A и IS-733 рассмотрены принципы построения, функциональные блоки и аппаратная реализация на цифровом сигнальном процессоре (ЦСП) вокодера, использующего линейное предсказание с возбуждением от кода (QCELP-вокодера). Приведены количественные характеристики качества его функционирования.

QCELP-вокодер состоит из двух независимых функциональных узлов: кодирования речевых выборок и декодирования параметров сигнала. Кодирование речевых выборок, т.е. определение параметров входного речевого сообщения, выполняется с помощью методов линейного предсказания [1] и процедуры анализа через синтез (Analysis by Synthesis) [5]. Для этого узел кодирования содержит три блока: блок линейного предсказания, блок анализа речевого сигнала и блок синтеза сигнала. Структурные схемы двух последних блоков представлены на рис. 1 и рис. 2 соответственно. Узел декодирования синтезирует речевой сигнал из принятых параметров сигнала. Структурная схема узла декодирования показана на рис. 3.

Рассмотрим алгоритм работы узла кодирования QCELP-вокодера. Поступающее на вход вокодера речевое сообщение $x(n)$ в стандартном ИКМ формате 13 бит с частотой дискретизации $F_д$ = 8000 Гц подается в блок сегментации. Сегментация выполняется путем деления выборок на неперекрывающиеся кадры (фреймы) длительностью 20 мс (160 выборок). Сегментированные фреймы подаются на блок удаления постоянной составляющей сигнала и блок задержки сигнала на 60 отсчетов (рис. 1).

Для компенсации смещения по постоянной составляющей входного сигнала исследовались два метода. В первом случае сигнал $x(n)$ подавался на низкочастотный фильтр с характеристикой

$$Y(z) = 0{,}75\,Y(z^{-1}) + 0{,}25X(z).$$

Оценка величины постоянной составляющей (DC) текущего фрейма определялась с помощью деления суммы всех 160 отсчетов профильтрованного сигнала $y(n)$ на 160. Компенсация постоянной составляющей на текущем фрейме осуществлялась путем вычитания из сигнала $x(n)$ полученной таким образом оценки величины DC.

Во втором случае, который оказался более предпочтительным, входной сигнал $x(n)$ подавался на низкочастотный фильтр с характеристикой $Y(z) = 0{,}94615[X(z) - 2X(z^{-1}) + X(z^{-1})] + 1{,}88\,Y(z^{-1}) - 0{,}8836\,Y(z^{-2})$.

При этом постоянная составляющая сразу устранялась более чем на 80 дБ.

Для минимизации нежелательных "концевых" эффектов [1,8] сегментированный речевой сигнал без постоянной составляющей поступает на блок предыскажения. Предыскажения осуществляются путем умножения сигнала текущего фрейма на оконную функцию (или окно), в качестве которой выбрано окно Хемминга:

$$x(n) = x(n)\,h(n),\ n = 0, 1, \ldots, 159,$$

где $h(n) = 0{,}54 - 0{,}46\cos(2\pi n/159)$.

Алгоритм вычисления коэффициентов краткосрочного предсказания использует ав-

токорреляционный метод [1, 8]. Для каждого фрейма вычисляются значения автокорреляционной функции:

$$R(k)=\sum_{k=0}^{159-k} x(i)\,x(i-k),$$

где k = 0, 1, ..., 10 для стандарта IS-96-A и k = = 0, 1, ..., 16 для стандарта IS-733. Далее с использованием алгоритма Дарбина из 11 значений $R(k)$ вычисляются 10 коэффициентов краткосрочного или линейного предсказания (КЛП) a_i, i = 0, 1, ..., 10, которые являются коэффициентами фильтра кратковременного предиктора (синтезирующего фильтра) с системной функцией $H(z)$ вида [1]:

$$H(z) = 1/A(z),\ A(z)=1-\sum_{i=1}^{10} a_i z^{-i}. \qquad (1)$$

При разработке системы, работающей на небольшой скорости передачи данных, одним из самых критичных факторов является квантование и кодирование параметров. Как известно

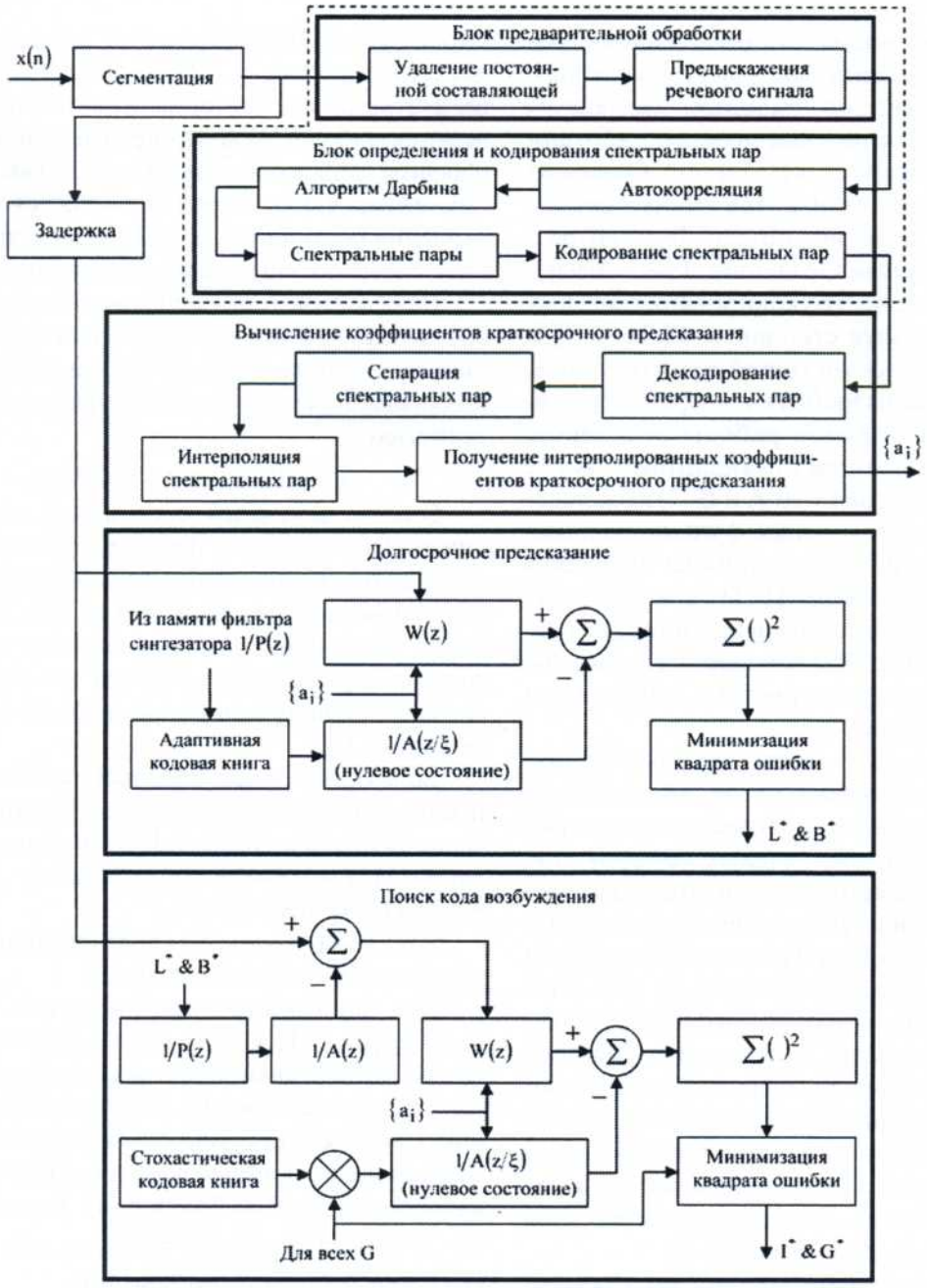


Рис. 1



Рис. 2

[5–7 и др.], КЛП обладают широким динамическим диапазоном. В этой связи непосредственное квантование КЛП достаточно сильно сказывается на изменении амплитудно-частотной характеристики (АЧХ) синтезирующего фильтра и его устойчивости. Для устранения этой нестабильности вместо КЛП предлагается использовать так называемые линейные спектральные пары (ЛСП) [2, 3], которые связаны с КЛП взаимно однозначным преобразованием и являются более устойчивыми к нежелательным эффектам квантования и ошибкам в канале. С этой целью формируются два полинома:

$$P_A(z) = A(z) + z^{-11}A(z^{-1}) = 1 + p_0z^{-1} + \ldots + p_4z^{-5} + p_4z^{-6} + \ldots + p_0z^{-10} + z^{-11}, \quad (2)$$

$$Q_A(z) = A(z) - z^{-11}A(z^{-1}) = 1 + q_0z^{-1} + \ldots + q_4z^{-5} - q_4z^{-6} - \ldots - q_0z^{-10} - z^{-11}, \quad (3)$$

где $A(z)$ определяется из (1), $pi = -a_i - a_{11-i}$, $qi = -a_i + a_{11-i}$, $1 \le i \le 10$. Далее для случая $z = \exp(j\omega/F_д)$ находятся корни ω_i, i = 1, 3, 5, 7, 9 полинома $P_A(z)$ и корни ω_i, i = 2, 4, 6, 8, 10 полинома $Q_A(z)$. Частоты ω_{2i-1}, i = 1,5 образуют ЛСП, между которыми выполняются соотношения [6, 7]:

$$0 < \omega_i < \pi,\ \omega_i < \omega_{i+1}. \quad (4)$$

Отметим особенности стандарта IS-96-A [6]. Вследствие свойства упорядоченности спектральных пар частотные разности имеют гораздо меньший диапазон по сравнению с самими ЛСП. Это позволяет квантовать разности $\omega_{i+1} - \omega_i$ всего 4 битами. Квантованные разности подаются в блок декодирования спектральных пар. Ошибки квантования и передачи через радиотракт кодированных спектральных пар могут привести к неустойчивости синтезирующего фильтра. Для устранения этого эффекта применяется процедура сепарации ЛСП [2,6], которая разносит спектральные пары друг от друга на расстояние не менее 80 Гц.

Частотная характеристика речи плавно меняется в течение фрейма [2]. Поэтому для уменьшения скачкообразности изменения значений спектральных пар фрейм речевого сигнала делится на 4 субфрейма, после чего для каждого субфрейма вычисляются соответствующие ему спектральные пары с помощью процедуры интерполяции:

$\omega_i\{1\} = 0{,}75\omega_{i(\text{предыд})} + 0{,}25\omega_{i(\text{текущ})}$
для 1-го субфрейма;

$\omega_i\{2\} = 0{,}5\omega_{i(\text{предыд})} + 0{,}5_{\ (\text{текущ})}$
для 2-го субфрейма;

$\omega_i\{3\} = 0{,}25\omega_{i(\text{предыд})} + 0{,}75\omega_{i(\text{текущ})}$
для 3-го субфрейма;

$\omega_i\{1\} = 0\omega_{i(\text{текущ})}$ для 4-го субфрейма.

Здесь $\omega_{i(\text{предыд})}$ и $\omega_{i(\text{текущ})}$ – соответственно ЛСП предыдущего и текущего фреймов. Интерполированные спектральные пары преобразуются в интерполированные КЛП с помощью процедуры, обратной процедуре нахождения корней уравнений (1), (2) [6].

При реализации стандарта IS-733 применяются следующие вычислительные операции [7]. Первоначально рассчитывается чувствительность спектральных пар SW_i, $i = \overline{1, 10}$. Затем с помощью полученных значений SW_i и пяти таблиц закодированных разностей последовательных ЛСП по критерию минимума ошибки между вычисленными и кодированными ЛСП определяется наилучший вектор, соответствующий полученным спектральным парам. Вектор вычисляется путем перебора всех элементов таблиц и нахождения наилучших позиционных значений, после чего кодируется 32 битами. В блоке синтеза кодирующего узла по 32 битам находятся ЛСП и далее производятся процедуры, аналогичные определяемым стандартом IS-96-A.

Входной сигнал $x(n)$, прошедший блок задержки, поступает в блок долгосрочного предсказания, где для каждого субфрейма определяются параметры основного тона: период основного тона L и коэффициент усиления B.

Определение этих параметров происходит с помощью поиска в адаптивной кодовой книге, заполнение которой происходит с выхода фильтра долговременного предиктора в синтезаторе речевого сигнала (рис. 3). Длина книги составляет 168 элементов. Поиск проводится для всех целых L от 17 до 143 и B от 0 до 2 через 0,25 для стандарта IS-96-A и всех L от 17 до 143 через 0,5 и B от 0 до 2 через 0,25 для стандарта IS-733. Интерполяция значений $p(L)$ адаптивной кодовой книги для дробных L может производиться, например, с помощью интерполяционного фильтра [7]:

$$p(L)=\sum_{i=-4}^{3} hamm\sin(i+0{,}5)\,p(L+i+0{,}5),$$

$$hamm\sin(x)=\frac{\sin(\pi x)}{\pi x}\left[0{,}5+0{,}4\cos\left(\frac{\pi x}{4}\right)\right].$$

Сорок последовательных значений кодовой книги, взятых с задержкой L и умноженных на коэффициент усиления B, проходят через фильтр с характеристикой $A(z/\xi)$, где ξ = 0,8, и вычитаются из сорока значений текущего субфрейма, пропущенных через персептуальный взвешивающий фильтр с характеристикой

$$W(z)=\frac{A(z)}{A(z/\xi)}=\frac{1-\sum_{i=1}^{10}a_i z^{-i}}{1-\sum_{i=1}^{10}a_i \xi^i z^{-i}}. \qquad (5)$$

Персептуальный взвешивающий фильтр применяется для уменьшения эффектов квантования и улучшения восприятия синтезированной речи [2, 6, 7]. Далее для всех значений L и B находится энергия разностного сигнала, минимальное значение которой определяет оптимальные величины L^* и B^*. Полученные для текущего субфрейма оценки периода основного тона L^* и коэффициента усиления B^* используются для определения характеристики фильтра долговременного предиктора

$$\frac{1}{P(z)}=\frac{1}{1-Bz^{-L}},$$

входным сигналом которого являются соответствующие значения адаптивной кодовой книги. С выхода долговременного предиктора сигнал поступает в фильтр кратковременного предиктора с характеристикой $1/A(z)$ (1). Восстановленный сигнал вычитается из значений сигнала текущего субфрейма. Полученный разностный сигнал проходит через персептуальный взвешивающий фильтр $W(z)$ (5), образуя остаточный сигнал.

Далее текущий субфрейм входного сигнала $x(n)$ делится на два (при использовании стандарта IS-96-A) или четыре (при использовании стандарта IS-733) подсубфрейма и поступает в блок поиска кода возбуждения, где для каждого подсубфрейма определяются параметры кода возбуждения, индекс стохастической кодовой книги I и коэффициент усиления G [6, 7].

Определение параметров кода возбуждения происходит с помощью поиска в циклической кодовой книге. Длина книги составляет 128 элементов. Поиск проводится для всех значений I от 0 до 127 и значений G от –6 дБ до 66 дБ. Кодирование коэффициента усиления G проводится с помощью трех бит в дифференциальном

Таблица 1

Распределение информационных бит внутри речевого фрейма по стандарту IS-96-A

Десять кодированных спектральных пар	Четыре периода основного тона L	Четыре коэффициента усиления B	Восемь индексов кодового возбуждения I	Восемь коэффициентов усиления G
40	28	12	56	24

Таблица 2

Распределение информационных бит внутри речевого фрейма по стандарту IS-733

Десять кодированных спектральных пар	Четыре периода основного тона L	Четыре коэффициента усиления B	Шестнадцать индексов кодового возбуждения I	Восемь коэффициентов усиления G
32	32	12	164	24

предсказывающем кодере. Первые два бита определяют изменение величины G текущего подсубфрейма от величины G предыдущего подсубфрейма на –4 дБ, 0 дБ, 4 дБ и 8 дБ, а третий бит определяет знак коэффициента усиления.

Двадцать последовательных значений кодовой книги, взятых с задержкой I и умноженных на коэффициент G, проходят через фильтр с характеристикой $A(z/\xi)$ (1) при $\xi = 0{,}8$ и вычитаются из двадцати (десяти) значений остаточного сигнала. Для всех значений I и G находится энергия разностного сигнала, минимальное значение которой соответствует оптимальным значениям L^* и G^*.

Распределение информационных бит одного кодированного речевого фрейма CELP-вокодера по стандартам IS-96-A и IS-733 представлено соответственно в таблицах 1 и 2.

Таким образом, вокодер преобразует речевой фрейм в пакет из 160 бит кодированной информации при использовании стандарта IS-96-A или в пакет из 264 бит при использовании стандарта IS-733. Соответственно выходная скорость речевого кодера составляет 8000 бит/с (для стандарта IS-96-A) или 13200 бит/с (для стандарта IS-733).

Синтез сигнала выполняется с использованием оптимальных параметров, полученных в кодере. Структурная схема синтеза сигнала представлена на рис. 3. Большая часть функциональных блоков декодера совпадает с блоками, описанными в кодере. Новыми функциональными блоками являются постфильтр и блок контроля усиления (рис. 3).

Адаптивный постфильтр используется для повышения качества синтезированной речи и обладает следующей характеристикой:

$$PF(z) = B(z)\frac{A(z/p)}{A(z/s)},$$

где $A(z)$ определяется из (3), $p = 0{,}5$, $s = 0{,}8$, а

$$B(x) = (1 - \gamma z^{-1})/(1 + \gamma z^{-1}). \qquad (6)$$

Параметр γ в (6) представляет собой функцию от среднего значения $M(\omega_i)$ десяти интерполированных линейных пар и может быть рассчитан как [6,7]

$$\gamma = \begin{cases} 0{,}25, & M(\omega_i) \le 0{,}24; \\ -25\left[M(\omega_i) - 0{,}25\right], & 0{,}24 < M(\omega_i) < 0{,}26; \\ -0{,}25, & M(\omega_i) \ge 0{,}26. \end{cases}$$

Распределение вычислительных затрат реализации вокодера по стандартам IS-96-A и IS-733 на ЦСП с фиксированной точкой TMS320VC5410A представлено в таблицах 3 и 4 соответственно.

Аппаратная реализация вокодера при использовании какого-либо из рассматриваемых стандартов показана на рис. 4.

Оценка качества синтезированной речи проводилась по отношению сигнал/шум (ОСШ) для синтезированного сигнала. С этой целью входное речевое сообщение делилось на N сегментов (N варьировалось от 200 до 2000). Длина каждого сегмента принималась равной длине субфрейма (40 выборок). Среднее ОСШ вычислялось по следующей формуле:

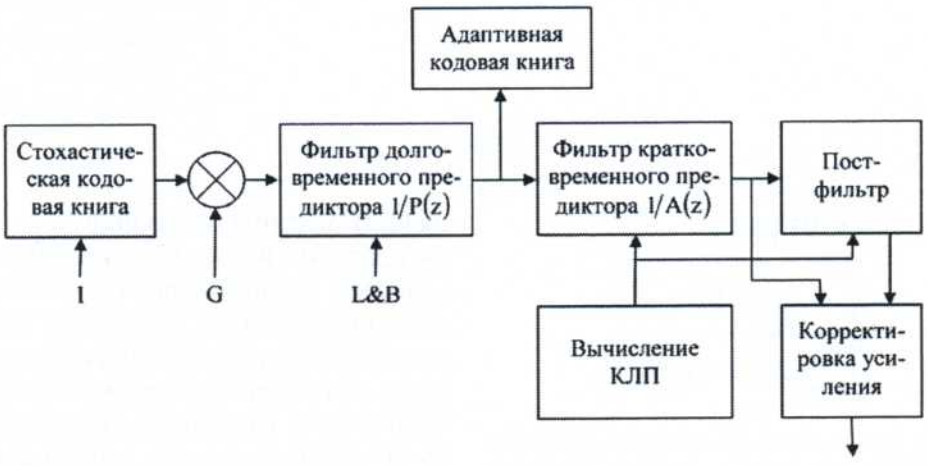


Рис. 3

Таблица 3

Распределение вычислительных затрат вокодера, синтезированного по стандарту IS-96-A

Функциональный блок	Максимальное количество требуемых циклов процессора, MIPS	Среднее количество требуемых циклов процессора, MIPS	Необходимая память	
			программ, килослов	данных, килослов
Кодер QCELP-вокодера	29	25	3100	4300
Декодер QCELP-вокодера	3	3	800	500
Суммарные затраты	32	28	3900	4800

Таблица 4

Распределение вычислительных затрат вокодера, синтезированного по стандарту IS-733

Функциональный блок	Максимальное количество требуемых циклов процессора, MIPS	Среднее количество требуемых циклов процессора, MIPS	Необходимая память	
			программ, килослов	данных, килослов
Кодер QCELP-вокодера	35	30	4700	6500
Декодер QCELP-вокодера	4	4	1200	800
Суммарные затраты	39	34	5900	7200

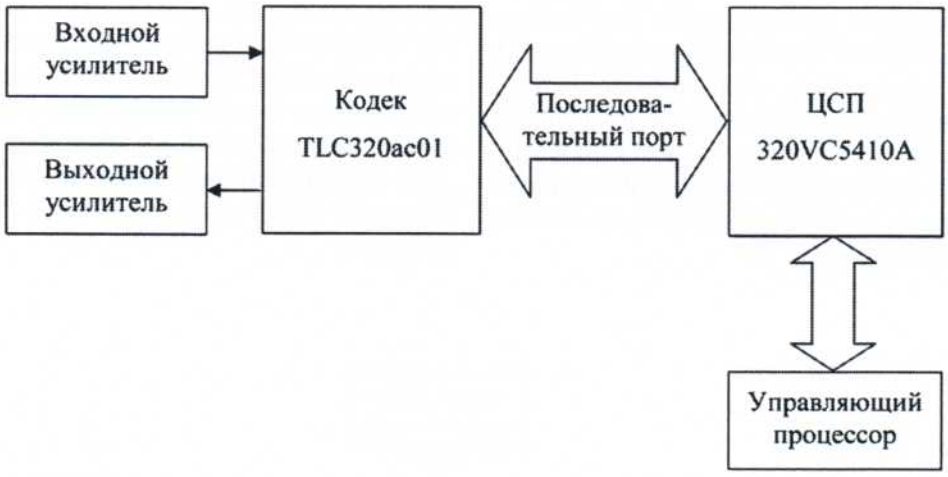


Рис. 4

$$\text{ОСШ } [\partial\text{Б}] = \frac{1}{N}\sum_{i=0}^{N-1} 10\lg\left[\frac{\sum_{n=0}^{39} s_i^2(n)}{\sum_{n=0}^{39}\left(s_i(n)-\acute{s}_i(n)\right)^2}\right],$$

где N – длина речевого сообщения в субфреймах, $s_i(n)$ – выборки входного речевого сигнала, $\hat{s}_i(n)$– выборки синтезированного речевого сигнала. Измерения проводились на 100 записанных речевых файлах (55 мужских и 45 женских голосов). Все полученные значения ОСШ при этом находились в диапазоне 15–25 дБ (для вокодера, синтезированного по стандарту IS-96-A), что характеризует качество синтезированной речи как близкое к коммерческому, и 20–30 дБ (для вокодера, синтезированного по стандарту IS-733), что характеризует качество синтезированной речи как коммерческое [9].

Таким образом, рассмотренные вокодеры, использующие линейное предсказание с возбуждением от кода и совместимые со стандартами IS-96-A и IS-733, могут быть рекомендованы к применению в системах связи общего и специального назначения, обеспечивающих пропускную способность 8000–13200 бит/с.

СПИСОК ЛИТЕРАТУРЫ

1. **Рабинер Л.Р., Шафер Р.В.** Цифровая обработка речевых сигналов: Пер. с англ. – М.: Радио и связь, 1981. – 593 с.

2. **Коротаев Г.А.** Эффективный алгоритм кодирования речевого сигнала на скорости 4.8 кбит/с и ниже // Зарубежная радиоэлектроника. – 1996. – № 3. – С. 52–68.

3. **Максимов М.И., Сидорова Н.А., Черноярова О.В.** Проектирование низкоскоростных речепреобразующих устройств для каналов с высоким процентом ошибок // Электросвязь. – 2008. – №7. – С. 48–50.

4. **Невдяев Л.М.** CDMA: IS-95 // Сети. – 2000. – №3. – С.6-8.

5. **A.S. Spanias** "Speech coding: A tutorial review" // Proc. of the IEEE, Vol. 82, No. 10, October 1994, pp. 1541–1582.

6. TIA/EIA/IS-96-A Speech Service Option Standard for Wideband Spread Spectrum Digital Cellular System / Telecommunications Industry Association, 1995. – 96 p.

7. TIA/EIA/IS-733 High Rate Speech Service Option 17 for Wideband Spread Communication System / Telecommunications Industry Association, 1998. – 124 p.

8. **Маркел Дж., Грэй А.Х.** Линейное предсказание речи. – М.: Связь, 1980. – 248 с.

9. **Назаров М.В., Прохоров Ю.Н.** Методы цифровой обработки и передачи речевых сигналов. – М.: Радио и связь, 1985. – 176 с.

Кукунин Д. С.

Модель системы с адаптивным выбором кода

Существует огромное количество способов оценки качества канала, для которых разработаны схемы, содержащие оптимальные алгоритмы контроля статистических характеристик. Логические схемы предполагают использование наряду с блоками измерения также блоки принятия решения. В качестве решения может выступать запрос к передающей стороне о повторе поврежденного блока, удаление блока, а также более сложные реакции системы на состояние канала. Ведущиеся в настоящее время многочисленные поиски решения задачи адаптации системы передачи к состоянию канала подчеркивают актуальность проблемы эффективного распределения ресурсов системы. Построение адаптивной системы должно способствовать повышению скорости передачи данных по каналу и оптимальному расходованию ресурсов памяти с максимально возможным при этом уровнем достоверности, достигающейся за счет применения предложенного в работе метода декодирования двоичных циклических кодов с использованием двойственного базиса поля Галуа [1].

Применение метода декодирования двоичных циклических (n,m) кодов с использованием двойственного базиса [1] обеспечивает высокий уровень достоверности передачи данных, приближаясь по данной характеристике к существующим методам декодирования, а в некоторых случаях превосходит их. Дополнительным преимуществом использования данного метода является анализ структуры "леса", по результатам которого определяется диапазон вероятности ошибки в канале [2].

Оценка канала при помощи рекуррентных последовательностей, обеспечивает требуемую точность определения $p_{ош}$ и может быть использована при построении системы с адаптивным кодированием. В качестве базового метода для такой системы выберем метод с непрерывным накоплением элементов, обеспечивающий

наилучшие результаты оценки канала по сравнению с остальными. При этом кодирующее устройство данной системы будет осуществлять переключение режима кодирования между кодами БЧХ (15,4), (15,6) и (15,11) [3].

Для построения имитационной системы, соответствующей таким требованиям, воспользуемся функциями калькулятора Галуа [4], обеспечивающими управление системой и мониторинг ведущейся статистики в режиме реального времени.

Таким образом, структура программной модели адаптивной системы состоит из компонент, разработанных в среде Microsoft Visual C++ 6.0 с применением библиотек MFC и включает следующие модули [3]:

- система, содержащая приемник, передатчик, кодирующее и декодирующее устройство для кодов (15,4), (15,6) и (15,11), а также генератор ошибок стационарного двоичного симметричного канала;
- контроллер, обеспечивающий регулирование параметров системы: критерий оценки и вероятность ошибки в канале;
- монитор, на который поступает информация о состоянии системы в реальном режиме времени.

Рассмотрим схему взаимодействия модулей данной системы (рис. 1).

Как видно на схеме (рис. 1), в задачи передатчика входит формирование кодовой комбинации в соответствии с выбранным кодом. Приемник в свою очередь производит декодирование принятых комбинаций с использованием двойственного базиса поля Галуа и определяет диапазон ($p_{0\min}$; $p_{0\max}$).

Задача элементов управления в *Контроллере* сводится к воздействию на генератор ошибок системы при помощи задания $p_{ош}$ в канале. Здесь определяется критерий оценки: вероятность правильного P_c или ложного P_e декодирования. Также задается параметр NA, определяющий длительность стационарного состояния системы, то есть количество принятых последовательностей N, свыше которого начинается проверка соответствия критерию оценки результатов работы используемого метода и становится возможным переход на более подходящий код.

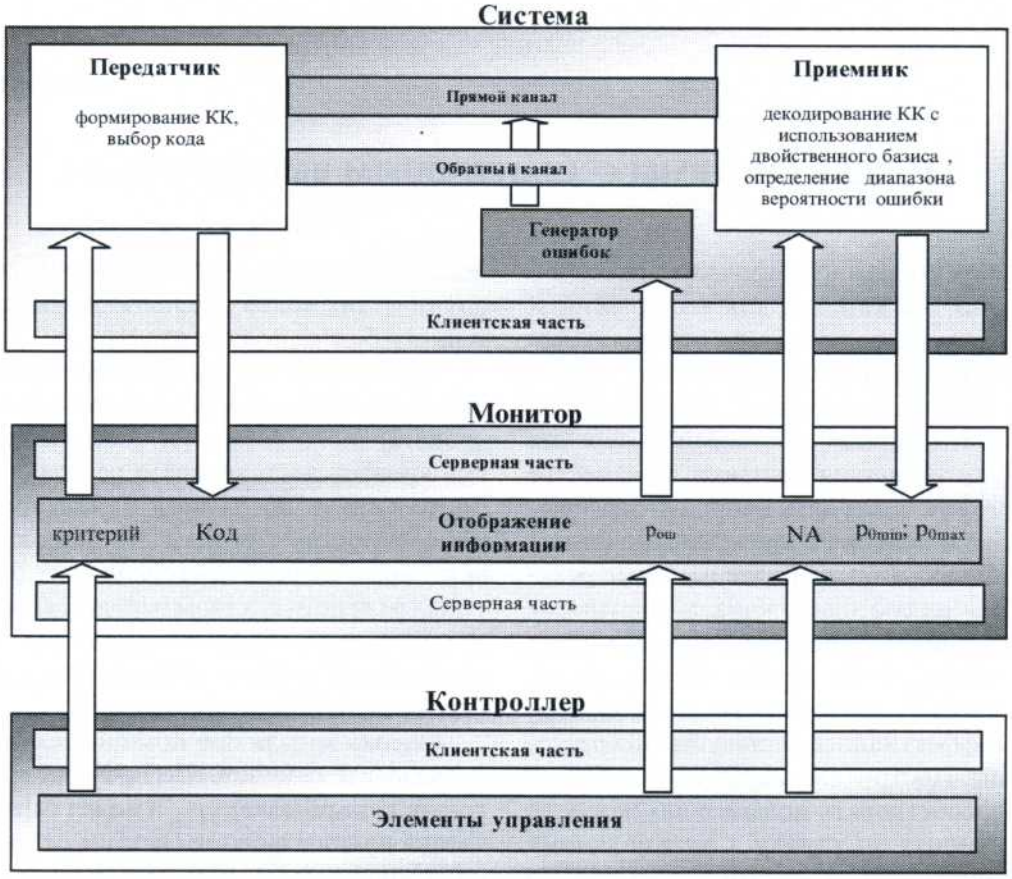


Рис. 1. Структурная схема взаимодействия модулей имитационной модели системы с адаптивным кодированием

Монитор, предназначенный для отображения статистической и управляющей информации, является связующим звеном между *системой* и *контроллером* и обеспечивает взаимодействие между ними при помощи сокетов. После запуска всех модулей начинается обработка принимаемых последовательностей с заданными по умолчанию параметрами: $p_{ош} = 0$; $P_c = 1$; $NA = 1000$. При этом используется самый быстрый код (15,11).

Рассмотрим алгоритм работы данной системы с критерием P_c (рис. 2).

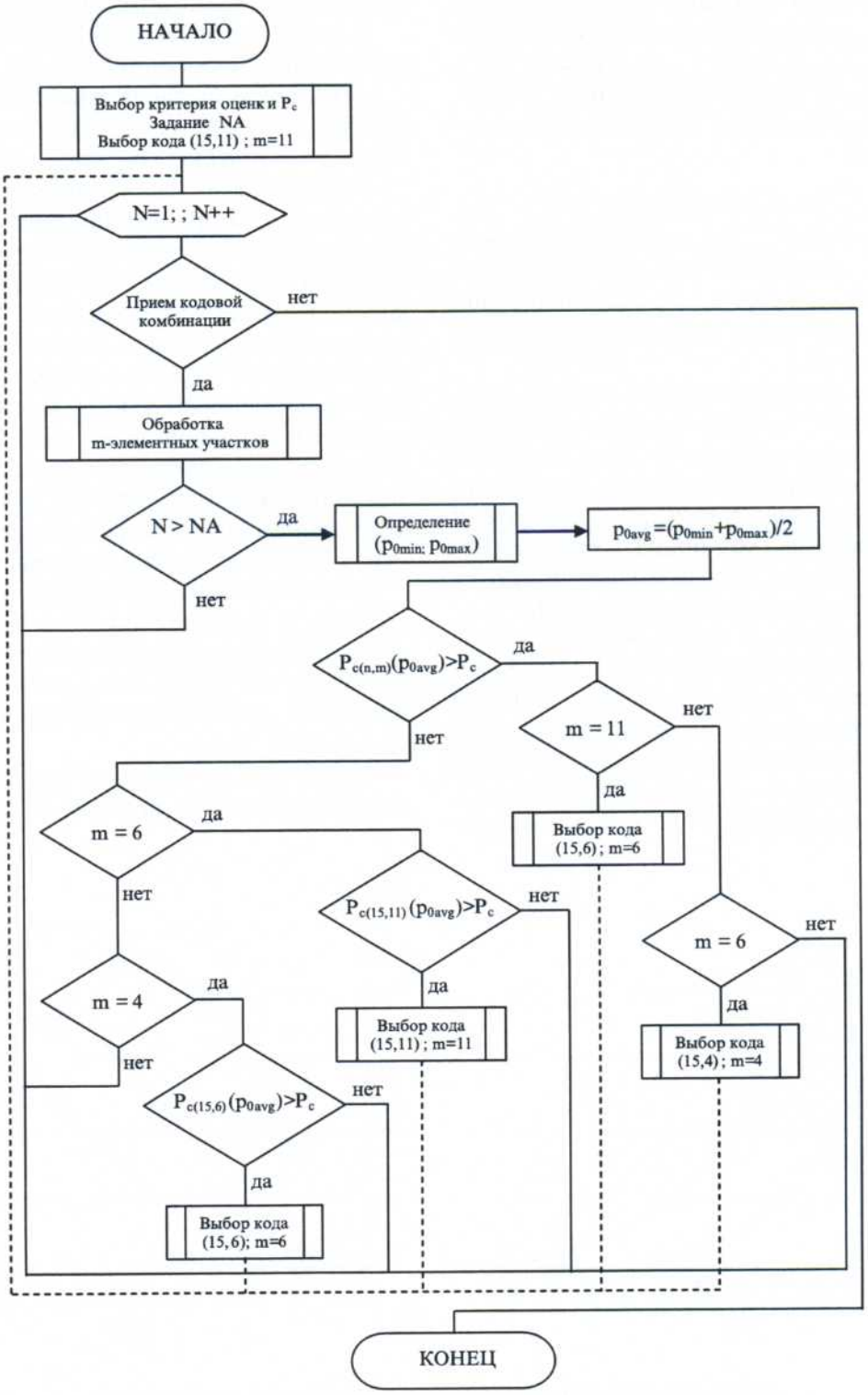


Рис. 2. Алгоритм работы адаптивной системы с критерием оценки P_c

Определение диапазона (p_{0min}; p_{0max}) (рис. 2) производится при обработке m-элементных участков двойственным базисом методом с непрерывным накопления элементов. В качестве анализируемого параметра выступает величина p_{0avg} – среднее значение в диапазоне. При выполнении условия $N>NA$ начинается проверка соответствия вероятности критерия оценки $P_{c(n,m)}(p_{ош})$ или $P_{e(n,m)}(p_{ош})$ при $p_{ош}=p_{0avg}$ соответственно заданной величине P_c или P_e. Таким образом, должно выполняться одно из двух условий:

$$P_{c(n,m)}(p_{0avg}) > P_c \text{ или } P_{e(n,m)}(p_{0avg}) < P_e. \quad (1)$$

Зависимости $P_{c(n,m)}(p_{ош})$ и $P_{e(n,m)}(p_{ош})$ можно построить экспериментально, произвести набор статистики декодирования достаточно большого количества кодовых комбинаций ($N = 10^5$) (рис. 3).

В случае невыполнения условий (1) осуществляется переход на более помехоустойчивый код, если такой доступен в системе. При этом происходит обнуление счетчика N, и снова начинается накопление элементов с обработкой m-элементных участков принятых последовательностей. Покажем переходные процессы также на временной диаграмме (рис. 4).

Как видно (рис. 4), при $p_{ош}$=0,023 в системе использовался циклический код (15, 11). В определенный момент времени значение $p_{ош}$ выросло до 0,098. Для адаптивной системы канал является стационарным, поэтому реакция на данное событие потребовала некоторого времени. С накоплением N>3000 (L>4,5·10⁴) выяснилось, что при P_c=0,95 условия (1) не выполняются. В связи с этим принимается решение о переходе на более помехоустойчивый код (15,6). Дальнейший анализ при использовании кода (15,6) показал, что данный код также не удовлетворяет условиям (1) и, следовательно, требует перехода на еще более помехоустойчивый код (15,4).

Переход на более помехоустойчивый код целесообразен также и в случае получения p_{0max}>0,5, когда большая вероятность ошибки в канале не позволяет с достаточной степенью точности определить (p_{0min}; p_{0max}) при $N>NA$.

Таким образом, выбор наиболее помехоустойчивого кода при максимально допустимой информационной скорости передачи ставит целью не только повышение достоверности передачи, но и решает задачу оптимального расходования ресурсов системы. Поэтому целесообразно рассмотреть ситуацию, когда наблюдается улучшение состояния канала, характеризуемое уменьшением величины $p_{ош}$, или происходит снижение требований к параметрам P_c и P_e, что способно привести к переходу на менее помехоустойчивый код, который должен повысить информационную скорость передачи.

Для этого сначала происходит проверка условий (1) для кода, по результатам которой может быть осуществлен переход на данный код. В случае перехода на другой код счетчик N обнуляется, и накопление элементов начинается сначала (рис. 5).

Видно (рис. 5), что при снижении $p_{ош}$ с 0,083 до 0,018 произошел переход от кода (15,4) к

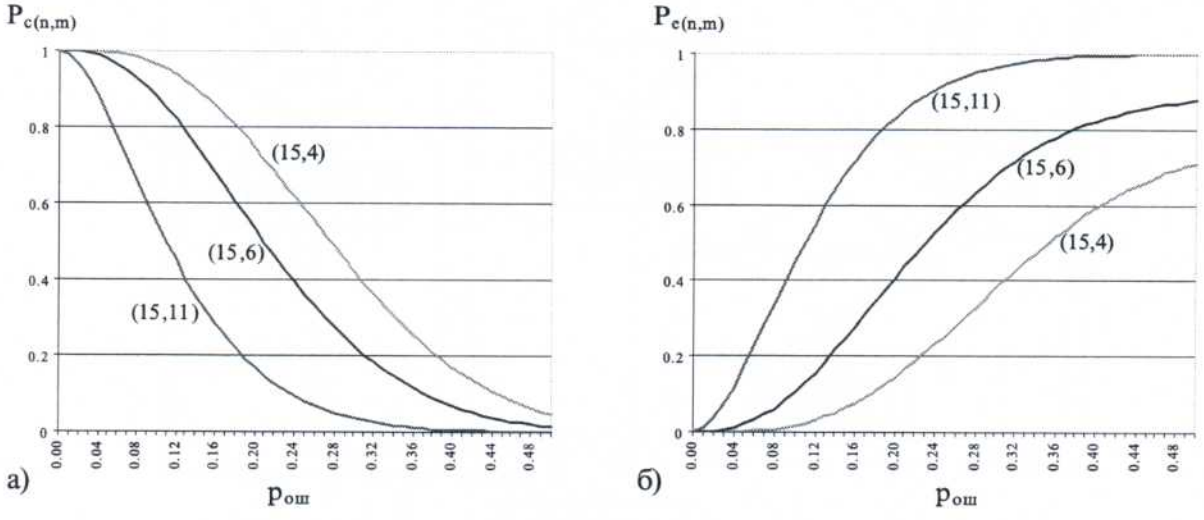


Рис. 3. Экспериментальные зависимости для различных кодов:
а – $P_{c(n,m)}(p_{ош})$; б – $P_{e(n,m)}(p_{ош})$

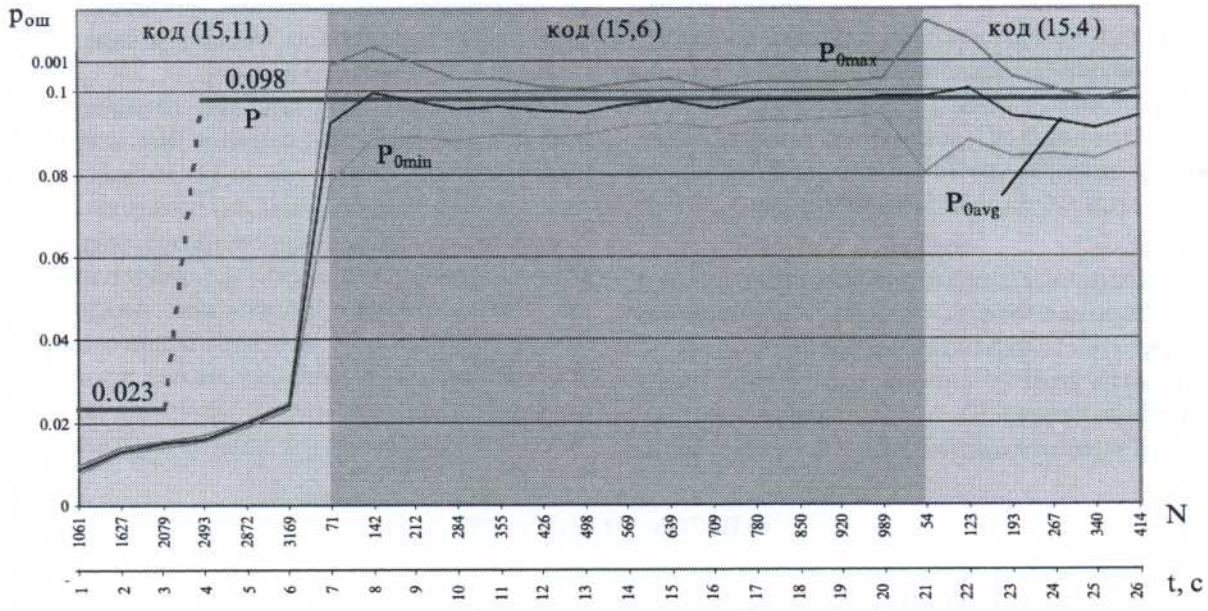


Рис. 4. Графическая диаграмма перехода системы на более помехоустойчивый код

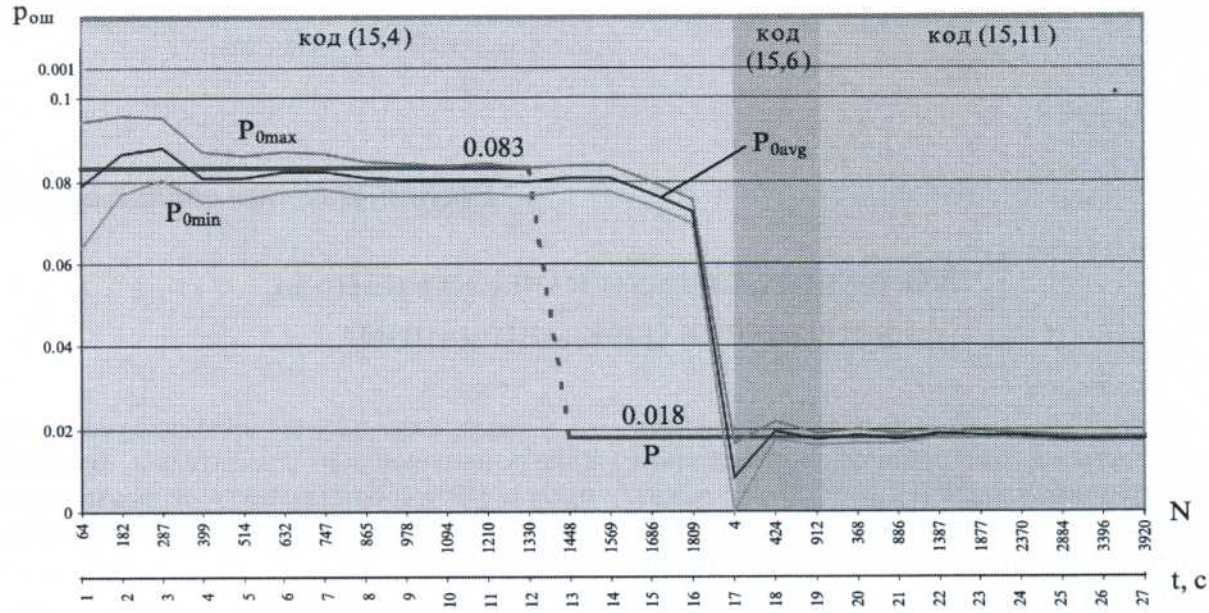


Рис. 5. Графическая диаграмма перехода системы на менее помехоустойчивый код

коду (15,6) и далее к коду (15,11). При этом соблюдается условие (1), обеспечивающее заданный уровень достоверности, и достигается максимальная информационная скорость передачи. Необходимо отметить в том числе относительно рассмотренных примеров (рис. 5), что подобная система может быть построена с учетом перехода от текущего кода к оптимальному, минуя промежуточные, но в данном случае ставилась цель показать процесс поэтапного перехода между кодами в соответствии с критерием оценки канала.

Рассмотренные примеры работы системы (рис. 4 и 5) демонстрируют применимость

метода непрерывного накопления элементов для оценки качества канала, позволяющего по достаточному числу обработанных m-элементных участков сделать заключение о величине $p_{ош}$, что в дальнейшем позволяет данной системе адаптироваться к состоянию канала, поддерживая информационную скорость передачи при заданном уровне достоверности.

Вывод

Созданная для практических исследований имитационная модель системы с адаптивным кодированием позволяет наглядно продемонстрировать работу алгоритмов адаптации к состоянию канала. Как и ожидалось, с накоплением m-элементных участков и соответственно принятых последовательностей наблюдается сближение границ диапазона, в пределах которого лежит вероятность ошибки в канале, что повышает точность ее оценки.

С ростом вероятности ошибки в канале или при повышении требований к достоверности принимаемой информации возможна ситуация, когда используемый циклический код не обеспечивает заданный уровень достоверности. В этом случае система осуществляет переход на более помехоустойчивый код. Аналогично при улучшении состояния канала или снижении требований к его качеству может произойти переход на менее помехоустойчивый, но более быстрый код с меньшей избыточностью.

СПИСОК ЛИТЕРАТУРЫ

1. **Когновицкий О.С.** Алгебраический метод нахождения двойственного базиса в поле Галуа и его практическое применение // Сборник научных трудов учебных институтов связи, – Л., 1982.

2. **Когновицкий О.С., Кукунин Д.С.** Метод оценки качества канала передачи данных при декодировании циклических кодов как рекурсивных последовательностей // 59-я НТК: мат. конф./ СПбГУТ. СПб, 2007.

3. **Когновицкий О.С., Кукунин Д.С.** Адаптация системы передачи к состоянию канала // 60-я НТК: мат. конф./ СПбГУТ. СПб, 2008.

4. **Когновицкий О.С., Кукунин Д.С.** Сетевая версия калькулятора Галуа // 57-я Юбилейная НТК: мат. конф./ СПбГУТ. СПб, 2005.

Кукунин Д.С., Когновицкий О.С.

Методика оценки качества канала в процессе передачи данных

Предлагаемая методика позволяет оценить вероятность ошибки в стационарном двоичном симметричном канале [1]. В ее основе лежит использование рекуррентных последовательностей и их обработка двойственным базисом поля Галуа [2].

Для примера приведем результат декодирования комбинации кода (15,6) методом, основанным на использовании двойственного базиса [3]. Верным решением в данном случае является совокупность элементов (ε^{10}, μ) (рис. 1).

Высота соответствующего им "дерева" превышает остальные "деревья" и равна 5. В случае отсутствия ошибок в принятой кодовой комбинации $c(x)$ высота "дерева" (ε^{10}, μ) должна строго равняться 15, то есть длине кода БЧХ (15,6).

Предлагается производить оценку качества канала на основании показателя J, который представляет собой среднюю скорость роста количества "деревьев" и определяется из формулы:

$$J = \frac{K_L}{L}, \qquad (1)$$

где K_L – количество "деревьев", полученное при обработке L m-элементных участков двойственным базисом.

Показатель K_L представляет собой сумму

$$K_L = \sum_{j=1}^{L} K_j, \qquad (2)$$

в которой каждое слагаемое K_j определяется следующим образом:

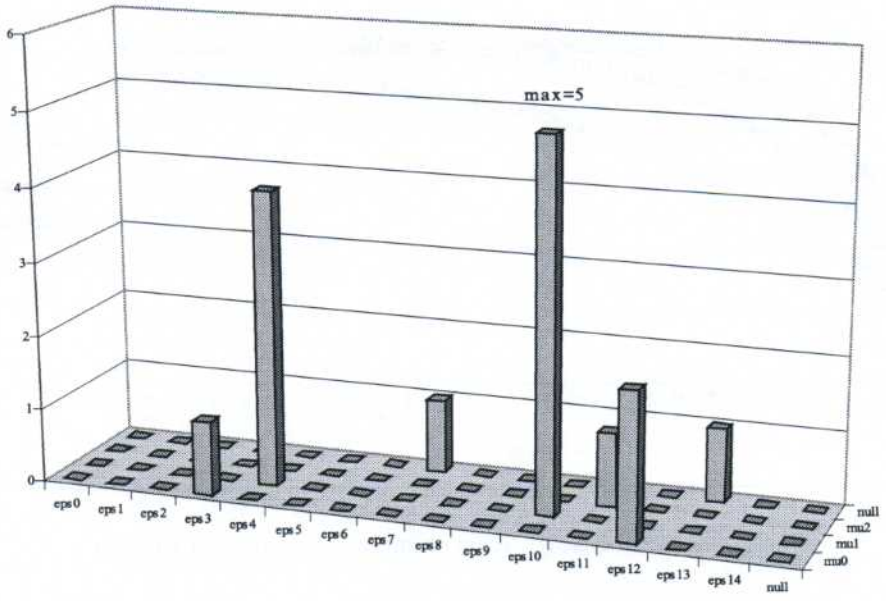


Рис. 1. Декодирование последовательности кода БЧХ (15,6) с параметрами: $P(x) = x^6 + x^5 + x^4 + x^3 + 1$, $c(x) = 101000111111011$, $e(x) = 010011000000000$

$$K_j = \begin{cases} 1, & \text{если } C_{i\tau} = 0 \\ 0, & \text{если } C_{i\tau} > 0 \end{cases}, \qquad (3)$$

где $C_{i\tau}$ – количество элементов поля ε^i, полученных в результате обработки двойственным базисом m-элементных участков принимаемой τ кодовой комбинации циклического (n,m) кода. Таким образом, при отсутствии ошибок в комбинации первый ее m-элементный участок определит $K_j = 1$, при обработке остальных получим $K_j = 0$.

Общее количество k-элементных участков L определяется как сумма m-элементных участков во всех принятых кодовых комбинациях N:

$$L = \sum_{\tau=1}^{N} L_\tau. \qquad (4)$$

При этом количество m-элементных участков L_τ в каждой комбинации τ связано с числом N_{b_τ} принятых ее элементов соотношением:

$$L_\tau = \begin{cases} N_{b_\tau}, \text{ если } N_{b_\tau} = n \\ N_{b_\tau} - (m-1), \text{ если } m \le N_{b_\tau} < n \end{cases}, \qquad (5)$$

так как передача всей последовательности целиком позволяет произвести обработку m-элементных участков по замкнутым в цикл элементам.

Вместе с этим общее количество анализируемых элементов складывается из количества элементов в каждой кодовой комбинации:

$$N_b = \sum_{\tau=1}^{N} N_{b_\tau}. \qquad (6)$$

Учитывая (1)–(6), запишем показатель J в виде:

$$J = \frac{\sum_{j=1}^{L} K_j}{\sum_{\tau=1}^{N} L_\tau}. \qquad (7)$$

Для эквидистантного кода (15,4) и кодов БЧХЭ (15,6) и (15,11) построим зависимость $J(p_{ош})$. Для точности возьмем $N_b = 1{,}5 \cdot 10^7$, что соответствует 10^6 кодовым комбинациям (рис. 2).

Длина выборки является достаточно большой ($N_b = 1{,}5 \cdot 10^7$), чтобы значение J считать близким к математическому ожиданию случайной величины K_j.

Для K_j определим среднеквадратическое отклонение σ и оценим доверительный интервал d для случайной величины J [4]:

$$\sigma = \sqrt{\sum_{j=1}^{L} \frac{(K_j - J)^2}{L}}; \quad d = t\sigma / \sqrt{L}, \qquad (8)$$

где t – квантиль функции распределения Лапласа.

Для определения доверительного интервала вероятности ошибки необходимо производить накопление N_b двоичных элементов по мере приема кодовых комбинаций.

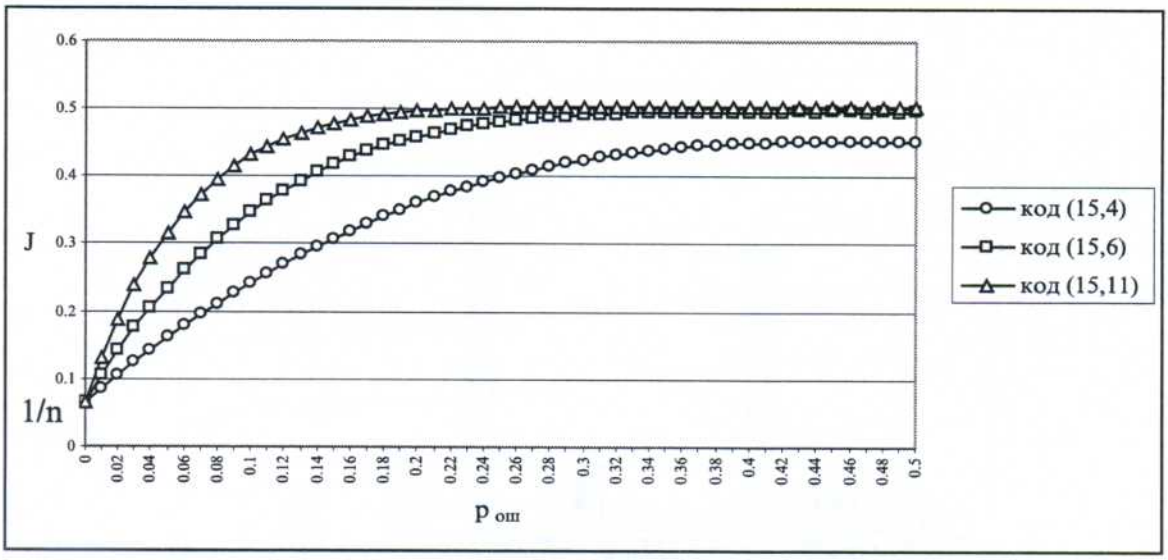


Рис. 2. Зависимость $J(p_{ош})$ для различных циклических кодов

После вычисления показателя J для заданного L определяется оценочное значение вероятности ошибки $p_0=x_0$ из решения уравнения:

$$v(x_0) = J. \qquad (9)$$

При решении уравнения (9) учитывается условие, что искомый корень находится в диапазоне (0;0,5), на котором производится аппроксимация функций при помощи программы Advanced Grapher 2.11 (рис. 2, табл. 1).

Таким образом, определяется единственный корень, который соответствует p_0. Далее по формуле (8) определяется доверительный интервал $\pm d(p_0,L)$ для J с учетом $P_{дв}$.

Границы доверительного интервала вероятности ошибки $p_{0min} = x_{min}$ и $p_{0max} = x_{max}$ определяются из уравнений:

$$v(x_{min}) = J - d(p_0, L);\ \ v(x_{max}) = J + d(p_0, L). \qquad (10)$$

Учитывая, что зависимости $d(p_{ош}, L)$ могут быть определены заранее, значения p_{0min} и p_{0max} для любого J при обработке L m-элементных участков также можно определить заранее. Покажем зависимость $p_{0min}(J)$ и $p_{0max}(J)$ при $L = 1{,}5\cdot10^3$ ($N = 100$) для различных кодов (рис. 3).

Аппроксимирующие функции $Z_{min}(J)$ и $Z_{max}(J)$ для зависимостей $p_{0min}(J)$ и $p_{0max}(J)$ (рис. 3) определены при помощи Advanced Grapher 2.11 (табл. 2).

Применяя функции (табл. 2), можно определить границы диапазона вероятности ошибки в канале:

$$p_{0min} = Z_{min}(J);\ p_{0max} = Z_{max}(J). \qquad (11)$$

Функции $Z_{min}(J)$ и $Z_{max}(J)$ (табл. 2) определены на отдельных участках J. Их можно использовать для определения p_{0min} и p_{0max} при реальной $p_{ош} < 0{,}3$ для эквидистантного кода (15,4). Для кодов (15,6) и (15,11) порог вероятности ошибки составляет $p_{ош} < 0{,}22$ и $p_{ош} < 0{,}15$ соответственно.

Рост $p_{ош}$ в канале приводит к расширению интервала $(p_{0min}; p_{0max})$ для всех кодов, однако для кодов с меньшим числом избыточных элементов расширение данного диапазона на всей длине J происходит быстрее, чем для кодов с большей избыточностью (рис. 3).

Покажем, как зависит величина интервала $(p_{0min}; p_{0max})$ от роста $p_{ош}$ в канале и значения L

Таблица 1

Код	Аппроксимирующие функции
(15,4)	$v_1(x) = 53{,}038x^8 - 173{,}881x^7 + 223{,}884x^6 - 151{,}454x^5 + 61{,}832x^4 - 14{,}091x^3 - 1{,}265x^2 + 1{,}975x + 0{,}0668$
(15,6)	$v_2(x) = -776{,}248x^8 + 1651{,}429x^7 - 1415{,}045x^6 + 623{,}738x^5 - 162{,}526x^4 + 40{,}884x^3 - 15{,}128x^2 + 4{,}018x + 0{,}0667$
(15,11)	$v_3(x) = 226{,}899x^8 - 224{,}333x^7 - 238{,}574x^6 + 552{,}064x^5 - 435{,}333x^4 + 189{,}261x^3 - 48{,}706x^2 + 7{,}017x - 0{,}0665$

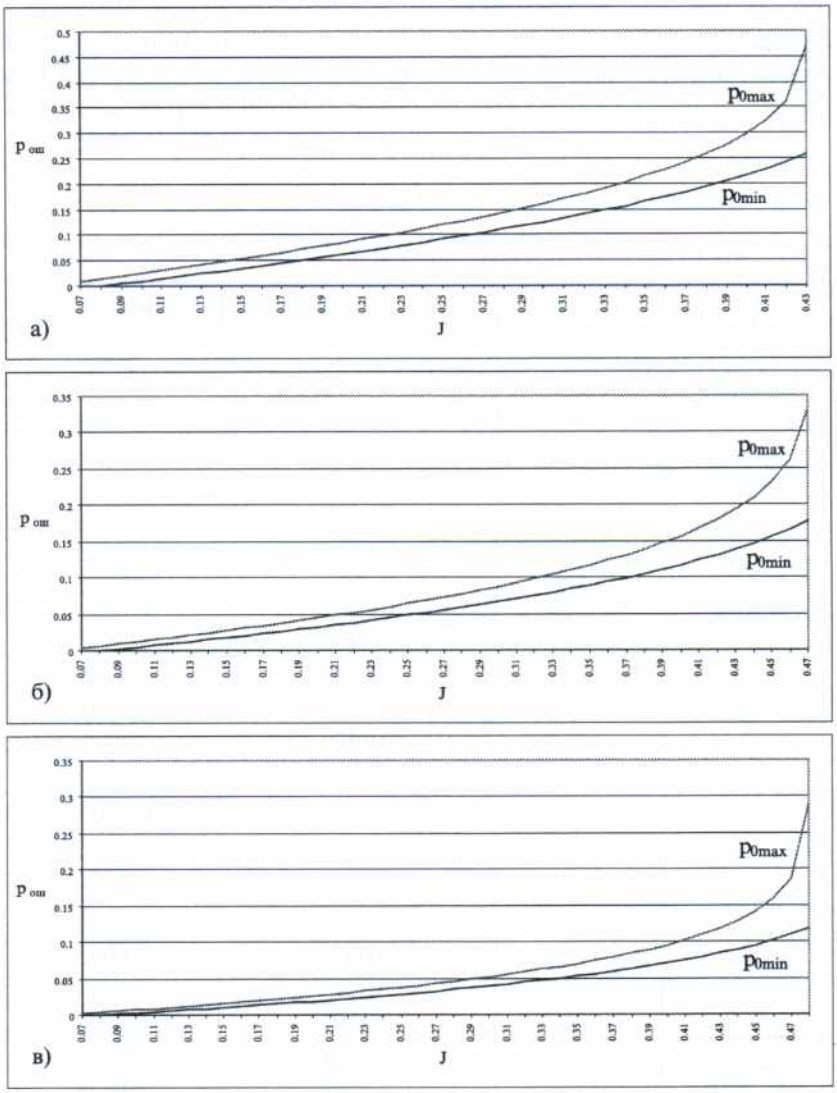


Рис. 3. Зависимость $p_{0\min}(J)$ и $p_{0\max}(J)$ при $P_{дв} = 0{,}95$ и $L = 1500$ для различных циклических кодов: а – (15,4); б – (15,6); в – (15,11)

Таблица 2

Код	Аппроксимирующие функции	
(15,4)	$Z_{\min}(J)$	$33288.4449J^8 - 69546.7850J^7 + 61870.3317J^6 - 30464.3176J^5 + 9041.5073J^4 - 1648.4049J^3 + 179.8675J^2 - 10.2289J + 0.2269$
	$Z_{\max}(J)$	$(5.0804 \cdot 10^5)J^8 - (9.5602 \cdot 10^5)J^7 + (7.5904 \cdot 10^5)J^6 - (3.3077 \cdot 10^5)J^5 + 86172.4659J^4 - 13681.8466J^3 + 1287.6480J^2 - 64.8845J + 1.3404$
(15,6)	$Z_{\min}(J)$	$8526.4677J^8 - 18751.4896J^7 + 17578.6367J^6 - 9125.5367J^5 + 2856.7414J^4 - 549.3324J^3 + 63.3209J^2 - 3.7414J + 0.0846$
	$Z_{\max}(J)$	$(1.2627 \cdot 10^5)J^8 - (2.5505 \cdot 10^5)J^7 + (2.1700 \cdot 10^5)J^6 - (1.0111 \cdot 10^5)J^5 + 28076.5097J^4 - 4732.8999J^3 + 470.7970J^2 - 24.8735J + 0.5351$
(15,11)	$Z_{\min}(J)$	$4729.2101J^8 - 10404.8881J^7 + 9763.7081J^6 - 5075.4900257J^5 + 1591.6190J^4 - 306.5525J^3 + 35.3854J^2 - 2.0890J + 0.0471$
	$Z_{\max}(J)$	$(1.9693 \cdot 10^5)J^8 - (4.0742 \cdot 10^5)J^7 + (3.5464 \cdot 10^5)J^6 - (1.6884 \cdot 10^5)J^5 + 47845.5215J^4 - 8219.1946J^3 + 831.5738J^2 - 44.9287J + 0.9907$

при $P_{дв} = 0{,}95$ (табл. 3), где $p_{ош}$ – вероятность ошибки модели канала, P_L – реальная средняя вероятность ошибки в L обработанных k-элементных участках.

Как и следовало ожидать, с увеличением L происходит уменьшение доверительного интервала (p_{0min}; p_{0max}), что приводит к более точной оценке вероятности ошибки в канале.

Таблица 3

Код (15,4)						
$p_{ош}$	L	P_L	J	p_0(оценка по методике)	Доверительный интервал	
					p_{0min}	p_{0max}
0.01	150	0.0067	0.08000	0.0067	0.0000	0.0283
	$1{,}5 \cdot 10^3$	0.0073	0.08133	0.0074	0.0006	0.0141
	$1{,}5 \cdot 10^4$	0.0097	0.08573	0.0096	0.0074	0.0118
	$1{,}5 \cdot 10^5$	0.0099	0.08625	0.0099	0.0092	0.0106
0.1	150	0.1400	0.31333	0.1535	0.0991	0.2294
	$1{,}5 \cdot 10^3$	0.0927	0.23000	0.0917	0.0789	0.1051
	$1{,}5 \cdot 10^4$	0.0975	0.23547	0.0953	0.0911	0.0996
	$1{,}5 \cdot 10^5$	0.1016	0.24407	0.1011	0.0997	0.1024
0.25	150	0.2067	0.37333	0.2140	0.1377	0.4099
	$1{,}5 \cdot 10^3$	0.2600	0.41333	0.2726	0.2329	0.3334
	$1{,}5 \cdot 10^4$	0.2471	0.39627	0.2445	0.2331	0.2570
	$1{,}5 \cdot 10^5$	0.2522	0.39969	0.2497	0.2459	0.2537
Код (15,6)						
$p_{ош}$	L	P_L	J	p_0(оценка по методике)	Доверительный интервал	
					p_{0min}	p_{0max}
0.01	150	0.0067	0.09333	0.0068	0.0000	0.0190
	$1{,}5 \cdot 10^3$	0.0120	0.11467	0.0125	0.0084	0.0167
	$1{,}5 \cdot 10^4$	0.0097	0.10320	0.0094	0.0082	0.0106
	$1{,}5 \cdot 10^5$	0.0098	0.10469	0.0098	0.0094	0.0102
0.1	150	0.0733	0.30667	0.0797	0.0502	0.1192
	$1{,}5 \cdot 10^3$	0.0813	0.30533	0.0791	0.0690	0.0900
	$1{,}5 \cdot 10^4$	0.0953	0.33720	0.0948	0.0909	0.0988
	$1{,}5 \cdot 10^5$	0.0998	0.34672	0.0999	0.0986	0.1012
0.25	150	0.2933	0.45333	0.1891	0.1157	>0.5
	$1{,}5 \cdot 10^3$	0.2587	0.47133	0.2207	0.1794	0.3584
	$1{,}5 \cdot 10^4$	0.2484	0.48207	0.2495	0.2270	0.2845
	$1{,}5 \cdot 10^5$	0.2518	0.48249	0.2509	0.2428	0.2602
Код (15,11)						
$p_{ош}$	L	P_L	J	p_0(оценка по методике)	Доверительный интервал	
					p_{0min}	p_{0max}
0.01	150	0.0067	0.11333	0.0070	0.0000	0.0149
	$1{,}5 \cdot 10^3$	0.0127	0.14600	0.0123	0.0096	0.0152
	$1{,}5 \cdot 10^4$	0.0095	0.13013	0.0097	0.0089	0.0105
	$1{,}5 \cdot 10^5$	0.0099	0.13149	0.0099	0.0096	0.0102
0.1	150	0.0733	0.34667	0.0596	0.0377	0.0934
	$1{,}5 \cdot 10^3$	0.0820	0.40733	0.0852	0.0733	0.1000
	$1{,}5 \cdot 10^4$	0.0985	0.42973	0.0983	0.0933	0.1038
	$1{,}5 \cdot 10^5$	0.0996	0.42815	0.0973	0.0957	0.0990
0.25	150	0.2133	0.44000	0.1056	0.0644	>0.5
	$1{,}5 \cdot 10^3$	0.2347	0.48200	0.1527	0.1197	>0.5
	$1{,}5 \cdot 10^4$	0.2517	0.50533	0.2909	0.1948	>0.5
	$1{,}5 \cdot 10^5$	0.2513	0.50207	0.2246	0.2062	0.2616

При этом на основании результатов (табл. 3) можно заключить, что для кодов с большей избыточностью имеется возможность производить оценку канала при больших вероятностях ошибки, чем для кодов с меньшим количеством проверочных элементов.

Также из табл. 3 видно, что для кодов с большей избыточностью величина доверительного интервала вероятности ошибки имеет большее значение, то есть для кода (15,11) доверительный интервал (p_{0min}; p_{0max}) является более узким, чем для кода (15,4) при равных вероятностях ошибки в канале. Это означает, что применение данного метода в случае с кодом (15,11) позволяет с большей степенью точности произвести оценку качества канала, чем при использовании кода (15,4).

При этом необходимо учитывать ограничения, накладываемые на код (15,11), для которого при $p_{ош} = 0{,}25$ не удалось определить верхнюю границу диапазона вероятности ошибки ($p_{0min} > 0{,}5$) даже при $L = 1{,}5 \cdot 10^4$. Для кода (15,6) аналогичная сложность возникла при $L = 150$.

Для наглядности построим зависимости p_{0min} и p_{0max} от L для всех кодов при $p_{ош} = 0{,}01$ и 0,1 (рис. 4).

Алгоритм работы метода с непрерывным накоплением элементов основан на обработке m-элементных участков принятых кодовых

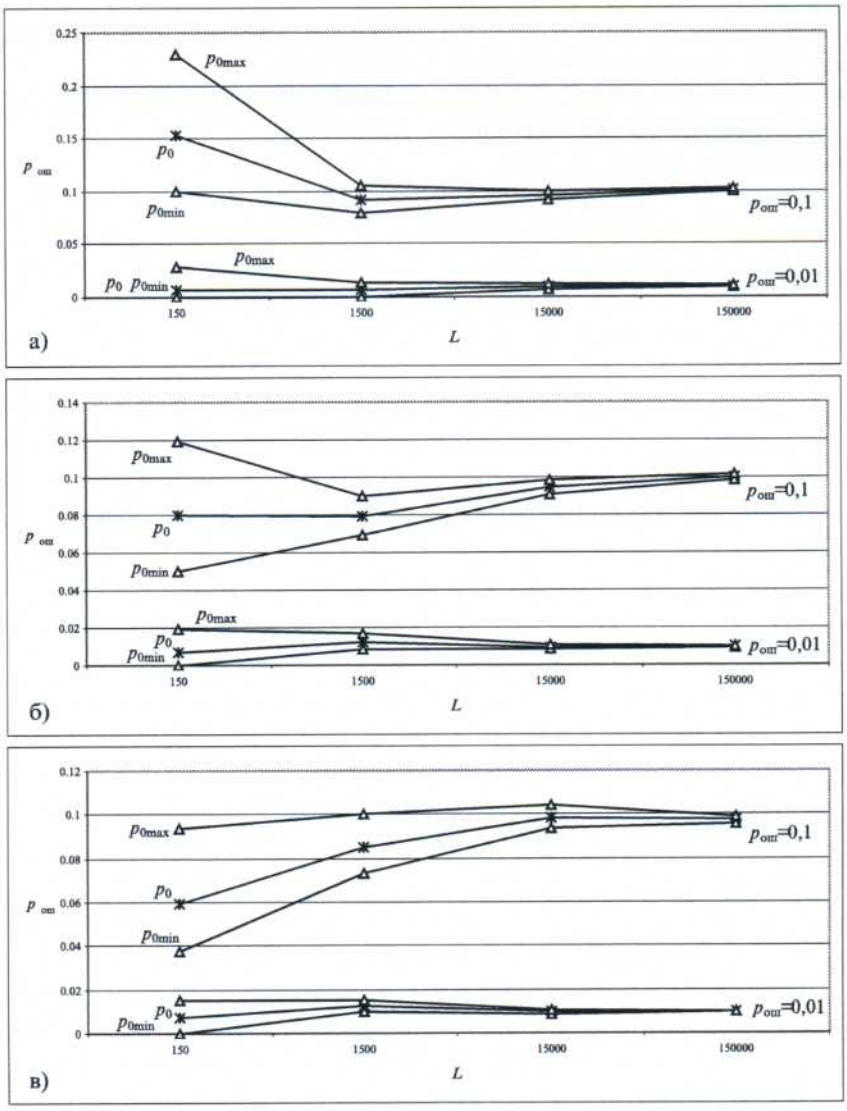


Рис. 4. Оценка канала методом с непрерывным накоплением элементов, определение границ p_{0min} и p_{0max} при $P_{дв}$=0,95 при декодировании комбинаций циклических кодов: а – (15,4); б – (15,6); в – (15,11)

комбинаций циклического (n,m) кода. Таким образом, предложенная методика обладает двумя важными свойствами. Во-первых, она обеспечивает контроль состояния канала в процессе передачи данных. Во-вторых, процесс вычисления интервала $(p_{0\min}; p_{0\max})$ по накопленной статистике не связан с результатом декодирования кодовых комбинаций, что позволяет подключить блок для оценки вероятности ошибки в канале параллельно работающему декодеру. Таким образом, устройство, реализующее данный метод, может получать поступившую информацию одновременно с декодирующим устройством, что значительно ускоряет процесс накопления статистики.

Что касается оперативности вычисления диапазона вероятности ошибки в канале, отметим возможность предварительного построения функций $Z_{\min}(J)$ и $Z_{\max}(J)$ для определенных значений L. В случае применения таких функций для множества L целесообразно использование соответствующих поверхностей, заданных в векторном виде.

СПИСОК ЛИТЕРАТУРЫ

1. **Зюко А.Г., Кловский Д.Д., Назаров М.В., Финк Л.М.** Теория передачи сигналов: Учебник для вузов. – М.: Радио и связь, 1986.

2. **Когновицкий О.С.** Алгебраический метод нахождения двойственного базиса в поле Галуа и его практическое применение // Сборник научных трудов учебных институтов связи, – Л., 1982.

3. **Когновицкий О.С.** Циклические коды БЧХЭ как рекурсивные последовательности // Труды учебных заведений связи / СПбГУТ. СПб, 2006.

4. **Худсон Д.** Статистика для физиков. – М.: Мир, 1967.

Новиков Ф.А., Тихонова У.Н.

Определение проблемно-ориентированных языков интерпретируемыми автоматами

Создание программного обеспечения – это сложный процесс. Наиболее критичной частью этого процесса является непосредственно программирование приложения, так как программистам приходится работать с двумя предметными областями: предметной областью целевой задачи и используемым языком программирования. Программист осуществляет отображение одной предметной области в другую, поэтому от него требуется знание обоих предметных областей. Программирование в смысле такого отображения может быть упрощено с помощью проблемно-ориентированных языков (domain-specific languages, DSL). Проблемно-ориентированные языки позволяют решать целевую задачу в терминах этой задачи, а не в терминах вычислительной машины [1] и в идеале могут использоваться непосредственно специалистами в данной области. Такой подход облегчает разработку программ и повышает их качество.

Проблемно-ориентированный язык, как и язык программирования общего назначения является средством выражения решения задачи. Он определяется синтаксисом и семантикой. Абстрактный синтаксис описывает класс допустимых языком программ (в смысле решений некоторых задач). Конкретный синтаксис определяет представление для этого класса программ. Семантика придает программам содержание и смысл, определяет их поведение. Следовательно, описание языка программирования включает в себя определение всех этих трех составляющих. На практике обычно формально определяется только конкретный синтаксис языка (в виде его грамматики), а семантика и абстрактный синтаксис описываются неформально и скрываются в реализующий язык транслятор. Такой подход влечет немало неудобств как для использования языка, так и для его поддержки. Одной из технологий, разработанных с целью устранить этот недостаток, является языковый инструментарий Meta Programming System (MPS) [2].

Другим методом определения языков программирования, включающих в себя не только формальное описание конкретного синтаксиса языка, но и формальное определение его абстрактного синтаксиса и семантики, является Венский метод [3]. Для описания абстрактного синтаксиса в нем используется абстрактная грамматика, а для определения семантики языка – интерпретирующие автоматы.

На данный момент автоматы широко применяются как для описания и проектирования алгоритмов, так и для их реализации [4]. Программирование с использованием автоматов, или автоматное программирование, основано на работе виртуальной машины, интерпретирующей автоматы. Согласно этому подходу определение семантики языка с помощью автоматов является одновременно интерпретатором данного языка. Кроме того, автоматное программирование позволяет рассматривать определение конкретного синтаксиса языка в виде диаграммы состояний и переходов как синтаксический анализатор языка [5].

В данной статье описывается метод определения проблемно-ориентированных языков с помощью системы автоматов, интерпретируемых виртуальной машиной автоматного программирования. Структура языка, его абстрактный синтаксис описывается в виде диаграммы классов в нотации UML и определяет представление конкретной программы, которое используют автоматы. Семантика языка задается системой автоматов, которые реализуют интерпретацию программы как экземпляра абстрактного синтаксиса. Конкретный синтаксис задается с помощью системы автоматов,

реализующих разбор программы. При этом прототипом такой системы автоматов является структура языка. Автоматы интерпретируются виртуальной автоматной машиной, поэтому такая спецификация языка и есть программа, его реализующая. Полученный таким образом языковый процессор не использует грамматического описания языка, а основан на связывании распознающих и интерпретирующих автоматов непосредственно с абстрактной структурой языка.

Предлагаемый метод опробован на примере описания спецификации языка СЛОН (слежение и обработка наблюдений) [6] системы ЭРА (Эфемеридные Расчеты Астрономии) [7] – прикладной проблемно-ориентированной системы программирования, предназначенной для решения разнообразных задач астрономии.

Ниже рассмотрены основные идеи предлагаемого метода, примеры его применения для описания языка СЛОН и программная реализация.

Определение абстрактного синтаксиса с помощью диаграммы классов

Для описания абстрактного синтаксиса с помощью диаграмм классов в предлагаемом методе определения проблемно-ориентированных языков используются следующие конструкции UML [**Ошибка! Источник ссылки не найден.**]:

- классы – для представления понятий определяемого языка;
- атрибуты классов – для представления свойств понятий;
- обобщение – для классификации понятий;
- композиция и (реже) ассоциация – для представления отношений между понятиями.

Аналогично тому, как формальная грамматика некоторого языка определяет синтаксическую структуру программы на этом языке, модель абстрактного синтаксиса (метамодель) определяет абстрактную структуру программы и составляющих ее частей. При этом выразительные средства диаграммы классов UML могут заменить все выразительные средства формальных грамматик следующим образом.

1. Множество нетерминалов заменяется множеством классов.

2. Множество терминалов разделяется на "семантически значимые" токены (например, имена и значения переменных, знаки операций) и "разделители" (скобки, запятые и т.д.). Семантически значимые токены заменяются значениями атрибутов соответствующих классов (например, идентификатор переменной – это атрибут класса *Переменная*), а разделители в метамодели не указываются.

3. Множество правил заменяется отношениями между классами.

4. В качестве начального нетерминала выступает класс, определяющий абстрактную программу.

5. Отношение выводимости заменяется отношением конкретизации (будем говорить, что объект a и класс A связаны отношением конкретизации, если объект a является экземпляром класса A).

Таким образом, выразительные средства диаграммы классов UML не слабее выразительных средств контекстно-свободных формальных грамматик. На рис. 1 приведен фрагмент метамодели языка СЛОН и указаны соответствия между выразительными средствами формальной грамматики и выразительными средствами диаграммы классов.

Определение конкретного синтаксиса с помощью распознающих автоматов

В классической теории трансляции для определения конкретного синтаксиса языка используются формальные грамматики. Одним из способов представления грамматических правил вывода являются диаграммы переходов. Если исключить в них появление нетерминалов на переходах, то можно рассматривать систему таких диаграмм как анализатор программы. Этот подход используется авторами статьи [5] для создания системы автоматического завершения ввода на основе парадигмы автоматного программирования.

В предлагаемом методе для описания конкретного синтаксиса проблемно-ориентированного языка используется система конечных автоматов, определенных следующим образом.

1. Входным алфавитом системы автоматов является множество терминалов языка.

2. Для описания взаимодействия автоматов используется нотация диаграммы состояний UML: составное состояние – это переход к соответствующему автомату.

При этом под терминалом понимается элемент нотации языка, или с практической точки зрения событие, посылаемое лексическим анализатором текста, или графическим редак-

<TableExpr > ::= number (<TableBinOp > | <NamedTable > | <Denotation >) '?' '('<Predicate >')' ...
<TableBinOp > ::= <TableExpr > <TTerm> <TableExpr >

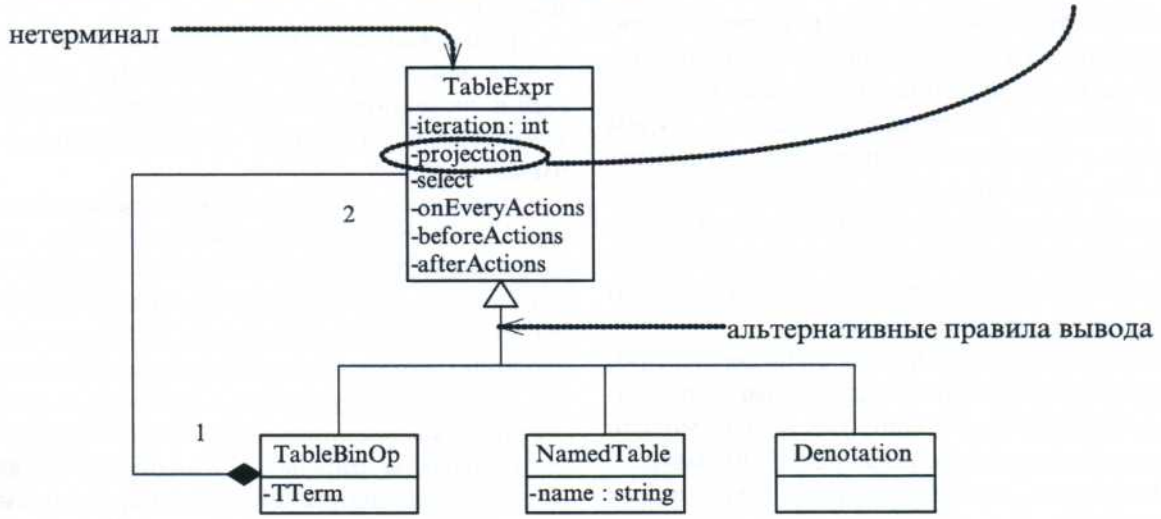


Рис. 1. Фрагменты грамматики и метамодели языка СЛОН: табличное выражение

тором, или диалоговым окном – любым источником событий, что позволяет использовать не только текстовое представление программы.

Прототипы распознающих конечных автоматов могут быть получены из метамодели языка с помощью следующих правил сведения.

1. Автомат-распознаватель определяется для каждой сущности (каждого класса) метамодели.

2. Отношение обобщения сводится к ветвлению в автомате.

3. Отношение ассоциации в зависимости от кратности полюсов сводится к циклу или к последовательности составных состояний в автомате.

Внесение необходимых изменений в структуру прототипов автоматов-распознавателей и задание элементов нотации (терминалов) языка на переходах определяют конкретный синтаксис языка. На рисунке 2 приведен автомат, распознающий фрагмент языка СЛОН – табличные выражения. Соответствующий фрагмент метамодели языка приведен на рисунке 1.

Определенная таким образом спецификация конкретного синтаксиса языка одновременно реализует задачу распознавания для языка (так как есть виртуальная машина, интерпретирующая автоматы спецификации).

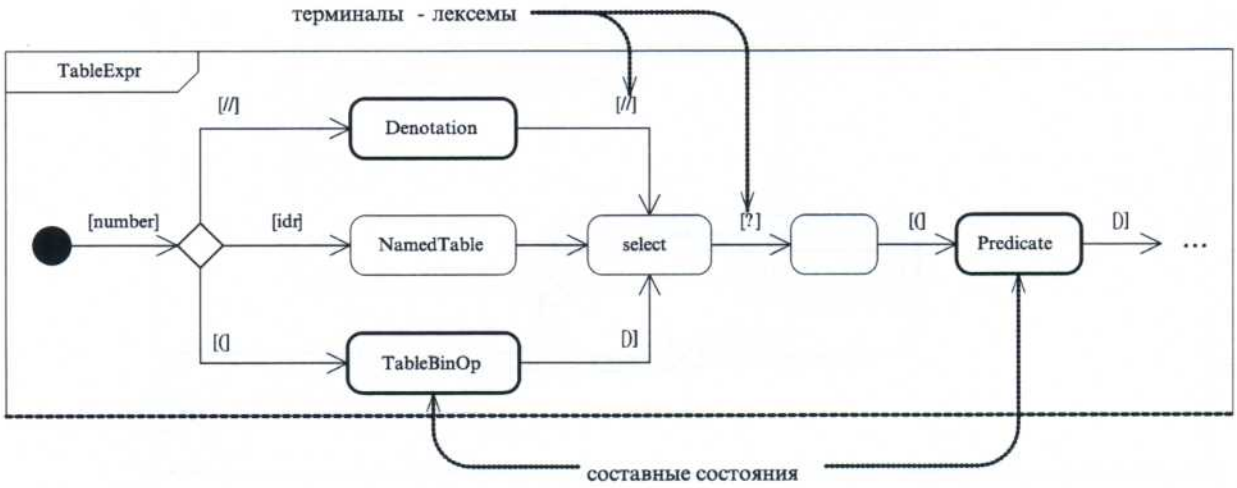


Рис. 2. Автомат, распознающий табличные выражения языка СЛОН

Определение семантики с помощью интерпретирующих автоматов

Операционный подход к определению семантики языка предполагает описание алгоритма интерпретации программы в терминах некоторой абстрактной машины. Будем строить алгоритм интерпретации программы на проблемно-ориентированном языке в виде системы конечных автоматов. При этом интерпретируемая программа (вход алгоритма) – это дерево, являющееся экземпляром абстрактного синтаксиса (метамодели) языка.

Для описания алгоритма выполнения проблемно-ориентированной программы в предлагаемом методе используется следующая модель (детерминированного конечного) автомата.

1. Автомат выполняется, используя входные события и предикаты.

2. Результатом выполнения автомата являются некоторые эффекты (выходные воздействия) и точки выхода (exit points).

3. Для взаимодействия между автоматами используются составные состояния и точки выхода.

Согласно парадигме автоматного программирования автомат управляет некоторым объектом управления. Здесь объектом управления является интерпретируемая программа – экземпляр метамодели проблемно-ориентированного языка. Таким образом, предикаты вычисляются и эффекты выполняются для экземпляра метамодели.

Точка выхода – это состояние, в котором может завершить свое выполнение вложенный в составное состояние автомат. По сути, точка выхода автомата – это возвращаемое им значение.

Именно с помощью такой модели системы взаимодействующих автоматов удалось описать семантику наиболее сложного фрагмента проблемно-ориентированного языка СЛОН – табличных выражений. Язык СЛОН реализует таблично ориентированное программирование [7], которое рассматривает таблицу, с одной стороны, как массив данных, с другой стороны как программу, определяющую последовательность присваивания значений переменным – величинам предметной области. Как массив данных таблица может храниться в (внешней) памяти. Таблица как программа – это итератор, перебирающий кортежи таблицы, чтобы выполнять с ними некоторые действия. Табличные выражения реализуют механизм алгебры таблиц, позволяющий строить из имеющихся таблиц произвольные таблицы.

Например, операция умножения таблиц (в метамодели языка она представлена классом бинарных табличных операций *TableBinOp*, см. рис. 1) позволяет последовательно объединить

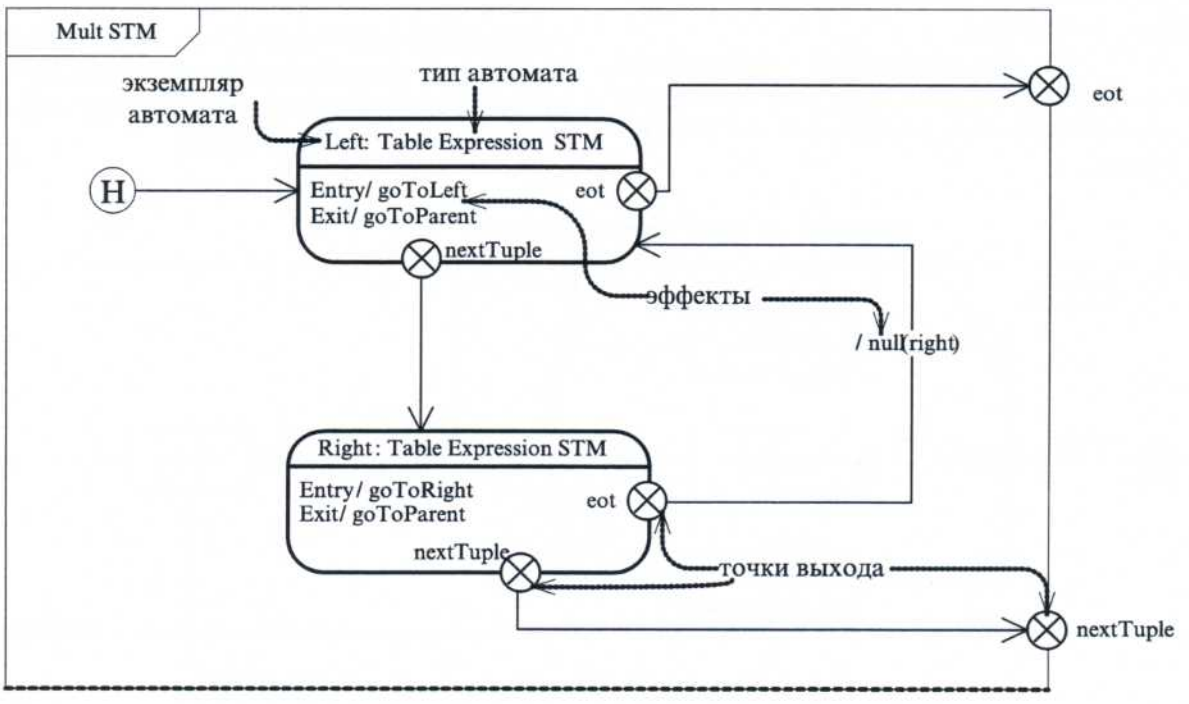


Рис. 3. Семантика операции умножения таблиц

все кортежи двух таблицы. Если рассматривать таблицу как программу (или как цикл), то умножению таблиц соответствует вложенность циклов (рис. 3).

Кроме того, для описания семантики табличных операций языка СЛОН используется следующая модель выполнения (интерпретации) системы автоматов:

- структура автомата (его состояния и переходы между ними) определяет класс автоматов;
- при выполнении (интерпретации) автомата создается соответствующий экземпляр класса автоматов;
- в процессе выполнения алгоритма, описанного с помощью системы интерпретирующих автоматов, может создаваться столько экземпляров автоматов, сколько потребуется.

Такая модель выполнения системы автоматов позволяет использовать рекуррентное определение семантики с сохранением локальных данных каждого автомата (смыслового элемента этого определения).

Программная реализация метода с помощью машины автоматного программирования

Виртуальная машина автоматного программирования позволяет рассматривать определение проблемно-ориентированного языка как реализующую его программу. Интерпретация распознающих автоматов, описывающих конкретный синтаксис, решает задачу распознавания для языка. Интерпретация интерпретирующих автоматов, определяющих семантику, реализует интерпретатор языка.

Машина автоматного программирования реализована с помощью применения метода раскрутки (bootstrapping) к предлагаемому методу: автоматное программирование – это проблемно-ориентированный язык. Абстракт-

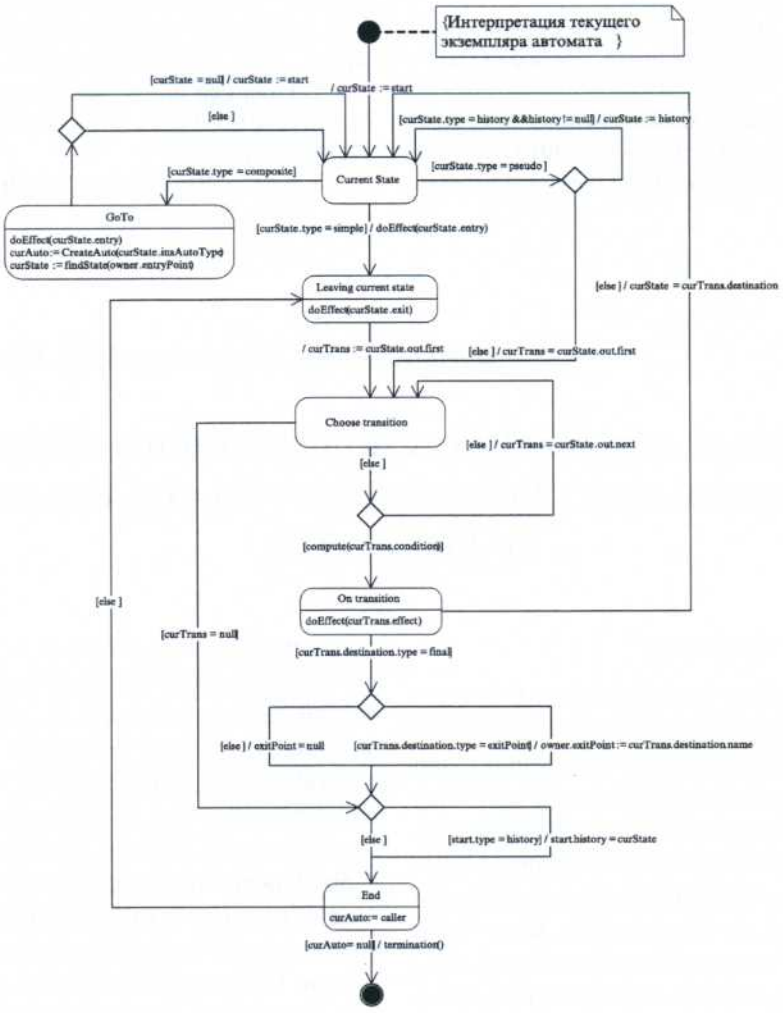


Рис. 4. Алгоритм интерпретации (семантика) автоматной программы

ный синтаксис этого языка определен в виде метамодели автоматной программы. Семантика выполнения автоматной программы описана в виде автоматной программы (рис. 4). Конкретный синтаксис автоматной программы задан с помощью автоматной программы, распознающей конкретное представление автоматной программы.

Реализованная согласно этой спецификации машина автоматного программирования была применена для интерпретации этой же спецификации.

Заключение

С помощью рассмотренного метода определения проблемно-ориентированных языков удалось описать наиболее сложный с точки зрения семантики фрагмент языка СЛОН – табличные выражения. Машина автоматного программирования позволяет получить языковый процессор этого фрагмента. Рассмотренный в статье метод показал свою выразительную мощность и возможность настройки на предметную область.

СПИСОК ЛИТЕРАТУРЫ

1. **Martin Fowler** "Language Workbenches: The Killer-App for Domain Specific Languages?" www.martinfowler.com/articles 12 June 2005

2. **Sergey Dmitriev** "Language Oriented Programming: The Next Programming Paradigm" www.onboard.jetbrains.com February 2005

3. **Оллонгрен А.** "Определение языков программирования интерпретирующими автоматами", Москва Мир 1977

4. **Шалыто А. А.** Сайт по автоматному программированию и мотивации к творчеству http://is.ifmo.ru/ СПбГУИТМО Кафедра "Технологии программирования"

5. **Гуров В. С., Мазин М. А., Шалыто А. А.** "Автоматическое завершение ввода условий в диаграммах состояний", http://is.ifmo.ru/works/_2008-02-28_auto_stop.pdf 2008

6. **Krasinsky G. A., Novikov F. A., Skripnichenko V. I.** "Problem Oriented Language for Ephemeris Astronomy and its Realization in System ERA" Cel. Mech., 1989. Vol. 45, P. 219–229

7. **Новиков Ф. А.** "Архитектура системы ЭРА – табличный подход к обработке данных" Ленинград ИПА РАН, 1990

8. **Буч Г., Якобсон А., Рамбо Дж.** UML. 2-е издание. СПб., Питер, 2006

Дунаев А. В., Ларченко А.В., Бухановский А.В.

Инструментальная оболочка поддержки принятия решений разработчика высокопроизводительных приложений в Грид-среде

Введение

На современном этапе развития технологий распределенных вычислений и систем понятие Грид подразумевает географически распределенную, согласованную, открытую и стандартизованную среду разделения вычислительных и информационных ресурсов [1]. Оно обобщает частные определения, независимо сформулированные I. Foster и C. Kesselman [2] и M. Livny [3], которые по мере внедрения технологий Грид в научно-исследовательскую и производственную практику расширяются и модифицируются. В настоящее время наиболее привычной является инфраструктурная интерпретация Грид как среды, предоставляющей совокупность высокопроизводительных вычислительных ресурсов для выполнения независимых задач многих пользователей [4]. Однако в рамках концепции [1] допустим альтернативный подход, который рассматривает Грид как специфическую ("мягкую", *soft*) среду параллельных вычислений наряду с более традиционными (например, кластерными, гибридными, P2P) параллельными архитектурами.

Проблема эффективной организации параллельных вычислений в Грид связана с не-

обходимостью объединения и синхронизации большого количества вычислительных систем для решения одной задачи. Ее решение связано с учетом ряда специфических факторов, в общем случае отрицательно влияющих на параллельную производительность. К ним относятся неоднородность вычислительных ресурсов и стохастическая изменчивость параметров коммуникационных сетей и вычислительных систем, характеризуемая нестационарным поведением во времени [5]. Потому принципы проектирования эффективных параллельных вычислительных приложений в Грид существенно отличаются от традиционных подходов, характерных, например, для кластерных систем [6]. Вследствие этого "ручной" процесс разработки таких приложений является весьма трудоемким и требует высокой квалификации разработчиков. Это ограничение может быть отчасти устранено за счет развития специализированного программного инструментария (технологических сред и оболочек проектирования), в частности инструментальной оболочки семейства PEG (Parallel Execution on Grid), представляемой в данной статье.

Интеллектуальный подход к задаче проектирования высокопроизводительных приложений для Грид-архитектур

Целью проектирования является формализация и обоснование внутренней структуры приложения (элементов и связей между ними), обеспечивающей оптимальные значения пользовательских характеристик при заданных ограничениях. Применительно к задачам высокопроизводительных вычислений такими характеристиками являются общее время выполнения, а также связанные с ним параллельное ускорение и эффективность. Эти величины совокупно зависят от особенностей применяемых параллельных алгоритмов, характеристик исходных данных и специфики параллельной вычислительной архитектуры [7].

Отличительной особенностью процесса проектирования приложений в Грид является изначальная ориентация на использование предметно-ориентированных вычислительных компонентов – прикладных Грид-сервисов (ПГС). Под ПГС подразумевается прикладная программа, зарегистрированная в Грид посредством ее оборачивания (*wrap*) соответствующей сервисной оболочкой, которая позволяет использовать ПГС другим системным и прикладным сервисам. Таким образом, решение задачи проектирования сводится к поиску необходимых пользователю сервисов и их взаимной организации в рамках соответствующего композитного приложения. При этом базовыми механизмами формирования оптимальной параллельной программной архитектуры такого приложения являются выбор конкретных реализаций сервисов, выбор эффективных способов распараллеливания по каждому из сервисов (внутренний параллелизм), выбор эффективных способов распараллеливания на уровне композитного приложения, статическая и динамическая балансировки, миграция задач между узлами в Грид.

В общем случае преимущества выбора (или тем более комбинации) механизмов формирования оптимальной архитектуры неочевидны в силу относительно слабой формализации требований и ограничений, изменчивости внешней среды (в данном случае инфраструктуры Грид) и неопределенности характеристик задачи. Как следствие, решение такой задачи целесообразно осуществлять не формальными методами, а средствами искусственного интеллекта [8]; при этом программный инструментарий поддержки процесса проектирования в заданной предметной области должен быть реализован в форме интеллектуальной системы реального времени.

На рис. 1 приведена концептуальная схема, иллюстрирующая принцип действия интеллектуальной системы поддержки принятия решений (ИСППР) разработчика высокопроизводительных приложений в Грид. Человеко-компьютерное взаимодействие в ИСППР осуществляется посредством графической нотации *workflow* (WF), или потока задач [9]. Пользователь описывает композитное приложение в форме мета-*workfow* (MWF). MWF представляет собой формальное описание пользовательской задачи в терминах предметной области; в общем случае – без указаний условий ее реализации в Грид. На первом этапе работы ИСППР выполняется интерпретация MWF, на основании которой осуществляется семантический поиск сервисов, зарегистрированных в Грид, в рамках заданной предметной области.

Таким образом, создается набор *активных* ПГС, установленных на конкретных вычислительных системах, входящих в Грид и готовых к использованию. Результатом является набор

абстрактных *workflow* (AWF), которые отличаются от MWF тем, что нечеткое описание сервисов в терминах предметной области заменено конкретными реализациями ПГС. Второй этап работы ИСППР связан с созданием оптимального расписания, которое обеспе-

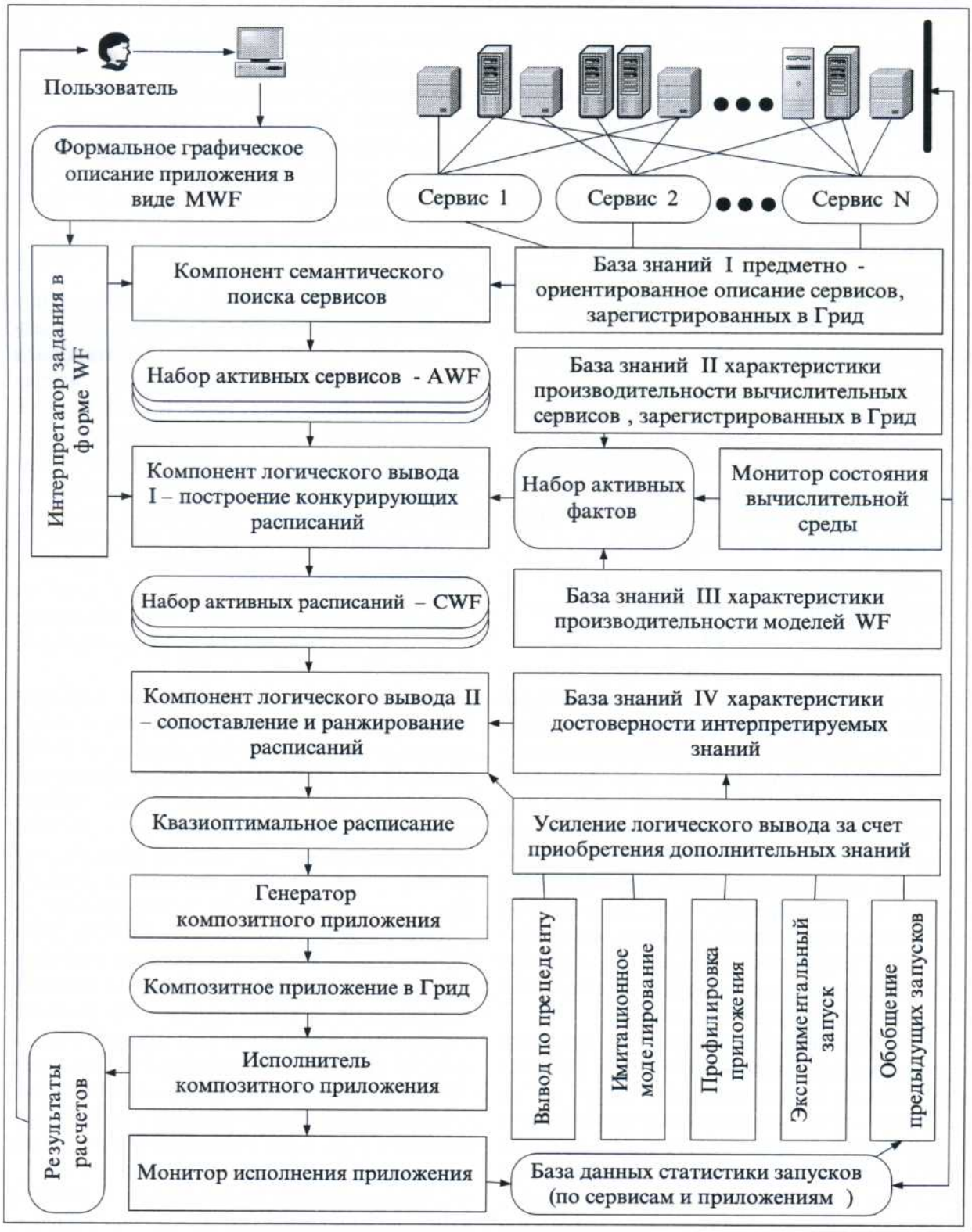


Рис. 1. Концептуальная схема ИСППР разработчика высокопроизводительных приложений в Грид

чивало бы наибольшую производительность. Для этого используются экспертные знания о производительности ПГС, предоставляемые их разработчиками при регистрации сервиса в Грид, например, в форме параметрической модели времени выполнения в зависимости от параметров задачи и характеристик вычислительной системы. Используя данные мониторинга текущего состояния Грид и известные характеристики пользовательской задачи, определяются активные факты в форме времени выполнения конкретных ПГС. Эти факты используются для построения системы конкурирующих расписаний; при этом конкуренция обусловлена как различными эвристиками, применяемыми для ускорения процесса построения расписания, так и разнообразием AWF, порождаемых одним MWF. Как следствие, выбор оптимального расписания требует выполнения процедуры сопоставления альтернатив путем их ранжирования в условиях неопределенности входных данных, стохастической изменчивости параметров Грид и экспертного характера знаний о производительности ПГС. В результате пользователю предлагается один или несколько (в том случае, если достоверно нельзя отдать приоритет ни одному) конкретных workflow (CWF). Каждый CWF является полным описанием параллельного выполнения задачи в Грид. Таким образом, пользователь может непосредственно трансформировать CWF в соответствующие коды и отправить на выполнение в Грид или использовать автоматически сгенерированные коды как каркас для дальнейшей "ручной" доработки. В результате выполнения приложения пользователь получает результат в виде набора данных, которые отправляются в указанное хранилище Грид или на машину пользователя.

Существенным отличием данной ИСППР реального времени от традиционных экспертных систем является наличие адаптивной компоненты, которая позволяет усиливать процедуру логического вывода системы за счет приобретения дополнительных знаний в ходе ее функционирования в условиях постоянно изменяющейся среды. Дополнительные знания могут приобретаться косвенным путем за счет поиска по прецеденту, имитационного моделирования, профилировки приложения, а также экспериментальных измерений производительности отдельных ПГС. При этом динамически формируется и обновляется база данных статистики предыдущих запусков; ее анализ позволяет не только уточнить интерпретируемую информацию (например, коэффициенты моделей производительности), но и судить о качестве знаний, предоставляемых экспертами, регулируя соответствующие коэффициенты достоверности. Таким образом, предлагаемая ИСППР позволяет автоматизировать процесс разработки параллельных приложений в Грид даже в условиях существенной неопределенности, когда разработчик не сообщает достоверной информации о производительности своих ПГС.

Программная реализация: инструментальные оболочки семейства PEG (Parallel Execution on Grid)

Концептуальная схема ИСППР, приведенная на рис. 1, реализуется в семействе инструментальных оболочек PEG (Parallel Execution on Grid) поддержки принятия решений разработчика высокопроизводительных приложений в Грид. Семейство PEG включает в себя следующие образцы.

- PEG1: инструментальная оболочка параллельного исполнения приложений. PEG1 позволяет создавать композитные приложения из готовых или разрабатываемых ПГС и оперативно отслеживать процесс их исполнения в Грид.
- PEG2: инструментальная оболочка информационной и технологической поддержки процесса проектирования приложений. PEG2 поддерживает визуальную среду прототипирования композитного приложения в терминах CWF и предоставляет набор специфичных для Грид инструментов проектирования (монитор Грид, контроллер ресурсов, монитор исполнения и пр.).
- iPEG (Intelligent PEG): интеллектуальная среда, реализующая всю последовательность операций, отображенную на рис. 1, от задания пользователем MWF до выполнения композитного приложения, созданного на основе квазиоптимального CWF.

На рис. 2 представлен процесс эволюции архитектуры оболочек семейства PEG. В PEG1 реализована основная функциональность работы с инфраструктурой Грид на основе Intel Grid Programming Environment (GPE). Он использует библиотеки GPE API версии 1.5, которые в свою очередь являются надстройкой над средой Globus. Это решение позволяет

изолировать реализацию инструментальной оболочки от особенностей архитектуры Грид. Элемент исполнения в оболочках PEG отвечает за выполнение пользовательского задания на удаленных целевых системах: он осуществляет вызовы интерпретатора WF, балансировочных алгоритмов, а также производит мониторинг. В инструментальной оболочке PEG2 по сравнению с PEG1 вводятся дополнительные функциональные элементы, в частности визуальная оболочка проектирования, элемент представления WF, элемент мониторинга Грид и элемент прогнозирования времени выполнения приложений в Грид. Визуальная оболочка реализует процесс человеко-компьютерного взаимодействия в графической форме. Элемент представления WF, по сути, является интерпретатором пользовательской постановки задачи для дальнейшего использования остальными элементами программного комплекса. Элемент балансировки и построения расписаний использует элементы мониторинга среды Грид и прогнозирования времени работы композитного приложения для построения CWF, имеющего наибольшую производительность.

В интеллектуальной среде iPEG вводится ряд принципиально новых элементов, главным из которых является элемент управления знаниями. Он предоставляет основной функционал для работы с базами знаний, в том числе инструменты их использования, активизации, извлечения, пополнения и обновления, а также проверки целостности и достоверности. Дополнительно в iPEG входит элемент планирования, который необходим для построения набора активных расписаний с использованием семантического поиска сервисов по нечеткому пользовательскому описанию задачи в форме MWF. Элемент планирования взаимодействует с элементом управления знаниями, получая от него информацию о доступных сервисах и их онтологических описаниях. Это обеспечивает соответствие общей архитектуры оболочек семейства PEG (см. рис. 2) принципам работы ИСППР, изображенной на рис. 1.

Применение инструментальной оболочки iPEG для создания высокопроизводительных композитных приложений в Грид

Интеллектуальная среда iPEG позволяет осуществлять полномасштабную информационную поддержку принятия решений разработчика высокопроизводительных приложений в Грид. Проиллюстрируем основные принципы работы системы на примере композитного приложения, осуществляющего расчет климатических спектров морского волнения [10]. В процессе вычислений производится статистическое обобщение массива таблично заданных функций (24x25 значений) распределения энергии волн по частотам и направлениям в фиксированной точке пространства за длительный временной интервал. Каждой из таблиц сопоставляется нелинейная функция двух

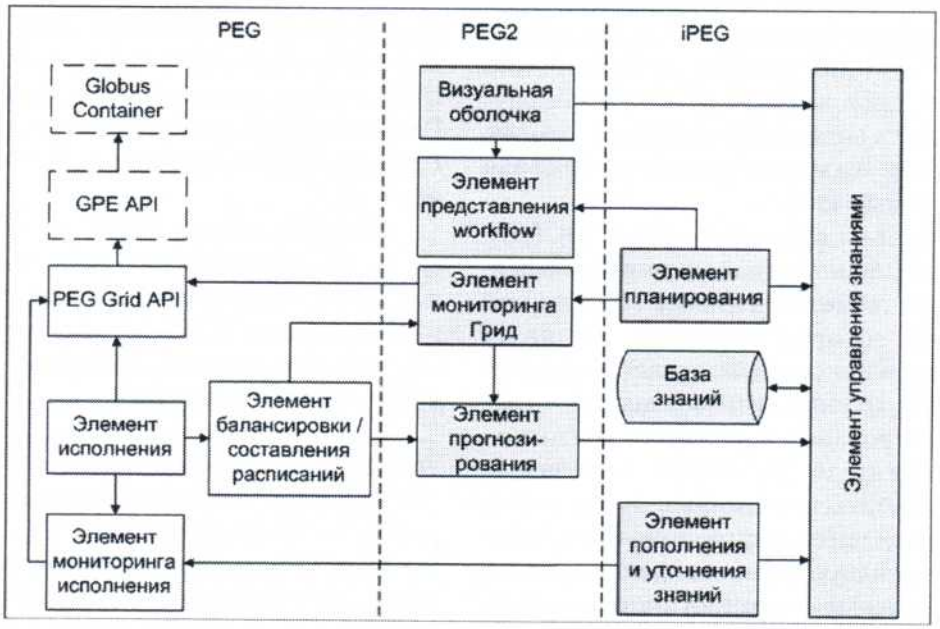


Рис. 2. Эволюция архитектуры оболочек семейства PEG

аргументов, параметры которой определяются методом адаптивного случайного поиска. Обработка таблиц ведется независимо, что позволяет выполнять распараллеливание вычислительного процесса по данным.

На рис. 3 приведены характеристики времени обработки массива из 8 тысяч спектров, полученные на экспериментальном стенде – корпоративной Грид-системе, состоящей из 24 целевых систем на базе процессоров Intel Pentium D и Intel Core 2 Quad.

По ходу работы в ИСППР, реализуемой инструментальной оболочкой iPEG, рассматривались три конкурирующих расписания: равномерное распределение данных, прямая каскадная схема и обратная каскадная схема, реализующие синхронные способы взаимодействия с вычислителями. Видно, что *в среднем* для небольшого количества вычислителей ($p = 2 \div 4$) каскадные схемы являются более предпочтительными; с увеличением p накладные расходы на осуществление синхронного взаимодействия возрастают, и, как следствие, предпочтительным становится использование равномерной схемы распределения данных по вычислителям.

На рис. 3 также приведены ядерные оценки плотности распределения времени работы для каждой из трех схем распараллеливания по данным, полученные путем обобщения результатов измерений. Видно, что с увеличением количества вычислителей разброс времени выполнения возрастает, что связано с ростом влияния стохастических эффектов коммуникаций в Грид. При этом взаимное расположение плотностей распределения определяет правила ранжирования конкурирующих расписаний. Например, в случае "А" (см. рис. 3) для $p = 2$ прямая и обратная каскадные схемы приводят в среднем к практически одинаковым оценкам времени работы. Различие между ними существенно меньше соответствующего диапазона изменчивости, что не позволяет автоматически выбрать наилучший способ. Однако разброс времени работы для прямой каскадной схемы в 1,5 раза больше, чем для обратной каскадной схемы. Как следствие, окончательный выбор схемы распараллеливания остается за пользователем, исходя из предпочитаемой им стратегии (меньший риск – меньшая производительность, больший риск – большая производительность).

С другой стороны, существуют ситуации (например, "Б" при $p = 24$), когда плотности распределения времени работы для разных схем перекрываются, однако различие между средними значениями сопоставимо с их диапазоном изменчивости. Тогда ранжирование конкурирующих расписаний должно вестись с учетом заданного уровня значимости ошибки; в частности, в ситуации "Б" в 5% случаев обратная каскадная схема будет давать лучший по

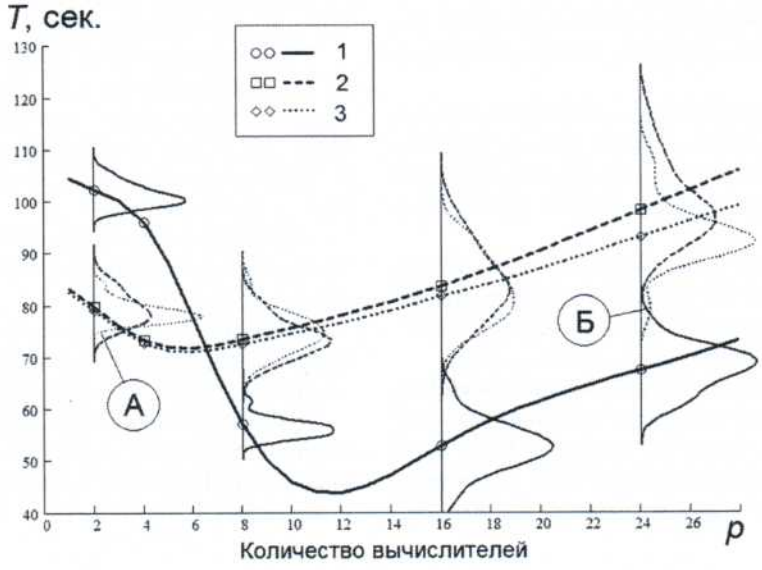


Рис. 3. Время выполнения иллюстративного композитного приложения по обработке 8 тысяч спектров морского волнения в Грид: 1 – равномерная схема; 2 – каскадная схема; 3 – обратная каскадная схема (А, Б – см. по тексту)

производительности результат, чем равномерная, несмотря на то, что *в среднем* равномерная схема работает в 1,5 раза быстрее.

Из рис. 3 также видно, что кривые производительности имеют ярко выраженный минимум (для каскадных схем при $p = 6$, для равномерной при $p = 12$), которые и определяют оптимальный режим выполнения задачи. Для рассмотренного примера максимальное ускорение в среднем составляет около 2,5 раз, а выигрыш по сравнению с конкурирующими каскадными расписаниями — в 2 раза. Столь невысокие показатели ускорения характерны для вычислений в Грид в силу существенного вклада коммуникационной составляющей; однако они являются только одним из определяющих факторов (наравне с объемом доступной памяти, дискового пространства и пр.), которые заставляют пользователей обратиться к технологиям распределенных вычислений и систем.

Таким образом, в ходе работы интеллектуальной системы iPEG рассмотренный выше анализ проводится автоматически, что позволяет обосновать выбор квазиоптимальной схемы распараллеливания композитного приложения применительно к доступным в Грид вычислительным ресурсам и ПГС.

Заключение

Проектирование высокопроизводительных приложений в Грид является сложным и ресурсоемким процессом, требующим высокой квалификации разработчика и специализированного программного инструментария. Представленное в статье семейство инструментальных оболочек PEG позволяет реализовать интеллектуальный подход к решению задачи проектирования в рамках последовательности MWF-AWF-CWF, автоматизируя процесс разработки композитных приложений на основе ПГС и гибко управляя их параллельной производительностью, исходя из конкретных условий эксплуатации инфраструктуры Грид.

Работа выполнена в рамках НИР 2007-4-1.4-20-01-025 "Разработка инструментальной оболочки проектирования высокопроизводительных приложений для Грид-архитектур в целях создания прикладных сервисов компьютерного моделирования и обработки данных".

СПИСОК ЛИТЕРАТУРЫ

1. **Stockinger H.** Defining the Grid — a snapshot of the current view // Journal of Supercomputing. — Springer Science+Business Media, 2007, P. 3–17.

2. **Foster I., Kesselman. C.** The Grid: Blueprint for a New Computing Infrastructure. Morgan-Kaufman, 1999, 701 p.

3. **Thain D., Tannenbaum T., and Livny M.** Condor and the Grid // Berman F., (Editor), Fox G. (Editor), Hey J.G. A. (Editor) Grid Computing: Making The Global Infrastructure a Reality. – John Wiley, 2003, P. 299–335

4. **Дмитриев В.Ю.** Сети высокопроизводительных вычислений — новое направление ИТ // Информационный бюллетень. JetInfo. Выпуск 175 — М.: компания "Инфосистемы Джет", 2007. — 28 с.

5. **Дунаев А.В., Ларченко А.В., Бухановский А.В.** Моделирование параллельных вычислительных процессов в среде Грид на примере Intel Grid Programming Environment // ПаВТ 2008 — сборник трудов (электронное издание), 2008. — С. 383–389.

6. **Гергель В.П.** Теория и практика параллельных вычислений: учебное пособие — М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. — 423 с.

7. **Ковальчук С. В., Иванов С. В., Колыхматов И. И., Бухановский А. В.** Особенности проектирования высокопроизводительных программных комплексов для моделирования сложных систем // Информационно-управляющие системы, 2008, № 3. — С. 10–18.

8. **Нечаев Ю.И.** Искусственный интеллект: концепции и приложения. — СПб.: Изд. СПбГМТУ, 2000. — 294 с.

9. **Blythe J., Deelman E.** et al. The Role of Planning in Grid Computing // ICAPS 2003, [http://www.isi.edu/~gil/papers/icaps03-submission.pdf].

10. **Boukhanovsky A.V., Lopatoukhin L.J.**, Guedes Soares C. Spectral wave climate of the North Sea // Applied Ocean Research, Volume 29, Issue 3, July 2007, P. 146–154

Рачинский Е. С.

Программная среда для оценки производительности клиент-серверных систем при нагрузочном окружении

Введение

В связи с развитием информационных систем и технологий, ростом уровня информатизации многих видов человеческой деятельности увеличиваются и нагрузки на информационные системы. Учитывая эти изменения и факт непрерывного увеличения сложности и масштабов программных комплексов, вопросы обеспечения требуемой производительности клиент-серверных систем становятся особенно актуальными. В программной инженерии задачами производительности занимается отдельная дисциплина – инженерия производительности ПО – ИПП. ИПП – это набор методов поддержки разработки ПО, ориентированных на аспекты, связанные с производительностью, при этом данный набор методов покрывает все фазы цикла разработки. Главная цель ИПП – гарантировать требуемое качество ПО, относящееся к производительности [1]. ИПП включает в себя моделирование, анализ и планирование производительности, а также тесно соприкасается с такими областями инженерии ПО как проектирование, обеспечение качества и сопровождение. Подробное обсуждение задач и методов инженерии производительности программного обеспечения можно найти в [1, 2, 10].

Для решения разнообразных задач ИПП, помимо методов аналитического и имитационного моделирования, также активно применяются методы непосредственного измерения показателей производительности при помощи экспериментов. Непосредственные измерения весьма эффективны для анализа в случае наличия готовой системы или ее прототипа. Суть этих методов заключается в создании нагрузочного окружения и измерении как внешних, так и внутренних показателей производительности. Эти методы используется в стресс-тестировании, в оценке производительности прототипов систем, в определении параметров моделей производительности, в определении качества обслуживания (QoS) и во многих других задачах ИПП.

Основная проблема в этой области заключается в отсутствии универсального подхода к созданию нагрузочного окружения и автоматизации задач ИПП. Это связано с тем, что нагрузочное окружение существенно зависит от исследуемой системы и от целей исследования. Например, различные коммуникационные технологии требуют разных подходов к генерации потока запросов и измерению показателей производительности. Существующие подходы и инструменты в основном предназначены для стресс-тестирования систем с использованием простых протоколов и не покрывают задачи ИПП комплексно. Это зачастую вынуждает отказаться от применения методов оценки и анализа производительности программных систем в процессе промышленной разработки.

В данной работе для решения этой проблемы предлагается разрабатывать инструментарий в зависимости от конкретных задач создания нагрузочного окружения и особенностей исследуемой системы. Этот метод основан на использовании программной платформы, предоставляющей базовые сервисы, используя которые разработчик или тест-инженер имеет возможность быстро создать требуемый инструментарий. Необходимо отметить, что мы предлагаем не очередной инструмент, а подход к созданию таких инструментов, учитывающих специфику проекта. Преимущества подхода заключаются в гибкости и адаптируемости под конкретные нужды. Применение этого подхода позволяет программно контролировать все аспекты создания нагрузочного окружения. В качестве примера использования этого подхода можно привести возможность программно изменять характер нагрузки в зависимости от реакции целевой системы. А это существенно упрощает автоматизацию задач ИПП, связанных с оценкой характеристик производительности информационных систем.

Подход разрабатывается автором в исследовательском отделении компании Сименс (Германия, Мюнхен). Результаты исследований применяются в реальных промышленных проектах.

Имитация нагрузочного окружения

Нагрузочное окружение – это совокупность всех потоков запросов к системе. Имитация нагрузочного окружения применяется для оценки производительности систем путем генерации потоков запросов и измерении внешних и внутренних характеристик производительности. Среди задач, для которых применяется имитация нагрузочного окружения, можно выделить три основные группы.

1. Тестирование производительности.

• Определение показателей производительности при заданной нагрузке.

• Оценка стабильности и корректности функционирования системы под нагрузкой.

• Стресс тестирование и определение значений ключевых характеристик производительности (QoS).

2. Анализ производительности.

• Сравнение производительности альтернативных конфигураций системы.

• Анализ факторов, влияющих на производительность.

• Поиск узких мест в системе ("bottlenecks").

• Планирование производительности и мощностей.

3. Поддержка моделирования производительности.

• Вывод параметров аналитических и имитационных моделей системы.

Общая процедура исследования производительности под имитируемой нагрузкой состоит из следующих трех шагов.

I. Моделирование нагрузки в зависимости от целей исследования.

II. Генерация нагрузки и измерение требуемых внешних и внутренних показателей производительности.

III. Анализ результатов измерений в соответствии с поставленными задачами исследования.

Моделирование нагрузки включает в себя определение исследуемой функциональности и вероятностных характеристик потока запросов. Например, для определения среднего времени ответа сайта электронной коммерции описывается последовательность HTTP запросов к вебстраницам и распределение времени ожидания между запросами. Для моделирования применяются как визуальные модели [2], например диаграммы описания поведения пользователя, так и аналитические модели, например фрактальные модели сетевого трафика [3]. Для генерации нагрузки используются специализированные приложения, осуществляющие запросы к системе в нескольких программных потоках. Существенной сложностью является генерация потоков с высокой интенсивностью таким образом, чтобы генератор запросов сам не стал узким местом и не повлиял бы на результат измерений. В связи с этим генераторы нагрузки имеют сложную и, как правило, распределенную архитектуру. Одновременно с генерацией нагрузки измеряются различные показатели производительности, например время ответа каждого запроса, загрузка процессора, частота операций ввода-вывода и т.д. В зависимости от задач эксперимента показатели могут быть статистически обработаны в процессе выполнения. Например, для управления стресс-тестом необходимо наблюдать за изменением среднего времени ответа с ростом нагрузки. После завершения эксперимента собранные измерения подвергаются статистическому анализу. Например, для поиска зависимости между средним временем ответа и потреблением аппаратных ресурсов.

Несмотря на то, что основные шаги достаточно просты, в деталях они сильно зависят от целей исследования и типов систем, для которых создается нагрузочное окружение. Подобная ситуация часто встречается в областях программной инженерии, связанных с поддержкой разработки крупных программных систем, например в управлении сборкой и тестировании.

Для решения этой проблемы автором предлагается подход, заключающийся в разработке инструментария имитации нагрузочного окружения в зависимости от особенностей исследуемой системы и целей эксперимента. Для поддержки этого подхода предлагается специализированная программная платформа, которая облегчает создание такого инструментария. Платформа предоставляет базовые сервисы и точки расширения, используя которые можно создавать требуемый инструментарий для любых задач экспериментального исследования производительности.

Существующие подходы

На данный момент существует несколько десятков программ, используемых для создания нагрузочного окружения. Наиболее популяр-

ные среди них HP Loadrunner, Apache JMeter, Open STA, Empirix E-Loader, Visual Studio Team System, Grinder, Rational Performance Tester; TPTP (полное перечисление можно найти в [4]). Большинство из них предназначены для стресс-тестирования веб-приложений и слабо адаптируемы для других типов систем и задач ИПП. Общий недостаток существующих методов — это отсутствие комплексного подхода к задачам инженерии производительности ПО, где все аспекты, связанные с особенностями исследуемой системы, методами измерений и целями создания нагрузки, могли бы быть одновременно учтены.

Из главных недостатков существующих инструментов необходимо выделить следующие.

• Отсутствие возможности создавать нагрузки, соответствующие открытой системе массового обслуживания, а также сложность в управлении статистическими параметрами потока запросов. В работе [5] отмечалось, что характеристики времени ответа могут существенно различаться для открытой и закрытой нагрузкок.

• Сложность интеграции автоматизированных процедур нагрузочного тестирования с другими методиками промышленной разработки ПО. Например, большинство существующих средств не позволяют внедрить автоматическое тестирование производительности в процедуру регулярной сборки или в процедуру автоматизированного тестирования.

• Отсутствие возможности изменять параметры нагрузочного окружения в зависимости от реакции системы. В этом мы видим ключевой недостаток, так как он не позволяет применять автоматические методы оценки производительности, например: автоматическое определение максимальной пропускной способности клиент-серверного приложения.

• Сложность расширения функциональности инструментария, например: добавление новых видов статистической обработки данных во время генерации нагрузки или поддержка специфических видов клиент-серверного взаимодействия, таких как асинхронные вызовы.

Указанные выше недостатки часто вынуждают отказаться от использования практик оценки производительности, либо ведут к некорректному применению инструментов и неверной интерпретации результатов.

Среда создания нагрузочного окружения

Среда представляет собой платформу времени исполнения, предоставляющие базовые сервисы и набор точек расширения. Адаптация либо создание инструментария заключается в разработке программных модулей, отвечающих за отдельные аспекты нагрузочного окружения и выполнения измерений. Платформа обращается к модулям через механизм расширений. Этот архитектурный принцип называется "обращение контроля" и используется для уменьшения связанности между компонентами [6]. Компоненты позволяют реализовывать следующие области функциональности:

• сценарий нагрузки;

• описание нагрузки;

• механизм точного измерения показателей производительности;

• механизм статистической обработки измерений в режиме реального времени;

• управление шагами эксперимента (старт, остановка и т.д.).

Платформа также предоставляет следующие базовые сервисы, используемые компонентами:

• генератор нагрузки, обеспечивающий выполнение большого количества одновременных запросов;

• передача, хранение и обработка результатов измерений;

• управление жизненным циклом компонент.

На рис. 1 представлена принципиальная схема платформы. Платформа состоит из двух основных частей: консоли управления и генераторов нагрузки. Далее будут подробно описаны основные области расширений.

Сценарий нагрузки

Сценарий нагрузки – это последовательность инструкций для выполнения запросов к исследуемой системе. Источником сценария обычно является модель нагрузки. В зависимости от целей создания нагрузки модель может быть основана на реальных данных, полученных путем анализа трасс-запросов к системе, либо сформирована искусственно, что часто называется синтетической нагрузкой. Большинство типов моделей состоят из следующих иерархически связанных элементов: сценарий, шаг (часто называемый транзакцией) и запрос (рис. 2).



Рис. 1

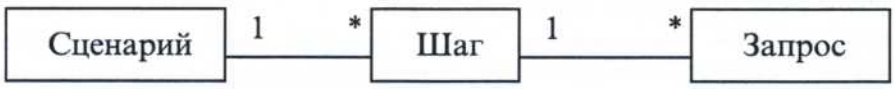


Рис. 2

В зависимости от типа модели элементы могут иметь различную семантику. Например, сценарий может описывать бизнес-процесс, где шаг – это действие в бизнес-процессе, состоящее из низкоуровневых вызовов удаленных функций. В случае веб-систем сценарий описывает поведение пользователя веб-сайта, где шаги – это страницы, состоящие из HTTP – запросов к ресурсам представления (CSS, JavaScript, HTML). Необходимо также отметить, что описанная иерархичная модель характерна для интерактивных систем. В случае интерфейса удаленного вызова процедур модель может быть упрощена до одного запроса в сценарии.

Разнообразные инструменты нагрузочного тестирования предлагают описывать сценарии на специализированных скриптовых языках, придерживаясь строго заданной структуры. В данном подходе предлагается описывать сценарий нагрузки на высокоуровневом языке программирования, таком как Java. Это позволяет запрограммировать любую логику сценария, а также использовать готовый набор средств разработки и отладки. Для отображения элементов моделей на код сценария используется три типа классов: сценарий, транзакция и запрос. Каждый из классов имеет абстрактный метод 'run'. Время выполнения этого метода регистрируется как время выполнение сценария, транзакции и запроса соответственно. На рис. 3 представлен пример простейшего сценария.

```
public class Service1Scenario extends
WebScenario {

Transaction1 transaction1 = new Transaction1();

 @Override
 public void onStart() {
  //Initialization
 }
 @Override
 public void onStop() {
  //Post mortem
 }
 @Override
 public void run() {
  runTransaction(transaction1);
  sleep(1000); //sleep for awhile
  runTransaction(transaction2);
  }
}
```

```
public class Transaction1 extends WebTransaction {
 Request1 request1 = new Request1 ();
@Override
 public void run() {
  runRequest(request1);
 }
}
```

```
public class Request1 extends WebRequest {
  final String url = "http://noname";
 @Override
 public void run() throws Exception {
          HttpResponse response = null;
          response =
              getContext().
                  getClient().execute(url     );
  }
}
```

Рис. 3

Описание нагрузки

В отличие от сценария нагрузки, который описывает функциональную часть нагрузочного окружения, описание нагрузки соответствует количественной части, то есть специфицирует, с какой интенсивностью поступают запросы. В зависимости от задач исследования параметры нагрузки могут быть выведены из трасс запросов и соответствовать реальной нагрузке или так же могут быть заданы искусственно, что называется синтетической нагрузкой. В данной работе предлагается три различных способа описания нагрузки.

Первый способ основан на модели закрытой системы массового обслуживания (СМО) [7], где генератор запросов и обслуживающий сервер представляют собой одну саморегулирующуюся систему. В этом случае нагрузка описывается как две функции времени: $f_{user}(t) = n$, $f_{think}(t) = T$, где $f_{user}(t)$ определяет количество программных потоков ("виртуальных пользователей"), исполняющих запросы в цикле, а $f_{think}(t)$ определяет вид и параметры распределения "времени обдумывания" между отдельными итерациями цикла. Параметр t соответствует времени, прошедшему с начала эксперимента. Если как пример взять стресс-тест, то $f_{user}(t)$ определяется как равномерно возрастающая ступенчатая функция, а $f_{think}(t)$ не зависит от времени. Эти функции могут быть описаны как аналитически, так и таблично.

Второй способ основан на модели открытой СМО [7]. В этом случае нагрузка описывается как $f_{rate}(t) = \lambda$, где $f_{rate}(t)$ описывает статистические характеристики потока запросов. Генерация запросов по такому описанию требует большого количества активных программных процессов. Для снижения нагрузки на сам генератор до начала эксперимента мы генерируем последовательность временных точек, в которые инициируются запросы. Эта последовательность раскладывается на несколько подпоследовательностей и распределяется по разным генераторам нагрузки.

Третий способ принципиально отличается от первых двух. В нем статистические параметры потока запросов регулируются в зависимости от реакции системы. Механизм регуляции может быть различным и зависит от целей создания нагрузки. Механизм описывается программно и использует сервис статистической обработки для наблюдения за исследуемой системой (см. следующую главу). Простым примером может служить задача определения среднего времени ответа системы с заданным доверительным интервалом при заданной нагрузке. В этом случае запросы генерируются до тех пор, пока объем выборки времен ответа не станет достаточным для определения среднего времени ответа с заданным доверительным интервалом.

Представленные три способа описания нагрузки позволяют решить задачи генерации нагрузки, максимально приближенной к реальной, симулировать "всплески нагрузки", создавать поток запросов, имеющий фрактальный

ную структуру, генерировать поток запросов с заданными статистическими характеристиками и автоматизировать задачи оценки производительности.

Измерение показателей производительности

Главной регистрируемой величиной является время исполнения отдельного запроса. Одновременно с ней регистрируется и абсолютное время начала запроса. Пример журнала регистраций приведен в таблица 1. Конкретный набор регистрируемых значений зависит от протокола взаимодействия и особенностей конкретной системы. Например, для HTTP запроса среди прочих значений регистрируется также и время, соответствующее определению IP адреса по доменному имени. Регистрируемые значения объединяются в статистические величины на заданных интервалах времени. Предлагаемая платформа предоставляет средства для расчета следующих статистических величин.

Выборочное среднее. Для каждого интервала времени длины l, для регистрируемой величины R вычисляются $S = \Sigma r_i, S = \Sigma r_i^2, t_{start}, n, l$, где n – количество измерений на заданном интервале, r_i – i-ое значение R, t_{start} – абсолютное время начала интервала. Тогда выборочное среднее равно $\overline{R} = \frac{S}{n}$ и выборочная дисперсия равна $\overline{V} = \frac{SS}{n} - \left(\frac{S}{n}\right)^2$. Преимущество такого представления заключается в том, что для рядом стоящих интервалов можно легко вычислить общее выборочное среднее и дисперсию. Пример: возьмем интервалы I_1, I_2 и соответствующие величины S_1, SS_1, t_{start_1}, n_1, l_1; S_2, SS_2, t_{start_2}, n_2, l_2, где $t_{start_1} + l_1 = t_{start_2}$, тогда на объеденном интервале времени $I_3 = I_1 + I_2$ $\overline{R} = \frac{(S_1 + S_2)}{n_1 + n_2}$, $\overline{V} = \frac{SS_1 + SS_2}{n_1 + n_2} - \left(\frac{(S_1 + S_2)}{n_1 + n_2}\right)^2$. Выборочное среднее применяется для таких измерений, как время ответа.

Средняя частота событий. Пускай на промежутке времени $[t_{start}, t_{stop}]$ произошло n событий в последовательные точки времени $t_1, t_2, \ldots, t_n$, $t_i \in [t_{start}, t_{stop}]$, тогда средняя частота событий (среднее число событий в единицу времени) равна $\lambda = \frac{n}{t_{stop} - t_{start}}$. Так как для каждого интервала времени сохраняется t_{start}, t_{stop}, n, то расчет средней частоты событий на двух соседних интервалах I_1, I_2 происходит следующим образом: $\lambda_3 = \frac{n_1 + n_2}{t_{stop_2} - t_{start_1}}$. Частота событий применяется для расчета средней частоты запросов, пропускной способности системы (транзакций/сек.), частоты отказов и т.д. Перерасчет средних значений на больших интервалах позволяет, например, легко увидеть тенденцию изменения наблюдаемой величины. При малых интервалах времени и большой дисперсии общая тенденция обычно плохо видна на графике. Компактное представление позволяет пересылать значения с удаленных генераторов нагрузки на центральную консоль мониторинга во время проведения эксперимента.

Платформа предоставляет возможность рассчитывать средние статистические значения для любых регистрируемых значений. Одновременно с этим предоставляется возможность создавать новые типы статистических величин.

Реализация

Платформа реализована на языке программирования Java с применением компонентной технологии OSGi [8]. OSGi обладает следующими преимуществами: простая компонентная модель, возможность динамической загрузки и выгрузки компонент в режиме реального времени, управление версиями компонент, управление зависимостями между компонентами, механизм мета-описания компонент. Платформа состоит из двух основных частей:

Таблица 1

ID потока	Итерация	Время начала запроса (t_i)	Продолжительность исполнения (r_i)	Код ошибки
51	0	3256750	141	0
63	2	3256781	140	0
13	3	3256813	156	0

консоли управления и генератора нагрузки. Обе части представляют собой отдельные контейнеры приложений, где приложение – это набор связанных компонент, каждая из которых отвечает за описанные выше области функциональности. Консоль управления реализована в виде расширений Eclipse [9] и взаимодействует с распределенными генераторами нагрузки посредствам TCP/IP. Консоль управления предоставляет точки расширения для отображения графиков измерений, обработки и хранения измерений, взаимодействия с пользователем и управления экспериментом. На стороне генератора программные модули отвечают за сценарий нагрузки, регистрацию значений, поддержку протокола взаимодействия с исследуемой системой, реализацию статистических преобразований. Перед началом эксперимента программные модули автоматически загружаются в соответствующие контейнеры.

Среда разработки Eclipse и набор сопутствующих технологий существенно облегчают создание и отладку компонент предлагаемой платформы. Благодаря компонентной технологии OSGi возможно повторное использование компонент платформы в различных проектах и типах экспериментов.

Заключение

В статье представлен новый подход к разработке инструментария создания нагрузочного окружения. Показано, что применение универсальной платформы существенно упрощает разработку и использование данного инструментария. В качестве дальнейших исследований автор видит разработку полностью автоматизированных методов тестирования и анализа производительности, основанных на предлагаемой платформе.

СПИСОК ЛИТЕРАТУРЫ

1. Performance Engineering: state of the art and current trends / Reiner Dumke (ed.). – Berlin: Springer, 2001 (Lecture notes in computer science; Vol. 2047)

2. Производительность Web-служб. Анализ, оценка и планирование. Пер. с англ./ Даниел Менаске, Алмейда Виргилио. — СПб: ООО "ДиаСофтЮП", 2003 – 480с.

3. Traffic Model and Performance Evaluation of Web Servers, Zhen Liu and Nicolas Niclausse. Performance Evaluation. Volume 46 , Issue 2-3 (October 2001)

4. Load and Performance Test Tools - www.softwareqatest.com/qatweb1.html, интернет ресурс

5. Open versus closed: a cautionary tale, Bianca Schroeder and Adam Wierman and Mor Harchol-Balter, Proceedings of the 3rd conference on Networked Systems Design & Implementation — Volume 3, 2006

6. Design Patterns Explained: A New Perspective on Object-Oriented Design (2nd Edition), Alan Shalloway. — Addison-Wesley, 2004

7. **Матвеев В.Ф., Ушаков В.Г.** Системы массового обслуживания. М.: МГУ, 1984

8. OSGi Service Platform, Core Specification, Release 4, Version 4.1, ISBN 978-90-79350-01-8

9. Расширение Eclipse: принципы, шаблоны и подключаемые модули, Эрик Гамма, Кент Бек. — КУДИЦ-Образ, 2005 г.

10. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling, R. K. Jain. – Wiley, 1991

Старостенко А.С., Самочадин А.В.

Программные средства дистанционного управления очередями потребителей

Введение

Одним из подходов к улучшению процесса обслуживания для различных видов сервисов является автоматизация процесса прохождения очереди. Системы управления очередями позволяют эффективно организовать работу персонала, сократить время обслуживания клиентов и повысить качество обслуживания.

В статье описывается подход к созданию системы управления очередями, основанный на использовании Интернет и SMS-сервисов для выполнения основных функций системы.

Системы управления очередями

Существующие системы управления очередями ориентированы на использование специального оборудования (номерковые автоматы, электронные табло и т.д.). Основными областями применения электронных очередей на базе номерковых аппаратов являются операционные залы банков, центры выплат страховых компаний, клиентские центры операторов сотовой связи, розничные торговые сети, салоны красоты, медицинские центры, автосалоны, автосервисы, нотариальные и адвокатские конторы, посольства и консульские учреждения, авиа- и железнодорожные кассы, центры трудоустройства и т.д.

Такие системы обладают множеством достоинств.

1. Клиент избавлен от ожидания в очереди. Взяв талон с номером, он может находиться в любом месте операционного зала. Система сообщает ему о состоянии очереди.

2. Можно обеспечить конфиденциальность за счет отсутствия посторонних людей в момент обслуживания.

3. Улучшение условий труда обслуживающего персонала позволяет более эффективно планировать работу.

4. Система позволяет более эффективно администрировать обслуживание, постоянно выдавая информацию о количестве посетителей. При необходимости тем или иным услугам может быть отдан приоритет. Статистика, собранная управляющим компьютером, дает возможность оптимальным образом комплектовать организацию персоналом.

Более подробно способы организации обслуживания с помощью специализированного оборудования рассмотрены в [2].

В то же время применение систем такого типа ограничено рядом факторов. Определяющим ограничением является необходимость использования дорогого оборудования, которое требует постоянной высококвалифицированной технической поддержки.

К функциональным недостаткам систем этого типа, ограничивающим их использование, следует отнести:

– необходимость присутствия клиента в торговом (операционном) зале в ожидании обслуживания;

– отсутствие возможности удаленной "записи" в очередь на удобное для клиента время;

– отсутствие возможностей удаленного мониторинга состояния очереди или состояния выполнения заказа (при длительном обслуживании).

Также существует большой класс услуг, при организации которых система на базе номерковых аппаратов не может быть использована, это оказание услуг "на дому" у клиента. К таким услугам относятся, например, вызов врача на дом; вызов мастера для установки, ремонта бытовой техники; вызов ветеринара и т.д. При этом клиенту сообщается планируемый достаточно продолжительный интервал времени оказания услуги (в течение дня, после обеда) и клиент испытывает неудобства при вынужденном ожидании услуги. Обычно для уточнения времени оказания услуги клиент связывается с оператором по телефону, что отнимает много времени у персонала и клиента.

Технологии быстрого оповещения (посредством мобильной связи или с помощью Интернет) делают крайне привлекательной идею создания очереди, управляемой дистанционно.

Предлагается заменить элементы системы электронной очереди (полностью или опционально) на мобильный телефон, используемый в качестве SMS-терминала. Очередь из ожидающих людей становится виртуальной, не требует "физического" присутствия клиента.

Для оперативного оповещения клиента о прогнозируемом времени обслуживания предлагается использовать подсистему оповещения о состоянии услуги.

Таким образом, программный комплекс, реализующий дистанционное управление очередью, может быть разделен на две дополняющие друг друга подсистемы: "Электронная очередь" и "Оповещение".

Электронная очередь

Алгоритм работы подсистемы

Клиент производит запись в очередь на сайте или посылает SMS на специальный номер (текст SMS зависит от услуги). После записи система сообщает (через сайт или с помощью SMS) номер очереди, пароль и прогнозируемое время ожидания.

Оператор, закончивший работу с очередным клиентом, посылает SMS, означающий "Следующий", на сервер. Система посылает SMS-оператору о номере и пароле очередного клиента. Клиенту посылается SMS с номером окна, куда ему нужно подойти.

Клиент подходит к окну, указанному в SMS, сообщает пароль и получает требуемую ему услугу.

Клиенту могут быть представлены дополнительные возможности, такие как информирование (через сайт или SMS) о прохождении очереди, о прогнозируемом времени ожидания.

Представленный алгоритм реализует вариант подсистемы "Электронная очередь", когда телефон заменяет номерковый аппарат (инфокиоск), табло и пульт вызова. В случае необходимости подсистема может заменять лишь часть аппаратуры, например номерковый аппарат и табло или только номерковый аппарат.

Все варианты исполнения подсистемы имеют достоинства традиционных систем и не имеют их недостатков. Стоимость подсистемы, ее инсталляции и поддержки существенно меньше, чем у традиционных систем управления на базе номерковых аппаратов.

Оповещение

Алгоритм работы подсистемы

Клиенты записываются на услуги через Интернет или через оператора. В результате для каждой услуги формируется список обслуживания (последовательность обслуживания клиентов). Список обслуживания должен содержать номер в очереди на обслуживание и номер мобильного телефона клиента. Списки обслуживания по всем услугам являются входной информацией для системы оповещения. После формирования списка по конкретной услуге клиентам по SMS сообщается номер в списке и приблизительное время предоставления услуги.

Закончив работу с очередным клиентом, сотрудник, обслуживающий клиентов, посылает SMS, означающий "Следующий". На основании информации об обслуженных клиентах Система рассчитывает уточненное время обслуживания для ожидающих клиентов.

Клиентам из списка сообщают информацию (через SMS) об изменении номера в списке и о прогнозируемом времени ожидания.

Подсистема "Оповещение" обеспечивает:

– формирование списка обслуживания (последовательности обслуживания клиентов);

– мониторинг прохождения обслуживания;

– оповещение клиентов о деталях обслуживания.

В совокупности программный комплекс "Электронная очередь" и "Оповещение" включает следующий функционал.

1. Создание очереди (в соответствии с заданными параметрами).

2. Обработка заявок на постановку в очередь.

3. Обработка событий очереди (обслуживание).

4. Расчет и прогнозирование времени ожидания в очереди.

5. Оповещение о прохождении очереди.

6. Оповещение заказчиков дополнительных сервисов.

7. Сбор статистики о результатах обслуживания.

Систему "Электронная очередь" и "Оповещение" предполагается использовать как для крупных организаций путем конфигурирования ее для конкретных потребностей организации [1], так и для небольших организаций.

Идея предлагаемого подхода для небольших организаций состоит в том, что им предлагается не система управления очередью, а услуги по управлению очередью. Такой подход позволяет организации не покупать систему и не поддерживать ее эксплуатацию.

Система, реализующая дистанционное управление очередями, позволяет построить модель обслуживания [3] со следующими характеристиками.

– Заявки в очередь могут подаваться как на ближайшее возможное время обслуживания, так и на фиксированное время (превышающее ближайшее возможное). Отказ от обслуживания/пропуск очереди клиентом анализируется раздельно для этих типов заявок.

– Каждая очередь является однолинейной [2], но может обслуживаться несколькими операторами, обладающими индивидуальными характеристиками (производительность, график работы и т.п.).

– Обслуживание очереди каждым из операторов может быть разбито на интервалы простоя, ожидания, обслуживания, где ожидание характеризует фазу между двумя идущими подряд обслуживаниями. При этом решение о клиенте, который будет обслужен, и его оповещение происходит заранее. Интервалы ожидания анализируются для всех операторов очереди вместе.

– Заявки в очередь могут поступать по различным каналам: Интернет-заявки, с помощью

мобильной связи. Интенсивность поступления заявок и отказов от обслуживания по различным каналам анализируется раздельно.

– Ближайшее возможное время обслуживания определяется на основе статистики данной очереди, ее параметров обслуживания, индивидуальных характеристик операторов. Оповещение клиента о наступлении обслуживания происходит при известном обслуживающем операторе очереди.

Построение модели и получение с ее помощью оптимальных параметров обслуживания очереди и оповещения заинтересованных лиц является предметом дальнейшего исследования и развития системы дистанционного управления очередями.

Заключение

В статье рассмотрен подход к созданию системы дистанционного управления очередями потребителей, основанный на использовании Интернет и SMS-сервисов для выполнения основных функций системы. Рассмотрены требования к функциональному составу системы и модульной структуре. Обозначены основные положения модели обслуживания. Приведены решения ключевых проблем, связанных со спецификой области применения системы и с внедрением системы.

СПИСОК ЛИТЕРАТУРЫ

1. **Самочадин А.В., Старостенко А.С.** Разработка методики прецедентов для обследования организации и составления технического задания. // "Научно-технические ведомости", №2, СПб.: СПбГПУ, 2008. 114–118 с.

2. **Чейз Р.Б., Эквилайн Н.Дж., Якобс Р.Ф.** Производственный и операционный менеджмент. М.: Вильямс, 2004. 704с.

3. **Cooper R.** Introduction to Queueing Theory. New-York, 1981. 361c.

Куракин А.С.

Использование мультиагентного подхода при разработке комплексов программ систем корпоративного дистанционного образования

В процессе роста и развития бизнеса большинство крупных компаний сталкиваются с необходимостью обучения и переобучения персонала. При этом предметом обучения могут быть как базовые навыки, так и знания, специфические для данной компании. Успехи, достигнутые в области современных информационных технологий, делают актуальной задачу создания и использования для обучения персонала коммерческих предприятий системы дистанционного обучения (СДО). Современная система ДО, в частности корпоративная, представляет собой программную оболочку (learning managing system, LMS), которую размещают на сервере внутренней компьютерной сети предприятия или на сервере провайдера. Внедрение современных СДО позволяет компаниям добиться следующих основных преимуществ [4].

1. Значительное снижение затрат на обучение персонала по сравнению с аудиторными занятиями.

2. Обеспечить единый стандарт знаний для сотрудников.

3. Достичь существенной экономии времени, потраченного на обучение.

4. Обеспечить эффективный контроль хода и результатов обучения.

5. Обеспечить массовость обучения в короткие сроки.

Необходимо отметить, что на российском рынке корпоративных СДО именно отечественные разработки обеспечивают более выгодное соотношение цена/функциональность. Большинство систем ДО, завоевавших себе устойчивую репутацию на рынке систем для обучения, разработаны и внедрены специа-

листами компаний NauLearning (NAUMEN), Training Ware (ГиперМетод), ДОЦЕНТ (Униар), WebTutor (WebSoft), СКО (ФИЗИКОН), Прометей, REDCLASS (REDCENTER).

В США 62% предприятий численностью свыше 1000 человек уже внедрили у себя системы электронного дистанционного обучения [7]. В России такие системы используют пока лишь некоторые крупные компании: "Сибнефть", "Норникель", "Татнефть", "Вимм-Билль-Данн", "Северсталь", "Вымпелком", РЖД. Например [см. 4], проект по СДО в РЖД был запущен в январе 2005 года. Использовалась технология WebTutor, разработанная компанией Websoft. Система интегрирована в учебный портал, содержащий информацию и о других источниках знаний в компании. Система очень популярна в регионах, а дистанционные курсы в работе учебного центра занимают почти треть от общего объема обучения. В холдинге "Норникель" система дистанционного обучения работает с 2002 года. В качестве LMS используется российская разработка "Прометей".

Появление новых форм сетевых технологий неизбежно оказывает влияние на методы и приемы дистанционного обучения. Сейчас в центре внимания специалистов сопоставление и осмысление в сравнении технологий Web 1.0 и 2.0 и соответственно E-Learning 1.0 и 2.0. Тип коммуникации на широкую аудиторию унаследован ДО из самых древних времен и воплощается преимущественно в форме E-Learning 1.0. [См. 8]. Web и E-Learning 2.0, основу которых составляют технологии wiki, блогов, социальных сетей и проч. в сравнении с Web 1.0, напротив, чрезмерно свободны и плохо поддаются формальному управлению и контролю. Здесь концептуально отсутствует орган, уполномоченный решать, чему и как учить и как преодолевать разногласия.

Сформулируем общую схему функционирования современной системы управления процессом обучения (LMS), в котором решающее значение имеет фактор интеллектуальности этих систем, понимаемый как реализация творческих функций, таких как оценка индивидуальных особенностей и текущего уровня знаний и умений обучаемого и выбор в соответствии с ней частей учебного материала, учебных заданий, способов их подачи, частоты и продолжительности и т.д. Детальный анализ таких систем проведен в работе [7]. Необходимо отметить, что представленная структура является общей для всех LMS-систем независимо от того, является ли она классически академической или используется для корпоративного обучения и достижения при этом специфических учебных целей.

Основными подсистемами такого комплекса программ являются подсистема формирования начальных условий, подсистема формирования управляющего воздействия и подсистема контроля знаний, умений и навыков (ЗУН). Подсистема формирования управляющего воздействия предоставляет возможность определить текущий уровень ЗУН обучаемого, сформировать стиль и индивидуальную программу обучения. Подсистема контроля

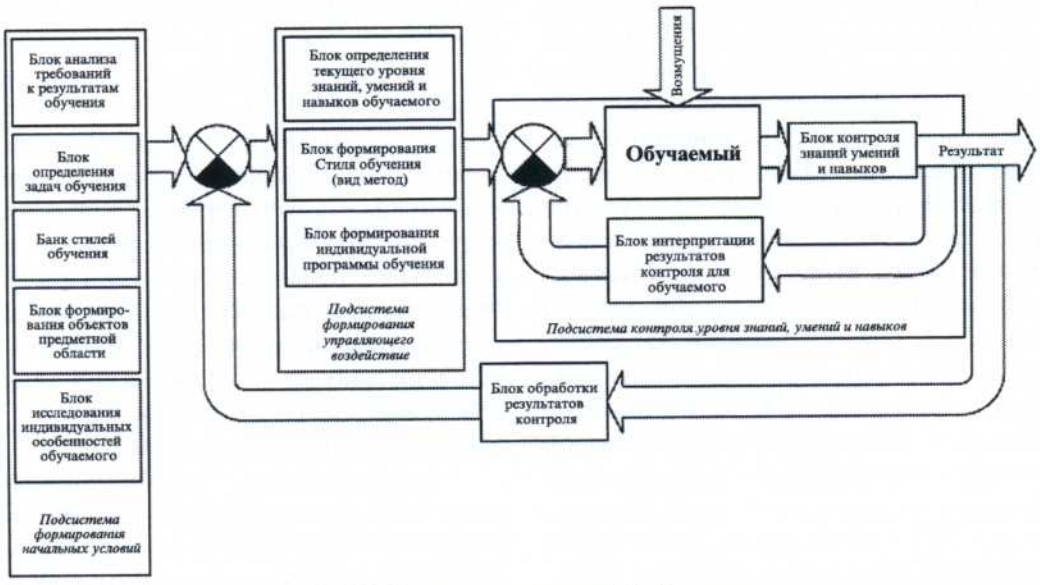


Рис. 1. Обобщенная структура LMS-системы

ЗУН обеспечивает возможность проведения различного вида контроля ЗУН, обработку и представление полученных результатов.

Теперь определим наиболее существенные отличия академического стиля дистанционного обучения от систем корпоративного обучения с целью выявить особенности построения комплексов программ корпоративных СДО.

Во-первых, корпоративное обучение прежде всего ставит перед собой задачи обеспечить освоение обучающимися определенных умений и навыков на уровне, определяемом внутрикорпоративными стандартами. Здесь особенно важно подчеркнуть практический аспект обучения, его направленность на ближайшие цели, жесткие требования к уровню освоения материала.

Во-вторых, при корпоративном обучении значительно возрастают требования к автоматизации процесса обучения. Это обусловлено тем, что для изучения большинства практических вопросов невозможно наладить эффективное взаимодействие в цепочке "носители умений и навыков – создатели курса – тьютор – обучаемый". Дело в том, что носителем практических знаний и особенно навыков являются конкретные специалисты (сотрудники компании или приглашенные), которые участвуют в создании учебного курса путем взаимодействия с группой разработчиков курса – людей, являющихся специалистами в вопросах корпоративного обучения и программистов, но уже не обладающих специальными знаниями и умениями. Соответственно тьюторы, обеспечивающие реализацию дистанционного курса, могут лишь частично заменить носителей практических знаний, и поэтому их роль в оперативном управлении процессом обучения оказывается существенно ограниченной. Следовательно, обеспечить эффективное проведение всего процесса дистанционного обучения по курсу возможно лишь при наличии гибкой программной среды, изначально заложенной в структуру курса и в максимальной степени обеспечивающей учет специфики предмета. Такая управляющая программная структура может быть создана исключительно совместными усилиями технических разработчиков курса и людей-носителей практических умений и навыков, излагаемых в курсе. Именно на этапе программной разработки курса дистанционного обучения в него уже должны быть заложены основные управляющие конструкции процесса обучения, особенности которых может предусмотреть только практический специалист, передающий свои умения и навыки слушателям курса опосредованно, с использованием разработанного программного комплекса СДО.

Уровень программной разработки должен быть достаточно высок для того, чтобы фактически заменить участие квалифицированного преподавателя и обеспечить эффективную работу тьюторов по координации прохождения курса слушателями с учетом того факта, что тьюторы сами не обладают достаточными знаниями и навыками (прежде всего практическими) в предметной области курируемого ими курса дистанционного обучения.

Изложенные соображения также свидетельствуют о том, что в системах корпоративного дистанционного обучения возможности эффективного использования методик web 2.0 и стандартов e-learning 2.0 очень ограниченны по сравнению с традиционными академическими формами обучения. Отличительной чертой обучения сотрудников в компаниях является высокая централизация процесса обучения и невозможность (либо низкая эффективность) средств совместной работы слушателей и тьюторов.

Итак, наиболее высокие и специализированные требования при разработке комплексов программ корпоративных СДО предъявляются к подсистеме формирования управляющего воздействия. При программной разработке этой подсистемы необходимо минимизировать роль участия специалиста в принятии решений по организации или модификации процесса обучения на каждом этапе, но при этом обеспечить высокий уровень интеллектуальности данного модуля. В результате прохождения полного процесса обучения слушатель должен овладеть специальными умениями и навыками в необходимом с точки зрения сложившейся корпоративной среды объеме и при минимизации размера условных затрат на обучение. В период обучения по каждому из запланированных к овладению умений и навыков должен программно реализовываться следующий набор функций для отвечающего за данный учебный модуль элемента комплекса программ:

– *автономность* – способность элемента комплекса программ функционировать без вмешательства человека и при этом осуществлять самоконтроль над своими действиями и внутренним состоянием;

– *общественное поведение* – способность функционировать в сообществе с другими элементами используемого комплекса программ, обмениваясь с ними сообщениями с помощью специализированного языка коммуникаций;

– *реактивность* – способность воспринимать состояние среды и своевременно реагировать на те изменения, которые в ней происходят;

– *про-активность* – способность элемента комплекса программ генерировать цели и действовать рационально для их достижения.

Фактически это означает, что программные модули комплекса программ СДО выполняют роли слабых интеллектуальных агентов в смысле [2].

Обучающая система на основе мультиагентных технологий строится из интеллектуальных рефлекторных агентов (ИА), которые представляют собой законченные автономные системы [см. 6]. Логический вывод в системе осуществляет каждый агент на основе внутренней базы динамических знаний, а обмен производится только статическими знаниями. Кооперирующиеся агенты, взаимодействуя между собой, функционируют в различных программных модулях, программах и на различных аппаратных средствах.

Модель обучающей системы на основе мультиагентных технологий включает в себя базы знаний (хранилище единиц знаний), клиентов (слушателей и тьюторов) и мультиагентную систему, состоящую из программного интерфейса и интеллектуальных агентов (рис. 2). Центральным звеном такой обучающей системы являются базы знаний, которые выступают по отношению к другим компонентам в качестве содержательной подсистемы, составляющей основную ценность [6]. Принцип действия корпоративной обучающей системы на основе мультиагентных технологий заключается в

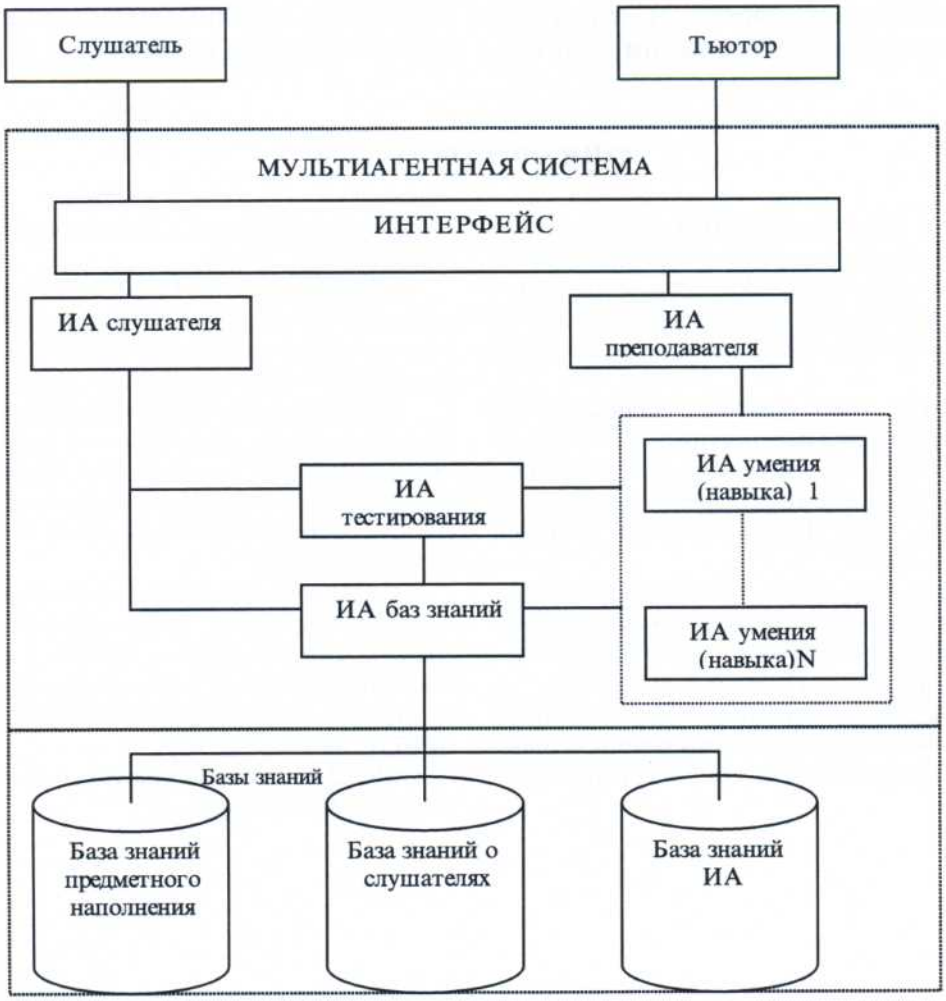


Рис. 2. Модель корпоративной обучающей системы на основе мультиагентных технологий

специфической компоновке агентов и их взаимодействии (см. рис. 2).

Основа этого подхода – построение системы как совокупности агентов (агенты пользователя, агенты преподавателя, агенты лекций и даже агенты отдельных объектов знания: определений понятий и правил, задач, методов, результатов, лабораторных работ, комментариев и т.д.) [ср. 1]. Каждый из агентов имеет семантическое описание своего поля деятельности и соответствует экспертной системе с традиционной структурой. Агент обладает всеми свойствами экспертных систем, а также памятью своей деятельности. Основная идея применения агентов заключается в том, что каждый агент имеет собственные ресурсы для достижения собственных целей, взаимодействия с другими агентами и разрешения конфликтов с целями других агентов для достижения общей цели. Это позволяет свободно выбирать те цели, которые преследуются на данный момент объектом управления, и соответственно целям выбирать тот эталон (представленный соответствующим агентом), соответствие которому достигается моделью обучаемого на данный момент [5]. Вопросы взаимодействия агентов разной архитектуры могут быть решены применением соответствующего языка коммуникации агентов (ACL) и языка обмена информацией, которые дают возможность агентам эффективно понимать друг друга, несмотря на разницу в подходах их построения и функционирования.

Таким образом, для каждой конкретной задачи обучения в рамках корпоративной СДО составляется определенный коллектив агентов, что обеспечивает смену структуры и целей решающей системы в зависимости от поставленной задачи. Формирование коллективов агентов для решения задач корпоративного обучения позволяет реализовать любой уровень адаптации, т.к. эта процедура предполагает формирование каждый раз структуры системы, ее представления об объекте управления, т.е. обучаемом, и целей обучающей системы, адаптируемые под цели, преследуемые на данный момент объектом управления.

СПИСОК ЛИТЕРАТУРЫ

1. **Батищев С.В., Скобелев П.О.** Проблемы реализации мультиагентных систем дистанционного обучения в сети Интернет. - Материалы Международной конференции "Развитие новых технологий в системе образования РФ", Россия, г. Самара, 2005 г. С. 50–54.

2. **Водяхо А.И., Пантелеев М.Г., Татаринов Ю.С.** Концепция построения перспективных систем распределенного обучения на базе интеллектуальных агентов// Современные технологии обучения: Сб. науч.-метод. тр. СПб: Изд-во СПбГЭТУ "ЛЭТИ", 2003. Вып. 6, с. 29–39

3. **Демьянов А. В.**, Системный подход к синтезу интеллектуальной компьютерной обучающей системы // Надежность и качество: тр. Междунар. симп. – Пенза : Инф.-изд. центр ПГУ, 2005. С. 153–155.

4. **Клыков В.В.**, Интерактивные компьютерные тренажеры // Международная конференция "Новые информационные технологии в университетском образовании". Тез. докл. Новосибирск: изд-во СибГУТИ, 2003. С. 85–86.

5. **Нарожный А.В.** Проектирование структуры автоматизированной системы в условиях дистанционного обучения// Вестник ХНТУ "ХПИ" – Харьков: ХНТУ "ХПИ", 2005. № 54. С. 62–67.

6. **Пантелеев М.Г., Жандаров В.В., Семенов В.С.** Мультиагентная среда аннотирования образовательных ресурсов на основе технологий Семантического Web// Материалы VII Международной конф. по мягким вычислениям и измерениям SCM'04. – СПб.: СПбГЭТУ, 2004

7. **Рабинович П.Д.**, Исследование и разработка математической модели, алгоритмов и программного средства для представления и передачи знаний в компьютерных обучающих системах. Труды II научной конференции "СИМ'2005" – Москва, МФТИ, 2005

8. Reed D. The Law of the Pack // Harvard Business Review, 2008, February

Ицыксон В.М., Моисеев М.Ю., Цесько В.А., Карпенко А.В.

Исследование систем автоматизации обнаружения дефектов в исходном коде программ

В настоящее время программное обеспечение (ПО) является составной частью большинства технических систем. Качество ПО, его надежность и безопасность являются важными характеристиками, влияющими на функционирование системы в целом. В то же время известно, что большинство программных систем содержит ошибки: дефекты, недоработки, уязвимости. Природа возникновения программных ошибок связана с особенностями самих программных систем как одних из самых сложных продуктов деятельности человека. Современные программные системы состоят из множества составных частей и могут содержать до нескольких десятков миллионов строк кода.

Общего решения проблем управления процессом разработки, контроля качества, тестирования и верификации таких сложных систем в настоящее время не существует. Многие фирмы-разработчики вкладывают значительные средства в различные направления программной инженерии, позволяющие повысить качество ПО за счет уменьшения количества дефектов [1]. Некоторыми из этих направлений являются:

- разработка интегрированных средств сквозного проектирования ПО и автоматическая генерация программного кода;
- автоматизация динамического анализа и тестирования;
- верификация программного обеспечения (model checking);
- статический анализ программного кода.

Способы обнаружения дефектов можно разделить на три группы:

- ручные методы анализа кода: аудит кода, парное программирование, ручное тестирование, ручная верификация программы на соответствие спецификации;
- динамические методы: различные виды тестирования (модульное, функциональное, производительности, стресс-тестирование), анализ трасс выполнения программы, профилирование;
- статические методы: компиляция, статический анализ исходного кода, проверка на соблюдение стиля кодирования, методы верификации на основе моделей.

Процесс выявления дефектов в ПО является чрезвычайно трудоемким и может занимать значительную часть от всего времени разработки, поэтому задача автоматизации выявления дефектов является актуальной [2]. В данной статье проводится сравнительный анализ существующих средств автоматизации обнаружения дефектов в исходном коде программ.

Автоматизация поиска дефектов

Различные способы обнаружения дефектов характеризуются различной степенью автоматизации. Методы ручной проверки кода имеют высокую трудоемкость и обычно используются только для небольших частей программной системы. Возможности автоматизации этого процесса ограничиваются обеспечением удобного просмотра и навигации по исходному коду программы, а также управлением списком дефектов.

Динамические методы, в частности различные виды тестирования, допускают частичную автоматизацию некоторых стадий (запуск, выполнение, анализ простых результатов). При этом создание тестов, организация воздействия на программу при функциональном тестировании, анализ сложных результатов трудно формализуем и обычно выполняется вручную. Количество тестов, обеспечивающее приемлемый уровень тестового покрытия для любой нетривиальной программы, достаточно велико. Так, общее количество строк кода модульных тестов обычно сопоставимо с размером программы, сложность создания таких тестов зачастую не уступает сложности разработки самой программы. Задачи получения и анализа трасс выполнения программы также плохо автоматизируются.

Среди статических методов обнаружения дефектов компиляция программы выполняется полностью автоматически, при этом современные компиляторы могут обеспечивать обнаружение некоторых дефектов. Средства обнаружения несоответствия стилю кодирова-

ния также решают свою задачу автоматически. Средства, реализующие верификацию кода программы на соответствие спецификации, требуют ручного создания модели программы и формального описания требований спецификации, автоматизируется только сам процесс проверки модели. Известны попытки автоматизации процесса извлечения моделей программ из исходного кода, однако в настоящее время такие подходы имеют серьезные ограничения [3]. Методы статического анализа исходного кода программ также допускают частичную или полную автоматизацию. В этой области ведутся активные исследования [4], результаты которых реализуются в средствах автоматического поиска дефектов.

Можно сделать вывод, что статические методы обнаружения дефектов, особенно методы статического анализа исходного кода, наилучшим образом подходят для автоматизации, поэтому далее будут рассматриваться средства автоматизации поиска дефектов, основанные на этих методах.

Классификация дефектов

Для проведения сравнительного анализа средств поиска дефектов необходимо выделить множество программных дефектов, выявление которых позволит характеризовать качество алгоритмов обнаружения. Использование классификации дефектов при проведении экспериментов позволяет более точно оценить возможности и специализацию каждого исследуемого средства обнаружения дефектов.

Одним из возможных решений является использование одного из существующих стандартов в области информационной безопасности в качестве основы для построения классификации.

На данный момент в области безопасности исходного кода не существует международных или общепринятых стандартов [5], в частности стандартов таких комитетов, как ISO, IEC и национальных комитетов – NIST, BSI. На статус общепринятого стандарта претендуют стандарты CERT Secure Coding Standard в вариантах для C и C++. Их содержание обобщает многочисленные практики и руководства безопасного кодирования, принятые в различных организациях. Интерес к совместной работе над стандартами с CMU CERT проявляет группа OWGV при комитете ISO/IEC JTC 1/SC 22.

Помимо инициативы CMU CERT существует ряд отраслевых стандартов в области безопасного кодирования. Например, для автомобильной промышленности таким стандартом является MISRA C. Специфика программного обеспечения в этой области накладывает значительный отпечаток на требования к безопасности исходного кода, ограничивая использование многих конструкций языка программирования, что делает этот стандарт неприменимым к программному обеспечению общего назначения.

В области классификации дефектов нужно различать классификации проявлений дефектов, или сбоев, и дефектов как ошибок в исходном коде. В отличие от большого количества классификаций сбоев, ошибки в исходном коде обычно классифицируются неформально. Отдельные организации имеют свои внутренние классификации дефектов. Общепринятыми классами ошибок являются ошибки выделения ресурсов, инициализации, переполнения буфера и др.

В данной работе предлагается собственная классификация дефектов, учитывающая наиболее важные свойства дефектов:

- тип – определяется сущностью дефекта и способом его проявления (способом его влияния на программу);
- частота проявления – насколько часто возникают условия для проявления дефекта в данной программе;
- серьезность – каково негативное влияние проявления дефекта на программу в целом (выдача неверных результатов, зависание, авария);
- распространенность – насколько часто данный дефект встречается в реальных программах;
- стадия – стадия разработки, на которой был внесен дефект;
- сложность – трудоемкость поиска и исправления дефекта, вероятность внесения новых дефектов в процессе исправления;
- приоритет – насколько важно найти и исправить данный дефект.

Разработанная классификация дефектов по типам приведена в таблице 1. Классификация дефектов по типам основана на соотнесении дефектов с объектами и свойствами объектов. Например, тип дефектов RES соотносится с объектами, являющимися ресурсами (динамическая память, файловый дескриптор и другие).

Таблица 1

Классификация дефектов по типам

Номер	Тип / Подтип	Описание / Примеры
RES	Ошибки управления ресурсами	
RES-01	Утечка динамической памяти	Повторное выделение памяти, потеря ссылки
RES-02	Утечка дескрипторов	Повторное открытие, потеря дескриптора
RES-03	Множественное освобождение памяти	Освобождение уже освобожденного указателя
RES-04	Множественное освобождение дескрипторов	Повторное закрытие
RES-05	Корректная обработка ошибок выделения ресурса	Отсутствие проверок успешности операции выделения ресурса до его использования
RES-06	Ошибки протокола работы с ресурсом	Запись в файл, открытый на чтение
BUF	Ошибки работы с буферами/массивами/строками	
BUF-01	Ошибка индекса массива	Индекс выходит за границу массива
BUF-02	Ошибка арифметики указателей	Разыменование указателя, выведенного за границу буфера
BUF-03	Ошибка переполнения буфера	Попытка записи в буфер недостаточного размера
BUF-04	Ошибка отсутствия завершающего нуля строки символов	Завершающий символ изменен
BUF-05	Арифметические операции с указателями на разные массивы	Вычитание, сравнение таких указателей
BUF-06	Арифметические операции с указателем на объект, не являющийся массивом	Инкрементирование такого указателя
INI	Ошибки отсутствия инициализации	
INI-01	Использование неинициализированной переменной	В выражениях, условиях, в качестве индекса массива или аргумента аннотированной функции
INI-02	Использование неконтролируемого значения переменной	Значение из потока ввода используется в качестве индекса массива, аргумента аннотированной функции
INI-03	Разыменование неинициализированного (освобожденного) или нулевого указателя	Нулевой указатель передается в аннотированную функцию
INI-04	Разыменование неконтролируемого указателя	Значение из потока ввода используется в качестве указателя, производится разыменование
INI-05	Чтение неинициализированного (освобожденного) или неконтролируемого указателя без разыменования, или чтение неинициализированной переменной	Не является непосредственно ошибкой, однако может быть потенциальной угрозой
FRM	Уязвимость форматной строки	
FRM-01	Использование в качестве строки формата неконтролируемого значения	Например, прочитанного из потока ввода
FRM-02	Использование опасных функций ввода	Функции без контроля размера вводимых данных (*gets*)
FRM-03	Проверка соответствия списка аргументов форматной строке	В форматных функциях (*printf*)
TYP	Ошибки типов	
TYP-01	Неявное приведение несовместных типов	Преобразование переменной типа int к типу char
TYP-02	Преобразование типов с разным выравниванием	Преобразование типов для массивов и структур

Окончание табл. 1

Номер	Тип / Подтип	Описание / Примеры
TYP-03	Разыменование в неправильный тип	Преобразование указателя к указателю на другой тип и разыменование
TYP-04	Ошибки разыменования указателей на функции	Вызов такой функции с некорректным набором аргументов
TYP-05	Наличие объявлений функций без аргументов	Могут быть вызваны с любым набором аргументов
SCP	Ошибки нарушения области видимости	
SCP-01	Выдача из функции указателей на локальные переменные	Выдача указателя на локальную переменную
EXP	Ошибки в выражениях	
EXP-01	Ошибки применения побитовых операторов	Побитовый сдвиг переменной с отрицательным значением
EXP-02	Наличие побочных эффектов в логических выражениях и некоторых функциях	Использование оператора ++ для аргумента sizeof
EXP-03	Предположение определенного порядка вычисления подвыражений в выражении	Порядок вычисления значений аргументов в вызове функций не установлен
EXP-04	Ошибка использования переменных типа int для float-арифметики	Вычисления части выражения могут быть целочисленными
MAC	Ошибки макросов	
MAC-01	Ошибки изоляции аргументов	Отсутствие скобок вокруг аргументов макроса в теле макроса
MAC-02	Многократное вычисление аргумента	Использование вызовов функций в качестве аргумента макроса

Примером связи дефектов со свойствами объектов программы является тип дефектов INI, в котором анализируется свойство наличия/отсутствия инициализации. Выделение подтипов дефектов позволяет детализировать объекты и их свойства, а также структурировать возможные представления дефекта в исходном коде.

Методика исследований средств обнаружения дефектов

Для сравнения возможностей средств автоматизации обнаружения дефектов необходимо провести ряд экспериментов. Методика проведения экспериментов включает в себя выбор показателей, разработку тестов, запуск исследуемых средств на этих тестах, фиксацию результатов и расчет выбранных показателей. Целью проведения этих экспериментов является оценка эффективности обнаружения дефектов различными средствами. Эффективность обнаружения дефекта – комплексное свойство, отражающее качество получаемого результата, степень автоматизации процесса анализа и сложность его организации, ресурсоемкость, применимость для программ различного размера и класса.

Нас будут интересовать только качество получаемых результатов, которое можно оценить с помощью показателей эффективности. Предлагается использовать в качестве показателей полноту и точность результатов анализа.

Полнота результатов анализа – это показатель, характеризующий долю обнаруживаемых дефектов среди всех дефектов в программе.

Точность результатов анализа – это показатель, характеризующий долю истинных дефектов среди всех обнаруженных дефектов.

Эти показатели зависят от набора дефектов, выявляемых данным средством, а также от возможностей анализа сложных конструкций языка. Рисунок 1 поясняет рассмотренные показатели.

С учетом этих факторов сформированы две группы тестов: простые тесты и комплексные тесты. Простые тесты предназначены для определения перечня обнаруживаемых дефектов,



Рис. 1. Показатели эффективности

комплексные тесты – для оценки возможностей анализа сложных конструкций языка. Любой тест представляет собой программу на языке С, состоящую из одного файла. Программа содержит один или несколько контролируемых дефектов – дефектов, специально внесенных в код программы. Тестовая программа не содержит других дефектов. Тестовые программы не используют библиотечных функций, требующих аннотирования для выполнения анализа. Решение проводить эксперименты на специально разработанных тестах объясняется необходимостью контролировать все дефекты в анализируемых программах, чего практически невозможно добиться для любой реальной программной системы.

При проведении экспериментов фиксируются пропуски имеющихся в анализируемой программе дефектов и ложные предупреждения. На основе числа пропусков, ложных предупреждений и общего известного числа дефектов рассчитываются выбранные показатели. Результаты обнаружения фиксируются отдельно для каждого типа дефектов, что позволяет сравнивать эффективность обнаружения разных типов дефектов выбранными средствами. Ложные предупреждения фиксируются только для комплексных тестов. Это связано с тем, что простые тесты состоят всего из 10–20 строк кода, в них отсутствуют конструкции языка, не связанные с представляемым дефектом, поэтому они плохо подходят для оценки показателя точности.

Простые тесты направлены на определение способностей обнаружения незамаскированных дефектов рассматриваемыми средствами. В каждом простом тесте присутствует один или несколько локализованных дефектов одного типа. Тестовая программа содержит одну функцию – функцию *main*.

Разработанный набор простых тестов покрывает все рассмотренные типы и подтипы дефектов, имеющиеся в разработанной классификации. Для каждого подтипа дефектов разработан один или несколько тестов. Некоторые дефекты реализованы в нескольких тестах в разных вариантах представления.

Комплексные тесты разработаны с целью выявления свойств реализованных алгоритмов, они не содержат конструкций, которые нельзя проанализировать статическими методами. В отличие от простых тестов в комплексных тестах дефекты замаскированы в коде программы, для этого используются различные возможности языка программирования, представляющие сложность для статического анализа. Некоторые дефекты являются распределенными: отдельные части дефекта находятся в разных частях программы.

Всего было разработано четыре комплексных теста, покрывающих основные возможности языка С, при этом каждый комплексный тест включает в себя несколько дефектов разных типов. Распределение дефектов по комплексным тестам приведено в таблице 2.

Тест UNI-01 использует основные возможности изменения потока управления, операторы *if, for, while, break, continue и goto*, а также их различные комбинации. Тест UNI-02 состоит из нескольких процедур с вложенными вызовами и рекурсией. Тест UNI-03 использует указатели, в том числе указатели на функции. Тест UNI-04 является программой, которая решает задачу интегрирования функции: этот тест использует все возможности языка, использованные в предыдущих тестах.

Таблица 2

Распределение дефектов по комплексным тестам

Номер	Описание	RES	BUF	INI	FRM	TYP	SCP	MAC	EXP
UNI-01	Ветвления, циклы, break, continue, goto	13	5	6	7	0	0	0	1
UNI-02	Вызовы процедур, в том числе и сложные, рекурсии	11	7	9	4	0	0	0	3
UNI-03	Указатели, в том числе на функции	4	4	8	0	0	0	0	4
UNI-04	Программа расчета интеграла функции	2	6	7	3	0	0	0	0

Проведение экспериментов с использованием комплексных тестов позволяет оценить такие свойства алгоритмов анализа, как способность анализировать указатели и сложные типы данных, способность вычисления условий переходов и циклов, наличие выведения и проверки типов, наличие межпроцедурного анализа и другие свойства.

Все разработанные тесты доступны в [7].

Выбор средств поиска дефектов

Выбранные для исследований средства должны обладать необходимой функциональностью, широко использоваться на рынке проектирования ПО, должны осуществляться их поддержка и развитие разработчиками.

Для проведения экспериментальных исследований были выбраны следующие средства: IBM Rational Software Analyzer 7.0.0, Gimpel FlexeLint 9.0, Microsoft Code Analysis 2008, Fortify Source Code Analysis 5.2, Splint 3.1.2. Среди выбранных продуктов присутствуют средства от лидеров мировой индустрии разработки программного обеспечения: IBM и Microsoft, наследники одного из первых средств поиска дефектов lint – коммерческий продукт компании Gimpel – FlexeLint и поддерживаемое сообществом разработчиков средство Splint, а также продукт компании Fortify Software, специализирующейся на разработке средств анализа программ. Другие программные продукты не исследовались по причине несоответствия предъявленным требованиям или из-за трудностей их получения.

Рассмотрим средства, выбранные для исследования.

Коммерческий продукт IBM RSA предназначен для поиска дефектов в коде на языках C/C++ и Java, интегрируется в среду Eclipse, существуют версии для ОС Linux и Windows. Статический анализ производится согласно набору правил, структурированных по категориям и важности. Анализ кода разделен на три этапа:

- ревизия кода – каждый исходный файл анализируемой программы проверяется синтаксическим анализатором, который осуществляет поиск шаблонов кода, нарушающих установленные правила;
- поиск зависимостей – осуществляется поиск зависимостей между исходными файлами и межпроцедурный анализ;
- анализ сложности – выполняется расчет метрик сложности программы, определяются части программы с избыточной сложностью и выдаются рекомендации для проведения рефакторинга.

Коммерческие продукты PC-Lint для ОС Windows и FlexeLint для ОС Unix, Linux, Mac OS X, Solaris предназначены для поиска дефектов в программах на языках C/C++. Система производит синтаксический анализ исходного кода, анализ потоков данных и управления, осуществляет межпроцедурный анализ. Средство PC-Lint может быть интегрировано в среду Visual Studio. Основные обнаруживаемые дефекты: ошибки нулевого указателя, работы с памятью,

потока управления, переполнения буфера, переполнения при арифметических операциях, а также нарушение стиля кодирования.

Коммерческий продукт Microsoft Code Analysis является инструментом статического анализа кода на языках C/C++, интегрированным в среду разработки Visual Studio. Средство реализовано в виде набора правил, сгруппированных по категориям. Имеются следующие возможности: создание политики проверки кода, настройка интерпретации найденных уязвимостей. Средство позволяет обнаруживать достаточно широкий перечень дефектов. Принципы построения и используемые методы анализа, реализованные в данном средстве, производителем не описываются.

Коммерческий продукт Fortify SCA предназначен для анализа кода на языках C/C++ и Java. Продукт является кроссплатформенным и может быть интегрирован в различные среды разработки (Visual Studio, Eclipse). В данном средстве реализован межпроцедурный анализ потоков данных и управления. Производится поиск уязвимостей в более чем 200 категориях. Поддерживается возможность глубокой параметризации и настройки проводимого анализа с целью уменьшения числа ложных срабатываний.

Некоммерческий проект Splint предназначен для статического анализа программ на языке C на наличие дефектов. Splint поддерживает внешнее аннотирование, реализует функциональность препроцессора. Выявляемые дефекты: разыменование нулевого указателя, использование неинициализированных переменных или возвращение не полностью определенных структур и объединений, несоответствие типов ошибки области видимости, ошибки управления памятью, в том числе утечки памяти, ошибки работы с аннотированными сущностями, ошибки потока управления, ошибки переполнения буфера, ошибки в макроопределениях, нарушение стиля кодирования.

Проведение экспериментов и анализ результатов

Выбранные средства были исследованы в соответствии с разработанной методикой. Каждый продукт был протестирован на всем множестве простых и комплексных тестов, результаты экспериментов приведены в таблице 3 и таблице 4. Для простых тестов приводится показатель полноты результатов анализа, для комплексных тестов в ячейках таблицы – показатель полноты / показатель точности.

Таблица 3

Результаты простых тестов

Номер теста	IBM RSA	FlexeLint	Microsoft CA	Fortify SCA	Splint
RES	0%	33%	0%	33%	11%
BUF	3%	48%	16%	44%	6%
INI	16%	73%	50%	47%	32%
FRM	57%	86%	71%	43%	85%
TYP	0%	46%	50%	0%	38%
SCP	0%	100%	100%	100%	0%
MAC	0%	100%	0%	0%	0%
EXP	11%	88%	33%	11%	50%

Таблица 4

Результаты комплексных тестов

Номер теста	IBM RSA	FlexeLint	Microsoft CA	Fortify SCA	Splint
UNI-01	38% / 41%	56% / 100%	34% / 91%	53% / 100%	0% / -
UNI-02	32% / 50%	47% / 100%	35% / 92%	56% / 100%	38% / 100%
UNI-03	35% / 39%	20% / 100%	35% / 100%	20% / 80%	20% / 67%
UNI-04	56% / 90%	22% / 100%	22% / 100%	22% / 100%	6% / 100%
Итог	40% / 55%	35% / 100%	32% / 96%	34% / 95%	16% / 89%

Результаты, полученные на простых тестах, существенно различаются, при этом FlexeLint заметно опережает остальные средства. Однако на комплексных тестах все средства, за исключением Splint, показали примерно одинаковые результаты по показателю полноты.

Точность получаемых результатов на комплексных тестах у исследуемых средств достаточно высока, за исключением IBM RSA. Несмотря на то, что это средство показало плохие результаты на простых тестах, оно достигло наилучшей полноты результатов на комплексных тестах. Возможно, это связано с низким порогом принятия решения об обнаружении дефекта в IBM RSA. Обобщенные результаты сравнительного анализа рассмотренных средств приведены на рис. 2.

Заключение

В ходе исследования была предложена классификация дефектов программного обеспечения и методика проведения сравнительного анализа средств обнаружения дефектов, использующая наборы простых и комплексных тестовых программ. На основе предложенной методики были получены оценки показателей эффективности выбранных средств анализа кода.

Все рассмотренные средства обеспечивают автоматизацию процесса выявления дефектов в исходном коде программ. Полнота результатов наилучшего из рассмотренных средств не превышает 40%. Кроме того, все средства анализа, за исключением FlexeLint, позволяют обнаруживать только дефекты определенных типов. Необходимо отметить, что использование этих средств применительно к реальным системам характеризуется высокой сложностью и трудоемкостью анализа результатов.

Разработанные тесты являются достаточно простыми и несопоставимы по сложности с реальными программными системами. Можно ожидать ухудшения показателей эффективности рассмотренных средств при анализе реальных программ.

Проведенные исследования выявили существенные различия в результатах работы рассмотренных средств. Это позволяет сделать вывод о том, что в средствах используются различные алгоритмы анализа, сама область автоматизированного обнаружения дефектов в исходном коде активно развивается, нет устоявшихся решений и подходов.

Возможные перспективы развития в данной области заключаются в разработке новых, более эффективных алгоритмов статического анализа, комплексировании различных подходов с целью повышения полноты и точности результатов работы средств автоматизированного обнаружения дефектов [6].

Исследование выполнено в рамках работ по государственному контракту № 02.514.11.4081 "Исследование и разработка системы автоматического обнаружения дефектов в исходном коде программного обеспечения" Федерального агентства по науке и инновациям в рамках Федеральной целевой программы "Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2012 годы".

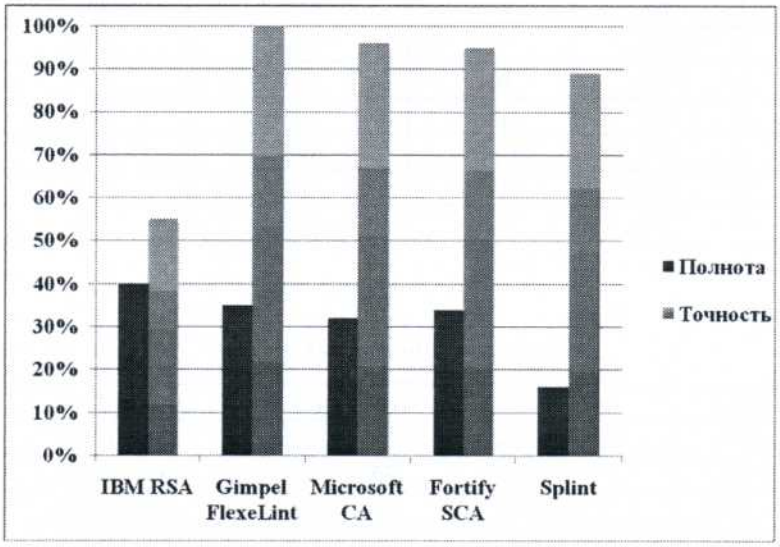


Рис. 2. Эффективность средств статического анализа

СПИСОК ЛИТЕРАТУРЫ

1. Static analysis for security / B. Chess, G. McGraw // IEEE Security & Privacy. – IEEE, 2004. – Volume 2, Issue 6. – p. 76–79.

2. **Брукс Ф.** Мифический человеко-месяц или как создаются программные системы. – Символ-Плюс, 2006. – 304 с.

3. First Steps in the Verified Software Grand Challenge / J. Woodcock // IEEE Computer. – IEEE, 2006. – Volume 39, Number 10, p. 57–64.

4. Software Analysis: a Roadmap] / D. Jackson, M. Rinard // Proceedings of the Conference on The Future of Software Engineering. – Limerick, Ireland: ACM, 2000. – p. 133–145.

5. Secure Coding: Principles & Practices / Mark G. Graff, Kenneth R. Van Wyk. – O'Reilly, 2003. – 200 p.

6. Principles of Program Analysis / Flemming Nielson, Hanne R. Nielson, Chris Hankin. – Corr. 2nd printing. – Berlin [etc.]: Springer, 2005. – XXI, 452 p., 56 illus.

7. http://aivt.ftk.spbstu.ru/projects/gk2008/tests/nov08

Ицыксон В.М., Захаров А.В., Ахин М.Х., Мяснов А.В.

АВТОМАТИЧЕСКОЕ ОБНАРУЖЕНИЕ ДЕФЕКТОВ ПРОГРАММНЫХ СИСТЕМ НА ОСНОВЕ МЕТОДА ПРОВЕРКИ МОДЕЛИ

Введение

В настоящее время одной из основных особенностей большинства программных систем (ПС) является их сложность. В некоторых областях объемы исходных кодов ПС могут достигать десятков миллионов строк. Такие объемы и сложная структура ПС, сложность языков программирования могут приводить к появлению ошибок в ПС. Особое место занимает специальный класс ошибок – программные дефекты – ошибки, приводящие к деградации программного обеспечения.

Программные дефекты, как правило, возникают на этапе кодирования по следующим причинам:

- неправильная интерпретация семантики языковых конструкций;
- некорректное использование системных и/или библиотечных функций;
- использование программных объектов и ресурсов без соблюдения требуемых протоколов вызовов;
- и др.

Дефекты в ПС в процессе работы могут приводить к отказам ПС, а также могут быть причиной возникновения потенциальных уязвимостей, тем самым значительно ухудшая их безопасность и надежность. То есть программные дефекты являются причиной снижения качества ПС [1].

Стоимость выявления и исправления ошибок в ПС очень высока, при этом стоимость возрастает на каждой следующей стадии жизненного цикла проектирования программного обеспечения. Таким образом, задача разработки технологий автоматического или автоматизированного обнаружения программных дефектов является актуальной.

Состояние предметной области

Выделяют две основные группы методов обнаружения дефектов: на базе статического или динамического анализа [2].

Динамический анализ заключается в проверке программного обеспечения во время или по результатам его выполнения. Основным достоинством такого подхода является то, что анализ выполняется на основе информации о реальном поведении программы во время выполнения. К недостаткам динамического анализа следует отнести сложность повторения пути выполнения программы, изменение временной диаграммы анализируемой системы в результате внедрения в исходный код дополнительных служебных вызовов, учет ограниченного множества возможных путей выполнения программы.

Статический анализ осуществляет проверку программного обеспечения без его выполнения, только на основе исходного кода программ. Преимуществом статического анализа перед

семейством динамических подходов является учет всех возможных путей выполнения программы и возможность его использования на любых этапах разработки. Также статический анализ не требует выполнения программы, что позволяет использовать его до появления первой работающей версии, тем самым сокращая расходы на устранение дефектов за счет уменьшения времени между появлением и обнаружением программных ошибок. К недостаткам относятся потенциальное наличие большого числа ложных срабатываний и значительная вычислительная сложность, в том числе из-за наличия различных неопределенностей времени выполнения относительно поведения программы (неопределенных условий, рекурсий, исключений, полиморфизма, и т.д.). Это усложняет использование статического анализа для крупных проектов.

Оба подхода, описанные выше, показывают удовлетворительные результаты при поиске определенных типов дефектов, таких как выход за границы массива, ошибки инициализации и т.п. В то же время для обнаружения важной группы дефектов, связанной с корректной последовательностью вызовов определенных функций, так называемых ошибок протоколов вызовов, использование таких подходов не позволяет достичь приемлемых результатов.

Ошибки протоколов вызовов могут приводить к утечкам ресурсов, нарушениям доступа к памяти, непредсказуемому поведению программы, зацикливанию, зависанию и пр. Для обнаружения подобных дефектов наиболее эффективным является использование аппарата темпоральной логики и основанного на нем метода проверки моделей (model checking) [3].

Данная работа посвящена разработке комплексного подхода, предназначенного для обнаружения указанной группы дефектов, который основывается на расширении статического анализа методами проверки моделей. Ключевыми задачами, решаемыми при обнаружении дефектов с помощью метода проверки модели, являются автоматическое построение модели программы и ее верификация на основе проверки темпоральных свойств.

Первая из указанных задач – автоматическое построение модели программы – является самостоятельной сложной задачей. Существует множество работ, посвященных этой проблеме, однако полного законченного решения для реальных языков программирования не существует. Основные причины этого следующие:

- отсутствие формальной семантики для большинства языков программирования;
- принципиальные различия языков программирования и входных языков верификаторов;
- необходимость обеспечения адекватности модели;
- сложность моделирования некоторых типов данных языка программирования на языке верификатора;
- необходимость применения абстракции для построения такой модели реальной ПС, которая может быть верифицирована за приемлемое время с использованием ограниченных вычислительных ресурсов.

Тем не менее, к настоящему времени уже создано несколько подходов и программных средств, позволяющих в той или иной степени решить указанную задачу. Для построения моделей программ на языке Java существуют проекты Java2Spin в составе JCAT [4], Java PathFinder 1 [5], Bandera [6]; для языка C – modex в составе FeaVer [7], c2sdl [8].

Более подробный анализ проблемы автоматического построения модели по коду программы является предметом отдельного исследования и находится за рамками данной статьи.

Настоящая статья посвящена рассмотрению второй задачи – обнаружению дефектов с помощью метода проверки моделей.

Метод проверки моделей для обнаружения ошибок протоколов вызовов

В данной статье рассматривается подход на основе метода проверки моделей в применении к обнаружению дефектов, связанных с нарушением корректного протокола работы с библиотеками, модулями, компонентами в исходном коде программ на языке C. Нарушение корректного протокола работы может приводить, например, к утечке ресурсов, ошибкам работы с памятью и аварийному завершению приложения. В этой статье рассматриваются дефекты в программах на языке C, так как язык C является одним из самых используемых языков программирования, который часто применяется для разработки системного программного обеспечения, а также для построения критически важных систем. Однако предложенный подход может быть легко адаптирован и для других языков программирования высокого уровня.

В данной работе для проверки модели используется свободно-распространяемый верификатор Spin [9], в котором требования спецификации представляются в виде формул линейной темпоральной логики. Выбор верификатора Spin не накладывает ограничений на разрабатываемый подход, но значительно облегчает создание модели. В данном верификаторе используется язык Promela, выразительная мощность которого несколько лучше языков других верификаторов подходит для построения моделей программ на языке С: близкая система типов, наличие конструкций ветвления и циклов и т.п.

Цель предлагаемого подхода – автоматическая проверка корректности протокола вызовов функций в ПС на языке С. Основные положения предлагаемого подхода:

• построение модели ПС осуществляется с использованием статического анализа исходного кода ПС на языке С;

• для описания семантики анализируемых функций используются внешние аннотации;

• спецификация корректного протокола вызовов функций выражается с помощью формул линейной темпоральной логики;

• метод проверки модели применяется для обнаружения дефектов и получения трассы.

Для построения модели используется статический анализ исходного кода программы, который включает следующие этапы:

• лексический анализ, результатом которого является последовательность токенов;

• синтаксический анализ, в результате которого строится абстрактное синтаксическое дерево;

• построение графа потока управления;

• построение модели программы на языке верификатора *Spin*.

Аннотации служат для спецификации анализируемого набора функций. Использование аннотаций позволяет однозначно идентифицировать требуемые функции в исходном коде ПС, а также создавать модель указанных функций на языке верификатора.

В результате применения статического анализа производится построение модели программы на входном языке верификатора Spin, при этом

• неинтересующие программные объекты и типы удаляются из модели;

• интересующие программные объекты представляются с помощью переменных встроенных типов языка *Promela*;

• указатели моделируются с помощью переменных типа *int* языка *Promela*; присваивание указателю другого указателя или арифметические операции с указателями моделируются соответствующими операциями над целочисленными переменными модели;

• операции разадресации указателей моделируются индексированием в массиве указателей;

• вызов функций моделируется запуском процессов, передача возвращаемого значения – с помощью каналов;

• ветвления моделируются операторами ветвления языка *Promela*;

• циклы моделируются циклами языка *Promela*;

• в зависимости от возможностей моделирования программных конструкций и интересующих программных условий операторы циклов и ветвлений могут моделироваться условиями над соответствующими переменными модели или недетерминированным выбором.

Для аннотированного набора функций разрабатывается формула (множество формул), выражающая темпоральные свойства корректной последовательности вызовов. С помощью верификатора модель программной системы проверяется на соответствие данной формуле. По результатам проверки можно судить о правильности работы со специфицированными протоколами.

На рисунке 1 приведена общая схема процесса анализа программной системы на основе предлагаемого подхода к обнаружению дефектов.

Анализируемый набор функций, как правило, предназначен для решения определенного круга задач, например файловый ввод–вывод, работа с динамической памятью, построение изображений, многопоточное исполнение, синхронизация потоков исполнения, передача данных по сети и т.д. Для решения указанных задач данным набором функций используются общие программные объекты, которые с точки зрения программного интерфейса могут выглядеть как целочисленные дескрипторы, указатели, структуры. Чаще всего для представления ресурсов используются дескрипторы и указатели, поэтому в данной статье рассматриваются именно эти способы представления. В таких наборах функций можно выделить функции-генераторы, функции-пользователи и функции-уничтожители ресурсов некоторого типа. Для моделирования

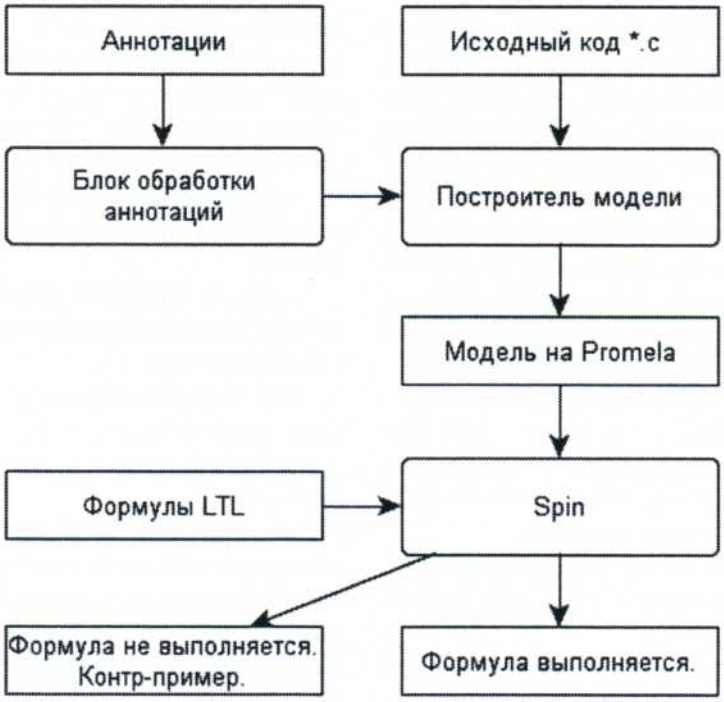


Рис. 1. Общая схема процесса анализа

подобных функций предлагается использовать следующий подход.

Количество доступных ресурсов задается параметром модели N. Для учета выделенных ресурсов создается массив соответствующей размерности N, в котором ведется учет занятых ресурсов. При выделении ресурса в модели функции-генератора выполняется поиск первого свободного ресурса в массиве. Если свободный ресурс найден, он отмечается как используемый и возвращается как результат работы функции. В зависимости от типа ресурса вводится биективное отображение индекса в массиве на множество допустимых значений указателя или дескриптора, например:

$$f(i) = i + 1,$$
$$f_{обр}(i) = i - 1.$$

Если свободный ресурс не найден, возвращается код ошибки.

В модели функции-пользователя ресурсов с помощью $f_{обр}$ получается индекс i в массиве, проверяется, что значение индекса лежит в диапазоне массива, и данный ресурс отмечается как выделенный.

В модели функции-уничтожителя ресурсов с помощью $f_{обр}$ получается индекс i в массиве, проверяется, что значение индекса лежит в диапазоне массива, ресурс по индексу i был выделен, после чего ресурс освобождается.

Верификатор *Spin* позволяет в формулах линейной темпоральной логики оперировать только глобальными переменными. Для формализации свойства "была вызвана функция f" предлагается ввести глобальную переменную *cmd*: при моделировании вызова функции в начале выполнения соответствующего процесса переменной cmd присваивается константа *op_f*, в конце выполнения – значение *op_init*.

Тогда для проверки того, что ресурс i был выделен функцией f и освобожден функцией g, можно использовать следующую обобщенную формулу темпоральной логики:

#define APf ((cmd == op_f) && (res[i] > 0))
#define APg ((cmd == op_g) && (res[i] == 0)),
[] (APf –> (<> APg))

где

- *APf* – атомарный предикат, выражающий, что ресурс был выделен;
- *APg* – атомарный предикат, выражающий, что ресурс был освобожден;
- *res* – массив ресурсов;
- *[]* – оператор всеобщности (*G*);
- <> – оператор "когда-то в будущем" (*F*);
- –> – импликация.

Пример применения подхода для функций работы с динамической памятью

Для иллюстрации работоспособности разработанного подхода применим его для

стандартных функций работы с памятью. Для простоты возьмем сокращенный набор функций управления памятью: *malloc, realloc, free*.

• Функция *malloc* выделяет память указанного размера. При успешном завершении выполнения функции возвращается указатель на выделенную память; в противном случае возвращается нулевой указатель.

• Функция *realloc* изменяет размер выделенной области памяти. В функцию передается указатель на выделенную ранее (с помощью *malloc* или *realloc*) область памяти и новый размер выделяемой памяти в байтах. Переданный указатель может быть нулевым. Функция возвращает указатель на вновь выделенную область памяти, при этом он может совпадать или не совпадать с ранее выделенным. Если не удается выделить запрошенное количество байт, функция возвращает *NULL*.

• Функция *free* освобождает выделенную ранее память.

Можно выделить следующие возможные ошибки, связанные с рассмотренным набором функций.

1. Разыменование нулевого указателя. Функции *malloc* и *realloc* могут вернуть указатель *NULL*, который затем может быть ошибочно разыменован.

2. Разыменование указателя, указывающего на непринадлежащую программе память.

3. Утечка памяти. Память, выделенная с помощью *malloc* и/или *realloc*, не освобождается после завершения работы с ней.

4. Многократное освобождение памяти. Многократный вызов *free* к одному и тому же указателю.

Допустимую последовательность действий (корректный протокол работы) для заданного набора функций можно выразить недетерминированным конечным автоматом, представленным на рисунке 2.

Для демонстрации работоспособности предложенного подхода был разработан пример программы на языке С, в котором специально нарушается естественный порядок вызова библиотечных функций работы с динамической памятью [10].

На основе данной программы была построена модель на языке *Promela* [10].

Для моделирования функций *malloc* и free используются специальные процессы с соответствующими именами. Процесс *malloc*

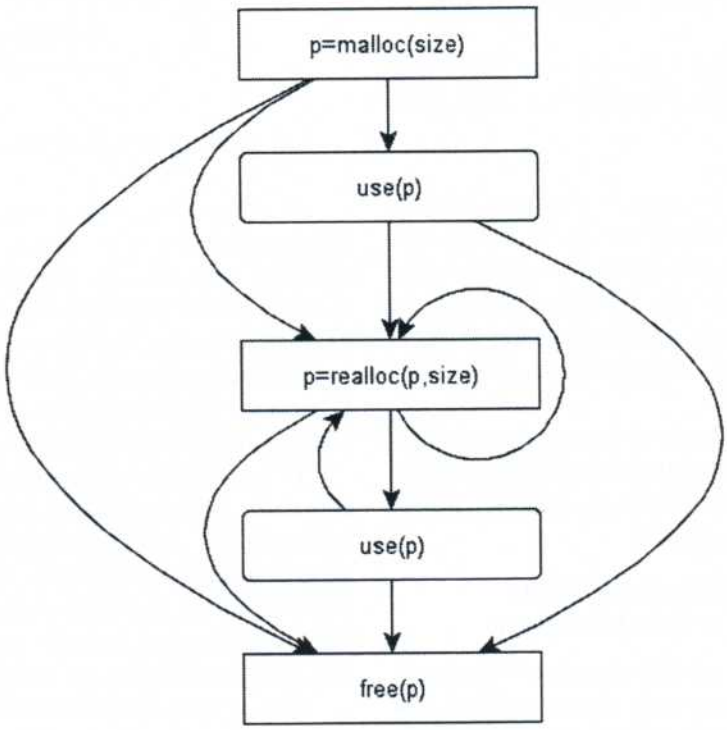


Рис. 2. НКА для протокола работы с *malloc*, *realloc* и *free*

моделирует выделение памяти. Возвращаемый указатель является индексом в массиве указателей, в котором отмечается, что память по данному индексу выделена. При удалении указателя с помощью оператора проверки (assert) обеспечивается защита от повторного освобождения памяти.

Темпоральное правило, которое необходимо проверить, – указатель, выделенный с помощью malloc, по всем возможным путям вычислений будет освобожден при помощи вызова free, – выражается следующей формулой темпоральной логики:

#define m ((cmd == op_malloc) && (ptr[i] > 0))
#define f ((cmd == op_free) && (ptr[i] == 0)).
[] (m –> (<> f))

Построенная модель тестовой программы и указанная формула темпоральной логики были поданы на вход верификатора Spin. В результате проверки данной модели программы была определена невыполнимость указанной выше темпоральной формулы, что говорит о наличии дефекта (ошибки протокола вызова) в данной программе. Место и условия проявления ошибки можно идентифицировать путем анализа полученного контр-примера.

Правило, проверяющее, что функция free никогда не будет вызвана для невыделенного ресурса, можно выразить следующей формулой:

#define free_unallocated ((cmd == op_free) && (sec == sec_before) && (free_arg == (i+1)) && (ptr[i] == 0)).
[] (!free_unallocated)

В этой формуле вводится понятие секции (sec == sec_before). Это позволяет указать, что перед выполнением функции free должны быть проверены остальные условия формулы: для указателя, переданного в виде параметра функции free, ресурс по соответствующему индексу не выделен.

Построенная модель и рассматриваемая формула были проверены с помощью *Spin*. В результате был получен контр-пример, показывающий наличие дефекта.

Пример применения подхода для функций работы с файлами

Второй пример иллюстрирует применимость подхода для анализа функций работы с файлами. Рассмотрим сокращенный набор функций, входящий в библиотеку "*stdio.h*": *fopen, fprintf, fclose*.

• Функция *fopen* открывает файл с заданным именем и режимом доступа. При невозможности открытия файла функция возвращает нулевой указатель.

• Функция *fprintf* выводит в файл отформатированную строку. При успешном завершении выполнения возвращается количество записанных байт, при ошибке – отрицательное значение.

• Функция *fclose* закрывает файл. Значение 0 соответствует корректному закрытию файла, значение *EOF* – ошибке работы с файлом.

Допустимую последовательность вызовов для заданного набора функций можно выразить недетерминированным конечным автоматом (рис. 3):

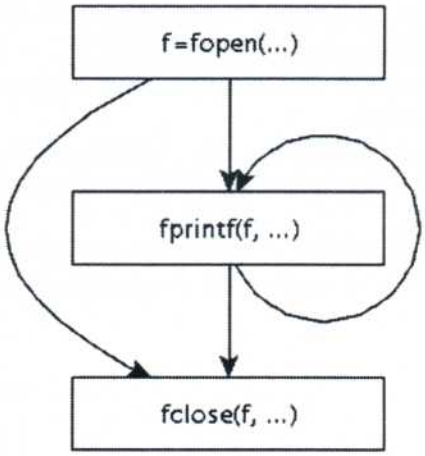


Рис. 3. НКА для функций *fopen, fprintf, fclose*

Для демонстрации работоспособности предложенного подхода был разработан пример программы на языке С, в котором специально нарушается естественный порядок вызова функций работы с файлами [10].

По рассматриваемой программе на языке С может быть построена соответствующая модель на языке *Promela* [10].

В данном примере рассмотрено множество функций, используемых для протоколирования программы. Функция *log_init* инициализирует внутренние переменные протоколирования; функция *log_message* используется для вывода сообщений в файл протокола; функция *log_deinit* освобождает ресурсы.

В построенной модели функция *fopen* представляется процессом *fopen*, в котором возвращаемый функцией *fopen* указатель на структуру *FILE* моделируется индексом в

массиве ресурсов. По индексу в массиве отмечается, что ресурс был выделен (файл открыт). Процесс fclose сначала выполняет проверку того, что освобождаемый ресурс действительно был выделен, после чего моделирует освобождение ресурса – обнуление соответствующего элемента массива.

Правило, которое необходимо проверить, – указатель, выделенный с помощью fopen, по всем возможным путям выполнения рано или поздно будет освобожден при помощи fclose, может быть задано следующей формулой темпоральной логики:

#define fo ((cmd == op_fopen) && (fdes[i] > 0))

#define fc ((cmd == op_fclose) && (fdes[i] == 0))

[] (fo –> (<> fc)).

В результате проверки для данной модели программы верификатор Spin выдает контрпример, который может быть проанализирован для идентификации места и условия возникновения дефекта.

Заключение

Для выбранного класса дефектов – ошибки протоколов вызовов функций – предложен подход к обнаружению этих дефектов и показана его применимость на примере работы с двумя типами ресурсов. Предложенный подход позволяет достаточно просто выражать темпоральные свойства программ и применять эффективные существующие средства проверки моделей, которые используют специальные подходы для сокращения пространства состояний и позволяют за конечное время проверить соответствие модели программы и спецификации. Существенным достоинством подхода является возможность в случае нарушений требований спецификации получить контрпример, идентифицирующий место и условие проявления дефекта.

Предложенные в статье подходы к моделированию программных ресурсов позволяют во многих случаях решить такие сложные проблемы статического анализа, как обнаружение утечек динамической памяти, файловых дескрипторов и т.п.

Одним из направлений развития описанного подхода является его адаптация для решения задачи обнаружения дефектов в многопоточных приложениях, которые обычно плохо поддаются классическому статическому анализу. Использование подхода на основе проверки моделей позволяет справиться с этой проблемой.

Исследование выполнено в рамках работ по государственному контракту № 02.514.11.4081 "Исследование и разработка системы автоматического обнаружения дефектов в исходном коде программного обеспечения" Федерального агентства по науке и инновациям в рамках Федеральной целевой программы "Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2012 годы".

СПИСОК ЛИТЕРАТУРЫ

1. ISO/IEC 9126–1:2001

2. Software Analysis: a Roadmap / D. Jackson, M. Rinard // Proceedings of the Conference on the Future of Software Engineering. – 2000. – p.133–145

3. Верификация моделей программ: Model Checking / Э. Кларк, О. Грамберг, Д. Пелед. – М. : МЦНМО, 2002.

4. Modeling and Validation of Java Multithreading Applications Using SPIN / C. Demartini, R. Iosif, R. Sisto // In Proc. 4th International SPIN Workshop. – 1998

5. Model Checking Java Programs Using Java PathFinder / K. Havelund, T. Pressburger // International Journal on Software Tools for Technology Transfer (STTT). – 2000

6. Bandera: Extracting Finite–state Models from Java Source Code / J. C. Corbett, M. B. Dwyer, J. Hatcliff, S. Laubach et al. // In Proc. of the 22st International Conference on Software Engineering. – 2000

7. Automating Software Feature Verification / G.J. Holzmann, M.H. Smith // Bell Labs Technical Journal. – 2000. – Volume 5, Number 2. – p.72–87

8. Formal Verification of Software Source Code Through Semi–Automatic Modeling / C. Eisner // Software and Systems Modeling (SOSYM). – 2005. – Volume 4, Number 1

9. http://www.spinroot.com/

10. http://aivt.ftk.spbstu.ru/projects/gk2008/verification/samples/

Савельев М.В., Аликов А.Ю., Федосеев С.В.

Синтез простых структур и наборов импульсных управляющих воздействий

При проектировании системы управления многомерным объектом решается задача выбора набора управляющих величин. Традиционно такой выбор очевиден и определяется сложившейся традицией управления данным классом объектов или обусловлен отсутствием достаточно широкого набора альтернативных величин, допускающих их использование в качестве управляющих переменных. При этом рациональный выбор в некотором смысле наилучшего набора таких величин оказывается неочевидным, особенно, если учитывать требования к динамике процессов управления и ограничения, накладываемые на переменные управления и состояния.

Последующий обоснованный выбор конкретного набора управляющих воздействий, не обладающего избыточностью и вместе с тем достаточного для реализации заданных динамических характеристик системы управления, требует проведения весьма трудоемких расчетов, особенно при необходимости учета динамики вспомогательных механизмов управления, а также ограничений, накладываемых на переменные управления и состояния.

В связи с высокой трудоемкостью анализа вариантов синтезируемых систем рассматриваемого класса сложился эмпирический подход, предполагающий последовательное наращивание размерности вектора управления и амплитуд управляющих воздействий до уровня, достаточного для обеспечения назначенных динамических требований к системе управления. Такое наращивание осуществляют путем последовательного добавления к вектору управления очередного управляющего воздействия, упорядоченного в порядке убывания предпочтительности использования очередного воздействия. Указанный подход обеспечивает выявление неизбыточного по структуре вектора управления. Однако он не может быть признан достаточно строгим методом выявления наилучшего варианта управления, так как не предполагает исследования всего множества сочетаний управляющих воздействий и не определяет явно и достаточно строго правило сравнения вариантов по их предпочтительности.

Задача исчерпывающего перебора и анализа всех возможных сочетаний управляющих величин из заданного списка рассмотрена в работе [1] на примере синтеза системы. В ней выделение наилучшего варианта основано на использовании суммы бальных оценок технической реализации той или иной переменной управления. В указанной работе не ставится и не решается задача выделения всего множества минимально-факторных решений на основе правила минимально-факторного выбора, оценка приемлемости вариантов ведется без учета ограничений на диапазон допустимых значений переменных управления и состояния, а также допустимых отклонений получаемой траектории от программной.

Отличие предлагаемого подхода к формализации задачи поиска неизбыточных наборов управляющих величин состоит в том, что он предполагает выявление множества всех наборов воздействий, необходимых и достаточных для реализации требуемых траекторий движения динамической системы, а также то, что он приводит к достаточно просто проверяемым условиям приемлемости варианта управления, вместе с тем учитывающим ограничения, накладываемые на переменные управления и состояния, а также на отклонение получаемой траектории от программной.

При формализации рассматриваемой задачи будем учитывать следующие особенности.

Задачи синтеза систем управления, решение которых математически описывается конечным вектором

$$x = (x_1, x_2, \dots, x_i, \dots, x_n),$$

с компонентами принимают значение из некоторых в общем случае произвольных множеств.

Структурой решения будет разбиение множества компонент вектора решения на подмножества нулевых и активных компонент. Структуру решения можно однозначно определить, задав набор активных компонент вектора решения и указав набор

$$S = \{q, \ldots, p\}$$

номеров активных координат вектора решения. При этом из $i \in S$ следует, что x_i – активная компонента,

Объект управления определяется системой уравнений

$$dx/dt = Ax + Bu,\ x(0) = x_0 \qquad (1)$$

где x – n-мерный вектор состояния, u – m-мерный вектор управления, t – время, A, B – известные матрицы соответствующих размерностей, компоненты x, u, A, B принимают значения из множества действительных чисел.

Назначена контрольная траектория движения $x^*(t,x_0)$, $t \in [0,T]$, которую необходимо осуществить с заданной точностью.

Искомое управление должно удовлетворять условию $u \in U$, а соответствующая ему траектория движения системы (1) – условию $x \in X$, где U, X – области допустимых значений векторов u и x соответственно.

Требуется найти все наборы S компонент вектора управления u, отклонение которых от нулевого значения есть необходимое и достаточное условие осуществления с заданной точностью контрольной траектории движения. При этом все компоненты вектора u, не вошедшие в набор S, полагаются тождественно равными нулю.

Вектором решения рассматриваемой задачи является вектор u. Его компоненты $u_j(t)$ – функции времени, отражающие изменение тех или иных физических величин, допускающих их использование в качестве переменных управления. Он удовлетворяет принятому в данной работе определению вектора решения.

Действительно, в соответствии с принятым нами определением вектора решения вектор u, во-первых, является конечномерным (содержит m компонент). Во-вторых, каждая его координата $u_j(t)$ указывают на наличие либо отсутствие элемента структуры синтезируемого объекта. Компонента $u_j(t)$, тождественно равная нулю, содержательно означает отказ от использования соответствующей ей физической величины в качестве переменной управления. Наоборот, наличие в векторе u компоненты, не равной тождественно нулю, означает использование соответствующей ей физической величины в качестве переменной управления. В-третьих, координаты вектора u, отличные от нуля, определяют количественные характеристики синтезируемого объекта – векторного закона управления $u(t)$, а именно – конкретные функции времени $u_j(t)$, обеспечивающие достижение поставленной цели управления: воспроизведение с заданной точностью назначенной траектории движения $x^*(t,x_0)$, $t \in [0,T]$.

Структура решения определяется набором S компонент вектора u, которые могут быть отличны от нуля. Учитывая, что каждой компоненте вектора u соответствует конкретная физическая величина, из множества величин, допускающих использование в качестве переменных управления, набор S определяет перечень физических величин, используемых для управления. Минимально-факторная структура решения S в данном случае определяет набор таких физических величин, использование которых для управления объектом есть необходимое и достаточное условие обеспечения заданной точности воспроизведения назначенной траектории $x^*(t,x_0)$, $t \in [0,T]$.

Иными словами, минимальная структура S вектора управления u не содержит компонент (физических величин, претендующих на использование в качестве переменных управления), которые можно исключить из S^0, обеспечив приемлемость получаемой траектории Движения объекта за счет допустимого изменения функций $u_j(t)$, оставшихся в структуре закона управления $u(t)$.

Важным частным случаем рассматриваемых векторов управления с минимальной структурой являются векторы, в которых число физических величин, используемых для управления, минимально. Такие векторы управления имеют минимальную структуру. Они являются предпочтительными по отношению к остальным векторам с минимально-факторной структурой, если все физические величины однородны (неразличимы) в смысле предпочтительности их включения в набор S.

В случае, когда представляется возможным указать весовые коэффициенты, оценивающие использования каждой физической величины, целесообразным оказывается

выявление множества векторов управления с минимально-взвешенной структурой.

Таким образом, в рамках рассматриваемой задачи исследуем различные варианты формализации правила сравнения сложности анализируемых структур векторов управления: правила минимально-факторного $\pi_{мф}$, минимально-взвешенного $\pi_{мв}$ сравнения и правила сравнения сложности структур по числу элементов $\pi_{м}$. Выполним поиск множества простых структур вектора управления u, где в качестве правила сравнения сложности могут использоваться правила .

Получим условия допустимости структуры решения S рассматриваемой задачи.

Представим требование близости реальной и контрольной траекторий в форме условий, допускающих относительно простую проверку.

Решение уравнения (1) можно, используя формулу Коши [2], записать в виде:

$$x(t,x_0)=x^0(t)+\int_0^T K(t,\tau)B(\tau)u(\tau)d\tau, \qquad (2)$$

где $x^0(t)$ – свободное движение объекта (1), соответствующее условиям $\{x(0) = x_0, u \equiv 0\}$; $K(t, \tau)$ – матрица Коши системы (1).
Для дискретного времени $\{t_0 = 0, t_1, t_2, \ldots t_{\mu-1}, t_\mu, \ldots, t_M = T\}$

$$x(\mu,x_0)=x^0(\mu)+$$
$$+\sum_{\xi=0}^{\mu-1}K(\mu,\xi+1)B_S(\xi)u_S(\xi),\ \mu=0,1,\ldots,M.$$

С учетом заданной структуры S вектора u имеем:

$$x(\mu,x_0)-x^0(\mu)=$$
$$=\sum_{\xi=0}^{\mu-1}K(\mu,\xi+1)B(\xi)u(\xi),\quad \mu=0,1,\ldots,M, \qquad (3)$$

где $B_S(\xi)$, $u_S(\xi)$ получены соответственно из $B(\xi)$, $u(\xi)$ исключением элементов, номер которых отсутствует в наборе S, перечисляющем номера активных компонент вектора u. Систему (3) представим в виде:

$$\Gamma_S(\mu)u_S = x(\mu, x_0) - x_0(\mu),\ \mu = 0, 1, \ldots, M, \qquad (4)$$

где $\Gamma_S(\mu)$ – матрица, получаемая в результате перехода от системы (3) к системе (4), $u_S = u_S(\xi)_{\xi=0,M-1}$. В конечном итоге приходим к системе

$$\Im_S u_S = \chi,$$

где $\Im_S = \Gamma_S(\mu)_{\mu=0,M}$, $\chi = (x(\mu)) - x^0(\mu)_{\mu=0,M}$.

Требование точного совпадения получаемой и контрольной траектории в каждый из моментов времени $\mu = 0, 1, \ldots, M$ представляется условием

$$\Im_S u_S = \chi^*, \qquad (5)$$

где $\chi^* = (x^*(0, x_0) - x^0(0), \ldots x^*(\mu, x_0) - x^0(\mu), \ldots, x^*(M, x_0) - x^0(M)$ – вектор-столбец, получаемый подстановкой в χ вместо дискретных значений $x(\mu, x_0)$ дискретных значений $x^*(\mu, x_0)$ для каждого $\mu = 0, 1, \ldots, M$.

Для заданного варианта S, учитывая равенство нулю всех координат вектора u_S, номера которых не включены в S, систему (5) для заданного S можно представить в виде:

$$\sum_{j\in S}\Im_j u_j = \chi^*, \qquad (6)$$

где $\Im_i$ – j-ый столбец матрицы $\Im$; μ_i – j-ая компонента вектора u.

С переходом из (6) получим следующие варианты условий достаточной близости получаемой и контрольной траекторий:

$$\chi^- \prec \sum_{j\in S}\Im_j u_j \prec \chi^+, \qquad (7)$$

$$(\sum_{j\in S}\Im_j u_j - \chi^*)^T(\sum_{j\in S}\Im_j u_j - \chi^*) \le \Delta_\aleph, \qquad (8)$$

где χ^-, χ^+ – назначенный диапазон допустимых значений вектора $\chi_S = \Im_S u_S = \sum_{j\in S}\Im_j k_j$; вектор χ^* является "номиналом" вектора χ; $\Delta_\aleph$ – квадрат максимально допустимой длины невязки $\chi_S - \chi^*$ (заметим, что среднеквадратическое отклонение траектории $x(t, x_0)$ от $x^*(t, x_0)$ в моменты времени $\mu = 0, 1, \ldots, M$ есть $\Delta_\aleph/(M + 1)$.

Требование $u \in U$ конкретизируем в форме условия

$$U^- \le D_S u_S \le U^+, \qquad (9)$$

где U^-, U^+, D – заданные векторы и матрица соответствующих размерностей. Ограничения на переменные состояния $x \in X$,

Необходимо определить программное импульсное управление, обеспечивающее прохождение траектории движения динамического объекта через заданное множество точек.

К такой задаче может быть приведена задача расчета программы управления ракетой с импульсными (например, пиротехническими)

корректирующими двигателями, осуществляющей на начальной фазе полета программный маневр выхода в луч, в котором далее ее удерживает система стабилизации [3].

Объект управления при импульсном управлении определяется системой уравнений:

$$dx/dt = Ax + bu,\ x(0) = x_0, \qquad (10)$$

где x – n-мерный вектор состояния, u – скалярная функция управления, A, b – известные матрица и вектор соответствующих размерностей, компоненты x, u, A, B принимают значения из множества действительных чисел. В общем случае элементы матрицы A и вектора b могут быть функциями времени.

Назначено множество точек $x^*(t_i), i = 1, \ldots, N$, через которые должна проходить программная траектория движения.

Управляющее воздействие носит импульсный характер и допускает идеализированное представление в виде:

$$u(t) = \sum_{j \in S} \delta(t - t_j)u_j, \qquad (11)$$

где $\delta(t)$ – дельта-функция Дирака, u_j – амплитуда управления в j-ый момент времени, набор S указывает номера моментов времени из заданного ряда $\{t_0 = 0, t_1, t_2, \ldots t_{\mu-1}, t_\mu, \ldots, t_M = T\}$, в которые управление может быть отлично от нуля, т.е. $S \subseteq \{0, 1, \ldots, M\}$.

Решение рассматриваемой задачи определяется вектором $u = (u_j)_{j=0,M}$. Структура решения определяется набором S номеров активных компонент вектора u. Набор S, таким образом, определяет перечень моментов времени, в которые происходит подача импульсов управления. Минимально-факторная структура решения в данном случае определяет набор S^0 таких моментов времени, в которые подача управляющих импульсов есть необходимое и достаточное условие прохождения траектории движения объекта (10) через множество назначенных точек $x(t_i), i = 1, \ldots, N$.

Иными словами, программа управления с простой (минимально-факторной) структурой не содержит импульсов, которые можно исключить, обеспечив приемлемость получаемой траектории за счет изменения амплитуд оставшихся импульсов, то есть она не содержит избыточных импульсов.

Важным частным случаем программ управления с минимально-факторной структурой являются программы, предполагающие подачу минимально возможного числа управляющих импульсов, необходимых для реализации назначенной траектории движения с заданной точностью, то есть программы с минимальной структурой.

Такие программы являются предпочтительными по отношению к остальным законам управления с минимально-факторной структурой, если номера моментов времени подачи управляющих импульсов однородны (неразличимы) в смысле предпочтительности их включения в набор S.

Получим условия допустимости структуры решения S рассматриваемой задачи. Воспользуемся представлением системы (10) вида (2). Учитывая характер управления (11), получим:

$$x(t, x_0) = x^0(t) + \sum_{j \in S \cap Q(t)} K(t, t_j)b(t_j)u_j, \qquad (12)$$

где $Q(t)$ – множество номеров моментов времени, составляющих последовательность $\{t_0, t_1, \ldots, t_j \le t\}$; $x^0(t)$ – свободное движение объекта (10), соответствующее условиям $\{x(0) = x_0, u \equiv 0\}$; $K(t, t_j)$ – матрица Коши системы (10) (отметим, что рассматриваемый класс задач характеризуется скорее допустимостью идеализации (12), чем (11)).

С учетом (12) требование прохождения траектории движения системы (10) через точки $x^*(t_i)$, $i = 1, \ldots, N$ эквивалентно условию

$$\sum_{j \in S \cap Q(t)} K(t, t_j)b(t_j)u_j = x^*(t_i) - x^0(t_i),\ i = 1, \ldots, N, \qquad (13)$$

или

$$\Gamma_S(i)u_S = x^*(t_i, x_0) - x^0(t_i),\ i = 1, \ldots, N, \qquad (14)$$

где $\Gamma_S(i)$ – матрица, получаемая в результате перехода от системы (13) к системе (14), $u_S = (u_i)_{j \in S}$. В конечном итоге приходим к системе уравнений:

$$\Im_S u_S = \chi^*,$$

где $\Im_S = (\Gamma_S(i))_{i=0,N}$, $\chi^* = (x^*(t_i, x_0) - x^0(t_i))_i = 0, N$. Полученная система эквивалентна системе (5) и отражает условие точного прохождения траектории движения через назначенные точки $x^*(t_i)$, $i = 1, \ldots, N$.

Переход от требований точной к требованием приближенной реализации назначенной траектории приводит к условиям допустимости структуры S вектора управления u в виде одного из соотношений (6), (7), (8) совместно с требованиями (9) в зависимости от варианта формализации задачи.

Применительно к рассматриваемой задаче на практике в качестве ограничения, накладываемого на управление, помимо условия (9), часто рассматривается также условие одинаковости модулей всех ненулевых управляющих импульсов, т.е. условие

$$u_j = 0,$$

или

$$|u_j| = u_m, j = 0, 1, \ldots, M, \qquad (15)$$

где u_m – заранее назначенное либо выбираемое число.

Полученные результаты позволяют сделать следующие выводы.

Рассматриваемую задачу синтеза программных импульсных управлений с неизбыточной структурой можно представить как задачу поиска простых структур. Задача поиска неизбыточных структур в общем случае состоит в определении всех допустимых структур, для каждой из которых нельзя указать менее сложную допустимую структуру. В формальной интерпретации данная задача состоит в определении множества

$$\Omega^0 = \{S^0 \in \Omega_\partial : \{S \in \Omega_\partial : S\pi S^0\} = \varnothing,$$

где π – бинарное отношение, отражающее понятие "проще, чем ... ".

Варианты математической постановки представленной задачи поиска неизбыточных структур могут различаться используемым правилом сравнения сложности структур и характером описания множества допустимых структур Ω_∂, в которой условием допустимости структуры является выполнение одного из соотношений (6), (7), (8) совместно с системой требований (9) либо (15).

Из анализа полученных условий допустимости следует, что анализируемая задача сводится к задаче поиска простых структур с линейными ограничениями в случае использования условий (6) или (7) совместно с системой (9) и к задаче поиска простых структур с частично линейными ограничениями в случае использования иных сочетаний полученных условий допустимости.

В более общем случае при математическом описании рассматриваемой задачи для сравнения структур, кроме правила $\pi_{мф}$, можно использовать также правило сравнения $\pi_м$ или $\pi_{мв}$. При этом анализируемая задача сводится к общей постановке задачи поиска структурно-неизбыточных решений, а именно к задаче поиска множества $\Omega_{i\partial}$ простых (минимально-факторных) структур состоит в определении множества

$$\Omega_{мф} = \{S^0 \in \Omega_\partial : \{S \in \Omega_\partial : S\pi_{мф}S^0\} = \varnothing$$

поиска простых структур с теми же ограничениями, определяющими условия допустимости сравниваемых структур. В результате осуществленной формализации анализируемая задача сводится к задаче поиска простых структур с частично линейными ограничениями.

СПИСОК ЛИТЕРАТУРЫ

1. **Понтрягин Л.С., Болтянский В.Г., Гамкрелидзе Р.В., Мищенко Е.Ф.** Математическая теория оптимальных процессов. – М.: Наука, 1969. – 232 с.

2. **Красовский Н.Н.** Теория управления движением. – М.: Наука, 1968. – 476 с.

3. **Параев Ю.И., Смагин В.И.** Задачи упрощения структуры оптимальных регуляторов. // Автоматика и телемеханика. – 1975. № 26. – С. 180–183.

Чебыкин С.В.

Методика производственного планирования выпуска листового стекла

Современный рынок функционирует в условиях жесткой конкуренции и требует от предприятий рационального и взвешенного подхода к планированию деятельности и управлению производственными процессами. В настоящее время существует ряд методов и методологий производственного планирования, а уровень развития информационных технологий позволяет использовать эти методы практически без ограничений.

Однако задачу создания эффективных автоматизированных систем планирования и управления нельзя считать полностью решенной: универсальных методов, способных учитывать все особенности процессов планирования и управления на каждом конкретном предприятии, не существует. Имеющиеся методы и системы или представляют собой достаточно общие решения или стандарты (прежде всего это относится к системам класса ERP), или ориентированы на узкий круг задач конкретного предприятия. В любом случае внедрение комплексного решения для управления предприятием требует времени, солидных капиталовложений и несет в себе серьезные риски [5]. Помочь решить названные выше проблемы призваны концепции производственного планирования и управления APS (Advanced Planning & Scheduling), MES (Manufacturing Execution Systems). С одной стороны, они довольно жестко регламентируют структуру и функции систем управления предприятиями. С другой стороны, оставляют широкие возможности для разработки и реализации в их рамках методик планирования и управления с учетом специфики конкретного производства.

В настоящей работе представлена методика производственного планирования, согласующаяся с концепцией Advanced Planning and Scheduling и предназначенная для производственного объединения (ПО) "Полированное стекло" ОАО AGC Борский стекольный завод – крупнейшего в России производителя полированного и автомобильного стекла.

В структуре управления предприятием можно выделить 3 уровня иерархии, различающиеся горизонтом и оперативностью управления, а также решаемыми задачами. Это уровень стратегического управления предприятием, уровень организационного управления производством и уровень организационно-технологического управления участками и оперативного управления технологическим оборудованием.

Производственное планирование, находящееся на среднем уровне иерархии, осуществляется в рамках ПО предприятия. Вопросы согласования деятельности производственных объединений решаются на верхнем уровне иерархии.

ПО "Полированное стекло" функционирует в рамках AGC БСЗ как самостоятельное предприятие, основная задача которого – производство листового полированного стекла различного назначения. Задачи материально-ресурсного обеспечения производства вынесены за его пределы.

В производственном объединении установлены две линии производства стекла суммарной производительностью около 1150 т/сутки, одна высокопроизводительная линия резки-раскроя стекла и два складских помещения.

Производство листового стекла флоат-способом имеет свои особенности, накладывающие ограничения на выбор методов производственного планирования [2]. Основная особенность – многотоннажность и инерционность технологического процесса выработки листового стекла. Несмотря на относительную узость номенклатуры выпускаемой продукции, инерционность процесса варки-выработки является здесь определяющим фактором при формировании портфеля заказов и календарном планировании производства. Так, переналадка технологической линии, связанная с изменением толщины стекла, может потребовать несколько десятков минут. При высокой производительности линии варки-выработки (до 600 т/сутки) потери при переналадках могут составить существенную величину.

Планово-учетной единицей в производстве листового стекла является заказ, предусматривающий изготовление определенного количества

листов стекла определенных размеров и толщин. Номенклатура заказа может варьироваться в широких пределах, однако параметры заказа – длина, ширина, а в особенности толщина листов стекла – могут принимать ограниченный набор значений. Поэтому при планировании производства имеет смысл объединять заказы в партии по той позиции номенклатуры, что требует больше затрат на ее изменение.

В этих условиях формирование портфеля заказов на производство и календарное планирование производства представляют собой достаточно сложную оптимизационную задачу. Эта задача осложняется еще тем, что на производстве имеется две линии варки-выработки стекла, что требует решать задачу оптимальной загрузки производственных мощностей. Кроме этого, при планировании производства необходимо учитывать необходимость регулярного проведения профилактических и ремонтных работ на технологическом оборудовании.

Поступление заказов на производство стекла носит случайный характер. Поток заказов не является стационарным [4], он подвержен воздействию со стороны различных экономических факторов (в частности, сезонные колебания спроса на различные виды продукции). Вследствие этого

а) распределение плотности вероятности потока заказов на производство является отправной точкой для определения продолжительности учетного периода, на который формируется портфель заказов; для ПО "Полированное стекло" учетный период соответствует одному календарному месяцу;

б) для обеспечения отгрузки заказов точно в срок требуется поддержание запасов готовой продукции и заготовок для дальнейшей переработки, что также следует учитывать при планировании производства.

К особенностям производственного планирования относится также необходимость согласования календарного плана выпуска листового стекла с планом проведения ремонтно-профилактических мероприятий на технологических линиях.

Основные задачи производственного планирования, решаемые в ПО "Полированное стекло", –

1) формирование портфеля заказов на производство,

2) расчет оптимального календарного плана производства,

3) внутрицеховое планирование, в т.ч. оперативная корректировка календарного плана и оптимизация загрузки производственных мощностей.

При этом задачи (2) и (3) используют в качестве входных данных информацию о портфеле заказов на производство. Вследствие этого задачу (1) можно рассматривать как общую задачу планирования производства, а задачи (2) и (3) как ее подзадачи. В то же время данные задачи взаимосвязаны и не должны рассматриваться изолированно друг от друга.

Эффективность производственного планирования имеет смысл оценивать с экономической точки зрения, опираясь на цели, стоящие перед руководством предприятия. Основной целью деятельности любого предприятия является максимизация прибыли, получаемой на некотором временном интервале. Прибыль – один из важнейших критериев, по которым оценивается деятельность предприятия. Поскольку количество заказов, поступивших на предприятие, может превышать его возможности, в текущий портфель должны выбираться заказы, приносящие наибольший доход.

При имеющемся портфеле заказов на листовое стекло в качестве дополнительного критерия оптимизации может быть выбрана цеховая себестоимость стекла. Структура формирования цеховой себестоимости имеет множество составляющих [2], которые в рассматриваемом случае можно разделить на условно-постоянные и управляемые затраты. В рамках решаемой задачи к условно-постоянным относятся те затраты, которые связаны с использованием энергетических, материальных и человеческих ресурсов: покупка сырья, топлива и электроэнергии, основная и дополнительная зарплаты сотрудников, расходы на эксплуатацию оборудования и т.д.

К управляемым затратам могут быть отнесены затраты, связанные с потерями стекла при переналадках технологического оборудования, с потерями при хранении готовой продукции и заготовок на складе. Эти затраты соотносятся с тем, как осуществляется производственное планирование, и зависят от принимаемых управленческих решений.

Следующим критерием является время выполнения заказа, т.е. срок, по истечению которого заказ должен быть отгружен потребителю. В случае нарушения срока на предприятие накладываются штрафные санкции, и здесь

задачей оптимизации является исключение или минимизация потерь вследствие налагаемых штрафов.

Таким образом, выделены следующие критерии эффективности производственного планирования в ПО "Полированное стекло":

1) прибыль, получаемая от реализации заказов,

2) размер штрафных санкций P за невыполнение планов по отгрузке заказов потребителям,

3) цеховая себестоимость вырабатываемого стекла C_{org}.

Прибыль, получаемая от реализации заказов, функционально связана с себестоимостью стекла, поэтому вместо критерия (1) следует использовать значение выручки G от реализации заказов.

Альтернативные варианты решения задачи производственного планирования должны оцениваться по нескольким критериям. Для оптимизации данная задача должна быть преобразована в однокритериальную. В качестве метода преобразования в настоящей работе было выбрано агрегирование критериев – свертывание частных критериев в один путем введения весовых коэффициентов:

$$K = a_1 G + a_2 C_{org} + a_3 P, \qquad (1)$$

где $a_1 \ldots a_3$ – весовые коэффициенты, которые могут принимать как положительные, так и отрицательные значения в зависимости от того, какой вклад соответствующий частный критерий вносит в общий критерий K. Значения весовых коэффициентов могут быть определены экспертным путем.

Выручка G, получаемая от реализации заказа, вычисляется как произведение стоимости 1 м^2 стекла определенной толщины C_S на площадь заказанного стекла S_{gl} этой толщины:

$$G = \sum_{i=1}^{N} C_s^i S_{gl}^i, \qquad (2)$$

где i – номер позиции спецификации заказа, N – число позиций заказа.

Размер штрафных санкций P зависит от количества дней D, на которое было задержано выполнение заказа:

$$P = D \cdot k_p, \qquad (3)$$

где k_p – коэффициент штрафа, отражающий размер неустойки за задержку с отгрузкой на один день.

Цеховая себестоимость C_{org} содержит следующие компоненты (исключая те, что рассматриваются в рамках настоящей работы как условно-постоянные):

- недополученная прибыль C_c, вызванная потерями при переналадках технологических линий на выпуск стекол различной толщины;
- складские расходы C_w.

Недополученная прибыль C_c рассчитывается следующим образом:

$$C_c = S'_{gl} \cdot t_t \cdot C_s^{\delta}, \qquad (4)$$

где S'_{gl} – площадь стекла, вырабатываемого за минуту; t_t – продолжительность переходного процесса, минут; C_s^{δ} – стоимость одного квадратного метра стекла толщины δ, на которую был осуществлен переход.

Складские потери C_w зависят от средней стоимости транспорта стекла, стоимости погрузочно-разгрузочных работ и процента боя стекла на складе. Формула нахождения складских потерь выглядит следующим образом:

$$C_w = S_{agl} \cdot (C_{tr} + C_{lu} + C_{av} k_b), \qquad (5)$$

где S_{agl} – суммарная площадь складируемого стекла; C_{tr} – средняя удельная стоимость транспорта стекла, руб/м^2; C_{lu} – удельная стоимость погрузочно-разгрузочных работ, руб/м^2; k_b – процент боя стекла, C_{av} – средняя удельная стоимость стекла, руб/м^2.

Задача производственного планирования изготовления листового стекла в ПО "Полированное стекло" может быть сформулирована как задача обеспечения выпуска требуемого количества стекла точно в срок и с наименьшими затратами.

Задачу формирования портфеля заказов, рассматривая ее в качестве базовой задачи планирования производства, можно сформулировать как задачу о рюкзаке [1]. Формальная постановка задачи выглядит следующим образом. В портфель нужно выбрать такие заказы, чтобы целевая функция K имела максимальное значение

$$K = \sum_{i=1}^{N} K(z_i) \cdot x_i \to \max \qquad (6)$$

с учетом ограничений на имеющиеся объемы производственных мощностей [6], которые можно выразить следующим соотношением:

$$T^* = \sum_{i=1}^{N} t(z_i) \cdot x_i; \; T^* \le T, \qquad (7)$$

где z_i – i-й заказ из текущего портфеля заказов, для которого определена толщина листов

стекла, их суммарная площадь, $x_i \in \{0, 1\}$ показывает, включен ли i-й заказ в портфель, $K(z_i)'$ – значение целевой функции, рассчитанной для i-го заказа, $t(z_i)$ – трудоемкость выполнения i-го заказа на технологическом оборудовании, T^* – трудоемкость выполнения рассматриваемого портфеля заказов, T – календарный фонд времени на плановый период.

В настоящей работе предлагается следующий алгоритм решения задачи производственного планирования.

1. В период, предшествующий плановому, предприятие получает заявку – совокупность заказов, которую можно представить как вектор $Z = \{z_1, z_2, \ldots, z_m\}$, где $z_i = \langle\{w_{ij}, s_{ij} \mid i = 1..m, j = 1..k\}, G(z_i), p_i\rangle$ – кортеж, описывающий i-й заказ, k – количество позиций заказа, w_{ij} – толщина стекла в j-й позиции заказа, s_{ij} – площадь листов стекла в j-й позиции заказа, $G(z_i)$ – выручка от реализации заказа z_i, $p_i \in \in \{0...N'\}$ – внешне заданный приоритет выполнения заказа. Здесь N' – натуральное число; равенство $p = 0$ означает, что для заказа приоритет не определен.

2. Из общей заявки Z удаляются те заказы, для обеспечения которых продукция уже находится на складе. В заявку Z добавляются с соответствующими приоритетами те заказы, что не были выполнены в предыдущем плановом периоде.

3. Заказы с одинаковой толщиной ленты стекла консолидируются и представляются в общей заявке в виде единого заказа. В результате создается начальный портфель заказов $Z' = \{z'_1, \ldots, z'_x\}$, в котором заказы отличаются только толщиной стекла.

4. Заказы из вновь сформированного портфеля предварительно распределяются по двум линиям производства стекла таким образом, чтобы обеспечивалась равномерная загрузка линий.

5. На основе статистических данных о фактической производительности технологических линий рассчитывается трудоемкость T выполнения начального портфеля заказов Z'. При определении трудоемкости учитывается производительность технологических линий и время, необходимое для переналадок технологических линий на выпуск стекла другой толщины.

6. Календарный фонд времени на период планирования для всей производственной системы составляет 1 месяц, так как заявка на производство стекла поступает на предприятие один раз в месяц. Если трудоемкость выполнения текущего портфеля заказов оказывается больше календарного фонда времени на период планирования, из портфеля Z необходимо исключить некоторое множество заказов Z^* так, чтобы для каждой технологической линии выполнялись условия оптимальности (6) и ограничения (7). Данная задача сводится к задаче о рюкзаке. В настоящее время существует множество алгоритмов для ее решения [1].

7. Заказы Z^* переносятся на следующий плановый период, и их приоритет $p_i = p(z_i)$ увеличивается на единицу.

8. Если условие (7) показывает, что трудоемкость выполнения заказов на технологических линиях меньше календарного фонда времени, которым располагает предприятие, т.е.

$$T - \sum_{i=1}^{N} t(z_i) > T_\varepsilon, \qquad (8)$$

где $T\varepsilon$ – некоторое пороговое значение разницы между календарным фондом и трудоемкостью, определяемое экспертным путем, то портфель заказов Z дополняется "фиктивными" заказами Z^f, параметры которых рассчитываются на основе модели потока заказов [4].

9. Если условия (7) и (8) выполняются для обеих технологических линий, возникает необходимость минимизировать суммарное время для перехода с одной толщины ленты стекла на другую. Это достигается путем оптимального распределения заказов по технологическим линиям и формирования правильной очередности выполнения заказов на них. Поскольку явной зависимости времени перехода от характеристик заказов по статистическим данным не выявлено [3], для формирования очередности заказов следует воспользоваться рядом формальных правил, обусловленных особенностями флоат-процесса [2]:

а) на обеих технологических линиях переходы должны осуществляться по возможности на минимальную разницу в толщине;

б) работу по заказам текущего портфеля следует начинать с толщины, на которой заканчивается работа по предыдущему портфелю;

в) время между переходами не должно превышать период профилактического обслуживания технологических линий.

Что касается оптимального распределения заказов по технологическим линиям (п. 9а настоящего алгоритма), это отдельная опти-

мизационная задача балансировки загрузки производственных мощностей, для решения которой в настоящее время разработано достаточно много алгоритмов [1].

10. График проведения профилактических мероприятий рассчитывается на основе нормативов, определенных для технологических линий, и на основе графика переходов. Профилактические мероприятия должны выполняться только в моменты переходов, поскольку они, как и переналадки, нарушают ход технологического процесса и негативно влияют на качество стекла.

Предлагаемая в настоящей статье методика позволяет спланировать производственный процесс таким образом, чтобы максимизировать выручку от производства и реализации заказов и сократить производственные издержки. Представленная в статье методика проходит в настоящее время апробацию в ПО "Полированное стекло". Создание модуля СППР, реализующего описанные в настоящей статье алгоритмы, для APS-системы ПО "Полированное стекло", и внедрение его в производственный процесс планируется после подтверждения эффективности разработанной методики.

СПИСОК ЛИТЕРАТУРЫ

1. **Гэри М., Джонсон Д.** Вычислительные машины и труднорешаемые задачи. М. Мир, 1982 г. 416 с., ил.

2. **Макаров Р.И., Хорошева Е.Р., Лукашин С.А.** Автоматизация производства листового стекла. Учебное пособие. – М.: Изд-во АСВ, 2002 г., 192 с., ил.

3. **Макаров Р.И., Нагибин Г.С., Чебыкин С.В., Щукин М.В.** Использование статистических методов моделирования при разработке алгоритмов планирования выработки флоат-стекла. Владимир, 2006. – С. 264–268

4. **Макаров Р.И., Нагибин Г.С., Чебыкин С.В., Щукин М.В.** Моделирование потока заказов в производстве листового стекла. Ярославль, 2007. – С. 335–341

5. **Мертенс П.** Интегрированная обработка информации. Операционные системы в промышленности. М.: Финансы и статистика, 2007 г., 424 с.

6. **Наянзин Н.Г.** Формирование и оптимизация ресурсных потоков: графокомбинаторные методы и модели. Владимир: ВГПУ, 2006г., 190 с.

Лавров А.П., Молодяков С.А.

Возможности построения процессоров обработки сигналов в виде гибридных оптоэлектронных микросхем

Введение

Современная электронная технология обладает широкими возможностями и продолжает развиваться. Высокие тактовые частоты, компактность, малое энергопотребление, широкое использование автоматизированного проектирования - несомненные достоинства электроники. Однако для разработки каждого последующего поколения процессоров с новой технологией с пониженным топологическим размером требуются большие капиталовложения. Приходит время, когда электронные технологии приближаются к теоретическому пределу своих возможностей. Решение проблемы повышения производительности процессоров многим исследователям видится в привлечении оптоэлектронной технологии. Ее преимуществами является возможность параллельной обработки информации, простота реализации множества связей между оптическими элементами, возможность гибкой коммутации каналов связи. Не заменяя существующую электронную элементную базу, оптоэлектронные (ОЭ) процессоры и коммутаторы оптимально дополняют ее, резко ускоряя решение сложных задач, неудобных для традиционной электроники [1–3]. Специалисты все больше внимания уделяют исследованию и созданию новых архитектур мультипроцессорных (многоядерных) систем с разнородными процессорами. В рамках мультипроцессорных систем гибридные ОЭ процессоры особенно эффективно применять на этапе предварительной обработки информации в системах обработки потоков данных в мультимедийных задачах, задачах астрономии, связи и др. Производительность таких процессоров может достигать тысячи гигагерц [3].

Наиболее перспективными типами оптоэлектронных процессоров обработки сигналов являются процессоры, выполняющие операции вектор-матричного перемножения и корреляции/свертки. Схемы таких процессоров, построенных на объемных оптических элементах, известны давно, их обычно относят к классу систолических процессоров [2]. Современный уровень развития элементной базы делает возможной реализацию процессоров в виде гибридных микросхем и микромодулей. В гибридных микросхемах оптоэлектронные устройства ввода и вывода компонуются в единую жестко связанную структуру, которая может быть смонтирована на электронной плате подобно обычной электронной микросхеме. Настоящая работа посвящена анализу методов построения гибридных микросхем и исследованию их применения для обработки сигналов.

Базовая схема гибридной микросхемы

Рассматриваемый нами гибридный ОЭ процессор включает три слоя элементов: набор излучателей (линейка или матрица), транспарант и линейку или матрицу фотоприемника. Простейший элемент транспаранта пропускает или нет луч излучателя (светодиода) на соответствующий фотоприемник (фотодиод). Набор операций, которые может выполнять такая структура, зависит от возможностей каждого слоя и сигналов управления работой каждого элемента. Причем на каждом такте параллельно вводятся и обрабатываются несколько сигналов. Простейшей операцией в такой структуре является коммутация каналов: за счет управления элементами транспаранта происходит соединение или разрыв связи между светодиодом и фотодиодом. Подобного вида ОЭ коммутаторы известны, их широкое внедрение ожидается в будущем в сетевых архитектурах [3, 4]. Трехслойная структура позволяет проводить и более сложные операции.

На рис. 1а представлена схема вектор-матричного перемножителя на объемных элементах, которая рассматривается в работах [2–4]. Схема является основой для рассмотрения работы гибридной микросхемы. C_i – вектор

входного сигнала, i = 1,..., p; $M_{i,j}$ – матрица транспаранта, j = 1,..., l; R_j – выходной вектор. Для формирования вектора входного сигнала использовалась линейка светоизлучающих диодов (СИД), транспарант формировался на фотопленке, а в качестве фотоприемника использовалась линейка приборов с зарядовой связью (ФПЗС). Для равномерной засветки каждой строки транспаранта и формирования изображения на ФПЗС использовались элементы объемной оптики, которые не показаны на схеме. Каждый элемент СИД засвечивает свою строку транспаранта, в результате происходит перемножение: $F_{i,j} = C_i \cdot M_{i,j}$. Каждый столбец транспаранта проецируется на один элемент фотоприемника, в котором выполняется суммирование по столбцу матрицы $F_{i,j}$. Таким образом, выполняется известная операция вектор-матричного произведения:

$$R_j = \sum_i M_{i,j} C_i. \qquad (1)$$

Данная операция широко используется в задачах обработки информации, в частности как аппаратная реализация нейронных сетей [3]. Возможны различные способы реализации операции (1) [4]. Вместо линейки фотоприемника можно использовать матрицу ФПЗС (рис. 1б). В этом случае операцию суммирования (1) можно проводить либо в элементах ФПЗС (выходном регистре или выходном устройстве), либо в цифровом процессоре при постобработке. В качестве линейки СИД можно использовать линейку с вытянутой геометрией светодиодов. В результате при совпадающих размерах СИД, транспаранта и ФПЗС можно исключить промежуточную формирующую (фокусирующую) оптику и перечисленные элементы наложить друг на друга. Получается "сэндвич" – структура гибридной микросхемы [5, 6].

Одной из первых реализаций такого типа гибридных ОЭ микросхем явилась схема фирмы Mitsubishi (1995), которая имела слой из 32 полосчатых СИД, слой пространственно-временных модуляторов света и слой из 32 перпендикулярных светодиодам полосчатых фотодиодов. Слои непосредственно наложены друг на друга. Гибридная микросхема была использована для нейросетевых вычислений по модели Хопфилда [4]. Нами совместно с МИФИ в это же время начали проводиться исследования с макетированием гибридных ОЭ микросхем [6, 7].

Схема, представленная на рис. 1б, позволяет проводить операцию свертки входного сигнала с функцией, записанной на транспаранте:

$$R_{l,k} = \sum_j M_{i,j} C_{j+k}\ . \qquad (2)$$

Рассмотрим выполнение операции свертки. Она построена на алгоритме конвейерного накопления перемноженных в оптике дискретных сигналов. Процесс начинается в тот момент, когда элементы вектора C_i в зависимости от входного сигнала получают свои значения, которые высвечиваются на линейке СИД. Каждый элемент вектора C_i перемножается с каждым элементом строки транспаранта. Полученные произведения сохраняются в ФПЗС-матрице. Затем строки зарядового рельефа сдвигаются на один элемент в направлении выходного регистра. По окончании такта предыдущая

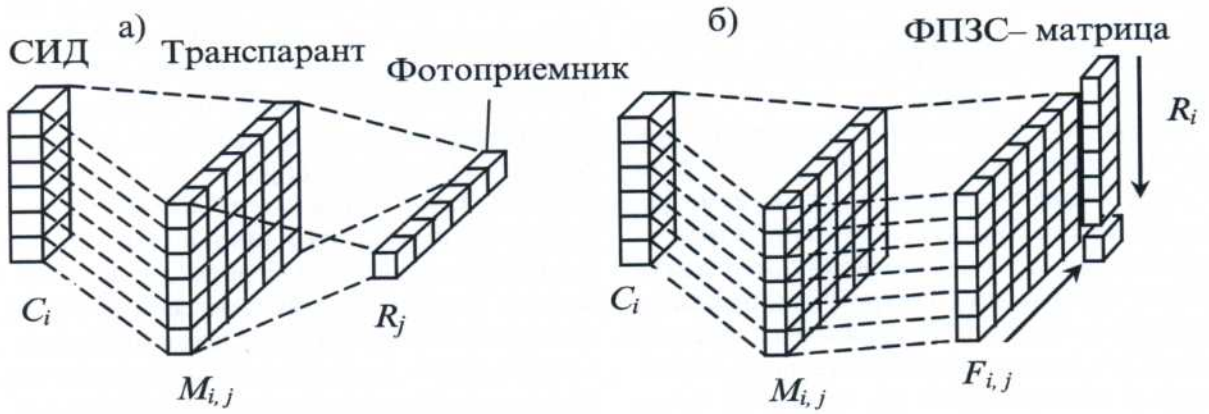


Рис. 1. Базовые схемы ОЭ процессора с линейным (а) и матричным (б) фотоприемниками

выборка сигнала сдвигается по линейке излучателей на одну позицию, и к ней добавляется новый отсчет, вновь происходит перемножение с накоплением зарядов в ФПЗС. Когда в результате сдвигов первый столбец матрицы ФПЗС окажется на месте последнего, то после следующего сдвига его значения перемещаются в выходной регистр, а оттуда через выходное устройство передается для дальнейших преобразований. Если функция пропускания транспаранта имеет вид $(1+\sin(ij/N))$, то ОЭ процессор выполняет спектральный анализ входного сигнала и каждая строка отвечает за свою частоту. Эквивалентную производительность такого ОЭ процессора, имеющего 10^6 пикселей в маске и выполняющего сдвиги зарядов в ФПЗС с частотой 1 МГц, можно оценить величиной 10^{12} операций "умножение + сложение"/с. Можно выполнять ряд других матричных операций и использовать другие схемы введения сигнала в ОЭ процессор (водить сигнал с использованием транспаранта) [5].

Основные архитектуры ОЭ процессоров и элементная база

Как было отмечено, основными операциями, выполняемыми ОЭ процессором, построенным на основе одной гибридной микросхемы, являются матричные алгебраические операции и операции свертки. Основной параметр этих операций – это количество одновременно обрабатываемых отсчетов, которое определяет базу обрабатываемого сигнала. Отсюда повышенные требования к количеству элементов (пикселей) в компонентах гибридной микросхемы. При необходимости увеличить базу обрабатываемого сигнала можно использовать более сложные архитектуры или параллельно несколько микросхем. Возможности параллельной обработки отдельных отсчетов в оптике реализуются просто, поэтому и новые архитектуры достаточно просты.

На рис. 2а и 2б показаны две схемы гибридных микросхем: типа "сэндвич" и развернутого типа. Первая схема предложена нами [6], проведено ее макетирование. Она характеризуется простотой и позволяет создавать жесткие конструкции, но требует сопряжения размеров всех слоев. Вторая схема, предложенная для создания межкристальных и межплатных коммутаторов [8], позволяет легко сопрягать размеры, но требует использования фокусирующих микролинз для формирования оптических пучков. Транспарант во втором случае может быть значительных размеров, в том числе и жидкокристаллический. Другое достоинство конструкции развернутого типа – легко сформировать матрицу светоизлучателей из отдельных модулей СИД (рис. 3а). На рис. 3б приведена предлагаемая нами схема параллельного включения двух гибридных микросхем (ГМ), которые выполняют и матричные операции, и операции свертки над двойным количеством пикселей. В этом случае две секции накопления ФПЗС работают как одна. В схеме применена возможность электрической передачи сигнала из ГМ1 в ГМ2 с использованием устройств вывода (вых.у.) и электрического ввода (вх. у.) ФПЗС [9].

Рассмотрим современные возможности элементной базы ОЭ микросхем.

1. Интегральные наборы излучателей могут быть построены на основе светодиодов или полупроводниковых лазеров. Современный уровень развития оптоэлектронных компонент позволяет изготовить линейку СИД со специальной сильно вытянутой вдоль одной координаты излучающей апертурой диодов, соответствующей по размерам столбцу ФПЗС. Мы использовали линейчатые светодиоды, состоящие из 64 СИД [6]. Известны линейки из 1200 светодиода на дюйм , которые используются для оптической головки в принтерах [10]. Они состоят из набора матриц СИД. Наиболее перспективными являются матрицы лазеров с вертикальным резонатором (Vertical-Cavity Surface-Emitting Laser – VCSEL) [8]. Однако при желании иметь 10^6 пикселей [4] максимальные размеры матриц составляют всего 128x128 лазеров (Sandia National Laboratories).

2. Перспективными транспарантами для создания управляемых пространственно-временных модуляторов света ОЭ микросхем являются жидкокристаллические модуляторы [1]. Технология ЖК модуляторов постоянно совершенствуется. Количество пикселей и их размер близки размерам фотоприемника, однако из-за рассеивания излучения на ЖК модуляторе создание простых ОЭ микросхем (рис.2.а) затруднено. Известны ОЭ матрицы Smart пикселов с оптическими входами/выходами [4]. Они используются для создания оптических коммутаторов, а для гибридных микросхем требуют дальнейшего совершенствования. В ряде случаев для упрощения ГМ можно отказаться от транспаранта (смотри ниже применение ГМ для пульсарной машины).

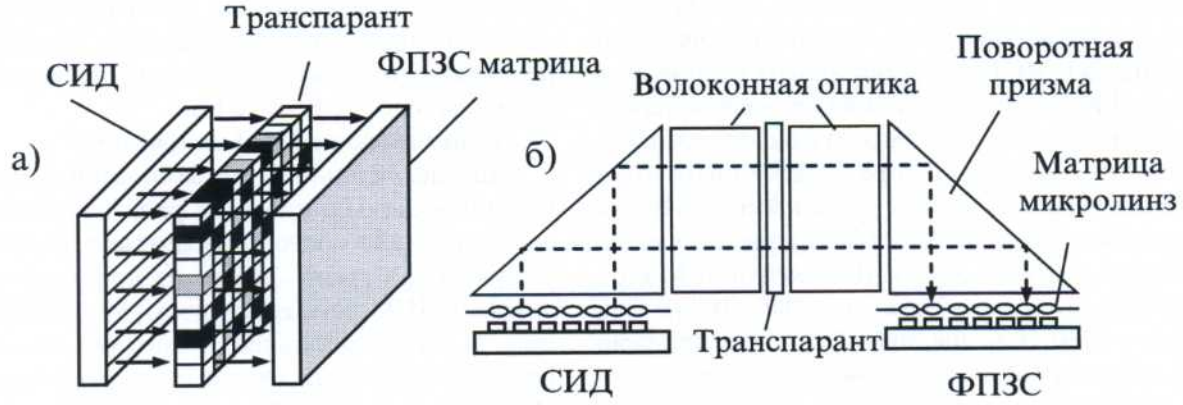


Рис. 2. Структурные схемы гибридных микросхем: а) типа "сэндвич"; б) развернутого типа

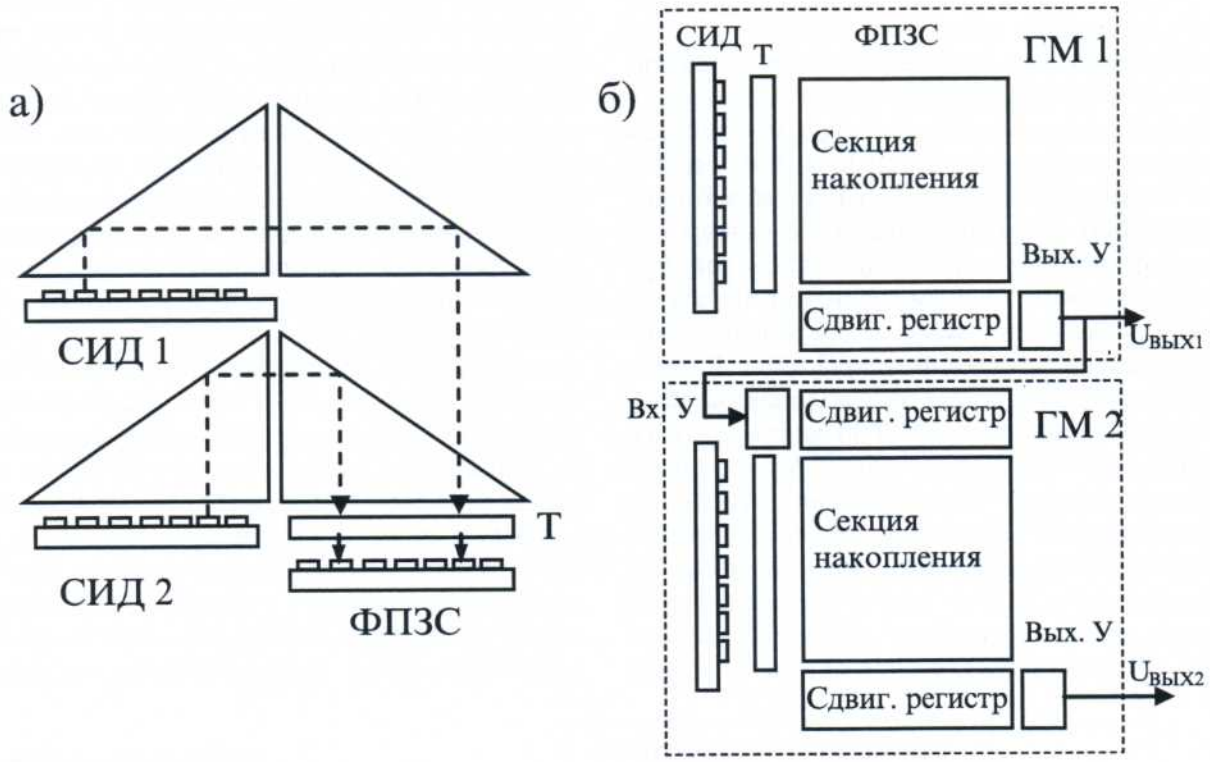


Рис. 3. Схемы гибридной микросхемы с двумя СИД (а) и параллельного включения двух гибридных микросхем (б)

3. Многоэлементные наборы фотоприемников являются наиболее сложным элементом. Перспективными технологическими типами являются ФПЗС и КМОП матрицы. Для организации операции свертки на самом фотоприемнике применимы только ФПЗС [9]. В этом случае ФПЗС находится в специальном режиме временной задержки и накопления ВЗН (Time Delay Integretion TDI). КМОП фотоприемники не обладают возможностями ПЗС по обработке зарядовых пакетов, но за счет интеграции на одном кристалле и фотоприемника, и цифровых элементов позволяют также проводит обработку данных, но уже в цифровом виде. Происходит интенсивное развитие всех узлов КМОП фотоприемников. Недостатки сенсорной части в большинстве случаев с успехом компенсируются возможностями по обработке сигналов в цифровых цепях фотоприемника. КМОП фотоприемник, кроме сенсорной части,

включает АЦП, буферную память, сумматора и др. элементы. Алгоритмы и скорость обработки данных в КМОП фотоприемниках отличаются от ФПЗС тем, что возможна параллельная обработка сигналов буквально каждого пикселя. Скорость обработки определяется местоположением АЦП. Примером расположения АЦП на каждом столбце является быстродействующая КМОП-матрица MT9M413 компании Micron Technology Inc (разрешение 1280 × 1024). Она имеет 1280 10 разрядных АЦП. Необходимо отметить новое направление фотоприемников – фотодиоды с управляемой чувствительностью, развиваемое на объединении "Позитрон". Транспарант в этом случае может быть совмещен с фотоприемником.

4. Дополнительными элементами гибридных микросхем являются световолоконные шайбы, микролинзы и электронное обрамление. Световолоконные шайбы – пластины, выполненные на основе волокон с малым – единицы микрон – диаметром. Такие пластины переносят изображение из плоскости входных торцов в плоскость выходных торцов, т.е. они выполняют функции отображающих каскадов, при необходимости – и с изменением масштаба [7]. Они обеспечивают жесткое соединение отдельных слоев гибридной микросхемы. В схеме нашего макета (рис. 2а) световолоконные шайбы находились между СИД и транспарантом и между транспарантом и ФПЗС. Микролинзы, обеспечивающие фокусирование отдельных световых пучков, являются элементами MEMS (Microelectromechanical Systems) технологии. Однако до сих пор их использование является сложной технологической задачей. Электронное обрамление для ОЭ микросхем в настоящее время строится на основе программируемой логики и сигнальных процессоров. Они обеспечивают требуемые тактовые частоты ввода и вывода данных 10-50 МГц.

Макет гибридной ОЭ микросхемы

Для исследования особенностей работы ОЭ микросхемы и проверки ряда технических решений были созданы несколько макетов. Структурная схема одного из них включает линейку СИД, транспарант и матричный ФПЗС (рис. 4). Такая гибридная ОЭ микросхема была предложена и исследована нами совместно с НИИ Радиооптики (Москва) и НИИ "Пульсар" (Москва) [6].

В микросхеме использовался многоэлементный светоизлучатель 1 с 64-мя ленточными диодами с размерами 31 × 7600 мкм, расположенных – с шагом 63 мкм и длиной волны излучения λ = 0,6 мкм. Маска 2 была рассчитана на компьютере с точностью 0,1% и изготовлена на фотопленке с высоким разрешением. Она представляет собой часть зонной пластинки Френеля. Волоконная шайба 3 была выполнена в виде плоскопараллельной пластины из волокон диаметром 6 мкм, расположенных упорядоченно в гексагональной упаковке при длине волокон (толщине шайбы) 2,5 мм. Матричный фотоприемник 4 типа ФПЗС-13М имел 264 столбца по 290 элементов. Шайба была установлена в качестве покровного стекла на кристалле фотоприемника. Шаг СИД и шаг трех столбцов ФПЗС были равны; пикселом считался сдвоенный элемент в строке с пропуском соседнего одиночного элемента. Пропуск элемента обеспечивал необходимую развязку между каналами. Контроллер обеспечивал работу ФПЗС как в обычном телевизионном режиме датчика изображения (при

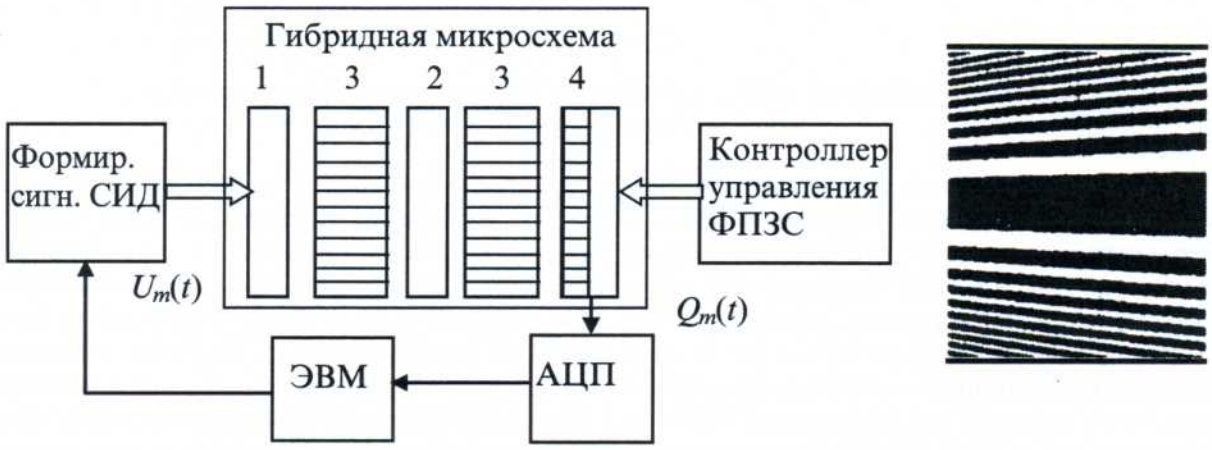


Рис. 4. Макет гибридной микросхемы и вид Френелевской маски

настройке и тестах), так и в рабочем режиме ВЗН. Совмещение режимов выполнено за счет применения памяти команд управления ФПЗС: один набор команд для режима кадрового, другой для режима ВЗН. Так режим ВЗН сводится к трем командам (шагам).

1. Перенос зарядовых пакетов из секции накопления ФПЗС матрицы в сдвиговый выходной регистр.

2. Вывод зарядов из регистра.

3. Переход на шаг 1 по очередному тактовому импульсу. Входные тестовые сигналы $U(t)$ генерировались в управляемом источнике сигналов на основе ЭВМ и подавались на линейку СИД через схему формирования сигналов. Выходной электрический сигнал обрабатывался в ЭВМ.

Нами проведено исследование линеек СИД (разработка НИИ "Сапфир", г. Москва) с использованием оптической схемы с переносом изображения в плоскость фотоприемника. Каждый из светодиодов имел независимое управление, при этом подводка токоведущих проводников к диодам была чередующейся: четные СИД запитывались с одного конца, а нечетные с противоположного. Сила света излучающего элемента при токе 30 мА составляет 380 мкКд; напряжение на элементе – ~2 В. При установке волоконной пластины ослабление излучения не превышало 5%. Распределение интенсивности в среднем достаточно равномерно, однако падение напряжения по длине диода вызывает небольшой наклон распределения, противоположный для четных и нечетных диодов. Проведенные исследования линейки СИД позволили определить требуемую для работы ФПЗС мощность излучения диодов и соответствующую ей силу питающего тока [6, 7].

В макете обрабатывались сигналы вида:

$$U_m(n)= \\ =\begin{cases} 0.5\left\{1+A\cos\left[\beta_m\left(n-n_0\right)^2\right]\right\}, & |n-n_0|\le N/2 \\ 0.5, & |n-n_0|>N/2 \end{cases}, \quad (3)$$

где $A < 1$ – глубина модуляции; n – номер отсчета; n_0 – номер отсчета, соответствующего середине сигнала; β_m – коэффициент, зависящий от номера m канала. На рис. 5.а показан тестовый входной сигнал длительностью $N = 290$ отсчетов с $m = 17$ ($\beta_m = 2.24 \cdot 10^{-3}$). На рис. 5.б показано изменение коэффициента пропускания маски по каналу, т.е. импульсный отклик процессора (в макете использовалась спадающая к краям засветка). На рис. 5.в представлен выходной сигнал (аппаратная функция) процессора для рассматриваемого канала. Анализ показывает, что измеренный отклик хорошо соответствует расчетному по форме, ширине и уровню боковых лепестков.

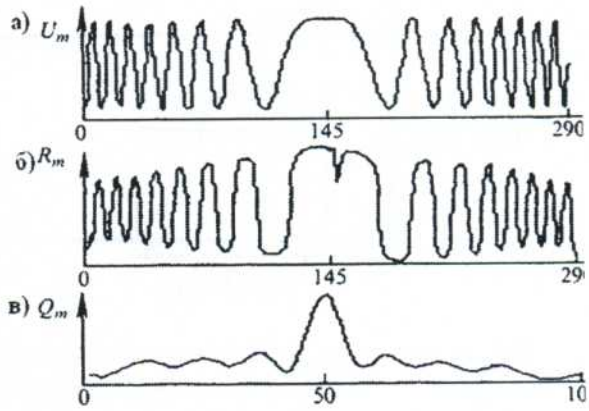


Рис. 5. Входной сигнал (а), сигнал маски (б) и выходной сигнал (в) одного канала ОЭ микросхемы

Задача управления линейкой СИД значительно упрощается при переходе от амплитудно-импульсной модуляции (АИМ) излучения светодиодов к широтно-импульсной (ШИМ). На рис. 6 показана временная диаграмма преобразования АИМ в ШИМ при подаче на гибридную микросхему импульсов сигнала, следующих с периодом T_n. В этом случае величина тока, текущего через СИД, фиксируется на одном уровне (бинарное управление), а длительность свечения определяется амплитудой управляющего сигнала. Появляется возможность перехода от аналоговых элементов схем введения сигнала в ОЭ микросхему к цифровым. Как следствие, устраняются температурные и временные нестабильности и нелинейности, свойственные аналоговым схемам; упрощаются схемы управления излучателями.

Применение гибридной технологии в фильтровом пульсарном процессоре

Описанная ОЭ микросхема (микросборка) (рис. 4) явилась основой при разработке макета обработки сигналов радиолокатора с синтезированной апертурой РСА, представленного ранее в работе [11]. Ниже мы представляем оригинальное применение гибридной технологии в радиоастрономическом приемном комплексе, предназначенном для обработки и регистрации сигналов пульсаров [12].

Пульсары представляют собой нейтронные звезды с высокой и очень стабильной скоро-

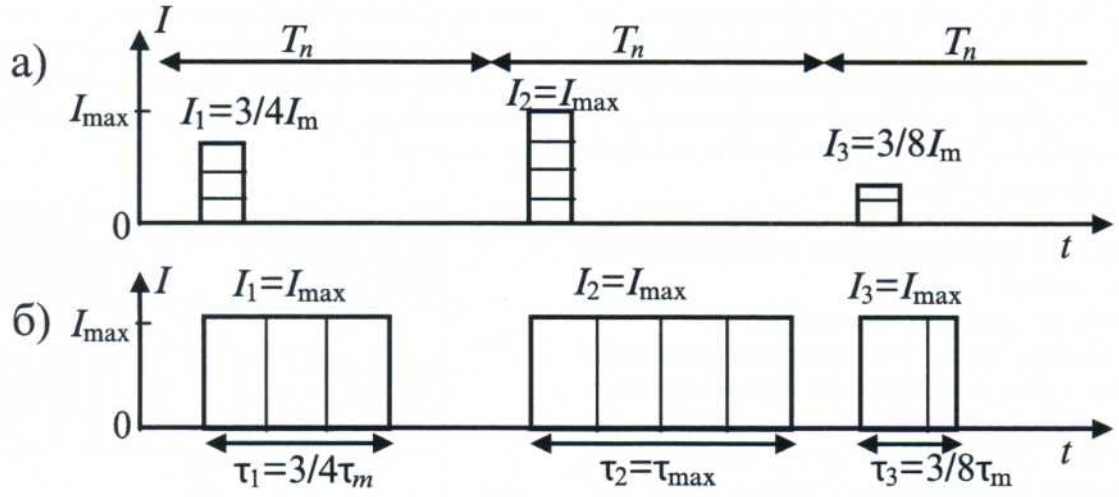


Рис. 6. Временная диаграмма управления светодиодами с преобразованием исходной АИМ (а) в ШИМ (б) с установкой всех импульсов на максимальном уровне

стью вращения. Импульсное радиоизлучение пульсаров имеет ряд особенностей, главная из которых – это дисперсия в межзвездной среде. Из-за дисперсии короткий импульс радиоизлучения пульсара дрейфует по частоте в полосе приема Δf с центральной частотой f от высоких частот к низким. Дрейф приводит к расширению импульса на выходе приемника и потере информации о его тонкой временной структуре. Пульсарный процессор приемного комплекса радиотелескопа должен учитывать, компенсировать дрейф.

Приемный комплекс (рис. 7) включает высокочастотный приемник, банк фильтров с N детекторами, гибридную ОЭ микросхему и ЦСП. ОЭ микросхема состоит из линейки СИД, волоконнооптической шайбы (ВШ) и линейного ФПЗС. Выходы детекторов управляют интенсивностью излучения СИД. Шайба, как уже отмечалось, позволяет жестко соединить СИД и ФПЗС.

Каждый радиоимпульс пульсара последовательно проходит от первого до последнего светодиода линейки СИД, в результате в фотоприемнике формируется двумерный массив отсчетов $S_{n,k}$, где n – номер канала, k – номер отсчета во времени. ФПЗС одновременно накапливает и перемещает зарядовые пакеты (работает в режиме ВЗН), устраняя тем самым влияние дисперсии и повышая отношение сигнал/шум (с/ш):

$$S_P(k)=\sum_{n=1}^{N} S_{n,k+T\cdot n}, \qquad (4)$$

где T – соответствующий сдвиг во времени, зависящий от меры дисперсии наблюдаемого пульсара.

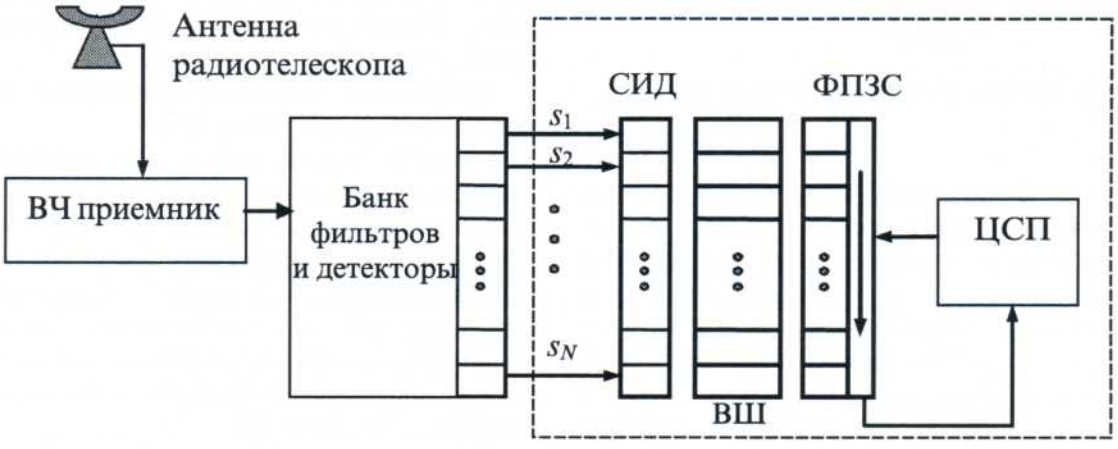


Рис. 7. Гибридная ОЭ микросхема для фильтровой пульсарной машины

При необходимости (для слабых пульсаров) дальнейшее увеличение отношения сигнал/шум происходит за счет суммирования отдельных импульсов пульсара в ЦСП. Таким образом, в фильтровом приемном комплексе гибридная микросхема выполняет несколько функций. Во-первых, происходит компенсация дрейфа импульса пульсара по частотным каналам приемника (компенсация дисперсии) в реальном времени. Во-вторых, за счет накопления в ФПЗС происходит повышение отношения сигнал/шум. В-третьих, обеспечена электрическая развязка аналоговой и цифровой частей комплекса. В-четвертых, реализован ОЭ коммутатор, подключающий таким необычным образом параллельные каналы банка фильтров на один выход фотоприемника. Нами проведено макетирование всех элементов представленной схемы. Ряд результатов описано в работах [7, 12].

Представленные нами реализации специализированных ОЭ процессоров в виде гибридных микросхем привлекательны прежде всего благодаря своей простоте и возможностям миниатюризации. Различные варианты архитектур микросхем и микромодулей могут применяться при решении широкого спектра задач параллельной обработки массивов данных. Дальнейшее развитие гибридной оптоэлектроники связано с совершенствованием элементной базы, прежде всего матриц микролазеров, а также с всесторонним освоением новой схемотехники, позволяющей проектировать и размещать на одной подложке и оптические и электронные компоненты.

Авторы выражают благодарность коллегам Центра оптоэлектронных проблем информатики СПбГПУ за совместную работу, Сотниковой Г.Ю., ст.н.с. ФТИ им. А.Ф.Иоффе, за участие в разработке и макетировании фотоприемника с волоконной шайбой, сотрудникам МИФИ Евтихиеву Н.Н. и Старикову Р.С. за совместную работу и обсуждение результатов.

Работа частично поддержана грантом РФФИ №07-02-01211.

СПИСОК ЛИТЕРАТУРЫ

1. **Bradley G. Boone**, Signal Processing Using Optics: Fundamentals, Devices, Architectures, and Applications. – Oxford University Press, 1997. – 416 p.

2. Оптическая вычислительная техника (тематический выпуск) //ТИИЭР. – 1984. – Т.72. – №7.

3. **McAulay A.D.** Optical computer architectures: the application of optical concepts to next generation computers. – J.Wiley & Sons, 1991. – 531 p.

4. Информационная оптика./ Н.Н.Евтихиев, О.А.Евтихиева, И.Н.Компанец и др.- М.Изд. МЭИ. – 2000. – 612с.

5. **Забулонов М. И., Иванов П. А., Евтихиев Н. Н. и др.** Разработка оптических вычислителей в виде гибридных микросхем и микромодулей: компьютерное моделирование и экспериментальное макетирование.// Накоемкие технологии. – 2005. – №5. – С. 20–28

6. **Евтихиев Н.Н., Есепкина Н.А., Лавров А.П. и др**. Оптоэлектронный процессор в виде гибридной микросхемы.// Квантовая электроника. – 1995. – Т. 22. – №10. – С. 985–990.

7. **Есепкина Н.А., Гаврилов Г.А., Лавров А.П. и др.** Оптоэлектронный процессор на основе матричного ФПЗС с волоконной шайбой.// Письма в ЖТФ. – 1992. – Т. 8. – №1. – С. 32–37.

8. **Chateauneuf, M., A.G. Kirk, D.V. Plant et al**. 512-channel vertical cavity surface emitting laser based free-space optical link.// Applied Optics. – 2002. – V. 41. – P. 5552–55561.

9. **Лавров А.П.** Оптоэлектронные процессоры радиосигналов с использованием сканирующих ПЗС-фотоприемников. Дис. д-ра ф.-м. наук, СПб: СПбГПУ, 1999. – 346 с.

10. **Masumi Koizumi, et.al.** High-Speed Chip-Matrix 1200 dpi LED Printhead.// Proceedings of SPIE. – 2001. – V. 4300. – P. 249–255.

11. **Dolgy V.A., Evtihiev N.N., Lavrov A.P. et al**. Optoelectronic hybrid microcheme for synthetic aperture radar signal processing.// Proc. SPIE. – 1993. – V. 2051. – P. 236–238.

12. **Есепкина Н.А., Лавров А.П., Молодяков С.А.** Акустооптический компенсатор дисперсии для наблюдения радиоизлучения пульсаров.// Антенны. – 2006. – №7. – С. 69–76.

Зайцев Э.Ф., Гузенко К.В., Молдаванова Е.Н.

Анализ волн в ферритодиэлектрическом волноводе вариационным методом

Ферритодиэлектрический волновод (ФДВ) является основой нового перспективного класса антенн – интегральных фазированных антенных решеток [1, 2], которые работают в миллиметровом диапазоне волн и обеспечивают простое и эффективное электрическое управление лучом. Для создания таких антенн необходимо решить волноводную электродинамическую задачу отыскания параметров распространяющихся мод в ФДВ: постоянных распространения, коэффициентов затухания, коэффициентов связи с излучателями на поверхности.

ФДВ представляет собой многомодовый невзаимный волновод со сложным поперечным сечением (рис.1), поэтому его анализ возможен только численными методами. В данной работе использован вариационный метод Ритца.

Поле распространяющейся волны имеет вид $E(x, y)e^{-i\gamma z}$, $H(x, y)e^{-i\gamma z}$, где z – продольная координата, γ – постоянная распространения, $E(x, y)$, $H(x, y)$ – векторные функции поперечных координат. Они удовлетворяют уравнениям:

$$i\gamma z_0 \times E = \mathrm{rot}E + ik\mu H, \qquad (1)$$

$$i\gamma z_0 \times H = \mathrm{rot}H + ik\mu E. \qquad (2)$$

Здесь $k = \omega/c = 2\pi/\lambda$ – волновое число, ε, μ – тензоры относительных проницаемостей, z_0 – единичный вектор. В уравнениях (1), (2) и далее под H понимается магнитный вектор, умноженный на константу $w = \sqrt{\mu_0/\varepsilon_0} = 120\pi$ Ом. Тогда E и H имеют одинаковую размерность, что делает формулы более удобными.

Умножим уравнение (1) на H^* (комплексно сопряженное), а уравнение (2) – на E^*, вычтем их и проинтегрируем по поперечному сечению:

$$i\gamma\int(E \times H^* + E^* \times H)z_0 dS = \iint[(\mathrm{rot}E + ik\mu H)H^* - (\mathrm{rot}H - ik\varepsilon E)E^*]dS.$$

Отсюда получаем:

$$\gamma = \frac{-i\int[H^*(\mathrm{rot}E + ik\mu H) - E^*(\mathrm{rot}H - ik\varepsilon E)]dS}{\int(E\times H^* + E^* \times H)z_0 dS}. \qquad (3)$$

Знаменатель этого выражения – вещественная величина. В числителе $(H^*\mu H)^* - H^*\mu^* H$ вещественно, если тензор μ эрмитов, т.е. нет потерь мощности (под μ^* понимается эрмитово сопряженный тензор). То же справедливо и для $E^*\varepsilon E$. Наконец, рассмотрим:

$$i\int(E^*\mathrm{rot}H - H^*\mathrm{rot}E)dS = i\int(H\mathrm{rot}E^* - H^*\mathrm{rot}E)dS + i\int\mathrm{div}(H \times E^*)dS.$$

Последнее слагаемое преобразуется в интеграл по внешнему контуру L поперечного сечения $i\oint_L(H\times E^*)n dl$ (n – нормаль к L), который обращается в нуль, если касательная составляющая E на внешней стенке волновода равна нулю:

$$E_\tau|_L = 0. \qquad (4)$$

Таким образом, выражение (3) вещественно, причем это верно и тогда, когда E и H не обязательно являются решениями уравнений (1), (2), лишь бы выполнялось условие (4).

Выражение (3) представляет собой функционал от E и H. Можно показать, что он принимает экстремальные значения, когда E, H являются точными решениями исходной системы (1), (2). Доказательство, ввиду его громоздкости, опускаем. Заметим только, что поиск экстремума функционала (3) эквивалентен отысканию экстремума числителя при условии постоянства знаменателя. Эту задачу на условный (относительный) экстремум удобно решать методом множителя Лагранжа, согласно которому необходимо найти экстремум функционала:

$$F = i\int(E^*\mathrm{rot}H - H^*\mathrm{rot}E)dS + k\int(H^*\mu H + E^*\varepsilon E)dS - \gamma\int(E \times H^* + E^* \times H)z_0 dS. \qquad (5)$$

Следуя методу Ритца, найдем приближенное решение для E и H в виде конечных разложений:

$$E_x = \sum_{n=1}^{N} a_n\phi_{xn}(x,y), \quad E_y = \sum_{n=1}^{N} b_n\phi_{yn}(x,y), \quad E_z = \sum_{n=1}^{N} c_n\phi_{zn}(x,y), \quad H_x = \sum_{n=1}^{N} d_n\psi_{xn}(x,y), \quad H_y = \sum_{n=1}^{N} e_n\psi_{yn}(x,y), \quad H_z = \sum_{n=1}^{N} f_n\psi_{zn}(x,y), \qquad (6)$$

где a_n, ..., f_n – неизвестные коэффициенты (комплексные), $\varphi_{xn}(x, y)$, ..., $\psi_{zn}(x, y)$ – некоторая система известных функций, именуемых базисными. Эти функции должны быть такими, чтобы выполнялось условие (4). Кроме того, они должны обладать необходимой полнотой в том смысле, что при $N \to \infty$ по ним может быть разложена любая функция с интегрируемым (по области S) квадратом модуля. В остальном базисные функции произвольны.

После подстановки (6) в (5) функционал F превращается в функцию многих неизвестных a_n, ..., f_n, имеющую вид билинейной формы этих величин. Для отыскания экстремума F надо приравнять к нулю частные производные по всем неизвестным. Последние являются комплексными, например $a_n = a'_n + ia''_n$, аналогично остальные. Следовательно, надо приравнять $\frac{\partial F}{\partial a'_n} = 0$ и $\frac{\partial F}{\partial a''_n} = 0$. Это равносильно одному комплексному равенству $\frac{\partial F}{\partial a'_n} + i\frac{\partial F}{\partial a''_n} = 0$, что в свою очередь равносильно равенству

$$\frac{\partial F}{\partial a^*_n} = 0. \qquad (7)$$

Приравняв нулю частные производные F по всем неизвестным (комплексно сопряженным), получим однородную линейную систему $6N$ уравнений относительно a_n, ..., f_n. Каждое нетривиальное решение этой системы дает нам постоянную распространения γ соответствующей собственной моды волновода и коэффициенты разложения полей по базисным функциям.

Расчетная модель ФДВ показана на рис. 2. Она отличается от реального неэкранированного ФДВ наличием вспомогательных электрических стенок по бокам и сверху феррито-диэлектрической структуры на некотором расстоянии от нее. Это необходимо для выполнения условия (4) на внешнем контуре. В воздушных зазорах поле собственной моды ФДВ быстро убывает, так что вспомогательные стенки на некотором удалении от структуры уже перестают влиять на собственные моды.

В качестве базисных функций были выбраны произведения тригонометрических функций вида

$$\begin{Bmatrix} \cos\frac{n\pi x}{a} \\ \sin\frac{n\pi x}{a} \end{Bmatrix} \cdot \begin{Bmatrix} \cos\frac{2m\pi y}{p} \\ \sin\frac{2m\pi y}{p} \end{Bmatrix}, \qquad (8)$$

где a – высота волновода, p – его ширина. Эти функции такие же, как у полей в пустом волноводе, они заведомо удовлетворяют условию (4) и обладают необходимой полнотой для представления только симметричных мод, у которых составляющие E_y, H_x и H_z являются четными функциями y. Несимметричные моды (с нечетной зависимостью E_y, H_x и H_z) не рассматриваются, так как они не возбуждаются на входе реального ФДВ и не взаимодействуют с излучателями.

Каждая базисная функция (8) имеет два целочисленных индекса m и n, поэтому коэффициенты в (6) будут тоже двухиндексными: a_{mn}, ..., f_{mn}. Всего будет $6MN$ неизвестных и столько же уравнений.

Для реализации метода Ритца в среде MATLAB целесообразно переобозначить неизвестные так, чтобы они имели только один индекс, пробегающий $6MN$ значений. Тогда система уравнений примет вид:

$$Ac = \gamma Dc, \qquad (9)$$

где c – столбец неизвестных, A, D – квадратные матрицы.

Расчетная программа вычисляет все элементы матриц A, D, а затем использует стандартную процедуру вычисления собственных значений γ_j и собственных векторов c_j. Тем самым задача полностью решена.

В рассматриваемой конструкции ФДВ феррит работает при малом намагничивающем поле, поэтому его свойства на СВЧ достаточно хорошо описываются соотношением

$$\mu H = \mu_1 H - i\mu_a H \times m_0, \qquad (10)$$

где m_0 – единичный вектор в направлении постоянной намагниченности, $\mu_1 = 0{,}9 \div 0{,}95$ – постоянная, а параметр μ_a зависит от намагниченности и может принимать значения от – 0,4 до + 0,4. Формула (10) инвариантна к выбору направления осей координат.

При проведении исследований выполнены расчеты двух конструкций ФДВ. Первая – открытый ФДВ (рис.1), который применяется в сканирующих антеннах с дипольными излучателями. Для каждой распространяющейся моды рассчитывались коэффициент замедления $q = \gamma/k$, постоянная затухания α, учитывающая потери мощности в диэлектрике, феррите и металлическом экране, и сопротивление связи моды z_d с излучающим диполем [3].

Установлено, что оптимальная величина боковых зазоров между ФДВ и вспомогательными электрическими стенками (см. рис.2)

составляет около 1 мм, а верхний зазор – 3 мм. Необходимая точность расчетов достигается при числе базисных функций $MN = 100$.

На рис. 3 приведены графики зависимости q от μ_a для различных мод на частоте 36 ГГц при следующих размерах: толщина феррита – 0,7 мм, толщина диэлектрика – 0,3 мм, ширина феррита – 7,2 мм, ширина диэлектрика – 3,4 мм. Видно, что самая низшая (рабочая) мода хорошо управляется за счет изменения намагниченности, в то время как некоторые моды вовсе не управляются. Это связано с различием структуры полей разных мод. По рис. 3 следует также отметить, что при некоторых значениях намагниченности возможно вырождение мод (пересечение линий на графике). Кроме того, важно помнить, что в открытом ФДВ не могут существовать моды с замедлением $q < 1$. Их наличие в расчетных результатах вызвано наличием вспомогательных электрических стенок в расчетной модели реального открытого волновода. Фактически такие моды должны быть отброшены.

В целом полученные результаты хорошо согласуются с расчетами по методу Галеркина и экспериментальными данными [4].

Вторая конструкция ФДВ, которой уделено основное внимание, – полностью экранированный ФДВ, предназначенный для сканирующих антенн со щелевыми излучателями. Такие антенны имеют ряд преимуществ перед дипольными, но пока почти не исследованы. Этот волновод отличается от изображенного на рис. 2 только тем, что верхний проводящий экран вплотную примыкает к верхнему слою феррита. В отличие от предыдущего случая здесь могут распространяться и ускоренные моды, имеющие $q < 1$.

На рис. 4 построены графики зависимости q от μ_a для разных мод при тех же размерах, как для рис. 3, на той же частоте. По сравнению с открытым ФДВ коэффициент замедления основной моды немного уменьшается вследствие влияния верхнего экрана. Для закрытого ФДВ также имеет место уже отмеченное для рис.3 явление вырождения мод. Кроме того, следует подчеркнуть, что наличие ускоренных мод в данном случае имеет смысл, но, поскольку они являются самыми высшими, расчет их параметров производится с существенной погрешностью вследствие ограниченного количества базисных функций. В связи с этим три самые нижние кривые на рис. 4 лишь примерно отражают ход соответствующих реальных зависимостей. Для их уточнения следует увеличить число базисных функций.

В табл. 1 приведены результаты расчета параметров мод при разных поперечных размерах, когда феррит размагничен ($\mu_a = 0$). При указанных изменениях размеров число распространяющихся мод не меняется, постоянная затухания α немного уменьшается с увеличением толщины. Важным параметром является y_s – проводимость связи моды с излучающей щелью [5], которая позволяет определить долю мощности волны, излучаемую щелью в пространство, и долю мощности, расходуемую на возбуждение всех волноводных мод. При увеличении толщины ферритовых слоев величины y_s основной и большинства других мод заметно уменьшаются, что дает возможность обеспечить оптимальное соотношение между проводимостями за счет выбора размеров ФДВ.

Таблица 1

Влияние толщины феррита на параметры мод ФДВ

Толщина феррита	Номер моды	1	2	3	4	5	6	7
0,7 мм	q	3,76	3,40	2,67	2,10	1,91	1,27	0,83
	α, дБ/см	0,44	0,26	0,39	0,35	0,63	0,73	1,06
	$\lvert y_s \rvert$, $\sqrt{\text{См}}$	0,0259	0,0141	0,0086	0,0312	0,0525	0,0110	0,0862
0,85 мм	q	3,87	3,38	2,83	2,36	2,25	1,88	1,58
	α, дБ/см	0,41	0,26	0,35	0,42	0,35	0,49	0,58
	$\lvert y_s \rvert$, $\sqrt{\text{См}}$	0,0177	0,0122	0,0081	0,0303	0,0233	0,0139	0,0504
1 мм	q	3,92	3,37	2,94	2,62	2,38	2,18	1,91
	α, дБ/см	0,39	0,25	0,32	0,35	0,34	0,43	0,47
	$\lvert y_s \rvert$, $\sqrt{\text{См}}$	0,0126	0,0106	0,0082	0,0200	0,0175	0,0103	0,0397

Таким образом, в работе решена электродинамическая задача отыскания параметров распространяющихся мод в электрически управляемом ФДВ методом Ритца. Получено выражение для функционала, принимающего экстремальные значения на решениях задачи, выбрана система базисных функций для представления решения, реализован алгоритм Ритца в среде MATLAB. Выполнены расчеты для различных конструкций ФДВ. Результаты будут способствовать улучшению параметров электрически сканирующих антенн миллиметрового диапазона волн.

СПИСОК ЛИТЕРАТУРЫ

1. **Zaitsev E. F. et al.** MM-wave Integrated Phased Arrays with Ferrite Control. // IEEE Transaction on Antennas and Propagation, vol. 42, No. 3, pp. 1262–1368, March, 1994.

2. **Зайцев Э.Ф., Гуськов А.Б., Черепанов А.С..** Новые электрически сканирующие антенны миллиметрового диапазона волн. // Известия высших учебных заведений России. Радиоэлектроника. №4, 2003, стр. 3–12.

3. **Zaitsev E.F., Guskov A.B., Cherepanov A.S.** Mathematical Model of Ferrite Integrated Phased Array. // Eighth Biennial Conference on Electromagnetic Field Computation, Tucson, Arizona, June 1–3, 1998.

4. **Гузенко К.В., Зайцев Э.Ф.** Волны в многослойном ферритодиэлектрическом поперечно намагниченном волноводе с электрически управляемым коэффициентом замедления // Журнал Технической Физики, т. 73, № 9, 2003. С. 95–100.

5. **Зайцев Э.Ф., Явон Ю.П.** Параметры излучающей щели, прорезанной в стенке невзаимного волновода. // Радиотехника и электроника, 1983, №4.

Макатов М. Р., Вердиев М. М.

Информационная модель описания объектов и ситуаций проблемной среды

При решении задач планирования поведения интеллектуальных систем актуальной проблемой является описание образов и ситуаций проблемной среды (ПС). Рассмотрим модель ПС, в которой функционирует специализированный интеллектуальный робот (ИР), имеющий развитую информационно-измерительную систему, способный выполнять множество действий $B = \{b_i\}\ i = 1, n_1$ (например, сложить числа, взять предмет, повернуть влево, перейти в заданную точку и т.д.). Важную и основополагающую роль в организации планирования интеллектуальных систем (ИС) играет структура представления знаний о закономерностях внешнего мира.

Рассмотрим теоретико-множественное описание объектов проблемной среды. Для систематизированного представления предметных знаний все объекты проблемной среды по семантическому признаку можно разделить на пять основных списков (рис.1):

$O = \{o_{i_1}\}$, i_1=1, n_1 – множество объектов;

$P = \{p_{i_5}\}$, i_5=1, n_5 – множество составных частей объектов;

$\psi = \{\psi_{i_6}\}$, i_6=1, n_6 – множество свойств или возможностей объектов;

$L = \{l_{i_7}\}$, i_7 = 1, n_7 – множество признаков объектов и их составных частей;

$H = \{h_{i_8}\}$, $i_8 = 1, n_8$ – множество имен объектов и их составных частей.

Определение 1.1. Предметами называются объекты и понятия ПС, имеющие законченный, самостоятельный семантический смысл и содержание (например, птица, стол, дерево решений).

Определение 1.2. Частями объектов называются образы ПС, которые являются составными элементами, определяющими структуру объекта (например, крыло птицы).

В самом общем случае одно и то же понятие может определять как категорию объекта, так и категорию части объекта, например, понятие "ручка" определяет категорию объекта, если речь идет о пишущей ручке, а не о ручке чемодана и т.д. Иными словами, просматривается многозначность в толковании понятий. Обойти указанную трудность можно, если при представлении объектных знаний ИС многозначные понятия, фактически определяющие составные части сложных объектов, согласно их смысловому содержанию представлять в виде составных понятий (например, дверная ручка).

Определение 1.3. Под свойствами и умениями объектов и их составных частей следует понимать соответственно их функциональное назначение и действия, которые они оказывают друг на друга.

Функциональное назначение объектов определяется типом инструмента, в качестве которого они могут быть использованы в процессе поведения, и тем, каким образом осуществляется применение этого инструмента.

Определение 1.4. Признаками частей объектов, а следовательно, и признаками самих объектов называются образы ПС (характеристики), которые описывают их форму, химический состав, цвет, геометрические размеры и т.п.

Определение 1.5. Именами и названиями объектов и их составных частей называются абстрактные биективно соответствующие им понятия, выраженные на формальном языке ИС (возможно естественном).

Задаваясь на элементах множества объектов О родо-видовыми отношениями, его можно представить как объединение трех подмножеств:

$$O = O_1 \cup O_2 \cup O_3,$$

где $O_1 = \{o_{i_9}\}$, $i_9 = 1, n_9$ – множество объектов, определяемых понятием "вид" (например, токарный станок № 3);

$O_2 = \{o_{i_{10}}\}$, $i_{10} = 1, n_{10}$ – множество объектов, определяемых понятием "род" (например, птицы);

$O_3 = \{o_{i_{10}}\}$, $i_{11} = 1, n_{11}$ – множество объектов, определяемых понятием "класс" (например, животные).

При этом понятие "вид" определяется рядом конкретных индивидуумов, рассматриваемых как идентичные объекты, или объекты, различающиеся между собой по несущественным характеристикам.

Согласно определениям 1.1–1.5 каждый объект $o_{i_1} \in O$ описывается при помощи соответствующего множества характеристик X_{i_1}, состоящего из трех подмножеств P_{i_1}, ψ_{i_1} и L_{i_1}, т. е. $X_{i_1} = P_{i_1} \cup \psi_{i_1} \cup L_{i_1}$.

Следовательно, объект o_{i1} по характерным только для него элементам этих подмножеств определяет понятие одного вида.

Некоторое множество видов по характерной только для него совокупности описывающих его элементы подмножеств P_{i_1}, ψ_{i_1} и L_{i_1} объединяется в список и определяет понятие одного рода. Некоторое множество родов по характерному только для него пересечению описывающих его элементы подмножеств P_{i_1}, ψ_{i_1}, L_{i_1} объединяется в список и определяет понятие одного класса.

Множество составных частей объектов представляет собой объединение следующих трех подмножеств:

$P_1 = \{p_{i_{12}}\}$, $i_{12} = 1, n_{12}$, $P_1 \subset P$ – множество обобщенных частей объектов или частей объектов, состоящих в свою очередь из нескольких сложных элементов;

$P_2 = \{p_{i_{13}}\}$, $i_{13} = 1, n_{13}$, $P_2 \subset P$ – множество частей обобщенных частей объектов;

$P_3 = \{p_{i_{14}}\}$, $i_{14} = 1, n_{14}$, $P_3 \subset P$ – множество неделимых частей объектов.

Свойства объектов можно разделить также на три подмножества:

$\psi_1 = \{\psi_{i_{15}}\}$, $i_{15} = 1, n_{15}$, $\psi_1 \subset \psi$ – прямые свойства, определяющие умение объектов и непосредственные формы их взаимодействия без активного участия ИС;

$\psi_2 = \{\psi_{j_2}\}, j_2 = 1, m_2, \psi_2 \subset \psi$ – вспомогательные свойства, характеризующие предварительные формы взаимодействия объектов (например, сближение одного объекта с другим и т.п.).

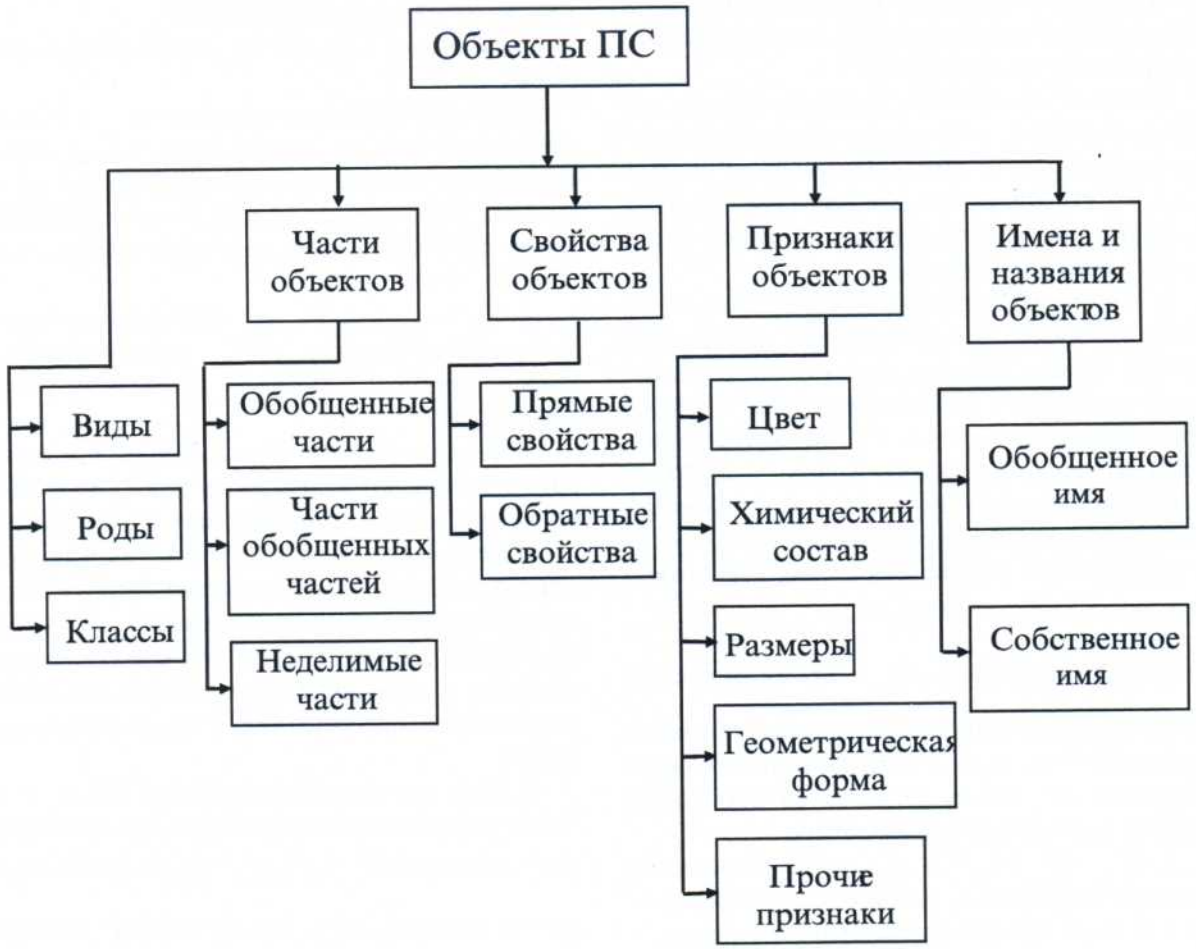


Рис. 1. Разделение объектов проблемной среды по семантическим признакам

$\psi_3 = \{\psi_{j_3}\}, j_3 = 1, m_3, \psi_3 \subset \psi$ – инструментальные свойства, определяющие, в качестве какого инструмента могут быть использованы объекты и каким образом их следует использовать.

Множество признаков объекта L разбивается на следующие непересекающиеся подмножества:

$L_1 = \{l_{i_{16}}\}$, $i_{16} = 1, n_{16}$, $L_1 \subset L$ – множество признаков, определяющих различные цвета объектов и их частей;

$L_2 = \{l_{i_{17}}\}$, $i_{17} = 1, n_{17}$, $L_2 \subset L$ – различные геометрические формы объектов и их частей;

$L_3 = \{l_{i_{18}}\}$, $i_{18} = 1, n_{18}$, $L_3 \subset L$ – признаки, задающие размеры различных объектов;

$L_4 = \{l_{i_{19}}\}$, $i_{19} = 1, n_{19}$, $L_4 \subset L$ – элементы химического состава объектов;

$L_5 = \{l_{i_{19}}\}$, $i_{20} = 1$, n_{20} – прочие признаки объектов.

Каждому объекту $o_{i_1} \in O$ можно присвоить абстрактное имя или понятие на языке ИР. Следует различать обобщенные и собственные имена. Обобщенные имена присваиваются некоторому подмножеству объектов (объектов, относящихся к одному виду), а собственные к конкретному индивидууму.

Помимо сказанного, каждый объект проблемной среды $o_{i_1} \in O$ можно охарактеризовать множеством допустимых состояний $F = \{f_{i_{21}}\}$, $i_{21} = 1, n_{21}$, которые он может принимать и изменять в результате оказываемых на него внешних и внутренних воздействий.

К одному из эффективных способов структурированного представления объектных знаний можно отнести сетевые модели, которые позволяют достаточно полно в общем виде описывать объекты ПС и осуществлять ассоциативные принципы поиска и выборки необходимых данных в процессе принятия решений. В этом случае каждый объект $o_{i_1} \in O$ описывается соответствующей ему семантической макросетью $M_{i_1} = \{m^{i_1}_{i_{22}}\}$, $i_{22} = 1, n^{i_1}_{i_{22}}$, где каждая микросеть $m^{i_1}_{i_{22}} \in M_{i_1}$ представляет собой взаимосвязанные между собой элементы множеств O, P, ψ, L, F и H. При этом каждая макросеть M_{i_1} может состоять из следующих четырех типов автономных сетей.

1. $E^*_1 = \{e_{i_{23}}\}$, $i_{23} = 1, n_{23}$ – сеть возможных состояний объектов, для которых микросеть $e_{i_{23}} \in E^*_1$ определяется высказыванием вида $e_{i_{28}} = f_{i_{21}}\, r_{i_2}\, f'_{i_{21}}$ или высказыванием $e'_{i_{29}} = o_{i_1}\, r'_{i_2}\, f_{i_{21}}$, где роль отношения r_{i_2} выполняет элемент из множества свойств ψ, а отношение r'_{i_2} задается высказыванием "быть состоянием объекта".

Рассмотрим случай, когда в роли отношения $r_{i_2} \in R$ выступает действие ИС $b_{i_2} \in B$, преобразующее состояние $f_{i_2} \in F$ в состояние $f'_{i_{21}} \in F$. При этом высказывание $e_{i_{23}} = f_{i_{23}}\, r_{i_2}\, f'_{i_{21}}$ заменяется импликативным решающим правилом $f_{i_{21}}$ & & $\{b^1_{j_1}, b^2_{j_1}, ..., b^k_{j_1}\} \to f'_{i_{21}}$. Приведенная запись означает, что если объект находится в состоянии $f_{i_{21}}$, то действия $\{b^1_{j_2}, b^2_{j_2}, ..., b^k_{j_2}\}$ при их выполнении в заданной последовательности переводят объект в состояние $f'_{i_{21}}$. Выполняя операцию композиции над элементами множества $E^*_1 \subset M_{i_1}$, можно сформировать цепочку действий следующего вида $f^0_{i_{21}}$ & $b^1_{j_1} \to f^1_{i_{21}}$ & $b^2_{j_1} \to$ $... \to f^*_{i_{21}}$ позволяющую перевести объект из состояния $f^0_{i_{21}}$, в требуемое состояние $f^*_{i_{21}}$.

2. $E^*_2 = \{e_{i_{24}}\}$, $i_{24} = 1, n_{24}$ – сеть составных частей объекта. В ней каждая микросеть $e_{i_{24}} \in$ $\in E^*_2$ определяется высказыванием следующего содержания:

$p_{i_{13}}\, r_{i_1}\, p_{i_{14}}$, $o_{i_1}\, r_{i_2}\, p_{i_{12}}$, $p_{i_{12}}\, r_{i_2}\, p_{i_{13}}$, где r_{i_2} – отношение "быть составной частью".

3. $E^*_3 = \{e_{i_{25}}\}$, $i_{25} = 1, n_{25}$ – сеть свойств объекта.

В этой сети каждая микросеть $e_{i_{25}} \in E^*_3$ есть тройка вида $o_{i_1}\, r_{i_2}\, \psi_{i_{12}}$, а роль отношения r_{i_2} могут выполнять высказывания следующего содержания: "обладать способностью", "обладать умением" и "выступать в качестве инструмента для ".

4. $E^*_4 = \{e_{i_{26}}\}$, $i_{26} = 1, n_{26}$ – сеть описания признаков объекта, где $e_{i_{26}} \in E^*_4$ – высказывание l_{i_7} "иметь признак", например "иметь красный цвет" и т.д.

В неопределенных условиях функционирования зачастую невозможно дать подробное четкое описание произвольных объектов ПС, а также количественно оценить параметры, определяющих их характеристик. Обойти отмеченные трудности можно путем представления множества характеристик ПС субъективным образом в виде нечеткого множества и качественного их описания в содержательной форме.

В этом случае каждый объект ПС $o_{i_1} \in O$ будет определяться нечетким множеством характеристик $X_{i_1} = \{<\mu^{i_1}_{j_4}, x^{i_1}_{j_4}>\}$, $j_4 = 1, m_4$ и обозначаться $o_{i_1}(X_{i_1})$, где $\mu^{i_1}_{j_4} \in [0,1]$ – степень присущности характеристики $x^{i_1}_{j_4}$ к нечеткому

множеству X_{i_1} или к соответствующему этому множеству объекту. В случае, когда для всех $x_{j_4}^{i_1} \in X_{i_1}$ выполняется условие "$\mu_{j_4}^{i_1} = 1$", т.е. все степени принадлежности равны единице, имеем четкое множество характеристик описания объекта.

В свою очередь каждая характеристика объекта ПС может быть задана тройкой $< x_{j_4}^{i_1}, r_{i_{29}}^{j_4}, R_{i_{29}}^{j_4, j_5} >$, где $r_{i_{29}}^{j_4}$ – количественное значение параметров характеристики $x_{j_4}^{i_1}$; $R_{i_{29}}^{j_4, j_5}$ – качественное значение параметров характеристики, определяемое термом лингвистической $R_{i_{29}}$ переменной $R_{i_{29}} \in$ R, описывающей эту характеристику (на практике обычно $j_5 = 1, 5$). $R = \{R_{i_{29}}\}$, $i_{29} = 1, n_{29}$ – множество известных лингвистических переменных, биективно соответствующих семантическому определению известных ИР понятий.

В общем случае каждая лингвистическая переменная может быть представлена набором [1] $R_{i_{29}} = (F_{i_{29}}, T_{i_{29}}, U_{i_{29}}, Z_{i_{29}}, N_{i_{29}})$, где $F_{i_{29}}$ – наименование лингвистической переменной, например "расстояние между объектами ПС"; $T_{i_{29}} = R_{i_{29}}^{j_5}$ – терм-множество или множество всех названий лингвистических значений переменной $R_{i_{29}}$, определяющихся нечеткой переменной со значениями из универсального множества; $U_{i_{29}}$ – универсальное множество значений с базовой переменной $r_{i_{29}}$; $Z_{i_{29}}$ – синтаксические правила, порождающие название термов лингвистической переменной; $N_{i_{29}}$ – семантические правила, которые определяют смысловое содержание $N_{i_{29}}(R_{i_{29}})$ нечеткой переменной, относящейся к $R_{i_{29}}$.

При этом следует иметь в виду, что множество лингвистических переменных R можно разбить на два подмножества $R_1, R_2 \subset R$, таких, что в R_1 входят только те переменные, которые могут быть оценены как качественно, так и количественно, а в множество R_2 входят только те переменные, которые могут быть оценены лишь качественно, например, лингвистическая переменная "высота" может быть оценена как качественно, так и количественно, а переменная "красота" может быть оценена только качественно. Таким образом, сравнение качественно оцениваемых переменных может быть выполнено только на содержательном нечетком уровне описания. Следовательно, нечетко заданная модель представления объектов ПС является более адекватной и более удобной для распознавания и сравнения образов ПС.

Резюмируя вышеизложенное, следует отметить, что предложенная модель позволяет систематизировать и упорядочить предметные данные в памяти ИР и на этой основе выбирать объекты ПС, обладающие необходимыми характеристиками, например, для их использования в качестве инструмента в процессе целенаправленной деятельности.

СПИСОК ЛИТЕРАТУРЫ

1. **Заде Л.** Понятие лингвистической переменной и его применение к принятию приближенных решений. – М.: Мир 1976 – 163 с.

Исмаилов Ш.А., Магомедов Ш.Г.

Алгоритмы и структуры преобразования числовых данных из позиционной системы счисления в систему остаточных классов

Для повышения быстродействия выполнения арифметических операций могут быть использованы специализированные процессоры с представлением данных в системе остаточных классов. Это позволяет повысить скорость выполнения, например, операций умножения на 1–2 порядка в сравнении с выполнением операций при представлении данных в классической системе счисления [1].

Алгоритм и структура преобразования двоично-десятичных чисел в код остаточных классов первого типа

Пусть задано k – разрядное двоично-десятичное число $R = (r_1, r_2, \ldots, r_m, \ldots, r_k)$, $m = 1,$

k, где $r_m = \{0, 1, \ldots, 9\}$, m-ая двоичная тетрада этого числа, которая вычисляется следующим образом:

$$r_m = \sum_{i=1}^{4} a_{im} * 2^{i-1},$$

и принимает следующее значение $r_m = (a_1\ a_2\ a_3\ a_4)$, $a_1 = \{0, 1\}$. Задан также набор модулей системы остаточных классов(СОК) $P = (P_1, P_2, \ldots, P_i, \ldots, P_n)$, $i = 1, n$. Необходимо представить число R в заданном наборе модулей СОК, т.е. осуществить операцию преобразования

вида R= => $(\alpha_1, \alpha_2, \ldots, \alpha_i, \ldots, \alpha_n)$, где

$\alpha_1 = R \bmod P_1$, $\alpha_1 = \{0, 1, \ldots, P_1 - 1\}$

$\alpha_2 = R \bmod P_2$, $\alpha_2 = \{0, 1, \ldots, P_2 - 1\}$

....

$\alpha_n = R \bmod P_n$, $\alpha_n = \{0, 1, \ldots, P_n - 1\}$.

Обычно для решения этой задачи рассматривают процедуру получения α_i по модулю: $\alpha_i = R - \left[\frac{R}{pi}\right] P_1$, где $\left[\frac{R}{pi}\right]$ – целая часть дроби $\frac{R}{p1}$, при этом всегда соблюдается условие $\alpha_1 < P_1$.

Введем обозначение $B_i = (b^i_1, b^i_2, \ldots, b^i_m, \ldots, b^i_k)$ – множество остатков, образующихся в процессе деления числа R на модуль P_1. Очевидно, что получение b^i_m можно записать как $b^i_{m-1} \circ r_m = b^i_m$ где $\circ$ означает операцию конкатенации. Тогда процедуру получения остатка α_i по модулю P_i с учетом того, что $b^i_0 = 0$ можно представить в следующем виде.

$b^i_0 \circ r_k = b^i_1 \bmod P_i$, $b^i_1 = \{0, 1, \ldots, P_i - 1\}$

$b^i_1 \circ r_{k-1} = b^i_2 \bmod P_i$, $b^i_2 = \{0, 1, \ldots, P_i - 1\}$

....

$b_{k-2} \circ r_{k-1} = b^i_{k-1} \bmod P_i$, $b^i_{k-1} = \{0, 1, \ldots, P_i - 1\}$

$b_{k-1} \circ r_k = b^i_k \bmod P_i$, $b^i = \{0, 1, \ldots, P_i - 1\}$.

Очевидно, что $b^i_k \bmod P_1$ есть искомый остаток α_i, т.е. $\alpha_i = R \bmod P_i$. Для получения остатков $B_i = = (b^i_1, b^i_2, \ldots, b^i_m, \ldots, b^i_k)$ можно воспользоваться таблицей соответствия. Пример ее формирования для модуля системы остаточных классов, равного $P_1 = 3$, приведен в табл. 1.

Размерность таблиц соответствия будет определяться следующим образом. Обозначим через L^1_1 количество различных значений, принимаемых двоичным элементом b^i_m множества B_i, а через L^1_2 – количество различных значений, принимаемых двоичной тетрадой числа R, $L^1_2 = = 10$. Тогда количество сочетаний S, определяемых различными значениями b^i_m и r_m в операции конкатенации будет равным

$$S = L^1_1 \times L^1_2.$$

Временные затраты этого преобразования зависят от разрядности преобразуемого числа R.

Алгоритм 1. Преобразования двоично-десятичного числа в систему остаточных классов по модулю P_1 содержит следующие основные операции.

Входные переменные: $R = (r_1, r_2, \ldots, r_m, \ldots, r_k)$, $m = 1, \ldots, k$ – двоично-десятичное число, P_i, $i = 1, \ldots, n$ – модули системы остаточных классов.

Выходные переменные: $A = (\alpha_1, \alpha_2, \ldots, \alpha_i, \alpha_n)$, где $\alpha_1 = R \bmod P_i$.

Метод

1. Начало.
2. $m := k$; $b^i_0 = 0$ – начальные условия.
3. $b_{m-1} \circ r_m \ \gamma \ b_m$, $m := m - 1$ выборка
4. Проверить условие "$m = 1$", да, перейти к п. 5, нет, m: = $m - 1$, перейти к п. 3
5. Конец.

Таблица 1

Таблица соответствия вида $r_m \circ b_{m-1} \Rightarrow b_m$ для получения остатка по mod $P = 3$

b_m		r_m				b_{m-1}	
0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1
1	0	0	0	0	0	1	0
0	0	0	0	0	1	0	1
0	1	0	0	0	1	1	0
1	0	0	0	0	1	0	0
0	0	0	0	1	0	1	0
0	1	0	0	1	0	0	0
1	0	0	0	1	0	0	1
0	0	0	0	1	1	0	0
0	1	0	0	1	1	0	1
1	0	0	0	1	1	1	0
0	0	0	1	0	0	0	1
0	1	0	1	0	0	1	0
1	0	0	1	0	0	0	0
0	0	0	1	0	1	1	0
0	1	0	1	0	1	0	0
1	0	0	1	0	1	0	0
0	0	0	1	1	0	0	0
0	1	0	1	1	0	0	1
1	0	0	1	1	0	1	0
0	0	0	1	1	1	0	1
0	1	0	1	1	1	1	0
1	0	0	1	1	1	0	0
0	0	1	0	0	0	1	0
0	1	1	0	0	0	0	0
1	0	1	0	0	0	0	1
0	0	1	0	0	1	0	0
0	1	1	0	0	1	0	1
1	0	1	0	0	1	1	0

Рассмотрим структуру устройства преобразования чисел из двоично-десятичной системы счисления в код системы остаточных классов, позволяющую реализацию данного алгоритма аппаратным путем. На рис. 1 приведена общая структура устройства преобразования двоично-десятичных чисел в код остаточных классов, содержащая n независимых блоков, количество которых соответствует набору заданных модулей СОК $P = (P_1, P_2, \ldots, P_i, \ldots, P_n)$. На входы всех n преобразующих блоков по соответствующим модулям P поступает в последовательном коде, начиная со старшей тетрады. Преобразуемое двоично-десятичное число A, представленное в коде с весами, 8, 4, 2, 1. После обработки последней (младшей) тетрады соответствующим блоком на выходе устройства получим код системы остаточных классов по заданным модулям:

$$A = (\alpha_1, \alpha_2, \ldots, \alpha_i, \alpha_n).$$

На рис. 2 приведена структура отдельного i-го преобразующего блока для модуля P_i, которая содержит блок памяти (БП), разделенный на ассоциативную часть (АЧ) и информационную часть ИЧ, шины синхронизации ШС1 и ШС2 и группу логических элементов. Работу отдельного i-го преобразующего блока устройства по модулю P_i можно описать следующим образом.

По управляющему импульсу (УИ), поданному на ШС 1, на АЧ БП, поступают разряды старшей тетрады исходного преобразуемого числа, т.е. r_n. По указанному адресу из ИЧ БП считывается слово, представляющее собой остаток b_1, полученный в результате деления числового значения старшей тетрады r_n на заданный модуль P_i, т.е. $r_n\, b_0 \bmod P_i$, и представленный в двоично-десятичном коде. Полученный код остатка b_1 задерживается на один такт с помощью элемента задержки. При поступлении второго

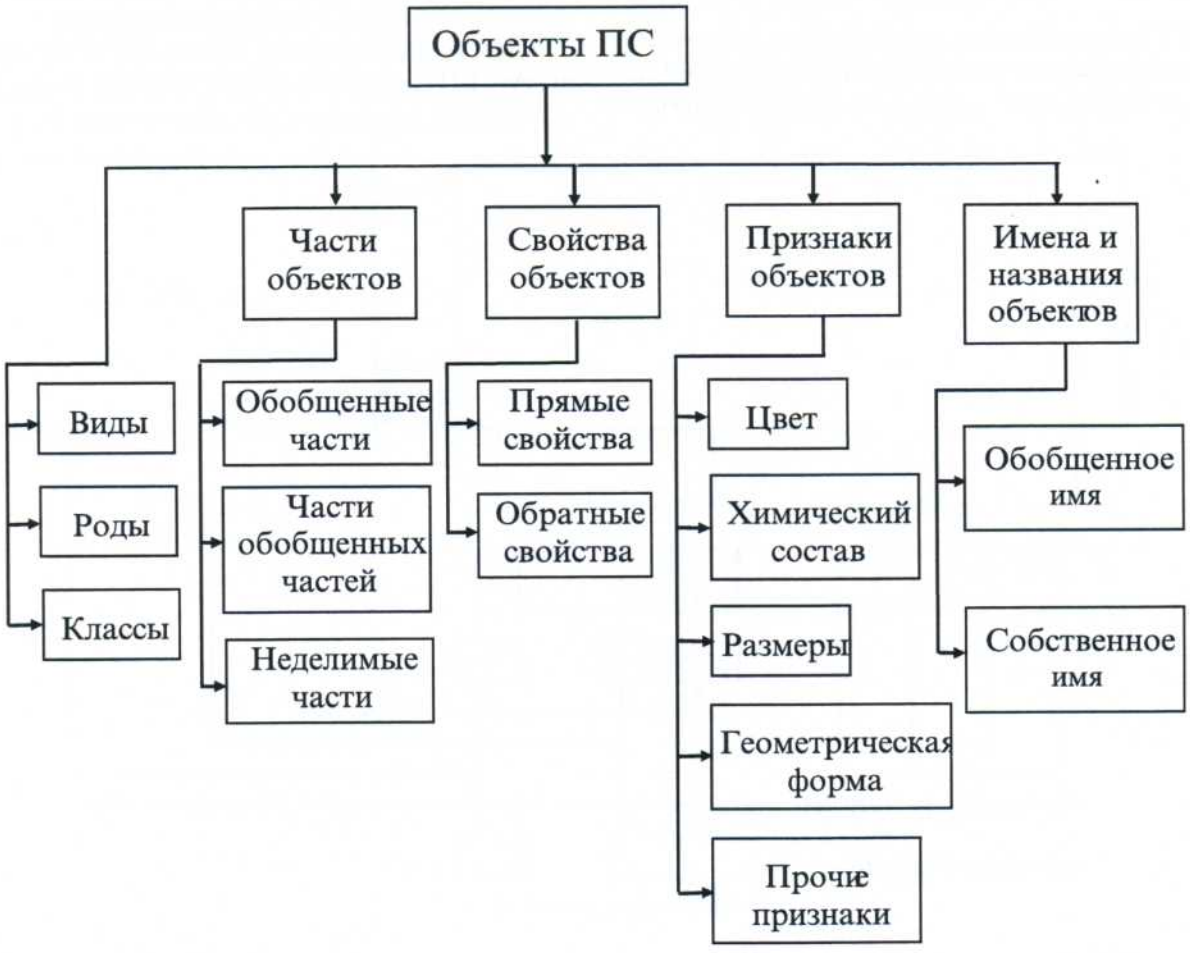


Рис 1. Структура устройства первого вида, служащая для преобразования двоично-десятичных чисел в код системы статочных классов

УИ на ШС 1 на АЧ БП поступает комбинация в виде двух или более тетрад (количество тетрад, необходимых для представления остатка b_i в двоично-десятичном коде, зависит от значения модуля P_i) $\alpha_2 b_1$. По указанному адресу из ИЧ БП считывается слово, представляющее собой остаток b_2, полученный в результате деления числового значения второй тетрады в совокупности с числовым значением предыдущего остатка b_1 на заданный модуль P_i, т.е.:

$$r_2 b_1 \bmod P_i = b_2.$$

Данный процесс повторяется до тех пор, пока не будут обработаны все тетрады преобразуемого числа $A = (r_n, r_{n-1}, \ldots, r_n)$. После обработки последней (младшей) тетрады r_1 в совокупности с тетрадами, предоставляющими значения остатка b_{n-1}, и при наличии УИ на ШС 2 получим искомый результат $\alpha_i = b_n$, соответствующий преобразованию исходного числа A в код остаточных классов по заданному модулю P_i:

$$r_n b_{n-1} \bmod P_i = b_n = \alpha_i.$$

Таким образом, предлагаемое устройство преобразования чисел из двоичной СС в СОК по модулю работает согласно следующей схеме:

$r_1 b_0 \bmod P_i = b_1$, $b_1 = 0$
$r_2 b_1 \bmod P_i = b_2$
$r_3 b_2 \bmod P_i = b_3$
.....
$r_n b_{n-1} \bmod P_i = b_n$, $b_n = \alpha_i = A \bmod P_i$.

Аналогичным образом получаются остатки и по другим модулям системы.

На рис. 3 приведена структурно-функциональная схема устройства для модуля $P_i = 3$ с прошивкой матрицы блока памяти (БП). Пусть, например, задано число $A = 193$, представленное в исходном виде в двоично-десятичной системе кодирования:

$$A = (0001\ 1001\ 0011).$$

Определим остаток данного числа по заданному модулю согласно приведенной структуре преобразующего блока. По информационным входам преобразующего блока при подаче УИ на ШС 1 на АЧ БП поступает старшая тетрада числа A $r_3 = 0001$. По указанному адресу 00 0001 из ИЧ БП считывается слово $b_1 = 01$, которое задерживается на один такт элементами задержки.

При подаче второго УИ на ШС 1 на входы АЧ БП поступает вторая тетрада числа A, $r_2 = 1001$ в совокупности с остатком b_1, т.е.

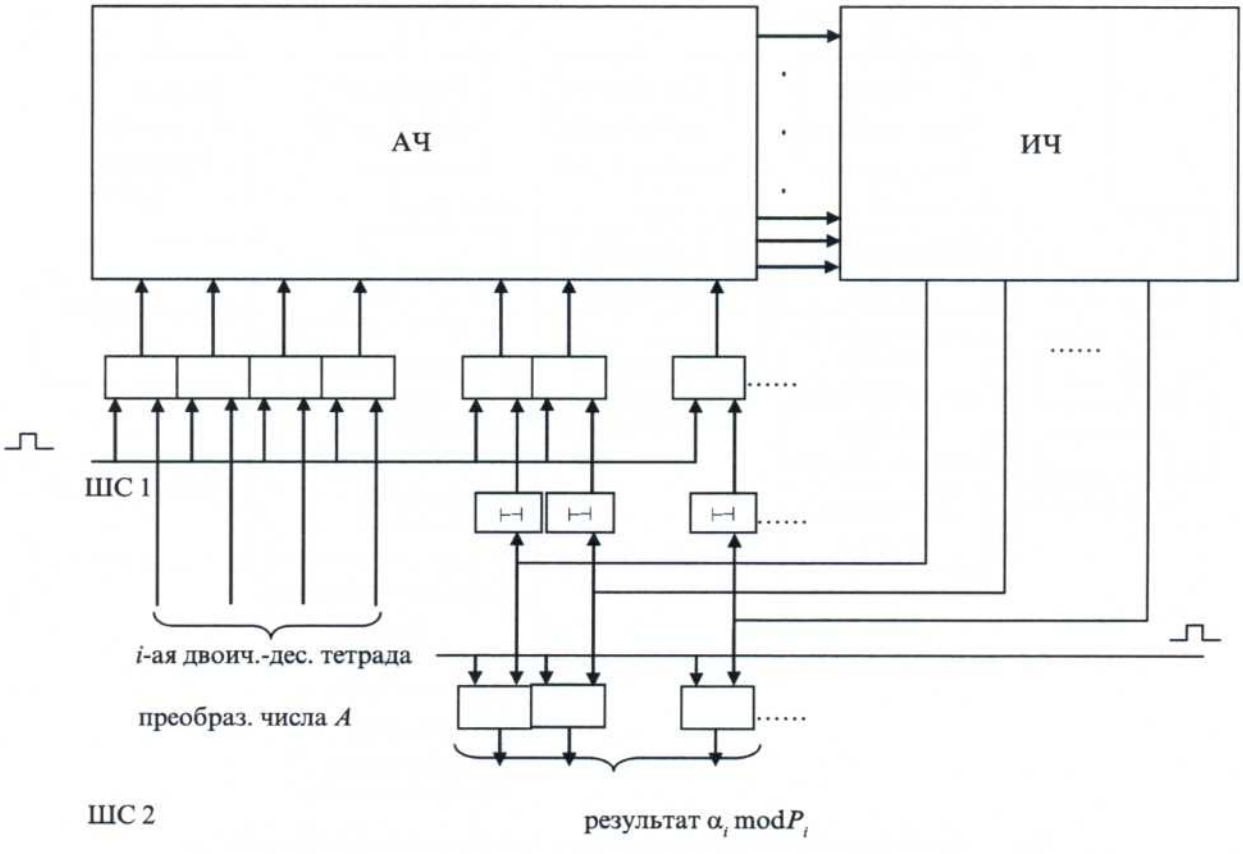


Рис. 2. Структура i-го преобразующего блока по модулю P_i

образуется адрес 01 1001, по которому из ИЧ БП считывается слово $b_2 = 01$.

Соответственно при подаче третьего УИ на ШС 1 на входы АЧ БП поступает последняя (младшая) тетрада числа r_1, которая в совокупности с остатком b_2 и образует адрес 010011, по которому из ИЧ БП считывается слово $b_3 = 01$, которое при подаче УИ по ШС 2 поступает на выходные шины устройства и представляет собой искомый результат преобразования числа A по модулю $P_i = 3$.

Оценим затраты памяти, необходимые для структурной реализации алгоритма преобразования чисел из двоично-десятичной СС в СОК.

Для определения затрат ПЗУ в словах для i-го преобразующего блока по модулю P_i необходимо учитывать разрядность входного слова, которая равна

$$m_i = r + [\log_2 P_i],$$

где r – количество двоичных разрядов, необходимых для представления десятичной цифры числа A в двоично-десятичной СС. Тогда

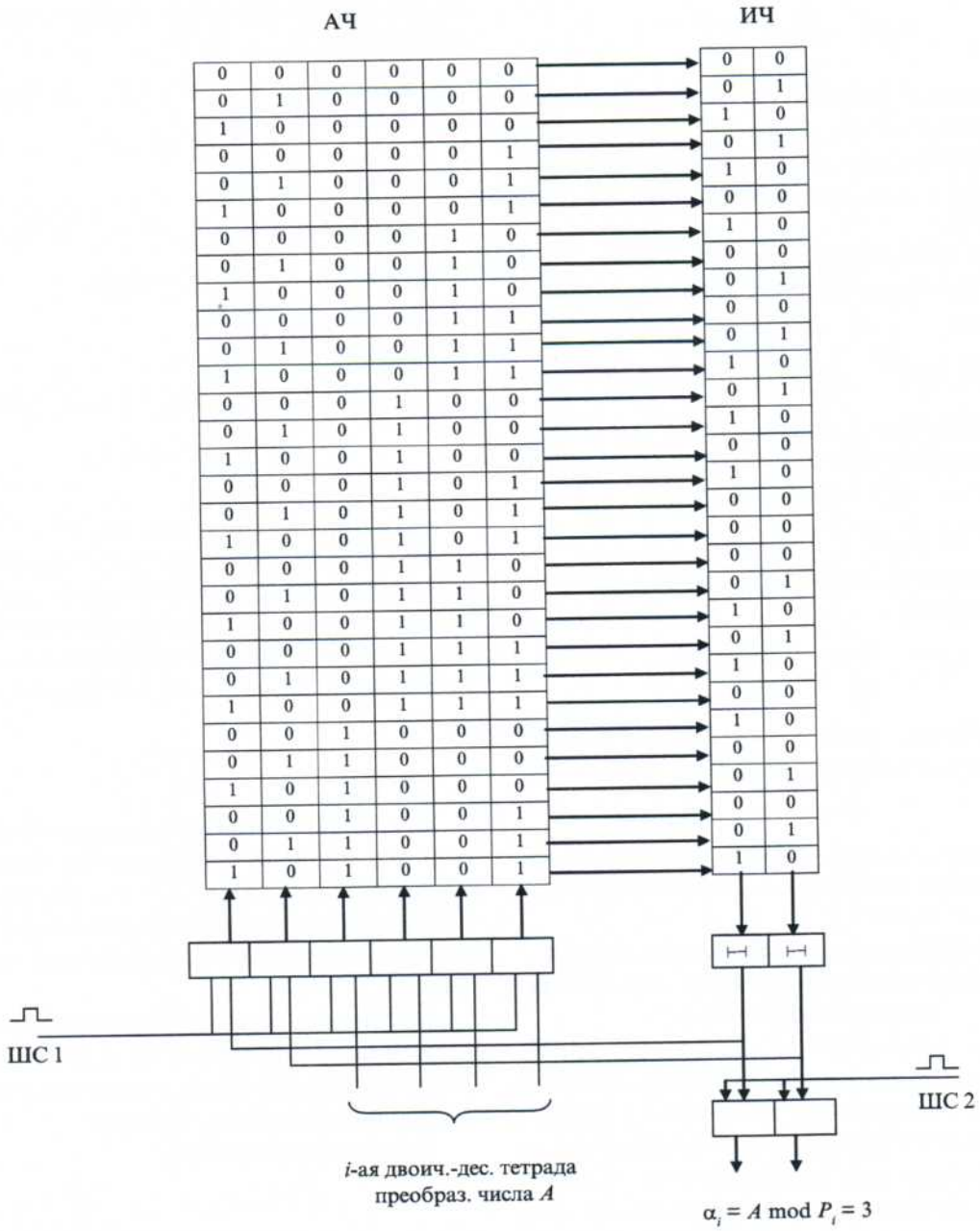


Рис.3. Пример реализации i-го преобразующего блока двоично-десятичных чисел в код СОК по mod $P_i = 3$

затраты памяти ПЗУ в словах $V_{ic}^{\text{ПЗУ}}$ для i-го преобразующего блока можно определить по формуле:

$$V_{ic}^{\text{ПЗУ}} = 2^{r+[\log_2 P_i]}.$$

Если задана система модулей $P = (P_1, P_2, \ldots, P_n)$, то общие затраты памяти ПЗУ в словах $V_c^{\text{ПЗУ}}$ для всех преобразующих блоков по модулям СОК определяются как $V_c^{\text{ПЗУ}} = V_{1c}^{\text{ПЗУ}} + V_{2c}^{\text{ПЗУ}} + \ldots + V_{nc}^{\text{ПЗУ}}$. Подставляя соответствующие значения $V_{ic}^{\text{ПЗУ}}$, получим:

$$V_{ic}^{\text{ПЗУ}} = \sum_{i=0}^{n} 2^{r+[\log_2 P_i]}.$$

Для определения затрат памяти ПЗУ в битах для i-го преобразующего блока по модулю P_i необходимо учитывать разрядность выходного слова I_i в двоичном коде, которая равна $I_i = [\log_2 P_i]$.

Тогда затраты памяти ПЗУ в битах $V_{iB}^{\text{ПЗУ}}$ для i-го преобразующего блока можно определить как

$$V_{iB}^{\text{ПЗУ}} = V_{ic}^{\text{ПЗУ}} * I_i = 2^{r+[\log_2 P_i]} * [\log_2 P_i].$$

Общие затраты памяти ПЗУ в битах $V_B^{\text{ПЗУ}}$ для всех преобразующих блоков по модулям $P = (P_1, P_2, \ldots, P_n)$ определяются как

$$V_B^{\text{ПЗУ}} = V_{1B}^{\text{ПЗУ}} + V_{2B}^{\text{ПЗУ}} + \ldots + V_{nB}^{\text{ПЗУ}}.$$

Подставляя соответствующие значения $V_{iB}^{\text{ПЗУ}}$, получим

$$V_B^{\text{ПЗУ}} = \sum_{i=0}^{n} 2^{r+[\log_2 P_i]} * [\log_2 P_i].$$

Временные затраты в тактах сравнения T определяются следующим выражением:

$$T = \tau * k,$$

где τ – время считывания информации из памяти, k – количество цифр r_i в исходном двоично-десятичном числе A.

Алгоритм и структура преобразования двоично-десятичных чисел в код остаточных классов второго вида

Выше показано, что алгоритм 1 преобразования первого вида состоит в последовательном нахождении остатков b_m, причем $b_k = \alpha_i$, который определяется k-ой комбинацией последовательных промежуточных величин. Отличие алгоритма второго вида заключается в параллельном нахождении величин b_m^{*i} для разложения числа R:

$$r_k * 10^{k-1} \bmod P_1 = b_1^{*1}, r_{k-1} * 10^{k-2} \bmod P_1 = b_2^{*1}, \ldots, r_1 * 10^0 \bmod P_1 = b_k^{*1}$$

$$r_k * 10^{k-1} \bmod P_2 = b_1^{*2}, r_{k-1} * 10^{k-2} \bmod P_2 = b_2^{*2}, \ldots, r_1 * 10^0 \bmod P_2 = b_k^{*2}$$

$$r_k * 10^{k-1} \bmod P_n = b_1^{*n}, r_{k-1} * 10^{k-2} \bmod P_n = b_2^{*n}, \ldots, r_1 * 10^0 \bmod P_n = b_k^{*n}$$

Тогда для нахождения

$$\alpha_1 = R \bmod P_1, \alpha_2 = R \bmod P_2, \ldots, \alpha_n = R \bmod P_n,$$

можно воспользоваться алгоритмом для суммирования чисел в остаточных классах по модулям $P_1, P_2, \ldots, P_i, P_n$:

$$\alpha_1 = (b_1^{*1} + b_2^{*1} + \ldots + b_m^{*1} + \ldots b_k^{*1}) \bmod P_1 = \sum_{m=1}^{k} (b_m^{1*}) \bmod P_1$$

$$\alpha_2 = (b_1^{*2} + b_2^{*2} + \ldots + b_m^{*2} + \ldots b_k^{*2}) \bmod P_2 = \sum_{m=1^*}^{k} (b_m^{2*}) \bmod P_2$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$\alpha_n = (b_1^{*n} + b_2^{*n} + \ldots + b_m^{*n} + \ldots b_k^{*n}) \bmod P_n = \sum_{m=1}^{k} (b_m^{n*}) \bmod P_n$$

Для определения величин b_1^{*1}, b_1^{*2} … b_1^{*n} можно использовать таблицы соответствия, количество которых определяется максимальной разрядностью позиционного числа R, так как диапазон Д представления чисел в СОК по модулям $P = (P_1, P_2, \ldots, P_n)$ определяется как

$$\text{Д} = \prod_{i=1}^{n} P_i.$$

В случае трехразрядного двоично-десятичного числа R справедливы следующие таблицы соответствия 2, 3 и 4. Размерность S таблиц соответствия определяется количеством значений, принимаемых двоичной тетрадой числа R.

Обозначим через $L_i^{\text{СОК}}$ сумматор, функционирующий в СОК по модулю P_i, тогда алгоритм 2 преобразования чисел второго вида содержит следующие основные операции.

Входные переменные:

$$R = (r_1, r_2, \ldots, r_m, \ldots r_k),$$

$m = 1, \ldots, k$ – двоично-десятичное число, P_i, $i = 1, \ldots, n$ – модули системы остаточных классов.

Выходные переменнные:

$A = (\alpha_1, \alpha_2, \ldots, \alpha_i, \alpha_n)$, где $\alpha_1 = R \bmod P_i$.

Метод

1. Начало.
2. Загрузить нуль в сумматор $L_i^{\text{СОК}}$
3. Определить b_m^{*i} по цифрам r_m по таблицам соответствия.
4. Выполнить операцию параллельного суммирования в СОК по модулю P_i.
5. Конец. Содержимое сумматора $L_i^{\text{СОК}}$ есть представление числа R в СОК по модулю P_i.

Рассмотрим структуру устройства преобразование чисел из двоично-десятичной СС в код системы остаточных классов.

Принцип работы устройства чисел из двоично-десятичной СС в СОК второго вида приводится на рис. 4 и заключается в следующем. Каждая двоичная тетрада числа $A^{(2-10)}$, являющаяся адресом, поступает на АЧ, количество которых определяется разрядностью k преобразуемого числа $A^{(2-10)}$.

Из ИЧ соответствующих БП считываются остатки $b_1^*, b_2^*, \ldots, b_n^*$:

Таблицы соответствия для примера преобразуемого трехразрядного двоично-десятичного числа A в код СОК

Таблица 2

$r_2 * 10^2$				b_1^{2*}				b_1^{3*}				b_1^{4*}			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0
0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0
0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	1	0	1	0	1	0	0	0	0	0	0	1	1	0	0
0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	0
0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0

Таблица 3

$r_2 * 10^1$				b_2^{2*}				b_2^{3*}				b_2^{4*}			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0
0	0	1	0	0	1	0	0	0	0	0	0	1	1	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	1	0	0	0	0	0	1	0	1	0
0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0
0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0

Таблица 4

$r_2 * 10^0$				b_3^{2*}				b_3^{3*}				b_3^{4*}			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	1	0	0	1	0	0	1	0	0	0
0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0
0	0	1	1	1	0	0	0	1	1	0	1	1	0	0	0
0	1	0	0	0	0	1	1	0	0	1	0	0	1	0	0
0	1	0	1	1	1	0	0	0	0	1	0	1	0	0	0
0	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0
0	1	1	1	1	0	1	0	1	0	0	0	0	1	0	0
1	0	0	0	0	1	0	0	1	1	0	0	1	0	0	0
1	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0

$$r_1 * 10^0 \bmod P_1 = b_1^{1*},\ r_1 * 10^0 \bmod P_2 = b_2^{1*},\ \ldots,\ r_1 * 10^0 \bmod P_n = b_n^{1*}$$

$$r_2 * 10^1 \bmod P_1 = b_1^{2*},\ r_2 * 10^1 \bmod P_2 = b_2^{2*},\ \ldots,\ r_2 * 10^1 \bmod P_n = b_n^{2*}$$

.....

$$r_k * 10^{k-1} \bmod P_1 = b_1^{k*},\ r_k * 10^{k-1} \bmod P_2 = b_2^{k*},\ \ldots,\ r_k * 10^{k-1} \bmod P_n = b_n^{k*}$$

Остатки по модулю b_1^{1*}, полученные с ИЧ всех блоков ассоциативной памяти, поступают на входы параллельного сумматора чисел по модулю P_i, в результате чего на выходе сумматора по модулю P_i сформируется результат $d_i \bmod A$. Аппаратурные затраты устройства определяются затратами параллельных сумматоров по модулям $P_1, P_2, \ldots, P_n$ и затратами БП, которые в словах равны $V_c = 10$, а в битах определяются выражением:

$$V_B = 10n * \sum_{i=0}^{n} [\log_2 P_i].$$

Временные затраты зависят от характеристик параллельного сумматора чисел, функционирующего в остаточных классах.

Алгоритм и структура устройства преобразования двоичных чисел в систему остаточных классов

Разработка алгоритма преобразования чисел из двоичной системы счисления в СОК по аналогии с алгоритмом 1 осуществляется следующим образом.

Пусть дано преобразуемое m – разрядное двоичное число $A^{(2)}$, а также система модулей СОК $P = (P_1, P_2, \ldots, P_n)$. Необходимо определить:

$\alpha_1 = A^{(2)} \bmod P_1$,
$\alpha_2 = A^{(2)} \bmod P_2$,
.....
$\alpha_n = A^{(2)} \bmod P_n$.

Как было показано ранее, остаток определяется согласно выражению:

$$\alpha_i = A - \left[\frac{A}{Pi}\right] P_1 .$$

Разобьем исходное преобразуемое m разрядное двоично число $A^{(2)}$ на S равных компонентов, причем i-ая компонента пусть содержит r_i двоичных разрядов. Тогда множество $R = (r_1, r_2, \ldots, r_j, \ldots r_s), j = 1, s$, определяет структуру разбиения исходного преобразуемого двоичного числа $A^{(2)}$.

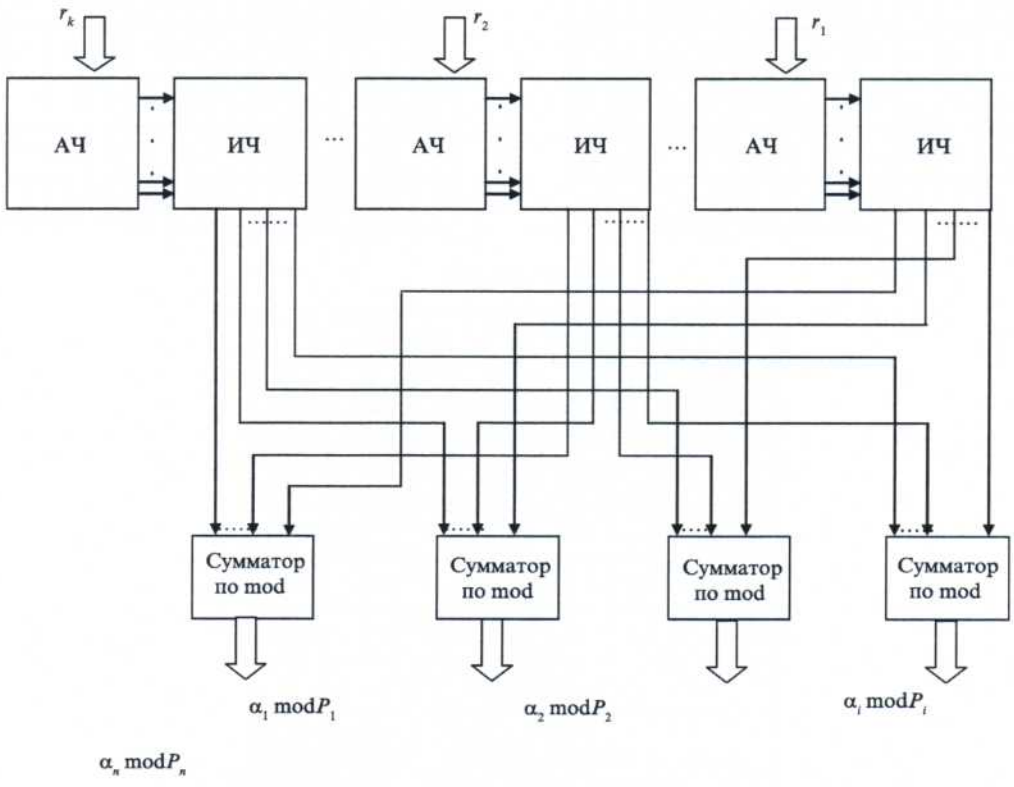


Рис. 4. Структура устройства преобразования чисел второго вида из двоично-десятичной СС в код системы остаточных классов

Согласно данной структуре разбиения, число $A^{(2)}$ может быть представлено последовательностью упорядоченных множеств следующего вида:

$$A^{(2)} = \{a_{ir1}\}, \{a_{ir2}\}, \ldots, \{a_{irj}\}, \ldots, \{a_{irs}\},$$
$$ir_j = 1,\ldots, r_j,$$

где $a_{irj} = a_{i+(r1+r2+\ldots+rj-1)}$, a_{irj} есть значение разряда в исходном представлении числа $A^{(2)}$, который соответствует i-ой позиции в r_1 компоненте структуры разбиения числа $A^{(2)}$. Тогда число $A^{(2)}$ в зависимости от значения r_j, $j = 1,\ldots, s$ можно представить в двоичной ($r_1 = 1$), четверичной ($r_1 = 2$), восьмеричной ($r_1 = 3$), шестнадцатеричной ($r_1 = 4$) и других системах счисления.

На основании этого можно получить таблицы соответствия, представляющие собой результаты деления исходного числа $A^{(2)}$ на модули $P_1, P_2, \ldots, P_n$ с нахождением промежуточных остатков $b_0, b_1, \ldots, b_s$, то есть

$\alpha_s \circ b_1 \bmod P_i = b_2 = \alpha_s, b_1 = 0$

$\alpha_{s-1} \circ b_2 \bmod P_i = b_3$

.....

$\alpha_i \circ b_{s-1} \bmod P_i = b_s = \alpha_i, \alpha_i = A^{(2)} \bmod P_i$

Пример организации таблицы соответствия для случая, когда $r_1 = 4$ (т.е. двоичное число $A^{(2)}$ представлено в шестнадцатеричной системе счисления) и $P_1 = 3$ приведен в табл. 5.

Таблица 5

Таблица соответствия вида $b_{i-1} \circ r_i \Rightarrow b_i$

b_{i-1}		r_i				b_i	
0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	1
0	0	0	0	1	0	1	0
0	0	0	0	1	1	0	0
0	0	0	1	0	0	0	1
0	0	0	1	0	1	1	0
0	0	0	1	1	0	0	0
0	0	0	1	1	1	0	1
0	0	1	0	0	0	1	0
0	0	1	0	0	1	0	0
0	0	1	0	1	0	0	1
0	0	1	0	1	1	1	0
0	0	1	1	0	0	0	0
0	0	1	1	0	1	0	1
0	0	1	1	1	0	1	0
0	0	1	1	1	1	0	0
0	1	0	0	0	0	0	1
0	1	0	0	0	1	1	0
0	1	0	0	1	0	0	0
0	1	–	–	–	–	–	–
0	1	1	1	1	0	0	0

Окончание табл. 5

b_{i-1}		r_i				b_i	
0	1	1	1	1	1	0	1
1	0	0	0	0	0	1	0
1	0	0	0	0	1	0	0
1	0	0	0	1	0	0	0
1	0	–	–	–	–	–	–
1	0	1	1	1	1	1	0

Временные затраты в тактах алгоритма преобразования чисел из двоичной системы счисления в СОК по модулям P_1 равны $T = S$, где S – количество разбиений исходного числа $A^{(2)}$ на равные компоненты r_j. Объем таблицы соответствия в словах по модулю P_1 равен:

$$V = [\log_2 P_i] * 2^{\eta}, \quad i = \overline{1,n}, \; j = \overline{1,S}.$$

Структурная организация устройства преобразования чисел из двоичной СС в код остаточных классов по модулям $P = (P_1, P_2, \ldots, P_n)$ отличается от принципов построения устройств преобразования первого вида из двоично-десятичной СС в СОК только тем, что в этом случае необходимы другие таблицы соответствия, определяемые от структур разбиения исходного преобразуемого двоичного числа $A^{(2)}$. На рис. 5. приведен пример реализации структуры преобразования двоичных чисел в код СОК для случая разбиения $A^{(2)}$ на тетрады с определением кода остатка по модулю $P = 3$.

Работу устройства рассмотрим на примере преобразования двоичного числа $A^{(2)}$ = 1011 0111 1101.

Разобьем исходное число $A^{(2)}$ на тетрады r_3 = 1011, r_2 = 0111, r_1 = 1101, т.е. представим его в двоично-шестнадцатеричной форме счисления $A^{(2-16)}$.

В первом такте по сигналу, поданному на ШС1, на вход АЧ БП поступает старшая тетрада r_3 = 1011, которая совместно с информацией, считанной в предыдущем такте из ИЧ блока памяти, образует ассоциативный признак. Так как в предыдущем такте из ИЧ БП не считывалась информация, т.е. b_0 = 00, то АП имеет вид 00 1011. По указанному адресу из ИЧ БП считывается слово b_1 = 10.

Во втором такте по сигналу, поданному на ШС1, на АЧ БП поступает АП, имеющий вид 10, 0111, по которому из ИЧ БП считывается слово b_2 = 00.

В третьем такте по сигналу, поданному на ШС1, на вход АЧ БП поступает АП, имеющий вид 00 1101, по которому из ИЧ БП считывается слово b_3 = 01, которое при подаче тактового им-

пульса на ШС2 поступает на выход устройства $b_3 = \alpha = A^{(2)} \bmod 3$.

Временные затраты алгоритма в тактах сравнения определяются структурой разбиения исходного числа $A^{(2)}$, т.е. количеством равных компонентов S в структуре разбиения, и равны $T = S$.

Аппаратурные затраты составляют для i-го преобразующего блока в словах определяются выражением:

$$V_c = 2^{ri} * P_i;\ V_B = \{2^{ri} * P_i\} * \{r_i + [\log_2 P_i] * 2\}.$$

Общие затраты преобразующего устройства по всем модулям P = (P_1, P_2, …, P_n) вычисляются соответственно по формулам:

$$V_c^0 \sum_{i=0}^{n} 2^{ri} * P_i;\ \ V_c^0 \sum_{i=0}^{n} 2^{ri} * P_i * (r_i + 2 * [\log_2 P_i]).$$

Резюмируя вышеизложенное следует отметить, что предложенные в работе алгоритмы и их аппаратная реализация позволяют эффективным образом выполнять преобразования числовых данных из позиционной системы счисления

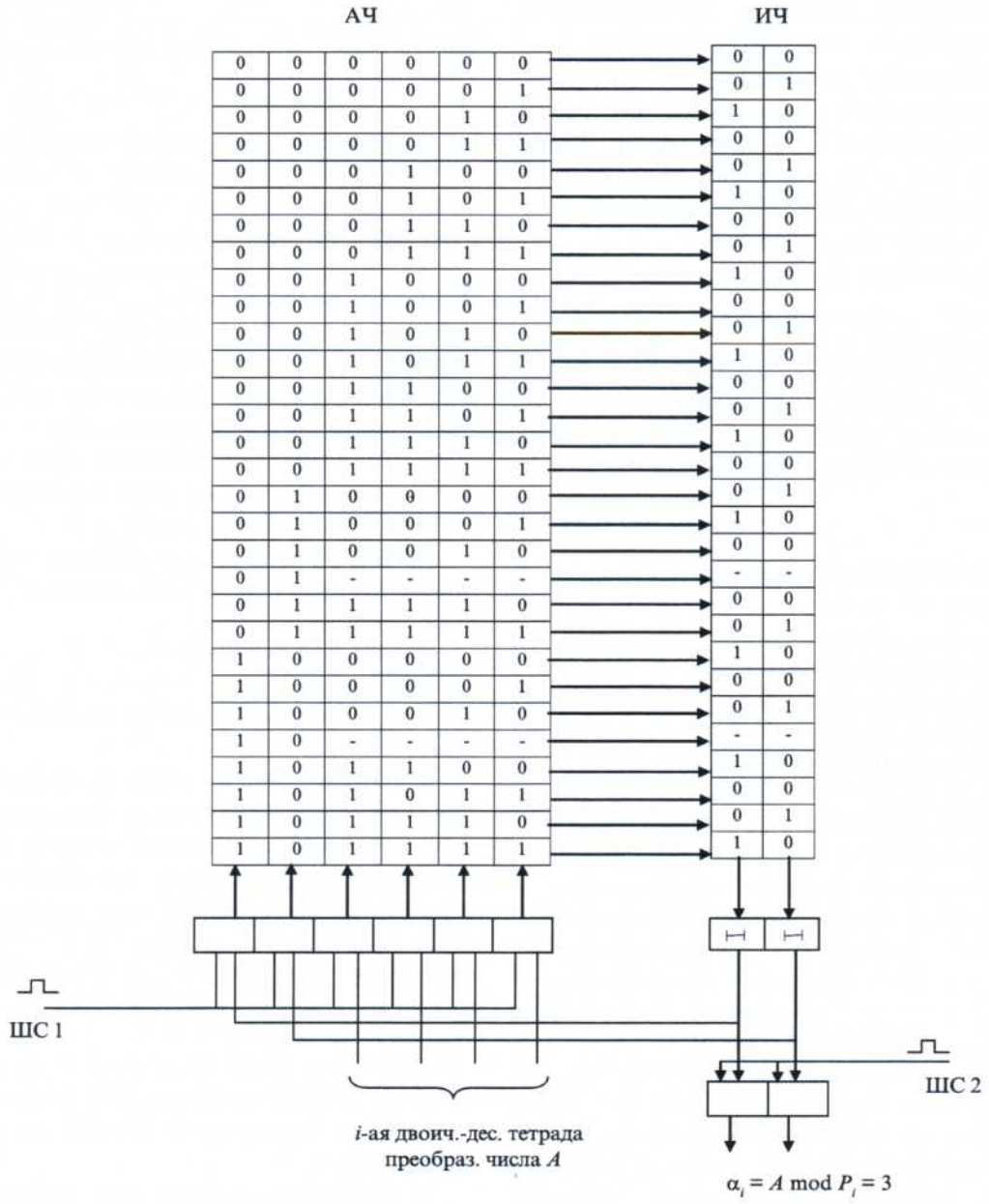


Рис. 5. Пример реализации структуры устройства преобразования двоичных чисел в СОК

в систему остаточных классов и на этой основе проектировать специализированные вычислительные устройства для нейрокомпьютеров, обладающие высоким быстродействием.

СПИСОК ЛИТЕРАТУРЫ

1. **Акушский И.Я., Юдицкий Д.И.** Машинная арифметика в остаточных классах. – М. : Сов. радио. 1968. – 440 с.

Новиков К.В.

Быстродействующий алгоритм сегментации изображений на основе спектральных свойств анизотропных областей

Одной из распространенных задач в области обработки изображений является задача сегментации, понимаемая как разбиение изображения на однородные области. В качестве признака сегментации обычно используется какая-либо интегральная характеристика окрестности точки, по которой определяется принадлежность ее к соответствующей области [1]. В данной работе рассматривается сегментация изображений на анизотропные и изотропные области, где анизотропным областям соответствуют участки изображения, имеющие ярко выраженные корреляционные связи в определенном направлении (выраженную анизотропность). Искомые анизотропные области здесь – следы в виде параллельных царапин, оставленные режущей (царапающей) кромкой некоторого инструмента на некоторой поверхности. Результатом сегментации является бинарное изображение, где уровню черного соответствует изотропная область, уровню белого – анизотропная. В дальнейшем изображения найденных анизотропных областей используются для решения задачи идентификации инструментов, оставивших эти царапины.

В данной статье предлагается новый вариант быстродействующей реализации алгоритма сегментации на основе [2]. Необходимо отметить, что в алгоритме из [2] применяется достаточно распространенный в литературе подход, использующий характерные спектральные свойства анизотропных областей (см., например, [4], [5]), а также выбор спектрального признака анизотропии на основе теоремы о центральном слое (см., например, [6]) наилучшим образом описывающего рассматриваемую в [2] анизотропию.

Алгоритм сегментации

В [2] первоначально посредством обработки в скользящем окне изображение стационаризируется на основе известного подхода (см., например, [8]) компенсацией локальных среднего и дисперсии при помощи соотношения вида:

$$s_{i,j} = \frac{z_{i,j} - \hat{m}_{i,j}}{\hat{\sigma}_{i,j}}, \qquad (1)$$

где $z_{i,j}$ и $s_{i,j}$ – соответственно значения яркости исходного и преобразованного изображений в точке с координатами i, j; $\hat{m}_{i,j}$; $\hat{\sigma}_{i,j}$ – соответственно оценка математического ожидания и среднеквадратического отклонения (СКО) в окне размером $m \times m$ в окрестности точки с координатами i, j.

Второй этап состоит в том, что осуществляется обработка стационаризированного изображения скользящим окном размера $n \times n$. Для каждого положения скользящего окна вычисляется спектр изображения в этом окне и производится поиск максимума функции вида:

$$e(\theta) = \sum_{k=1}^{n/2} \left| S([k \cdot \cos(\theta)],[k \cdot \sin(\theta)]) \right|^2, \qquad (2)$$

где $S(,)$ – дискретный амплитудный спектр фрагмента с размерами $n \times n$ изображения S, [] – функция округления. Здесь $S(,)$вычисляется в декартовой системе координат.

На третьем этапе данного алгоритма формируется препарат в соответствии с выражением:

$$v_{i,j} = \max_{\theta} e(\theta). \qquad (3)$$

Так как $S(,)$ обладает центральной симметрией, то функция $e(\theta)$ имеет два одинаковых максимума: при $\theta = \theta_{max}$ и $\theta = \theta_{max} + \pi$. Следовательно, поиск максимума можно производить в пределах от $-\pi/2$ до $\pi/2$.

На четвертом этапе происходит пороговая обработка препарата v и формирование итогового препарата в соответствии со следующим правилом:

$$r_{i,j} = \begin{cases} 1 \text{ (анизотропная область)}, & v_{i,j} \geq C \\ 0 \text{ (изотропная область)}, & v_{i,j} < C \end{cases}, \qquad (4)$$

где C – некоторый порог. Так как изображение на первом этапе было стационаризировано с помощью (1), то можно говорить, что на втором этапе участки стационаризированного изображения S для различных положений скользящего окна при достаточно большом n, будут иметь примерно одинаковое среднее (около нуля) и примерно одинаковую дисперсию (около единицы). Поэтому признак анизотропии на основе (2)–(4), по сути, означает, что за анизотропный участок принимается тот, у которого основная доля всей энергии сосредоточена в гармониках, расположенных вдоль прямой, проходящей через начало координат энергетического спектра. Направление этой прямой несущественно и обозначает лишь направление анизотропии (направление сильных корреляционных связей данного участка). Таким образом, выбирая порог c в выражении (4) мы, по сути, определяем, какая доля всей энергии участка должна быть сосредоточена вдоль этой прямой, чтобы мы могли считать данный участок областью с выраженной анизотропией.

В [2] уже говорилось, что в соответствии с теоремой о центральном слое (см., например, [6]), предложенный алгоритм на втором и третьем этапах является, по сути, алгоритмом поиска наилучшей в среднеквадратическом смысле аппроксимации участка изображения функцией вида "цилиндр":

$$g(x, y) = f(x \cdot \cos(\xi) + y \cdot \sin(\xi)), \qquad (5)$$

где $g(x, y)$ – двумерная функция, описывающая "цилиндр", $f(\cdot)$ – функция одной переменной, ξ – угол поворота линий постоянного уровня "цилиндра", которые являются прямыми. Причем $\xi = \theta + \frac{\pi}{2}$, а $e(\theta)$ в выражении (2) с точностью до константы является энергией аппроксимации $g(x, y)$. Таким образом, в предлагаемом здесь алгоритме определяется энергетический вклад найденной аппроксимации в функцию, описывающую анализируемый участок изображения. И, следовательно, порог C в выражении (4) определяет, какая доля энергии изображения должна быть сосредоточена в его наилучшей (в смысле среднеквадратического отклонения) аппроксимации (5), чтобы данное изображение считалось анизотропным. Все это определяет пределы изменения порога на этапе "обучения" алгоритма для обнаружения требуемых видов анизотропии.

Необходимо отметить, что $f(\cdot)$ в аппроксимации (5), с точностью до константы, соответствует одной из так называемых проекций радоновского образа исследуемого участка изображения. Таким образом, $e(\theta)$ в выражении (2), с точностью до константы, является энергией проекции соответствующего радоновского образа. В связи с этим можно сказать, что предложенный алгоритм на втором и третьем этапах близок к известному в литературе (см., например, [8], а также [9]) алгоритму обработки изображений (построения поля направлений), основанному на поиске у анализируемого участка изображения проекции радоновского образа с максимальной дисперсией. В [8] также отмечается, что методы, основанные на непосредственном вычислении преобразования Радона, являются достаточно громоздкими и трудоемкими, если требуется вычислять радоновские образы с достаточно малым шагом дискретизации ξ.

Из всего вышеизложенного следует, что предложенный алгоритм сегментации на втором и третьем этапах может быть реализован в пространственной области следующим образом. Для каждого положения скользящего окна значение ξ меняется N раз в пределах от $-\pi/2$ до $\pi/2$ с соответствующим шагом дискретизации, и таким образом, вычисляется набор из N проекций радоновского образа. Из этих проекций выбирается та, чья энергия является

наибольшей. Эта энергия соответствует $v_{i,j}$ выражения (3). В [9] рассматривается ситуация, где радоновские образы необходимо вычислять для каждого положения скользящего окна. Данная методика получила название – локализированное преобразование Радона. Там же отмечается высокая помехоустойчивость этого преобразования при обнаружении локальных линейных объектов на изображениях. Локализированное преобразование Радона в [9] представлено следующим образом:

$$\tilde{f}(\xi,\rho,\sigma) = \int_{X_{\min}}^{X_{\max}} \int_{Y_{\min}}^{Y_{\max}} s(x,y)\cdot\delta(\rho - x\cdot\cos(\xi) - y\cdot\sin(\xi))dxdy, \quad (6)$$

$$X_{\min} = \min[\rho\cdot\cos(\xi) - \sigma\cdot\sin(\xi), \rho\cdot\sin(\xi) - (\sigma+\lambda)\cdot\sin(\xi)],$$

$$X_{\max} = \max[\rho\cdot\cos(\xi) - \sigma\cdot\sin(\xi), \rho\cdot\sin(\xi) - (\sigma+\lambda)\cdot\sin(\xi)],$$

$$Y_{\min} = \rho\cdot\sin(\xi) + \sigma\cdot\cos(\xi),\ Y_{\max} = \rho\cdot\sin(\xi) + (\sigma+\lambda)\cdot\cos(\xi),$$

где $\tilde{f}(\xi,\rho,\sigma)$ – локализированное преобразование Радона; ξ и ρ – параметры, однозначно определяющие прямую линию (наклон и местоположение); λ – длина отрезка, расположенного на данной линии, вдоль которого происходит интегрирование; σ – параметр, определяющий положение (сдвиг) отрезка λ на данной линии; $s(x, y)$ – обрабатываемое изображение; $\delta(\cdot)$ – дельта-функция Дирака. При фиксированном ξ $\tilde{f}(\xi,\rho,\sigma)$ является функцией двух переменных. Где пары значений ρ и σ, по сути, однозначно соответствуют координатам точек изображения, а значение $\tilde{f}(\xi,\rho,\sigma)$ – интеграл (в дискретном варианте – сумма) значений изображения вдоль отрезка длинной λ и наклоном, соответствующим ξ в точке изображения задаваемой координатами ρ и σ. Таким образом, функция двух переменных $\tilde{f}(\xi,\rho,\sigma)$ (при фиксированном ξ) однозначно соответствует препарату, где каждой точке с координатами x, y ставится в соответствие интеграл (в дискретном варианте – сумма) значений изображения вдоль отрезка с наклоном $\theta+\frac{\pi}{2}$, длиной λ, середина данного отрезка имеет координаты x, y. Как известно, такой препарат можно получить сверткой отсчетов изображения s с импульсной характеристикой вида:

$$h_\theta^1(x,y) = \begin{cases} \delta(x\cdot\cos(\theta+\frac{\pi}{2}) + y\cdot\sin(\theta+\frac{\pi}{2})), & \text{при} \quad \sqrt{x^2+y^2} \le \frac{\lambda}{2} \\ 0, & \text{в противном случае.} \end{cases} \quad (7)$$

Данный препарат на основе свертки в дискретной области можно реализовать при помощи быстрого преобразования Фурье (БПФ) (см., например, [5]) либо при помощи рекурсии (см., например, [7]), что даст алгоритм с достаточно малыми вычислительными затратами. Необходимо отметить, что совокупность значений данного препарата вдоль отрезка длинной λ с центром в точке с координатами x, y, проходящего под углом θ, представляет собой функцию одной переменной, которая есть проекция радоновского образа изображения в скользящем окне с координатами x, y для угла $\theta+\frac{\pi}{2}$. Следовательно, сумма квадратов значений данного препарата вдоль отрезка длиной λ, с центром в точке с координатами x, y, проходящего под углом θ представляет собой энергию проекции радоновского образа изображения в скользящем окне с координатами x, y, полученной для угла $\theta+\frac{\pi}{2}$. В связи с этим алгоритм нахождения препарата, где каждому значению с координатами x, y соответствует энергия проекции радоновского образа изображения в скользящем окне с координатами x, y в соответствии с заданным углом θ, состоит из следующих этапов.

1) Свертка изображения с импульсной характеристикой (7).

2) Возведение отсчетов полученного препарата в квадрат.

3) Свертка препарата найденного на этапе 2) с импульсной характеристикой вида:

$$h_\theta^2(x,y) = \begin{cases} \delta(x\cdot\cos(\theta) + y\cdot\sin(\theta)), & \text{при} \quad \sqrt{x^2+y^2} \le \frac{\lambda}{2} \\ 0, & \text{в противном случае.} \end{cases} \quad (8)$$

Таким образом, мы сформируем набор N препаратов для N дискретных значений угла θ. Каждой точке с координатами x, y соответствует N значений полученных препаратов. Далее из этого набора значений выбирается максимальное и присваивается точке с координатами x, y окончательного препарата. Этот оконча-

тельный препарат будет соответствовать препарату v, полученному на основе (3). На рис. 1. представлена блок-схема данной реализации алгоритма.

предложенного алгоритма для реальных изображений. На рис. 2 слева приведено исходное изображение, где присутствуют следы в виде совокупности практически параллельных ца-

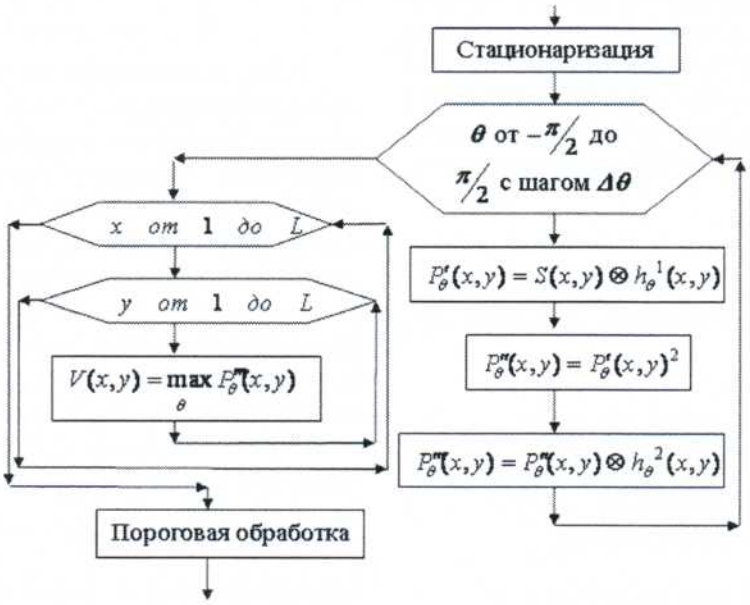


Рис.1. Блок-схема алгоритма сегментации

На рисунке 1 $L \times L$ – размеры сегментируемого изображения, $P'_\theta(x, y)$, $P''_\theta(x, y)$; $P'''_\theta(x, y)$ – соответственно препараты, вычисленные на первом, втором и третьем этапах обработки стационарного "изображения" s.

Изображения, представленные на рис. 2 – рис. 4. иллюстрируют результаты работы

рапин, оставленных неким инструментом. Данные следы образуют искомую анизотропную область. Также на этом изображении помимо искомых царапин инструмента, есть еще и множество объектов, похожих на царапины, – ложных целей. Результат обработки исходного изображения алгоритмом сегментации

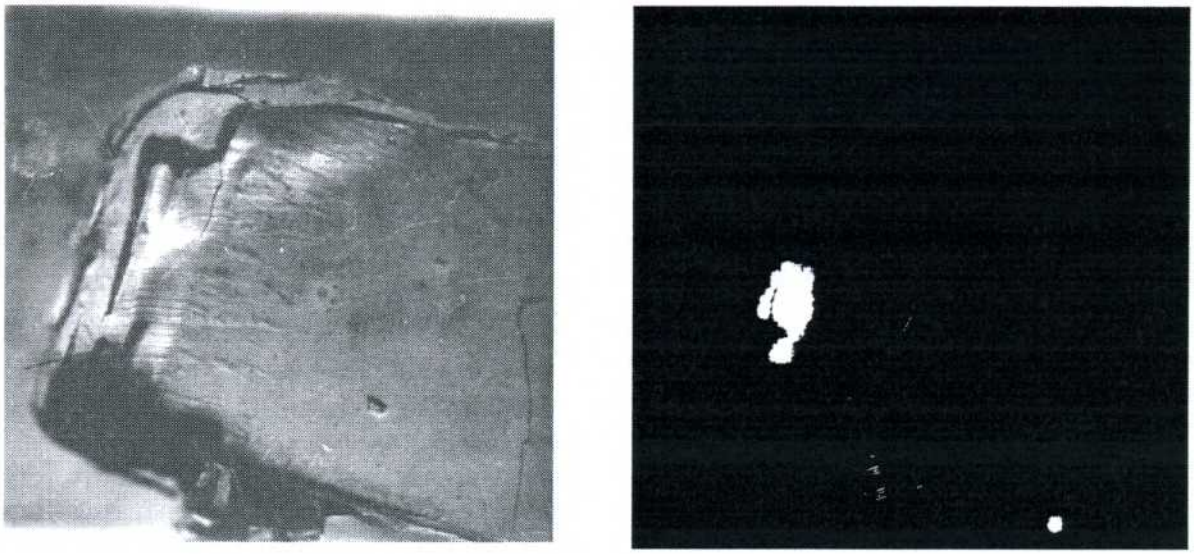


Рис.2. Исходное изображение с множеством ложных царапин и полученный для него бинарный препарат на выходе алгоритма сегментации



Рис.3. Совмещение исходного изображения и найденных предлагаемым алгоритмом анизотропных областей

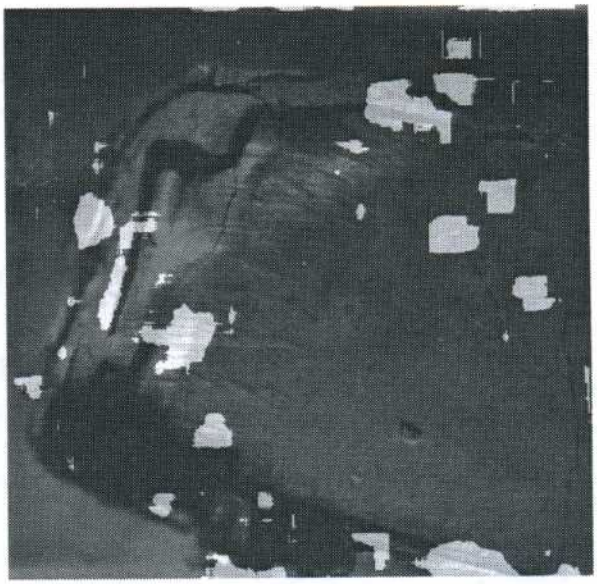


Рис.4. Совмещение исходного изображения и анизотропных областей, найденных алгоритмом из [11]

представлен на рис. 2 справа. Здесь белым обозначены области, распознанные алгоритмом как анизотропные, черным – изотропные. На рис. 3 для наглядности совмещены исходное изображение и найденные алгоритмом анизотропные области. Из рис. 3 видно, что алгоритм достаточно успешно справился с поставленной задачей: обнаружил искомую анизотропную область практически полностью, а также выдал лишь несколько ошибочных областей незначительной площади. Для сравнения на рис. 4 приведены результаты сегментации при помощи алгоритма из [10]. Параметры алгоритма [10] подбирались для получения максимально эффективного результата. Как видно из рис. 4, алгоритм [10] обнаружил лишь часть искомой анизотропной области, плюс к этому выдал множество ложных областей со значительной площадью. Таким образом, оказалось, что в ситуации, когда изображение достаточно сильно осложнено помехами, предлагаемый в данной статье алгоритм работает более эффективно, чем представленный в [10].

Таким образом, предложенный в данной статье алгоритм для сегментации анизотропных изображений в виде царапин (трасс) на некоторой поверхности обладает приемлемыми вычислительными требованиями, а также повышенной помехоустойчивостью по сравнению с алгоритмом, предложенным в [10].

СПИСОК ЛИТЕРАТУРЫ

1. **Прэтт У.** Цифровая обработка изображений: Пер. с англ. – М.: Мир, 1982. – Кн. 2. – 480 с.

2. **Новиков К.В.** Обнаружение анизотропных областей на изображениях с помощью преобразования Фурье. Материалы VI международной конференции "Актуальные проблемы электронного приборостроения" (АПЭП-2002). Новосибирск 2002.

3. **Abbie L. Warrick, Pamela A.** Delaney Detection of Linear Features using a Localized Radon Transform with a Wavelet Filter, In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'97) – Volume 4, 1997, P. 2769.

4. **D. V. Satish Chandra** Target Orientation Estimation Using Fourier Energy Spectrum, In: IEEE Transactions on Aerospace and Electronic Systems July 1998, v. 34, No. 3, P. 1009 – 1012.

5. **Tsitsipis P., Kontogeorgos A., Hillaris A., Moussas X., CaroubalosC., Preka-Papadema P.** Fast estimation of slopes of linear and quasi-linear structures in noisy background, using Fourier methods, In: Pattern Recognition 40 (2007), P. 563–577.

6. **Троицкий И.Н.** Статистическая теория томографии. – М.: Радио и связь, 1989. – 240 с.

7. Nima Bigdely Shamlo Matlab toolbox for high resolution vector field visualization with application in improving the u nderstanding of crack propagation mechanisms, In: A Thesis Presented to the Faculty of San Diego State University, San Diego, 2005, P. 52.

8. Методы компьютерной обработки изображений / Под ред. Сойфера В.А. – М.: Физматлит, 2003. – 784 с.

9. **Antony C. Copeland, Gopalan Ravichandran, Mohan M. Trivedi** Localized Radon Transform – Based Detection of Ship Wakes in SAR Images, In: IEEE Transactions on Geoscience and Remote Sensing, 1, January 1995, v.33, No. 1, P. 35–45.

10. **Грузман И.С., Новиков К.В.** Быстрый алгоритм сегментации анизотропных изображений на основе локальных спектральных моментов // Радиоэлектроника. – 2005. – № 3. – С. 50–56.

Суров А.Н., Игизьянова Н.А., Потапов В.И.

О СХОДИМОСТИ РЕШЕНИЯ РАЗНОСТНОЙ СХЕМЫ В МАТЕМАТИЧЕСКОЙ МОДЕЛИ ГЕНЕРАЦИИ ТЕПЛОВОЙ ЭНЕРГИИ

В данной работе проводится анализ устойчивости разностной схемы, которая гарантирует сходимость ее к решению исходной дифференциальной задачи.

При электрошлаковом наплавлении (ЭШП) сплошных или полых слитков большого диаметра появляется неоднородность температурного поля в шлаковой ванне. Эта неоднородность особенно существенна при наплавлении полых слитков большого диаметра. Если при наплавлении сплошных слитков шлаковая ванна достаточно компактна по объему и конвективные потоки относительно выравнивают температурное поле, тогда как при наплавлении полых слитков этого не происходит.

Формирование температурного поля происходит вследствие выделения тепловой энергии при прохождении электрического тока в жидком проводнике – шлаковой ванне с большим сопротивлением. В межэлектродном промежутке канала выделение энергии будет распределенное, так как распределенной будет сила тока. Если пренебречь теплообменом стенок канала с окружающими средами (стенками кристаллизатора, зеркалом жидкометаллической ванны, воздухом), то распределенное выделение энергии по длине канала и определит температурное поле. Можно утверждать, что каждой точке шлакового пространства будет соответствовать точечный источник энергии. Неоднородность температурного поля приводит к неравномерности оплавления торца расходуемого электрода, неравномерности теплообмена с окружающими средами. Поэтому изучение анизотропности температурного поля в шлаковом пространстве представляет как научный, так и практический интерес.

В данной работе рассматривается оценка анизотропии энергетического поля в шлаковом пространстве при ЭШП. При этом исходили из следующих допущений: так как при ЭШП энергия для переплава генерируется в шлаковой ванне электрическим током, то за основу были взяты фундаментальные законы электромагнитной динамики; теплообмен шлаковой ванны с окружающими средами, который искажает температурное поле, создаваемое источниками энергии, рассмотрен отдельно [1] и в данном случае не учитывается; теплообмен между каплями жидкого металла, стекающими с торца расходуемого электрода, и шлаком незначителен по сравнению с общей энергетикой процесса; процесс электродинамический считается осесимметричным, установившимся.

Исходя из принятых допущений, за основу были взяты уравнения Максвелла, адаптированные для данного процесса [2]. Уравнения, описывающие электромагнитные процессы в электроде, шлаковой ванне, имеют вид:

Поле потенциала в шлаковой ванне описывается уравнением Лапласа:

$$\frac{\partial^2 U}{\partial r^2}+\frac{1}{r}\frac{\partial U}{\partial r}+\frac{\partial^2 U}{\partial z^2}=0. \qquad (1)$$

Граничные условия имеют вид:
в области входа электрода в шлаковую ванну –

$$\{r,z: 0<r<r_э, z=d\}\ \frac{\partial \vec{U}}{\partial z}=-\frac{\vec{j}_э}{\sigma}; \qquad (2)$$

в области свободной поверхности шлаковой ванны –

$$\{r,z: r_э<r<R, z=d\}\ \frac{\partial U}{\partial z}=0; \qquad (3)$$

в области дна шлаковой ванны –

$$\{r, z: 0<r<r_э, z=c\}\ U=0;$$

$$\{r,z: r_э<r<R, z=c\}\ \frac{\partial U}{\partial z}=0; \qquad (4)$$

на боковой поверхности шлаковой ванны –

$$\{r,z: r=0, c<z<d\}\ \frac{\partial U}{\partial r}=0;$$

$$\{r,z: r=R, c<z<d\}\ \frac{\partial U}{\partial n}=0. \qquad (5)$$

Напряженность электрического поля $\vec{E}$ определяется из уравнений:

$$\vec{E}=-\nabla\vec{U},\ \vec{E}=\frac{\partial\vec{U}}{\partial n},\ \vec{E}_r=-\frac{\partial\vec{U}}{\partial r}, \qquad (6)$$

$$E_\varphi=-\frac{\partial U}{\partial z},\ E_Z=-\frac{\partial U}{\partial z},$$
$$\left|\vec{E}\right|=\sqrt{E_r^2+E_\varphi^2+E_Z^2}. \qquad (7)$$

Плотность электрического тока в шлаковой ванне вычисляется из соотношении:

$$\vec{j}=\sigma\vec{E}.$$

В уравнениях (1–7) приняты обозначения:

U – потенциал, b; r, z – координаты точек пространства шлаковой ванны, м; $\vec{j}_э$ – плотность тока на пятне электрода, кА/м^2; σ – удельная проводимость шлака, (Ом · м$^{-1}$); c, d – координаты границ шлаковой ванны; n – нормаль боковой поверхности ванны.

По известному вектору плотности электрического тока в каждой точке шлакового пространства вычисляется сила тока:

$$\vec{I}=\int_s \vec{j}ds.$$

Интегрирование распространяется на всё поперечное сечение s проводника (шлака). Плотность постоянного тока одинакова по всему сечению s проводника. Поэтому для постоянного тока.

$$\vec{I}=\vec{j}s.$$

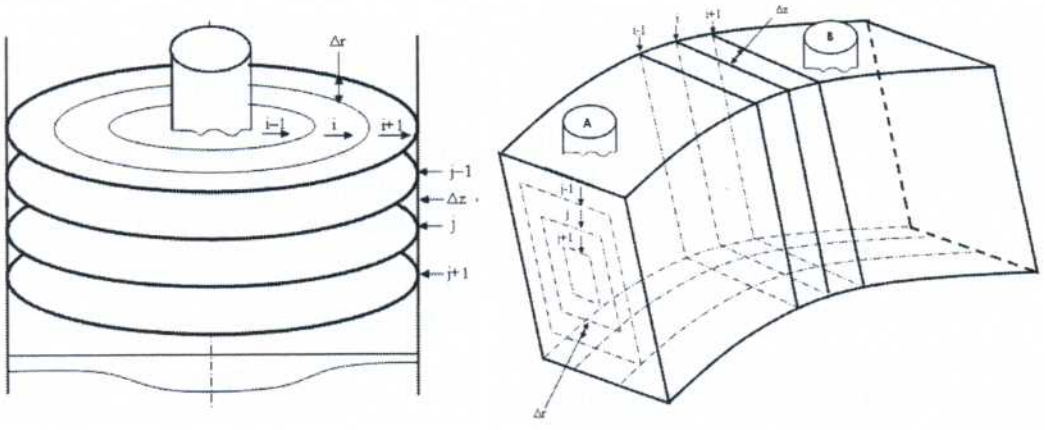
Решение уравнений (1–7) проводилось численным методом конечных разностей. Заменили частные производные в уравнениях (1–7) приближенными разностными производными в направлении переменных z, r, получили разностные уравнения. Схема дискретизации пространства шлаковой ванны в однофазной и бифилярной схемах включения печи ЭШП приведена на рис.1. Был использован шаблон разностного уравнения по явной схеме. Для вычисления значения разностного решения на $(i+1)$-м слое используются значения искомого решения на двух предыдущих слоях: i-м и $(i-1)$-м. При этом при $i=0$ и $i=1$ используются граничные условия (2–5).

В качестве сетки при дискретизации шлакового пространства D принята совокупность прямых $r=u\Delta r$, $z=mh$, где $u=1, 2, \ldots, k$ $m=0, 1, 2, \ldots, l$; $\Delta r>0$; $h>0$.

Производные в уравнениях (1–7) заменили разностными аналогами вида:

$$U_r=(2\Delta r)^{-1}[U(mh,(u+1)\Delta r)-U(mh,(u-1)\Delta r];$$
$$U_{rr}=\Delta r^{-2}[U(mh,(u+1)\Delta r)-$$
$$-2U(mh,u\Delta r)+U(mh,(u-1)\Delta r)];$$
$$U_z=(2h)^{-1}[U(m+1)h,u\Delta r)-U((m-1)h,u\Delta r)];$$
$$U_{zz}=h^{-2}[U((m+1)h,u\Delta r)-$$
$$-2U(mh,u\Delta r)+U((m-1)h,u\Delta r)].$$

При такой дискретизации шлакового пространства D шаг по r определяет цилиндр радиуса r_i, а шаг по z – слой высотой h. Для упрощения вычислительной схемы расчета приняли шлаковую ванну при бифилярном включении электродов в виде сектора тора.



а) б)

Рис. 1. Схема дискретизации пространства шлаковой ванны:
а – при однофазном включении цепи; б – при бифилярном

После дискретизации уравнений (1–7) была получена система алгебраических уравнений относительно неизвестных значений функций в узлах сетки. Если $U(mh, u\Delta r)$ обозначить U^u_m, то дифференциальное уравнение примет вид

$$\frac{U^{u+1}_m - 2U^u_m - U^{u-1}_m}{\Delta r^2} + \frac{U^{u+1}_m - U^{u-1}_m}{2u\Delta r^2} + \\ + \frac{U^u_{m+1} - 2U^u_m + U^u_{m-1}}{h^2} = 0.$$

Для оценки погрешности вычислений рассмотрим определение устойчивой разностной схемы из работы [3]. Пусть для приближенного вычисления решения u дифференциальной краевой задачи

$$Lu = f \qquad (8)$$

составлена разностная схема

$$L_h u^{(h)} = f^{(h)}, \qquad (9)$$

которая аппроксимирует задачу (8) на решении u с некоторым порядком h^k. Это значит, что невязка $\delta f^{(h)}$

$$L_h[u]_h = f^{(h)} + \delta f^{(h)},$$

возникающая при подстановке таблицы $[u]_h$ решения u в уравнение (9), удовлетворяет оценке вида

$$\|\delta f^{(h)}\|_{F_h} \le C_1 h_k, \qquad (10)$$

где C_1 – некоторая постоянная, не зависящая от h.

Будем называть разностную схему (9) устойчивой, если существуют члены $h_0 > 0$ и $\delta > 0$, такие, что при любом $h < h_0$ и любом $\varepsilon^{(h)} \in F_h$, $\|\varepsilon^{(h)}\|_{F_h} < \delta$ разностная задача

$$L_h z^{(h)} = f^{(h)} + \varepsilon^{(h)}, \qquad (11)$$

полученная из задачи (9) добавлением к правой части возмущения $\varepsilon^{(h)}$, имеет одно и только одно решение $z^{(h)}$, причем это решение отклоняется от решения $u^{(h)}$ невозмущенной задачи (9) на сеточную функцию $z^{(h)} - u^{(h)}$, удовлетворяющую оценке

$$\|z^{(h)} - u^{(h)}\|_{U_h} \le \|\varepsilon^{(h)}\|_{F_h}, \qquad (12)$$

где C – некоторая постоянная, не зависящая от h.

Теорема. Пусть разностная схема $L_h u^{(h)} = f^{(h)}$ аппроксимирует задачу $Lu = f$ на решении u с порядком h^k и устойчива. Тогда решение $u^{(h)}$ разностной задачи $L_h u^{(h)} = f^{(h)}$ сходится к $[u]_h$, причем имеет место оценка

$$\|[u]_h - u^{(h)}\|_{U_h} \le (CC_1)h^k, \qquad (13)$$

где C и C_1 – числа, входящие в оценки (10) и (12).

Оценим точность, достигнутую при решении дифференциального уравнения разностным методом по принципу Рунге.

Пусть нам известно, что порядок погрешности решения при помощи разностей шага h есть n, т.е. погрешность $\varepsilon_n(r, z)$ в какой-либо точке (r, z) может быть приближенно представлена так:

$$\varepsilon_h(r, z) \approx k(r, z)h^n,$$

где функция $k(r, z)$ от h не зависит.

Пусть мы имеем приближенные решения $U_{2h}(r, z)$ и $U_h(r, z)$ шага $2h$ и h. Тогда, обозначая точное решение через $u(r, z)$, имеем:

$$u(r, z) = U_{2h}(r, z) + \varepsilon_{2h}(r, z),$$
$$u(r, z) = U_h(r, z) + \varepsilon_h(r, z),$$

откуда, вычитая, получим:

$$U_h(r, z) - U_{2h}(r, z) = \varepsilon_{2h}(r, z) - \varepsilon_h(r, z);$$

и так как

$$\varepsilon_{2h}(r, z) \approx k(r, z)2^n h_n \approx 2^n \varepsilon_h(r, z),$$

то

$$U_h(r, z) - U_{2h}(r, z) \approx \varepsilon_h(r, z)[2^n - 1],$$

откуда

$$\varepsilon_h(r,z) \approx \frac{U_h(r,z) - U_{2h}(r,z)}{2^n - 1}.$$

Из этой формулы вытекает следующее правило для приближенной оценки погрешности решения с разностями шага h.

Следует взять в соответствующих точках разности решений при шаге h и при удвоенном шаге. Будучи поделены на $2^n - 1$, где n-й порядок погрешности, эти разности приблизительно дадут абсолютные погрешности решения шага h.

При решении уравнения Лапласа в квадратной сетке шага h с исправлением граничных значений по Коллатцу погрешность будет иметь порядок h^2 и, значит, будет приблизительно равна одной трети разности решений при шагах h и $2h$.

Рассмотрим разностное уравнение:

$$a_n u_{n-1} + b_n u_n + c_n u_{n+1} = f_n, \\ n = 1, 2, \ldots, N - 1, \qquad (14)$$

во внутренних точках $0 < n < N$ сеточного отрезка $0 \le n \le N$, где сеточная функция $\{u_n\}$ принимает заданные значения

$$u_0 = \varphi, \; u_N = \psi. \qquad (15)$$

Определение [3]. Разностная краевая задача (14), (15) с коэффициентами a_n, b_n, c_n, ограниченными в совокупности, $|a_n|, |b_n|, |c_n| < M$ хорошо обусловлена, если при всех достаточно больших N она имеет одно и только одно решение $\{u_n\}$ при произвольных правых частях φ, ψ и $\{f_n\}$ и если числа $u_0, u_1, \ldots u_N$, образующие решение, удовлетворяют оценке:

$$|u_n| \le M \max\left\{|\varphi|, |\psi|, \max_m |f_m|\right\}. \qquad (16)$$

где M – некоторое число, не зависящее от N.

Критерием хорошей обусловленности задачи с переменными коэффициентами, ограниченными в совокупности $|a_n|, |b_n|, |c_n| < M$ и ни при каком n одновременно не могут быть малыми, т.е.

$$d_n = \max\{|a_n|, |b_n|, |c_n|\} \ge B > 0,$$

где M и B не зависят от N и n, является теорема.

Теорема. Пусть коэффициенты задачи (14), (15) удовлетворяли условиям

$$\left.\begin{aligned} &|a_k - a_l| \le D\left|\frac{k-l}{N}\right|^{\omega}, |b_k - b_l| \le D\left|\frac{k-l}{N}\right|^{\omega}, \\ &|c_k - c_l| \le D\left|\frac{k-l}{N}\right|^{\omega}, D > 0, \omega > 0. \end{aligned}\right\} \qquad (17)$$

Тогда для хорошей обусловленности задачи (14), (15) необходимо и достаточно, чтобы корни q_1 и q_2 квадратного уравнения

$$a_n + b_n q + c_n q^2 = 0, \; 0 < n < N, \qquad (18)$$

удовлетворяют условию

$$|q_1| < 1 - \frac{\theta}{2}, \left|q_2^{-1}\right| < 1 - \frac{\theta}{2}, \qquad (19)$$

где $\theta > 0$ – некоторое число, не зависящее от N и n.

В области под электродом $0 < r < r_э$ из граничных условий $\frac{\partial \vec{U}}{\partial z} = -\frac{\vec{j}_э}{\sigma}$ следует $\frac{U^u_{m+1} - U^u_{m-1}}{2h} = -\frac{j}{\sigma}, j = \frac{I}{s}$, то $U^u_{m+1} = U^u_{m-1} - \frac{2Ih}{\sigma s}$, $U^u_m = U^u_{m-1} - \frac{Ih}{\sigma s}$. Тогда разностное уравнение примет вид:

$$\left(1 + \frac{1}{2u}\right)U^{u+1}_m - 2U^u_m + \left(1 - \frac{1}{2u}\right)U^{u-1}_m = \frac{3I\Delta r^2}{h\sigma s}. \qquad (20)$$

Проверим все условия для хорошей обусловленности задачи:

$$d_n = \max\left\{1 + \frac{1}{2u}; 2; 1 - \frac{1}{2u}\right\} > 0,$$

$$\forall k, l: \left|1 + \frac{1}{2k} - 1 - \frac{1}{2l}\right| = \left|\frac{1}{2k} - \frac{1}{2l}\right| \le D\left|\frac{k-l}{N}\right|^{\omega},$$

что соответствует (17). Для уравнения

$$\left(1 + \frac{1}{2u}\right)U^{u+1}_m - 2U^u_m + \left(1 - \frac{1}{2u}\right)U^{u-1}_m = 0$$

корни $q_{1,2} = 1 \pm \frac{1}{2u}$, т.е. удовлетворяют условию (19). Значит, критерий хорошей обусловленности выполняется и решение $\{U^n_m\}$ исходной задачи удовлетворяет оценке: [3]

$$\left|U^u_m\right| \le M \max\left(\left|\frac{I}{\sigma s}\right|, \max\left|\frac{3I\Delta r^2}{h\sigma s}\right|\right).$$

В результате компьютерного моделирования были получены значения в узлах сетки: потенциала U, плотности тока $\vec{j}$, силы тока I, энергии Q, выделяемой в шлаке током I, разность температур ΔT.

На рис. 2 приведены изолинии плотности тока в сечении шлаковой ванны. Расчетные данные показывают, что плотность тока в точках шлаковой ванны распределяется неравномерно: от значений, равных нулю, изолинии под номером 0 и до 10618 кА/м^2 на зеркале жидкометаллической ванны. Наибольшая плотность тока наблюдается при радиусе $r = r_{эл}$, $r_{эл}$ – радиус расходуемого электрода. Под электродом плотность тока более равномерная и при $r_{эл} < r < r_k$, при $z = m_i h$, $(i = \overline{5,10})$ плотность тока нулевая, где r_k – радиус кристаллизатора. В этой части пространства не наблюдается и выделение энергии, эта область есть самое холодное пространство шлаковой ванны. Конечно, в действительности происходит некоторое выравнивание выделенной энергии конвективными потоками жидкого шлака, теплопроводностью.

По найденным значениям плотности и силы тока определялась выделенная энергия в точках шлакового пространства по известному фено-

менологическому соотношению $Q = I^2R_ш \cdot t$, т.е. рассчитывались элементы энергетической матрицы $Q = [q_{ij}]$, $i = 1, 2, \ldots, k$, $j = 0, 1, 2, \ldots, m$, где q_{ij} – это величина энергии Дж, в узлах сетки с номерами i, j.

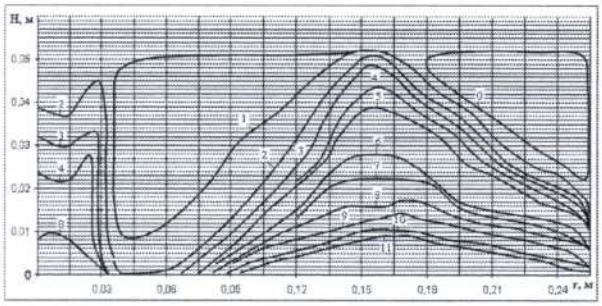


Рис. 2. Изолинии амплитуды плотности тока в шлаковой ванне: 0–0; 1–28,3; 2–30,5; 3–40,0; 4–50,0; 5–100,0; 6-150,0; 7–200,0; 8–300,0; 9–500,0; 10–2000,0; 11–12000,0 кА/м^2

Исходя из принятого допущения, что внешний теплообмен шлаковой ванны отсутствует, тогда внутри ванны распределение тепла происходит теплопроводностью. Из соотношения для теплового потока

$$Q_i = \lambda_{шл} \cdot S_i \frac{T_{i+1} - T_i}{\Delta r} \qquad (21)$$

от слоя к слою была вычислена разность температур $T_{i+1} - T_i = \Delta T$.

Теплопроводность шлака $\lambda_{шл} = 4{,}64$ Вт/(м · К) $r_э = 0{,}15$ м, $r_k = 0{,}26$ м. Площадь поперечного сечения i-го слоя вычислялась по формуле: $S_i = \Pi(r_{i+1} + r_i)\Delta r$. Шаг по r был принят равным 0,15 м, а по высоте ванны $h = 0{,}005$ м. Соотношение между шагами Δr, h удовлетворяет условию, при котором разностная схема устойчива, для гиперболического уравнения (1) и уравнений (2–7).

По формуле (21) была вычислена матрица температур $\Delta T = [\Delta t_{ij}]$ $i = 1, 2, \ldots, k$, $j = 1, 2, \ldots, m$. На рис.3 приведено изменение ΔT_j при $i = 5$, т.е. на середине высоты шлаковой ванны. Максимальное значение температуры наблюдается в области при $r = r_э$, и в этой периферийной зоне электрода выделяется максимум энергии и происходит перегрев шлака. При внешнем теплообмене этот пик перегрева сглаживается, но анизотропия температурного поля остается очень значительная. Замеры температуры проводились путем просвечивания ванны рентгеновским лучом и по степени затемнения пятна луча на пленке определяли температуру. При этом следует заметить, что луч, проходя по диаметру ванны, дает интегральную характеристику затемнения пятна на пленке.

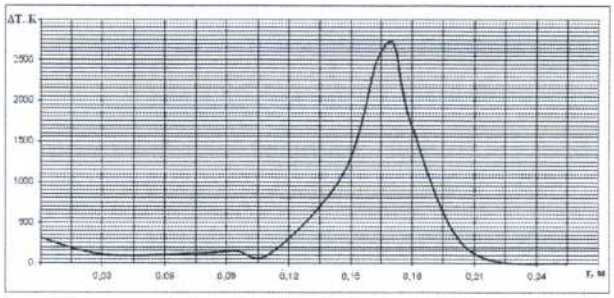


Рис. 3. Изменение разности температур между слоями шлаковой ванны на половине ее глубины, по радиусу

Вычисленная среднеинтегральная температура в сечении (рис. 3) по формуле:

$$\overline{\Delta T} = \frac{2}{r_k^2} \sum_{i=1}^{17} \Delta T_i r_i \Delta r = 341{,}1\,K.$$

При наличии внешнего теплообмена эта температура будет ниже, и что соответствует экспериментальным данным.

В результате исследований, проведенных в данной работе, можно сделать следующие выводы:

- получена математическая модель генерации тепловой энергии в шлаковой ванне;
- разработана методика расчета температурного поля в шлаковой ванне; подтверждено предположение о значительной анизотропии энергетического поля в шлаке;
- выявлено наличие зоны, где энергия не генерируется, и зоны перегрева шлака на периферии торца расходуемого электрода (это и приводит к оплавлению электрода в виде конуса).

Если в работе [1] в математических моделях теплофизических процессов при ЭШП источник энергии принимался в качестве граничного условия в среде для шлаковой ванны, то при наличии модели, полученной в данной работе, можно объединить электродинамику и теплофизику при ЭШП в одну математическую модель электрошлакового переплава.

СПИСОК ЛИТЕРАТУРЫ

1. **Суров, А.Н.** Расчет температурных полей в полых слитках при электрошлаковом переплаве. / А.Н. Суров, В.И. Потапов, М.С. Бугаев // Вестник ЮУрГУ, серия "Металлургия". – 2006. – Вып. 7, №10. – С. 73–75.

2. Электродинамические процессы при ЭШП на постоянном токе и их математическое моделирование / В.И. Потапов, Н.А. Игизьянова, И.В.Чуманов, Д.А. Пятыгин // Современные проблемы электрометаллургии стали: материалы XIII Междуна. Конф. / под ред. В.Е.Рощина. – Челябинск, 2007. – 4.3 – 216 с.

3. **Годунов С.К.** Разностные схемы (введение в теорию): учеб. пособие / С.К.Годунов, В.С.Рябенький. – М.: Наука, 1973.

Павлов Д. А.

Алгоритм построения расширенного универсального базиса Гребнера

Введение

Теория базисов Гребнера в настоящее время является основным алгоритмическим инструментом вычислительной коммутативной алгебры. Базисы Гребнера применяются для нахождения нормальных форм полиномов относительно идеала, решения систем полиномиальных уравнений, вычисления размерностей алгебраических многообразий и многих других задач.

Классическая теория базисов Гребнера основана на рассмотрении допустимых упорядочений мономов. То есть, любой базис Гребнера подразумевает некоторое конкретное допустимое упорядочение (или семейство упорядочений), на котором основан алгоритм редукции полиномов относительно идеала.

Однако, классическая теория может быть обобщена различными способами на некоторые другие упорядочения, не являющиеся допустимыми [1]. Мы рассматриваем т.н. слабодопустимые упорядочения, согласованные только с делением мономов. Некоторые слабодопустимые упорядочения позволяют определить базис идеала, который также может использоваться для вычисления нормальных форм полиномов. Конечные слабодопустимые упорядочения в точности соответствуют многомерным таблицам Юнга, а бесконечные — некоторым путям в графе Юнга [2]. Таким образом, комбинаторика мономиальных упорядочений тесно связана с комбинаторикой таблиц и диаграмм Юнга [3].

Обозначения

Обозначим M множество всех мономов от d переменных. Множество переменных обозначим $V = \{x_1, \ldots, x_d\}$. Мы будем рассматривать как тотальные, так и конечные упорядочения на множестве M.

Определение. Тотальное упорядочение $<$ называется слабодопустимым, когда оно согласовано с делением мономов:

$$m_1, m_2 \in M \quad m_1 | m_2 \Rightarrow m_1 < m_2.$$

Определение. Тотальное слабодопустимое упорядочение $<$ называется допустимым, когда оно согласовано с умножением:

$$m, m_1, m_2 \in M \quad m_1 < m_2 \Rightarrow mm_1 < mm_2.$$

Определение. Конечным слабодопустимым (допустимым) упорядочением степени n называется цепочка $M' = (m_1, \ldots, m_n)$, удовлетворяющая условию слабодопустимого (допустимого) упорядочения и содержащая с каждым мономом все его делители.

Обозначим за α некоторый идеал в бесконечномерном векторном пространстве полиномов $K[x_1,\ldots,x_d]$. Обозначим за A факторалгебру идеала α.

Множество мономов M лежит в $K[x_1,\ldots,x_d]$, таким образом, каждый моном $m \in M$ имеет канонический образ в A.

Редуцированный базис Гребнера идеала относительно некоторого мономиального упорядочения обозначим $G(\alpha)$.

Определение. Мы будем называть мономы линейно-независимыми относительно α,

если линейно-независимы их образы в его факторалгебре A. Здесь мономы рассматриваются как одночленные полиномы, а линейная независимость мономов означает линейную независимость их образов в факторалгебре $A = K[x_1,\ldots,x_d]/\alpha$.

Определение. Коидеалом в M называется множество мономов, которые не делятся ни на один моном из заданного конечного набора S (т.е. дополнение до возрастающего идеала в решетке мономов). Будем говорить, что коидеал построен на множестве мономов S. Коидеал, построенный на старших членах базиса Гребнера, порождает факторалгебру идеала. Коидеал может быть как конечным, там и бесконечным.

Определение. Идеал α называется нульмерным, если его факторалгебра A имеет конечную размерность, т.е. порождена конечным числом мономов.

Максимальные линейно-независимые мономиальные множества

Мы будем изучать максимальные линейно-независимые относительно идеала α множества мономов, которые являются коидеалами в M.

Определение. MLIMS(α) (Maximal Linearly-Independent Monomial Set) — это максимальный по включению коидеал в M, элементы которого линейно-независимы относительно α:

$$\dim(L(MLIMS(\alpha))\cap\alpha)=0,$$

где L(S) обозначает линейную оболочку множества S в векторном пространстве $K[x_1,\ldots,x_d]$. Очевидно, у идеала может иметься несколько MLIMS.

Утверждение. Коидеал C, построенный на старших членах какого-либо $G(\alpha)$, является MLIMS(α). Размерность пересечения L(C) с α равна 0, т.к. в L(C) нет полиномов из α (если бы в L(C) был полином из α, то C содержало бы его старший член, чего не может быть по определению базиса Гребнера). C является максимальным по включению, потому что мономы, которые можно добавить в C – это старшие члены полиномов базиса Гребнера. Поскольку все младшие члены полиномов базиса уже есть в C (по определению редуцированного базиса), то после добавления монома $L(C)$ содержит полином из базиса и $\dim(L(C)\cap\alpha)=1$.

Определение. Диаграммой Юнга базиса Гребнера нульмерного идеала называется множество мономов, являющихся делителями старших членов базиса (но не равные им).

Утверждение. Диаграмма Юнга базиса Гребнера нульмерного идеала α является его MLIMS(α). Это следует из того, что коидеал, построенный на старших членах базиса нульмерного идеала, конечен и является его диаграммой Юнга. Мономы, принадлежащие диаграмме, порождают факторалгебру идеала.

Утверждение. В произвольном MLIMS(α) нульмерного идеала α не больше dim(A) мономов.

В любую диаграмму Юнга всегда можно добавить моном и получить новую диаграмму Юнга. Как правило, добавить моном можно несколькими способами. Если MLIMS(α) является диаграммой Юнга (т.е. имеет конечный размер), то любое добавление монома образует множество мономов (m_1, ..., m_k), линейно-зависимых в идеале, причем есть лишь одно соотношение вида

$$p = f_1 m_1 + \ldots + f_k m_k,$$

такое, что полином $p \in \alpha$. Всего таких полиномов столько, сколько мономов можно добавить в MLIMS(α). Мы будем говорить, что MLIMS(α) образует набор полиномов (p_1, ..., p_s).

Определение. MLIMS(α) называется согласованным с полным упорядочением <, когда все его мономы меньше (в смысле упорядочения <) всех мономов, которые в нем не содержатся. Такое множество мы будем обозначать MLIMS(α, <).

Алгоритм FGLM [4], разработанный в 1994 году, делает пересчет базиса Гребнера нульмерного идеала из одного допустимого упорядочения мономов в другое допустимое. В этом алгоритме используется тот факт, что набор всех полиномов, образованных MLIMS(α, <) при допустимом < (т.е. согласованным с умножением на моном), является базисом Гребнера идеала. Вообще говоря, для нульмерных идеалов возможно, построение MLIMS(α, <) для всех конечных упорядочений, и вычисление соответствующих базисов [2]. Ансамбль всех базисов называется веером Гребнера Мора-Роббиано (Grubner fan [5]), и для его эффективного построения есть специализированный алгоритм (Grubner walk), не связанный с диаграммами Юнга и реализованный в пакете GFan [6].

Расширенный универсальный базис Гребнера

Мы рассматриваем базисы полиномиальных идеалов, соответствующие не только допус-

тимым упорядочениям мономов, но принимаем в расчет и слабодопустимые упорядочения, согласованные только с делением мономов. Свойство линейной независимости MLIMS(α, <) как раз является естественным условием для образования базиса, по которому можно вычислять нормальные формы полиномов. При наращивании MLIMS(α, <) любым мономом получается множество мономов, линейно зависимых в α. Некоторая их линейная комбинация образует полином, принадлежащий α. При допустимом упорядочении < набор всех таких полиномов является базисом Гребнера, а MLIMS(α, <) порождает A. Если взять более общий случай слабодопустимого <, то полученное множество полиномов может не порождать исходный идеал. Такие множества полиномов мы будем называть псевдо-базисами. Кроме того, при слабодопустимом < может оказаться |MLIMS(α, <)| < dim(A). Однако даже в случае |MLIMS(α, <)| = dim(A) можно показать, что среди полученных псевдо-базисов могут присутствовать базисы-идеала, отсутствующие в веере Гребнера Мора-Роббиано [2].

Назовем такие базисы обобщенными базисами Гребнера (обычные базисы Гребнера присоединим к ним тоже), а объединение всех таких базисов — расширенным универсальным базисом Гребнера (Extended Universal Grıbner Basis, EUGB). Очевидно, что "обычный" UGB(α) Мора-Робиано содержится в EUGB(α), т.к. любое допустимое упорядочение < по определению является слабодопустимым. Можно показать [1], что EUGB(α) состоит из конечного числа полиномов.

Алгоритм построения EUGB

Мы продемонстрируем алгоритм построения EUGB как множества полиномов, а не как объединения базисов. Необходимое условие принадлежности некоторого полинома к EUGB можно сформулировать в терминах размерности пересечения идеала и некоторого линейного подпространства в кольце полиномов [1].

Определение. Диаграмма Юнга $Y(p)$ полинома p – это множество мономов, которые являются делителями термов p (включая их самих).

Утверждение. Для полиномов p из обобщенных базисов и только для них выполняется условие $\dim(L(Y) \cap \alpha) = 1$, при этом $Y(p)$ является минимальной по включению диаграммой, соблюдающей это условие то есть уменьшение $Y(p)$ на один моном образует множество мономов, линейно-независимых в α.

Отличие этого условия от MLIMS в том, что оно описывает минимальные по включению, а не максимальные по включению диаграммы. Если мы возьмем какую-нибудь большую диаграмму Юнга Y и вычислим $\dim(L(Y) \cap \alpha) > 1$, то точно будем знать, что из Y можно вычеркнуть как минимум $\dim(L(Y) \cap \alpha) - 1$ мономов. Какой-то результат в любом случае получится. Из диаграммы Юнга с помощью гауссовой элиминации нетрудно получить полином, являющийся линейной комбинацией ее мономов-неделителей и принадлежащий идеалу. Это и будет первый полином EUGB.

Получить последующие полиномы этой процедурой уже не так просто. Требуется модификация, предотвращающая повторение найденных полиномов. Однако мы рассмотрим модификацию, предотвращающую повторение найденных диаграмм. Она не спасает от повторения полиномов, однако гарантирует, что они будут найдены все и за конечное время а дубликаты можно тривиально удалить.

Пусть на каком-то этапе работы алгоритмом найдено K диаграмм Юнга. При нахождении следующей надо сделать так, чтобы она не совпадала ни с одной из K предыдущих. Это значит, что она не содержит и не содержится ни в одной из них, т.к. искомые диаграммы минимальны по включению. Чтобы исключить совпадение искомой диаграммы с найденной, достаточно из первоначальной "большой" диаграммы удалить один из мономов-неделителей найденной диаграммы, а также все кратные ему (чтобы диаграмма осталась таковой). Чтобы исключить совпадение искомой диаграммы с K найденными, достаточно удалить K мономов-неделителей найденных диаграмм по одному из каждой и все кратные им.

После удаления может оказаться, что мономы в диаграмме уже линейно-независимы, и получить из нее диаграмму с $\dim(L(Y) \cap \alpha) = 1$ невозможно. В таком случае надо выбрать другую комбинацию K мономов, удаляемых из начальной диаграммы.

Утверждение. Если ни одна комбинация K мономов не привела к образованию диаграммы с линейно-зависимыми мономами, то значит других диаграмм, отвечающих заданным условиям, нет.

Если же мономы в диаграмме оказались линейно-зависимы (скажем $\dim(L(Y) \cap \alpha) =$

$u > 0$), то из нее следует удалить u-1 моном и вычислить $\dim(L(Y) \cap \alpha)$ снова. Когда u станет равно 1, надо по одному удалять мономы из диаграммы, пока не получится минимальная по включению, т.е. когда любое уменьшение диаграммы приведет к тому, что мономы станут линейно-независимыми.

Особое внимание стоит обратить на то, какой должна быть начальная "большая" диаграмма Юнга. Чтобы алгоритм находил все диаграммы Юнга, отвечающие условиям, начальная диаграмма Юнга содержать их всех в себе. Проще говоря, она должна содержать в себе все диаграммы Юнга размера $\dim(A)$, где $\dim(A)$ – размерность факторалгебры идеала. Это то же самое, что диаграмма, содержащая все d-мерные "параллелепипеды" размера не более $\dim(A)$. Следует отметить, что для ненульмерных идеалов, размерность факторалгебры которых бесконечна, такой подход работать не будет. Вместо этого необходимо иметь какую-либо оценку на максимальный размер произвольного базиса Гребнера рассматриваемого идеала.

Вычисление псевдобазисов по полиномам EUGB

Среди множества полиномов EUGB, найденных методом из предыдущего раздела, нас интересуют подмножества полиномов, которые порождают идеал α. Для каждого такого подмножества $\{p_1,\ldots, p_n\}$ необходимо построить MLIMS(α), который можно нарастить n способами, и получить в результате эти полиномы $\{p_1,\ldots,p_n\}$. Если такой MLIMS(α) удастся найти, значит $\{p_1,\ldots, p_n\}$ составляют псевдо- базис. Если при этом MLIMS(α)| = $\dim(A)$, то $\{p_1,\ldots, p_n\}$ образуют обобщеный базис.

Нахождение MLIMS(α) начинается с того, что в полиномах $\{p_1,\ldots,p_n\}$ выделяются старшие члены, так чтобы ни один старший член ни одного p_i не делил ни одного терма, ни одного p_i, кроме самого себя. Если есть несколько способов выделения старших членов, то перебираются все.

Если после выделения старших членов оказалось, что набор $\{p_1,\ldots, p_n\}$ нарушает условия редуцированного базиса (т.е. если какой-то член одного полинома делится на старший член другого полинома), то этот набор убирается из рассмотрения, т.к. он не может быть обобщенным базисом.

На выделенных старших членах строится коидеал. Поскольку мы знаем, что MLIMS(α) конечен и в нем не больше $\dim(A)$ мономов, то мы исключаем те случаи, когда коидеал бесконечен или в нем больше $\dim(A)$ мономов. В противном случае найденный коидеал – кандидат на MLIMS(α). Последняя операция – самая дорогостоящая – проверка линейной независимости мономов найденного коидеала. Если мономы линейно независимы, значит мы нашли MLIMS(α) и соответствующий ему обобщенный базис идеала.

Реализация

Описанный алгоритм был реализован на языке Common Lisp и запущен на ОС Debian GNU/Linux на процессоре Intel Core 2 Duo с использованием компилятора SBCL. Для вычисления стандартного веера Гребнера использовался пакет GFan [6]. В данной реализации применялась достаточно прямолинейная стратегия отсечения мономов из рассматриваемых диаграмм, поэтому имеются резервы для существенного повышения эффективности работы программ за счет применения более оптимальных стратегий отсечения. Также в дальнейшем предполагается оптимизировать алгоритм построения базисов из полиномов, принадлежащих EUGB.

Пример

Рассмотрим идеал α, порожденный набором полиномов $\{x^5 + y^5, x^2y^3 - y^5, xy^5 + y^6, y^7 + y^5\}$. Его веер Гребнера состоит только из двух ячеек :

$\{x^5 + y^5, x^2y^3 - y^5, xy^5 + y^6, y^7 + y^5\}$,

$\{x^7 + x^5, x^5y + x^6, x^2y^3 + x^5, y^5 + x^5\}$.

Размерность факторалгебры этого идеала равна 21. Наша процедура нашла еще два обобщенных базиса, которые также могут использоваться для вычисления нормальных форм:

$\{x^5 + x^2y^3, x^2y^4 + x^3y^3, y^5 - x^2y^3, x^4y^3 + x^2y^3\}$

$\{x^6 + x^5y, y^5 + x^5, x^2y^3 + x^5, x^5y^2 + x^5\}$.

Расширенный универсальный базис EUGB(α) в этом примере состоит из 10 полиномов,

$\{x^5 + y^5, x^2y^3 - y^5, xy^5 + y^6, y^7 + y^5, x^7 + x^5, x^5y + x^6, x^2y^3 + x^5, x^2y^4 + x^3y^3, x^4y^3 + x^2y^3, x^5y^2 + x^5\}$,

три из которых не принадлежат вееру Гребнера и не могут быть построены пакетом Gfan.

Заключение

Описан метод построения расширенных универсальных базисов Гребнера полиномиальных идеалов. Метод не требует привлечения допустимых упорядочений мономов, а исполь-

зует только естественный частичный порядок, основанный на делимости. В дальнейшем планируется обобщить этот метод с нульмерных идеалов на произвольные, а также сделать ряд оптимизаций производительности.

Благодарности

Автор благодарит Н.Н. Васильева за большое внимание к проделанной работе и множество ценных советов и находок.

СПИСОК ЛИТЕРАТУРЫ

1. **Vasiliev, N.**: Monomial Orderings, Young Diagrams and Grцbner Bases. Proceedings of the International Conference "Computer Algebra in Scientific Computing" (CASC). Technical University of Munchen (2003)

2. **Vasiliev, N., Pavlov, D.**: Enumeration of finite monomial orderings and Young tableaux. In "Polynomial Computer Algebra" Proceedings of the International Conference PCA'08, pp. 71–76, St. Petersburg, 2008

3. **Stanley, R.P.**: Enumerative combinatorics vol. 1 and 2. Cambridge University Press (1997 and 1999)

4. **Faugire J. C., Gianni P., Lazard D., Mora T.** Efficient computation of zero-dimensional Grцbner bases by change of ordering. Journal of Symbolic Computation 16, Issue 4 (1993) 329–344

5. **Mora T., Robbiano L.** The Grцbner fan of an ideal. Journal of Symbolic Computation, 6, 2–3 (988) 183–208

6. **Jensen, Anders N.** Gfan, a software system for Grцbner fans. http://www.math.tu-berlin.de/~jensen/software/gfan/gfan.html

Манцеров Д. И.

Система верификации для параметрических классов задач по математике

В настоящее время актуальным является процесс компьютеризации школ. Во всех школах, начиная с небольших деревенских и заканчивая большими специализированными городскими школами, создаются компьютерные классы. Идея создания таких классов – научить учеников работать с компьютером. В связи с этим актуальным становится создания специального программного обеспечения(ПО) для школ. В этой статье будет описана система, которая позволяет ученику решать предложенные математические задачи, совершенствовать свои знания в математике. Все это ученик может проделывать с помощью компьютера и без помощи своего учителя.

Система верификации параметрических классов задач включает в себя как средства для создания задач, так и средства для проверки решений предложенных задач на множестве базовых примеров и организации обучающего диалога.

Система построена на параметрических классах объектов и включает в себя набор базовых примеров этих объектов, демонстрирующих основные особенности этих объектов, которые входят в школьный курс математики. По техническим соображениям набор примеров подбирается так, чтобы, не перегружая программу, обеспечить верификацию ответов достаточную, чтобы подтолкнуть решающего к собственной идее решения, увидеть в частичном решении противоречия или недочеты. Среда по своей сути не является ни средством обучения, ни средством контроля. Ее роль – поддержка самостоятельной работы ученика в процессе решения задачи.

Примерами классов объектов могут служить, например, изучаемые в школе классы функций: линейные функции $y = kx + b$, логарифмические функции $y = \log_a(bx + c)$ и т.д.

Система верификации построена на следующих *методических* принципах.

• Система допускает несколько форм представления математических объектов[3]: возможно использование формульного, графического и вербального представлений объектов. Это дает возможность использовать в качестве элемента решения задачи переход от одной формы

представления объекта к другой и создает основу для генерирования различных видов задач на одном параметрическом классе.

• Система обрабатывает любые синтаксически правильные ответы ученика, в том числе, когда они являются лишь частичными решениями. Ученик может посмотреть несколько примеров, демонстрирующих неполноту или противоречивость ответа.

• Система не ограничивает ученика в форме записи ответа (предиката, описывающего множество решений), благодаря чему ученику предоставляется возможность найти собственное решение задачи, без "подгонки" под эталонную форму записи.

Система верификации построена на следующих *технологических* принципах.

• Система верификации создана на языке программирования JAVA. Применение этого языка позволило реализовать среду как в виде апплета, так и в виде программного приложения.

• Для хранения предикатов, примеров и задач в системе верификации используется формат XML.

• После создания в среде учителя задача является независимым объектом и может быть использована любой системой ученика, интерпретирующей внутренний формат задачи.

Для реализации системы использована компонентная модель организации программных сред [5]. К известным достоинствам компонентной модели организации программных сред для решения задачи – создания программного обеспечения (ПО) педагогического назначения – добавляются следующие факторы:

• разработка специального ПО предполагает его открытость и легкость модификации, в том числе регулярное изменение старых и добавление новых модулей (создание педагогического программного обеспечения, как правило, не может начаться с точной спецификации и представляет собой диалог с учителями математики, предполагающий непрерывную коррекцию программы в процессе ее создания);

• компоненты программы связаны с определенными сущностями из предметной области "методика преподавания математики"[2, 3], в данном случае это формы представления математических знаний; сопоставления отдельных компонентов программы этим сущностям позволяет существенно расширить функциональность создаваемого ПО;

• использование компонентной модели[6] предоставляет большую свободу в конструировании интерфейса, что является наиболее важным в создании ПО педагогического назначения.

Структура системы верификации

Система верификации состоит из трех независимых компонентов: оболочки ученика, оболочки учителя, виртуальной клавиатуры; четырех вспомогательных компонент: базы данных параметрических классов объектов, базы данных примеров, базы данных задач. Общая структура среды представлена на рисунке 1. Основу среды составляет две компоненты: оболочка ученика, которая необходима для решения задач, и оболочка учителя, которая используется для создания задач и параметрических классов задач.

Отдельно выделены компоненты: виртуальная клавиатура учителя и виртуальная клавиатура ученика.

Оболочка ученика – это компонента, которая позволяет ученику быстро и удобно вводить ответ на предложенные задачи.

Главной особенностью компоненты ученика является механизм анализа ответа ученика.



Рис. 1. Структура среды

Ответ сравнивается с эталонным на множестве примеров, связанных с данным параметрическим объектом. Отличия введенного ответа от эталонного демонстрируется на примере, который, как правило, демонстрируется в нескольких формах (графической, формульной) и сопровождается вербальным комментарием.

В интерфейсе используются различные визуальные ассоциации, так, примеры, которые удовлетворяют введенному ответу, но не удовлетворяют условию задачи, помечаются красным цветом; примеры, которые удовлетворяют условию задачи, но не удовлетворяют введенному условию, помечаются желтым цветом; примеры, удовлетворяющие обоим условиям, помечаются зеленым цветом.

Оболочка учителя предназначена для создания новых и редактирования существующих параметрических классов, базы примеров и базы предикатов для параметрических классов задач и собственно задач (для оболочки ученика). Основу любой задачи составляет параметрический класс. Параметрический класс описывается параметрами, примерами и предикатами. Они определяют конструируемый объект, его свойства объекта и предоставляют достаточную информацию для его исследования.

Примеры строятся посредством конкретизации параметров параметрического класса. Свойства параметрического класса вводятся как предикаты от его параметров, причем этим предикатам должен быть присвоен содержательный смысл (например, "функция возрастает", "уравнение не имеет действительных корней") В дальнейшем эти объекты используются для создания задач (по существу, элементарные предикаты уже являются примерами таких задач). Для редактирования параметрических классов, существующих в системе баз и задач, в оболочке учителя используется виртуальная клавиатура учителя.

Физически оболочки ученика и учителя могут быть установлены и использоваться как на одном компьютере, так и на разных.

Файл описания параметрического класса представляется в XML-формате и является элементом базы параметрических классов. Каждый класс включает в себя все необходимые данные, которые вводятся через форму, изображенную на рисунке 2. Файл с описаниями параметрического класса создается и редактируется только в оболочке учителя.

База примеров – это база значений параметров, представленных в виде xml-файлов. Каждый xml-файл содержит значения параметров для одного параметрического класса задач. Все действия по созданию новых баз примеров и редактированию существующих баз осуществляется только через оболочку учителя.

База предикатов – предикаты представляются в виде xml-файлов. Каждый xml-файл содержит предикаты одного параметрического класса. Предикаты задают свойства объектов определенного параметрического класса задач. Базу предикатов так же, как и базу примеров, можно создавать и редактировать только через оболочку учителя.

Виртуальная клавиатура используется для имитации компьютерной клавиатуры в оболочках учителя и ученика, расширенной параметрами класса и необходимыми математическими знаками. Данный компонент представляет собой объединение нескольких панелей с кнопками: панель функций, панель цифр, панель с операциями отношений, панель с логическими операциями и панель с параметрами параметрического класса задач. Каждому параметрическому классу и задачам, созданным на его основе, соответствует своя виртуальная клавиатура. Отдельные виртуальные клавиатуры отличаются друг от друга только составом кнопок на панели параметров. Хотя часть символов можно вводить и с помощью обычной клавиатуры, есть математические символы, которым на обычной клавиатуре кнопок не соответствует (например, "больше или равно"), в этом случае можно использовать только виртуальную клавиатуру. На рис. 2, 3, 4 изображены виртуальные клавиатуры в оболочках учителя и ученика.

База задач – это база созданных задач на основе всех параметрических классов. Каждая задача представляет собой самодостаточный объект, который используется в оболочке ученика для обеспечения интерактивности его работы в процессе решения задачи. Как видно из рис. 1, база данных задач создается и редактируется только в оболочке учителя.

Создание нового параметрического класса задач

Основным объектом среды верификации является параметрический класс. Каждый параметрический класс является фреймом (класс-шаблоном), определяющим в конечном счете особенности построенного на нем множества задач.

В данной системе параметрические классы задаются следующими элементами:

• *общая формула функции, или параметрическое описание математического объекта*, это символьное представление объекта, которое содержит параметры и независимые переменные;

• *параметры формулы функции или математического объекта;*

• *предикат функции или математического объекта* – условие, наложенное на параметры класса (определяющая область допустимых значений, на котором будет идти верификация; например, для $y = \log_a(bx + c)$ это условие $a > 1 \ \& \ a \neq 1$.

На рис. 2 представлена форма для создания нового параметрического класса задач. Формулы и параметры класса перечисляются через запятую. Кроме полей для основных трех элементов описания математических объектов, есть два дополнительных поля: поле имени класса и поле для ввода независимых переменных (на данный момент система работает с одной независимой переменной).

Название класса будет использовано для дальнейшей работы с классом и базами данного класса.

Заметим, что переход от школьной математики к другой предметной области будет заключаться в изменении представления класс-шаблона. Описание параметрического класса не обязательно будет представлено в виде функции. Это может быть, к примеру, конечное множество объектов с параметрами (например, таблица Менделеева в химии).

После создания нового класса задач создается базовое множество примеров.

Наполнение базы примеров

В основе идеи использования примеров в нашей системе верификации лежит идея Патрика Уинстона [Winston, 1975a] о машинном обучении на основе примеров и контрпримеров. Оказывается, что при правильном чередовании сообщений самообучающейся программе о том, обладает ли данный пример некоторым свойством или нет, данной информации достаточно, чтобы программа быстро смогла "овладеть" сутью того или иного понятия. Такой метод обучения является также составной частью методики обучения математике.

При формировании множества примеров важно, чтобы в базе примеров появились как примеры, удовлетворяющие эталонному предикату (положительные примеры), так и примеры, ему не удовлетворяющие (отрицательные примеры).

Для подбора примеров в системе используется следующий метод: сначала формируется равномерная сетка примеров путем выбора диапазона изменения параметра и шага изменения, затем пишется условие, по которому из всех этих примеров отбираются нужные (отрицательные и положительные отдельно).

На рис. 3 изображен модуль для генерации объектов по заданному условию.

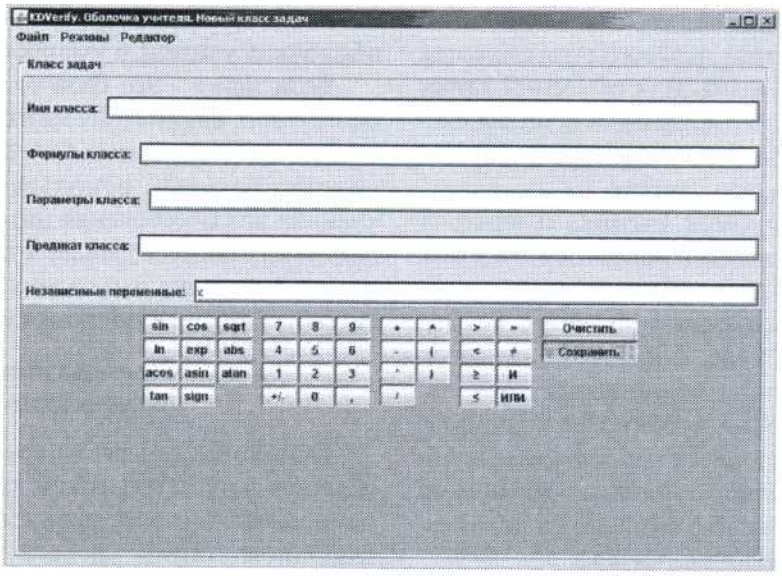
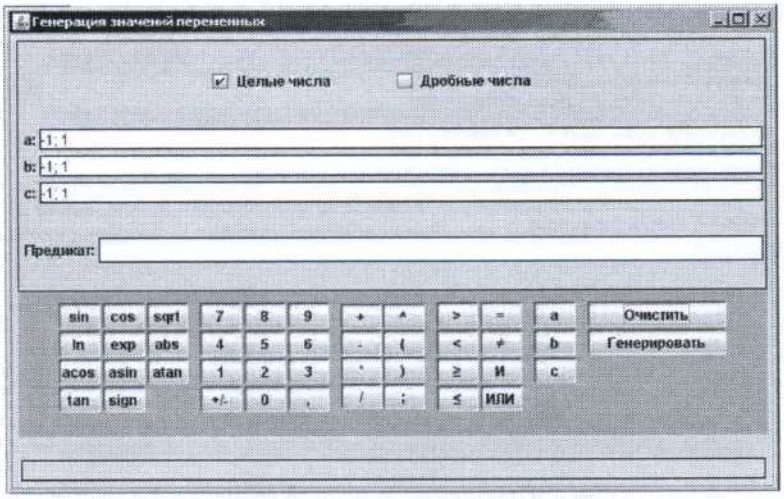


Рис. 2. Форма для нового параметрического класса задач



Ри. 3. Генератор примеров

Рассмотрим на примере создание объектов параметрического класса задач. На рисунке 3 представлен модуль, разработанный автором для создания объектов. В качестве примера использован параметрический класс, формула которого имеет вид $f(x) = kx + b$ – линейная функция. После ввода интервалов для параметров k и b и предиката на данные параметры происходит автоматическое создание примеров. Для демонстрации создания положительных и отрицательных примеров рассмотрим следующую задачу. При каком условии на параметры k и b линейная функция возрастает? Для данной задачи положительными примерами являются те объекты, для которых неравенство $k > 0$ приминает истинное значение. Остальные объекты будут относиться к отрицательным примерам.

База предикатов

Как было отмечено выше, на параметрах класс-шаблона можно определять предикаты, описывающие свойства математического объекта, заданного класс-шаблоном. Все предикаты помещаются в базу предикатов. Множества предикатов разных параметрических классов задач не пересекаются, даже если они имеют одинаковый набор параметров.

Доступ к базе предикатов класс-шаблона предоставляется только после его создания. После того как создан класс-шаблон, переходим к созданию базы предикатов, которая изначально пуста. Чтобы добавить предикат параметрического класса, необходимо перейти в редактор предикатов. В редакторе задается сам предикат и его название. Каждый предикат класс-шаблона строится из имен параметров, логических операций, отношений (численные значения параметров не допускаются).

Название предиката необходимо для удобства его использования. Например, выберем класс-шаблон, формула которого имеет вид $f(x) = ax^2 + bx + c, a \neq 0$ – квадратичная функция общего вида, – и рассмотрим предикат $c - \frac{b^2}{4a} > 0$. С первого взгляда трудно сказать, какое свойство задает данный предикат. Здесь будет полезно название предиката – "вершина параболы лежит внутри верхней полуплоскости", (это одно из возможных названий нашего предиката). Однако более важная функция имени предиката в том, что появляется возможность предоставить ученику возможность работать не только с параметрами класса, составляя из них предикаты решений, но и с элементарными предикатами – свойствами объекта, из которых ученик также может строить логические выражения.

Из рис. 1 видно, что изменения в базе предикатов никак не отразятся на содержании базы примеров. Обратное также верно.

База примеров и предикатов используются всегда вместе.

На рис. 4 представлено окно с базой предикатов для линейной функции вида $y = kx + b$.

Создание задачи

После того как создан параметрический класс задач базы примеров и предикатов, можно переходить к созданию задач для оболочки ученика. Для этого в оболочке учителя

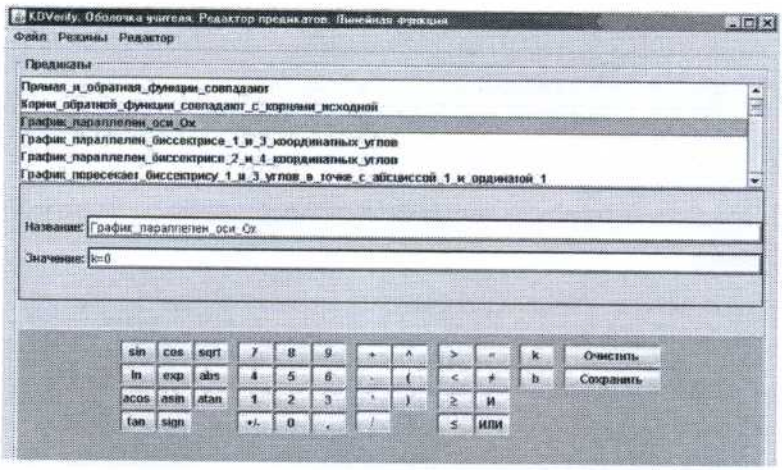


Рис. 4. Предикаты линейной функции

открываем компоненту для создания задач на основе данного параметрического класса задач. Задача будет представлена в виде файла XML-формата, в котором будет содержаться информация о наборе базовых примеров, предикатах, которые используются в качестве свойств на виртуальной клавиатуре (если такие есть), предикат эталонного ответа и условие задачи. Эталонный ответ может быть представлен в виде логического выражения, содержащего как параметры, так и предикаты параметрического класса. После того как будет создана задача, ее можно передавать в оболочку ученика.

Поддержка решения задачи в оболочке ученика

В оболочке ученика среды верификации реализована идея Пойа [1] об исследовательском подходе к математической задаче. В его методике большое значение придается самостоятельной деятельности по открытию и исследованию математических закономерностей. Процесс решения задачи подробно описан нами в статье [4].

Заключение

В проведенном исследовании используется созданная автором программная система для поддержки исследовательского подхода к обучению математике, развитого в работах Д. Пойа. Для технической реализации использовалась идея обучения на примерах и контрпримерах, реализация которой для машинного самообучения описана в работах П. Уинстона.

В результате проведенного исследования выявлена структура среды, которая позволяет реализовать предложенный подход. В этой структуре выделены следующие компоненты: описание параметрического математического объекта, являющегося основой для постановки исследовательских задач, множество предикатов, описывающих свойства объекта, множество примеров и контрпримеров, позволяющих разделять свойства объекта, задачи с параметрами на исследование математических объектов.

На основе теоретического анализа создана среда верификации, которая была апробирована в ходе создания инновационных учебно-методических комплексов и информационных источников сложной структуры в 2006–2008 годах.

СПИСОК ЛИТЕРАТУРЫ

1. **Дж. Пойа** Математическое открытие. – Москва, Главная редакция физико-математической литературы. Издание второе, стереотипное. Перевод В.С. Бермана. Издательство "Книга", 1976.

2. **Pozdnyakov S.** Computer technologies and productive learning in mathematics. 9th International Congress on Mathematical Education (ICME9) on behalf of the International Commission on Mathematical Instruction (ICMI).Tokyo/Makuhari, Japan, from July 31 to August 6, 2000.

3. **Pozdnyakov S.** Mathematical models of intellectual dialogue in Computer Aided Learning. World Conference on Twenty-First Century Mathematics. Proseedings. School of Mathematical Sciences, GC University, Lahore, Punjab (Pakistan), 2005.

4. **Манцеров Д.И.** Среда Verifier-KD: верификация решений задач по математике. "Компьютерные инструменты в образовании". – Санкт-Петербург, Автономная некоммерческая организация "Центр Информатизации Образования "КИО", 2006. – С. 36–41

5. http://www.osp.ru/os/1999/04/179785/_p1.html: Надежда Куцевич. Компонентные технологии в системах промышленной автоматизации

6. The JavaBeans 1.01 specification. - Sun Microsystems, July, 1997.

Канунникова Е.А., Колтунов Л.И., Потапенко А.Н.

Численное моделирование распределенных электротехнических систем в полубезграничных областях на основе метода инверсии

Введение

В прикладных исследованиях электротехнических систем встречается необходимость расчета электрических полей (ЭП) для высоковольтных *линий электропередач* (ЛЭП) переменного и *постоянного тока* [1, 2], для систем молниезащит [3], для высоковольтных электростатических систем типа однозонных и двухзонных электрофильтров в системах очистки газа [4, 5], *распределенных систем ионизации воздуха* типа модификации "Аэроион-Про" [6] *и др.*

В исследованиях электрических полей *ЛЭП* одним из основных показателей, определяющих степень влияния *высоковольтных ЛЭП* на биологические объекты, является уровень напряженности ЭП вблизи поверхности земли, для систем молниезащит – особенности распределения ЭП относительно стержневых молниеотводов, для *распределенных систем ионизации воздуха* необходимо в окружающем пространстве определять распределения потенциала и напряженности поля, так как на коронирующие электроды подается потенциал высокого напряжения и др. Задачи расчета ЭП в распределенных электротехнических системах типа [1–3] *относятся к внешним краевым задачам в полубезграничных областях, также к ним можно отнести и исследования распределенных систем ионизации воздуха* [6] с учетом соотношений размеров, например для спортивных залов:

$$H_l / H_s \le 1000, \qquad H_s / d \le 100,$$

где H_l – расстояние между противоположными стенами зала; H_s – расстояние от коронирующего провода до стены; d – диаметр провода.

Решения внешних краевых задач связаны с определенными сложностями, и поэтому для их преодоления предлагаются новые методы, среди которых метод полос [7], метод точных искусственных граничных условий [8], метод квазиравномерных сеток [9], метод инверсии для численного решения внешних краевых задач [10] и др. В этой работе для численного моделирования распределенных электротехнических систем в полубезграничных областях используется метод [10] в виде модифицированного метода инверсии для полубезграничных сред (МИПБС). При этом в [10] показывается, что в численном методе моделирования применяется переход от исследуемой области D к составной из D'и D*, причем при переходе от D' до D* с помощью инверсии сохраняется шаблон расчетной схемы в силу свойства конформных отображений сохранения углов и постоянства растяжений.

1. Постановка задач для исследования распределенных электротехнических систем

1.1. Математическая модель для исследования ЭП высоковольтной линии электропередачи на примере ЛЭП (Cahora Bassa)

ЛЭП *представляет собой* высоковольтную биполярную линию постоянного тока мощностью 1920 МВт и длиной 1420 км, состоящей из двух линий напряжением +533 кВ и –533 кВ, расположенных на расстоянии 1 км [2], т.е. их электрические поля не влияют друг на друга. Схема ЛЭП постоянного тока показана на рис. 1а и соответствующая ей схема моделирования – на рис. 1б. Определяется ЭП одной линии ЛЭП с расщепленным полюсом A_p, состоящим из

четырех проводов бесконечной длины. Полюс расположен в полубезграничной области над проводящей поверхностью земли Z_k, причем грозозащитный провод находится над центральной частью полюса (см. рис. 1а). Исходные условия и обозначения следующие: высота подвеса до центральной части расщепленных проводов ЛЭП относительно земли H_1 = 8,55 м, расстояние между проводами l = 0,45 м, радиус провода R_1 = 0,0159 м, эквивалентный радиус расщепленных проводов $R_э$ = 0,215 м, высота расположения грозозащитного провода над центральной частью расщепленных проводов H_2 = 2,45 м, радиус грозозащитного провода R_2 = 0,0088 м [2]. Границы на схеме моделирования обозначены в виде Γ_i.

Cahora Bassa (*а*) и схема для ее моделирования (*б*)

Внешняя краевая задача сводится к решению уравнения Лапласа [11] в некоторой области $D(x, y)$:

$$\Delta\psi = 0, \qquad (1)$$

Δ – оператор Лапласа.

Граничные условия с учетом оси симметрии задачи следующие:

- на поверхности земли Z_k. (граница Γ_1):

$$\psi(x, y) = 0; \qquad (2)$$

- на границе $\Gamma\infty$:

$$\psi(\infty) = 0; \qquad (3)$$

- на оси симметрии (граница Γ_4):

$$\partial\psi/\partial n = 0; \qquad (4)$$

- на грозозащитном проводе (граница Γ_{23}):

$$\psi(x, y) = 0; \qquad (5)$$

- на проводах (границы Γ_{21}, Γ_{22}):

$$\psi(x, y) = \psi_A, \qquad (6)$$

где $D(x, y)$ – расчетная область, ограниченная границами Γ_1, Γ_{21}, Γ_{22}, Γ_{23}, Γ_4, Γ_∞; Γ_3 – внутренняя граница в области $D(x, y)$ (искусственная граница для внешней краевой задачи с учетом применения численного моделирования на основе МИПБС); $\Gamma\infty$ – условная граница на бесконечности; ψ_A – заданный потенциал на расщепленных проводах.

С учетом применения модифицированного МИПБС для плоско-параллельных задач исследуемая область $D(x,y)$ для полубезграничных областей может быть в виде квадрата, прямоугольника и др. (с возможностью упрощения применяемого численного метода решения задачи). Пусть, например, некоторая искусственная область $D_k(x, y)$, входящая в состав области $D(x, y)$, имеет форму четверти круга с радиусом R с исходными границами Γ_1, Γ_{21}, Γ_{22}, Γ_{23}, Γ_3, Γ_4. Для реализации предлагаемого метода необходима дополнительная область $D^*_k(x,y)$ такого же размера, как и $D_k(x,y)$, причем граница Γ_3 – общая граница этих областей. Согласно [10] часть исходной области $D(x, y)$, находящейся между границами Γ_3 и $\Gamma\infty$, отображается на область

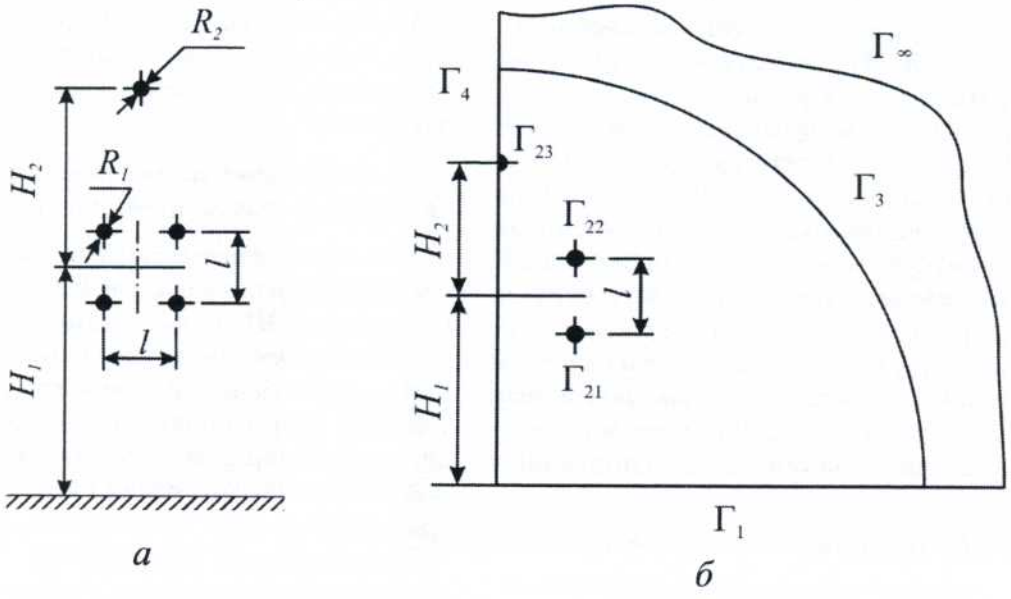


Рис. 1. Исходная схема высоковольтной линии постоянного тока

$D^*_k(x, y)$ с учетом метода инверсии. В узлах на границе области $D^*_k(x, y)$, расположенной по ее диаметру, присваивается нуль, что совпадает со значением потенциала ψ на границе Γ_1 области $D_k(x, y)$ и на бесконечности Γ∞.

1.2. Математическая модель для исследования ЭП на примере трехфазной ЛЭП напряжением 735 кВ (канадской компании "Hydro-Quebec")

Схема ЛЭП напряжением 735 кВ показана на рис. 2а соответствующая ей схема моделирования – на рис. 2б. Исходные условия следующие: схема расположения фаз горизонтальная, в каждой фазе по 4 провода с расстоянием между ними $l = 0{,}457$ м, высота подвеса фаз $H_1 = 27{,}43$ м, радиус проводов расщепленной фазы $R_1 = 0{,}0153$ м, расстояние между центрами фаз $l_1 = 13{,}72$ м, высота расположения грозозащитных проводов над центрами фаз $H_2 = 12{,}8$ м, расстояние между грозозащитными проводами $l_2 = 18$ м, радиус грозозащитных проводов $R_2 = 0{,}0061$ м [7]. (На рис.2 в jpg вместо L используется l).

Внешняя краевая задача также сводится к определению поля потенциала ψ и напряженности поля, т.е. предполагается стационарность электромагнитных полей и применение уравнения Лапласа в некоторой области $D(x, y)$ с учетом того, что для трехфазной ЛЭП расчеты выполнялись для момента времени t_i при выполнении одного из условий [7], а именно на средней фазе $U1 = 0$, на крайних фазах $U2$, и $U3$ соответственно равны 624,75 В и –624,75 В.

Поле потенциала ψ в области $D(x,y)$ определяется из уравнения (1) с учетом симметрии задачи, граничных условий (2)–(4) и трех дополнительных (см. рис. 2б):

• на проводах расщепленной фазы (границы Γ_{21}, Γ_{22}):

$$\psi(x, y) = U1; \qquad (7)$$

• на проводах расщепленной фазы (границы Γ_{23}, Γ_{24}, Γ_{25}, Γ_{26}):

$$\psi(x, y) = U2; \qquad (8)$$

• на грозозащитном проводе (граница Γ_{27}):

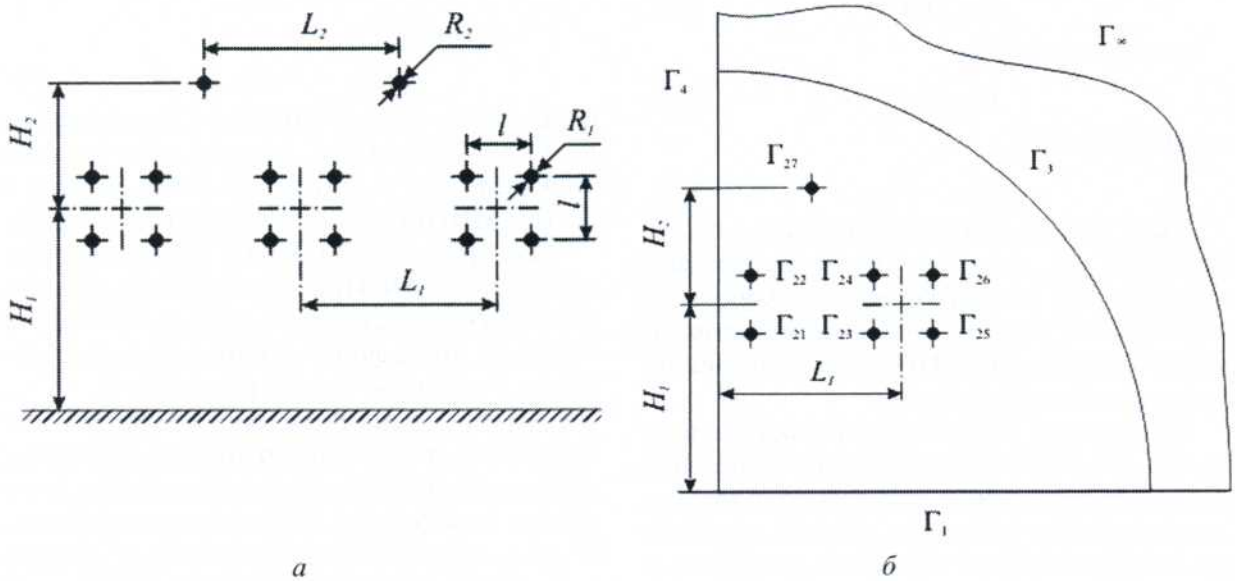
$$\psi(x, y) = 0; \qquad (9)$$

где $D(x, y)$ – расчетная область, ограниченная границами Γ_1, Γ_4, Γ∞, Γ_{21} - Γ_{27}, Γ_3 – внутренняя граница в области $D(x,y)$ (искусственная граница для внешней краевой задачи с учетом применения модифицированного МИПБС на основе [10]).

1.3. Математические модели для исследования ЭП ионизатора воздуха и стержневого молниеотвода системы молниезащиты

Ионизаторы типа "Аэроион-Про" [6] предназначены для создания в больших помещениях отрицательных аэроионов, представляют собой тонкий провод, расположенный по периметру помещения вблизи потолка, подключенный к отрицательному потенциалу высоковольтного блока. Схема для моделирования этого типа ионизатора показана на рис.3а.

Системы **молниезащит в основном содержат** молниеотводы, токоотводы-проводники и за-



а *б*

Рис. 2. Исходная схема ЛЭП напряжением 735 кВ (*а*) и схема для ее моделирования (*б*)

землители. Среди систем **молниезащит можно выделить** молниеотводы стержневого типа, схема для моделирования которого показана на рис. 3б.

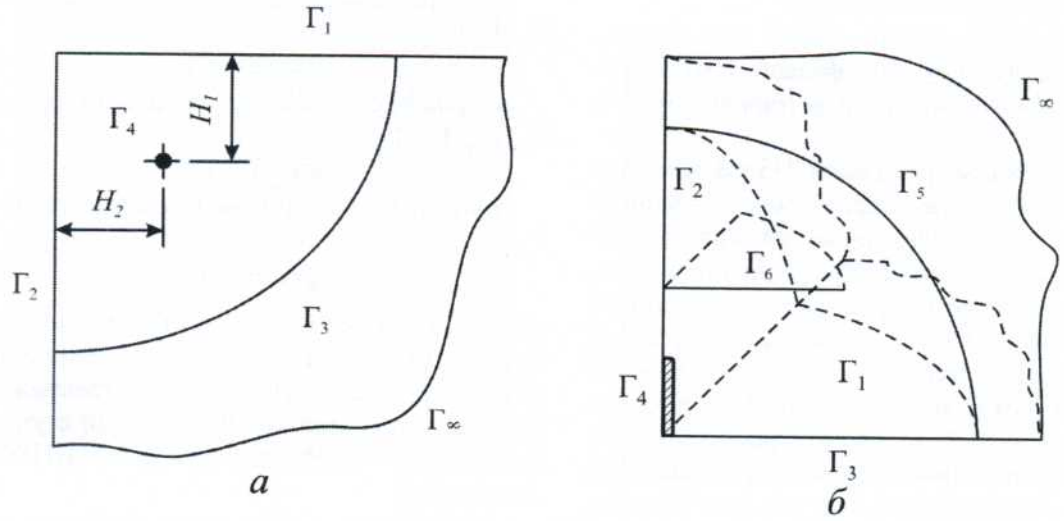


Рис. 3. Схемы моделирования распределенной системы ионизации воздуха (*а*) и стержневого молниеотвода системы молниезащиты (*б*)

Краевая задача (см. рис. 3, *а*) для распределенной системы ионизации воздуха с учетом плоско-параллельной постановки сводится к определению поля потенциала ψ на основе уравнения Лапласа (1) в исследуемой области $D(x,y)$, как, например, в [12], при следующих условиях:

- на границах Γ_1 и Γ_2:

$$\psi(x, y) = 0; \qquad (10)$$

- на границе $\Gamma\infty$:

$$\psi(\infty) = 0; \qquad (11)$$

- на границе Γ_4:

$$\psi(x, y) = \psi_B \qquad (12)$$

Здесь $D(x, y)$ – расчетная область с границами Γ_1, Γ_2, Γ_4, $\Gamma\infty$; Γ_3 – внутренняя граница в области $D(x, y)$ (искусственная граница для этой краевой задачи с учетом использования модифицированный МИПБС); ψ_B – потенциал на коронирующем проводе.

Для расчета систем **молниезащит** принят подход [13] с возможностью определения ЭП в окрестностях стержневых молниеотводов, при котором вместо объемных зарядов на облаке задается потенциал и принимается напряженность поля в пределах E_0 = 1 В/м и менее. Краевая задача (см. рис. 3б) в трехмерной постановке с учетом части сферы сводится к определению поля потенциала ψ и напряженности поля в области $D(x,y,z)$ на основе уравнения Лапласа (1) при следующих условиях:

- на границе Γ_1 и на заземленном стержне (границе Γ_4):

$$\psi(x, y, z) = 0; \qquad (13)$$

- на границе $\Gamma\infty$:

$$\psi(\infty) = 0; \qquad (14)$$

- на границах Γ_2, Γ_3:

$$\partial\psi/\partial n = 0; \qquad (15)$$

- на границе Γ_6:

$$\psi(x, y, z) = \psi_C; \qquad (16)$$

где $D(x, y, z)$ – расчетная область, ограниченная Γ_1, Γ_2, Γ_3, Γ_4, Γ_6, $\Gamma\infty$; Γ_5 – внутренняя граница в области $D(x, y, z)$ (искусственная граница с учетом МИПБС); ψ_C – потенциал на стержне.

Для трехмерной задачи с учетом модифицированного МИПБС некоторая выделенная область $D_m(x, y, z)$, входящая в состав области $D(x, y, z)$, имеет форму некоторой части полусферы с радиусом R_0 и с границами $\Gamma_1 - \Gamma_6$ (с учетом упрощения численного моделирования область $D_m(x, y, z)$ может быть в виде куба, параллелепипеда и др.). Для реализации этого метода необходима дополнительная область $D^*_m(x, y, z)$ с учетом вида области $D_m(x, y, z)$ для того, чтобы часть исходной области $D(x, y, z)$ между границами Γ_5 и $\Gamma\infty$, отобразилась на об-

ласть $D^*_m(x, y, z)$ с учетом подхода [10]. При этом на границе области $D^*_m(x, y, z)$, расположенной по диаметру некоторой части полусферы, присваивается значение, совпадающее со значением потенциала ψ на бесконечности Γ_∞.

2. Дискретная модель и численный метод исследования

В качестве примера для одной из внешних краевых задач с учетом уравнения (1) и граничных условий (2)–(6) рассмотрим представление исходных уравнений в операторной форме для дискретной области D_d с учетом определения поля электрического потенциала ψ в узле (i,j) в виде:

$$W_h\psi^{(h)} \equiv \begin{cases} W_{xx}\psi_{ij} + W_{yy}\psi_{ij} = 0, \ (x_i, y_j) \in D_d; \\ W_n\psi_{ij} = 0, \ (x_i, y_j) \in \Gamma_4; \\ \psi_{ij} = \psi_A, \ (x_i, y_j) \in \Gamma_{21}, \Gamma_{22}; \\ \psi_{ij} = 0, \ (x_i, y_j) \in \Gamma_1, \Gamma_{23}, \Gamma_\infty. \end{cases} \quad (17)$$

Здесь $W_{xx} = \partial^2/\partial x^2$, $W_{yy} = \partial^2/\partial y^2$, $W_n = \partial/\partial n$, производные потенциала представляются конечно-разностными аппроксимациями. Исследуемая область D_d, включающая области $D_{kd}(x, y)$ и $D^*_{kd}(x, y)$, является дискретной, причем из основных вариантов многомерных эйлеровых сеток выбрана универсальная регулярная сетка в виде прямоугольной с учетом того, что исследуемая область – часть круга с расположенными в нем проводами круглого сечения (см. рис. 1б). Аналогичный подход применяется и в остальных задачах.

Аппроксимация криволинейных границ области D_d выполняется на основе метода Брезенхема. В узлах областей $D_{kd}(x, y)$ и $D^*_{kd}(x, y)$ значения рассчитываются методом конечных разностей и с учетом метода инверсии внешней бесконечной области [10]. На общей границе Γ_3 этих областей расчет ведется по формулам аналогично внутренних узлов, а значения в них, не входящих в основную область $D_{kd}(x, y)$, берутся из дополнительной области $D^*_{kd}(x, y)$, и наоборот. В связи с этим условие на общей границе этих областей отсутствует в постановке задачи.

На оси симметрии Γ_4 с учетом того, что относительно вертикальной оси $\psi_{i+1,j} = \psi_{i-1,j}$, тогда потенциал $\psi_{i,j}$ рассчитывается по формуле:

$$\psi_{i,j} = \frac{2\psi_{i+1,j} + \psi_{i,j+1} + \psi_{i,j-1}}{4}. \quad (18)$$

Так как во внутренних узлах сетки выполняется условие сходимости итерационного метода Гаусса-Зейделя, то во всех расчетных внутренних узлах области $D_{kd}(x, y) \cup D^*_{kd}(x, y)$ потенциал $\psi_{i,j}$ рассчитывается с помощью экстраполяционного метода Либмана [14] по формулам:

$$\psi_{i,j} = \frac{\psi_{i+1,j} + \psi_{i-1,j} + \psi_{i,j+1} + \psi_{i,j-1}}{4}, \quad (19)$$

$$\psi^S_{i,j} = \psi^{S-1}_{i,j} + \alpha\left(\psi^{S0}_{i,j} - \psi^{S-1}_{i,j}\right), \quad (20)$$

где α – ускоряющий коэффициент ($1 \le \alpha \le 2$); $\psi^{S-1}_{i,j}$ – значение в узле $\psi_{i,j}$, вычисленное на предыдущей итерации; $\psi\Sigma^0_{i,j}$ – значение в узле $\psi_{i,j}$, вычисленное в текущей итерации согласно (19); $\psi^S_{i,j}$ – новое значение.

Модуль напряженности поля $|\mathrm{E}|$ в каждой i-ой точке исследуемой области определяется следующим образом:

$$|E| = \sqrt{E_x^2 + E_y^2}. \quad (21)$$

Компоненты напряженности электрического поля E_x и E_y с использованием центрально-разностной аппроксимации определяются как

$$E_{i,j} = -(\partial\psi_{i+1,j} - \partial\psi_{i-1,j})/2h; \quad E_{i,j} = -(\partial\psi_{i,j+1} - \partial\psi_{i,j-1})/2h, \quad (22)$$

где i – изменение по координате x; j – изменение по координате y.

3. Результаты численных расчетов

Тестирование применяемого метода на примере одиночного провода относительно поверхности земли дано в [15], при этом использовались методы инверсии и зеркальных изображений. Потенциал произвольной точки M на основе метода зеркальных изображений определялся с учетом выражения [11]:

$$\psi_M = \frac{\tau}{2\pi\varepsilon\varepsilon_0}\ln\frac{b}{a} + C, \quad (23)$$

где ε_0 – электрическая постоянная; ε – относительная диэлектрическая проницаемость среды; τ – линейная плотность заряда; a – расстояние от точки M до геометрической оси провода; b – расстояние от точки M до зеркального отображения геометрической оси провода.

При численных расчетах напряженности поля вблизи проводника и на некотором удалении от земли установлено [15], что средние

погрешности при расчете поля потенциала проводника, например, вблизи поверхности земли находятся в пределах от 0,410 % до 5,258 % при изменении расстояния от проводника до поверхности земли.

Результаты расчета плоскопараллельного ЭП с учетом расщепленных и грозозащитного проводов для ЛЭП постоянного тока показаны на рис. 4, причем на рис. 4а показан характер распределения поверхностей равных напряженностей электростатического поля с заданным шагом ΔE между грозозащитным проводом и поверхностью земли, а на рис. 4б показано распределение этих же поверхностей поля вблизи верхней части полюса в увеличенном масштабе.

дугу в сторону оси симметрии, направляется вниз. Полученные результаты с учетом того, что E_{max} находится на поверхностях проводов, соответствуют [2], так как экстремальные значения напряженности E находятся на нижних и верхних проводах расщепленного полюса строго под углом 45°. Следует отметить, что, согласно расчетам на основе математической модели с учетом некоторого провода с эквивалентным радиусом $R_э$ [1] для ЛЭП [2], линия 1 модуля напряженности E_{max} находится по оси симметрии. Для случая с учетом (рис. 4), чтобы эта линия была направлена от центра полюса в сторону проводящей поверхности Z_k, необходим некоторый эквивалентный радиус $R_{эк} = R_k$, отличающийся от расчетного $R_э = 0{,}215$ м более чем в 5 раз. Следовательно,

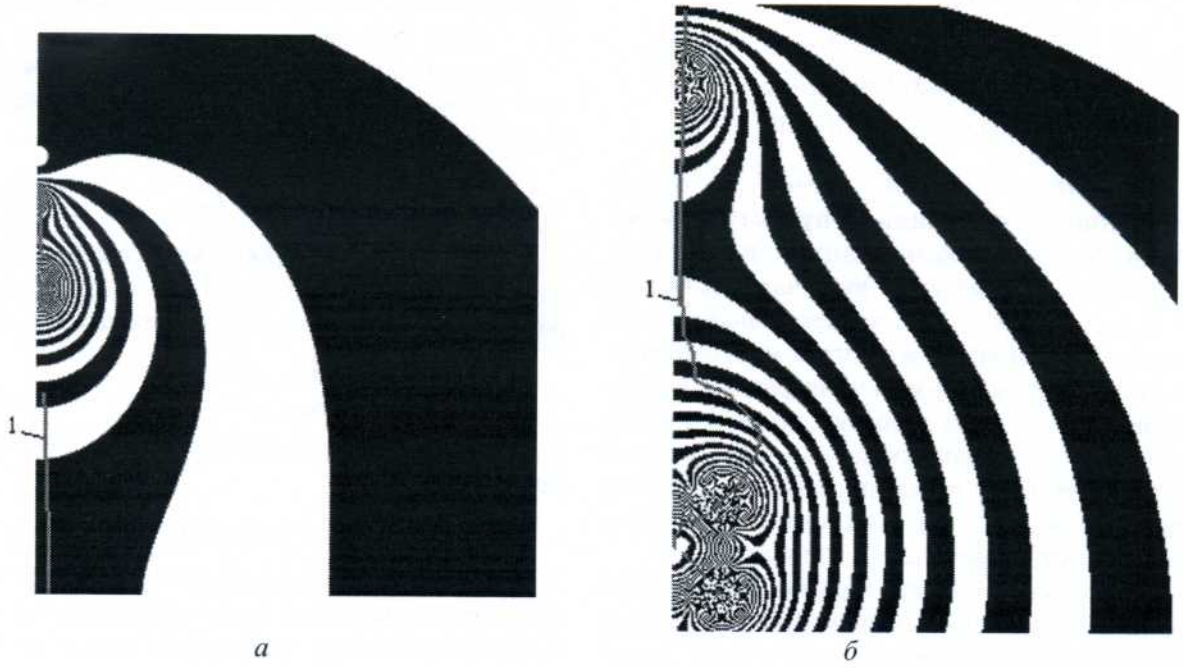


Рис. 4. Распределение поверхностей равных напряженностей поля с шагом ΔE для ЛЭП постоянного тока для разных масштабов

Анализ результатов расчета показывает (см. рис. 4б), что для верхнего провода полюса линия 1 модуля максимальной напряженности поля (E_{max}) вначале направлена под углом порядка 45° (относительно грозозащитного провода), а затем, описав некоторую дугу в сторону оси симметрии, направляется вверх параллельно этой оси.

Аналогичные результаты наблюдаются и в нижней части полюса, только линия 1 модуля напряженности E_{max} вначале направлена под углом порядка 45° к поверхности земли и выходит из нижнего провода, а затем, описав некоторую

расчеты в окрестностях проводов расщепленного полюса по модели с эквивалентным радиусом [1] являются некорректными.

Результаты численных расчетов для количественной оценки распределения напряженности поля для ЛЭП постоянного тока на разных уровнях относительно поверхности земли, показаны на рис. 5, причем эти результаты представлены в относительных величинах:

$$E^* = -\partial\psi^* / \partial s^*; \quad \psi_i^* = \psi_i / \psi_A; \quad s_i^* = s_i / l,$$

где l – расстояние между проводами; ψ_A – потенциал расщепленных проводов.

Из полученных результатов расчета следует (см. рис. 5), что на разных уровнях h^* относительно поверхности земли и с удалением от оси симметрии полюса величина напряженности поля изменяется по некоторому колоколообразному закону. При этом вблизи полюса максимальная величина $E^* = f(S^*_L)$ находится под крайними расщепленными проводами, а на уровне земли находится под центральной частью полюса ЛЭП Cahora Bassa. С удалением от расщепленных проводов от h^*_3 до h^*_2, т.е. с изменением уровня в 1,3 раза, максимум напряженности поля резко уменьшается (более чем в 4 раза), а затем при дальнейшем уменьшении h^*_i максимум $E^* = f(S^*_L)$ также уменьшается, но с существенно меньшим градиентом.

Результаты расчетов поля для трехфазной ЛЭП напряжением 735 кВ в виде характеристики $E = f(S_L)$ показаны на рис. 6, причем выполнены для момента времени t_i с учетом одного из условий [7], а именно на средней фазе $U1 = 0$, на крайних фазах $U2 = 624{,}75$ В и $U3 = -624{,}75$ В.

Анализ зависимостей $E = f(S_L)$ показал, что максимум напряженности поля вблизи Z_k находится не под центром крайних расщепленных фаз (см. рис. 6, где l_1 – центр симметрии крайней расщепленной фазы), а смещается во внешнюю сторону относительно крайних фазовых проводов при приближении к земле. В теоретических исследованиях этот эффект не изучался, но подтверждается многочисленными экспериментальными исследованиями [1].

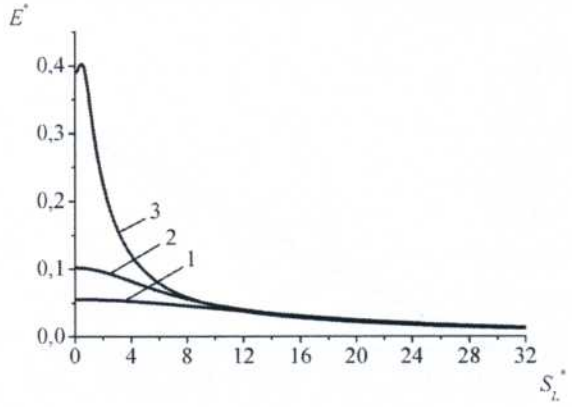


Рис. 5. Характеристика $E^* = f(S^*_L)$ относительно поверхности земли: 1 – $h^*_1 = 4{,}44$; 2 – $h^*_2 = 13{,}33$; 3 – $h^*_3 = 17{,}78$

При расчетах ионизаторов воздуха исследовалось влияние изменения соотношений H_1/H_2 (см. рис. 3а) на величину напряженности поля на некотором уровне относительно границы Γ_1. Результаты расчета напряженности E в окрестностях провода при условии $H_1 = H_2$ (провод расположен симметрично относительно Γ_1 и Γ_2) в виде распределения поверхностей $E_i = \text{const}$ с шагом ΔE показаны на рис. 7а. Результаты численных расчетов распределения модуля напряженности в виде $E^* = f(S^*_L)$ на уровне $H^*_k = 2H_1/h$ показаны на рис. 7б (в относительных единицах, где h – нормирующая величина, равная H_1).

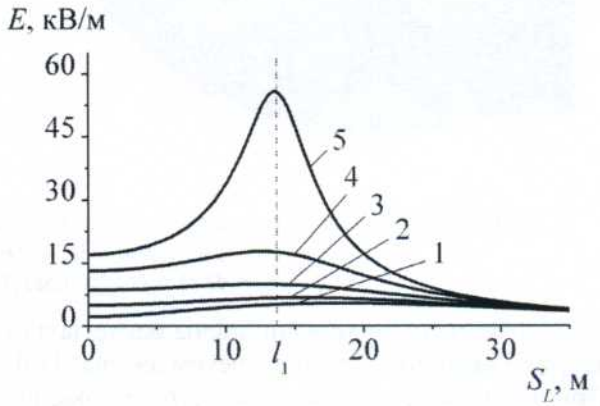
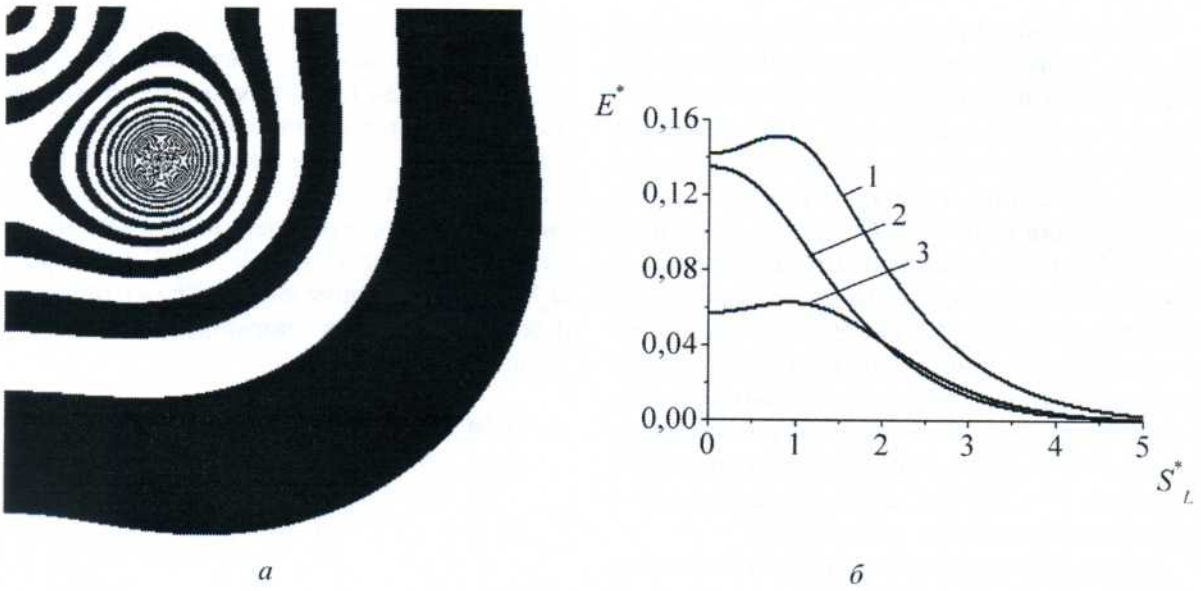


Рис. 6. Характеристика $E = f(S_L)$ относительно поверхности земли: 1 – $H_1 = 5$ м, 2 – $H_2 = 10$ м, 3 – $H_3 = 15$ м, 4 – $H_4 = 20$ м, 5 – $H_5 = 25$ м

Из анализа результатов (см. рис. 7б) следует, что на уровне H^*_k при смещении провода к границе Γ_2 максимальная величина модуля напряженности поля для $E^* = f(S^*_L)$ меньше, чем при симметричном его расположении, но значительно больше (более 2-х раз), чем при смещении провода к границе Γ_1. Следовательно, при смещениях провода к любой из границ, Γ_1 или Γ_2 относительно оси симметрии максимальная напряженность поля на уровне H^*_k от границы Γ_1 уменьшается по сравнению с его симметричным расположением, но при этом максимальная напряженность на проводе увеличивается в пределах 10 %. Эти результаты необходимо учитывать при монтаже коронирующего провода для ионизаторов воздуха типа "Аэроион-Про" [6].

При численных расчетах также исследовался стержневой молниеотвод на основе модифицированного МИПБС с учетом данных [13]. В первом варианте задачи граница Γ_6 находилась на высоте $H = 3l_s$ и ее радиус был равен $10l_s$. Для сравнительного анализа распределения потенциала ψ результаты расчетов по этому варианту представлены на рис. 8а с учетом данных [13].

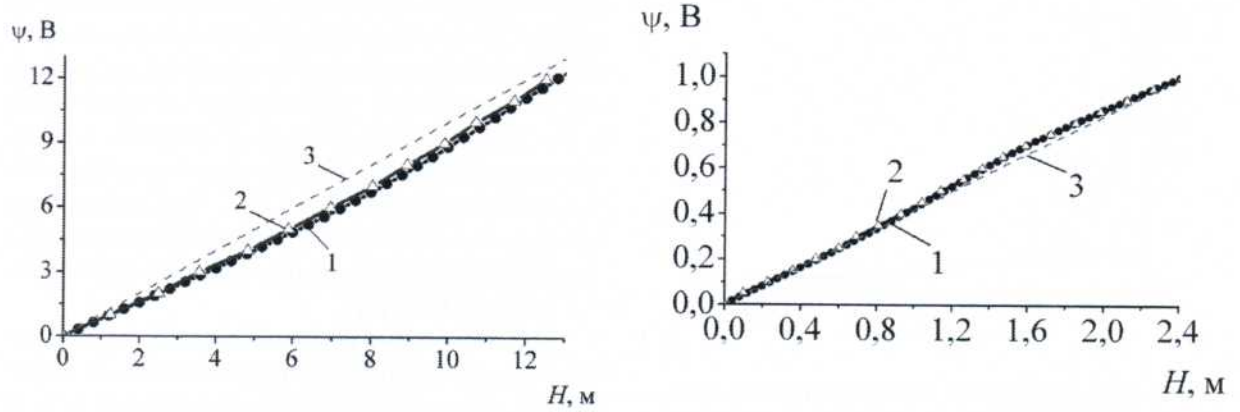


а *б*

Рис. 7. Распределение поверхностей E_i = const с шагом ΔE при $H_1 = H_2$ (*а*) и характеристика $E^* = f(S^*_L)$ на уровне $H_k^* = 2H_1/h$ (*б*) для случаев: 1 – $H_1 = H_2$; 2 – $H_1^* = H_1/h$ и $H_{2m}^* = 0{,}5H_2/h$; 3 – $H_{1m}^* = 0{,}5H_1/h$ и $H_2^* = H_2/h$

Кроме того, были выполнены расчеты для второго варианта задачи с учетом данных [13]. При этом исследовалось два стержня, расположенных соответственно на границах Γ_1 и Γ_6, причем смещенных друг относительно друга в вертикальной плоскости и направленных навстречу друг другу (по аналогии с рис. 3б). Постановка задачи соответствует уравнению Лапласа (1) и граничным условиям (13)–(16), так как эти стержни расположены соответственно на границах Γ_1 и Γ_6. При этом исходное условие на границе Γ_6 со стержнем с учетом данных [13] соответствует ψ_C = 1 В, а стержень на границе Γ_1 заземленный. Расстояние от плоскости Γ_1 до Γ_6 составляло величину 2,4 м. С учетом симметрии задачи вертикальная ось OY проходит между стержнями посередине (по аналогии с рис. 1б), при этом расстояние между стержнями в горизонтальной плоскости равно 0,4 м. Длина стержня, находящегося на границе Γ_1, равна 1 м, длина стержня, находящегося на границе Γ_6, равна 1,3 м. Результаты расчетов по этому варианту представлены на рис. 8б.

Сравнение результатов расчета по 2-м методам для стержневого молниеотвода проводилось в вертикальной плоскости на расстоянии



а *б*

Рис. 8. Распределение ψ для стержневого молниеотвода при E_0 = 1 В/м (*а*) и для 2-х стержней на границах Γ_1 и Γ_6, при E_0 = 0,42 В/м (*б*): 1 – расчет МИПБС; 2 – данные [13]; 3 – распределение ψ при заданном E_0

S_{k1} = 3,2 м от центра молниеотвода (см. рис. 8а), причем максимальная высота распределения поля потенциала была представлена в [13] до уровня порядка H_{k3} = 13 м. Погрешность расчета (см. рис. 8а, кривая 1 и 2) распределения потенциалов ψ по вертикали на расстоянии S_{k1} менее 0,5 %. Анализ результатов расчета для двух стержней по 2-м методам проводился также в вертикальной плоскости на расстоянии S_{k2} = 0,6 м по оси OX относительно оси OY. В этом случае максимальная погрешность распределения потенциалов ψ до 4,0 % (см. рис. 8б, кривая 1 и 2), т.е. больше, чем в первом случае, так как величина потенциала меньше. Распределения потенциала ψ, полученные при расчетах по 2-м методам в виде характеристик 1 и 2 (см. рис. 8), хорошо коррелируют между собой.

Заключение

Анализ результатов численного моделирования распределенных электротехнических систем с учетом их сравнения с данными других авторов показал возможность применения модифицированного метода инверсии для полубезграничных сред при исследованиях особенностей этих систем.

1. При исследованиях электрических полей *ЛЭП* показана возможность определения одного из важных показателей степени влияния *высоковольтных ЛЭП* на биологические объекты в виде распределения напряженности ЭП на уровне земли. При этом установлен закон изменения модуля максимальной напряженности поля относительно поверхности земли для расщепленных проводов на примере ЛЭП Cahora Bassa, соответствующий колоколообразному закону. Выявлено, что максимальное значение напряженности поля для трехфазных ЛЭП (компании "Hydro-Quebec") смещается в сторону от крайних фазовых проводов при приближении к земле, а это важно учитывать в условиях нахождения людей вблизи ЛЭП при переменном рельефе поверхности земли.

2. При численном моделировании распределенных систем ионизации воздуха установлены характерные закономерности, важные для монтажа таких систем в помещениях зданий или сооружений. Выявлено, что коронирующий провод необходимо устанавливать по периметру помещения вблизи потолка, но с учетом его асимметричного расположения в угловых зонах, так как это приведет к существенному уменьшению максимума напряженности поля под этим проводом.

3. Исследования стержневых молниеотводов для систем молниезащит показали качественные и количественные совпадения результатов численного моделирования исследуемых объектов с результатами других авторов, причем на основе предлагаемого метода расчета этих систем существенно упрощается определение особенностей распределения ЭП относительно стержневых молниеотводов, что важно при оценке некоторой области для защиты от молнии.

СПИСОК ЛИТЕРАТУРЫ

1. Дальние электропередачи 750 кВ. В 2ч. Ч.1. Воздушные линии/ Под общ. ред. А.М. Некрасова и С.С. Рокотяна. – М.: Энергия, 1974. – 224 с.

2. **Holtzhausen, J.P.** Corona On The Cahora Bassa DC Line: Theory And Tests On A Scale Model/ J.P. Holtzhausen, H. Hendricks, P.J. Pieterse// SAUPEC 2002, Vanderbijlpark, January 2002. – P. 283–287.

3. **Базелян, Э.М.** Физические и инженерные основы молниезащиты/ Э.М. Базелян, Б.Н. Горин, В.И Левитов. – Л.: Гидрометеоиздат, 1978. – 327 с.

4. **Гоник, А.Е.**Электрические и механические фильтры кассетного типа для очистки воздуха от сварочных аэрозолей / А.Е. Гоник, Н.Н. Жуков// Защита окружающей среды, здоровье, безопасность в сварочном производстве: Труды I Междунар. научно-практической конференции. – Одесса: Изд-во Астропринт, 2002. – С.169–174.

5. Патент №2145910 (Россия). Двухзонный электрофильтр для очистки газов/ Гоник А.Е., Жуков Н. Н., Потапенко А.Н., Белоусов А.В., Штифанов А.И. – Опубл. в Бюл. № 6, 2000 г.

6. http://www.ion.moris.ru

7. **Yang, Y.** The strip simulation method for computing electric field on conductor surfaces/ Y. Yang, D. Dallaire, J. Ma, and F. P. Dawalibi// Proceedings of the Third IASTED International Conference on Power and Energy Systems, EuroPES 2003, Marbella, Spain, Sept. 3–5. – 2003. – P. 353–357.

8. **Софронов, И.Л.** Точные искусственные граничные условия для некоторых задач аэродинамики и дифракции: автореферат дис. ... доктора физико-математических наук: 01.01.07/ Софронов Иван Львович. – М.: ИПМ РАН им. М.В. Келдыша. – 1999. – 28 с.

9. **Альшина, Е.А.** Численное решение краевых задач в неограниченной области/ Е.А. Альшина, Н.Н. Калиткин, С.Л. Панченко// Математическое моделирование. – 2002. – т. 14. – №11. – С. 10-22.

10. **Дыльков, М.И.** Метод инверсии для численного решения внешних краевых задач для уравнений эллиптического типа: дис...канд. физ.-мат. наук: 05.13.18: защ. 17.09.04: утв. 25.12.04/ Дыльков Михаил Иванович. – Белгород, 2004. – 140с.

11. **Бессонов, Л.А.** Теоретические основы электротехники. Электромагнитное поле: Учебник/ Л.А. Бессонов. – М.: Гардарики, 2001. – 317 с.

12. **Потапенко, А.Н.** Основы математического моделирования двухзонных электрофильтров и некоторые особенности их применения в энергоэффективных вентиляционных системах/ А.Н. Потапенко, А.В. Белоусов, Л.И. Колтунов, Е.А. Потапенко// Известия вузов. Проблемы энергетики. – 2006. – № 5–6. – С. 69-84.

13. **Резинкина, М.М.** Расчет трехмерных электрических полей в системах, содержащих тонкие проволоки/ М.М. Резинкина// Электричество. – 2005. – № 1. – С. 44–49.

14. **Сипайлов, Г.А.** Электрические машины (специальный курс)/ Г.А. Сипайлов, Е.В. Кононенко, К.А. Хорьков – М.: Высш. шк., 1987. – 287 с.

15. **Потапенко, А.Н.** Особенности метода численного исследования воздушных линий электропередач, его тестирование и результаты расчетов/ А.Н. Потапенко, Е.А. Канунникова, М.И. Дыльков// Вестник БГТУ им В.Г. Шухова. – 2007. – №4 – С. 111–121.

Ушмоткин А. С., Коржик В. И.

Оценка эффективности использования системы цифровых "водяных" знаков для изображений в условиях коалиционных атак

1. Введение

Технология ЦВЗ используется для защиты авторских прав легальных распространителей музыки, видео и изображений. В этом случае информация, вкладываемая в эти продукты, содержит идентификационные данные легального покупателя, что позволяет обнаружить впоследствии тех нечестных покупателей, которые копируют и нелегально распространяют купленные ими продукты. В дальнейшем такие нелегальные распространители могут быть привлечены к суду. С другой стороны, нелегальные распространители копий стремятся принять все меры, чтобы сделать их отслеживание невозможным. В частности, они могут объединяться в группы (коалиции) с целью такой совместной обработки своих индивидуальных продуктов, при которых получается копия, не позволяющая отследить ни одного из участников такой коалиции. Подобные действия называются обычно *коалиционной атакой*.

При разработке системы ЦВЗ (которая в данном случае обычно называется системой с *цифровым отпечатком пальца*) необходимо учитывать возможность коалиционной атаки и выбирать параметры ЦВЗ так, чтобы обеспечить малые вероятности как *пропуска* участника коалиции – *Pm*, так и *ложного обнаружения Pfa* пользователя, на самом деле не входящего в коалицию.

Эффективность системы ЦВЗ определяется парой этих вероятностей и объемом вкладываемой информации, в зависимости от качества продукта и количества возможных участников коалиции. Очевидно, что вероятность образования коалиции будет тем меньше, чем больше количество ее участников.

В настоящей работе производится теоретический анализ эффективности систем ЦВЗ для неподвижных изображений в условии коалиционной атаки, а также экспериментальное исследование данной системы при помощи моделирования.

2. Оценка вероятностей ошибок P_m и P_{fa} при использовании информированного декодера

Рассмотрим метод вложения ЦВЗ с использованием ШПС [1]:

$$C_{wj}(n) = C(n) + \alpha\pi_j(n),$$
$$n = 1, 2, \ldots N, i = 1, 2 \ldots L, \qquad (1)$$

где $C(n)$ – неподвижное изображение (покрывающее сообщение (ПС)), передаваемое одновременно L легальным покупателям; $\pi_i(n)$ – псевдослучайная ± 1–ичная

последовательность, которая вырабатывается по секретному *стего-ключу* индивидуально для каждого *i*-го пользователя (покупателя); $C_{wi}(n)$ – сигнал с вложенным ЦВЗ для i-го пользователя; α – амплитудный коэффициент, определяющий "глубину" вложения ЦВЗ и влияющий на качество ПС после вложения ЦВЗ; N – количество отсчетов (пикселей) изображения; L – общее количество легальных покупателей изображения $C(n)$.

Известно множество различных коалиционных атак, но, как показано в [2], наилучшей из них является атака *усреднением*:

$$C'_w(n) = \frac{1}{K}\sum_{j\in Sc} C_{wj}(n) + \varepsilon(n), \qquad (2)$$

где S_c – множество пользователей, объединившихся в коалицию, т.е. так называемых "пиратов"; $K = |S_c|$ – количество участников коалиции; $\varepsilon(n)$ – дополнительный аддитивный шум со средним значением $E\{\varepsilon(n)\} = 0$ и с дисперсией $Var\{\varepsilon(n)\} = \sigma_\varepsilon^2$; $C'_w(n)$ – копия, которая будет далее нелегально распространяться от лица всех участников коалиции.

Задача легального распространителя продукта состоит в том, чтобы, получив доступ к пиратской копии $C'_w(n)$, определить с достаточной надежностью желательно всех, или как минимум одного, из участников коалиции. В этом случае легальный дистрибьютор (ЛД) может направить в суд иск о нарушении его авторских прав.

Предположим сначала, что ЛД обладает точным образцом продукта $C(n)$. Тогда обнаружение участия *i*-го пользователя в коалиции S_c может быть проведено по следующему правилу:

$$i \in S_c, \text{ если } \Lambda_i > \lambda,$$
$$i \notin S_c, \text{ если } \Lambda_i \le \lambda, \qquad (3)$$

где

$$\Lambda_i = \sum_{n=1}^{N}(C'_w(n) - C(n))w_i(n), i = 1,2...N \qquad (4)$$

$$w_i(n) = \alpha\pi_i(n),$$

$\pi_i(n)$ – та же ПСП, которая использовалась при вложении ЦВЗ.

Для решающего правила (3), вероятность пропуска *i*-го пользователя P_{mi}, который действительно участвовал в коалиции, определяется выражением

$$P_{mi} = \Pr\{\Lambda_i \le \lambda \mid i \in S_c\}, \qquad (5)$$

где $\Pr\{A\}$ означает вероятность события A.

Вероятность ложного обнаружения *i*-го пользователя P_{fai} как участника коалиции, тогда как в действительности его там не было, определяется выражением

$$P_{fai} = \Pr\{\Lambda_i > \lambda \mid i \notin S_c\}. \qquad (6)$$

Для расчета P_{mi} и P_{fai} преобразуем Λ_i, подставляя (2) в (4):

$$\Lambda_i = \sum_{n=1}^{N}(\frac{1}{K}\sum_{j\in S_c} w_j + \varepsilon(n))w_i(n). \qquad (7)$$

Если общее количество пикселей изображения N велико, то в силу центральной предельной теоремы теории вероятностей [3] случайные величины Λ_i будут иметь в асимптотике гауссовское (нормальное) распределение, т.е.

$$\Lambda_i = \Lambda_{i1} = N(E\{\Lambda_{i1}\}, Var\{\Lambda_{i1}\}),$$
$$\Lambda_i = \Lambda_{i0} = N(E\{\Lambda_{i0}\}, Var\{\Lambda_{i0}\},$$

где Λ_{i1} – совпадает с Λ_i, если $i \in S_c$; Λ_{i0} – совпадает с Λ_i если $i \notin S_c$; $E\{\cdot\}$ – означает математическое ожидание, $Var\{\cdot\}$ – дисперсию.

Тогда для вероятностей P_{mi} и P_{fai} получим выражения:

$$P_{mi} = Q\left(\frac{E\{\Lambda_{i1}\} - \lambda}{\sqrt{Var\{\Lambda_{i1}\}}}\right), \qquad (8)$$

$$P_{fai} = 1 - Q\left(\frac{E\{\Lambda_{i0}\} - \lambda}{\sqrt{Var\{\Lambda_{i0}\}}}\right), \qquad (9)$$

где $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-\frac{t^2}{2}}dt$.

Видно, что расчет P_{mi} и P_{fai} сводится к нахождению $E\{\Lambda_{i0}\}$, $E\{\Lambda_{i1}\}$, $Var\{\Lambda_{i0}\}$, $Var\{\Lambda_{i1}\}$. Используя свойство взаимной независимости последовательностей $w_i(n)$ и элементов при $n' \ne n$ в каждой из таких последовательностей, получаем из (7):

$$E\{\Lambda_{i1}\} = \frac{\alpha^2 N}{K}; E\{\Lambda_{i0}\} = 0, \qquad (10)$$

$$Var\{\Lambda_{i1}\} = N(\sigma_\varepsilon^2\alpha^2 + \frac{\alpha^4(K-1)}{K^2});$$
$$Var\{\Lambda_{i0}\} = N(\sigma_\varepsilon^2\alpha^2 + \frac{\alpha^4}{K}). \qquad (11)$$

Подставляя (10) и (11) в (8) и (9), получаем:

$$P_{mi} = Q\left(\frac{\frac{\alpha^2 N}{K} - \lambda}{\sqrt{N(\sigma_\varepsilon^2\alpha^2 + \frac{\alpha^4(K-1)}{K^2})}}\right), \qquad (12)$$

$$P_{fai} = 1 - Q\left(\frac{-\lambda}{\sqrt{N(\sigma_\varepsilon^2\alpha^2 + \frac{\alpha^4}{K})}}\right). \qquad (13)$$

Для упрощения теоретических расчетов рассмотрим важный частный случай, когда целесообразно выбрать такой порог $\lambda = \lambda_o$, при котором $P_{mi} = P_{fai}$. Приравнивая (12) и (13), получаем выражение для выбора такого порога:

$$\lambda_o = \frac{\frac{\alpha^2 N}{K}}{\left(\sqrt{\frac{K^2\sigma_\varepsilon^2 + \alpha^2(K-1)}{K^2\sigma_\varepsilon^2 + K\alpha^2}}\right) + 1}. \qquad (14)$$

Если число участников коалиции $K >> 1$, то получаем из (14):

$$\lambda_o = \frac{\alpha^2 N}{2K}. \qquad (15)$$

Рассчитаем качество ПС сразу после погружения ЦВЗ и после коалиционной атаки в терминах *отношения сигнал/шум* (η_w и η_a соответственно):

$$\eta_w = \frac{\sigma_c^2}{\alpha^2}, \qquad (16)$$

$$\eta_a = \frac{\sigma_c^2}{Var\{C(n) - C'_w(n)\}} = \frac{\sigma_c^2}{\sigma_\varepsilon^2 + \frac{\alpha^2}{K}}. \qquad (17)$$

Из соотношения (17) получаем, что $\eta_a \approx \eta_w K$ при малом аддитивном шуме атаки ($\sigma_\varepsilon^2 \approx 0$), т.е. оно даже улучшается в K раз после коалиционной атаки.

Подставляя (15), (16) и (17) в (12), (13), получаем для $K >> 1$:

$$P_{mi} = P_{fai} = P_e = Q\left(\frac{1}{2}\sqrt{\frac{N}{K^2\eta_\varepsilon + K}}\right), \qquad (18)$$

где $\eta_\varepsilon = \frac{\sigma_\varepsilon^2}{\alpha^2}$.

Если аддитивный шум мал, то

$$P_e \approx Q\left(\frac{1}{2}\sqrt{\frac{N}{K}}\right). \qquad (19)$$

Из соотношения (19) видим, что при $K >> 1$, коалиционная атака может быть скомпенсирована (т.е. обеспечиваются те же вероятности P_{mi}, P_{fai}, что и при отсутствии атаки), если длина N ПСП будет увеличена в K раз.

Однако оптимальной стратегией коалиции, состоящей из K членов, является выбор такого уровня аддитивного шума, при котором обеспечивается заданное отношение сигнал/шум η_a после атаки. Выразим параметр η_ε через η_a и η_w, представленных выражениями (16), (17), и подставим его в (19):

$$P_e = Q\left(\frac{1}{2}\sqrt{\frac{N}{K^2\eta}}\right), \qquad (20)$$

где $\eta = \eta_w / \eta_a$.

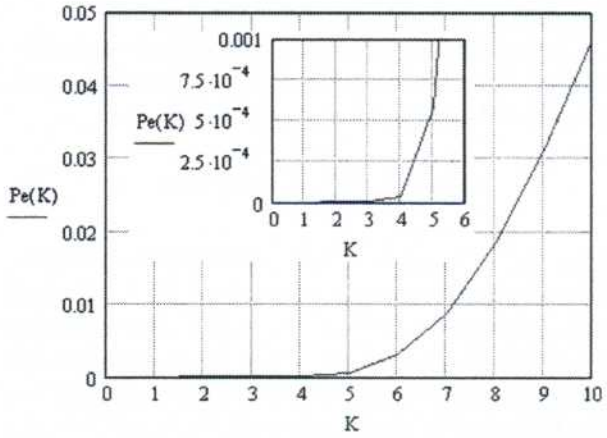


Рис. 1. Зависимость вероятности ошибки обнаружения пирата от числа участников коалиции, $Pe(K)$ при $K < 10$

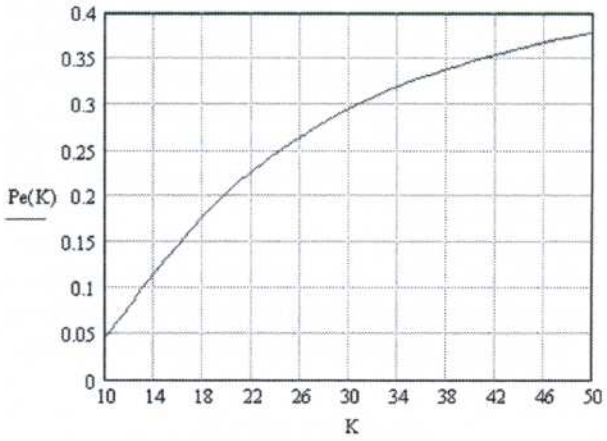


Рис. 2. Зависимость вероятности ошибки обнаружения пирата от числа участников коалиции, $Pe(K)$ при $10 < K < 50$

Из выражения (20) видно, что даже если атакующая коалиция выбирает $\eta = 1$ (а делать искусственно $\eta < 1$ не имеет практического смысла и является даже подозрительным, свидетельствуя о присутствии коалиционной атаки), то компенсация коалиционной атаки группой из K "пиратов" требует увеличения N (а следовательно, и снижения объема вложения) в K^2 раз.

На рис. 1 представлена зависимость $P_{mi} = P_{fai} = P_e$ по (12), (13) при выборе порога λ_o по (14) от числа участников коалиции $K < 10$ при типичной дисперсии изображения $\sigma^2_c = 2500$, а также при выборе параметров $\alpha = 5$, $\sigma^2_\varepsilon = 5$, $N = 1024$.

На рис. 2 представлена зависимость P_e по упрощенной формуле (20) от числа участников коалиции $10 \le K \le 50$ при выборе $N = 1024$ и $\eta = 1$.

На рис. 3 показаны соответственно исходное изображение, изображение после вложения ЦВЗ при $\alpha = 5$, $N = 1024$ и после коалиционной атаки с $\eta = 1$ (при $K = 5, \sigma^2_\varepsilon = (1 - \frac{1}{K})\alpha^2$).

Для более эффективной защиты от коалиционной атаки собственник продукта может использовать при вложении ЦВЗ по (1) *ортогональные последовательности* $\pi_i(n)$, $i = 1, 2 \ldots L$ т.е. такие, что:

$$\sum_{n=1}^{N} w_i(n) w_j(n) = \begin{cases} \alpha^2 N, i = j \\ 0, i \neq j \end{cases}. \qquad (21)$$

Тогда вместо соотношений (10), (11) получим:

$$E\{\Lambda_{i1}\} = \frac{\alpha^2 N}{K}; E\{\Lambda_{i0}\} = 0, \qquad (22)$$

$$Var\{\Lambda_{i1}\} = Var\{\Lambda_{i0}\} = N\sigma^2_\varepsilon \alpha^2. \qquad (23)$$

Подставляя (22), (23) в (8) и (9), будем иметь:

$$P_{mi} = Q\left(\frac{\frac{\alpha^2 N}{K} - \lambda}{\sqrt{N\sigma^2_\varepsilon \alpha^2}}\right), \qquad (24)$$

$$P_{fai} = 1 - Q\left(\frac{-\lambda}{\sqrt{N\sigma^2_\varepsilon \alpha^2}}\right). \qquad (25)$$

Выбирая порог λ по (15), получим совпадение (24) и (25), т.е.:

$$P_{mi} = P_{fai} = P_e = Q\left(\frac{1}{2}\sqrt{\frac{N}{K^2 \eta_\varepsilon}}\right). \qquad (26)$$

Выражая аналогично предыдущему параметр η_ε через η_a и η_w, будем иметь из (26):

$$P_e = Q\left(\frac{1}{2}\sqrt{\frac{N}{K^2 \eta - K}}\right). \qquad (27)$$

Сравнивая (20) (27), видим, что при больших величинах K, их разница оказывается небольшой, и, следовательно, переход к ортогональным ЦВЗ не дает существенной защиты от коалиционной атаки.

Таблица 1

Расчет вероятности ошибки P_e по теории и моделированием

K	Моделирование	Расчет
5	0	$1.91 \cdot 10^{-4}$
15	0.012	0.159
20	0.046	0.230
25	0.082	0.280
30	0.092	0.316
35	0.151	0.343
40	0.166	0.364
45	0.180	0.38
50	0.221	0.393

Результаты имитационного моделирования для вероятностей ошибки P_e при вложении ЦВЗ по (1), коалиционной атаки по (2), принятия решения по (3) и при выборе порога $\lambda = \lambda_o$ по (14) для исходных параметров, как в предыдущих примерах и для различного числа участников коалиции K приведены в таблице 1.



Рис. 3. Исходное изображение, изображение после вложения ЦВЗ, изображение после коалиционной атаки

В этой же таблице показаны значения P_e для теоретических расчетов по формулам (12), (13) при $\lambda = \lambda_o$ по (14).

На основании проведенных расчетов можно сделать следующие выводы

1. Теоретические формулы в основном подтверждаются результатами имитационного моделирования.

2. Качество изображения, определяемое как по отношению сигнал/шум (см. 16), так и непосредственно визуальным сравнением (см. рис. 3), несколько ухудшается после вложения ЦВЗ, но даже улучшается после коалиционной атаки (см. (17) и рис. 3).

3. Приемлемая вероятность ошибки (около 10^{-3}) достигается, как это видно из рис. 1, при объединении в коалицию не более 5[ти] нечестных покупателей продукта. Однако, учитывая, что общее количество пикселей изображения имеет порядок 60000, можно увеличить N, по крайней мере, в 60 раз, что позволит довести размер отслеживаемой коалиции до 40.

3. Оценка вероятностей ошибок Pm и Pfa при использовании "слепого" декодера

Пусть вложение ЦВЗ выполняется по тому же методу (1), что и ранее, коалиционная атака выполняется в соответствии с (2), но при обнаружении декодером i-го "пирата" исходный продукт $C(n)$, $n = 1, 2 \ldots N$ оказывается неизвестным, а известно лишь его среднее значение $m_c = E\{C(n)\}$, которое во многих случаях полагается равным нулю. Тогда обнаружение присутствия i-го пользователя в коалиции S_c может быть проведено по следующему правилу:

$$i \in S_c, \text{ если } \tilde{\Lambda}_i > \lambda,$$
$$i \notin S_c, \text{ если } \tilde{\Lambda}_i \le \lambda, \qquad (28)$$

где $\tilde{\Lambda}_i = \sum_{n=1}^{N} (C'_w(n) - m_c) w_i(n), i = 1,2 \ldots N.$

Вероятности ошибок P_{mi}, P_{fai} рассчитаем по формулам (8), (9), где, однако, необходимо заменить $\Lambda_{i1(0)}$ на $\tilde{\Lambda}_{i1(0)}$ и рассчитать средние значения и дисперсии для последней случайной величины.

Следуя технике, аналогичной той, которая использовалась при выводе (10), (11), находим:

$$E\{\tilde{\Lambda}_{i1}\} = \frac{\alpha^2 N}{K}; E\{\tilde{\Lambda}_{i0}\} = 0,$$

$$Var\{\tilde{\Lambda}_{i1}\} = N(\sigma_c^2\alpha^2 + \sigma_\varepsilon^2\alpha^2 + \frac{\alpha^4(K-1)}{K^2});$$
$$Var\{\tilde{\Lambda}_{i0}\} = N(\sigma_\varepsilon^2\alpha^2 + \sigma_c^2\alpha^2 + \frac{\alpha^4}{K}). \qquad (29)$$

Для большого размера коалиции ($K>>1$), выбирая $\lambda = \frac{\alpha^2 N}{2K}$, получим из (8), (9):

$$\tilde{P}_{mi} = \tilde{P}_{fai} = \tilde{P}_e =$$
$$= Q\left(\frac{\alpha^2 N}{2K\sqrt{N(\sigma_c^2\alpha^2 + \sigma_\varepsilon^2\alpha^2 + \frac{\alpha^4}{K})}} \right). \qquad (30)$$

Преобразуем (30):

$$\tilde{P}_e = Q\left(\frac{1}{2}\sqrt{\frac{N}{K^2(\frac{\sigma_c^2}{\alpha^2} + \frac{\sigma_\varepsilon^2}{\alpha^2}) + K}} \right) = \qquad (31)$$

$$= Q\left(\frac{1}{2}\sqrt{\frac{N}{K^2\eta_w + K^2\eta - K + K}} \right) =$$
$$= Q\left(\frac{1}{2}\sqrt{\frac{N}{K^2(\eta_w + \eta)}} \right). \qquad (32)$$

Выбирая типичный случай $\eta = 1$, будем иметь окончательно:

$$\tilde{P}_e \approx Q\left(\frac{1}{2}\sqrt{\frac{N}{K^2\eta_w}} \right), \qquad (33)$$

поскольку $\eta_w >> 1$.

Сравнивая (33) с (20), видим их "драматическую" разницу в том смысле, что "слепой" декодер приводит по сравнению с информированным к увеличению N в η_w раз для обеспечения такой же вероятности ошибки. Так, для условий примера, рассмотренного в конце предыдущего раздела, получим, что при выборе $N \approx 60000$ и типичной величине $\eta_w = 100$ вероятность ошибки $\tilde{P}_e \approx 10^{-3}$ достигается при размере коалиции $K \approx 4$, что в 10 раз меньше, чем при информированном декодере.

В литературе по ЦВЗ известно два основных метода, позволяющих получить эффективность слепого декодера, приближающуюся к эффективности информированного декодера. Это *улучшенная ШПС* (ISS) [4] и *проективная квантованная модуляция* (QPD) [5]. В первом случае вложение ЦВЗ производится по правилу:

$$C_{wi}(n) = C(n) + (\beta - \alpha x_i)w_i(n),$$
$$n = 1, 2 \ldots N,\ i = 1, 2 \ldots L, \qquad (34)$$

где $x_i = \frac{1}{N\alpha^2}\sum_{n=1}^{N} C(n)w_i(n)$, β – некоторые параметры.

Сравнивая (34) с (1), видим, что в ISS коэффициент вложения ЦВЗ зависит от ПС $C(n)$, что позволяет при отсутствии коалиционной атаки частично скомпенсировать его воздействие на результат декодирования. Однако данный эффект не достигается при коалиционной атаке, поскольку после нее $E\{\Lambda_{i1}\} = \frac{\beta}{K}$, т.е. уменьшается в K раз по сравнению с отсутствием такой атаки. Аналогичная ситуация складывается и при использовании QPD. Данные теоретические выводы были также подтверждены результатами имитационного моделирования, которое показало пренебрежимо малый эффект улучшения системы ЦВЗ при переходе к ISS или QPD по сравнению с простым "слепым" декодером.

4. Заключение

В статье выведены теоретические формулы для оценки вероятностей ошибок при отслеживании участников пиратской коалиции, нелегально распространяющей изображения. Из данных соотношений видно, что при выполнении участниками коалиции оптимальной атаки усреднением индивидуальных продуктов с добавлением аддитивного шума оптимального уровня, обнаружение участников коалиции информированным декодером возможно при небольших размерах коалиции. Для типичных изображений с $8^{\text{ми}}$ битовыми градациями серого и хорошим качеством после вложения ЦВЗ и атаки эти величины K должны быть ограничены величиной порядка 30-40. Использование в качестве ЦВЗ ортогональных сигналов несколько увеличивает эффективность системы. Однако при больших величинах K это улучшение оказывается незначительным. Теоретические расчеты вероятностей ошибок, а также оценка качества изображений подтверждается результатами имитационного моделирования.

Также были получены теоретические формулы для расчета вероятностей ошибок обнаружения участников коалиции при использовании собственником продукта "слепого" декодера, в котором исходное изображение неизвестно. Они показали значительное сокращение размеров допустимой коалиции (примерно в 10 раз по сравнению с информированным декодером). Было показано также, что использование таких известных методов, как ISS и QPD не дает ощутимого эффекта в условиях коалиционной атаки. Поэтому данную проблему можно отнести пока к числу открытых. Правда, некоторым "утешением" можно считать то обстоятельство, что при отслеживании "пиратов" собственник продукта, как правило, имеет в своем распоряжении его оригинал (до внесения ЦВЗ).

СПИСОК ЛИТЕРАТУРЫ

1. **Cox, I.** Digital Watermarking/ I. Cox, et al. – Morgan – Kaufman publishers, 2002.

2. **Ray Liu, K.J.** Multimedia Fingerprinting Forensics for Traitor Tracing/ K.J. Ray Liu, Wade Trappe, Z. Jane Wang, Min Wu, Hong Zhao. – Hindawi Publishing Corporation, 2005.

3. **Гнеденко, Б.В.** Курс Теории вероятностей/ Б.В. Гнеденко. – М.: ФМ, 1961.

4. **Malvar, H.** Improved Spread Spectrum: A modulation technique for robust watermarking/ H. Malvar, D. Florencia // IEEE transaction on signal processing. – 2001. – №51. – P. 898.

5. **Perez-Freire.** Spread-spectrum versus quantization-based data hiding: Misconceptions and implications/ Perez-Freire, F. Perez-Gonzales. SPIE 17^{th} Symposium. – 2005.

Суслов Н.В.

Принципы построения виртуального обучающего пространства

На сегодняшний день существует много программных средств по созданию мультимедийного содержания. Это редакторы по работе с видео, 3-х мерной графикой, звуком, интерактивной анимацией и т.д., широко применяемые для подготовки учебного материала. Существуют также авторизированные средства по сведению готового материала, но они ориентированы в основном на представление бизнес-содержания. Программ по сведению готового мультимедийного содержания в реальном времени, с поддержкой сетевого распределенного взаимодействия и программным интерфейсом, доступным учителю, – нет. В данной статье мы рассмотрим основные принципы, следуя которым, ведется разработка инструментария по проектированию виртуальных обучающих пространств. Этот инструментарий может применяться также в качестве "микшерного, или учительского пульта" по организации мультимедиа, материала в реальном времени в учебном процессе [1].

Большинство существующих программных решений, ориентированных на создание виртуальных обучающих пространств, исповедуют бизнес-логику. В основном эти системы порождают комплексы для обучения, основывающиеся на шаблонном представлении содержания. Так, получили широкое распространение системы управления обучением (Learning Management Systems - LMS/LCMS), системы управления курсами (Cource Learning Management Systems - CLMS). Примерами таких систем являются: Lotus Learning Management System, Oracle RealTime Collaboration, Cisco Meeting Place, Macromedia Breeze , Moodle, MediaWiki и др. Данные средства нацелены в первую очередь на создание информационных баз данных, презентаций, проведения тестирований, сертификаций, то есть как можно более выгодно "продать" внешнюю форму, не заботясь о раскрытии смысла. Использование их в учебном процессе сводится к работе с текстом, показу слайдов, анимационных роликов, внедренных лабораторных экспериментов. Эксперимент проводит либо учитель, либо сам обучаемый, но не вместе. Примером такой системы LCMS может быть центр компьютерного обучения "Специалист" (www.specialist.ru). Все они порождают и жестко закрепляют форму представления содержания, навязывая бизнес-логику. Также для систем на базе LCMS характерна недоступная пользователю по сложности серверная установка, программирование в специализированных средах, закрытый или усложненный код приложения и т.д.

Учитель же заинтересован в обратном: содержание должно порождать психологическикомфортную форму представления знаний. Содержание и форма не должны зависеть от аппаратной платформы (настольный компьютер, сервер, телефон), языка программирования, операционной системы, системы управления содержанием. Не зависеть от программистов, и системных администраторов, и людей, не сведущих в предметной области учителя.

Изложим основные принципы, в соответствии с которыми ведется разработка нового программного обеспечения [2].

Генеративное, или человеко-ориентированное, программирование, где основными понятиями являются "открытость", "репликация", "самопорождение", дает возможность пересмотреть средства создания виртуальных обучающих пространств коренным образом. Технология движется по пути к полному отказу от таких терминов, как "информация", "данные", "теория передачи информации", "память" в любых формах ее представления и соответственно от операций "сохранить", "извлечь", "удалить", "защитить" и т.д. Новые понятия "репликация" и "самопорождение" призывают мыслить в терминах "реального времени", где всякий порождаемый объект генерируется заново, в каждый момент времени своей жизни.

Возможность сохранения или удаления объекта является признаком "мертвого кода". Свойство "самопорождения объекта" невозможно без свойства "открытости" системы в целом. "Открытость" кода гарантирует запущенной программе возможность самоизменения

и модификации во время своего исполнения, что прямо соответствует философии порождающего программирования по аналогии с живым организмом в биологии. Тогда такая программа может быть настроена на неограниченное количество сенсоров, снимающих показатели окружающей среды, и в ходе работы адаптироваться к ней, тем самым гармонично отвечая запросам человека. Только возможность непредопределенной самогенерации позволит моделирующемуся в реальном времени "виртуальному миру" адекватно совмещаться с реальностью, окружающей человека, где человек в итоге уже будет взаимодействать не с компьютером-роботом, а с окружающей его "расширенной реальностью", частью которой является и он сам. Главной частью программы будет являться сам человек с его внутренним и внешним реальным миром.

Термины "хакер", "пользователь", "вирус" изменят смысл, так как "хакерами" будут все "пользователи", а программы на основе "самогенерации" и будут основаны на "вирусах". "Вирус" станет критерием "живого кода". Программы будут формировать во времени собственную иммунную систему и в случае необходимости генерировать биллионы антител, готовых защитить от внешнего и внутреннего агрессоров. Вирусы будут составлять основу для антител, в последующем их будут называть ретровирусами.

Сейчас все привыкли к тому, что в компьютере существует программа в виде листинга или откомпилированная, которая может быть запущена на виртуальной машине ее языка программирования. Причем виртуальная машина языка также заранее откомпилирована и представляет собой "мертвый код", который всегда однозначно выдает один и тот же результат вычислений. Виртуальная машина "живого кода" не будет никогда откомпилирована и сможет в процессе выполнения программы менять собственный язык программирования, что сделает ее полностью неуязвимой от природоразрушающих агрессивных воздействий и будет гарантировать сохранение главного свойства – "открытости". Операционных систем и виртуальных машин будет ровно столько, сколько компьютерных программ и их копий, и ни одной повторяющейся версии. Тем самым отпадет необходимость в защите авторских прав на программную среду и порождаемое ими содержание.

"Репликация" позволяет в каждом узле сети генерировать один в один модели всех остальных узлов, участвующих в соединении [3]. Репликация, или разделение объектов, между группой пользователей в виртуальном обучающем пространстве возможна при применении распределенных вычислений в равноправной сети, например P2P. Это технология построения распределенной сети, где каждый узел может одновременно выступать как в роли клиента (получателя информации), так и в роли сервера (поставщика информации). Идея, заложенная в репликацию, следующая: объекты имеют определенный собственный набор поведений, которые активируются при операции обмена сообщениями. В основе лежат распределенные вычисления, и в первую очередь они служат для моделирования параллельно идущих процессов, асинхронно обменивающихся сообщениями. Сообщения, передающиеся по сети, не содержат прямых данных или результатов вычислений, процессы обмениваются только запросами на вычисления. Каждый узел сети содержит всю информацию о моделируемом виртуальном мире. Любое активное действие, произведенное на участников, взаимодействия мгновенно, реплицируется по сети и вызывает изменение состояний остальных.

Виртуальное обучающее пространство должно быть пронизано многообразием мультимедиа-сред, предоставлять возможность в интерактивном режиме взаимодействовать с его содержимым. Из этого принципа следует предпочтительность модульно-структурной организации инструментария. При этом модули являются как независимыми приложениями, так и программными компонентами, дополняющими друг друга и образующими единую среду создания и запуска виртуальных обучающих пространств. Они разрабатываются на открытых технологиях Open Source, мультиплатформенны и запускаются без инсталляции, позволяют проводить импорт мультимедиа содержания (2D, 3D, звук, видео и т.д.) реального времени, например, на основе библиотек OpenGL, OpenAL, Cairo. Поддержка русского (и любого национального) языка осуществляется через Unicode (стандарт кодирования символов, позволяющий представить знаки практически всех письменных языков). Для разметки содержания и симуляций создается встроенный язык программирования высокого уровня. Программы включают в себе одновременно

как среду разработки, так и среду запуска виртуальных обучающих пространств. Программы должны быть нетребовательны к аппаратной платформе и операционным системам.

Примерами уже созданных приложений на основе выше изложенных принципов, являются следующие.

1. Мультимедиа-диск (2D версия) "Уистлер и Россия" издан в Государственной Третьяковской галерее [4]. В настоящее время ведется разработка 3D версии на основе" Крестьянство" (рис. 1) с поддержкой совместной работы пользователей по сети в 3-х мерном пространстве, где каждый элемент содержания будет доступен для совместного обозрения и взаимодействия.

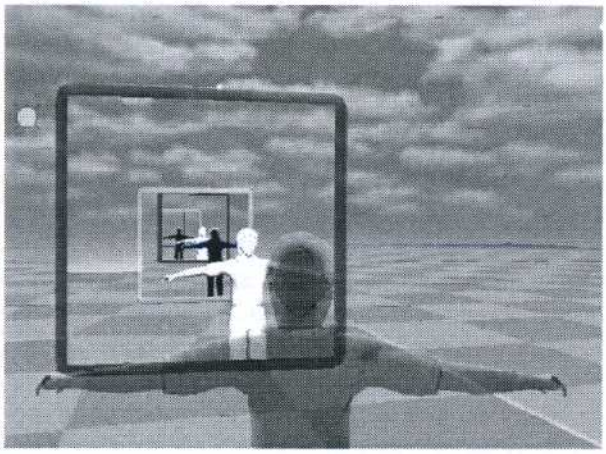


Рис. 1. Снимок экрана из 3D версии "Крестьянство"

2. На основе среды "Крестьянство/Смотрины" создан мультимедиа-диск "А.А. Иванов. Рисунок. Акварель. Библейские эскизы" (рис. 2). Диск позволяет работать с содержанием как одному, так и неограниченному числу пользователей одновременно посредством локальной сети или Интернет; создавать аннотации к произведениям, новые альбомы и сохранять их непосредственно в структуре диска; встроенная поддержка полнотекстового поиска. Диск не требует инсталляции, запускается с CD-Rom или переносного носителя и работает по аналогии с веб-сервером/ базой данных [5].

3. На основе среды "Крестьянство/Игрище" была создана мультимедиа-инсталляция "А.Иванов. Библейский эскизы" [6] и показана в государственной Третьяковской галерее в Москве (рис. 3) и государственном Русском музее в Санкт-Петербурге.



Рис. 2. Снимок экрана из диска "А.А. Иванов"

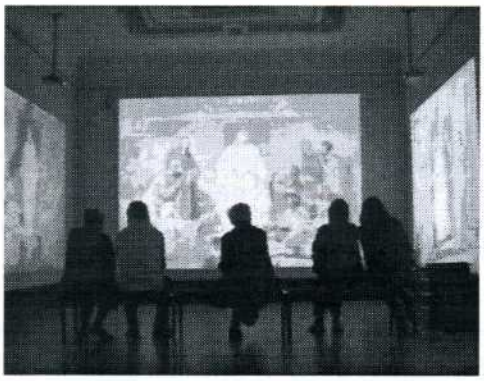


Рис. 3. Фото с инсталляции в залах ГТГ

Проект был поддержан Фондом Потанина для показа в более чем 10 городов России и получил диплом 1 степени АНО "Культурное наследие в цифровых технологиях" АДИТ – премия.

Это проект расширенной реальности, где сеть из компьютеров моделирует помещение с росписями фресок художника в залах галереи. Четыре компьютера, соединенных в сеть, смогли превратить пространство одного из залов галереи в исследовательскую лабораторию творчества художника [7]. На три активные стены проецируются стены виртуального пространства соответственно с полным совмещением с геометрией реальных. Сенсорная панель обеспечивает пульт управления виртуальным пространством. Посетитель, находящийся в центре зала перед сенсорной панелью, может с помощью нее управлять виртуальным освещением, менять конфигурации размещения изображений на стенах, изменять геометрию

помещения, например, путем добавления виртуальных углов, окон и т.д. отражающиеся и на звуковом сопровождении (акустика виртуального помещения); сохранять наиболее удавшиеся состояния и просматривать созданные искусствоведами. Пульт управления может быть не один и находиться не обязательно в экспозиции, что позволит управлять действом удаленно и сразу нескольким людьми одновременно. Самих же активных стен может быть так же неограниченное число, что позволит превратить в единый живой организм всю галерею целиком, включая фасады с внешней стороны. Приходящий человек становится основным звеном вычислений медиа-здания галереи. Главным становится не сам факт наличия того или иного произведения, а та формула художника, которую человек сможет применить к окружающему его миру в реальной жизни. Новые технологии сами собой предвещают рождение человека и общества нового типа, самовозникшего на принципах открытости, бескорыстности, коллективного благородного труда и творчества.

В заключение перечислим вышеизложенные принципы как необходимые условия существования и развития виртуальных обучающих пространств.

Принцип физического моделирования является необходимым условием для объективной виртуализации пространства-времени.

Принцип открытости является необходимым условием эволюционирования виртуального обучающего пространства как живой среды.

Принцип самопорождения является необходимым условием для возможности самоизменения и модификации виртуального обучающего пространства во времени.

Принцип репликации является необходимым условием обеспечения совместных действий в виртуальном обучающем пространстве.

Принцип реального времени является необходимым условием объединения среды разработки и среды запуска виртуальных обучающих пространств.

Таким образом, виртуальные обучающие пространства, созданные на этих принципах, позволят порождать, по новому формировать содержание учебного предмета в реальном времени, будут являться лабораториями для проведения естественнонаучных и педагогических экспериментов.

СПИСОК ЛИТЕРАТУРЫ:

1. **Суслов Н.В.** Виртуальное обучающее пространство "Крестьянство" // Молодые исследователи – регионам. Материалы Всероссийской научной конференции студентов и аспирантов. В 2-х т. – Вологда, ВоГТУ, 2005 г. Т.1. С. 349–350

2. Проект Российского фонда фундаментальных исследований № 07-07-00332 Виртуальное обучающее пространство – "Крестьянство". 2007–2008 г. // Создание и развитие ИВТР для фундаментальных исследований.

3. **Суслов Н.В.** Репликация виртуального обучающего пространства как необходимое условие его существования. Принципы методологии "Крестьянство" // Вузовская наука региону. Материалы 5-й Всероссийской научно-технической конференции. В 2-х.т.-Вологда: ВоГТУ, 2007. – Т.1. С. 357–359

4. **Суслов Н.В.** Мультимедиа CD-Rom "Уистлер и Россия". Москва. Государственная Третьяковская галерея. 2006

5. **Суслов Н.В.** Мультимедиа CD-Rom "А.А. Иванов. Рисунок. Акварель. Библейские Эскизы". Москва. Государственная Третьяковская галерея. 2007 г.

6. **Суслов Н.В.** Мультимедиа-инсталляция "А.А. Иванов. Библейские Эскизы". Москва. Государственная Третьяковская галерея. 2006–2008 г.

7. **Суслов Н.В.** "Игрище": от сценария к действу. // Информатизация процессов формирования открытых систем на основе СУБД, САПР, АСНИ и систем искусственного интеллекта: Материалы 4-й межд. научно-техн.конф. – Вологда: ВоГТУ, 2007. С. 212–216

СВЕДЕНИЯ ОБ АВТОРАХ

АЛИКОВ Алан Юрьевич – *доцент кафедры систем автоматизированного проектирования Северо-Кавказского горно-металлургического института (государственного технологического университета), кандидат технических наук, доцент.*

АЛЕКСЕЕВ Александр Иванович – *аспирант кафедры информационных систем и технологий Северо-Кавказского государственного технического университета.*

АХИН Марат Халимович – *магистрант кафедры автоматики и вычислительной техники Санкт-Петербургского государственного политехнического университета.*

БЕРЕЗКИН Александр Александрович – *аспирант кафедры основ передачи дискретных сообщений Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А.Бонч-Бруевича.*

БИРЯЛЬЦЕВ Евгений Васильевич – *заместитель генерального директора по науке и новым технологиям, ЗАО «Градиент», кандидат технических наук.*

БУЗЮКОВА Ирина Львовна – *аспирантка кафедры сетей связи Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича.*

БУХАНОВСКИЙ Александр Валерьевич – *директор НИИ НКТ, профессор кафедры информационных систем Санкт-Петербургского государственного университета информационных технологий, механики и оптики, доктор технических наук.*

БЫЛИН Алексей Романович – *заведующий лабораторией кафедры систем связи топливно-энергетического комплекса Петербургского энергетического института повышения квалификации.*

БЫЛИНА Мария Сергеевна – *доцент кафедры линий связи Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича, кандидат технических наук.*

ВЕРДИЕВ Мурад Микаилович – *аспирант кафедры вычислительной техники Дагестанского государственного технического университета.*

ГАЙДАМАКА Юлия Васильевна – *доцент кафедры систем телекоммуникаций Российского университета дружбы народов (г. Москва).*

ГЛАГОЛЕВ Сергей Федорович – *декан факультета многоканальных телекоммуникационных систем, профессор кафедры линий связи Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича, кандидат технических наук, доцент.*

ГУЗЕНКО Константин Викторович – *доцент кафедры радиофизики Санкт-Петербургского государственного политехнического университета, кандидат физико-математических наук.*

ДАНИЛОВИЧ Дмитрий Андреевич – *аспирант кафедры радиопередающих устройств и средств подвижной связи Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича.*

ДРОНЬ Вера Олеговна – *инженер ООО "Специальный технологический центр" (г. Санкт-Петербург).*

ДУНАЕВ Антон Валентинович – *младший научный сотрудник НИИ Наукоемких компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики (СПбГУ ИТМО).*

ЗАЙЦЕВ Эрнст Федорович – *профессор кафедры радиофизики Санкт-Петербургского государственного политехнического университета, кандидат технических наук, профессор.*

ЗАХАРОВ Алексей Владимирович – *аспирант кафедры автоматики и вычислительной техники Санкт-Петербургского государственного политехнического университета.*

ИГИЗЬЯНОВА Надежда Александровна – *старший преподаватель кафедры математики и вычислительной техники Южно-Уральского государственного университета (г. Челябинск).*

ИСМАИЛОВ Шейх-Магомед Абдуллаевич – *профессор Дагестанского научного центра Российской академии наук (г. Махачкала), доктор технических наук.*

ИЦЫКСОН Владимир Михайлович – *доцент кафедры автоматики и вычислительной техники Санкт-Петербургского государственного политехнического университета, кандидат технических наук.*

КАНУННИКОВА Елена Александровна – *аспирантка кафедры электротехники и автоматики Белгородского государственного технологического университета им. В.Г. Шухова.*

КАРПЕНКО Анатолий Владимирович – *студент кафедры автоматики и вычислительной техники* Санкт-Петербургского государственного политехнического университета.

КИПОТЬ Виктор Леонидович – *старший научный сотрудник Научно-исследовательского института математики и механики имени Н.Г. Чеботарева Казанского государственного университета, кандидат технических наук.*

КОГНОВИЦКИЙ Олег Станиславович – *заведующий кафедрой обработки и передачи дискретных сообщений Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича, кандидат технических наук, профессор.*

КОЛТУНОВ Леонид Иванович – *руководитель центра информационных технологий Белгородского государственного технологического университета им. В.Г. Шухова.*

КОРЖИК Валерий Иванович – *профессор кафедры информационной безопасности телекоммуникационных систем Санкт-Петербургского государственного университета телекоммуникаций им.проф. М.А. Бонч-Бруевича, доктор технических наук, профессор.*

КОТЛЯРОВ Всеволод Павлович – *профессор кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.*

КУДРЯШОВ Денис Алексеевич – *магистр техники и технологии, аспирант кафедры радиотехники и телекоммуникации Санкт-Петербургского государственного политехнического университета.*

КУКУНИН Дмитрий Сергеевич – *аспирант кафедры обработки и передачи дискретных сообщений Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича.*

КУРАКИН Антон Сергеевич – *ассистент кафедры интернет–технологии МАТИ – Российского государственного технологического университета имени К.Э.Циолковского.*

ЛАВРЕНТЬЕВ Никита Владимирович – *студент кафедры радиотехники и телекоммуникаций Санкт-Петербургского государственного политехнического университета.*

ЛАВРОВ Александр Петрович – *профессор кафедры радиофизики Санкт-Петербургского государственного политехнического университета, доктор физико-математических наук.*

ЛАРЧЕНКО Алексей Викторович – *младший научный сотрудник НИИ Наукоемких компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики.*

ЛУКЬЯНОВА Людмила Михайловна – *доцент кафедры систем управления и вычислительной техники Калининградского государственного технического университета, кандидат технических наук, доцент.*

МЕЗЕНЦЕВА Оксана Станиславовна – *доцент кафедры информационных систем и технологий Северо-Кавказского государственного технического университета, кандидат физико-математических наук, доцент.*

МАГОМЕДОВ Шамиль Гасангусейнович – *аспирант кафедры вычислительной техники Дагестанского государственного технического университета.*

МАКАТОВ Мурад Русланович – *старший преподаватель кафедры вычислительной техники Дагестанского государственного технического университета.*

МАКСИМОВ Максим Игоревич – *старший преподаватель кафедры теории электрической связи Московского технического университета связи и информатики.*

МАНЦЕРОВ Дмитрий Ирикович – *аспирант кафедры информатики Санкт-Петербургского государственного университета.*

МАРКОВ Алексей Михайлович – *ассистент кафедры радиоэлектронных средств защиты информации Санкт-Петербургского государственного политехнического университета.*

МОИСЕЕВ Михаил Юрьевич – *старший преподаватель кафедры автоматики и вычислительной техники Санкт-Петербургского государственного политехнического университета.*

МОЛДАВАНОВА Евгения Николаевна – *магистр техники и технологии кафедры радиофизики Санкт-Петербургского государственного политехнического университета.*

МОЛОДЯКОВ Сергей Александрович – *докторант, доцент кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.*

МЯСНОВ Александр Владимирович – *ведущий электроник телекоммуникационного центра факультета технической кибернетики Санкт-Петербургского государственного политехнического университета.*

НАСПЕР Максим Львович – *студент кафедры радиотехники и телекоммуникаций Санкт-Петербургского государственного политехнического университета.*

НАУМОВ Александр Сергеевич – *аспирант кафедры компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики, инженер ООО "Специальный технологический центр".*

НОВИКОВ Константин Владимирович – *ассистент кафедры теоретических основ радиотехники Новосибирского государственного технического университета.*

НОВИКОВ Федор Александрович – *доцент кафедры прикладной математики Санкт-Петербургского государственного политехнического университета, кандидат физико-математических наук.*

ПАВЛОВ Дмитрий Алексеевич – *аспирант кафедры прикладной математики Санкт-Петербургского государственного политехнического университета, магистр.*

ПОТАПЕНКО Анатолий Николаевич – *профессор кафедры электротехники и автоматики Белгородского государственного технологического университета им. В.Г. Шухова, кандидат технических наук, доцент.*

ПОТАПОВ Виктор Иванович – *профессор кафедры математики и вычислительной техники Южно-Уральского государственного университета (г. Челябинск), доктор технических наук, доцент.*

РАЧИНСКИЙ Евгений Станиславович – *аспирант кафедры системного программирования Санкт-Петербургского государственного университета.*

РЫЖОВ Василий Александрович – *аспирант кафедры радиофизики Казанского государственного университета.*

САВЕЛЬЕВ Михаил Владимирович – *профессор кафедры электронных вычислительных машин Южно-Российского государственного университета (Новочеркасского политехнического института), доктор технических наук, профессор.*

САМОЧАДИН Александр Викторович – *профессор кафедры интегрированных компьютерных технологий в промышленности Санкт-Петербургского государственного политехнического университета, профессор, кандидат технический наук.*

СИДОРОВА Надежда Александровна – *студентка факультета радио и телевидения Московского технического университета связи и информатики.*

СИДОРОВ Юрий Евгеньевич – *профессор кафедры радиотехники и телекоммуникаций Санкт-Петербургского государственного политехнического университета, доктор технических наук, профессор.*

СТАРОСТЕНКО Анастасия Сергеевна – *аспирантка кафедры интегрированных компьютерных технологий в промышленности Санкт-Петербургского государственного политехнического университета.*

СУСЛОВ Николай Викторович – *аспирант кафедры информационных систем и технологий Вологодского государственного технического университета.*

СУРОВ Алексей Николаевич – *старший преподаватель кафедры общей математики Южно-Уральского государственного университета.*

ТИХОМИРОВ Владимир Александрович – *аспирант кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета.*

ТИХОНОВА Ульяна Николаевна – *аспирантка кафедры прикладной математики, ассистент кафедры прикладной математики Санкт-Петербургского государственного политехнического университета.*

УШМОТКИН Антон Сергеевич – *аспирант кафедры информационной безопасности телекоммуникационных систем Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича.*

ФЕДОСЕЕВ Сергей Владимирович – *докторант Южно-Российского государственного технического университета (Новочеркасского политехнического института), кандидат технических наук.*

ЦЕСЬКО Вадим Александрович – *аспирант кафедры автоматики и вычислительной техники Санкт-Петербургского государственного политехнического университета.*

ЦИКИН Игорь Анатольевич – *заведующий кафедрой радиотехники и телекоммуникаций Санкт–Петербургского государственного политехнического университета, доктор технических наук, профессор.*

ЧЕБЫКИН Сергей Владимирович – *магистр техники и технологии, ассистент кафедры информационных систем и информационного менеджмента Владимирского государственного университета.*

АННОТАЦИИ

Лукьянова Л.М. МЕТОДИКА СТРУКТУРНО-ЦЕЛЕВОГО АНАЛИЗА И СИНТЕЗА ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ КОМПЛЕКСОВ.

Рассмотрена методика структурно-целевого анализа и синтеза организационно-технических комплексов производственной сферы, обеспечивающая логическую корректность и согласованность результатов системного анализа объектов данного класса.

СИСТЕМНЫЙ АНАЛИЗ. МЕТОДИКА. ЛОГИЧЕСКАЯ ПРАВИЛЬНОСТЬ ПРОЦЕДУР. ЛОГИЧЕСКАЯ КОРРЕКТНОСТЬ РЕЗУЛЬТАТОВ.

Савельев М.В., Аликов А.Ю., Федосеев С.В. ВЕРОЯТНОСТЬ СВЯЗНОСТИ СЕТИ С АДДИТИВНОЙ СТРУКТУРОЙ

Рассматриваются сети с аддитивной структурой, построенной итеративным присоединением к ее текущей границе подсетей дополнения. Предложены методы вычисления связности числа элементов в подсетях дополнения.

СВЯЗНОСТЬ. СЕТЬ. СТРУКТУРА. НАДЕЖНОСТЬ. СТРУКТУРНАЯ НАДЕЖНОСТЬ СЕТИ. АППРОКСИМАЦИЯ.

Тихомиров В.А., Котляров В.П. ВЕРИФИКАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПРИ ИНТЕГРАЦИИ НОВОЙ ФУНКЦИОНАЛЬНОСТИ В СУЩЕСТВУЮЩУЮ СИСТЕМУ.

После интеграции новой функциональности в уже существующую систему необходимо тщательное тестирование всей системы, что требует больших затрат. В работе предложен метод решения этой проблемы, основанный на широко известном верификационном подходе – проверка на моделях (model checking) и на теории частичных спецификаций (partial specifications).

ИНТЕГРАЦИОННОЕ ТЕСТИРОВАНИЕ. ЧАСТИЧНЫЕ СПЕЦИФИКАЦИИ. ВЕРИФИКАЦИИ. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ.

Мезенцева О. С., Алексеев А. И. ПРИМЕНЕНИЕ АППАРАТА ПОРОГОВЫХ СХЕМ И МОДУЛЯРНОЙ АРИФМЕТИКИ ДЛЯ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ.

Рассмотрены факторы, влияющие на производительность распределенных систем, основанных на аппарате пороговых схем, предложены методы повышения быстродействия, основанные на использовании параллельных алгоритмов модулярной арифметики, в частности, системы остаточных классов. Рассмотрена аппаратная платформа.

СИСТЕМА ОСТАТОЧНЫХ КЛАССОВ. ПОРОГОВЫЕ СХЕМЫ. РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ. ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ.

Дронь В.О., Марков А.М., Наумов А.С. СОКРАЩЕНИЕ ОБЪЕМА ВЫЧИСЛЕНИЙ ПРИ СЛУЧАЙНОМ ПОИСКЕ В ЗАДАЧАХ ПЕЛЕНГОВАНИЯ.

Рассмотрен вопрос сокращения времени определения направления на источник радиоизлучения путем оптимизации фазово-корреляционной функции методом случайного поиска.

РАДИОПЕЛЕНГАЦИЯ. ТЕОРИЯ ОПТИМИЗАЦИИ. МЕТОД СЛУЧАЙНОГО ПОИСКА. МЕТОД ФАЗОВОЙ КОРРЕЛЯЦИИ.

Березкин А.А. ПОСТРОЕНИЕ ОПТИМАЛЬНЫХ НЕЙРОННЫХ ДЕКОДЕРОВ БЛОКОВЫХ КОДОВ.

В статье рассмотрены основные подходы к построению нейронных декодеров блоковых кодов. Предложены структуры нейронных декодеров некоторых кодов БЧХ.

ДЕКОДИРОВАНИЕ. НЕЙРОННЫЕ СЕТИ. ПЕРЕДАЧА ДАННЫХ. ТЕЛЕКОММУНИКАЦИОННЫЕ СИСТЕМЫ.

Кудряшов Д.А., Цикин И.А. ВЕРОЯТНОСТНО-ВРЕМЕННЫЕ ХАРАКТЕРИСТИКИ ДВУНАПРАВЛЕННОЙ РАДИОСИСТЕМЫ ПЕРЕДАЧИ ТРЕВОЖНЫХ ИЗВЕЩЕНИЙ.

Приводятся аналитические выражения, связывающие вероятностно-временные характеристики РСПТИ с параметрами алгоритма функционирования. Описывается методика имитационного моделирования РСПТИ.

РАДИОСИСТЕМА. ДВУНАПРАВЛЕННЫЕ СИСТЕМЫ. ПОВТОРЕНИЕ ПАКЕТОВ. МОНИТОРИНГ. ТРЕВОЖНЫЕ ИЗВЕЩЕНИЯ. ВЕРОЯТНОСТНО-ВРЕМЕННЫЕ ХАРАКТЕРИСТИКИ.

Сидоров Ю. Е., Наспер М. Л. ОЦЕНКА ЭФФЕКТИВНОСТИ ОБНАРУЖИТЕЛЯ РАДИОСИГНАЛОВ В КАНАЛАХ С ЗАМИРАНИЯМИ ПРИ РАЗНЕСЕННОМ ПРИЕМЕ.

Получено оптимальное правило обнаружения радиосигнала с неизвестными характеристиками замираний. Проведена оценка его эффективности методом статистического моделирования.

ОБНАРУЖИТЕЛЬ. ЗАМИРАНИЯ. ОЦЕНКА ЭФФЕКТИВНОСТИ. РАЗНЕСЕННЫЙ ПРИЕМ. СТАТИСТИЧЕСКОЕ ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ. СИГНАЛ-ШУМ.

Былин А.Р., Былина М.С., Глаголев С.Ф. ПУТИ ПОВЫШЕНИЯ РАЗРЕШАЮЩЕЙ СПОСОБНОСТИ ИМПУЛЬСНЫХ ИЗМЕРЕНИЙ.

В статье проведен анализ возможности использования вместо традиционного однополярного прямоугольного зондирующего сигнала, так называемого «биимпульса», представляющего собой совокупность двух разнополярных прямоугольных импульсов. Показано, что использование «биимпульса» позволяет получить особую «биимпульсную» рефлектограмму. В статье также предложен алгоритм обработки зарегистрированных традиционных рефлектограмм.

ИМПУЛЬСНЫЙ ПРИБОР. ИМПУЛЬСНЫЙ РЕФЛЕКТОМЕТР. РЕФЛЕКТОГРАММА. ЗОНДИРУЮЩИЙ СИГНАЛ. РАЗРЕШАЮЩАЯ СПОСОБНОСТЬ.

Сидоров Ю. Е., Лаврентьев Н.В. ОБНАРУЖИТЕЛЬ РАДИОСИГНАЛОВ В КАНАЛАХ СВЯЗИ С НЕИЗВЕСТНЫМИ ХАРАКТЕРИСТИКАМИ ЗАМИРАНИЙ.

Получено оптимальное правило обнаружения радиосигнала с неизвестными характеристиками замираний. Проведена оценка его эффективности методом статистического моделирования.

ОБНАРУЖИТЕЛЬ СИГНАЛОВ. КАНАЛ СВЯЗИ. ЗАМИРАНИЯ. ОЦЕНКА ЭФФЕКТИВНОСТИ. СТАТИСТИЧЕСКОЕ ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ.

Рыжов В.А., Кипоть В.Л., Биряльцев Е.В. ПАРАМЕТРИЗАЦИЯ СПЕКТРОВ В ТЕХНОЛОГИИ НИЗКОЧАСТОТНОГО СЕЙСМИЧЕСКОГО ЗОНДИРОВАНИЯ НА ОСНОВЕ ВЕЙВЛЕТ АНАЛИЗА.

В статье предлагается решение проблемы оценки информативных параметров спектров микросейсм для технологии поиска нефтяных месторождений низкочастотного сейсмического зондирования. Рассматривается алгоритм параметризации спектральных максимумов, основанный на применение методов локальной оптимизации к вейвлет-образу спектра.

ВЕЙВЛЕТ. ПАРАМЕТР. СПЕКТР. СВЕРТКА. МИКРОСЕЙСМЫ. НСЗ. НЕФТЬ.

Данилович Д.А. ОПТИМИЗАЦИЯ ХАРАКТЕРИСТИК АНТЕНН И ВЫСОТ ИХ ПОДВЕСА НА МНОГОИНТЕРВАЛЬНЫХ ЦИФРОВЫХ РАДИОРЕЛЕЙНЫХ ЛИНИЯХ.

Рассматривается задача совместной оптимизации выбора антенн и высот их подвеса на многоинтервальных цифровых РРЛ с двухчастотными планами распределения частот в условиях наличия дополнительных неформальных ограничений. Разработаны математическая модель, алгоритм и методика нахождения заданного числа ранжированных лучших решений указанной задачи оптимизации.

ДВУХЧАСТОТНЫЙ ПЛАН. СИГНАЛЫ ОБРАТНОГО НАПРАВЛЕНИЯ. СОВМЕСТНАЯ ОПТИМИЗАЦИЯ. МИНИМИЗАЦИЯ СУММАРНЫХ ЗАТРАТ. ОБЪЕМ ВЫЧИСЛЕНИЙ.

Бузюкова И.Л., Гайдамака Ю.В. ИССЛЕДОВАНИЕ СИГНАЛЬНОЙ НАГРУЗКИ ИНТЕЛЛЕКТУАЛЬНЫХ СЕТЕЙ СВЯЗИ С УЗЛАМИ В РАЗНЫХ ЧАСОВЫХ ПОЯСАХ.

В статье анализируются различные конфигурации российской интеллектуальной сети связи федерального уровня с точки зрения требуемых объемов сигнального оборудования. Изменение сетевой конфигурации осуществляется путем введения дополнительных узлов сети и перераспределения нагрузки.

ИНТЕЛЛЕКТУАЛЬНАЯ СЕТЬ. СИГНАЛЬНАЯ НАГРУЗКА INAP. ЧИСЛО ЗВЕНЬЕВ СИГНАЛИЗАЦИИ.

Максимов М.И., Сидорова Н.А. ПРИНЦИПЫ РАБОТЫ И ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ РЕЧЕВОГО КОДЕКА НА ОСНОВЕ МЕТОДА ЛИНЕЙНОГО ПРЕДСКАЗАНИЯ С ВОЗБУЖДЕНИЕМ ОТ КОДА.

Рассмотрена практическая реализация речевых кодеков, совместимых со стандартами IS-96-A и IS-733, на базе цифрового процессора обработки сигналов TMS320VC5410A. Описаны принципы построения кодека на основе метода линейного предсказания с возбуждением от кода (CELP-вокодера). Приведена оценка вычислительных затрат и требуемой памяти для реализации каждого функционального блока кодека.

ВОКОДЕР. ЛИНЕЙНОЕ ПРЕДСКАЗАНИЕ. ЛИНЕЙНЫЕ СПЕКТРАЛЬНЫЕ ПАРЫ. КОДОВАЯ КНИГА. ВОЗБУЖДЕНИЕ ОТ КОДА.

Кукунин Д.С. МОДЕЛЬ СИСТЕМЫ С АДАПТИВНЫМ ВЫБОРОМ КОДА.

В статье описана работа системы с адаптивным кодированием, которая способна выбирать оптимальный циклический код в зависимости от состояния канала связи. Приведены примеры реакции системы на увеличение и уменьшение вероятности ошибки в канале.

ДВОЙСТВЕННЫЙ БАЗИС. ПОЛЕ ГАЛУА. ВЕРОЯТНОСТЬ ОШИБКИ. АДАПТИВНАЯ СИСТЕМА. ДОСТОВЕРНОСТЬ. КАЧЕСТВО КАНАЛА.

Кукунин Д.С., Когновицкий О.С. МЕТОДИКА ОЦЕНКИ КАЧЕСТВА КАНАЛА В ПРОЦЕССЕ ПЕРЕДАЧИ ДАННЫХ.

В статье показаны перспективы использования двойственного базиса поля Галуа применительно к задачам оценки качества канала. Предложена методика определения вероятности ошибки в канале в процессе передачи данных.

ДВОЙСТВЕННЫЙ БАЗИС. ПОЛЕ ГАЛУА. КАЧЕСТВА КАНАЛА. ВЕРОЯТНОСТЬ ОШИБКИ. ДОВЕРИТЕЛЬНЫЙ ИНТЕРВАЛ.

Новиков Ф.А., Тихонова У.Н. ОПРЕДЕЛЕНИЕ ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ ЯЗЫКОВ ИНТЕРПРЕТИРУЕМЫМИ АВТОМАТАМИ.

В статье описан метод определения проблемно-ориентированных языков с помощью интерпретируемых автоматов. Этот метод позволяет формально определить проблемно-ориентированный язык в виде трех составляющих: абстрактного синтаксиса, конкретного синтаксиса и семантики.

ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЙ ЯЗЫК. СЕМАНТИКА. МЕТАМОДЕЛЬ. АВТОМАТ. МАШИНА АВТОМАТНОГО ПРОГРАММИРОВАНИЯ.

Дунаев А.В., Ларченко А.В., Бухановский А.В. ИНСТРУМЕТАЛЬНАЯ ОБОЛОЧКА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ РАЗРАБОТЧИКА ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ПРИЛОЖЕНИЙ В ГРИД-СРЕДЕ.

В статье рассматривается новый подход к автоматизации процесса проектирования высокопроизводительных приложений в Грид-среде. Обосновываются базовые принципы функционирования интеллектуальной системы поддержки принятия решений разработчика, обсуждаются аспекты ее практической реализации в инструментальных оболочках семейства PEG. Представлены оценки эффективности композитных приложений.

ГРИД. ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ ВЫЧИСЛЕНИЯ. ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА. ПРОЕКТИРОВАНИЕ.

Рачинский Е.С. ПРОГРАММНАЯ СРЕДА ДЛЯ ОЦЕНКИ ПРОИЗВОДИТЕЛЬНОСТИ КЛИЕНТ-СЕРВЕРНЫХ СИСТЕМ ПРИ НАГРУЗОЧНОМ ОКРУЖЕНИИ.

В статье рассмотрены основные проблемы изучения производительности клиент-серверных систем под нагрузкой. Проанализированы общие подходы и существующие средства имитации нагрузки и измерения показателей производительности. Описаны основные принципы и идеи нового решения, его реализация и примеры использования.

ПРОИЗВОДИТЕЛЬНОСТЬ. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ. ТЕСТИРОВАНИЕ. ИНФОРМАЦИОННЫЕ СИСТЕМЫ. АНАЛИЗ.

Старостенко А.С., Самочадин А.В. ПРОГРАММНЫЕ СРЕДСТВА ДИСТАНЦИОННОГО УПРАВЛЕНИЯ ОЧЕРЕДЯМИ ПОТРЕБИТЕЛЕЙ.

Рассмотрены проблемы автоматизации процесса прохождения очереди при обслуживании конечных потребителей. Описывается подход к созданию системы управления очередями, основанный на использовании Интернет и SMS сервисов для выполнения основных функций системы.

УПРАВЛЕНИЕ ОЧЕРЕДЯМИ. СИСТЕМА АВТОМАТИЗАЦИИ. ИНТЕРНЕТ-СИСТЕМЫ. МОБИЛЬНЫЕ УСТРОЙСТВА. СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ.

Куракин А. С. ИСПОЛЬЗОВАНИЕ МУЛЬТИАГЕНТНОГО ПОДХОДА ПРИ РАЗРАБОТКЕ КОМПЛЕКСОВ ПРОГРАММ СИСТЕМ КОРПОРАТИВНОГО ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ.

В статье проведен анализ развития современных технологий программных комплексов корпоративного дистанционного обучения, выявлены особенности и специфика разработки и функционирования

таких систем. На основе проведенного анализа предложена мультиагентная структура построения программных комплексов систем дистанционного обучения.

ПРОГРАММНЫЕ КОМПЛЕКСЫ. СИСТЕМА. ДИСТАНЦИОННОЕ ОБУЧЕНИЕ. МУЛЬТИАГЕНТНАЯ СТРУКТУРА.

Ицыксон В.М., Моисеев М.Ю., Цесько В.А., Карпенко А.В. ИССЛЕДОВАНИЕ СИСТЕМ АВТОМАТИЗАЦИИ ОБНАРУЖЕНИЯ ДЕФЕКТОВ В ИСХОДНОМ КОДЕ ПРОГРАММ

Предложена классификация дефектов в исходном коде программ. Проведен сравнительный анализ существующих средств автоматизированного выявления дефектов на основе статического анализа исходного кода. Сформулированы направления дальнейшего развития в области автоматизированного выявления дефектов.

БЕЗОПАСНОСТЬ. ОБНАРУЖЕНИЕ ДЕФЕКТОВ. СТАТИЧЕСКИЙ АНАЛИЗ. АВТОМАТИЗАЦИЯ. КЛАССИФИКАЦИЯ ДЕФЕКТОВ. ПРОГРАММА.

Ицыксон В.М., Захаров А.В., Ахин М.Х., Мяснов А.В. АВТОМАТИЧЕСКОЕ ОБНАРУЖЕНИЕ ДЕФЕКТОВ ПРОГРАММНЫХ СИСТЕМ НА ОСНОВЕ МЕТОДА ПРОВЕРКИ МОДЕЛИ.

Разработан комплексный подход к обнаружению программных дефектов, основывающийся на расширении статического анализа методом проверки моделей. Показана применимость подхода на примере задачи обнаружения ошибок протоколов вызова.

ПРОВЕРКА МОДЕЛЕЙ. СТАТИЧЕСКИЙ АНАЛИЗ. ОБНАРУЖЕНИЕ ДЕФЕКТОВ. ОШИБКИ ПРОТОКОЛОВ ВЫЗОВОВ.

Савельев М.В., Аликов А.Ю., Федосеев С.В. СИНТЕЗ ПРОСТЫХ СТРУКТУР И НАБОРОВ ИМПУЛЬСНЫХ УПРАВЛЯЮЩИХ ВОЗДЕЙСТВИЙ.

В статье рассмотрено совершенствование методов синтеза систем управления с импульсным воздействием при формализации сложности и исключения избыточности структур. В результате выполненной формализации анализируемая задача сводится к задаче поиска простых структур с частично линейными ограничениями.

СИНТЕЗ. ФОРМАЛИЗАЦИЯ. СТРУКТУРА. СИСТЕМЫ УПРАВЛЕНИЯ. УПРАВЛЯЮЩИЕ ВОЗДЕЙСТВИЯ.

Чебыкин С.В. МЕТОДИКА ПРОИЗВОДСТВЕННОГО ПЛАНИРОВАНИЯ ВЫПУСКА ЛИСТОВОГО СТЕКЛА.

В статье рассматриваются вопросы оптимизации производственного планирования выпуска листового стекла. Описывается разработанная автором методика, позволяющая обеспечить эффективное производственное планирование выпуска листового стекла в соответствии с концепцией APS.

ОПТИМИЗАЦИЯ. ПОРТФЕЛЬ ЗАКАЗОВ. ПРОИЗВОДСТВЕННОЕ ПЛАНИРОВАНИЕ. УПРАВЛЕНИЕ.

Лавров А.П., Молодяков С.А. ВОЗМОЖНОСТИ ПОСТРОЕНИЯ ПРОЦЕССОРОВ ОБРАБОТКИ СИГНАЛОВ В ВИДЕ ГИБРИДНЫХ ОПТОЭЛЕКТРОННЫХ МИКРОСХЕМ.

Оптоэлектронные процессоры в виде гибридных микросхем являются новым направлением исследований. Представлены как схемы процессоров, так и схемы систем обработки сигналов. Обсуждаются результаты экспериментального макетирования.

ОПТОЭЛЕКТРОННЫЙ ПРОЦЕССОР. ГИБРИДНАЯ МИКРОСХЕМА. МАТРИЧНЫЕ ОПЕРАЦИИ. КОНВЕЙЕР. ШИРОТНО-ИМПУЛЬСНАЯ МОДУЛЯЦИЯ.

Зайцев Э.Ф., Гузенко К.В., Молдаванова Е.Н. АНАЛИЗ ВОЛН В ФЕРРИТОДИЭЛЕКТРИЧЕСКОМ ВОЛНОВОДЕ ВАРИАЦИОННЫМ МЕТОДОМ.

Решена электродинамическая задача о волнах в электрически управляемом ферритодиэлектрическом волноводе. Реализован алгоритм Ритца в среде MATLAB. Выполнены расчеты для различных конструкций волновода. Результаты применимы для усовершенствования электрически сканирующих антенн миллиметрового диапазона волн.

МИЛЛИМЕТРОВЫЕ ВОЛНЫ. СКАНИРУЮЩАЯ АНТЕННА.ФЕРРИТОДИЭЛЕКТРИЧЕСКИЙ ВОЛНОВОД. ВАРИАЦИОННЫЙ МЕТОД.

Макатов М. Р., Вердиев М. М. ИНФОРМАЦИОННАЯ МОДЕЛЬ ОПИСАНИЯ ОБЪЕКТОВ И СИТУАЦИЙ ПРОБЛЕМНОЙ СРЕДЫ.

Рассмотрена информационная модель описания объектов и ситуаций проблемной среды при решении задач планирования поведения интеллектуальных систем в недоопределенных средах.

ИНФОРМАЦИОННАЯ МОДЕЛЬ. СРЕДА. ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ. ПЛАНИРОВАНИЕ ПОВЕДЕНИЯ.

Исмаилов Ш.А., Магомедов Ш.Г. АЛГОРИТМЫ И СТРУКТУРЫ ПРЕОБРАЗОВАНИЯ ЧИСЛОВЫХ ДАННЫХ ИЗ ПОЗИЦИОННОЙ СИСТЕМЫ СЧИСЛЕНИЯ В СИСТЕМУ ОСТАТОЧНЫХ КЛАССОВ.

В статье рассматриваются проблемы разработки алгоритма разрядно - параллельного преобразования двоично-десятичных чисел в систему остаточных классов. Для преобразования чисел используются таблично - алгоритмические принципы обработки данных.

СИСТЕМА ОСТАТОЧНЫХ КЛАССОВ. БЛОК ПАМЯТИ. УПРАВЛЯЮЩИЙ ИМПУЛЬС. ШИНА СИНХРОНИЗАЦИИ. АССОЦИАТИВНАЯ ЧАСТЬ. ИНФОРМАЦИОННАЯ ЧАСТЬ.

Новиков К.В. БЫСТРОДЕЙСТВУЮЩИЙ АЛГОРИТМ СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ НА ОСНОВЕ СПЕКТРАЛЬНЫХ СВОЙСТВ АНИЗОТРОПНЫХ ОБЛАСТЕЙ.

Рассмотрен помехоустойчивый алгоритм сегментации анизотропных изображений, использующий свойства спектральной плотности мощности анизотропных областей. Предлагаются быстродействующие варианты реализации данного алгоритма.

СЕГМЕНТАЦИЯ ИЗОБРАЖЕНИЙ. ХАРАКТЕРИСТИКИ АНИЗОТРОПИИ. ЛОКАЛИЗИРОВАННОЕ ПРЕОБРАЗОВАНИЕ РАДОНА.

Суров А.Н., Игизьянова Н.А., Потапов В.И. О СХОДИМОСТИ РЕШЕНИЯ РАЗНОСТНОЙ СХЕМЫ В МАТЕМАТИЧЕСКОЙ МОДЕЛИ ГЕНЕРАЦИИ ТЕПЛОВОЙ ЭНЕРГИИ.

В статье рассмотрена математическая модель генерации тепловой энергии при прохождении электрического тока в жидком проводнике – шлаковой ванне с большим сопротивлением. В математической модели рассмотрены дифференциальные уравнения, которые решены методом конечных разностей. Получены численные результаты температурных полей и плотности тока.

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ. МЕТОД КОНЕЧНЫХ РАЗНОСТЕЙ. ТЕМПЕРАТУРНОЕ ПОЛЕ.

Павлов Д.А. АЛГОРИТМ ПОСТРОЕНИЯ РАСШИРЕННОГО УНИВЕРСАЛЬНОГО БАЗИСА ГРЕБНЕРА.

В статье описывается алгоритм построения расширенного универсального базиса Гребнера (EUGB) полиномиального идеала, основанный на связи между многомерными диаграммами Юнга и базисами Гребнера полиномиальных идеалов. Представленный алгоритм предназначен для нульмерных идеалов и включает в себя построение максимальных линейно-независимых множеств мономов (MLIMS) в факторалгебре по идеалу.

БАЗИСЫ ГРЕБНЕРА. УПОРЯДОЧЕНИЯ МОНОМОВ. ДИАГРАММЫ ЮНГА. ПОЛИНОМИАЛЬНЫЕ ИДЕАЛЫ

Манцеров Д.И. СИСТЕМА ВЕРИФИКАЦИИ ДЛЯ ПАРАМЕТРИЧЕСКИХ КЛАССОВ ЗАДАЧ ПО МАТЕМАТИКЕ.

В статье рассмотрена система для создания задач по математике и проверки знаний на основе параметрических классов задач. Изложены основы параметрических задач и структура системы верификации.

ПАРАМЕТРИЧЕСКИЙ КЛАСС ЗАДАЧ. ЭКЗЕМПЛЯР ПАРАМЕТРИЧЕСКОГО КЛАССА. КЛАСС-ШАБЛОН. ПРЕДИКАТЫ КЛАСС-ШАБЛОНА. ГЕНЕРАЦИЯ ПРИМЕРОВ.

Канунникова Е.А., Колтунов Л.И., Потапенко А.Н. ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ РАСПРЕДЕЛЕННЫХ ЭЛЕКТРОТЕХНИЧЕСКИХ СИСТЕМ В ПОЛУБЕЗГРАНИЧНЫХ ОБЛАСТЯХ НА ОСНОВЕ МЕТОДА ИНВЕРСИИ.

В статье рассмотрены возможности модифицированного метода инверсии для полубезграничных сред для решения внешних краевых задач на примере распределенных электротехнических систем. Представлены результаты численного моделирования исследуемых систем и сравнительный анализ с известными данными.

ВНЕШНЯЯ КРАЕВАЯ ЗАДАЧА. ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ. МОДИФИЦИРОВАННЫЙ МЕТОД ИНВЕРСИИ ЭЛЕКТРОТЕХНИЧЕСКИЕ СИСТЕМЫ.

Ушмоткин А.С., Коржик В.И. ОЦЕНКА ЭФФЕКТИВНОСТИ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ ЦИФРОВЫХ «ВОДЯНЫХ» ЗНАКОВ ДЛЯ ИЗОБРАЖЕНИЙ В УСЛОВИЯХ КОАЛИЦИОННЫХ АТАК.

Рассматривается система с вложением цифровых водяных знаков (ЦВЗ) в изображение, использующая широкополосные сигналы. В статье производится оценка эффективности системы ЦВЗ в терминах вероятности ошибок пропуска и ложного обнаружения участников такой коалиции.

ЦИФРОВОЙ ВОДЯНОЙ ЗНАК. АВТОРСКОЕ ПРАВО. ШИРОКОПОЛОСНЫЙ СИГНАЛ. КОАЛИЦИОННАЯ АТАКА. ВЕРОЯТНОСТЬ ОШИБКИ. ДЕТЕКТОР. МОДЕЛИРОВАНИЕ.

Суслов Н.В. ПРИНЦИПЫ ПОСТРОЕНИЯ ВИРТУАЛЬНОГО ОБУЧАЮЩЕГО ПРОСТРАНСТВА.

В статье представлена работа, нацеленная на исследование основных принципов построения виртуальных пространств. Приводятся конкретные примеры созданных обучающих пространств.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ. МУЛЬТИМЕДИА. ВИРТУАЛЬНОЕ ОБУЧАЮЩЕЕ ПРОСТРАНСТВО. РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ. ОТКРЫТЫЕ СИСТЕМЫ.

Annotation

Lukianova L. M. TECHNIQUE OF THE STRUCTURALLY-TARGET ANALYSIS AND SYNTHESIS OF ORGANISATIONAL-TECHNICAL COMPLEXES.

The technique of the structurally-target analysis and synthesis of organizational-technical complexes of the industrial sphere, providing a logical correctness and a coordination of results of the systems analysis of objects of the given class is considered.

SYSTEMS ANALYSIS. TECHNIQUE. LOGICAL PROCEDURES CORRECTNESS. LOGICAL REZULTS CORRECTNESS.

Saveljev M. V., Alikov A. J., Fedoseev S. V. PROBABILITY OF CONNECTIVITY OF THE NETWORK WITH ADDITIVE STRUCTURE.

Networks with the additive structure constructed by iterative connection to its current border sub networks of addition are considered. The probability of connectivity of a network is found. Methods of calculation of connectivity of number of elements in sub networks additions are developed.

CONNECTIVITY. NETWORK. STRUCTURE. RELIABILITY. STRUCTURAL RELIABILITY OF THE NETWORK. APPROXIMATION.

Tikhomirov V. A., Kotljarov V. P. VERIFICATION OF THE SOFTWARE AT INTEGRATION OF NEW FUNCTIONALITY INTO EXISTING SYSTEM.

After integration of new functionality into already existing system careful testing of all system that demands the big expenses is necessary. In operation the solution technique of this problem grounded on widely known верификационном approach – check on models (model checking) and on the theory of partial specifications (partial specifications) is offered.

INTEGRATION TESTING. PARTIAL SPECIFICATIONS. VERIFICATIONS. SOFTWARE.

Mezentseva O. S., Alekseev A. I. IMPROVING DISTRIBUTED COMPUTING SYSTEMS PERFORMANCE USING THRESHOLD SCHEMES AND MODULAR ARITHMETICS ALGORITHMS.

The paper deals with the aspects of distributed data representation model, based on (N, k)-threshold schemes and modular arithmetic, namely, the system of residual classes, what leads to a high degree of parallelism and, as follows, high execution speed of the proposed model. Also specified are some possible hardware solutions.

RESIDUAL CLASSES SYSTEM. THRESHOLD SCHEMES. DISTRIBUTED COMPUTING. PARALLEL ALGORITHMS.

Dron V. O., Markov A. M., Naumov A. C. REDUCTION OF CALCULATIONS VOLUME BY RANDOM SEARCH IN PROBLEMS OF DIRECTION FINDING.

The question of reduction of time of definition of a direction on a source of a radio emission by optimization of fazovo-correlation function by a method of casual search is considered.

RADIO DIRECTION FINDING. OPTIMIZATION THEORY. METHOD OF CASUAL SEARCH. METHOD OF PHASE CORRELATION.

Beryozkin A. A. CONSTRUCTION OF OPTIMAL NEURONAL DECODERS OF BLOCK CODES.

In clause basic method of construction neuronal decoders of block codes. Structures of neuronal decoders some BCH codes are suggested.

DECODING. NEURAL NETWORKS. DATA COMMUNICATION. TELECOMMUNICATION SYSTEMS.

Kudryashov D.A., Tsikin I.A. PROBABILITY-TIME CHARACTERISTICS OF DUPLEX ALARM RADIO SYSTEMS.

The analytic forms of probability-time characteristics are considered. The imitation models are described.

RADIO SYSTEM. TWO-FORKED SYSTEMS. REPETITION of PACKAGES. MONITORING. DISTURBING NOTICES. PROBABILITY-TIME CHARACTERISTICS.

Sidorov Yu. E., Nasper M. L. APPRAISING OF EFFICIENCY RADIO SIGNALS DETECTOR IN SEPARATED RECEPTION ON THE CHANNELS WITH FADING.

The optimum rule of detector radio signals with unknown character of fading received. We have appraised efficiency of detector by statistical simulation method.

DETECTOR. FADING. THE EFFICIENCY ESTIMATION. THE CARRIED RECEPTION. STATISTICAL SIMULATION MODELING. THE SIGNAL-NOISE.

Bylin A.R., Bylina M.S., Glagolev S.F. WAY OF INCREASE OF RESOLUTION OF PULSE MEASUREMENTS.

In clause the analysis of an opportunity of use instead of a traditional unipolar rectangular probing signal, so-called «bipolar pulse», representing set of two rectangular impulses of different polarity is lead. It is shown, that use of bipolar pulse allows to receive special «bipulsed» TDR trace. In clause the algorithm of processing registered traditional TDR trace also is offered/

TIME DOMAIN REFLECTOMETER. TDR TRACE. PROBING SIGNAL. RESOLUTION.

Sidorov Yu. E., Lavrentiev N. V. THE DETECTOR RADIO SIGNALS IN THE CHANNELS OF COMMUNICATION WITH UNKNOWN CHARACTER OF FADING.

The optimum rule of detector radio signals with unknown character of fading received. We have appraised efficiency of detector by statistical simulation method.

DETECTOR SIGNALS. CHANNEL OF COMMUNICATION. FADING. THE EFFICIENCY APPRAISING. STATISTICAL SIMULATION.

Rizhov V. A., Kipot V. L., Birialtsev E. V. THE ESTIMATION OF PARAMETERS OF SPECTRUM FOR LOW-FREQUENCY SEISMIC SENSING TECHNOLOGY ON BASIS OF WAVELET ANALYSIS.

The estimation of significant parameters of microseism specters for oil field geophysical prospecting in low-frequency seismic sensing technology is suggested. The algorithm estimation of parameters of spectrum maximums is described on basis of local optimization methods to applying to wavelet image.

WAVELET. PARAMETER. SPECTRUM. CONVOLUTION. MICROSEISM. LSS. OIL.

Danilovitch D.A. THE OPTIMIZATION OF ANTENNA CHARACTERISTICS AND HEIGHTS ON MULTITOP DIGITAL MICROWAVE LINKS.

Problem of associated optimization of antenna types and heights on multitop digital microwave links in case of dual-frequency plan of channel arrangement under additional informal constraints is considered in the stated material. Mathematical model, algorithm and a method of finding a predetermined number of ranged best decisions of optimization problem stated above are introduced.

DUAL-FREQUENCY PLAN. DISTURBING BACKWARD SIGNAL. ASSOCIATED OPTIMIZATION. TOTAL COSTS MINIMIZATION. CALCULATION VOLUME.

Buzyukova I.L., Gaidamaka Y.V. ANALYSIS OF SIGNALLING LOAD IN NATIONAL INTELLIGENT NETWORK WITH DIFFERENT TIME ZONES.

This paper examines various configurations of the large scale Russian National Intelligent Network in order to analyze the required volumes of signaling equipment. Network configuration is modified by adding network nodes and by sharing signaling load.

INTELLIGENT NETWORK. INAP SIGNALLING LOAD. NUMBER OF SIGNALLING LINKS.

Maximov M.I., Sidorova N.A. PRINCIPLES OF OPERATION AND PRACTICAL IMPLEMENTATION OF CODE EXCITED LINEAR PREDICTIVE SPEECH CODEC.

The practical realization of speech signal coding compatible with the IS-96-A and IS-733 standards on the digital signal processor TMS320VC5410A has been considered. The principles of speech codec construction by given standards have been described. The estimation of computing expenses and required memory for realization of each codec functional block has been determined.

VOCODER. LINEAR PREDICTION. LINEAR SPECTRAL STEAMS. CODE BOOK. EXCITATION FROM THE CODE.

Kukunin D.S. MODEL OF SYSTEM WITH AN ADAPTIVE CHOICE OF CODE.

In article work of system with adaptive coding which is capable to choose an optimum cyclic code depending on a communication channel condition is described. Examples of reaction of system on increase and reduction of probability of an error in the channel are resulted.

DUAL BASIS. GALOIS FIELD. PROBABILITY OF THE ERROR. ADAPTIVE SYSTEM. RELIABILITY. QUALITY OF THE CHANNEL.

Kukunin D.S., Kognovitskiy O.S. TECHNIQUE OF THE ESTIMATION OF QUALITY OF THE CHANNEL IN THE COURSE OF DATA TRANSMISSION.

In article prospects of use of dual basis of Galois Field with reference to problems of an estimation of quality of the channel are shown. The technique of definition of probability of an error in the channel in the course of data transmission is offered.

DUAL BASIS. GALOIS FIELD. QUALITIES OF THE CHANNEL. PROBABILITY OF THE ERROR. THE CONFIDENTIAL INTERVAL.

Novikov F.A., Tikhonova U.N. DEFINITION OF DOMAIN-SPECIFIC LANGUAGES BY INTERPRETED AUTOMATA.

The method of domain-specific languages definition by interpreted automata is described. This method enables one to define domain-specific language in the form of three parts: abstract syntax, concrete syntax and semantics. The program implementation of this method, virtual machine of automata-based programming, interprets automata of language specification and realizes interpretation and parsing of programs in the language thereby.

DOMAIN-SPECIFIC LANGUAGES. SEMANTICS. METAMODEL. AUTOMATA. VIRTUAL MACHINE OF AUTOMATA-BASED PROGRAMMING.

Dunaev A.V., Larchenko A.V, Boukhanovsky A.V. INTEGRATED DEVELOPMENT ENVIRONMENT OF DECISION SUPPORT FOR HIGH-PERFORMANCE APPLICATIONS DEVELOPER IN GRID.

The article examines a new approach to automation of high-performance applications development for Grid. Basic principles of intellectual decision support system functioning are described, issues to practical implementation of the system in PEG software family are discussed. Also assessments of composite applications effectiveness are presented.

GRID. HIGH-PERFORMANCE COMPUTING. INTELLECTUAL SYSTEM. DESIGNING. WORKFLOW.

Rachinskiy E.S. SOFTWARE FRAMEWORK FOR PERFORMANCE ANALYSIS OF CLIENT-SERVER SYSTEMS UNDER LOAD.

Main issues of client-server systems performance analysis under load are considered. Main methods and existing tools for load generation and performance characteristics measurement are presented. Need of a new approach is demonstrated. General principals and ideas of the new proposed approach, its implementation and application examples are described.

PERFORMANCE. SOFTWARE. TESTING. INFORMATION SYSTEMS. ANALYSIS/

Starostenko A., Samochadin A. SOWTWARE TOOLS FOR REMOTE QUEUING.

The problems of automation queuing systems in end-users services are considered. The approach to queue control via Internet and SMS services is given.

QUEUE CONTROL. AUTOMATION SYSTEMS. INTERNET-SYSTEMS. MOBILE DEVICES. MASS SERVICE SYSTEMS.

Kurakin A. S. USAGE MULTIAGENTAL THE APPROACH BY DEVELOPMENT OF COMPLEXES OF PROGRAMS OF SYSTEMS OF CORPORATE REMOTE DERIVATION.

In clause the analysis of development of modern technologies of program complexes of corporate remote training is lead, features and specificity of development and functioning of such systems are revealed. On the basis of the lead analysis the structure of construction of program complexes of systems of remote training is offered multiagenty.

PROGRAM COMPLEXES. SYSTEM. REMOTE TRAINING. MULTIADGEY STRUCTURE.

Itsykson V. M., Moiseev M. J., Tsesko V. A., Karpenko A. V. RESEARCH ON TOOLS FOR AUTOMATION OF DEFECTS DETECTION IN PROGRAM SOURCE CODE.

Taxonomy of defects in source code is proposed in this article. Comparative analysis of existing automated defects detection tools, based on static analysis of source code, and is carried out. Trends in the field of automated defects detection are presented.

SAFETY. DEFECTS DETECTION. STATIC ANALYSIS. AUTOMATION. DEFECTS TAXONOMY. PROGRAM.

Itsykson V. M., Zakharov A. V., Akhin M. K., Myasnov A. V. AUTOMATIC MODEL CHECKING BASED DEFECT DETECTION.

Complex technique for software defect detection, based on model checking enhanced static analysis, is introduced in this article. Its usability is proven through function call protocol error detection.

MODEL CHECKING. STATIC ANALYSIS. DEFECT DETECTION. FUNCTION CALL PROTOCOL ERRORS.

Saveljev M.V., Alikov A.J., Fedoseev S.V. SYNTHESIS OF SIMPLE STRUCTURES AND SETS OF PULSE MANAGING INFLUENCES.

In clause perfection of methods of synthesis of control systems with pulse influence is considered at formalization of complexity and exception of redundancy of structures. As a result of the executed formalization the analyzed problem is reduced to a problem of search of simple structures with in part linear restrictions.

SYNTHESIS. FORMALIZATION. STRUCTURE. THE CONTROL SYSTEMS. MANAGING INFLUENCES.

Chebykin S. V. TECHNIQUE OF PRODUCTION PLANNING OF FLOAT GLASS MANUFATURING.

This article describes the production planning optimization of float glass manufacturing. Principles providing the efficient production planning, based on Advanced Planning and Scheduling concepts, are also being considered in this article.

OPTIMIZATION. BACKLOG OF BUSINESS ORDERS. PLANNING. MANAGEMENT.

Lavrov A.P., Molodyakov S.A. POSSIBILITIES OF CONSTRUCTION OF SIGNAL PROCESSORS IN THE FORM HYBRID OPTOELECTRONIC MICROCIRCUITS.

The optoelectronic processors in the form hybrid microcircuits is a new trend of researches. Are submitted scheme of hybrid microcircuits, and scheme of signal processing systems. Experimental results are discussed.

OPTOELECTRONIC PROCESSOR. THE HYBRID MICROCIRCUIT. MATRIX OPERATIONS. THE CONVEYOR. PULSE-WIDTH MODULATION.

Zaitsev E.F., Guzenko K.V., Moldavanova E.N. ANALYSIS OF WAVES IN A FERRITE-DIELECTRIC WAVEGUIDE BY VARIATION METHOD.

The electrodynamics problem of waves along the electrically controlled ferrite-dielectric waveguide was solved. Ritz' algorithm has been realized in MATLAB. The calculations were made for different waveguide designs. The results can be applied to improve electrically scanning millimeter-wave antennas.

MILLIMETER WAVES. SCANNING ANTENNA. FERRITE-DIELECTRIC WAVEGUIDE. VARIATION METHOD.

Makatov M. R. Verdiev M. M. INFORMATION MODEL OF THE DESCRIPTION OF OBJECTS AND SITUATIONS OF THE PROBLEM ENVIRONMENT.

The information model of the description of objects and situations of the problem environment is considered at the decision of problems of planning of behavior of intellectual systems in not predetermined environments.

INFORMATIONAL MODEL. THE ENVIRONMENT. INTELLECTUAL SYSTEMS. BEHAVIOUR PLANNING.

Ismailov SH. A., Magomedov SH. G. ALGORITHMS AND STRUCTURES OF THE TRANSFORMATION NUMERIC GIVEN FROM POSITIONAL SYSTEM IN RESIDUE NUMBER SYSTEM.

In article are considered problems of the development of algorithm discharged – a parallel the transformation binary-decimal numbers in system of the remaining classes. For transformation of numbers are used tabular – an algorithmic principles data processing.

Novikov K.V. FAST IMAGE SEGMENTATION ALGORITHM BASED ON SPECTRAL CHARACTERISTICS OF ANISOTROPIC AREAS.

Noise proof algorithm for anisotropic images segmentation is considered. This algorithm used power spectrum properties of anisotropic images for image segmentation. Fast versions of this algorithm are suggested.

IMAGE SEGMENTATION. ANISOTROPY CHARACTERISTICS. LOCALIZED RADON TRANSFORM.

Surov A.N., Igizjanova N.A., Potapov V.I. ABOUT CONVERGENCE of DECISION DIFFERENCE SCHEME IN MATHEMATICAL MODEL of GENERATION THERMAL ENERGY.

The mathematical model is considered in clause of generation thermal energy at passage of an electric current to a liquid conductor – a slag bath with greater resistance. The differential equations are considered in mathemati-

cal model which are solved by a method of final differences. Numerical results are received of temperature fields and density of a current.

MATHEMATICAL MODEL. THE METHOD OF FINAL DIFFERENCES. THE TEMPERATURE FIELD.

Pavlov D. A. ALGORITHM FOR BUILDING THE EXTENDED UNIVERSAL GRЦBNER BASE.

The paper is devoted to the description of an algorithm for building the extended universal Grцbner base (EUGB) of a polynomial ideal. This algoritm is based on the relationship between multi-dimensional Young diagrams and Grцbner bases of polynomial ideals. The algorithm starts with construction of the maximal linearly-independent monomial sets (MLIMS).

GRЦBNER BASES. MONOMIAL ORDERINGS. YOUNG DIAGRAMS. POLYNOMIAL IDEALS.

Mantserov D. I. SYSTEM OF VERIFICATION FOR PARAMETRIC CLASSES OF TASKS IN MATHEMATICS.

In clause system for creation tasks in mathematics and verification of knowledge by using parametric classes of tasks. Also in clause is expounded the base of parametric classes of tasks and structure of verification system

PARAMETRIC CLASS OF TASKS. SPECIMEN OF A PARAMETRIC CLASS. CLASS-PATTERN. PREDICATES OF A CLASS-PATTERN. GENERATION OF SPECIMEN.

Kanunnikova E. A., Koltunov L. I., Potapenko A. N. NUMERICAL MODELING of DISTRIBUTED ELECTROTECHNICAL SYSTEMS IN SEMI-INFINITE REGIONS BASED ON THE INVERSION METHOD.

The article presents the possibilities of the modified inversion method for semi-infinite mediums for solving outer boundary problems using the example of the distributed electrotechnical systems. The results of numerical modeling of the systems researched and the comparative analysis with known data are presented.

OUTER BOUNDARY PROBLEM. NUMERICAL MODELING. MODIFIED INVERSION METHOD FOR SEMI-INFINITE MEDIUMS. ELECTROTECHNICAL SYSTEMS.

Ushmotkin A. S., Korzhik V. I. THE ESTIMATION OF AN EFFICIENCY OF DIGITAL WATERMARKING SYSTEM FOR IMAGES UNDER CONDITIONS OF A COALITION ATTACK.

Digital watermarking (DWM) system for images using spread spectrum signals is examined. The estimation of an efficiency of DWM system is produced in the article in terms of error probability of missing and false detecting of coalition participants.

DIGITAL WATERMARK. COPYRIGHT. SPREAD SPECTRUM SIGNAL. COALITION ATTACK. ERROR PROBABILITY. DETECTOR. MODELLING.

Suslov N.V. PRINCIPLES OF CONSTRUCTION OF VIRTUAL TRAINING SPACE.

The article describes the work which is done with the support of the Russian Foundation for Fundamental Research, which aim is to study the constructing principles of a virtual learning environment. Several examples of existing environments made up with virtual learning environment «Krestyanstvo» are also presented.

INFORMATION TECHNOLOGY. MULTIMEDIA. VIRTUAL LEARNING ENVIRONMENT. DISTRIBUTED COMPUTING. OPEN SYSTEMS.