

ТЕХНОЛОГИИ АВТОМАТНОГО ПРОГРАММИРОВАНИЯ И ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Генерация конечных автоматов для управления моделью беспилотного самолета	<i>Александров А.В., Казаков С.В., Сергушичев А.А., Царев Ф.Н., Шалыто А.А.</i>	3
Разработка системы управления малоразмерным вертолетом	<i>Клебан В.О., Шалыто А.А.</i>	12
Применение двухэтапного генетического программирования для построения модели танка в игре «ROBocode»	<i>Соколов Д.О.</i>	16
Применение машинного обучения для создания управляющих автоматов на примере игры «ROBocode»	<i>Чернявский И.И.</i>	22
Автоматический синтез системы управления мобильным роботом для решения задачи «КЕГЕЛЬРИНГ»	<i>Алексеев С.А., Калининченко А.И., Клебан В.О., Шалыто А.А.</i>	26
Анализ эффективности использования GPU для автоматического синтеза системы управления мобильным роботом	<i>Сергеев А.А., Клебан В.О., Шалыто А.А.</i>	32
Программно-аппаратный комплекс для исследования автоматного управления мобильными роботами	<i>Алексеев С.А., Клебан В.О., Шалыто А.А.</i>	37
Детекторы особенностей в методе Виолы-Джонса, построенные на основе конечных автоматов	<i>Скорынин П.А.</i>	40
Методы оптимизации стратегий в играх для двух участников с использованием генетических алгоритмов	<i>Трофимов Д.А., Шалыто А.А.</i>	45
Разработка методов построения конечных автоматов с использованием алгоритма имитации отжига на примере игры «Война за ресурсы»	<i>Заикин А.К.</i>	49
Метод представления автоматов линейными бинарными графами для использования в генетическом программировании	<i>Данилов В.Р., Шалыто А.А.</i>	54
Автоматический подбор параметров внешней среды при генерации автоматных программ с помощью генетических алгоритмов	<i>Кулев В.А.</i>	57
Применение генетического подхода для генерации клеточных автоматов	<i>Тихомиров А.В., Шалыто А.А.</i>	62
Применение генетических алгоритмов к генерации тестов для автоматных программ	<i>Законов А.Ю., Шалыто А.А.</i>	66
Генерация тестов для олимпиадных задач по программированию с использованием генетических алгоритмов	<i>Буздалов М.В.</i>	72
Совместное применение контрактов и верификации для повышения качества автоматных программ	<i>Борисенко А.А., Парфенов В.Г.</i>	77
Виртуальная лаборатория обучения методам искусственного интеллекта для генерации управляющих конечных автоматов	<i>Тяhti А.С.</i>	81

ОПТИЧЕСКИЕ И ОПТИКО-ЭЛЕКТРОННЫЕ СИСТЕМЫ. ОПТИЧЕСКИЕ ТЕХНОЛОГИИ

Обесцвечивание природных сапфиров	<i>Асеев В.А., Некрасова Я.А., Хомченко К.В.</i>	86
Представление экспериментальных данных об оптических параметрах атмосферы аналитическими зависимостями	<i>Демин А.В., Моисеева М.И.</i>	90

ФОТОНИКА И ОПТОИНФОРМАТИКА

Согласованная система квантовой рассылки криптографического ключа на поднесущей частоте модулированного света	<i>Рупасов А.В., Глейм А.В., Егоров В.И., Мазуренко Ю.Т.</i>	95
Фазовая самомодуляция однопериодных оптических волн	<i>Дроздов А.А., Козлов С.А.</i>	99
Диффузия света в среде с наноразмерными неоднородностями	<i>Альфимов А.В., Арысланова Е.М. и др.</i>	105

АНАЛИЗ И СИНТЕЗ СЛОЖНЫХ СИСТЕМ

Разработка модели решения обратной задачи вертикального зондирования ионосферы	<i>Гришеницев А.Ю., Коробейников А.Г.</i>	109
Программируемый формирователь траектории движения следящего электропривода	<i>Ловлин С.Ю., Цветкова М.Х., Жданов И.Н.</i>	113
Контроль затрат на управление при воспроизведении гармонических экзогенных воздействий: грамианный подход	<i>Бирюков Д.С., Ушаков А.В.</i>	117

КОМПЬЮТЕРНЫЕ СИСТЕМЫ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Основные принципы решения задачи преобразования объектно-ориентированного кода в формат RDF средствами семантического анализа	<i>Зараковский А.В., Клименков С.В., Ткаченко Н.И., Харитонов А.Е.</i>	123
Анализ методов построения траектории движущихся объектов на основе сегментации видеоданных	<i>Рубина И.С.</i>	127
Проблемы пополнения семантического словаря	<i>Боярский К.К., Каневский Е.А.</i>	132
Организация сетевого взаимодействия узлов распределенной системы хранения данных	<i>Лукьянов Н.М., Дергачев А.М.</i>	137
Метод сжатия динамического диапазона цифровых видеоплаграмм Фурье в задаче встраивания скрытых водяных знаков	<i>Старченко А.П., Гатчин Ю.А.</i>	141
Оценка количества информации в базах знаний	<i>Бессмертный И.А.</i>	146
Уязвимости мониторов виртуальных машин	<i>Евелев Ю.Е., Чернокнижный Г.М.</i>	149

БИМЕДИЦИНСКИЕ ТЕХНОЛОГИИ

Применение метода ВЧ-ближнепольного зондирования в диагностике биообъектов	<i>Арсеньев А.В., Волченко А.Н., Лихачева Л.В., Печерский В.И.</i>	154
Неинвазивные методы измерения билирубина, гемоглобина и глюкозы. Прибор гемобилиглюкометр	<i>Кожохина Е.В.</i>	157

ЭКОНОМИКА И ФИНАНСЫ. МЕНЕДЖМЕНТ

Применение имитационного моделирования для оценки качества бизнес-планов инновационных проектов	<i>Кравец А.Г., Дроботов А.С.</i>	163
Роль технопарков в развитии России как инновационного общества	<i>Абрамова Е.Ю.</i>	166

КРАТКИЕ СООБЩЕНИЯ



ГЛАВНЫЙ РЕДАКТОР.....

д.т.н., профессор В.О. Никифоров

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

д.т.н., профессор А.А. Бобцов, д.т.н. А.В. Бухановский,
д.т.н., профессор В.А. Валетов, д.ф.-м.н., ст.н.с. Т.А. Вартамян,
д.т.н. М.А. Ган, д.т.н., профессор Ю.А. Гатчин,
к.т.н., ст.н.с. Н.Ф. Гусарова, д.т.н., профессор А.В. Демин,
к.т.н., доцент Н.С. Кармановский (заместитель главного редактора),
д.ф.-м.н., профессор Ю.Л. Колесников, д.ф.-м.н., профессор С.А. Козлов,
д.т.н., профессор А.Г. Коробейников, д.т.н., профессор В.В. Курейчик,
д.т.н., доцент Л.С. Лисицына, к.т.н., доцент В.Г. Мельников,
д.т.н., профессор Ю.И. Нечаев, д.т.н., профессор Н.В. Никоноров,
д.т.н., профессор А.А. Ожиганов, д.т.н., профессор П.П. Парамонов,
д.ф.-м.н., ст.н.с. Е.Ю. Перлин, д.т.н., профессор И.Г. Сидоркина,
д.т.н. О.А. Степанов, д.т.н., профессор В.Л. Ткалич,
д.т.н., профессор А.А. Шалыто, д.т.н., профессор Ю.Г. Якушенков

Секретарь Г.О. Артемова

Редактор Н.Ф. Гусарова

Перевод Н.Г. Лещикова

Адрес: 197101, Санкт-Петербург, Кронверкский пр., 49, СПбГУ ИТМО, комн. 330

Телефон / факс (812) 233 45 51

<http://books.ifmo.ru/ntv/>

E-mail: karmanov@mail.ifmo.ru

ПРАВИЛА ПРЕДСТАВЛЕНИЯ РУКОПИСЕЙ

ОБЩИЕ ПОЛОЖЕНИЯ

Редакция принимает к рассмотрению оригинальные (нигде ранее не опубликованные, за исключением материалов научных конференций) работы по тематике журнала, представляющие новые научные результаты, полученные лично авторами статьи. Публикуются также обзоры, характеризующие современное состояние актуальных направлений развития науки и техники, сообщения о научных конференциях, материалы научных дискуссий и рецензии на новые книги.

Тематика журнала включает разделы, отражающие достижения науки, техники и технологии в области современного приборостроения по направлениям: оптика, оптические системы и технологии, оптическое материаловедение и нанотехнологии, информационные и компьютерные системы и технологии, методы и системы защиты информации, системный анализ, системы автоматического управления, мехатроника, системы автоматизированного проектирования, микроэлектроника, биомедицинские технологии. В качестве дополнительных разделов журнал также публикует работы по экономике и финансам.

Материал статьи должен быть изложен в соответствии со следующей структурой.

В аннотации, рассчитанной на самый широкий круг читателей, необходимо кратко (объемом до 100 слов) изложить научное содержание статьи.

Во введении необходимо представить содержательную постановку рассматриваемого вопроса, краткий анализ известных из научной литературы решений, критику их недостатков и преимущества (особенности) предлагаемого подхода.

В основном тексте статьи должна быть представлена строгая постановка решаемой задачи, изложены и обстоятельно разъяснены (доказаны) полученные утверждения и выводы, приведены результаты экспериментальных исследований или математического моделирования, иллюстрирующие сделанные утверждения. Основной текст статьи может быть разбит на отдельные разделы: «Постановка задачи», «Основной результат», «Результаты моделирования» и т.п.

В заключении необходимо кратко сформулировать основные результаты, прокомментировать их и, если возможно, указать направления дальнейших исследований.

Все статьи проходят процедуру рецензирования. После ее завершения автору направляется экземпляр рукописи с замечаниями рецензента и научного редактора для доработки. При опубликовании статьи указываются даты ее поступления и последующих переработок.

ОСНОВНЫЕ ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ СТАТЕЙ

Объем статьи, включая иллюстрации, таблицы и список литературы, не должен превышать 6 страниц.

Оригинал-макет статьи выполняется в редакторе Word for Windows. Шрифт Times New Roman, размер шрифта 12 pt, межстрочный интервал одинарный, поля сверху 20 мм, остальные поля 25 мм. Абзацный отступ 10 мм.

Аннотация печатается шрифтом Times New Roman 10 обычный, межстрочный интервал одинарный, выравнивается по ширине страницы. В аннотации не допускаются ссылки на цитированную литературу и громоздкие формулы.

Название раздела печатается шрифтом Times New Roman 12 полужирный, межстрочный интервал одинарный, отделяется от предыдущего раздела полужирной пустой строкой, а от последующего текста одной пустой строкой.

Название подраздела печатается шрифтом Times New Roman 12 полужирный, межстрочный интервал одинарный, пустыми строками не отделяется.

Формулы набираются в редакторе Microsoft Equation Ed. 3.0. Параметры стиля: текста — Times New Roman, прямой; функции — Times New Roman, прямой; переменной — Times New Roman, наклонный; греческих букв — Symbol, прямой; символов — Symbol, прямой; матрицы-вектора — Times New Roman, прямой, полужирный; чисел — Times New Roman, прямой. Размеры: обычный — 12 pt, крупный индекс — 10 pt, мелкий индекс — 8 pt, крупный символ — 16 pt, мелкий символ — 12 pt. Отдельные переменные в тексте допускается набирать в текстовом режиме с соблюдением требований, указанных для формул. Разделительный знак десятичных дробей — запятая.

При оформлении рисунков и графиков используется шрифт Times New Roman. Стиль написания переменных должен соответствовать требованиям к основному тексту.

Используются только единицы измерения, соответствующие системе СИ.

Список использованной литературы оформляется в соответствии с ГОСТ 7.0.5-2008. Нумерация позиций — согласно очередности ссылок в тексте.

Плата с аспирантов за публикацию рукописей не взимается.

Более подробную информацию смотрите на сайте <http://books.ifmo.ru/ntv/>

Подписано к печати 15.02.2011 Тираж 350 экз. Заказ № 20(72)

Отпечатано в учреждении «Университетские телекоммуникации»
Адрес: 197101, Санкт-Петербург, Кронверкский пр., 49

Журнал зарегистрирован Федеральной службой по надзору в сфере связи
и массовых коммуникаций, свидетельство ПИ № ФС77-33466 от 10.10.2008 г.
Подписной индекс 47 197 по каталогу агентства Роспечать
(Газеты. Журналы)

AUTOMATA-BASED PROGRAMMING AND ARTIFICIAL INTELLIGENCE TECHNOLOGIES		
Generation of finite state machines for unmanned airplane controlling	<i>A. Alexandrov, S. Kazakov, A. Sergushichev, F. Tsarev, A. Shalyto</i>	3
Small-sized helicopter control system development	<i>V. Kleban, A. Shalyto</i>	12
Double-step genetic programming application for tank model creation in "ROBOCODE" game	<i>D. Sokolov</i>	16
Machine learning application for control automata creation by "ROBOCODE" game example	<i>I. Chernyavskiy</i>	22
Automatic synthesis of mobile robot control system for solving the "KEGELRING" problem	<i>S. Alexeev, A. Kalinichenko, V. Kleban, A. Shalyto</i>	26
GPU effectiveness analysis for automatic synthesis of robot control system	<i>A. Sergeev, V. Kleban, A. Shalyto</i>	32
Hardware and software complex for finite state mobile robot control study	<i>S. Alexeev, V. Kleban, A. Shalyto</i>	37
Feature detectors in the viola-jones method on the basis of finite automata	<i>P. Skorynin</i>	40
Strategy optimization methods for two-player games with genetic algorithms	<i>D. Trofimov, A. Shalyto</i>	45
Development of finite automata creation methods with annealing simulation algorithm by the «WAR FOR RESOURCES» example	<i>A. Zaikin</i>	49
Representation of finite state automata by linear binary graphs in genetic programming	<i>V. Danilov, A. Shalyto</i>	54
Automatic selection of environment parameters during automata programs generation by genetic algorithms	<i>V. Kulev</i>	57
Genetic approach for cellular automata generation	<i>A. Tikhomirov, A. Shalyto</i>	62
Genetic algorithms application to test generation for automata programs	<i>A. Zakonov, A. Shalyto</i>	66
Tests generation for olympiad programming tasks using genetic algorithms	<i>M. Buzdalov</i>	72
Joint application of contracts and verification for automata-based programs quality enhancement	<i>A. Borisenko, V. Parfenov</i>	77
Virtual laboratory for artificial intelligence training methods for controlling finite state automata generation	<i>A. Tyakhti</i>	81
OPTICAL AND OPTICAL ELECTRONIC SYSTEMS. OPTICAL TECHNOLOGIES		
Discoloration of natural sapphires	<i>V. Aseev, Y. Nekrasova, K. Khomchenko</i>	86
Representation of experimental data on atmosphere optical parameters by analytic relations	<i>A. Demin, M. Moiseeva</i>	90
PHOTONICS AND OPTICAL INFORMATICS		
Consistent system of unique key quantum distribution on chopped light sideband	<i>A. Rupasov, A. Gleim, V. Egorov, Y. Mazurenko</i>	95
Phase self-modulation of single-cycle optical waves	<i>A. Drozdov, S. Kozlov</i>	99
Light diffusion in a medium with nanoscale heterogeneities	<i>A. Alfimov, E. Arysylanova et al.</i>	105
ANALYSIS AND SYNTHESIS OF COMPLEX SYSTEMS		
Solution model of inverse problem of ionosphere vertical sounding	<i>A. Grishentsev, A. Korobeynikov</i>	109
Programmed servo drive trajectory shaper	<i>S. Lovlin, M. Tsvetkova, I. Zhdanov</i>	113
Cost control estimation for harmonic exogenous actions: gramian approach	<i>D. Biryukov, A. Ushakov</i>	117
COMPUTER SYSTEMS AND INFORMATION TECHNOLOGIES		
Basic principles for solving the problem of transforming an object-oriented source code to RDF format using semantic analysis	<i>A. Zarakovsky, S. Klimenkov, N. Tkachenko, A. Kharitonova</i>	123
Creation methods analysis of moving objects trajectory on the basis of video data segmentation	<i>I. Rubina</i>	127
Problems of semantic dictionary replenishment	<i>K. Boyarsky, E. Kanevsky</i>	132
Organization of nodes network interaction in distributed repositories	<i>N. Luk'yanov, A. Dergachev</i>	137
Compression method for dynamic range of digital fourier hologram while embedding hidden watermarks	<i>A. Starchenko, Yu. Gatchin</i>	141
Quantitative assessment of information in knowledge bases	<i>I. Bessmertny</i>	146
On the vulnerability of virtual machine monitors	<i>Yu. Evelev, G. Chernoknizhny</i>	149
BIOMEDICAL TECHNOLOGIES		
Close-field high frequency probing method in biological diagnostics	<i>A. Arsen'yev, A. Volchenko, L. Likhacheva, V. Pecherskiy</i>	154
Non-invasive measuring methods of bilirubin, haemoglobin and glucose. Haemobilglucometer device	<i>E. Kozhokhina</i>	157
ECONOMICS AND FINANCES. MANAGEMENT		
Simulation modeling application for the quality assessment of innovative projects business plans	<i>A. Kravets, A. Drobotov</i>	163
The role of industrial parks in development of russia as innovative society	<i>E. Abramova</i>	166
BRIEF REPORTS		
		169



К ШЕСТИДЕСЯТИЛЕТИЮ ВЛАДИМИРА НИКОЛАЕВИЧА ВАСИЛЬЕВА

1 апреля 2011 года исполняется 60 лет со дня рождения ректора Государственного образовательного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет информационных технологий, механики и оптики» Васильева Владимира Николаевича.

В.Н. Васильев окончил Ленинградский ордена Ленина Политехнический институт им. М.И. Калинина (ЛПИ) по специальности «Теплофизика» (1974). До 1978 г. он обучался в аспирантуре ЛПИ, в 1978–1983 г.г. принимал участие в создании и становлении Ставропольского политехнического института, последовательно занимая должности старшего научного сотрудника, ассистента и доцента, в 1980 г. защитил кандидатскую диссертацию по специальности «Теплофизика и молекулярная физика».

С 1983 г. В.Н. Васильев работает в Ленинградском институте точной механики и оптики (ЛИТМО). В 1989 г. защитил докторскую диссертацию по специальностям «Оптические приборы» и «Теплофизика и молекулярная физика». Доктор технических наук (1990), профессор (1992).

В 1991 г. В.Н. Васильев основал и возглавил новую кафедру «Компьютерные технологии», которая на протяжении двадцати лет выпускает молодых специалистов в области прикладной математики и информатики, крайне необходимых во всех отраслях науки, техники и производства. О качестве подготовки и востребованности выпускников кафедры говорят многочисленные достижения кафедры и ее выпускников.

Студенты кафедры неоднократно были чемпионами России по программированию, победителями европейских и международных олимпиад по информатике. В 2000 г. они стали обладателями серебряных медалей, в 2001, 2002, 2003, 2005, 2007 г.г. удостоились золотых медалей, а в 2004, 2008 и 2009 г.г. стали абсолютными чемпионами мира и Европы. Выпускник кафедры «Компьютерные технологии» 2000 года Павел Белов был удостоен премии Президента Российской Федерации 2009 года в области науки и инноваций для молодых ученых за выдающийся вклад в развитие физики метаматериалов и разработку устройств передачи и обработки изображений со сверхразрешением, а в ноябре 2010 года защитил диссертацию на соискание ученой степени доктора физико-математических наук. В 2009 году Президент Российской Федерации Д.А. Медведев вручил Гран-при «Открытие» первой молодежной национальной премии «Прорыв» Евгению Южакову, Федору Цареву и Александру Штучкину, выпускникам кафедры «Компьютерные технологии», за разработку программного обеспечения первого мире коммуникатора 4G WiMax «Yota».

В 1995 г. сотрудники кафедры под руководством профессора В.Н. Васильева в содружестве с рядом российских университетов заложили основу федеральной университетской компьютерной сети RUNNet, и с этого времени профессор В.Н. Васильев является научным руководителем этой сети. В 2000 году за создание Федеральной университетской компьютерной сети RUNNet коллектив авторов под руководством профессора В.Н. Васильева был удостоен премии Правительства Российской Федерации в области образования.

В 1996 г. профессор В.Н. Васильев был избран ректором Санкт-Петербургского государственного университета информационных технологий, механики и оптики. Его избрание, подтвержденное конференцией трудового коллектива вуза в 2001 и в 2006 г.г., привнесло в развитие вуза принципиально новые

импульсы. Апробированная в рамках возглавляемой В.Н. Васильевым кафедры идея формирования «синтетического» специалиста, в котором на базе компьютерной подготовки объединяются оптика, механика и электроника, сегодня расширена практически на весь кластер современных наукоемких технологий. В настоящее время в Университете осуществляется подготовка высококвалифицированных инженерных и научных кадров в области компьютерных и информационных технологий, оптоэлектроники, лазерной техники и лазерных технологий, компьютеризированных приборных комплексов и систем управления, что является целью реализуемого в Университете специального образовательного проекта. В 2007 г. университет стал победителем конкурса инновационных образовательных программ вузов России на 2007–2008 г.г. Одним из главных итогов научно-образовательной деятельности университета стала его победа в 2009 г. в многоэтапном конкурсе, в результате которой ему была присвоена категория «Национальный исследовательский университет».

Профессор В.Н. Васильев – крупный ученый в области компьютерных телекоммуникаций и математического моделирования информационных систем и многофункциональных приборных комплексов, один из создателей информационных ресурсов высшей школы России и построения единого научно-образовательного информационного пространства России. Под его руководством были разработаны и внедрены новые системы управления и сопровождения спутниковой связи, передачи данных локальными и глобальными сетями, в том числе первая отечественная межгородская АТМ-система на линии Москва–Санкт-Петербург.

Профессор В.Н. Васильев являлся научным руководителем научно-исследовательских работ по ряду федеральных целевых программ, среди которых – «Электронная Россия», «Развитие единой образовательной информационной среды», «Интеграция науки и высшего образования России», «Федеральная программа развития образования», «Исследования и разработки по приоритетным направлениям развития науки и техники».

Под руководством В.Н. Васильева в Университете ИТМО создана и поддерживается насыщенная интеллектуальная атмосфера. На базе вуза регулярно проводятся международные и всероссийские конгрессы, конференции, семинары и симпозиумы по компьютерным технологиям и сетям, телематике, оптике и математическому моделированию. О высоком научном статусе СПбГУ ИТМО говорит, в частности, тот факт, что в вузе издаются четыре научно-технических журнала. По инициативе Владимира Николаевича создан и издается «Научно-технический вестник СПбГУ ИТМО», входящий в Перечень ведущих рецензируемых научных журналов Высшей аттестационной комиссии. Под руководством В.Н. Васильева создаются и внедряются в учебный процесс новые информационные технологии обучения, в том числе дистанционного. Все это, несомненно, способствует притоку в вуз талантливой молодежи.

Научно-исследовательскую и педагогическую работу профессор В.Н. Васильев сочетает с активной общественной деятельностью, являясь советником Губернатора Санкт-Петербурга, председателем Совета ректоров вузов Санкт-Петербурга, председателем трех специализированных диссертационных советов, членом редакционных советов журналов «Компьютерные инструменты в образовании», «Известия вузов. Приборостроение», «Оптический журнал» и ряда других, членом научно-технического совета Правительства Санкт-Петербурга, членом совета Национального фонда подготовки кадров, постоянным членом международных научных обществ и международных научно-технических рабочих групп.

Научно-педагогическая деятельности профессора В.Н. Васильева получила широкое общественное признание. Он является членом-корреспондентом Российской академии образования (2008), действительным членом Международной академии наук высшей школы (1999), действительным членом Академии инженерных наук им. А.М. Прохорова (2003), действительным членом и членом-корреспондентом ряда других международных и отраслевых академий, членом научно-технического совета Минобрнауки России, членом научного совета по проблемам информатизации образования. С 2004 г. профессор В.Н. Васильев возглавляет Совет ректоров вузов Санкт-Петербурга, а с 2006 г. является вице-президентом Российского Союза ректоров. В 2006 г. профессор В.Н. Васильев был избран президентом Оптического общества им. Д.С. Рождественского.

В 2000 г. В.Н. Васильеву присвоено звание «Заслуженный деятель науки Российской Федерации». Он дважды – в 2003 г. и в 2005 г. – становился лауреатом премии Президента Российской Федерации в области образования и дважды – в 2000 г. и в 2008 г. – лауреатом премии Правительства России в области образования.

В 2007 г. В.Н. Васильев был награжден орденом Почета.

В данном выпуске «Научно-технического вестника» в рубрике «Технологии автоматного программирования и искусственного интеллекта» помещены статьи сотрудников, студентов и аспирантов кафедры, возглавляемой Владимиром Николаевичем.

Коллектив редакции журнала «Научно-технический вестник СПбГУ ИТМО» сердечно поздравляет Владимира Николаевича Васильева с юбилеем и желает творческих успехов!

УДК 004.4'242

ГЕНЕРАЦИЯ КОНЕЧНЫХ АВТОМАТОВ ДЛЯ УПРАВЛЕНИЯ МОДЕЛЬЮ
БЕСПИЛОТНОГО САМОЛЕТА

А.В. Александров, С.В. Казаков, А.А. Сергушичев, Ф.Н. Царев, А.А. Шалыто

Для генерации автоматов управления объектами со сложным поведением предлагается применять генетическое программирование. При этом вместо известного подхода, в котором для оценки качества управляющего автомата используется моделирование, занимающее обычно большое время, применяется подход, в котором используется сравнение поведения автоматов с поведением, обеспечиваемым за счет управления человеком. Особенность рассматриваемого подхода состоит в том, что он позволяет использовать объекты управления не только с дискретными, но и с непрерывными параметрами. Применение подхода иллюстрируется на примере создания автомата, управляющего моделью самолета в режиме «мертвая петля».

Ключевые слова: конечные автоматы, генетическое программирование, беспилотный самолет.

Введение

В последнее время для программирования систем со сложным поведением все шире применяется автоматное программирование, в рамках которого поведение программ описывается с помощью конечных детерминированных автоматов (в дальнейшем – автоматов) [1].

В автоматном программировании программы предлагается строить в виде набора автоматизированных объектов управления. Каждый такой объект состоит из объекта управления и системы управления (системы управляющих автоматов). Система автоматов получает на вход события и переменные из внешней среды и от объекта управления. На основании этих данных система управления вырабатывает выходные воздействия для объекта управления.

Для многих задач управляющие автоматы удастся строить эвристически, но существуют задачи, для которых такое построение невозможно или затруднительно. К этому классу относятся, например, задача «Умный муравей» [2–4], задача «Умный муравей-3» [5] и задача об управлении моделью беспилотного летательного аппарата [6].

Существует несколько подходов к решению последней задачи. Один из них состоит в выделении «идеальной» траектории из нескольких полетов, выполненных человеком, и последующем следовании ей. Такой подход описан в работе [7].

Другой подход – использование автоматов для управления беспилотным летательным аппаратом и построение таких автоматов с помощью генетических алгоритмов, описанных в работах [8–12].

В работе [13] для генерации конечного автомата верхнего уровня, управляющего моделью беспилотного самолета, применяется алгоритм генетического программирования, основанный на использовании метода сокращенных таблиц для представления конечных автоматов. При этом вычисление функции приспособленности базируется на моделировании поведения самолета во внешней среде, которое занимает достаточно большое время. Этот алгоритм, как и описываемый в данной работе, относится к генетическому программированию, так как особь обладает сложной структурой [10].

Цель настоящей работы – разработка лишнего указанного недостатка метода, основанного на генетическом программировании, для построения автоматов, управляющих объектами со сложным поведением. Для этого предлагается строить автоматы управления таким объектами отдельно для каждого из их режимов работы с помощью генетического программирования, используя обучающие примеры, создаваемые для каждого режима. Задавая большое число обучающих примеров, можно, как и в работе [7], избавиться от неточностей, допускаемых человеком при управлении. Такой подход является развитием идей, предложенных в работе [14], в которой рассматривались только объекты, управляемые дискретными выходными воздействиями.

В настоящей работе указанный подход обобщается на объекты, которые управляются с помощью не только дискретных, но и непрерывных воздействий. При этом в качестве примера объекта со сложным поведением рассмотрена модель самолета в режиме «мертвая петля».

Для объединения автоматов, управляющих режимами, с помощью метода сокращенных таблиц [13] или эвристического метода строится головной автомат, каждое из состояний которого соответствует одному из режимов. В результате формируется иерархическая система взаимодействующих автоматов. Вопрос о построении головного автомата в данной работе не рассматривается.

Кроме более высокого быстродействия, метод обладает еще одним преимуществом по сравнению с вычислением функции приспособленности с помощью моделирования: в предложенном методе эту функцию не требуется изменять как при переходе от одного режима к другому, так и при переходе от одного объекта управления к другому.

Постановка задачи

Исходными данными для построения управляющего конечного автомата является набор обучающих примеров (тестов), структура которых подробно описана ниже. Тесты, задающие эталонное поведение, создаются человеком.

Задача алгоритма генетического программирования состоит в построении конечного автомата, который задает поведение объекта управления, наиболее близкое к эталонному.

Органы управления

Объект управления характеризуется набором органов управления, посредством воздействия на которые объектом можно управлять. Параметры, соответствующие органам управления, будем называть управляющими. Параметры некоторых органов управления могут принимать лишь конечное множество значений – такие органы называются дискретными. Параметры других органов управления характеризуются вещественными значениями – такие органы называются непрерывными. Будем также называть управляющие воздействия на непрерывные органы управления непрерывными, а управляющие воздействия на дискретные органы – дискретными. В данной работе в каждый момент времени значения управляющих параметров – это накопленные за предыдущие моменты времени значения управляющих воздействий.

Подход, предлагаемый в настоящей работе, ориентирован на объекты управления как с дискретными, так и с непрерывными органами управления. Если все органы дискретны, то следует использовать методику, изложенную в работе [14].

Непрерывное воздействие изменяет параметр органа управления на некоторую вещественную величину, а дискретное – устанавливает соответствующий орган управления в конкретное значение. Заметим, что последовательное выполнение действий с одним органом управления эквивалентно сумме воздействий на него в случае непрерывного воздействия и последнему воздействию – в случае дискретного.

Например, одним из непрерывных управляющих параметров является угол поворота руля самолета. Непрерывным воздействием на этот орган управления является изменение угла его поворота на некоторое значение. Тогда последовательность поворотов руля на x и y градусов эквивалентна повороту руля на $x + y$ градусов.

В свою очередь, примером дискретного исполнительного органа является стартер. Дискретное воздействие на него – включение или выключение. Тогда последовательность включений и выключений стартера эквивалентна последнему, совершенному над ним действию.

Структура теста

Тест T состоит из двух частей: $T.in$ и $T.ans$. Каждая из них является последовательностью длины $T.len$ – первая из них состоит из значений входных параметров, а вторая – из соответствующих эталонных значений управляющих параметров, которые записываются в ходе экспериментов, проводимых человеком.

Каждый элемент $T.in[t]$ последовательности входных параметров состоит из P чисел – значений этих параметров в момент времени t .

Элемент $T.ans[t]$ включает в себя два набора: $T.ans[t].d$ и $T.ans[t].c$. Последовательность $T.ans[t].d$ состоит из D дискретных, а $T.ans[t].c$ – из C непрерывных параметров. Таким образом, $T.ans[t]$ состоит из $D + C$ чисел.

Обозначим через $T.out$ набор управляющих параметров, выданных автоматом на тесте T . Структура $T.out$ совпадает со структурой $T.ans$.

Предлагаемый алгоритм генетического программирования

Существуют различные модели алгоритмов генетического программирования, например, классический и островной [15].

Классический алгоритм генетического программирования состоит из следующих шагов: 1. Создание начальной популяции. 2. Вычисление значений функции приспособленности. 3. Отбор особей для скрещивания. 4. Скрещивание. 5. Мутация.

Поколение, полученное после шага 5, становится текущим, и шаги 2–5 повторяются, пока не будет достигнуто условие останова.

Предлагаемый алгоритм отличается от классического тем, что перед шагом 2 для максимизации значения функции приспособленности введен дополнительный шаг – расстановка значений выходных

воздействий на переходах «скелета» автомата. Под «скелетом» понимается автомат, значения выходных воздействий которого будут определены в дальнейшем. «Скелет» задается в виде полной таблицы переходов – для каждого состояния и каждого условия истинности или ложности входной переменной задается переход. В скелете возможны два типа переходов. Для переходов первого типа символы выходных воздействий не указываются (предыдущие значения управляющих параметров не изменяются). Для переходов второго типа соответствующие символы указываются.

Ниже приведено описание этого алгоритма в порядке, удобном для понимания.

Особь в предлагаемом алгоритме генетического программирования

Конечный автомат в предлагаемом алгоритме представляется в виде объекта, который содержит описание переходов для каждого из состояний и номер начального состояния автомата, причем число состояний автомата фиксировано и является параметром алгоритма. Для каждого перехода задается условие, при котором он выполняется. Это условие имеет вид « x_i » или « $\neg x_i$ », где x_i – i -ое утверждение (предикат) о состоянии объекта управления (например, «двигатель включен» для объекта управления «самолет»). Эти условия составляются вручную до запуска генетического алгоритма и не изменяются в течение его работы. При этом на переходах в особи не задаются значения выходных воздействий z_j , где z_j – j -ый кортеж изменений управляющих параметров. Таким образом, в особи кодируется только «скелет» автомата, а конкретные выходные воздействия, вырабатываемые на переходах, определяются с помощью алгоритма расстановки действий.

Работа автомата

Будем считать, что генерируемый автомат является синхронным – все такты его работы одинаковы, а их величина определяется инерционностью объекта управления. При этом под тактом его работы будем понимать запуск автомата на одном наборе входных параметров.

На каждом такте система управления получает значения всех входных параметров. После этого вычисляются логические значения всех условий объекта управления в порядке увеличения их номеров. После вычисления соответствующего значения условия оно подается на вход автомата. Автомат в течение одного такта может совершить несколько переходов, и результирующее воздействие автомата в каждый момент времени t может быть составлено из нескольких последовательно выполненных воздействий на отдельных переходах. Напомним, что каждый параметр результирующего воздействия – сумма воздействий в случае непрерывного параметра и последнее воздействие – в случае дискретного.

На рис.1 изображен возможный «скелет» автомата.

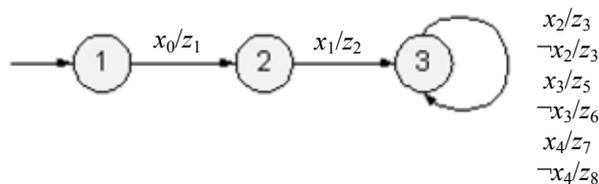


Рис. 1. «Скелет» автомата

Для данного скелета x_0 – x_4 – заданные предикаты, а значения выходных воздействий (кортежей) z_1, \dots, z_8 определяются в дальнейшем с помощью алгоритма их расстановки.

Функция приспособленности

Выбранная функция приспособленности отражает близость поведения автомата к эталонному и имеет вид

$$Fitness = 1 - \sqrt{\frac{1}{N} \sum_{i=1}^N \frac{Dist(T[i].out, T[i].ans)^2}{Dist(T[i].ans, 0)^2}} \quad (1)$$

Для вычисления ее значения на вход автомата подается каждая из N последовательностей $T[i].in$ – входной набор i -го теста и определяется последовательность значений управляющих параметров $T[i].out$, которую генерирует автомат.

Здесь под $Dist(out, ans)$ понимается «расстояние» между выходной и эталонной последовательностями управляющих параметров, которое вычисляется по формуле

$$Dist(out, ans) = \sqrt{\sum_{t=1}^{T.len} \left(\sum_{k=1}^D [out[t].d[k] \neq ans[t].d[k]] + \sum_{k=1}^C (out[t].c[k] - ans[t].c[k])^2 \right)}$$

Как предлагалось выше, перед вычислением функции приспособленности к «скелету» автомата применяется алгоритм расстановки воздействий. Этот алгоритм расставляет значения воздействий на переходах так, чтобы максимизировать значение функции приспособленности при заданном «скелете» автомата.

Алгоритм расстановки воздействий – дополнительный шаг в алгоритме генетического программирования

При использовании этого алгоритма на переходах «скелета» автомата на его вход подается набор параметров из каждого теста, который был ранее назван входным. При этом запоминаются переходы, совершаемые автоматом. После этого определяются значения выходных воздействий, при которых функция приспособленности достигает максимума – это происходит, когда сумма

$$\sum_{i=1}^N \frac{Dist(T[i].out, T[i].ans)^2}{Dist(T[i].ans, 0)^2}$$

минимальна.

Суммы по каждому параметру можно минимизировать независимо друг от друга. Значит, необходимо минимизировать сумму

$$\sum_{i=1}^N \frac{\sum_{t=1}^{T[i].len} [T[i].out[t].d[k] \neq T[i].ans[t].d[k]]}{Dist(T[i].ans, 0)^2} \tag{2}$$

для каждого k от 1 до D и сумму

$$\sum_{i=1}^N \frac{\sum_{t=1}^{T[i].len} (T[i].out[t].c[m] - T[i].ans[t].c[m])^2}{Dist(T[i].ans, 0)^2} \tag{3}$$

для m от 1 до C . Далее считаем индексы k и m зафиксированными.

Расстановка дискретных воздействий. Выделим из суммы (2) слагаемые, соответствующие каждому переходу:

$$\sum_{i=1}^N \sum_{j=1}^n \sum_{t \in Time_{i,j}} \frac{[T[i].out[t].d[k] \neq T[i].ans[t].d[k]]}{Dist(T[i].ans, 0)^2} = \sum_{j=1}^n \sum_{i=1}^N \sum_{t \in Time_{i,j}} \frac{[T[i].out[t].d[k] \neq T[i].ans[t].d[k]]}{Dist(T[i].ans, 0)^2}$$

Здесь $Time_{i,j}$ – множество таких моментов времени t , при которых номер последнего выполненного перехода автомата, запущенного на тесте i , равен j , а n – число переходов в автомате.

Таким образом, для каждого j от 1 до n необходимо минимизировать выражение

$$\sum_{i=1}^N \sum_{t \in Time_{i,j}} \frac{[T[i].out[t].d[k] \neq T[i].ans[t].d[k]]}{Dist(T[i].ans, 0)^2}$$

Зафиксируем j . Как и в случае, когда обучающий пример состоит из одного теста, $T[i].out[t].d[k]$ при $t \in Time_{i,j}$ равно значению k -го дискретного параметра на переходе j . Обозначим его через u .

Сгруппируем слагаемые с одинаковым значением $T[i].ans[t].d[k]$. При этом получим:

$$\begin{aligned} \sum_{h=1}^G \sum_{i=1}^N \sum_{t \in TimeV_{i,j,h}} \frac{[u \neq T[i].ans[t].d[k]]}{Dist(T[i].ans, 0)^2} &= \sum_{h=1}^G \sum_{i=1}^N \frac{[u \neq v_h]}{Dist(T[i].ans, 0)^2} \sum_{t \in TimeV_{i,j,1}} 1 = \\ &= \sum_{h=1}^G \sum_{i=1}^N \frac{[u \neq v_h]}{Dist(T[i].ans, 0)^2} |TimeV_{i,j,h}| = \sum_{i=1}^N \frac{\sum_{h=1}^G [u \neq v_h] \cdot |TimeV_{i,j,h}|}{Dist(T[i].ans, 0)^2} \end{aligned}$$

Здесь v_h – h -ое значение k -го дискретного параметра, G – число возможных значений этого параметра, $TimeV_{i,j,h}$ – множество тех моментов времени t из $Time_{i,j}$, при которых $T[i].ans[t].d[k]$ равно v_h .

Выбрав v_h в качестве значения дискретного параметра на i -ом переходе, получим следующее значение суммы:

$$\begin{aligned} & \sum_{i=1}^N \frac{1}{Dist(T[i].ans,0)^2} \left(|Time_{i,j}| - |TimeV_{i,j,h}| \right) = \\ & = \sum_{i=1}^N \frac{1}{Dist(T[i].ans,0)^2} |Time_{i,j}| - \sum_{i=1}^N \frac{1}{Dist(T[i].ans,0)^2} |TimeV_{i,j,h}|. \end{aligned} \quad (4)$$

Для минимизации суммы (4) необходимо выбрать то значение v_h , при котором $\sum_{i=1}^N \frac{1}{Dist(T[i].ans,0)^2} |TimeV_{i,j,h}|$ максимально. Таким образом, на j -ом переходе в качестве значения k -го дискретного параметра следует выбрать v_h , где $h = \arg \max_h \sum_{i=1}^N \frac{1}{Dist(T[i].ans,0)^2} |TimeV_{i,j,h}|$.

Расстановка непрерывных воздействий. Ввиду того, что величина изменения m -го непрерывного параметра в момент времени t ($T[i].out[t].c[m]$) равна сумме изменений этого параметра на всех выполненных переходах к этому моменту, из формулы (3) получим:

$$S = \sum_{i=1}^N \frac{1}{Dist(T[i].ans,0)^2} \sum_{t=1}^{T[i].len} \left(\sum_{j=1}^n \alpha_{i,j}[t] u_j - T[i].ans[t].c[m] \right)^2.$$

Здесь, как и раньше, u_j – неизвестная величина изменения m -го непрерывного параметра на j -ом переходе, а $\alpha_{i,j}[t]$ – число выполнений j -го перехода к моменту времени t автомата, запущенного на тесте i .

Производная S по u_h имеет вид

$$S'_{u_h} = \sum_{i=1}^N \frac{1}{Dist(T[i].ans,0)^2} \sum_{t=1}^{T[i].len} 2\alpha_{i,h}[t] \left(\sum_{j=1}^n \alpha_{i,j}[t] u_j - T[i].ans[t].c[m] \right).$$

Приравняв все S'_{u_h} к нулю, для каждого непрерывного параметра m получим систему из n уравнений:

$$\begin{cases} \sum_{j=1}^n \left(\sum_{i=1}^N \frac{1}{Dist(T[i].ans,0)^2} \sum_{t=1}^{T[i].len} \alpha_{i,1}[t] \alpha_{i,j}[t] \right) u_j = \sum_{i=1}^N \frac{1}{Dist(T[i].ans,0)^2} \sum_{t=1}^{T[i].len} T[i].ans[t].c[m] \alpha_{i,1}[t]; \\ \sum_{j=1}^n \left(\sum_{i=1}^N \frac{1}{Dist(T[i].ans,0)^2} \sum_{t=1}^{T[i].len} \alpha_{i,2}[t] \alpha_{i,j}[t] \right) u_j = \sum_{i=1}^N \frac{1}{Dist(T[i].ans,0)^2} \sum_{t=1}^{T[i].len} T[i].ans[t].c[m] \alpha_{i,2}[t]; \\ \dots \\ \sum_{j=1}^n \left(\sum_{i=1}^N \frac{1}{Dist(T[i].ans,0)^2} \sum_{t=1}^{T[i].len} \alpha_{i,n}[t] \alpha_{i,j}[t] \right) u_j = \sum_{i=1}^N \frac{1}{Dist(T[i].ans,0)^2} \sum_{t=1}^{T[i].len} T[i].ans[t].c[m] \alpha_{i,n}[t]. \end{cases}$$

Каждая из этих систем решается так же, как и в случае одного теста. Полученные u_j – искомые величины изменений рассматриваемого непрерывного m -го параметра на переходе j .

Отметим, что линейность полученной системы уравнений определяется видом выбранной функции приспособленности. Первоначально авторами в качестве этой функции было выбрано выражение

$$Fitness = \frac{1}{N} \sum_{i=1}^N \sqrt{1 - \frac{Dist(T[i].out, T[i].ans)^2}{Dist(T[i].ans,0)^2}},$$

которое приводило к системе нелинейных уравнений, что резко усложняло решение задачи.

Операторы алгоритма генетического программирования

Неотъемлемой частью любого алгоритма генетического программирования являются операторы отбора, мутации и скрещивания.

Оператор отбора. Этот оператор необходим для выделения из текущего поколения наиболее подходящих для решения задачи особей и добавления их в промежуточное поколение. Для сравнения особей применяется функция приспособленности, сопоставляющая каждой особи число, характеризующее, на-

сколько хорошо автомат, соответствующий особи, подходит для решения задачи. При этом отметим, что чем больше это число, тем лучшей считается особь.

В данной работе используется *турнирный метод* отбора [15]. В этом методе случайным образом выбирается k особей из текущего поколения. Среди них определяется лучшая особь, которая и добавляется в промежуточное поколение. Турнир проводится столько раз, сколько особей в поколении.

Оператор мутации. При применении этого оператора выполняется с заданной вероятностью каждое из действий:

- изменение начального состояния;
- добавление, удаление или изменение случайного перехода в «скелете» автомата.

Под изменением перехода понимается изменение состояния, в которое выполняется переход по условию, на случайно выбранное.

Оператор скрещивания. В скрещивании принимают участие две особи. Результатом применения оператора также являются две особи. Обозначим «родительские» особи как $P1$ и $P2$, а «дочерние» – $C1$ и $C2$. Тогда для стартовых состояний $C1.is$ и $C2.is$ справедливо одно из следующих утверждений:

- $C1.is = P1.is$ и $C2.is = P2.is$;
- $C1.is = P2.is$ и $C2.is = P1.is$.

Далее для каждой ячейки таблицы переходов $t[i][j]$ с равной вероятностью выбирается один из следующих вариантов:

- $C1.t[i][j] = P1.t[i][j]$ и $C2.t[i][j] = P2.t[i][j]$;
- $C1.t[i][j] = P2.t[i][j]$ и $C2.t[i][j] = P1.t[i][j]$.

Эффективность метода

Для проверки эффективности предложенного метода была поставлена задача генерации автомата, обеспечивающего управление самолетом в режиме «мертвая петля».

Взаимодействие сгенерированного автомата с моделью беспилотного самолета. Существуют симуляторы, моделирующие полет самолета. Авторами был выбран свободный кроссплатформенный симулятор *FlightGear* (<http://www.flightgear.org>), который применительно к настоящей работе позволяет осуществлять как ручное, так и автоматное управление моделью самолета. На рис. 2 представлен снимок экрана симулятора *FlightGear* в момент начала разгона.



Рис. 2. Снимок экрана симулятора *FlightGear* в момент начала разгона

Симулятор позволяет осуществлять сохранение параметров полета (скорость, направление полета и т.д.) и параметров самолета (положение руля, элеронов, состояние стартера и т.п.). Параметры полета являются входными параметрами для системы управления, а параметры самолета – управляющими, так как за счет их изменений выполняется управление самолетом. Значения управляющих параметров формируются автоматом (рис. 3).

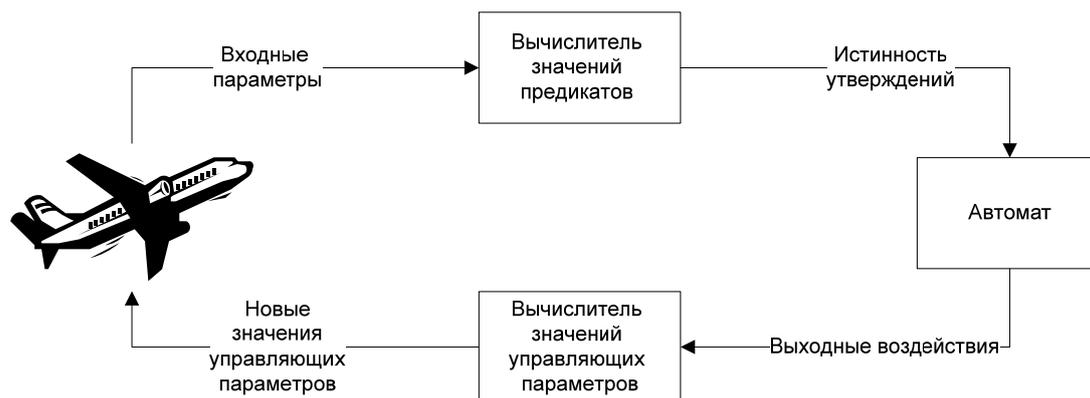


Рис. 3. Управление самолетом с помощью автомата

Задача генерации автомата, выполняющего «мертвую петлю». Эта задача может быть сформулирована следующим образом: требуется построить автомат, под управлением которого модель самолета делает мертвую петлю, а затем ровно пролетает несколько секунд.

Использование предлагаемого метода. Кратко рассмотрим основные шаги применения предлагаемого метода на рассматриваемом примере:

- сформирован необходимый и достаточный набор утверждений о состоянии самолета;
- записаны три набора тестов, которые рассматривались независимо друг от друга:
 - каждый набор состоял из 10 тестов;
 - каждый тест состоял из нескольких тысяч наборов входных и управляющих параметров;
 - опрос параметров производился 10 раз в секунду;
- алгоритм генетического программирования запускался для каждого из трех наборов тестов и каждого набора параметров алгоритма.

Перечислим параметры алгоритма и их значения:

- размер поколения – 100 особей;
- число состояний автомата – от двух до пяти;
- вероятность мутации – 0,5;
- метод отбора – турнирный;
- размер элиты – две особи;
- величина такта – 0,1 с.

Вычисления производились на одном ядре компьютера с процессором Intel Core 2 Duo T7250 с тактовой частотой 2 ГГц под управлением операционной системы Microsoft Windows XP. Язык программирования – Java.

В среднем время работы алгоритма составляло около 10 часов для одного набора параметров и одного набора тестов. При этом было вычислено около двух тысяч поколений. Таким образом, создание и обработка одного поколения в среднем занимала около 20 с или 0,2 с на один автомат, что значительно меньше, чем в работе [13], где это занимало около 5 минут.

Выбранные утверждения:

- x_0 – двигатель включен;
- x_1 – ускорение изменения направления движения самолета больше нуля;
- x_2 – скорость изменения направления движения самолета больше нуля;
- x_3 – величина отклонения от начального направления меньше одного градуса;
- x_4 – величина отклонения от начального направления больше нуля (отклонение влево);
- x_5 – ускорение изменения крена (угла наклона) больше нуля;
- x_6 – скорость изменения крена больше нуля;
- x_7 – крен самолета маленький (меньше одного градуса);
- x_8 – крен больше нуля (самолет завалился на правый бок);
- x_9 – ускорение изменения вертикальной скорости самолета больше нуля;
- x_{10} – скорость изменения вертикальной скорости больше нуля;
- x_{11} – вертикальная скорость маленькая (меньше 0,1 метра в секунду);
- x_{12} – вертикальная скорость больше нуля (самолет поднимается).

Управляющими органами являются магнето, стартер, дроссель, элероны, руль высоты и руль направления. Первые два органа являются дискретными, а остальные – непрерывными. При этом, напри-

мер, воздействие «включить стартер» является дискретным, а «повернуть руль на 0,5 градуса вправо» – непрерывным.

В результате запусков алгоритма генетического программирования было установлено:

- автоматы с тремя-четырьмя состояниями «ведут себя» достаточно хорошо;
- с увеличением числа состояний автомата его структура становится все менее логичной, а поведение ухудшается.

Записанные тесты. Приведем ссылки на некоторые видеозаписи полетов под управлением человека:

- успешное выполнение «мертвой петли» (этот полет вошел в обучающий набор) – <http://www.youtube.com/watch?v=G5Kcx0ohpNo>;
- неудачное выполнение – <http://www.youtube.com/watch?v=OGVTch-a97A>.

Полученные результаты. Было проведено около 50 запусков генетического алгоритма, в каждом из которых выбирался автомат с наибольшим значением функции приспособленности. Эти запуски отличались друг от друга используемыми параметрами и наборами тестов.

Полеты моделей самолетов, управляемых выбранными автоматами, были просмотрены авторами. После этого автомат, используемый в полете, больше других похожем на «идеальную» «мертвую петлю», был назван лучшим. Этот автомат имеет четыре состояния и 68 переходов. При этом отметим, что «идеальная» «мертвая петля» может отличаться от эталонной, которая выполняется вручную.

В процессе наблюдения за ходом выполнения «мертвой петли» под управлением лучшего автомата было установлено, что в зависимости от параметров среды и самолета при запуске автомата возможны три варианта выполнения «петли»:

1. В большинстве случаев модель самолета, как и при управлении вручную, выполняет одну «мертвую петлю» и летит дальше;
2. Иногда модель самолета может выполнить несколько «мертвых петель» с некоторым интервалом. Это происходит в случае, когда значения параметров модели самолета в конце выполнения «мертвой петли» схожи со значениями параметров в начале ее выполнения. Это приводит к тому, что если автомат после осуществления «петли» находится в том же состоянии, в котором был в начале, то поведение модели заикливается. В ходе экспериментов авторы неоднократно наблюдали выполнение двух «мертвых петель» подряд. Выполнение большего числа «петель» не наблюдалось;
3. Модель может вообще не выполнить «мертвую петлю», так как автомат не справляется с управлением, что, правда, бывает крайне редко.

Определение условий, при которых выполняется каждый из этих вариантов полета, требует дальнейших исследований.

Пример полета самолета под управлением лучшего из полученных автоматов

Ниже приведены ссылки на видеозаписи трех вариантов реализации «мертвой петли» под управлением лучшего автомата:

1. Выполнение «мертвой петли», близкое к «идеальному» (<http://www.youtube.com/watch?v=TzrLoJjVTA>);
2. Выполнение «мертвой петли» с заваливанием на левый борт с последующим выравниванием (<http://www.youtube.com/watch?v=C6WV7x2bqE8>);
3. Последовательное выполнение двух «мертвых петель» (<http://www.youtube.com/watch?v=yFiG4yz67Ks>).

Заключение

Разработан метод генетического программирования на основе обучающих примеров для построения конечных автоматов, управляющих объектом со сложным поведением. Выполнено экспериментальное исследование предложенного метода, и показано, что автоматически сгенерированный автомат может обеспечивать лучшее управление, чем ручное. В большинстве случаев автоматически сгенерированный автомат обеспечивает корректное выполнение задания. Показано, что предложенный метод позволяет значительно быстрее оценивать генерируемые автоматы по сравнению с прототипом [13]. Предложенный метод был применен к задаче управления моделью беспилотного самолета в режиме «мертвая петля».

Кроме более высокого быстродействия метода, еще одно его преимущество по сравнению с вычислением функции приспособленности с помощью моделирования состоит в том, что в предложенном методе эту функцию не требуется изменять как при переходе от одного режима к другому, так и при переходе от одного объекта к другому.

Исследование выполнено по Федеральной целевой программе «Научные и научно-педагогические кадры инновационной России на 2009–2013 годы» в рамках государственного контракта П1188 от 27 августа 2009 года.

Литература

1. Поликарпова Н.И., Шалыто А.А. Автоматное программирование. – СПб: Питер, 2010. – 176 с.
2. Angeline P., Pollack J. Evolutionary Module Acquisition // Proceedings of the Second Annual Conference on Evolutionary Programming. Cambridge: MIT Press, 1993. – P. 154–163.
3. Jefferson D., Collins R., Cooper C., Dyer M., Flowers M., Korf R., Taylor C., Wang A. The Genesys System: Evolution as a Theme in Artificial Life // Proceedings of Second Conference on Artificial Life. – MA: Addison-Wesley, 1992. – P. 549–578.
4. Царев Ф.Н., Шалыто А.А. Применение генетического программирования для генерации автомата в задаче об «Умном муравье» / Сборник трудов IV-ой Международной научно-практической конференции. – М.: Физматлит, 2007. – Т. 2. – С. 590–597.
5. Бедный Ю.Д., Шалыто А.А. Применение генетических алгоритмов для построения автоматов в задаче «Умный муравей». – СПбГУ ИТМО, 2007 [Электронный ресурс]. – Режим доступа: <http://is.ifmo.ru/works/ant>, свободный. Яз. рус. (дата обращения 09.02.2011).
6. Парашенко Д.А., Царев Ф.Н., Шалыто А.А. Технология моделирования одного класса мультиагентных систем на основе автоматного программирования на примере игры «Соревнование летающих тарелок». Проектная документация. – СПбГУ ИТМО, 2006. [Электронный ресурс]. – Режим доступа: <http://is.ifmo.ru/unimod-projects/plates/>, свободный. Яз. рус. (дата обращения 09.02.2011).
7. Coates A., Abbeel P., Ng A.Y. Learning for Control from Multiple Demonstrations // Proceedings of the 25th International Conference on Machine Learning. – Helsinki, 2008. – P. 144–151.
8. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. – М.: Физматлит, 2006.
9. Рассел С., Норвиг П. Искусственный интеллект: современный подход. – М.: Вильямс, 2006.
10. Koza J.R. Genetic programming: On the Programming of Computers by Means of Natural Selection. – Cambridge: MIT Press, 1992.
11. Курейчик В.М. Генетические алгоритмы. Состояние. Проблемы. Перспективы // Известия РАН. Теория и системы управления. – 1999. – № 1. – С. 144–160.
12. Курейчик В.М., Родзин С.И. Эволюционные алгоритмы: генетическое программирование // Известия РАН. Теория и системы управления. – 2002. – № 1. – С. 127–137.
13. Поликарпова Н.И., Точилин В.Н., Шалыто А.А. Метод сокращенных таблиц для генерации автоматов с большим числом входных переменных на основе генетического программирования // Известия РАН. Теория и системы управления. – 2010. – № 2. – С. 100–117.
14. Царев Ф.Н. Метод построения управляющих конечных автоматов на основе тестовых примеров с помощью генетического программирования // Информационно-управляющие системы. – 2010. – № 5. – С. 31–36.
15. Яминов Б. Генетические алгоритмы. – СПбГУ ИТМО, 2005. [Электронный ресурс]. – Режим доступа: <http://rain.ifmo.ru/cat/view.php/theory/unordered/genetic-2005>, свободный. Яз. рус. (дата обращения 09.02.2011).

<i>Александров Антон Вячеславович</i>	– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, alantbox@gmail.com
<i>Казаков Сергей Владимирович</i>	– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, kazakov_serгей_v@mail.ru
<i>Сергушичев Алексей Александрович</i>	– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, alsergbox@gmail.com
<i>Царев Федор Николаевич</i>	Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, fedor.tsarev@gmail.com
<i>Шалыто Анатолий Абрамович</i>	Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, зав. кафедрой, shalyto@mail.ifmo.ru

**РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ
МАЛОРАЗМЕРНЫМ ВЕРТОЛЕТОМ****В.О. Клебан, А.А. Шалыто**

Приводится пример построения системы автоматического управления беспилотным малоразмерным вертолетом с соосной схемой расположения винтов. В данной работе методы теории управления и цифровой обработки сигналов успешно дополнены парадигмой автоматного программирования, которая позволяет эффективно строить системы, обладающие сложным поведением.

Ключевые слова: имитационное моделирование, автоматное программирование, соосный вертолет.

Введение

В настоящее время ряд исследовательских групп изучает вопросы построения систем управления для беспилотных летательных аппаратов, действующих внутри помещений [1–4]. При этом можно выделить три основных направления исследований: аэродинамика летающих роботов, разработка алгоритмов планирования и управления полетом, разработка сенсорики для оценки состояния окружающего пространства и обеспечения безопасности полета в стесненных условиях. Например, в работе [5] представлена реальная система управления легким вертолетом, реализованная на находящейся в продаже серийной модели легкого вертолета.

Многие из проводимых исследований направлены на создание летающих роботов для спасательных операций, где одной из задач является обеспечение автономности и применение любых внешних ориентиров, таких как видеокамеры, невозможно. Разработка летающих моделей связана с тем, что наземные роботы медлительны, а их способность преодолевать пересеченную местность сильно ограничена. В отличие от наземного робота, летающий робот является практически идеальным инструментом для оценки обстановки на местности. В связи со спецификой решаемых задач такой робот должен иметь возможность совершать полет в закрытом помещении для обеспечения возможности сбора информации внутри зданий и сооружений.

Устройство микровертолета соосной схемы

В целях исследования динамики соосного микровертолета и разработки системы управления был приобретен соосный радиоуправляемый вертолет фирмы Walkera.

Управление вертолетом осуществляется по четырем каналам. Вертикальное движение осуществляется при помощи оборотов двигателя, управление курсом происходит при помощи рыскания, горизонтальный полет регулируется при помощи крена и тангажа. В силу того, что управление по высоте возможно лишь при помощи изменения оборотов двигателей, вертолет имеет достаточно медленный отклик по каналу вертикальной скорости.

Два несущих винта продуцируют все основные силы и моменты, действующие на вертолет. Управление верхним винтом производится при помощи аппарата Хиллера, который замедляет его реакцию на резкие изменения крена и тангажа за счет гироскопического момента, создаваемого стабилизатором. Нижний винт управляется посредством двух микросервомоторов, которые соединены с ним при помощи тарелки перекося. Скорость вращения винтов контролируется двумя бесколлекторными электродвигателями. Приобретенный вертолет спроектирован специально для полетов внутри помещения, что привело к ослаблению некоторых характеристик, не требующихся для подобных полетов.

Силовая установка и имеющаяся система радиосвязи приобретенного вертолета подверглись существенной модернизации. Также на вертолет были установлены бортовая ЭВМ и твердотельная курсовертикаль собственной разработки.

Модернизация силовой установки включала в себя замену коллекторных двигателей постоянного тока на бесколлекторные двигатели повышенной мощности. В дополнение к двигателям были установлены усилительно-преобразовательные устройства для управления двигателями при помощи широтно-импульсной модуляции и карманы для перехвата воздушного потока с целью охлаждения электродвигателей. Законцовки лопастей были утяжелены для снижения конусности несущего винта и позволили избавиться от флаттера, возникающего при больших оборотах двигателя.

Устройство бортовой ЭВМ

Бортовая ЭВМ (рис. 1) представляет собой контроллер на базе архитектуры ARM7, снабженный тремя датчиками угловых скоростей, акселерометром, магнитометром, а также входами для подключения датчиков выотомера и дальномеров.

На базе данной ЭВМ была разработана курсовертикаль, показывающая ориентацию вертолета в пространстве и вычисляющая углы Эйлера в режиме реального времени.

Основными проблемами при разработке курсовертикали оказались [1, 6, 7]: температурный дрейф и «уход нулей» показаний датчиков угловых скоростей, что приводило к линейно возрастающей угловой

ошибке; ошибки калибровки, приводящие к росту угловых ошибок в зависимости от количества движений вертикали; собственный шум датчика и шум датчика вследствие вибраций корпуса вертолета, приводящие к росту угловой ошибки.

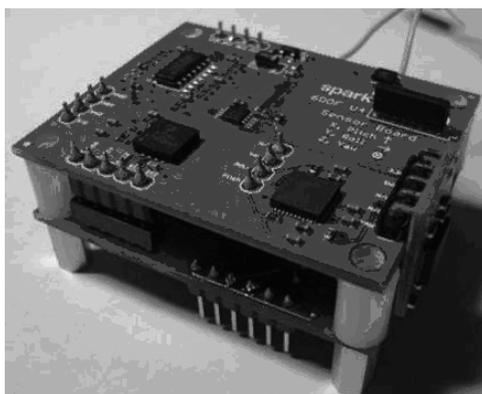


Рис. 1. Бортовая ЭВМ вертолета

В целях обеспечения достоверности показаний прибора производилась его тарировка путем многократных поворотов устройства вдоль различных осей и при различных температурах на специальном стенде. Это позволило добиться высокой стабильности и достоверности показаний курсовертикали.

Для решения задачи фильтрации показаний приборов была разработана схема фильтрации сигналов (рис. 2) позволяющая в значительной степени снизить негативные эффекты от попадания помех в систему [8, 9]. Отметим, что борьба с помехой производилась не только на уровне электронных схем, но и при компоновке вертолета.

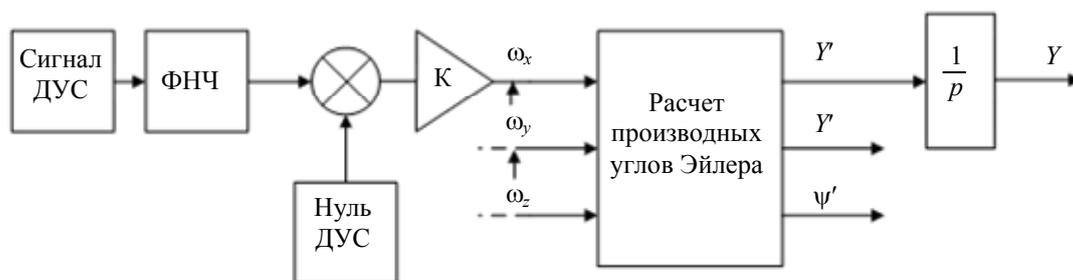


Рис. 2. Схема измерения угла по одному из каналов ДУС

К сожалению, избавиться от эффекта «ухода нуля» в предлагаемой схеме фильтрации не удалось, поэтому был разработан алгоритм, позволяющий производить оценку нулей датчиков угловых скоростей непосредственно в полете (рис. 3).

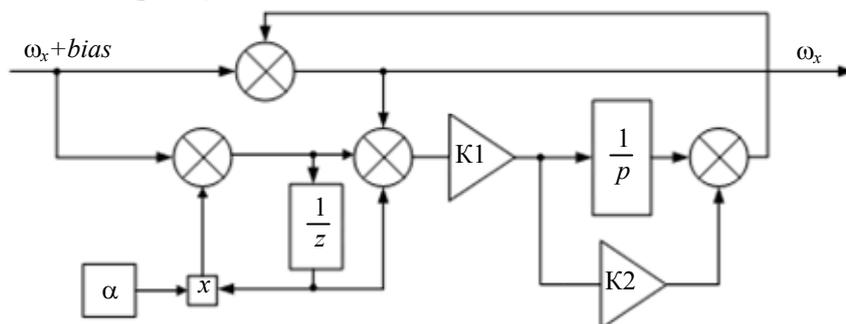


Рис. 3. Схема оценки нуля датчика угловой скорости в полете

Сбор и выдача данных происходят с частотой 200 Гц.

Устройство системы связи

Система связи микровертолета представляет собой цифровую радиолинию, работающую на частоте 2 ГГц. Скорость передачи данных составляет 115200 бит/с. Дальность радиосвязи составляет около полутора километров.

К сожалению, имеющиеся средства организации радиосвязи не удовлетворяют условиям разрабатываемой системы. Исследовательский характер проводимых работ делает использование пакетной радиосвязи неудобным. Это проявляется в том, что частые изменения состава передаваемых данных вынуждают каждый раз вносить изменения в состав пакетов. Кроме того, стояла задача реализовать такой протокол передачи данных, который бы не блокировал канал в случае неудавшейся передачи данных, что часто происходит при реализации протоколов, заголовок которых содержит объем принимаемых данных.

Для решения данной задачи было предложено пойти по пути расширения языковых возможностей языка С путем введения нового ключевого слова linked, означающего, что переменная будет синхронизирована между бортовой ЭВМ и наземным компьютером. Например, инструкция static linked int throttle = 0; означает, что целочисленная переменная с именем throttle и стартовым значением 0 будет доступна как на бортовой ЭВМ, так и на наземной, причем ее изменения будут производиться синхронно на обеих машинах. Все переменные, объявленные с модификатором linked, могут контролироваться из менеджера переменных, позволяющего производить изменение переменных вручную, а также наблюдать за их изменением в графическом виде. На программную реализацию данной идеи было получено авторское свидетельство № 2009615456 «Средство обеспечения взаимодействия автоматных программ».

Устройство системы обнаружения препятствий

Система обнаружения препятствий представляет собой набор инфракрасных датчиков расстояния [10], расставленных по периметру вертолета, и позволяет осуществлять контроль приближения вертолета к опасным объектам.

Применение автоматного программирования

Практически с самого начала экспериментов стало ясно, что объект обладает сложным поведением как с точки зрения логики согласования всех систем, так и с точки зрения непосредственно управления полетом. В связи с этим, для реализации системы управления решено было применить автоматное программирование, эффективность применения которого для реактивных систем и мобильных роботов показана в работах [11–18]. Технология автоматного программирования заключается в представлении программ в виде системы взаимодействующих автоматов. Взаимодействие автоматов может быть реализовано за счет вложенности, вызываемости, обмена сообщениями, работы над общим входом и т.д. Использование автоматного подхода при создании подобных систем управления обладает рядом достоинств:

- документуемость;
- возможность построения компактных моделей Крипке, а значит, верифицируемость;
- возможность осуществления формального и изоморфного преобразования графа переходов в код на языке программирования.

Основной цикл системы управления вертолетом (рис. 4) описывается автоматом из десяти состояний.

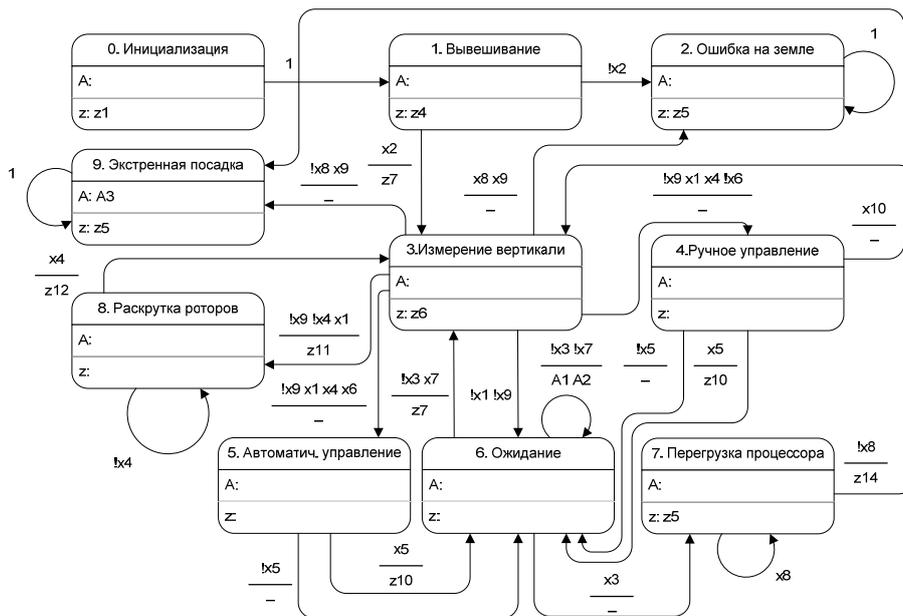


Рис. 4. Главный автомат управления вертолетом

На вход автомату передаются следующие переменные:

- x1 – запуск разрешен;
- x2 – успешное вывешивание;
- x3 – таймер истек до входа в Ожидание;
- x4 – ротор раскручен;
- x5 – такт управления;
- x6 – режим управления «Ручной»;
- x7 – истек таймер;
- x8 – нахожусь на земле;
- x9 – слишком низкая частота работы САУ;
- x10 – установлена связь с землей.

В качестве выхода автомат использует следующие воздействия:

- z1 – инициализация;
- z4 – вывешивание;
- z5 – сообщить об ошибке;
- z6 – фильтрация данных;
- z7 – сброс таймера;
- z9 – останов;
- z10 – вывод управления в канал ШИМ;
- z11 – раскрутка ротора;
- z12 – сброс курсовертикали;
- z14 – снижение частоты работы САУ;
- z15 – повышение частоты работы САУ.

Аналогично при помощи автоматов описываются все остальные системы вертолета: опрос датчиков, управление полетом, управление радиосвязью и т.д.

Заключение

Динамика полета вертолета – сложный нелинейный процесс с неучтенной динамикой. К сожалению, построение полной физической модели вертолета на данном этапе недоступно в силу ее сложности, наличия массы специфичных для каждой конкретной модели вертолета (или даже экземпляра) моментов, наличия неучтенной динамики и т.д.

Изложенный автоматный подход, позволяющий проводить имитационное моделирование управления вертолетом, обеспечил возможность создания системы управления без использования модели объекта управления. В данном случае методы теории управления и цифровой обработки сигналов успешно дополнены парадигмой автоматного программирования, которая позволяет эффективно строить системы, обладающие сложным поведением, а также сильносвязанные системы.

Дальнейшие исследования в этой области требуют значительно больших ресурсов. Вот, что, например, пишет А. Гоголь, ректор СПбГУ телекоммуникаций им. М.А. Бонч-Бруевича: «В одном из университетов США я наблюдал за продвижением группы из семи человек, занимающихся геликоптером. Бюджет – миллион долларов на каждого в год, причем без особых отчетов. Не знаю, на каком этапе они сейчас, но тогда их модели спокойно летали брюхом вверх. Я такого в жизни не видел».

Также вертолетами занималось и занимается группа из 15 человек в Цюрихском политехническом институте [19], и это, как отмечено выше, не единственный университет в мире.

Исследование выполнено по Федеральной целевой программе «Научные и научно-педагогические кадры инновационной России на 2009–2013 годы» в рамках государственного контракта П1188 от 27 августа 2009 года.

Литература

1. Castillo P., Lozano R., Dzul A.E. Modelling and Control of Mini-Flying Machines . –Springer, 2005.
2. Watkinson J. The Art of Helicopter. – Elsevier, 2004.
3. Bermes C., Sartori K., Schafroth D., Bouabdallah S., Siegwart R. Control of a Coaxial Helicopter with Center of Gravity Steering // Proceedings of Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS. – 2008. – P. 492–500.
4. Kotmann M. Software for Model Helicopter Flight Control. – Zurich: Institute of Computer Systems, 1999.
5. Барабанов А.Е., Ромаев Д.В. Автопилот вертолета по телевизионным наблюдениям // Гироскопия и навигация. – 2006. – № 4. – С. 106–107.
6. Локк А.С. Управление снарядами. – М.: Физматлит, 1958.
7. Горбатенко С.А., Макашов Э.М. Механика полета. – М.: Машиностроение, 1969.
8. Лайонс Р. Цифровая обработка сигналов. – М.: Бином, 2007.

9. Айфичер Э., Джервис Б. Цифровая обработка сигналов: практический подход. – М.: Вильямс, 2008.
10. Sharp distance sensors [Электронный ресурс]. – Режим доступа: <http://www.parallax.com/dl/docs/prod/acc/SharpGP2D12Snrs.pdf>, свободный. Яз. англ. (дата обращения 09.02.2011).
11. Туккель Н.И., Шалыто А.А. Система управления дизель-генератором (фрагмент). Программирование с явным выделением состояний. Проектная документация [Электронный ресурс]. – Режим доступа: <http://is.ifmo.ru/projects/dg/>, свободный. Яз. рус. (дата обращения 09.02.2011).
12. Туккель Н.И., Шалыто А.А. От тьюрингова программирования к автоматному // Мир ПК. – 2002. № 2. – С. 144–149 [Электронный ресурс]. – Режим доступа: <http://is.ifmo.ru/works/turing/>, свободный. Яз. рус. (дата обращения 09.02.2011).
13. Шалыто А.А. Switch-технология. Алгоритмизация и программирование задач логического управления. – СПб: Наука, 1998. – 628 с.
14. Шалыто А.А. Логическое управление. Методы аппаратной и программной реализации. – СПб: Наука, 2000. – 780 с.
15. Шопырин Д.Г., Шалыто А.А. Синхронное программирование // Информационно-управляющие системы. – 2004. – № 3. – С. 35–42 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/works/sync_prog/, свободный. Яз. рус. (дата обращения 09.02.2011).
16. Клебан В.О., Шалыто А.А. Использование автоматного программирования для построения многоуровневых систем управления мобильными роботами // Сборник тезисов 19 Всероссийской научно-технической конференции «Экстремальная робототехника». – СПб: ЦНИИ РТК, 2008. – С. 85–87.
17. Brooks R.A. A Robust Layered Control System for a Mobile Robot // IEEE Journal of Robotics and Automation. – 1986. – V. 2. – P. 14–23.
18. Harel D., Pnueli A. On the development of reactive systems / In «Logic and Models of Concurrent Systems». – NATO Advanced Study Institute on Logic and Models for Verification and Specification of Concurrent Systems. – Springer Verlag, 1985. – P. 477–498.
19. Flying Machine Arena [Электронный ресурс]. – Режим доступа: http://www.idsc.ethz.ch/Research_DAndrea/FMA, свободный. Яз. англ. (дата обращения 09.02.2011).

Клебан Виталий Олегович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, vk.developer@gmail.com

Шалыто Анатолий Абрамович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, зав. кафедрой, shalyto@mail.ifmo.ru

УДК 004.832.28

ПРИМЕНЕНИЕ ДВУХЭТАПНОГО ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ ДЛЯ ПОСТРОЕНИЯ МОДЕЛИ ТАНКА В ИГРЕ «ROBOCODE» Д.О. Соколов

Рассматривается применение генетического программирования для построения конечных автоматов, управляющих системами со сложным поведением. Приведен метод двухэтапного генетического алгоритма, основанный на идеях динамического программирования. Рассмотрено применение этого метода на примере игры «Robocode».

Ключевые слова: автомат, деревья разбора, генетическое программирование.

Введение

Задача построения управляющих систем для беспилотной техники в настоящее время является актуальной. Чаще всего эта задача решается вручную. Такой подход не всегда является эффективным как ввиду больших затрат ресурсов, так и в связи с низким качеством построенных систем. В некоторых случаях ввиду сложности системы построение управляющей системы при помощи ручного труда невозможно. Естественная идея – поручить построение управляющей системы компьютеру.

Эволюционные вычисления успешно применяются для автоматического создания программ [1]. Эффективность применения эволюционных алгоритмов напрямую зависит от способа представления программы в виде хромосомы [2]. Для многих управляющих систем их поведение удобно представлять в виде конечных автоматов [3, 4]. Одной из таких систем является система управления танком в игре «Robocode», которая и рассматривается в настоящей работе.

Выделим основные проблемы, которые решаются в данной работе, возникающие при использовании генетического программирования для решения задачи управления:

- большое время работы генетического алгоритма;
- необходимость обработки вещественных переменных при помощи конечных автоматов;
- попадание в локальные максимумы.

Автор предполагает, что читатель знаком с основными понятиями генетических алгоритмов: особь, генетические операторы (скрещивание, мутация), фитнес-функция, поколение, классический генетический алгоритм.

Постановка задачи управления

Для описания предлагаемого метода была выбрана задача управления танком в игре «Robocode». У танка необходимо управлять следующими устройствами: радаром (*Radar*), пушкой (*Gun*) и системой движения (*Body*).

Управление танком осуществляется аналогично работе [5]. В каждый момент времени анализируется текущая ситуация (положение танка, его скорость и т.д.). На основе этой информации формируются четыре действия: передвижение (вперед/назад), поворот, поворот пушки, стрельба. Указанные действия формируются в результате интерпретации функции управления, которая генерируется на основе генетических алгоритмов.

Конкретизируем постановку задачи управления. Обозначим текущее время в игре как $t \in N$. Обозначим множество позиций как S , а множество действий танка как A .

Позиция – элемент некоторого множества всех возможных ситуаций в игре в выбранный момент времени. Позиция включает в себя положение каждого танка, его скорость, направление, угол поворота пушки и радара, энергию, информацию о танке соперника и т.д.

Задача управления для рассматриваемой игры в общем случае состоит в задании для каждого момента времени t функции управления $f_t : S^t \Rightarrow A$, которая на основании информации о позициях во все предыдущие моменты времени (получаемой из среды «Robocode») определяет действие на объект управления в текущий момент времени. Таким образом, решение задачи управления танком – вектор $[f_1, f_2, \dots]$.

Упростим задачу управления. Пусть:

1. действие танка в текущий момент времени зависит не только от позиции в этот момент времени, а также от состояния самой системы управления;
2. зафиксирован алгоритм управления радаром – вращение по кругу;
3. при выборе действия анализируется лишь упрощенная позиция – элемент множества $S^t = (x, y, dr, tr, w, dh, GH, h, d, e, E)$, здесь: x, y – координаты танка соперника относительно нашего танка; dr – расстояние, которое осталось «доехать» нашему танку (после вызова метода *AdvancedRobot.setAhead*); tr – угол, на который осталось повернуться нашему танку; w – расстояние от нашего танка до края поля; dh – угол между направлением на танк соперника и пушкой нашего танка; GH – угол поворота пушки нашего танка; h – направление движения танка соперника; d – расстояние между нашим танком и танком соперника; e – энергия танка соперника; E – энергия нашего танка;
4. множество действий $A^t = g, p, d, h$, где g – угол поворота пушки; p – энергия снаряда (неположительные значения означают, что выстрел не производится); d – расстояние, на которое перемещается танк; h – угол поворота танка.

Учитывая изложенное, задача построения системы управления танком сведена к заданию функции $f : S^t \times S \rightarrow A^t \times S$, где S – конечное множество состояний системы. Естественно представлять данную функцию в виде автомата, который будем искать при помощи генетического алгоритма.

Основные идеи

Как отмечалось во введении, в данной работе рассматривается метод решения трех проблем, возникающих при использовании генетических алгоритмов для решения задачи управления. Рассмотрим методы решения.

Попадание в локальные максимумы. Для решения данной проблемы использован островной генетический алгоритм [6] (рис. 1), а также оператор большой мутации.

Основные отличия островного алгоритма от классического генетического:

- разделение популяции на несколько популяций, развивающихся независимо – разделение на острова;
- добавление оператора миграции [6].

Оператор большой мутации на фиксированной доле островов заменяет поколение на случайное. Данный оператор применяется через фиксированное число поколений.

Обработка вещественных переменных. Как уже отмечалось, будем строить конечный детерминированный автомат с множеством состояний S . В каждое состояние поместим обработчик переменных. Обработчик i -ого состояния представляет собой функцию $f_i : S^i \rightarrow A^i$ (см. раздел «Постановка задачи управления»).

Зафиксируем некоторое число k , $2^k \leq |S|$, сопоставим автомату еще один обработчик переменных – $g : S^k \rightarrow \{0,1\}^k$. Функция переходов автомата будет иметь вид $G : S \times S^k \rightarrow S, G(s) = H(S, g(S^k))$.

Задача получения автомата свелась к задаче получения функций f_i , g и H . При помощи выделения обработчиков, обработка вещественных переменных была «отделена» от автомата, таким образом, появилась возможность применять любой из имеющихся методов для получения функции H .

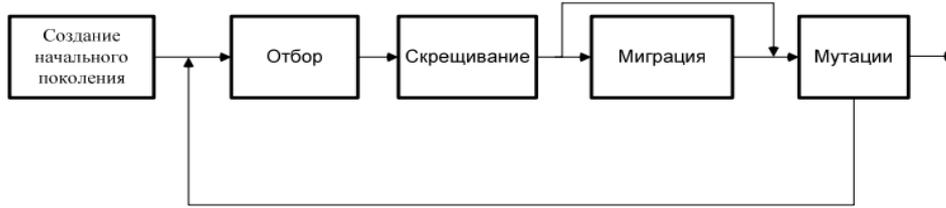


Рис. 1. Схема островного генетического алгоритма

Уменьшение времени работы генетического алгоритма. Время работы напрямую связано с размером пространства решений. Для уменьшения размера данного пространства будем использовать идеи динамического программирования [7].

В результате применения метода деревьев разбора задача разбилась на две части:

1. получение f_i ;
2. получение g и H .

В соответствии с принципом динамического программирования необходимо сначала решить подзадачи оптимальным образом. Подзадачами будем считать нахождение f_i . В нашем случае сложно сформулировать, что значит «оптимально», поэтому на первом этапе при помощи генетического алгоритма сформируем некоторое множество f_i – репозиторий обработчиков (состояний). На втором этапе будем получать функции g и H , но вместо случайных f_i будем случайным образом выбирать их из репозитория (сразу отметим, что финальная реализация будет несколько отличаться от этого, по причине того, что можно развить эту идею, пользуясь особенностями задачи).

Таким образом, осталось разработать метод генерации вещественных функций.

Представление обработчика вещественных переменных

Предлагаемый метод представления особи является комбинацией методов, описанных в работах [5, 8], f_i (см. определение в разд. «Основные идеи») будет представлять собой четверку так называемого дерева разбора.

Дерево разбора представляет собой дерево, в котором во внутренних вершинах находятся функции. У каждой внутренней вершины ровно столько потомков, какова аридность функции в данной вершине. В листья подаются входные переменные или заранее зафиксированные константы.

В данной работе, аналогично работе [5], использованы внутренние функции $if(x < y) return w else return v$; $\min(x, y)$; $x \times y$; $-x$; $\frac{1}{1 + \exp(-x)}$; $x + y + w$; $x \times y \times w$ и константы: 0,1; 0,5; 1;

2; 5; 10.

Пример дерева разбора изображен на рис. 2.

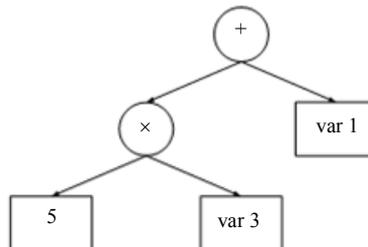


Рис. 2. Пример дерева разбора

Условное обозначение $var i$ означает, что в данных лист подается значение i -ой переменной. Генетические операторы для деревьев разбора аналогичны описанным в работе [8]. Таким образом, ав-

томат представляет собой массив, состоящий из четверок деревьев разбора, k деревьев разбора – функция g , а также таблица переходов – функция H .

Генетический алгоритм

Как говорилось ранее, на всех этапах построения используется островной генетический алгоритм. На первом этапе строятся четверки деревьев разбора, эти четверки сохраняются в репозитории. Причем для каждого из деревьев известно, за какое из действий танка он отвечает. На втором этапе генерируется функция переходов для автомата.

Общие для двух этапов элементы генетического алгоритма

В качестве основной стратегии формирования следующего поколения используется элитизм. При рассмотрении текущего поколения отбрасываются все особи, кроме некоторой доли наиболее приспособленных – элиты. Эти особи переходят в следующее поколение. После этого оно дополняется в определенной пропорции случайными особями – особями из текущего поколения, которые мутировали и являются результатами скрещивания особей из текущего поколения (отдельно отметим, что скрещиваться могут не только элитные особи, а все). Особи, «имеющие право» давать потомство, определяются «в поединке»: выбираются две случайные пары особей, и более приспособленная особь в каждой из них становится одним из родителей. Через фиксированное число поколений каждый остров меняется с другим случайным числом случайно выбранных элитных особей.

Через заранее заданное число поколений происходит *большая мутация* – фиксированная доля островов заменяется островами со случайными особями. Проведение мутации в момент, когда функция приспособленности «элитных» особей изменяется незначительно, невозможно, так как за счет миграции особей между островами среднее значение функции приспособленности постоянно изменяет свое значение.

Первый этап генетического алгоритма

На первом этапе генетического алгоритма выращиваются четверки деревьев разбора. Начиная с заранее заданного поколения, лучшая особь поколения добавляется в репозиторий. При генерации начального поколения все острова заполняются случайно сгенерированными особями. Скрещивание деревьев разбора происходит попарно.

Для подсчета функции приспособленности создается автомат из одного состояния (для него не нужны функции g и H), для которого f_1 является тестируемой четверкой деревьев разбора. При подсчете фитнес-функции используется моделирование соревнования между танком, управляемым тестируемой особью, и выбранным танком. Заранее выбрать фитнес-функцию не представляется возможным, так как на первом этапе генерируется не вся особь в целом, а лишь одно состояние. В данной работе для сравнения использовались различные фитнес-функции:

- $t.damage + t.bulletdamage$;
- $\frac{(t.survival - n)n}{t.bulletDamage}$;
- $\frac{t.score + t.survival * t.bulletDamage}{t.score + t.survival * t.bulletDamage + e.score + e.survival * t.bulletDamage}$;
- $\frac{t.score}{t.score + e.score}$;
- $t.score$.

Здесь: t – модель танка, управляемая тестируемой особью; e – модель танка противника; $damage$ – ущерб, нанесенный противнику за счет попаданий и столкновений; $bulletDamage$ – ущерб, нанесенный противнику только за счет попаданий; $survival$ – число раундов, в которых танк выжил; $score$ – число очков, набранных танком (вычисляется средой «Robocode»); n – число раундов в соревновании. Для избежания излишней пассивности танков (отсутствия стрельбы) часть запусков алгоритма проводилась с запретом на движение тестируемого танка.

Второй этап генерации особей

После первого этапа становится доступен репозиторий с деревьями разбора, где деревья сгруппированы по действиям, за которые они отвечают.

Теперь особь представляет собой тройку (S, H, g) , где S – множество состояний, каждое из которых представляет собой четверку деревьев разбора из репозитория; H – функция переходов автомата, представляет собой полную таблицу переходов, генетические операторы аналогичны операторам, описанным в работе [9]; g – подфункция H , по состоянию в игре возвращающая номер перехода, представляет собой k деревьев разбора.

При скрещивании элементы данной тройки скрещиваются попарно. При скрещивании S и S' элементы этих множеств также скрещиваются попарно. При скрещивании состояний P и Q возможны следующие ситуации (наследники P' и Q'):

- $P' := P, Q' := Q$;
- $P' := Q, Q' := P$;
- P' и Q' случайным образом выбирают деревья разбора от родителей.

При генерации случайного автомата, для каждого состояния из репозитория равновероятно выбирается по одному дереву разбора, отвечающему за каждое действие (в этом отличие от раздела «Основные идеи»: функции f_i можно разбить на части). Стартовое состояние генерируется случайным образом. Деревья разбора, отвечающие за функцию g , генерируются случайным образом.

Также отметим, что операторы скрещивания и мутации к деревьям разбора, находящимся внутри состояний, не применяются.

На этом этапе использовалась следующая фитнес-функция: $\frac{t.score}{t.score + e.score}$.

Основные особенности применения двухэтапного алгоритма

Для начала оценим размер пространства решений в случае применения двухэтапного генетического алгоритма.

Для деревьев, отвечающих за логические переменные, соотношение не изменилось: $(Z^E)^{V_c}$. Также не изменилось соотношение переходов: $N^{2^{V_c}}$. Однако для генерации состояния теперь достаточно лишь выбрать четыре дерева из репозитория. В этом случае имеет место соотношение S^4 , где S – размер репозитория. Таким образом, $|\Omega_{tree(new)}| = (Z^E)^{V_c} S^4 N^{2^{V_c}}$. При этом $|\Omega_{tree(new)}| \ll |\Omega_{tree}|$.

Основные настройки генетического алгоритма

Перечислим настройки генетического алгоритма, используемые при генерации особей: число островов – 4; число особей на острове – 52; доля элитных особей – 0,15; вероятность мутации – 0,02; период миграции – 11; период большой мутации – 140; доля островов, уничтожаемых при большой мутации – 0,8; $k=2$.

Результаты

В качестве основных соперников использовались следующие танки из стандартного набора игры «Robocode»: `sample.Fire`; `sample.Tracker`; `sample.Target`; `sample.Walls`.

На рис. 3 представлен граф переходов автомата, построенного с помощью генетического алгоритма.

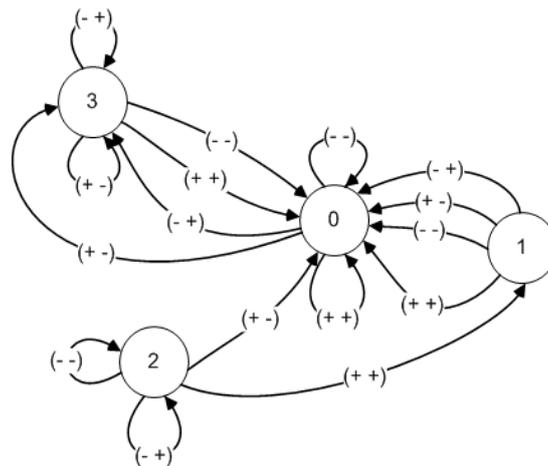


Рис. 3. Граф переходов автомата, полученного в результате работы второго этапа генетического алгоритма

Поясним используемые обозначения. На ребрах графа переходов находятся отметки вида (+, -). Они указывают на значение i -го бита функции g (+ соответствует единице).

Структуру состояний изобразить не представляется возможным из-за размеров деревьев разбора.

Результаты соревнования данной особи проиллюстрированы на рис. 4.

Rank	Robot Name	Total Score	Survival	Surv Bonus	Bullet Dmg	Bullet Bonus	Ram Dmg * 2	Ram Bonus	1sts	2nds	3rds
1st	core.tank.GVarTa...	4178 (70%)	3050	960	0	0	168	0	96	4	0
2nd	sample.Tracker	1822 (30%)	200	40	1500	35	48	0	4	96	0

Рис. 4. Результаты соревнования

На первом этапе проводились запуски танка для всех вариантов фитнес-функции против всех соперников. Таким образом, размер репозитория составил 304 дерева разбора для каждого из четырех типов.

По результатам сравнения предлагаемый в настоящей работе метод превосходит методы, предложенные в работе [5]. При этом отметим, что разница в результатах при применении методов с применением конечных автоматов и *karva*-деревьев лежит в пределах погрешности измерений функции приспособленности. Однако при использовании метода, предложенного в данной работе (несмотря на вдвое увеличенное число раундов соревнований при подсчете фитнес-функции), наблюдается по сравнению с работой [5] ускорение процесса получения особи (на втором этапе) с заданным значением функции приспособленности в среднем в полтора раза (примерно 1,5–2 часа).

Также данный метод позволяет достичь большей универсальности, так как на первом этапе генерируются различные *хорошие* стратегии поведения – каждое состояние представляет собой уже законченную стратегию поведения против конкретного поведения противника.

Из недостатков предложенного метода отметим длительное время проведения первого этапа алгоритма. На создание репозитория из 300 четверок деревьев потребовалось восемь суток.

Заключение

В данной работе задача построения управляющего автомата для систем со сложным поведением разбита на части:

1. Построение обработчиков вещественных переменных в одном отдельно взятом состоянии. При этом возможна более точная настройка желаемого поведения в данном состоянии;
2. Построение графа переходов и вспомогательных конструкций (например, дополнительные деревья разбора для генерации булевых переменных).

Разбиение на части позволило улучшить качество решений и сократить требуемые вычислительные ресурсы (для второго этапа) в случае, когда подсчет для первого этапа выполнен заранее. К недостаткам метода можно отнести длительное время работы первой части алгоритма.

Для каждой части были реализованы модификации островного генетического алгоритма.

Литература

1. Holland J.P. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. – The University of Michigan Press, 1975.
2. Koza J.R. Genetic programming: On the Programming of Computers by Means of Natural Selection. – Cambridge: MIT Press, 1992.
3. Aho A., Sethi R., Ullman J. Compiler Design: Principles, Tools, and Techniques. Наука, МА: Addison Wesley, 1986.
4. Шалыто А.А. Switch-технология. Алгоритмизация и программирование задач логического управления. – СПб: Наука, 1998. – 628 с.
5. Бедный Ю.Д. Применение генетических алгоритмов для генерации автоматов при построении модели максимального правдоподобия и в задачах управления. – СПбГУ ИТМО, 2006 [Электронный ресурс]. – Режим доступа: <http://is.ifmo.ru/papers/bednij/masters.pdf>, свободный. Яз. рус. (дата обращения 07.02.2011).
6. Яминов Б. Генетические алгоритмы. – СПбГУ ИТМО, 2005 [Электронный ресурс]. – Режим доступа: <http://rain.ifmo.ru/cat/view.php/theory/unordered/genetic-2005/>, свободный. Яз. рус. (дата обращения 07.02.2011).
7. Cormen T., Leiserson C., Rivest R., Stein C. Introduction to Algorithms. 3rd edition. – Cambridge: MIT Press, 2009.

8. Данилов В.Р. Технология генетического программирования для генерации автоматов управления системами со сложным поведением. – СПбГУ ИТМО, 2006 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/download/danilov_bachelor.pdf, свободный. Яз. рус. (дата обращения 07.02.2011).
9. Царев Ф.Н., Шалыто А.А. Применение генетического программирования для генерации автомата в задаче об «Умном муравье» // Сборник трудов IV-ой Международной научно-практической конференции. – 2007. – Т. 2. – С. 590–597.

Соколов Дмитрий Олегович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, dimoz_88@rambler.ru

УДК 004.85

ПРИМЕНЕНИЕ МАШИННОГО ОБУЧЕНИЯ ДЛЯ СОЗДАНИЯ УПРАВЛЯЮЩИХ АВТОМАТОВ НА ПРИМЕРЕ ИГРЫ

«ROBocode»

И.И. Чернявский

Рассматривается задача автоматического построения управляющих автоматов. Предлагается метод построения, основанный на применении машинного обучения, а также проводится сравнение предлагаемого метода с методом генетического программирования.

Ключевые слова: машинное обучение, управляющие автоматы, Robocode.

Введение

Построение автономных роботов-агентов является актуальной задачей. Одним из способов описания поведения таких агентов являются управляющие автоматы [1]. Построение автоматов вручную является трудоемким процессом, подверженным ошибкам. В связи с этим внимание исследователей привлечено к вопросу автоматического создания управляющих автоматов. В настоящей работе этот вопрос рассматривается на примере построения робота-танка для компьютерной игры «Robocode». При этом предлагается метод автоматического построения управляющего автомата для танка и проводится сравнение предлагаемого метода с методом генетического программирования.

Постановка задачи

Задача, решаемая в данной работе, состоит в разработке метода автоматического построения управляющих автоматов. Эта задача рассматривается на примере построения танка для игры «Robocode». Предлагаемый метод должен успешно справляться с задачей построения управляющего автомата для танка, побеждающего заданного противника из поставки игры (танки `sample.Tracker`, `sample.Fire` и `sample.Walls`).

Приведем краткое описание игры «Robocode». Игра представляет собой соревнование роботов-танков на прямоугольном поле. Танк состоит из корпуса, радара и пушки. Программно танк является классом, написанным на языке Java или языках .NET. Этот класс управляет стрельбой, движениями всех частей танка – корпуса, радара и пушки, а также занимается обработкой поступающих событий, к числу которых относятся обнаружение противника радаром, попадание снарядов в танк, поражение других танков, столкновения и т.д.

Игра состоит из последовательности сражений (раундов). В работе рассматриваются только игры с двумя сражающимися танками. В этом случае раунд продолжается до уничтожения одного танка другим. По результатам сыгранных раундов каждому танку начисляются баллы, зависящие от числа выигранных сражений, нанесенного другому танку урона и т.д. Победитель игры определяется наибольшим числом полученных баллов.

В работе [2] рассматриваются задачи «Умный муравей» и «Летающие тарелки», связанные с построением роботов-агентов, управляющих муравьем и летающей тарелкой соответственно, и показывается эффективность генетического программирования [3] в качестве метода построения управляющих автоматов.

В работе [4] исследуется задача автоматического создания управляющих автоматов для роботов-танков в игре «Robocode» при помощи генетического программирования, а в работе [5] предлагается метод двухэтапного генетического программирования для построения автомата.

Обучение с подкреплением

Подход, предлагаемый в настоящей работе, использует методы обучения с подкреплением [6] – класс методов машинного обучения, основной идеей которых является обучение агента через непосредственное взаимодействие с окружающей средой. Общая схема обучения для этого случая приведена на рис. 1.

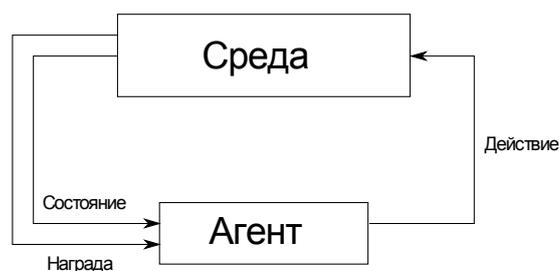


Рис. 1. Схема обучения

Обучение происходит пошагово. На каждом шаге агент «видит» состояние среды, совершает возможное из данного состояния действие и получает ответ от среды – награду. Награда – это число, являющееся оценкой действий агента в ближайшей перспективе. Заметим, что награда может быть отрицательной – быть наказанием. Цель агента – научиться выбирать свои действия так, чтобы максимизировать сумму получаемых им наград. Множества состояний и возможных действий известны заранее и не изменяются в процессе обучения. Также обучаемый агент в общем случае не знает заранее, какую награду дает среда за данное действие из данного состояния. Агент использует для обучения опыт взаимодействия со средой – последовательность из пройденных состояний, совершенных действий и полученных наград.

Рассмотрим метод обучения с подкреплением, используемый в данной работе – Q -обучение. При использовании этого метода агент пытается получить как можно более точные оценки значений Q -функции: $Q(s, a)$ – ожидание суммы последующих наград, получаемых агентом, находящимся в состоянии s и совершающим действие a . Зная значения этой функции, агент может выбирать правильные действия (из данного состояния достаточно выбрать действие, максимизирующее Q -функцию). Начальные оценки значений Q -функции полагаются нулевыми.

Пусть на шаге t агент, находясь в состоянии s_t , совершив действие a_t и получив награду r , перешел в состояние s_{t+1} . Тогда оценка $Q(s_t, a_t)$ обновляется по следующей формуле:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha (R_t - Q(s_t, a_t)),$$

где R_t – новая оценка:

$$R_t = r_{t+1} + \gamma \max_a Q(s_{t+1}, a).$$

Результатом обучения являются найденные значения Q -функции. Полученные значения можно хранить в памяти в виде таблицы. Данный способ представления не всегда удобен – число состояний может быть велико, и поэтому размер таблицы может быть слишком большим. В этом случае для представления Q -функции можно использовать нейронную сеть [7]. При таком способе представления для вычисления значения $Q(s, a)$ на входы нейронной сети подаются закодированные состояние s и действие a . Обновление оценки на каждом шаге производится путем применения алгоритма обратного распространения ошибки [7] – входами сети являются закодированные состояние s_t и действие a_t , а желаемым значением на выходе – новая оценка R_t .

Предлагаемый метод

Предлагаемый подход к построению автомата использует идею декомпозиции из работы [5]. Автомат строится в два этапа. На первом этапе строятся состояния автомата, а на втором – функция переходов автомата. В качестве методов, используемых на первом и втором этапах, предлагается использовать методы обучения с подкреплением.

На первом этапе робот обучается выполнению следующих действий:

- преследованию цели;
- уклонению («убеганию») от противника;
- наведению пушки на цель.

Данные действия соответствуют трем состояниям управляющего автомата. Состоянию A соответствует преследование цели, D – уклонение от противника, G – наведение пушки и стрельба. Находясь в каком-либо из состояний, робот выполняет соответствующее состоянию действие.

Цель второго этапа – найти функцию переходов управляющего автомата и, тем самым, получить общую стратегию игры робота, приводящую к победе над противником.

На первом этапе состояния строятся независимо друг от друга. Для построения состояний, как отмечалось выше, используется Q -обучение, результатом которого являются значения Q -функции. Эта функция может быть представлена в виде таблицы или в виде нейронной сети.

Таким образом, состояние автомата представляется либо в виде таблицы (набор всех значений Q -функции), либо в виде нейронной сети, вычисляющей Q -функцию. Нейронные сети, используемые в данной работе, имеют один скрытый слой. Нейроны этого слоя используют сигмоид-функцию

$$f(x) = \frac{1}{1 + e^{-x}}.$$

Остальные нейроны используют линейную функцию.

Сформулируем задачи обучения для построения состояний и функции переходов:

1. Обучение преследованию цели
 - состояния среды: два числа – угол между направлением движения и направлением на цель и расстояние до цели;
 - действия: сохранение направления движения или поворот влево/вправо на 10° ;
 - награда: «+4» – за приближение к цели.
2. Обучение уклонению от противника
 - состояния среды: относительное положение противника и относительное положение ближайшего препятствия (стены);
 - действия: сохранение направления движения или поворот влево/вправо на 10° ;
 - награда: «-5» – за приближение к противнику и «-10» – за столкновение со стеной.
3. Обучение наведению пушки на цель
 - состояния среды: угол между пушкой и целью;
 - действия: поворот пушки влево/вправо на 10° и на 2° или сохранение текущего направления пушки;
 - награда: «+2» – за наведение пушки на цель.
4. Обучение функции переходов
 - состояния среды: номер текущего состояния автомата и расстояние до цели;
 - действия: переход в состояние A, D, G ;
 - награда: «+5» – за победу в раунде, «-5» – за поражение (дается на последнем шаге раунда).

Результаты экспериментов

Схема эксперимента. Эксперимент проводится в два этапа. На первом этапе путем решения соответствующих задач обучения с подкреплением строятся состояния автомата. Результатом обучения является таблица или нейронная сеть. Полученный результат можно оценить численно – оценкой является средняя награда, получаемая роботом, использующим сеть, за один шаг.

Для сравнения состояния автомата строятся также с помощью генетического программирования. В данном случае особью является нейронная сеть, вычисляющая Q -функцию. Оценка особей производится путем сравнения среднего значения награды за один шаг.

Результаты, полученные с помощью обучения с подкреплением, сравниваются с результатами, полученными с применением генетического программирования.

На втором этапе эксперимента строится функция переходов автомата. Задача решается методом обучения с подкреплением, а также генетическим программированием – особью в этом случае является тройка нейронных сетей. Находясь в некотором состоянии автомата, робот использует соответствующую данному состоянию нейронную сеть для выбора следующего состояния автомата. Вычисляются выходные значения сети для трех входов, кодирующих возможное следующее состояние и расстояние до цели. Следующее состояние автомата определяется наибольшим полученным значением.

Обучение проводится путем проведения сражений против роботов-противников – `sample.Tracker`, `sample.Fire` и `sample.Walls`. Построенные автоматы сравниваются друг с другом по результатам игры с этими танками.

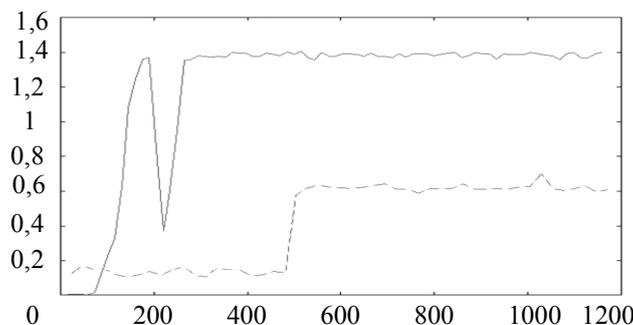


Рис. 2. Графики средней награды в задаче наведения пушки

Наведение пушки. Обучение проводилось с использованием нейронной сети. На рис. 2 сплошной и пунктирной линиями изображены графики средней награды за один шаг при использовании обучения и генетического программирования соответственно. По горизонтали на графиках отложено время, прошедшее с начала эксперимента, по вертикали – средняя награда. Робот обучился наведению пушки: средняя награда 1,35 означает, что на большинстве шагов пушка была направлена на цель. При использовании генетического программирования получено решение со средней наградой 0,7.

Преследование цели. На рис. 3 приведены результаты обучения и генетического программирования. Обучение проводилось с использованием таблиц.

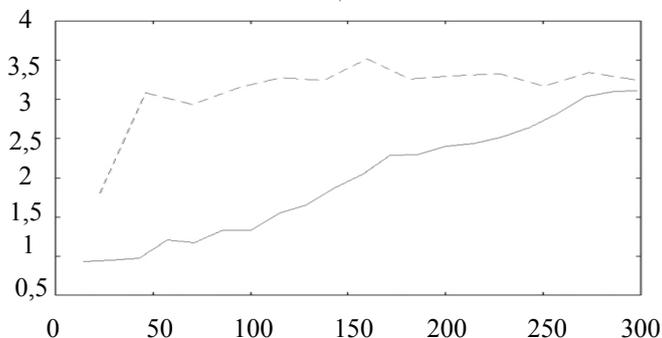


Рис. 3. Графики средней награды в задаче преследования цели

С помощью обучения была достигнута средняя награда за шаг 3,13. Метод генетического программирования быстро находит решение со средней наградой за шаг – 3,51.

Уклонение от противника. Результаты обучения и генетического программирования приведены на рис. 4. Наблюдаются сильные колебания средней награды в процессе обучения. Через 406 секунд была найдена нейронная сеть, соответствующая средней награде, равной «–1,6». Робот, использующий данную сеть, движется в сторону от противника, почти не сталкиваясь со стенами.

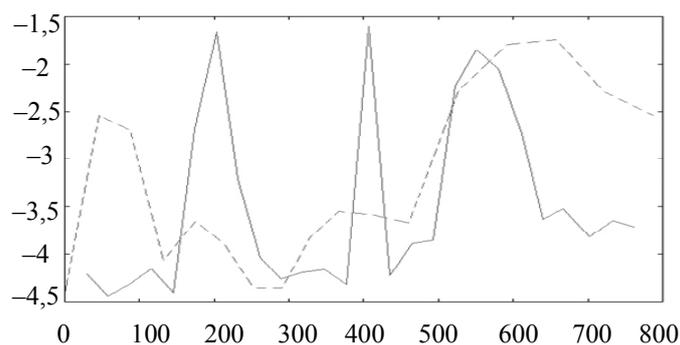


Рис. 4. График средней награды в задаче уклонения от противника

С помощью генетического программирования получено решение со значением средней награды, равным «–1,7».

Построение робота-танка. Роботы `sample.Tracker` и `sample.Fire` являются стандартными роботами поставки и реализуют простые стратегии игры. Робот `sample.Tracker` сначала пытается приблизиться к цели, а затем наводит пушку и стреляет. Робот `sample.Fire` не перемещается по полю, вращает пушку и стреляет из нее при обнаружении цели.

Сравниваемые методы (обучение с подкреплением (RL) и генетическое программирование (GP)) быстро справляются с построением роботов, побеждающих `sample.Tracker` и `sample.Fire`. С помощью генетического программирования уже на втором поколении строится робот, побеждающий танк `sample.Tracker` с отношением полученных баллов к общей сумме баллов, набранных сражающимися роботами, равным 0,7. Построенный робот наводит пушку, стреляет в цель и не перемещается по полю. Обучение с подкреплением за 500 раундов строит робота, демонстрирующего такую же стратегию игры. Результаты обучения против танков `sample.Tracker` и `sample.Fire` приведены в таблице.

Робот	RL	GP
<code>sample.Tracker</code>	8694:21471 (0,71)	9042:21253 (0,70)
<code>sample.Fire</code>	7512:22378 (0,75)	3264:18774 (0,85)

Таблица. Результаты обучения

В таблице указаны баллы, полученные роботами после 100 раундов игры, в скобках указано отношение баллов, полученных построенным роботом, к общей сумме баллов.

Робот `sample.Walls` имеет более сложное поведение – двигаясь вдоль стен, робот направляет пушку в сторону поля и стреляет в цель. С помощью генетического программирования было получено решение с отношением баллов (к общей сумме) 0,62. С помощью Q -обучения за 500 раундов был построен робот, побеждающий `sample.Walls` с отношением полученных баллов 0,71.

Заключение

В работе рассмотрена задача построения управляющего автомата для робота-агента на примере игры «Robocode». Для решения задачи предложен метод, основанный на двухэтапном построении управляющего автомата с помощью обучения с подкреплением. Предложенный метод успешно справляется с задачей построения автомата, управляющего танком в игре «Robocode».

Сравнение результатов применения Q -обучения с результатами, полученными методом генетического программирования, показывает, что на обоих этапах результаты работы методов не сильно отличаются друг от друга. При наведении пушки Q -обучение показало лучший результат, в то время как при преследовании цели лучший результат показывает метод генетического программирования. В задаче построения автомата против робота `sample.Fire` лучший результат показало генетическое программирование, а при построении автомата против робота `sample.Walls` – Q -обучение.

Полученные данные позволяют утверждать, что методы обучения с подкреплением применимы к построению управляющих автоматов для роботов-танков и не уступают в эффективности генетическому программированию.

Литература

1. Поликарпова Н.И., Шалыто А.А. Автоматное программирование. – СПб: Питер, 2009 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/books/_book.pdf, своб.
2. Царев Ф.Н. Разработка метода совместного применения генетического программирования и конечных автоматов. Бакалаврская работа. – СПбГУ ИТМО, 2007 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/papers/_2010_03_03_tsarev.pdf, своб.
3. Mitchell M. An Introduction to Genetic Algorithms. – The MIT Press, 1996.
4. Бедный Ю.Д. Применение генетических алгоритмов для генерации автоматов при построении модели максимального правдоподобия и в задачах управления. Магистерская диссертация. – СПбГУ ИТМО, 2008 [Электронный ресурс]. – Режим доступа: <http://is.ifmo.ru/papers/bednij/>, своб.
5. Соколов Д.О. Применение двухэтапного генетического программирования для построения автомата, управляющего моделью танка в игре «Robocode». Бакалаврская работа. – СПбГУ ИТМО, 2009.
6. Sutton R.S., Barto A.G. Reinforcement Learning. An Introduction. – The MIT Press, 1998.
7. Хайкин С. Нейронные сети. Полный курс. – М.: Вильямс, 2006.

Чернявский Илья Игоревич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, chernyavsky@rain.ifmo.ru

УДК 004.4'242

АВТОМАТИЧЕСКИЙ СИНТЕЗ СИСТЕМЫ УПРАВЛЕНИЯ МОБИЛЬНЫМ РОБОТОМ ДЛЯ РЕШЕНИЯ ЗАДАЧИ «КЕГЕЛЬРИНГ»

С.А. Алексеев, А.И. Калининченко, В.О. Клебан, А.А. Шалыто

Приводится пример автоматического синтеза системы управления мобильным роботом для решения задачи «Кегельринг». Автоматический синтез системы проводится с использованием генетического алгоритма, при помощи которого определяется структура управляющего автомата.

Ключевые слова: автоматное программирование, генетические алгоритмы, автоматический синтез систем управления.

Введение

Для построения систем управления мобильными роботами целесообразно использовать технологию автоматного программирования [1–3], в которой, в частности, предлагается строить программу как систему автоматизированных объектов управления, в которых управляющая программа представляет собой систему автоматов, взаимодействующих между собой за счет вложенности и вызываемости. Использование автоматного подхода при создании подобных систем обладает рядом достоинств, таких как возможность повышения уровня автоматизации процесса верификации, документируемость, упрощение внесения изменений и т.д.

На практике встречаются задачи, для которых известно, что они могут быть решены при помощи конечных автоматов, но эвристически построить для них автомат чрезвычайно сложно [4]. Для решения подобных задач могут быть использованы методы автоматического синтеза программ, одним из которых является генетическое программирование.

Эффективность применения генетического программирования для синтеза автоматов продемонстрирована в работах [5–8], но, к сожалению, ни в одной из них не проверялась работа синтезированных систем управления на реальных объектах.

В данной работе приводится пример автоматического синтеза системы управления роботом для задачи «Кегельринг» и его проверка на реальном мобильном роботе.

Постановка задачи

Цель робота в задаче «Кегельринг» состоит в выталкивании за пределы ринга расположенные в нем кегли за минимально возможное время. Ринг представляет собой площадку круглой формы диаметром один метр, которая ограничена темной контрастной полосой. При выполнении задания робот не должен выходить за пределы ринга. Порядок расстановки кеглей определен следующим образом:

- перед началом состязания на ринге расставляют восемь кеглей;
- робот устанавливается в центр ринга;
- методом жеребьевки из ринга убирают четыре кегли, что позволяет внести в задание элемент случайности.

Целью данной работы является автоматический синтез системы управления мобильным роботом для решения данной задачи.

Устройство робота

Робот представляет собой трехколесное шасси. В качестве движителя используются два электродвигателя постоянного тока, которые образуют дифференциальный привод, позволяющий роботу осуществлять «танковый» разворот на месте (рис. 1). На роботе установлены два типа датчиков: инфракрасный дальномер, предназначенный для детектирования кеглей, и датчик линии, предназначенный для сигнализации о том, что робот пересек ограничительную линию.

Выходными воздействиями для робота являются: вращение по часовой стрелке (z_0); вращение против часовой стрелки (z_1); движение вперед (z_2); движение назад (z_3); останов (z_4).

Входными воздействиями являются: наличие/отсутствие линии под роботом (x_0); перед роботом нет кегли (x_1); перед роботом есть кегля на большом расстоянии (x_2); перед роботом есть кегля на среднем расстоянии (x_3); перед роботом есть кегля на малом расстоянии (x_4); кегля на очень малом расстоянии (робот толкает кеглю) (x_5).

В процессе автоматического синтеза системы управления используются два робота, соответствующие по характеристикам друг другу: реальный и виртуальный.

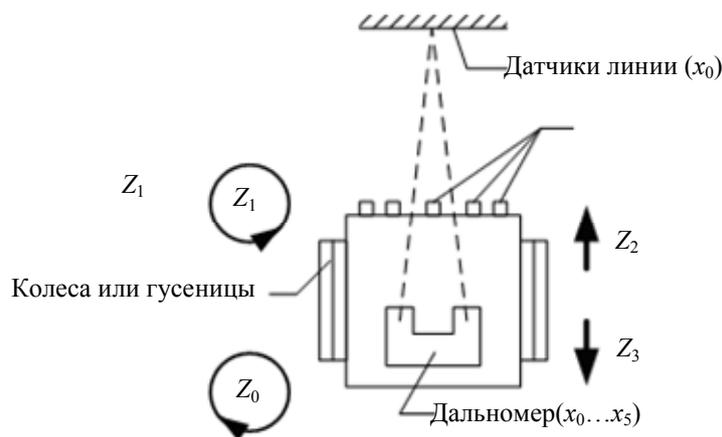


Рис. 1. Устройство мобильного робота

Схема работы генетического алгоритма

Для осуществления автоматического синтеза системы управления роботом разработан программно-аппаратный комплекс (рис. 2), который включает в себя симулятор, предназначенный для моделиро-

вания поведения реального робота, и приложение, реализующее генетический алгоритм (в дальнейшем просто генетический алгоритм).

Симулятор работает следующим образом: для каждой поданной ему на вход особи вычисляется функция приспособленности, которая используется при отборе особей в генетическом алгоритме. При работе симулятора желательно обеспечить наиболее точное соответствие виртуальной и реальной сред. Это необходимо для того чтобы синтезированная при помощи симулятора система управления оставалась работоспособной в реальной среде.

В качестве симулятора в работе использована программная среда Webots. Данная среда позволяет создавать виртуальные модели роботов, окружение для них, а также программировать физическое поведение и выполнять моделирование.

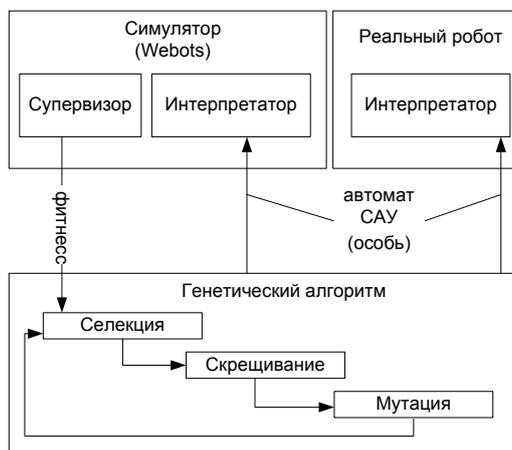


Рис. 2. Структурная схема разработанного комплекса

Среда эмуляции Webots позволяет оснастить виртуального робота моделями двигателей, сервоприводов и датчиков. Специально для решения поставленной задачи спроектирована виртуальная модель реального робота, а также виртуальный ринг с кеглями.

Виртуальная модель робота оснащена двумя двигателями, дальномером и датчиком линии, эквивалентными тем, которые установлены на реальном роботе.

Для обеспечения работы виртуальной модели на языке программирования Java был разработан интерпретатор автоматов управления (построенных при помощи генетического алгоритма).

Кроме того, был разработан контроллер-супервизор, который отслеживает перемещения модели робота и кеглей, и на основе этих данных вычисляет значение функции приспособленности.

В процессе работы генетический алгоритм обеспечивает процесс эволюции особей, которые в данном случае представлены в виде управляющих автоматов. Вначале алгоритм генерирует случайную начальную популяцию (набор случайных автоматов). Особи этой популяции подаются на вход симулятора Webots (рис. 2), который рассчитывает значение функции приспособленности для каждой из особей и предоставляет эти данные генетическому алгоритму. На следующем шаге генетический алгоритм осуществляет селекцию, скрещивание и мутацию в соответствии с полученными значениями функции приспособленности.

Любую из полученных в популяции особей, при необходимости, можно передать на вход интерпретатору реального робота для проверки функциональности особи в реальных условиях (рис. 3).

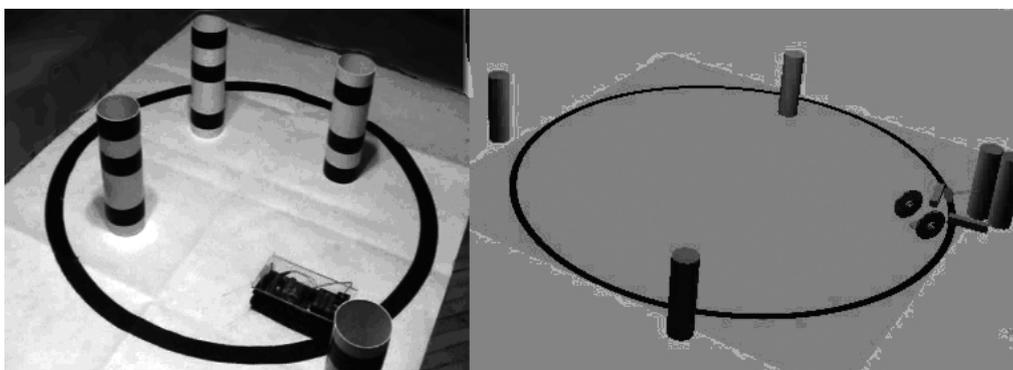


Рис. 3. Реальный робот и его компьютерная модель во время тестирования

Способ кодирования хромосом

Для применения генетического алгоритма к решению задач автоматического синтеза систем управления мобильными роботами необходима модель хромосомы. В рамках рассматриваемой задачи хромосома представляет собой закодированный специальным образом конечный автомат Мили, при помощи которого осуществляется управление роботом.

Опишем способ кодирования для конечного автомата Мили из N состояний, в каждом из которых возможно L входных воздействий. Первый байт строки – номер начального состояния (его значение меньше N). Каждое состояние кодируется следующим образом: первый байт – номер выходного воздействия (ехать вперед/назад, поворачивать), выдаваемого на исполнительные механизмы реального, либо виртуального робота, затем L байт – номера состояний, в которые автомат переходит в зависимости от входных воздействий. Таким образом, в начале байт-строки идет один байт – номер начального состояния, затем $N \times L$ байт – описание состояний.

Полученная модель хромосомы является одной из самых простых, что позволяет применять простейшие операторы скрещивания и мутации.

Операторы скрещивания и мутации

Как отмечено выше, генетический алгоритм осуществляет процесс эволюции особей, похожий на биологическую эволюцию. В ходе этого процесса применяются два основных инструмента – селекция и скрещивание. Для скрещивания особей применяется специальный оператор скрещивания. Его задача – создание особей нового поколения на основе особей предыдущего поколения.

В качестве оператора скрещивания используется одноточечный кроссовер. Генерируется случайное натуральное число C , величина которого меньше длины хромосомы. Затем обе родительские хромосомы делятся на две части, первая из которых содержит первые C генов, а вторая – оставшиеся. Две дочерние особи получаются путем составления нового генотипа на основе частей генотипа обоих родителей.

Рассмотрим пример применения операции одноточечного кроссовера к хромосомам $A(0\ 110\ 201)$ и $B(1\ 201\ 101)$. Длина каждой из хромосом равна семи. Допустим, что число C равно четырем. Тогда первая дочерняя хромосома будет состоять из четырех первых байт хромосомы A и трех последних байт хромосомы B . Вторая дочерняя хромосома будет состоять из четырех первых байт хромосомы B и трех последних байт хромосомы A . Части, которыми обмениваются конечные автоматы, выделены пунктирными линиями.

Для внесения многообразия в популяцию применяется оператор мутации, который осуществляет случайные изменения в генотипе случайных особей. При синтезе рассматриваемой системы управления применяется оператор мутации, который изменяет один случайно выбранный байт в хромосомах пяти случайных особей. Выбор именно пяти особей обусловлен тем, что с такими параметрами алгоритм показывал лучшую сходимость.

Функция приспособленности

Для эффективной работы генетического алгоритма необходимо выбрать соответствующую функцию приспособленности – отображение из множества хромосом в множество численных значений их приспособленности. Для рассматриваемой задачи эта функция зависит от того, покидал ли робот ринг, а также следующих параметров:

- число кеглей, которые в какой-либо момент времени находились за пределами ринга;
- число кеглей, которые не покинули ринг;
- расстояние, пройденное роботом;
- время, за которое робот решил задачу.

Особенностью функции приспособленности, использованной в данной задаче, является то, что способ ее вычисления зависит от текущего этапа развития робота. Поясним это. В первом поколении, когда хромосомы строятся случайным образом, робот редко может вытолкнуть хотя бы одну кеглю. Таким образом, если оставить для расчета функции приспособленности только такие параметры, как время, число кеглей и условие, что робот не должен покидать ринг, то большее значение вычисляемой функции часто будут иметь роботы, которые не предпринимают никаких действий и, как следствие, не покидают ринг.

Получается, что на первом этапе выращивания целесообразно использовать функцию приспособленности, включающую в себя «бонус» за пройденное расстояние. Тогда получают преимущество роботы, предпринимающие активные действия, например, использующие датчик линии, ограничивающей ринг, или перемещающиеся внутри ринга. Для вычисления функции приспособленности была использована процедура, приведенная в листинге.

После того, как хромосомы эволюционируют (рис. 4) и роботы научатся убирать кегли с ринга, целесообразно присваивать большее значение функции приспособленности тем роботам, которые способны решить изначально поставленную задачу: не выезжать за пределы ринга и не возвращать кегли на ринг (такое случается, если кегля «цепляется» за робота). Фактически происходит поэтапное обучение робота.

Листинг. Вычисление функции приспособленности

```

fitness = 0;          //Значение функции приспособленности
current_time = 0;    //Текущее время модели
max_time = const;   //Максимальное время выполнения задания
distance = 0;        //Путь, проделанный роботом
while current_time < max_time do
  if кегли_убраны and робот_не_покидал_ринг then
    fitness = max_time - current_time;
    return fitness;
  else
    fitness = fitness + distance;
    if кегля_покинула_ринг then
      fitness = fitness + 1;
    if кегля_вернулась_в_ринг then
      fitness = fitness - 0.5;
return fitness;

```



Рис. 4. Динамика роста функции приспособленности

Селекция

В качестве алгоритма селекции был применен отбор усечением. Эта стратегия использует отсортированное поколение, из которого выбирается наиболее приспособленная половина особей. Затем среди отобранных особей случайным образом выбираются пары, которые участвуют в скрещивании. Кроме того, применяется элитизм, при котором несколько наиболее приспособленных особей попадают в новое поколение без изменений.

Без использования этого приема поколения будут вырождаться, так как при скрещивании двух хорошо приспособленных особей имеется большая вероятность получить полностью неработоспособного потомка.

Калибровка модели и реального робота

Одной из серьезных проблем при использовании генетических алгоритмов является несоответствие между моделью задачи, используемой при синтезе системы управления, и реальным миром.

Автомат, отлично решающий поставленную задачу в эмуляторе, может полностью не справляться с ней в реальной среде. При разработке модели даже для такой простой задачи, как «Кегельринг», невозможно учесть все внешние факторы, влияющие на работу алгоритма: погрешности датчиков, особенности работы электродвигателей, материалы предметов, покрытие ринга и многое другое. Робот, выращенный на одной модели, должен демонстрировать работоспособность и на другой, немного отличающейся от исходной, это означает, что полученная стратегия управления должна быть как можно более робастной.

Рассмотрим возможные методы решения данной проблемы.

- Обучение робота на различных моделях. Недостатком данного метода является увеличение пространства поиска, что негативно сказывается на времени работы алгоритма.
- Обучение робота с зашумленными входами. Данный метод не увеличивает пространство поиска, но при этом адаптирует робота к выходу в реальный мир.
- Изменение функции приспособленности в сторону меньшей зависимости от параметров среды.

Наряду с перечисленными методами авторами предлагается использовать метод, суть которого заключается в изменении типа выходных воздействий синтезируемой системы управления. Поясним на примере.

Сначала при попытках обучить систему управления использовалось выходное воздействие «поворот», которое предполагало поворот робота на угол в 10° . В результате полученные автоматы использовали данное свойство, выстраивая выходные воздействия в цепочку и получая при этом поворот на необходимый угол. К сожалению, после переноса полученного автомата на реального робота оказалось, что система управления неработоспособна, так как осуществляемый по таймеру поворот происходит на разный угол в зависимости от условий. Избежать данного эффекта удалось, изменив тип выхода на «постоянное вращение». Вместо поворота на определенный угол робот начинал вращаться в заданном направлении. Это позволило добиться большего сходства между реальным роботом и его моделью.

Изменение типа выхода также положительно сказалось и на сходимости алгоритма.

Заключение

В данной работе приводится пример автоматического синтеза системы управления мобильным роботом для решения задачи «Кегельринг». Одним из условий успешного решения задачи служила возможность переноса синтезированной системы управления на реального робота без внесения каких-либо изменений.

Поставленные цели были достигнуты, и реальный робот выполнил поставленную задачу [8].

В ходе выполнения работы обнаружен прием, названный «изменение типа выхода», позволяющий добиться более устойчивой работы синтезированного автомата без увеличения пространства поиска.

Исследование выполнено по Федеральной целевой программе «Научные и научно-педагогические кадры инновационной России на 2009–2013 годы» в рамках государственного контракта П2236 от 11 ноября 2009 года.

Литература

1. Шальто А.А. Технология автоматного программирования // Труды Всероссийской научной конференции «Методы и средства обработки информации». – М.: МГУ, 2003 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/works/tech_aut_prog/, своб.
2. Клебан В.О., Шальто А.А. Использование автоматного программирования для построения многоуровневых систем управления мобильными роботами // Сборник тезисов 19 Всероссийской научно-технической конференции «Экстремальная робототехника». – СПб: ЦНИИ РТК, 2008. – С. 85–87.
3. Клебан В.О., Шальто А.А., Парфенов В.Г. Построение системы автоматического управления мобильным роботом на основе автоматного подхода // Научно-технический вестник СПбГУ ИТМО. – 2008. – № 53.
4. Царев Ф.Н., Шальто А.А. О построении автоматов с минимальным числом состояний для задачи об «Умном муравье» // Сборник докладов X международной конференции по мягким вычислениям и измерениям. – СПбГЭТУ «ЛЭТИ», 2007. – Т. 2. – С. 88–91 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/download/ant_ga_min_number_of_state.pdf, своб.
5. Данилов В.Р. Технология генетического программирования для генерации автоматов управления системами со сложным поведением. Бакалаврская работа. – СПбГУ ИТМО, 2007 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/papers/danilov_bachelor/, своб.
6. Поликарпова Н.И., Точилин В.Н. Применение генетического программирования для реализации систем со сложным поведением // Научно-технический вестник СПбГУ ИТМО. – 2007. – № 39.
7. Chambers L. Practical Handbook of Genetic Algorithms. Complex Coding Systems. – CRC Press, 1999. – V. III.
8. Видеоматериалы [Электронный ресурс]. – Режим доступа: <http://blog.savethebest.ru/?p=747>, своб

Алексеев Сергей Андреевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, alex.itmo@gmail.com

Калиниченко Александр Игоревич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, ittrium@gmail.com

Клебан Виталий Олегович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, vk.developer@gmail.com

Шальто Анатолий Абрамович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, зав. кафедрой, shalyto@mail.ifmo.ru

УДК 004.272.26

**АНАЛИЗ ЭФФЕКТИВНОСТИ ИСПОЛЬЗОВАНИЯ GPU
ДЛЯ АВТОМАТИЧЕСКОГО СИНТЕЗА СИСТЕМЫ УПРАВЛЕНИЯ
МОБИЛЬНЫМ РОБОТОМ****А.А. Сергеев, В.О. Клебан, А.А. Шалыто**

Исследуется эффективность использования графических сопроцессоров (GPU) для синтеза систем автоматического управления (САУ) мобильным роботом. В ходе работы учитываются и анализируются их особенности применительно к данной проблеме. В качестве модельной задачи для оценки эффективности синтеза выбрана параллельная реализация генетических алгоритмов на примере синтеза САУ роботом для соревнований «Сумо роботов».

Ключевые слова: производительность, GPU, генетический алгоритм, мобильный робот, синтез системы управления, конечный автомат, нейрон.

Введение

Синтез САУ, например, мобильным роботом может быть выполнен с помощью генетических алгоритмов (ГА), которые требуют значительного объема вычислений и являются в высокой степени параллельными. Поэтому актуально использование GPU для решения данной задачи. ГА в настоящее время уже реализованы на GPU, и они позволили получить существенное ускорение по сравнению с процессорной реализацией, однако применительно к синтезу САУ мобильных роботов такие исследования не выполнялись.

Постановка задачи

Цель работы – исследовать эффективность реализации синтеза системы управления мобильным роботом с использованием GPU. В качестве критерия эффективности рассматривается время, затрачиваемое на расчет фиксированной задачи при помощи GPU. Для оценки возможностей GPU рассматриваются особенности вычисления на ней и приводятся теоретические оценки.

В качестве эксперимента исследуется эффективность использования GPU для ГА на примере задачи «Сумо роботов». Оценка может быть получена путем анализа количества выращенных поколений и рассчитанных особей на машине с использованием GPU и без. Разрабатываются модели робота, его системы автоматического управления, выясняются особенности реализации ГА. Для проведения эксперимента и получения его результатов создается визуальная среда. Таким образом, в задачу входит подтверждение эффективности GPU для синтеза системы управления роботом на практике.

Сумо роботов

«Сумо роботов» представляет собой антагонистическую игру. Участие в ней принимают два робота. Они устанавливаются на плоскую круглую арену в выбранные хозяевами позиции и начинают движение по сигналу. Целью игры является выталкивание противника за пределы арены. Продолжительность поединка, как правило, ограничена 30 с. В начале игры роботы должны быть установлены за пределами ограничительных линий.

Традиционно робот имеет два колеса, скользящий шарик в качестве третьей точки опоры, индикатор линии и дальномер, например, инфракрасную пушку. Передвигается такой робот за счет разности хода между колесами.

Обзор

Исследований эффективности GPU для задач синтеза САУ мобильными роботами обнаружено не было. Существуют работы, например [1], посвященные обучению системы управления роботом с использованием ГА, но в них не применяются вычисления на GPU. Также известны работы [2–3], в которых выполняется оценка роста производительности при использовании GPU для генетических алгоритмов.

Архитектура комплекса

Тестовая машина имеет следующую конфигурацию: Intel Core i3 2.93GHz, 2 GB RAM, nVidia GeForce gtx 275 1792 MB. Для тестирования используется также видеокарта gt 240 512 MB. Пиковая производительность gtx 275 по данным nVidia составляет 1070 GFLOP/s. Для i3 это значение достигает порядка 40 GFLOP/s.

На самом деле, быстродействие всегда заметно ниже пикового, поскольку необходимо осуществлять доступ к памяти, выполнять медленные операции, синхронизировать потоки. Реально достижимая производительность на процессоре на большинстве параллельных звенья, использующих полную мощ-

ность видеокарты, составляет около 70% от максимальной. Для видеокарты это значение всегда меньше, около 50–60%. Если учесть эти оценки, то получим, что на данном стенде можно достичь ускорение порядка 20 раз, при этом максимальное ускорение оценим примерно как 27 раз (отношение пиковых производительностей).

Физическая модель робота

Физическая модель робота состоит из тела робота и нескольких колес. Тело представляет из себя цилиндр с основанием полигона. Для передвижения по поверхности арены роботы используют колеса, приводимые в движение двигателями. Таким образом, момент колеса ограничен по модулю. Колеса и тело робота взвешены и имеют ненулевой момент инерции.

Робот использует для идентификации себя в пространстве воображаемую камеру, а для наблюдения ему доступен угол обзора перед собой. Также робот снабжен датчиком цвета точки арены под собой. Это позволит ему немедленно среагировать и изменить поведение, находясь на краю арены.

Расчет физической модели

Импульс и момент импульса робота изменяются под воздействием сил трения колес о поверхность арены. Сила трения, действующая со стороны опоры на колесо, рассчитывается как

$$\|\mathbf{F}_\parallel\| = \mu \cdot \|\mathbf{N}\| \cdot \mathbf{f}((\boldsymbol{\omega} \times \mathbf{r}) - \mathbf{v}_\parallel).$$

Здесь \mathbf{F} – сила трения на колесо, \mathbf{v} – скорость колеса относительно поверхности арены, $\boldsymbol{\omega}$ – угловая скорость колеса, \mathbf{N} – сила реакции опоры (рисунок). Сила трения, таким образом, определяется относительной скоростью двух поверхностей и реакцией опоры. Скорость колеса рассчитывается из скорости центра масс робота и его угловой скорости как $\mathbf{v} = \mathbf{v}_{cm} + \boldsymbol{\omega} \times \mathbf{r}_{cm}$. Ускорение робота рассчитывается из

второго закона Ньютона, а угловое ускорение – из уравнения моментов: $\mathbf{F} = m\mathbf{a}$; $\mathbf{I}\beta = \sum_{i=1}^n \mathbf{M}_i$.

В случае столкновения на робота действуют добавочные силы. В последнем случае важно делать поправки для соблюдения закона сохранения энергии.

При малом проскальзывании по поверхности арены для двухколесного робота поворот происходит так, как изображено на рисунке. За счет разностей моментов сил создается разность угловых скоростей, обеспечивающая разность скоростей левого и правого борта робота. Тогда робот поворачивает вокруг обозначенной на рисунке точки O .

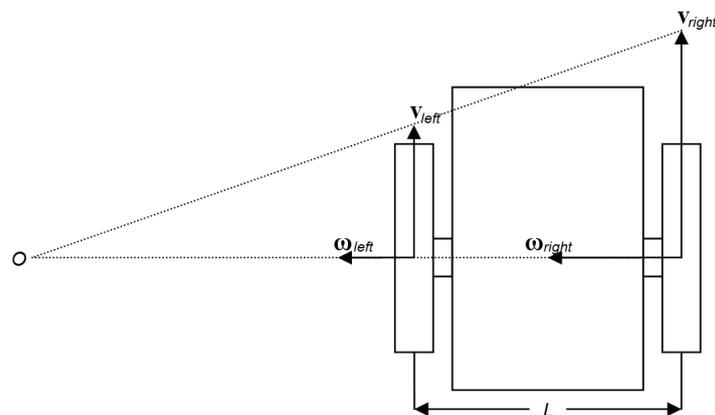


Рисунок. Поворот двухколесного робота

Используемая модель САУ

Единственное, что робот может использовать для изменения своего положение в пространстве, – это моменты сил, прикладываемые к колесам. Для реализации такого поведения необходимо на каждом шаге вычислять эти моменты. При выборе модели рассматривались следующие критерии для выбора модели системы управления:

1. Поведение робота должно в большинстве случаев непрерывно зависеть от входных параметров;
2. Робот избирает стратегию поведения в зависимости от действий противника;
3. Система управления должна иметь достаточно простую реализацию для обеспечения высокой скорости расчета действий и удобной для применения ГА.

Нейрон описывается линейной функцией от входных параметров:

$$output = f\left(\sum_{i=0}^{n-1} input[i] \cdot weight[i]\right).$$

Здесь f – передаточная функция. В частном случае положим ее тождественным оператором. Такой нейрон также называют звездой Гроссберга. Таким образом, вход нейрона есть фактически массив вещественных чисел, а выход – одно вещественное число.

Перцептрон есть модификация нейрона, реализующая пороговую функцию:

$$output = f\left(\sum_{i=0}^{n-1} input[i] \cdot weight[i]\right), \text{ где } f(x) = \begin{cases} 0, & x \leq threshold, \\ 1, & x > threshold. \end{cases}$$

Помимо весов, перцептрон опционально хранит еще и пороговое значение. Выходом его является уже целое число – нуль или единица.

Традиционно нейроны и их модификации применяются для построения нейронных сетей, где выход одного нейрона подается на вход другому. В данной работе использована несколько другая схема, основанная на передаче управления между нейронами при помощи конечного автомата.

Чтобы сделать поведение робота достаточно естественным и удовлетворить приведенным требованиям, была разработана следующая модель:

1. Моменты колес рассчитываются из входных параметров как выходы нейронов;
2. Для определения последовательности действий используется конечный автомат.

Каждое из состояний автомата имеет свои нейроны для расчета моментов, по одному нейрону на колесо. На вход нейронам подаются входные параметры робота, представляющие из себя вектор чисел. Таковыми являются, например, расстояние до противника, данные детектора близости линии. В зависимости от наличия противника в поле зрения этот вектор имеет различную длину и подается на вход нейронам. Приведенная модель автомата является автоматом Мура [2], в котором значение функции выхода есть моменты колес.

САУ робота описывается автоматом с 32-мя состояниями. Состояние имеет 8 пар нейронов (максимальное число колес) и 3 пары перцептронов. Первый нейрон или перцептрон в паре имеет длину входа в 5 параметров, второй – в 9. В зависимости от видимости противника активны первый или второй из них.

Для перехода в новое состояние происходит расчет значений перцептронов с такими же входными параметрами. Номер нового состояния формируется исходя из выходов перцептронов и индикатора наличия противника в поле видимости и индикатора линии.

Визуализация

Для тестирования и оценки результатов была разработана визуальная среда Sumo Robot Simulation. Это приложение написано на языке C++ с использованием для визуализации OpenGL [4], для реализации GUI – библиотеки Qt [5], для расчетов на GPU – динамическую библиотеку на CUDA [6].

Окно просмотра статистики роста позволяет получать всю необходимую информацию о росте популяции: просматривать график фитнес-функции и сохранять его, оценивать скорость генерации новых поколений, просматривать особей в таблице, сортировать их и сохранять в файл.

Используемая модель ГА

В данной работе будем использовать островную модель ГА. Выбранная реализация ГА использует также конкурентный элитизм. Для отбора особей применяется метод рулетки. В отборе на равных основаниях участвуют родители и их потомки. Хромосома представляет собой строку вещественных чисел одинарной точности, являющуюся последовательностью весов всех используемых нейронов и перцептронов.

Оценка приспособленности рассчитывается на основе результатов его поединков. Для этого создается набор из неизменных тестирующих особей, с каждой из которых происходит поединок, и после этого суммируются результаты. Всего в данной реализации предусмотрено 100 фиксированных особей для составления подобной оценки.

Особенности вычислений на GPU

Время доступа к регистрам и разделяемой памяти составляет четыре тактовых цикла против 400–600 для доступа к глобальной памяти [7]. На практике использование разделяемой памяти вместо глобальной может обеспечивать ускорение около 10–20 раз [8].

Код, выполняемый на устройстве, не имеет возможности напрямую выделять память на видеокарте или в оперативной памяти. Конфигурация робота и набор переменных, определяющих его движение, занимают более 10 Кбайт и достаточно велики, для того чтобы не рассматривать идею разместить их в разделяемой памяти. В таком случае задействуем как можно больше потоков.

Для расчета выходов нейронов, суммы моментов и суммы сил при движении будем использовать параллельную редукцию (reduction) [9].

Реализация ГА

Применительно к данной задаче наиболее требовательной к объему вычислений частью ГА является расчет фитнес-функции. Поэтому именно ее следует выполнять на GPU и параллельно генерировать новых особей, пользуясь возможностью асинхронного запуска ядер на GPU.

Параллельно на видеокарте происходит симуляция битв, загруженных в буфер. Каждая битва рассчитывается на фиксированное число шагов. Генерация новых экземпляров происходит на процессоре в результате скрещиваний и мутаций.

Выполнение на CPU теоретически возможно путем компилирования кода, написанного на CUDA, в режиме эмуляции. Однако использование эмуляции в настоящее время не рекомендуется (deprecated). Поэтому аналогичный код переписан для выполнения на процессоре.

Теоретическая оценка прироста производительности

Для параллельно выполняемой задачи справедлив закон Амдала: максимально возможное ускорение от параллельного выполнения равняется (для N потоков параллельно)

$$S = \frac{1}{(1-P) + \frac{P}{N}}$$

Оценка снизу распараллеливаемого объема вычислений P дает значение 0,99. Считая производительность видеокарты средней для математических задач, получим $S \approx 18,4$ (взято $N=20$ – средний коэффициент ускорения видеокарты относительно процессора). Значит, можно ожидать на порядок большую производительность при использовании GPU.

Результаты

Сравнивая время достижения максимальных значений фитнес-функции, полученное программно, можно судить о приросте производительности для CPU и разных моделей GPU. В табл. 1 приведены вычислительные мощности используемых устройств, цены приобретения и средние выигрыши в производительности.

	GFLOP/s	Цена, руб	Выигрыш
i3 2.93 GHz	40	4000	1,00
gt 240	330	3000	5,5
gt 275	1070	8000	13,5

Таблица 1. Сравнение производительностей и ценовых категорий устройств

	GPU	CPU	Средний прирост	Конфигурация
Простой ГА	4,08	81,88	20,1	AMD Athlon 2500+, nVidia GeForce 6800GT
ГА для оптимизации	0,001–0,1	0,25–0,26	30,0	Intel Core i7 920 3.2GHz, nVidia gtx 8800
Муравьиный алгоритм	1,06–0,85	22,49	6,0	Core 2 Quad Q9550, nVidia gtx 280

Таблица 2. Сравнение приростов производительностей для различных алгоритмов

Результат 20,1 (табл. 2) был получен при реализации на GPU простейшего ГА для фитнес-функции – несложной функции вещественных переменных. Здесь вычисление фитнес-функции имеет очевидные преимущества в скорости перед центральным процессором. Операции поиска лучших индивидуумов, кроссовер и мутация реализованы параллельно на GPU, что не представляет сложности для данной задачи. Здесь возможно эффективное использование разделяемой памяти (shared memory) в силу малого объема данных для отдельной подзадачи.

Результат 6,0 соответствует ускорению на муравьином алгоритме. Эта задача с высоким уровнем параллельности, тем не менее, не показала высокой степени ускорения.

Работа по использованию GPU для решения задач многомерной оптимизации (второй ряд) показывает высокий разброс результатов на различных задачах: выигрыш в скорости составил от 2,5 раза до тысяч раз. Это означает случайный разброс значений, поскольку показатели слишком высоки, а ускоре-

ние существенно превосходит соотношение производительностей устройств. В то же время средние показатели ускорения составляют 20–30 раз.

Для прироста системы управления роботом (табл. 1) был получен результат 13,5. Это несколько медленней лучших результатов для задач оптимизации и простого ГА. В то же время теоретический результат вполне подтверждается экспериментом. Потеря производительности наблюдается по следующим причинам:

1. Активное использование глобальной памяти. САУ и конфигурация роботов слишком велики для разделяемой памяти;
2. Часть потоков с подзадачами завершается довольно быстро, что позволяет процессору существенно экономить время.

Заключение

В работе проанализирована эффективность GPU для синтеза системы автоматического управления мобильным роботом. Рассмотрены особенности, характерные для реализации подобных задач с использованием GPU. Поставлен эксперимент на примере синтеза САУ для робота Сумо. Получена оценка прироста производительности по сравнению с решением без использования GPU. Согласно результатам анализа и поставленного эксперимента, использование GPU высокоэффективно для поставленной задачи.

В качестве главного недостатка GPU при решении задач синтеза САУ выделяется малый объем разделяемой памяти. Достоинствами являются возможность асинхронного запуска ядра и высокая вычислительная мощность GPU относительно CPU за счет большого числа параллельно рассчитываемых нитей.

Исследование выполнено по Федеральной целевой программе «Научные и научно педагогические кадры инновационной России на 2009–2013 годы» в рамках государственного контракта П2236 от 11 ноября 2009 года.

Литература

1. Jiming Liu, Shiwu Zhang. Multi-phase sumo maneuver learning // *Robotica* 22. – 2004. – № 1. – P. 61–75.
2. Qizhi Yu, Chongcheng Chen, Zhigeng Pan. Parallel Genetic Algorithms on Programmable Graphics Hardware. [Электронный ресурс]. – Режим доступа: <http://www.cad.zju.edu.cn/home/yqz/projects/gagpu/icnc05.pdf>, свободный. Яз. англ. (дата обращения 19.02.2011).
3. Pospichal P., Jaros J. GPU-based Acceleration of the Genetic Algorithm. [Электронный ресурс]. – Режим доступа: <http://www.gpgpgpu.com/gecco2009/7.pdf>, свободный. Яз. англ. (дата обращения 19.02.2011).
4. Wright R., Lipchak B. OpenGL superbible // SAMS publishing, 2005.
5. Земсков Ю.В. Qt4 на примерах. – СПб: БХВ Петербург, 2008.
6. Боресков А.В., Харламов А.А. Основы работы с технологией CUDA. – М: ДМК пресс. – 2010.
7. NVIDIA Corporation. Fermi Compatibility Guide for CUDA Applications. 2010. [Электронный ресурс]. – Режим доступа: http://developer.download.nvidia.com/compute/cuda/3_0/docs/NVIDIA_FermiCompatibilityGuide.pdf, свободный. Яз. англ. (дата обращения 19.02.2011).
8. NVIDIA Corporation. NVIDIA CUDA Programming Guide. 2010. [Электронный ресурс]. – Режим доступа: http://developer.download.nvidia.com/compute/cuda/2_3/toolkit/docs/NVIDIA_CUDA_ProgrammingGuide_2.3.pdf, свободный. Яз. англ. (дата обращения 19.02.2011).
9. NVIDIA Corporation. NVIDIA CUDA Best Practices Guide. 2010. [Электронный ресурс]. – Режим доступа: http://developer.download.nvidia.com/compute/cuda/2_3/toolkit/docs/NVIDIA_CUDA_BestPracticesGuide_2.3.pdf, свободный. Яз. англ. (дата обращения 19.02.2011).

Сергеев Антон Алексеевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, anton.a.sergeev@gmail.com

Клебан Виталий Олегович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, vk.developer@gmail.com

Шальто Анатолий Абрамович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, зав. кафедрой, shalyto@mail.ifmo.ru

УДК 004.4'236

ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ДЛЯ ИССЛЕДОВАНИЯ АВТОМАТНОГО УПРАВЛЕНИЯ МОБИЛЬНЫМИ РОБОТАМИ

С.А. Алексеев, В.О. Клебан, А.А. Шалыто

Управление роботами является задачей управления системами со сложным поведением, в настоящее время известно несколько подходов к решению таких задач, одним из которых является автоматное программирование. Рассматривается комплекс, предназначенный для проектирования и исследования автоматных программ управления мобильными роботами.

Ключевые слова: автоматное программирование, мобильные роботы, визуальное средство проектирования автоматных программ.

Введение

Роботы – это физические агенты, которые выполняют поставленные перед ними задачи, проводя манипуляции в физическом мире. Для исследования задач управления роботами существуют специальные программные средства. Главная их задача – предоставить пользователю возможность работать с максимальной эффективностью, ввиду этого очень важной частью программных средств становится эмулятор робота и среды окружения. Необходимость в эмуляторе связана со следующими проблемами:

- наиболее важной проблемой является необходимость в специальном стенде, постройка которого в реальном мире занимает много времени и средств; также во время реальных испытаний есть вероятность повредить оборудование;
- испытание в реальном мире занимает существенно большее время, чем при использовании эмулятора.

Однако при использовании существующих программных средств для исследования задач управления роботами место серьезные ограничения. Во-первых, эти средства проектировались для использования с конкретными моделями мобильных роботов, ввиду чего испытания разработанных программ на реальном роботе не всегда доступны. Во-вторых, в данных средствах отсутствует возможность использовать визуальное проектирование автоматных программ [1–3].

В данной работе авторы подробно рассматривают существующие средства для решения задач управления роботами, а также предлагают собственный вариант такого комплекса.

Постановка задачи

В рамках данной работы ставилась задача создать эффективный комплекс для сквозного проектирования и отладки автоматных систем управления мобильными роботами. В состав комплекса должны входить следующие компоненты: реальный робот; графический язык проектирования автоматных программ; модель робота в среде твердотельного моделирования; среда эмуляции и способ конвертирования модели робота в объект этой среды; контроллер робота с функцией отладки по шагам и функцией интерпретирования графической программы; протокол связи. Система сквозного проектирования должна предоставлять пользователю возможность осуществлять полный цикл создания и отладки мобильного робота, от чертежей деталей до программного обеспечения.

Для построения такого комплекса целесообразно использовать технологию автоматного программирования [4]. Применение автоматного подхода при создании подобных систем управления обладает рядом достоинств, таких как документируемость, возможность верификации, упрощение внесения изменений и т.д. Помимо этого, на практике достаточно часто встречаются задачи, для которых известно, что они могут быть решены при помощи конечных автоматов, но эвристически построить для них автомат чрезвычайно сложно. Попытки решения подобных задач выполняются, в частности, при помощи различных методов автоматического синтеза программ. Одним из таких методов является генетическое программирование [5]. Для применения генетического программирования необходим «быстрый» эмулятор робота, так как получение необходимого числа значений функции приспособленности для реального робота требует больших временных затрат.

Архитектура комплекса

На данный момент существует несколько лабораторий, предназначенных для эмуляции различных роботов и управления ими. Наиболее известные из них – это Microsoft Robotics Developer Studio, Player/Stage/Gazebo и Webots. Подробную информацию о данных лабораториях, а также об их преимуществах и недостатках можно найти в работе [6].

При разработке данного комплекса было принято решение использовать лабораторию Webots [7] ввиду ее гибкости при проектировании мобильных роботов и сцен.

Комплекс состоит из шести основных частей:

1. мобильный робот;
2. система управления с инструментом графического проектирования;
3. контроллер робота (реального и виртуального);

4. протокол связи;
5. модель робота в среде твердотельного моделирования Solidworks [8] с возможностью конвертирования в объект среды Webots;
6. эмулятор Webots;

Функционирование комплекса осуществляется за счет взаимодействия системы управления, контроллера робота и реального (виртуального) робота. Взаимодействие между системой управления и контроллером робота осуществляется по проводной или беспроводной связи при помощи команд протокола.

Робот

Объектом управления в данном комплексе является робот. Он включает в себя: аппаратную вычислительную платформу Arduino, выполненную на основе микроконтроллера Atmel AVR, два электродвигателя с редукторами и колесами, литий-ионный аккумулятор, датчик расстояния, четыре датчика линии.

Для данного робота была выбрана готовая аппаратная платформа Arduino Diecimila, обладающая интерфейсом RS-232 и микросхемой конвертера USB-to-Serial FT232R, благодаря которой связь с ПК осуществляется через USB-интерфейс.

Впоследствии к данной плате был припаян модуль BTM-11-CS-96, позволяющий осуществлять беспроводную связь с роботом, что увеличивает эффективность и удобство работы с комплексом. Решение о выборе вычислительной платформы было принято на основе ее функциональности и удобства использования. Вместе с вычислительной платформой также поставляется среда разработки Arduino, представляющая собой компилятор C++ с дополнительной функциональностью. Важными функциональными особенностями самой платы Arduino Diecimila являются возможность питания платы за счет соединения USB, возможность замены вышедшего из строя микроконтроллера Atmel AVR, наличие UART интерфейса, позволяющего добавлять к платформе различные передающие модули.

Готовый робот представлен на рисунке.

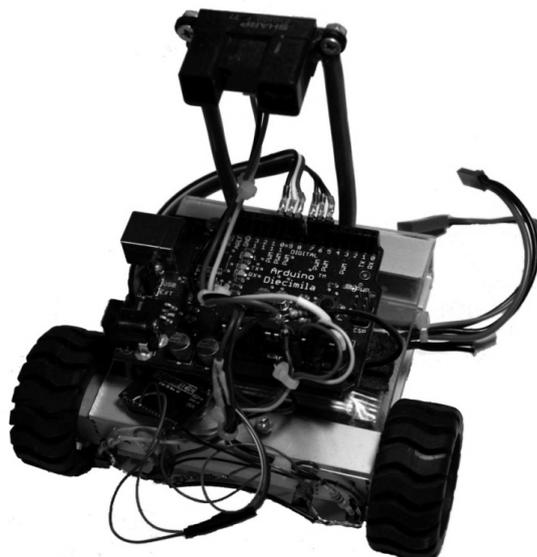


Рисунок. Фотография реального робота

Контроллер

Управление роботом осуществляется за счет изменения различных переменных, таких как скорость вращения двигателей, направление их вращения и т.д. Как реальным, так и виртуальным роботом управляют специально разработанные контроллеры, каждый из которых включает в себя два режима работы. Сразу после включения контроллер находится в состоянии ожидания, оно изменяется в случае получения контроллером какой-либо команды от системы управления. При получении от системы управления команды сброса контроллер возвращается в режим ожидания.

Первый режим, так называемый режим отладки, включается при получении контроллером одной из команд протокола, относящейся к взаимодействию с переменными («Leftspeed», «GetDistance» и т.д.). В режиме отладки система управления либо запрашивает показания различных датчиков робота, либо устанавливает необходимые значения переменных контроллера. В таком случае робот управляется непосредственно из системы управления на ПК, что позволяет отлаживать по шагам исследуемый автомат. В данном случае шагом является один опрос датчиков робота.

Второй режим – автономный. Он включается, если контроллер получает команду «SetAutomata». В этом режиме контроллер получает от системы управления представление конечного автомата и интер-

претирует его, автоматически изменяя значения переменных и опрашивая датчики. Контроллер спроектирован в среде разработки Arduino, включающей в себя редактор кода, компилятор и модуль передачи прошивки в аппаратную вычислительную платформу Arduino. Благодаря тому, что среда разработки Arduino является продуктом с открытым исходным кодом, в нее можно вносить произвольные изменения. Это является важным преимуществом по сравнению со средой разработки AvrStudio.

Протокол связи

Как реальный, так и виртуальный роботы управляются при помощи специального единого протокола. Это позволяет использовать идентичные интерфейсы контроллеров как для реального, так и для виртуального роботов. Благодаря этому отсутствует необходимость производить изменения системы управления при взаимодействии с контроллерами, имеющими различные архитектуры. Подробнее ознакомиться с особенностями протокола можно в работе [7].

Система управления

Система управления предназначена для создания, редактирования и отладки конечных автоматов. Она позволяет проектировать конечные автоматы при помощи встроенных инструментов, а также запускать и отлаживать их. Для запуска и отладки автоматов используется «модуль связи». В режиме отладки в системе управления хранится и обновляется набор переменных, отвечающих за управление роботом. Набор переменных делится на две части: переменные, осуществляющие входные воздействия, и переменные, осуществляющие выходные воздействия. На основе переменных, отвечающих за входные воздействия, выбирается следующее состояние автомата, а результатом выходных воздействий является изменение соответствующих переменных. После этого набор переменных в системе управления синхронизируется при помощи передающего модуля с набором переменных в роботе.

Модель робота в среде Solidworks

Виртуальная модель робота была спроектирована в системе твердотельного моделирования Solidworks. Это позволяет экспортировать ее в среду Webots, а также получать чертежи для изготовления реальных роботов. Кроме того, проектирование в среде Solidworks обладает рядом других достоинств.

Во-первых, при помощи встроенного в Webots 3D-редактора нельзя получить столь точные модели, как в среде Solidworks, так как встроенный редактор включает в себя лишь несколько примитивных объектов, таких как сфера, цилиндр, параллелепипед и т.д.

Во-вторых, в среде Solidworks есть различные инструменты для оценки спроектированной модели, такие как проверка интерференции, расчет массовых характеристик и т.д.

В Webots была загружена модель мобильного робота, спроектированная в Solidworks. Загрузка модели в Webots осуществлялась в несколько этапов:

- Выгрузка модели из Solidworks в формате VRML V2.0, при этом важно расположить одну ось координат параллельно оси колес, а начало координат в центре оси колес;
- Загрузка трехмерных объектов в Webots;
- Создание модели робота в Webots на основе загруженных объектов. Подразумевается назначение каждому массиву точек соответствующей роли, например, дальномер – объект DistanceSensor среды Webots и т.д.

Использование комплекса

При помощи комплекса в виртуальной среде была разработана система управления мобильным роботом на основе автоматного подхода. При этом система моделирования комплекса позволила успешно перенести полученную систему управления виртуальным роботом на реального робота без изменений.

Заключение

В данной работе рассматривались основные компоненты и возможности комплекса. С учетом вышесказанного можно считать, что поставленные задачи были выполнены.

- Разработан графический язык автоматного программирования с возможностью отладки, ориентированный на мобильных роботов.
- Разработана методика для конвертирования модели робота из среды SolidWorks в среду эмулятора Webots, с последующей возможностью эмуляции различных задач.
- На основе модели робота в среде SolidWorks изготовлен и протестирован мобильный робот.

Исследование выполнено по Федеральной целевой программе «Научные и научно-педагогические кадры инновационной России на 2009–2013 годы» в рамках государственного контракта П2236 от 11 ноября 2009 года.

Литература

1. Шалыто А.А. Технология автоматного программирования // Труды Всероссийской научной конференции «Методы и средства обработки информации». – М.: МГУ, 2003 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/works/tech_aut_prog/, своб.
2. Клебан В.О., Шалыто А.А., Парфенов В.Г. Построение системы автоматического управления мобильным роботом на основе автоматного подхода // Научно-технический вестник СПбГУ ИТМО. – 2008. – № 53. – С. 281–285.
3. Клебан В.О., Шалыто А.А. Использование автоматного программирования для построения многоуровневых систем управления мобильными роботами // Сборник тезисов 19 Всероссийской научно-технической конференции «Экстремальная робототехника». – СПб: ЦНИИ РТК, 2008. – С. 85–87.
4. Шалыто А.А., Туккель Н.И. Switch-технология – автоматный подход к созданию программного обеспечения «реактивных» систем // Программирование. – 2001. – № 5. – С. 22–28.
5. Царев Ф.Н., Шалыто А.А. О построении автоматов с минимальным числом состояний для задачи об «Умном муравье» // Сборник докладов X международной конференции по мягким вычислениям и измерениям. – СПбГЭТУ «ЛЭТИ». – 2007. – Т. 2. – С. 88 – 91. [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/download/ant_ga_min_number_of_state.pdf, своб.
6. Алексеев С.А. Программно-аппаратный комплекс для исследования автоматного управления мобильными роботами. – СПбГУ ИТМО. – 2010 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/papers/_alekseev_bachelor.pdf, своб.
7. Cyberbotics Ltd. Webots reference manual. 2009 [Электронный ресурс]. – Режим доступа: <http://www.cyberbotics.com/cdrom/common/doc/webots/reference/reference.html>, своб.
8. Тику Ш. Эффективная работа: SolidWorks. 2004. – СПб: Питер, 2005.

Алексеев Сергей Андреевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, alex.itmo@gmail.com

Клебан Виталий Олегович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, vk.developer@gmail.com

Шалыто Анатолий Абрамович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, зав. кафедрой, shalyto@mail.ifmo.ru

УДК 004.932.72'1

**ДЕТЕКТОРЫ ОСОБЕННОСТЕЙ В МЕТОДЕ ВИОЛЫ–ДЖОНСА,
ПОСТРОЕННЫЕ НА ОСНОВЕ КОНЕЧНЫХ АВТОМАТОВ**

П.А. Скорынин

Рассматривается модификация метода Виолы–Джонса – одного из самых эффективных методов классификации изображений. Вместо детекторов прямоугольных особенностей в предложенной модификации используются детекторы особенностей, построенные на конечных автоматах. Применение автоматов позволяет сократить число уровней в каскаде и число детекторов на каждом уровне за счет того, что детектор, управляемый автоматом, может обнаружить более сложные особенности, нежели простые прямоугольники.

Ключевые слова: классификация изображений, конечные автоматы, машинное обучение.

Введение

Задачи, связанные с компьютерным зрением, были всегда, пока существуют компьютеры. Основным шагом в создании эффективных алгоритмов в данной области стало применение в них методов машинного обучения. Одна из наиболее распространенных задач компьютерного зрения – задача о классификации. В рамках этой задачи требуется определить, принадлежит ли некоторое изображение или его часть к определенному классу. Примером такой задачи является задача о локализации лиц на изображении.

Один из примеров применения машинного обучения в задачах компьютерного зрения – метод Виолы–Джонса [1], являющийся одним из наиболее эффективных методов в своем классе. Основная его идея заключается в том, чтобы вместо одного сложного классификатора использовать каскад сильных классификаторов, построенных из слабых классификаторов, представляющих собой детекторы особенностей.

Слабые классификаторы составляют основу получаемого сильного классификатора и во многом определяют его эффективность. Заметим, что прямоугольные особенности не всегда достаточно хорошо могут описать характерные признаки искомого класса либо для этого требуется линейная комбинация достаточно большого числа таких особенностей. Таким образом, несмотря на высокую в среднем скорость работы получаемых классификаторов, в худшем случае требуется вычислять большое число особенностей. В работе Виолы и Джонса [1] для задачи детекции лиц был получен классификатор, в котором использовалось 4297 детекторов прямоугольных особенностей.

Постановка задачи и обоснование метода

Целью данной работы является модификация метода Виолы–Джонса с использованием детекторов особенностей на основе конечных автоматов. Синтезированный при помощи модифицированного метода классификатор должен распознавать признаки не хуже классификатора, синтезированного методом Виолы–Джонса.

Существует два способа повышения быстродействия методов компьютерного зрения. Первый – алгоритмическое улучшение, второй – аппаратная реализация методов. Использование детекторов особенностей на основе конечных автоматов может дать преимущество в эффективности и позволит уменьшить число слабых классификаторов в каскаде, поскольку прямоугольные особенности ограничены формой, а автомат, «двигаясь» по пикселям изображения, может распознавать не только прямоугольные элементы. При этом конечные автоматы легко реализуются на программируемой логической интегральной схеме (ПЛИС) [2]. Главное их преимущество – отсутствие сложных операций. Все, что необходимо – это сравнение и сложение. Поэтому использование конечных автоматов в детекторах особенностей гарантирует их легкую реализацию на ПЛИС с минимальными затратами ресурсов.

Описание управляющего автомата

В детекторах особенностей используется конечный автомат Мили. Выходные воздействия такого автомата генерируются в зависимости от текущего состояния и входного воздействия. В данной работе используются конечные автоматы Мили из восьми состояний.

В процессе своей работы автомат «двигается» по окну изображения размером 19×19 , начиная с некоторой фиксированной позиции. Находясь в некотором пикселе изображения, автомат «видит» текущий пиксель и еще четырех соседей. По разнице между текущим пикселем и соседями принимается решение о передвижении на один пиксель в одну из четырех сторон. Кроме того, на каждом шаге отдается голос «за» или «против» текущего окна.

Работа автомата заканчивается через некоторое число шагов. Опытным путем было выяснено, что наилучшим числом шагов для заданного размера окна является 10. При «выходе» автомата за край окна окно сразу отклоняется. Если число голосов «за» превышает пороговое значение T_v , рассматриваемое окно считается принятым. В противном случае оно также отклоняется. Оптимальный порог T_v необходимо находить в процессе обучения слабого классификатора.

Входные и выходные воздействия

Для описания входных воздействий применяется тернарная логика. Всего используется четыре входных переменных – по одной на каждого соседа текущего пикселя. Рассмотрим пример вычисления входной переменной для одного из соседних пикселей. Если значения в текущем и соседнем пикселях отличаются не более чем на пороговое значение T_d , то записываем в соответствующую переменную 0. В противном случае, если значение в текущем пикселе больше, записываем «1», иначе записываем «-1». Таким образом, получаем четыре значения для четырех соседей. Всего возможных вариантов оказывается $3^4 = 81$. На рис. 1 показан пример того, как вычисляется входное воздействие.

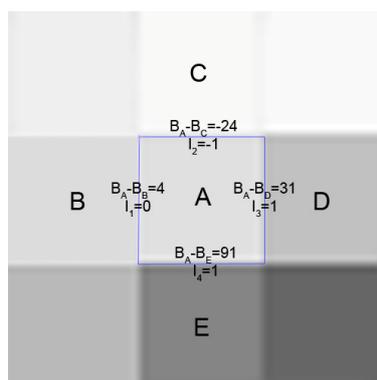


Рис. 1. Пример вычисления входного воздействия

На выходе автомат осуществляет следующие воздействия:

- направление движения (вверх, вниз, влево или вправо);
- голос «за»/«против»;
- новое состояние (одно из восьми).

Генерация автоматов

Чтобы оценить эффективность детекторов особенностей на конечных автоматах и сравнивать их между собой, определим следующую фитнес-функцию:

$$f = k \frac{T_p}{P} + m \frac{T_N}{N},$$

где T_p – число положительных тестов, на которых автомат ответил «Да»; T_N – число отрицательных тестов, на которых автомат ответил «Нет»; P – общее число положительных тестов, N – общее число отрицательных тестов; k и m – весовые коэффициенты. В данной работе $k=3$, а $m=1$. Выбор такой фитнес-функции обоснован необходимостью получения низкого уровня ошибок I рода, так как в каскаде ошибки накапливаются. Поэтому вес правильно определенных положительных тестов выше, чем вес правильно определенных отрицательных тестов.

Таблица переходов для каждого из восьми состояний автомата генерируется случайно. Среди сгенерированных автоматов выбираются только те, уровень ошибок I и II рода которых является приемлемым при некоторых пороговых значениях T_d и T_v . В данной работе для этих параметров были эмпирически подобраны значения в 5% и 60% соответственно.

После этого запускается процесс последовательного улучшения автомата. Для этого применяется (1+1) эволюционная стратегия [3]. Мутацией в данной работе является изменение случайно выбранного перехода. Так как при одной мутации изменяется лишь один из 648 переходов, можно говорить о малом изменении автомата в ходе мутации.

Сильные классификаторы

Слабым классификатором называется такой классификатор, который правильно классифицирует изображения с вероятностью более 50%. Усиление слабых классификаторов – подход к решению задачи классификации путем комбинирования нескольких слабых классификаторов в один сильный. Для этой задачи может подойти любой алгоритм усиления (boosting). В данной работе для выбора особенностей, а также для обучения сильного классификатора используется алгоритм AdaBoost [4].

Фройнд и Шапир (Freund, Schapire) в работе [4] доказали, что ошибка обучения у сильного классификатора приближается к нулю в экспоненциальной зависимости от числа циклов. Кроме того, метод AdaBoost показал высокую эффективность на практике.

Вычисление классификатора из двух особенностей на основе конечных автоматов требует около 80 команд микропроцессора. Сканирование же простых шаблонов изображений или вычисление простого персептрона [5] требует в 15 раз больше операций на одно окно.

Каскад

В работе [1] описан алгоритм построения каскада классификаторов, который позволяет увеличить эффективность классификации и сократить время вычислений. Идея алгоритма заключается в том, что могут быть построены меньшие и, следовательно, более эффективные сильные классификаторы, которые отклоняют большую часть отрицательных окон при обнаружении почти всех положительных. При этом простые классификаторы будут отклонять большую часть отрицательных окон до того, как будут использованы более сложные классификаторы для достижения низкого уровня ошибок I рода.

Уровни каскада строятся из сильных классификаторов, обученных с помощью AdaBoost. Начиная с сильного классификатора из одной особенности, эффективный классификатор может быть получен путем корректировки порога сильного классификатора для минимизации ошибки I рода.

На рис. 2 представлен общий вид процесса классификации – это вырожденное дерево решений, называемое *каскадом*.

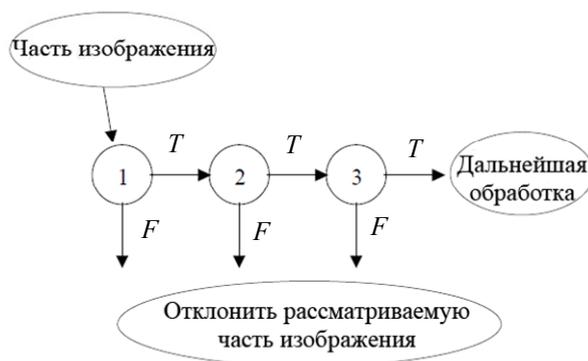


Рис. 2. Общий вид каскада сильных классификаторов

- Осуществляется последовательное применение классификаторов к изображению:
- простые классификаторы отбрасывают часть отрицательных окон, при этом принимая почти все положительные окна;
 - положительный отклик первого классификатора запускает вычисление второго, более сложного, классификатора, положительный отклик второго запускает третий и так далее;
 - отрицательный отклик на любом этапе приводит к немедленному отклонению окна.

Обучение каскада классификаторов

В большинстве случаев классификаторы с большим числом особенностей достигают более высокого уровня распознавания и низкого уровня ошибок I рода. В то же время классификаторы с большим числом особенностей требуют больше времени для вычисления. Это может послужить базой оптимизации, при которой для оптимизации ожидаемого числа особенностей N , решающих задачу для уровня ошибки II рода F и уровня распознавания для каскада классификаторов I , варьируются следующие величины: количество уровней классификатора, количество особенностей n_i на каждом уровне, порог на каждом уровне.

Поиск оптимальных значений параметров является трудной задачей. В настоящей работе применяется простой, но демонстрирующий хорошие результаты алгоритм оптимизации.

Структура полученного классификатора

При синтезе детекторов особенностей на конечных автоматах для решения задачи детекции лиц были использованы те же базы изображений, что и в работах [6] и [7]. На первом этапе работы метода было синтезировано 250 детекторов особенностей на конечных автоматах. Уровень детекции положительных примеров у них колеблется от 96% до 99%. Ошибка II рода составляет от 23% до 55%. Получившиеся детекторы удовлетворяют требованиям для слабых классификаторов.

Для обучения каскада были выбраны следующие значения параметров: максимально допустимый уровень ложных срабатываний – 50% на слой, минимально допустимый уровень обнаружения – 99% на слой. При таких параметрах обучения максимальная общая ошибка II рода в 0,1% может быть достигнута каскадом, состоящим всего из 10 слоев.

В результате работы обучающего алгоритма был построен каскад сильных классификаторов, состоящий из 10 слоев. Первый слой состоит из одного детектора особенностей, который отсеивает более 75% отрицательных окон. Уровень распознавания первого слоя близок к 100%. Второй слой состоит уже из трех детекторов. После него отсеиваются более 90% отрицательных окон при уровне распознавания по-прежнему близкому к 100%. Следующие слои представляют собой линейные комбинации слабых классификаторов и включают в среднем по 6 детекторов. Общее число использованных в каскаде слабых классификаторов равно 52.

Тестирование получившегося классификатора и сравнение с аналогами

Для тестирования классификатора использовалась база изображений лиц CMU (Carnegie Mellon University) [7]. Та же база использовалась и для тестирования классификатора в работе Виолы и Джонса [1]. База изображений включает в себя 472 изображений лиц и 23573 изображений других типов (не лиц). При тестировании использовались те данные, которые не использовались при обучении. Та же методика использовалась Виолой и Джонсом.

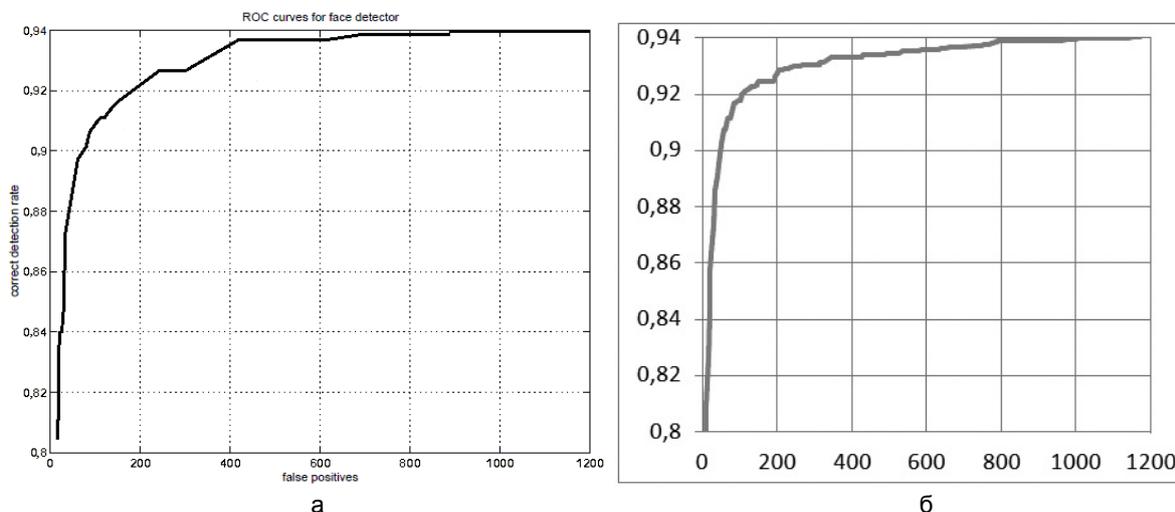


Рис. 3. ROC-кривая для классификатора, построенного: оригинальным методом Виолы-Джонса (а); на детекторах особенностей на конечных автоматах (б)

На рис. 3 показаны для сравнения ROC-кривые [8] для классификаторов, построенных в работе Виолы и Джонса и в данной работе.

В таблице приведены значения уровня распознавания для некоторых значений числа ложных срабатываний.

Число ложных срабатываний	10	31	50	65	78	95	110	167	422
Автоматы	81,5%	87,5%	89,9%	90,9%	91,4%	91,7%	92,0%	92,4%	93,3%
Виола-Джонс	78,3%	85,2%	88,8%	89,8%	90,1%	90,8%	91,1%	91,8%	93,7%

Таблица 1. Уровень распознавания при различном числе ложных срабатываний

Заключение

Результаты проделанной работы:

- построены слабые классификаторы на основе конечных автоматов;
- разработан и реализован метод автоматического синтеза слабых классификаторов на конечных автоматах;
- проведен анализ полученных результатов, который показал, что синтезированный классификатор не уступает по качеству классификатору, построенному методом Виолы–Джонса, но при этом обладает более высокой скоростью работы;
- реализован метод построения каскада классификаторов на основе детекторов особенностей на конечных автоматах.

В дальнейшем планируется разработка метода автоматического преобразования получаемого классификатора в язык описания схем и реализация полученного классификатора на реальном аппаратном обеспечении. Для достижения высокой надежности также возможно комбинирование аппаратной реализации метода с программной реализацией других методов: области, выделенные классификатором на автоматах, следует подвергать дополнительной проверке программными методами. При этом существенно сократится время программной обработки.

Кроме того, одним из направлений в развитии метода является усовершенствование структуры автомата в детекторах особенностей. Применение сокращенных таблиц [9] и увеличение числа состояний может помочь улучшить поведение автомата. Также существует возможность применять дополнительные фильтры над исходным изображением для того, чтобы подчеркнуть на нем некоторые особенности или, наоборот, сгладить шумы. При этом требуется, чтобы любые модификации не уменьшали скорость метода и не усложняли процесс реализации на ПЛИС.

Литература

1. Jones M. Robust Real-time Object Detection // Journal of Computer Vision. – 2004. – № 57(2). – P. 137–154.
2. Стешенко В.Б. ПЛИС фирмы Altera: элементная база, система проектирования и языки описания аппаратуры. – М.: Додэка-XXI, 2007.
3. Beyer H.-G. Evolution strategies – A comprehensive introduction // Natural Computing: an international journal. – 2002. – № 1. – P. 3–52.
4. Schapire R.E. Boosting the margin: A new explanation for the effectiveness of voting methods // Ann. Stat. – 1998. – № 26(5). – P. 1651–1686.
5. Kanade T. Neural network-based face detection // Patt. Anal. Mach. Intell. – 1998. – № 20. – P. 22–38.
6. Sung K.-K. Learning and Example Selection for Object and Pattern Detection: PhD thesis: 13.03.1996. Massachusetts Institute of Technology. 1996.
7. Heisele B. Face Detection in Still Gray images / A.I. memo AIM-1687, Artificial Intelligence Laboratory, MIT. – 2000.
8. Fawcett T. An introduction to ROC analysis // Pattern Recognition Letters. – 2006. – № 27. – P. 861–874.
9. Точилин В.Н. Метод сокращенных таблиц для генерации автоматов с большим числом входных воздействий на основе генетического программирования: Магистерская диссертация. – СПб: СПбГУ ИТМО, 2008. – 130 с.

Скорынин Павел Александрович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, pavel.skorynin@gmail.com

МЕТОДЫ ОПТИМИЗАЦИИ СТРАТЕГИЙ В ИГРАХ ДЛЯ ДВУХ УЧАСТНИКОВ С ИСПОЛЬЗОВАНИЕМ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

Д.А. Трофимов, А.А. Шалыто

Предложена модификация генетического алгоритма, позволяющая решать задачу оптимизации без задания целевой функции в явном виде, используя только функцию сравнения пары решений-кандидатов. Для этого используются так называемые турнирные схемы. Предложенный алгоритм позволяет применять основные принципы генетических алгоритмов в тех задачах, в которых применение генетических алгоритмов в классическом виде невозможно, либо слишком неэффективно. К указанному классу задач относятся задачи построения оптимальной стратегии против заранее неизвестного оппонента в играх двух участников.

Ключевые слова: генетические алгоритмы, игры для двух участников.

Введение

Во многих задачах многомерной оптимизации, где невозможно найти точное решение за приемлемое время, используются разнообразные эвристические алгоритмы. Достаточно часто для этой цели используются генетические алгоритмы, отличительной особенностью которых является наличие операций скрещивания и мутации между решениями-кандидатами [1].

К сожалению, не для всех задач генетические алгоритмы применимы в своем классическом виде. Это связано с тем, что для каждого решения необходимо вычислять значение так называемой функции приспособленности. При этом возникает проблема определения такой функции и алгоритма ее вычисления. Если функцию приспособленности задавать в точном соответствии с критерием оптимизации исходной задачи, временные затраты на вычисление функции в заданной точке могут быть слишком велики. Поскольку для получения качественного результата в ходе выполнения генетического алгоритма требуется выполнять такие вычисления большое число раз, общее время работы алгоритма становится неприемлемо высоким. По мнению авторов, это является главным недостатком генетических алгоритмов по сравнению с другими методами оптимизации.

В данной работе рассматривается класс задач, в которых при отсутствии заданного алгоритма вычисления функции приспособленности для одного отдельного взятого решения известен алгоритм сравнения двух решений, определяющий лучшее из них.

Известны следующие способы решения таких задач.

- Выбирается «эталонное» решение, далее задача сводится к нахождению такого решения, результат игры которого против эталонного максимален. Недостатки такого подхода: требуется наличие такого эталона, результат которого зависит от выбора эталона, если сравнение выполняется только качественно, но не количественно (т.е. можно определить, лучше ли данное решение, чем эталон, но нельзя определить, насколько), невозможно этим способом выбрать лучшее среди всех решений, побеждающих эталонное.
- Метод восхождения к вершине (hill climbing) [2] заключается в следующем. Случайным образом выбирается начальное решение. Далее на каждом шаге создается случайная модификация текущего решения. Полученное решение сравнивается с исходным и, если оно оказывается лучше, заменяет собой текущее, в противном случае оно отбрасывается. Процесс повторяется заданное число шагов.

В настоящей работе представлен способ модификации генетического алгоритма. Полученный алгоритм выполняет оптимизацию заданной игровой стратегии с выделенными параметрами оптимизации и не требует для своей работы вычисления в явном виде функции приспособленности отдельно взятой стратегии, используя только заданную функцию сравнения пар решений-кандидатов. Алгоритм принципиально отличается от обычного генетического алгоритма способом отбора. Алгоритм не требует возможности количественного сравнения и работает с недетерминированной (стохастической) функцией сравнения.

Рассматриваемый класс задач

Игра – процесс, в котором участвуют две стороны, ведущие борьбу за реализацию своих интересов [3]. Каждая из сторон имеет свою цель и использует некоторую стратегию, которая может вести к выигрышу или проигрышу в зависимости от поведения противоположной стороны.

Далее рассматриваются игры для двух игроков с *нулевой суммой*, в которых выигрыш одного участника равен проигрышу второго.

Стратегия – алгоритм поведения игрока, принимающего участие в игре, обладающий некоторым количеством настраиваемых параметров.

Во многих играх, в частности, в играх с большим количеством состояний или с неполной информацией невозможно (либо требует неприемлемо большого времени) построение детерминированной оптимальной стратегии. Для стратегий в таких играх, реализованных «вручную», типично наличие многих параметров (пороговых значений, весовых коэффициентов), которые подбираются эмпирически. Каждый такой параметр далее считается вещественным числом, принадлежащим промежутку $[0, 1)$.

Участник – стратегия с зафиксированными значениями параметров.

Пространство множества участников для стратегии с K параметрами представляет из себя пространство векторов $[0, 1)^K$.

Множество возможных участников игры U – это объединение по всем стратегиям множеств участников для каждой стратегии. Каждый участник задается идентификатором своей стратегии и вектором параметров этой стратегии.

Судья (или функция сравнения) – алгоритм, на вход которому подаются два участника, выдающий результат игры между ними – выигрыш первого участника (или, что тоже самое, проигрыш второго участника). В зависимости от правил игры возможны следующие варианты множества допустимых результатов:

- $\{-1; 1\}$ – «-1» означает поражение первого из входных участников, «1» – второго;
- $\{-1; 0; 1\}$ – «-1» означает поражение первого из входных участников, «1» – второго, «0» – ничью;
- вещественное число в интервале $[-1, 1]$ – количественный выигрыш первого участника.

Единственным способом оценки качества того или иного участника является сравнение его с другими посредством вызовов судьи. Внутренняя механика игры, а также семантика отдельных параметров стратегий алгоритму оптимизации в общем случае неизвестна, судья для него представляет собой «черный ящик».

Результат игры в общем случае не является детерминированным, т.е. не гарантируется идентичность результата нескольких игр между одними и теми же участниками.

Постановка задачи оптимизации

Целью алгоритма оптимизации является выбор наилучшей стратегии и набора параметров для нее.

В условиях, когда правила игры не дают количественной оценки отдельно взятого участника, выбор целевой функции (какого участника считать «наилучшим») не является однозначным. Далее за целевую функцию оптимизации принимается средний результат игры данного участника против каждого возможного оппонента, т.е. следующая величина: $f(A) = \int_{B \in U} E \frac{F(A, B) - F(B, A)}{2} dB$.

$$f(A) = \int_{B \in U} E \frac{F(A, B) - F(B, A)}{2} dB$$

Рассматривается поиск максимума этой величины в слабом смысле, т.е. если максимум достигается для нескольких решений, оптимальным считается любое из них.

К сожалению, вычисление данной величины нереализуемо на практике. Далее рассматривается приближение этой задачи – выбор лучшего участника из конечного множества, заданного прямым перебором. В этом случае значение функции для участника равно среднему выигрышу участника в играх против всех возможных оппонентов. Средний выигрыш вычисляется попарным сравнением всех участников из данного множества.

Модификация генетического алгоритма

Ниже описывается алгоритм, наследующий многие признаки генетических алгоритмов (селекция, мутация, кроссовер). Вместо заданной правилами игры функции приспособленности для одного участника, рассчитываемой независимо для каждого участника, алгоритм использует заданную правилами игры функцию сравнения двух участников.

На входе алгоритма – количество стратегий S , количество параметров для каждой стратегии $K_i (i \in [0..S-1])$, а также судья – функция, принимающая на вход упорядоченную пару участников и возвращающая результат игры между ними. Каждый участник кодируется парой $(i \in [0..S-1], V \in [0, 1)^{K_i})$. На выходе алгоритма – наилучший участник, полученный алгоритмом к моменту останова.

Алгоритм выглядит следующим образом:

1. Создать начальную популяцию из N участников. N раз случайно выбирается стратегия, независимо один от другого выбираются параметры этой стратегии;
2. Провести турнир среди N участников, выбрать M лучших;
3. Применяя операции мутации и кроссовера к M отобранным участникам, создать новую популяцию из N участников;
4. Повторить п.п. 2–3 до наступления условий останова;
5. В последнем поколении провести турнир, выявляющий одного лучшего участника, вернуть его в качестве ответа.

Алгоритм получения следующего поколения участников определяется аналогично таковому из классической схемы генетического алгоритма. Далее рассматривается следующий вариант:

1. Выбрать из N участников M ;
2. $N/2$ раз случайно равномерно выбрать пару из M участников с одинаковой стратегией, получить нового участника кроссовером между участниками этой пары;

3. $N/2$ раз случайно равномерно выбрать одного участника из M , создать нового участника мутацией выбранного;
4. N участников, полученных на предыдущих двух шагах – это и есть новое поколение.

Турнир – способ выделения из N участников M лучших, при этом M лучших между собой ранжировать не требуется. Часто используемые турнирные схемы [4, 5].

- Круговой турнир (round-robin, RR) – наиболее объективен, но требует $O(N^2)$ операций сравнения.
- Турниры с выбыванием после первого (single elimination, SE) или после двух поражений (double elimination, DE), требуют порядка $O(N)$ операций сравнения.
- Швейцарский турнир (Swiss-system, SW) – аналог кругового, но с меньшим количеством туров и специальным разбиением на пары, требует порядка $O(M \log NM)$ операций сравнения.

Мутация и кроссовер определены аналогично обычному генетическому алгоритму. Мутация заключается в замене одного случайного параметра участника случайным числом. Кроссовер заключается в том, что значение каждого параметра нового участника выбирается случайным образом между значениями параметров участников-«родителей».

Чтобы сравнить между собой результаты работы нескольких вариаций данного алгоритма, следует прерывать алгоритм спустя некоторое фиксированное, одинаковое для всех алгоритмов время. Результаты работы алгоритмов – лучших участников – сравниваются либо проведением кругового турнира между ними, либо сравнением суммарного результата игр против некоего тестового множества участников. Для измерения времени удобнее использовать не секунды, а число вызовов функции сравнения.

Стохастические функции сравнения

В ряде случаев функция сравнения может возвращать различные результаты при повторных вызовах с одними и теми же значениями аргументов. Это может быть обусловлено следующими причинами:

- случайность заложена в правила игры (например, в карточных играх, где раздача карт в каждой игре генерируется заново);
- даже если правила игры не используют случайных величин, их могут использовать один или оба участника. Например, участник делает не наилучший с его точки зрения возможный ход, а произвольный, при этом вероятность совершения того или иного хода пропорциональна его «качеству» с точки зрения участника.

В случае двух возможных исходов игры – «победа» или «поражение» первого из участников, результат сравнения участников A и B – случайная величина, задаваемая одним параметром $P(A, B)$ – вероятностью победы участника A . Назовем «истинным» более вероятный из двух исходов. Далее рассматривается частный случай, когда $P(A, B) = \text{const}(A, B) = p > 0,5$. Величина $q = 1 - p$ – вероятность функции сравнения выдать результат, противоположный истинному, далее называется *уровнем шума*.

При уровне шума выше некоторого порога сходимость алгоритмов, имевшая место в случае детерминированной функции, может нарушиться. Для борьбы с этим эффектом уровень шума функции сравнения уменьшается следующим алгоритмом: R раз провести игру между одними и теми же участниками, выдать более часто встретившийся исход. Оптимальное значение R для алгоритма с заданной турнирной схемой и ограничением на число вызовов функции сравнения подбирается экспериментальным путем.

Результаты

Для сравнения алгоритмов использовалась следующая игра. Два игрока называют по K вещественных чисел от 0 до 1, побеждает тот, у кого окажется больше позиций, в которых названное им на этой позиции число превысило таковое у противника. Стратегия для данной игры имеет K параметров – вектор чисел, называемых участником. Для моделирования стохастической функции сравнения рассматривался вариант, когда с заданной вероятностью $1 - P$ возвращался результат, противоположный результату исходной игры.

Оценкой алгоритма на всех этапах считался максимум величины $V(A)$ по всем векторам из последнего поколения, где $V(A)$ – медианный элемент вектора A , упорядоченного по возрастанию чисел.

Было поставлено несколько задач оптимизации, каждая из которых задавалась параметром P – вероятностью функции сравнения вернуть истинный результат и параметром C – максимально допустимым числом вызовов функции сравнения. Алгоритм решения задачи оптимизации задается четверкой (T, N, M, R) , где T – турнирная схема; N – число участников в одном поколении; M – число участников, порождающих следующее поколение; R – число повторных вызовов функции сравнения для уменьшения ее шума.

В ряде случаев рассмотренный в данной работе алгоритм показал лучшие результаты, чем простой алгоритм локального поиска (hill climbing, HC), заключающийся в следующем: взять случайное начальное решение, далее взять случайную мутацию текущего решения, сравнить с исходным, оставить в качестве текущего победившее, повторять процесс до истечения лимита времени.

Ниже приведены графики зависимости качества лучшего решения от времени для значения $C = 10000$ при $P = 0,55$ (рис. 1) и $P = 0,8$ (рис. 2) (показаны лучшие алгоритмы для каждой турнирной схемы).

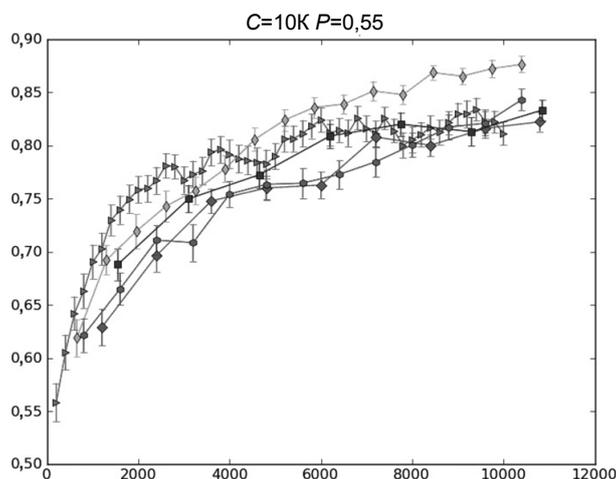


Рис. 1. График зависимости качества лучшего решения от времени для $C=10000$ и $P=0,55$;
 ► HC $R=100$; ◆ RR $R=10$ $N=16$ $M=4$; ■ SE $R=50$ $N=32$ $M=8$;
 ◆ DE $R=50$ $N=8$ $M=2$; ● SW $R=10$ $N=32$ $M=4$

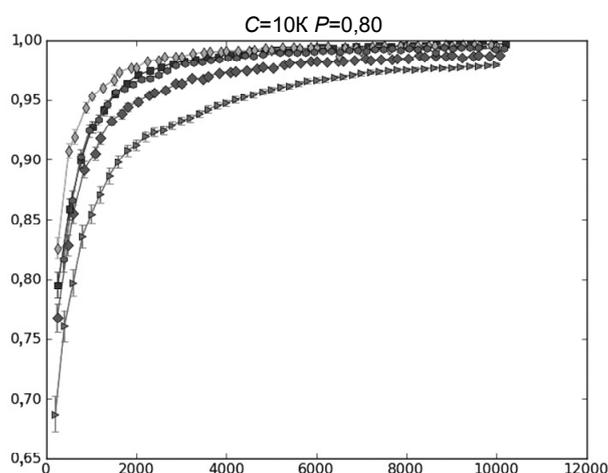


Рис. 2. График зависимости качества лучшего решения от времени для $C=10000$ и $P=0,8$;
 ► HC $R=50$; ◆ RR $R=1$ $N=16$ $M=4$; ■ SE $R=1$ $N=256$ $M=32$;
 ◆ DE $R=1$ $N=64$ $M=8$; ● SW $R=1$ $N=64$ $M=8$

Выводы

На основе полученных результатов сделаны следующие наблюдения и выводы:

- алгоритмы со временем улучшали значение приведенной выше целевой функции, несмотря на то, что они не выполняли ее вычисление ни для одного участника;
- среди рассмотренных турнирных схем не нашлось универсальной, оказавшейся лучшей для всех задач. Выбор оптимальной турнирной схемы зависит от свойств задачи, в том числе от ограничения на время работы;
- при значении $P \geq 0,8$ уменьшение шума функции сравнения оказалось бесполезным для всех турнирных схем (лучшие результаты показали варианты алгоритма с $R=1$).

Открытым вопросом является возможность применения полученного алгоритма на более широком классе задач. Например, представляют интерес задачи, где стохастические функции сравнения имеют более сложный характер, чем рассмотренная простая модель с одинаковым во всех точках уровнем шума. В ходе их исследования в дальнейшем можно использовать рассмотренные в данной работе алгоритмы и выводы.

Литература

1. Goldberg D.E. Genetic Algorithms in Search, Optimization, and Machine Learning. Reading. – MA: Addison-Wesley, 1989.
2. Гилл Ф., Мюррей У., Райт М. Практическая оптимизация. – М.: Мир, 1985.
3. Петросян Л.А., Зенкевич Н.А., Семина Е.А. Теория игр: Учеб. пособие для ун-тов. – М.: Высшая школа, Книжный дом «Университет», 1998.

4. Fayers M. Elimination Tournaments Requiring a Fixed Number of Wins [Электронный ресурс]. – Режим доступа: <http://www.maths.qmul.ac.uk/~mf/papers/meko.pdf>, свободный. Яз. англ. (дата обращения 09.02.2011).
5. Brad L. Miller, David E. Goldberg. Genetic Algorithms, Tournament Selection, and the Effects of Noise [Электронный ресурс]. – Режим доступа: <http://www.illgal.uiuc.edu/web/technical-reports/1995/01/24/genetic-algorithms-tournament-selection-and-the-effects-of-noise-13pp/>, свободный. Яз. англ. (дата обращения 09.02.2011).

Трофимов Дмитрий Алексеевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, dmitriy.tref@gmail.com

Шалыто Анатолий Абрамович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, зав. кафедрой, shalyto@mail.ifmo.ru

УДК 004.4*2

РАЗРАБОТКА МЕТОДОВ ПОСТРОЕНИЯ КОНЕЧНЫХ АВТОМАТОВ С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМА ИМИТАЦИИ ОТЖИГА НА ПРИМЕРЕ ИГРЫ «ВОЙНА ЗА РЕСУРСЫ»

А.К. Заикин

Представленные в работе алгоритмы имитации отжига применяются для генерации автоматов управления защитником в игре «Война за ресурсы». Рассматривается вопрос о применении исследуемых схем алгоритма имитации отжига к задаче построения конечного автомата, управляющего защитником в данной игре, и последующий анализ полученных результатов.

Ключевые слова: алгоритм имитации отжига, конечный автомат, игра «Война за ресурсы».

Введение

Эволюционные вычисления являются одним из активно развивающихся и перспективных направлений в искусственном интеллекте и программировании. Они доказали свою эффективность на практике при решении широкого спектра интересных и сложных задач в различных областях.

В работе [1] предложена игра «Война за ресурсы», на примере которой можно строить эффективные стратегии защиты от нападающего. Целью настоящей работы является применение схем алгоритма имитации отжига [2] для генерации автоматов управления защитником в этой игре. В данной работе исследовался отжиг Коши и его модификации. Для получения энергии решения были рассмотрены два способа оценки полученных автоматов. Выполнены оценка эффективности рассмотренных методов и сравнение их с генетическими алгоритмами.

Постановка задачи

«Война за ресурсы» – это игра для двух игроков на поверхности тора размером N на N . Каждая клетка представляет собой ресурс, за который борются соперники, и может быть свободна или захвачена одним из игроков. В начале игры все клетки, кроме занятых игроками, свободны. Первый игрок (защитник) занимает клетку $(1, 1)$, а второй (нападающий) – (N, N) .

Каждый игрок видит состояния четырех клеток – с севера, юга, запада и востока от себя. В процессе игры противники ходят по очереди. На каждом шаге участник осматривает видимые клетки и перемещается на одну из свободных или ранее захваченных им. Игрок, вставший на свободную клетку, захватывает ее до конца игры.

Игра заканчивается, когда на поле не остается свободных клеток или достигается ограничение по числу шагов. Защитник побеждает, если он захватил больше клеток, чем нападающий.

Нападающий действует по жадной стохастической стратегии: если он видит свободные клетки, то он ходит на произвольную из них; если свободных клеток в поле видимости нападающего нет, то он ходит на любую захваченную им клетку.

Максимальные значения функции приспособленности для полученных автоматов представлены в табл. 1.

	Число состояний									
	1	2	3	4	5	6	7	8	9	10
Значение	0,806	0,867	0,893	0,888	0,901	0,903	0,899	0,905	0,896	0,883

Таблица 1. Значения функции приспособленности лучшего найденного автомата

Метод отжига

Метод отжига [3] предназначен для поиска глобального минимума некоторой функции $f(x)$, заданной для x из некоторого множества S . Элементы множества S представляют собой состояния вообразаемой физической системы («энергетические уровни»), а значение функции $f(x)$ в этих точках используется как энергия системы $E = f(x)$. В каждый момент времени предполагается заданной температура системы T , которая, как правило, уменьшается с течением времени. После попадания в состояние x при температуре T следующее состояние системы выбирается в соответствии с заданным порождающим семейством вероятностных распределений $Q(x, T)$, которое при фиксированных x и T задает случайный элемент со значением $G(x, T)$ в пространстве S . После генерации нового состояния $x' = G(x, T)$ система с вероятностью $h(\Delta E, T)$ переходит к следующему шагу в это состояние, в противном случае процесс генерации x' повторяется. Здесь ΔE обозначает приращение функции энергии $f(x') - f(x)$. Если ΔE меньше нуля, то новое состояние принимается всегда. Величина $h(\Delta E, T)$ называется вероятностью принятия нового состояния.

В данной работе в качестве функции $h(\Delta E, T)$ взято ее приближенное значение:

$$h(\Delta E, T) = \exp(-\Delta E/T), \tag{1}$$

где $\Delta E = E' - E$ – изменение энергии; E' – энергия нового решения; E – текущая энергия; T – температура системы.

Схема алгоритма представлена на рис. 1. Опишем ее.

1. Случайным образом выбирается начальная точка $x = x_0, x_0 \in S$. Текущее значение энергии E устанавливается в значение $f(x_0)$.
2. k -я итерация основного цикла состоит из следующих шагов:
 - а. Сравнить энергию системы E в состоянии X с найденным на текущий момент глобальным минимумом. Если $E = f(x)$ меньше глобального минимума, то изменить значение глобального минимума;
 - б. Сгенерировать новую точку $x' = G(x, T(k))$;
 - в. Вычислить значение функции $E' = f(x')$ в ней;
 - г. Сгенерировать случайное число α из интервала $[0; 1]$;
 - д. Если $\alpha < h(E' - E, T(k))$, то установить $x \leftarrow x', E \leftarrow E'$ и перейти к следующей итерации. Иначе повторять шаг (б) до тех пор, пока не будет найдена подходящая точка x' .

Алгоритм может быть модифицирован на шаге 2, д: переход к следующей итерации может происходить и в том случае, если точка x' не являлась подходящей. При этом следующая итерация начинается с точки x , но уже с новым значением температуры.



Рис. 1. Схема алгоритма отжига

Математическая модель защитника

Для управления защитником использовался конечный автомат Мили [4]. Входные воздействия представляют собой состояния четырех видимых защитнику клеток. Каждая клетка может иметь три состояния. Поскольку все видимые клетки не могут быть захвачены нападающим, получается 80 вариантов входных воздействий.

Игрок в ходе игры может перемещаться в четырех направлениях. Выходным действием автомата будет направление перемещения игрока по полю. Следовательно, имеется 4 варианта выходных действий (С, Ю, З, В).

Представление автомата Мили

Функция переходов и функция выходных воздействий автомата Мили задается с помощью таблицы переходов. В каждом состоянии определены переходы для всех возможных входных воздействий.

Часть таблицы переходов для конечного автомата Мили, управляющего защитником, приведена в табл. 2. Сначала в этой таблице размещается номер текущего состояния, потом четыре столбца значений

входных воздействий (состояний видимых клеток), потом номер следующего состояния и выходное воздействие.

Состояние	x_1	x_2	x_3	x_4	Следующее состояние	Выходное действие
0	0	1	3	0	4	С
1	2	0	1	1	0	В

Таблица 2. Часть таблицы переходов

Для создания таблицы переходов необходимо для каждой пары «номер состояния, входное воздействие» (s, x) назначить номер следующего состояния и выходное действие.

Номер следующего состояния выбирается произвольно среди всех состояний автомата. Выходное действие выбирается среди допустимых выходных действий. Это множество не может быть пусто. Поскольку игра ведется против «жадной» стратегии, из полученного множества были выделены приоритетные действия – захват свободных клеток. Тогда выходное действие выбирается случайно – сначала среди множества приоритетных действий, а если оно пусто, то среди оставшихся допустимых действий.

Алгоритм имитации отжига

Конкретная схема алгоритма имитации отжига задается выбором трех параметров: закона изменения температуры, порождающего семейства распределений и вероятности принятия. Задача оптимизации автомата обладает собственной спецификой. Поэтому применение классических схем метода отжига в чистом виде невозможно.

Рассмотрим каждый этап алгоритма имитации отжига отдельно (рис. 1).

1. *Создание начального решения.* Создается случайный автомат. Для этого выбирается число состояний автомата, начальное состояние и задается таблица переходов.

2. *Оценка решения (оценка нового решения).* На каждом этапе оценки решения вычисляются два показателя: энергия решения и доля выигранных игр. Доля выигранных игр вычисляется аналогично способу, описанному в работе [1], для последующего сравнения. Энергия во всех случаях вычисляется следующим образом:

$$E = 1 - f(x) \in [0, 1],$$

где x – оцениваемое решение; $f(x)$ – оценка решения в интервале $[0, 1]$.

Рассмотрим способы получения оценки энергии решения.

Оценка решения по числу выигранных игр выполняется следующим образом:

$$f(x) = \frac{\text{число_выигранных_игр}}{\text{общее_число_игр}},$$

где x – оцениваемое решение.

Цель игры – захватить больше клеток, чем захватит нападающий. Можно оценивать автомат по числу захваченных во время игры клеток:

$$f'(i) = \frac{\text{число_захваченных_клеток_в_i_игре}}{\text{общее_число_захваченных_клеток_в_i_игре}},$$

где f' – функция оценки результата одной игры.

Оценка по результатам некоторого множества игр вычисляется как среднее арифметическое оценок каждой игры.

3. *Изменение решения случайным образом.* Случайное изменение автомата производилось одним из трех равновероятных способов: изменение начального состояния; случайное изменение таблицы переходов; обмен выходными действиями или номерами следующих состояний.

Некоторые способы изменения автомата используют вероятность изменения, которая передается в качестве параметра и задается следующей формулой:

$$P = \frac{T}{T_0},$$

где T – текущая температура системы; T_0 – начальная температура.

3.1. *Изменение начального состояния.* Новое начальное состояние выбирается случайно и равномерно.

3.2. *Случайное изменение таблицы переходов.* Каждая строка таблицы переходов изменяется с некоторой вероятностью. Если строку необходимо изменить, то равновероятно изменяются либо выходное действие, либо номер следующего состояния. Выходное действие равновероятно выбирается либо среди множества приоритетных, либо среди множества всех допустимых действий. Номер следующего состояния выбирается случайно и равномерно.

Для описания следующего способа изменения автомата представим таблицу переходов в виде таблицы, где в строках расположены номера состояний, а в столбцах – входные воздействия (рис. 2).

	x_0	x_1	x_2
s_0	s_1/y_1	s_2/y_2	s_3/y_2
s_1	s_3/y_1	s_2/y_1	s_1/y_2
s_2	s_2/y_2	s_1/y_1	s_3/y_2

Рис. 2. Представление таблицы переходов

3.3. *Обмен выходными действиями или номерами следующих состояний.* Сначала произвольным образом выбирается расстояние обмена по номерам состояний (Δs) и по входным воздействиям (Δx). Каждой ячейке (s_i, x_j) ставится в соответствие ячейка, которая циклически сдвинута относительно нее на Δx по горизонтали и на Δs по вертикали. Далее каждая пара с некоторой вероятностью обменивается номерами следующих состояний или выходными действиями между собой. Выбор элементов для обмена происходит случайно и равновероятно. На рис. 3 приведен пример данного способа изменения при $\Delta s=1$ и $\Delta x=1$.

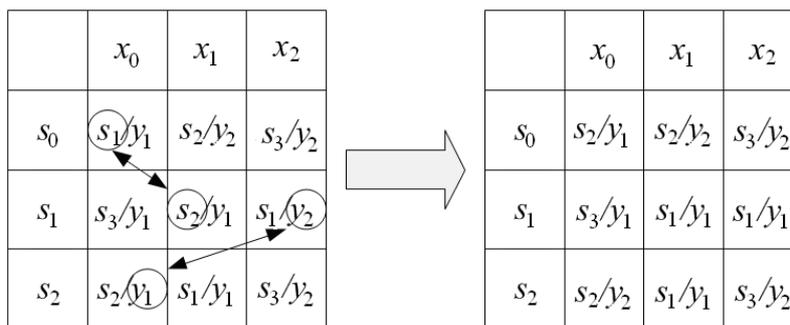


Рис. 3. Пример обмена номерами следующих состояний

4. *Критерий допуска.* В качестве критерия принятия использовалась формула (1).

5. *Изменение температуры.* В данной работе использовался отжиг Коши и две его модификации.

Отжиг Коши уменьшает температуру линейно в зависимости от числа итераций. Вероятность изменения в этой схеме алгоритма имитации отжига зависит только от номера шага:

$$P = \frac{T}{T_0} = \frac{T(k)}{T_0} = \frac{1}{k}, k > 0.$$

Эта формула показывает, что вероятность изменения уменьшается линейно. В результате на этапе случайного изменения решения автомат претерпевает незначительные изменения. Это значительно снижает результативность данной схемы метода отжига.

Для устранения этого недостатка в данной работе предлагается модифицировать эту схему двумя способами. Первая модификация отжига Коши связана с уменьшением температуры только при нахождении новой точки, вторая модификация – с уменьшением температуры только при нахождении нового решения. Если новое решение не обнаружено за определенное число итераций, то температура увеличивается до предыдущего значения.

Результаты экспериментов

Каждый из предложенных алгоритм запускался десять раз, и по результатам строился график среднего значения показателя выигрыша лучшего игрока в зависимости от числа построенных автоматов и сравнивался с аналогичным графиком для генетических алгоритмов (ГА) из работы [1]. Число состояний искомого автомата было равно либо шести, либо восьми.

Отжиг Коши и его модификации запускались со следующими параметрами: начальная температура – 1; число игр для оценки решения – 500. Вторая модификация отжига Коши увеличивала температуру, если новое решение не было найдено за 5000 итераций.

На рис. 4 приведен график лучшего выигрыша для отжига Коши и его модификаций при оценке по числу выигранных игр.

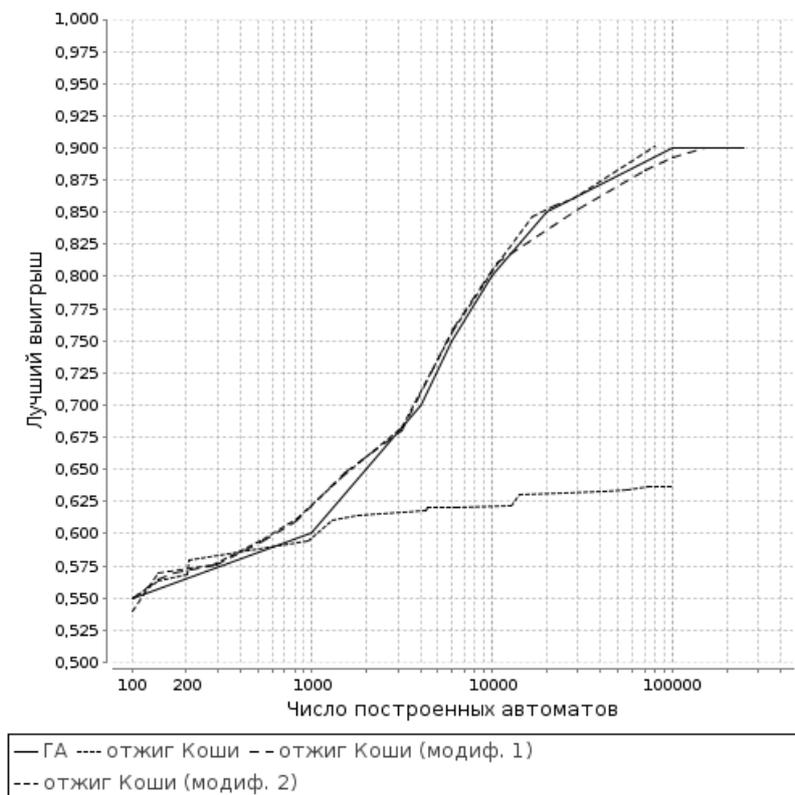


Рис. 4. График среднего значения лучшего игрока для отжига Коши и его модификаций и ГА

Обычный отжиг Коши не дает хороших результатов из-за быстрого уменьшения вероятности изменения. Первая модификация исправляет этот недостаток – ее результаты сравнимы с результатами ГА.

Вторая модификация отжига Коши улучшает результат ГА. На первых 10000 поколениях уменьшение температуры еще не используется, и график практически совпадает с графиком первой модификации. Далее наблюдается рост значения выигрыша.

В табл. 3 приведены значения выигрыша и округленные номера поколений, на которых удалось достичь таких значений.

Показатель	Первая модификация		Вторая модификация	
	Оценка по захваченным клеткам	Оценка по выигранным играм	Оценка по захваченным клеткам	Оценка по выигранным играм
0,6	620	620	600	620
0,7	3700	3700	3700	3600
0,8	9500	9300	9500	9300
0,89	90000	91000	60000	59000
0,9	140000	150000	80000	81000

Таблица 3. Результаты модификаций отжига Коши

Результаты игр автоматов из шести и восьми состояний, полученных с помощью отжига Коши и его модификаций и ГА, приведены в табл. 4.

	6 состояний	8 состояний
Отжиг Коши	0,653	0,636
Первая модификация	0,903	0,904
Вторая модификация	0,903	0,905
Генетический алгоритм	0,903	0,905

Таблица 4. Результаты игр автоматов полученных с помощью отжига Коши и его модификаций и ГА

Заключение

Исследования и эксперименты показали, что предложенный модифицированный метод отжига Коши оказался эффективнее как метода генетического программирования, так и других алгоритмов имитации отжига в рамках игры «Война за ресурсы». Таким образом, результатами работы подтверждена целесообразность применения метода отжига для построения конечных автоматов на примере игры «Война за ресурсы».

Исследование выполнено по Федеральной целевой программе «Научные и научно-педагогические кадры инновационной России на 2009–2013 годы» в рамках государственного контракта П2236 от 11 ноября 2009 года.

Литература

1. Spears W., Gordon D. Evolution of strategies for resource protection problems // Theory and Applications of Evolutionary Computation: Recent Trends. – Springer-Verlag, 2002.
2. Ingber L. Simulated Annealing: Practice Versus Theory // Mathl. Comput. Modelling. – 1993.
3. Лопатин А.С. Метод отжига // Стохастическая оптимизация в информатике. – 2005.
4. Поликарпова Н.И., Шальто А.А. Автоматное программирование. – СПб: Питер, 2009 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/books/_book.pdf, своб.

Заикин Александр Константинович – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, zaikin@rain.ifmo.ru

УДК 004.4'242

МЕТОД ПРЕДСТАВЛЕНИЯ АВТОМАТОВ ЛИНЕЙНЫМИ БИНАРНЫМИ ГРАФАМИ ДЛЯ ИСПОЛЬЗОВАНИЯ В ГЕНЕТИЧЕСКОМ ПРОГРАММИРОВАНИИ

В.Р. Данилов, А.А. Шальто

Предлагается метод представления автоматов в виде особой эволюционного алгоритма, основанный на использовании линейных бинарных графов. На примере выполнено сравнение этого метода с известными методами. Предлагаемый метод является более эффективным по сравнению с представлением функции переходов полными таблицами. При некоторых значениях числа состояний он более эффективен, чем метод представления функции переходов деревьями решений.

Ключевые слова: генетическое программирование, конечные автоматы, линейные бинарные графы.

Введение

Генетическое программирование [1] – метод автоматической генерации программ на основе эволюционных алгоритмов [2], использующий представление программ на высоком уровне абстракции. В работах [3, 4] рассматривалось применение генетического программирования для построения автоматов управления системами со сложным поведением. Автоматы такого рода характеризуются сложностью функции переходов из каждого состояния. Таким образом, методы, основанные на представлении функции переходов автоматов полными таблицами, оказываются неприменимыми на практике. Наиболее близким к предлагаемому в этой работе методу является метод представления автоматов, основанный на деревьях решений [4]. Недостатком такого подхода является повторное кодирование повторяющихся поддеревьев. В настоящей работе предлагается метод представления функции переходов автоматов управления, основанный на линейных бинарных графах [5], который лишен указанного недостатка.

Линейные бинарные графы

Разрешающая диаграмма [6] является удобным способом задания булевой функции. Она представляет собой помеченный ациклический ориентированный граф, в котором выделяют вершины двух типов:

- нетерминальные узлы;
- терминальные узлы.

При этом один из нетерминальных узлов является начальным. Все остальные узлы достижимы из начального. Из каждого нетерминального узла выходит по два ребра. Из терминальных узлов ребра не выходят. Метки в графе расставляются по следующим правилам:

- нетерминальные узлы помечаются названиями булевых переменных;
- терминальные узлы помечаются значениями функции;
- ребра помечаются значениями булевых переменных.

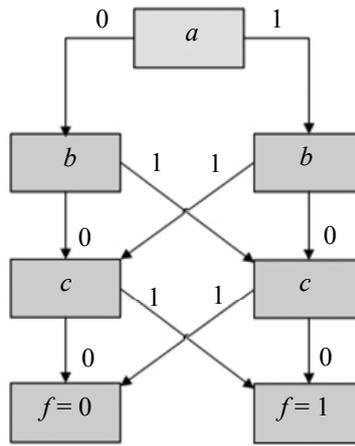


Рис. 1. Пример разрешающей диаграммы

Для определения значения функции по значениям булевых переменных необходимо пройти путь от стартового до терминального узла и определить значение, которым он помечен. При этом из вершины, помеченной булевой переменной x , переход производится по тому ребру, которое помечено тем же значением, что и значение переменной x . На рис. 1 приведен пример разрешающей диаграммы, реализующей булеву функцию $f = a \oplus b \oplus c$.

Важным частным случаем разрешающих диаграмм являются линейные разрешающие диаграммы или линейные бинарные графы. Узлы линейного бинарного графа могут быть пронумерованы таким образом, что из каждого нетерминального узла одно из ребер ведет в узел со следующим номером. При этом число элементов, необходимых для реализации булевой формулы бинарным линейным графом, линейно зависит от числа букв в формуле [5].

Пример линейного бинарного графа, реализующего булеву функцию $f = ab \vee c$, приведен на рис. 2.

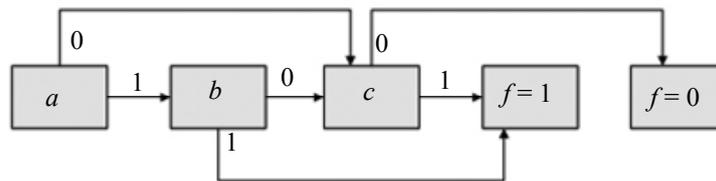


Рис. 2. Пример линейного бинарного графа

Представление автоматов бинарными линейными графами

Опишем предлагаемый метод представления автомата с помощью линейных бинарных графов. Для задания автомата необходимо выразить его функции переходов и действия. Осуществим следующее преобразование автомата: вместо задания функций переходов и действия для автомата в целом определим эти функции в каждом состоянии.

Более формально это выглядит следующим образом: зададим для каждого состояния $q \in Q$ функцию $\sigma_q : X \rightarrow Q \times Y$, такую, что $\sigma_q(x) = (\delta(q,x), \lambda(q,x))$ для $\forall x \in X$.

Функции σ_q соответствуют функциям переходов и действия в состоянии q . Каждая из таких функций может быть выражена с помощью некоторого линейного бинарного графа. Переменными этих графов являются входные переменные автомата, а множеством значений – все возможные пары (Новое Состояние, Действие). Таким образом, автомат в целом задается упорядоченным набором линейных бинарных графов.

Для использования представления автоматов в виде набора линейных бинарных графов в генетическом программировании необходимо определить генетические операции. Это может быть сделано с помощью генетических операций над линейными бинарными графами:

- случайное порождение автомата – в каждом состоянии создается случайный линейный бинарный граф;
- скрещивание автоматов – линейные бинарные графы в соответствующих состояниях скрещиваются;
- мутация автомата – в линейном бинарном графе, соответствующему случайному состоянию, производится мутация.

При этом считается, что число состояний в автомате фиксировано. Поэтому противоречий при выполнении определенных таким образом операций не возникнет.

Теперь определим представление линейных бинарных графов в виде хромосомы и соответствующие генетические операции. Линейный бинарный граф может быть представлен в виде хромосомы в следующей форме.

- Граф представляет собой упорядоченный список узлов. При этом считается, что узлы идут в таком порядке, что из каждого узла одно из выходящих ребер ведет в следующий узел.
- Для каждого нетерминального узла хранится переменная, которой помечен данный узел, значение переменной расщепления, соответствующее переходу в следующий узел, и номер узла, в который производится переход при ином значении переменной расщепления.
- Для каждого из терминальных узлов хранится значение функции, которым помечен соответствующий узел.

При этом терминальные узлы являются последними узлами линейного бинарного графа. Поэтому хромосому можно представить в виде строки, отдельно выделяя части, кодирующие терминальные и нетерминальные узлы.

Закодируем для примера в виде строки граф, приведенный на рис. 2. Заметим, что узлы этого графа могут быть пронумерованы в требуемом порядке слева направо. Тогда этому графу будет соответствовать следующая строка:

a13b04c15 10

Для скрещивания линейных бинарных графов можно применить любой из методов скрещивания строк, отдельно скрещивая части, описывающие нетерминальные и терминальные узлы. При этом часть, соответствующая терминальным узлам, должна быть скопирована полностью из одной родительской хромосомы.

Операция мутации может быть определена следующим способом. Выбирается произвольный нетерминальный узел. После этого в него вносится одно из следующих изменений:

- замена переменной, которой помечен модифицируемый узел;
- замена значения переменной, которым помечен переход в следующий в порядке нумерации узел;
- замена номера узла, в который производится переход при несовпадении значения переменной.

Заключение

Разработанный подход был протестирован на задаче «Умный муравей-2» [7]. Приведем описание задачи. Муравей находится на случайном игровом поле. Поле представляет собой тор размером 32×32 клетки. Муравей видит перед собой некоторую область (рис. 3).

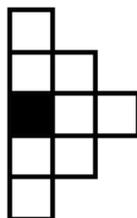


Рис. 3. Видимая муравью область

Еда в каждой клетке располагается с некоторой вероятностью μ . Значение μ является параметром задачи. Игра длится 200 ходов, за каждый из которых муравей может сделать одно из трех действий: поворот налево или направо, шаг вперед. Если после хода муравей попадает на клетку, где есть еда, то еда съедается. Целью задачи является построение стратегии поведения муравья, при которой математическое ожидание съеденной еды максимально.

Автомат управления муравьем в этой задаче имеет восемь (число видимых клеток) булевых входных переменных, каждая из которых определяет, есть ли еда в клетке, соответствующей переменной.

Эксперимент состоял в сравнении полученных значений фитнес-функции (объема съеденной еды) за фиксированное число шагов. Запуск производился со следующими настройками: стратегия отбора – элитизм, для размножения отбирались 25% популяции, имеющих наибольшее значение фитнес-функции; частота мутации – 2%; размер популяции – 200 особей; число популяций – 100; фитнес-функция – среднее значение съеденной еды на 200 случайных картах полей; карты внутри одной популяции совпадают; карты различных популяций различны. Последнее измерение фитнес-функции осуществлялось на случайном наборе из 2000 карт.

Результаты эксперимента приведены в таблице.

μ	0,04			
	Функция приспособленности			
Число состояний	2	4	8	16
Полные таблицы	17,18	15,94	15,03	13,68
Деревья решений	18,28	20,28	18,60	20,18
Предложенный метод	18,82	16,44	19,75	15,46

Таблица. Результаты экспериментов

Анализ результатов показывает, что предложенный метод является более эффективным по сравнению с представлением функции переходов полными таблицами. При некоторых значениях числа состояний предложенный метод более эффективен по сравнению с методом представления функции переходов деревьями решений.

Исследование выполнено по Федеральной целевой программе «Научные и научно-педагогические кадры инновационной России на 2009–2013 годы» в рамках государственного контракта П1188 от 27 августа 2009 года.

Литература

1. Koza J.R. Genetic programming: On the Programming of Computers by Means of Natural Selection. – Cambridge: MIT Press, 1992.
2. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. – М.: Физматлит, 2006.
3. Поликарпова Н.И., Точилин В.Н., Шальто А.А. Применение генетического программирования для генерации автоматов с большим числом входных переменных // Научно-технический вестник СПбГУ ИТМО. – 2008. – № 53. – С. 24–42.
4. Данилов В.Р. Метод представления автоматов деревьями решений для использования в генетическом программировании // Научно-технический вестник СПбГУ ИТМО. – 2008. – № 53. – С. 103–108.
5. Шальто А.А. Логическое управление. Методы аппаратной и программной реализации. – СПб: Наука, 2000. – 780 с.
6. Bryant R.E. Graph-based algorithms for boolean function manipulation // IEEE Transactions on Computers. – 1986. – № 8. – P. 677–691.
7. Бедный Ю.Д., Шальто А.А. Применение генетических алгоритмов для построения автоматов в задаче «Умный муравей». – СПбГУ ИТМО, 2007 [Электронный ресурс]. – Режим доступа: <http://is.ifmo.ru/works/ant>, свободный. Яз. рус. (дата обращения 09.02.2011).

Данилов Владимир Рюрикович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, Vladimir.Daniloff@gmail.com

Шальто Анатолий Абрамович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, зав. кафедрой, shalyto@mail.ifmo.ru

УДК 004.4'242

АВТОМАТИЧЕСКИЙ ПОДБОР ПАРАМЕТРОВ ВНЕШНЕЙ СРЕДЫ ПРИ ГЕНЕРАЦИИ АВТОМАТНЫХ ПРОГРАММ С ПОМОЩЬЮ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

В.А. Кулев

Предлагается новый метод генетического программирования, позволяющий улучшить способ построения функции приспособленности для генетических алгоритмов, проводящих тестирование особей путем симуляции окружающей среды. Метод позволяет автоматически подбирать параметры внешней среды, что приводит к улучшению качества генерируемых особей. В ходе исследований на примере задачи об «Итерированной дилемме узника» было показано, что разработанный подход применим для генерации автоматных программ в задачах, где поведение окружающей среды может быть описано с помощью конечного автомата.

Ключевые слова: генетическое программирование, конечный автомат, автоматное программирование.

Введение

Вычисление функции приспособленности на основе симуляции внешней среды получило наибольшее распространение благодаря своей очевидности и простоте. В некоторых ситуациях этот подход также представляется единственным возможным, при этом множество параметров среды, которое используется для вычисления функции приспособленности, намного меньше множества возможных параметров среды. Это множество, как правило, задается априори, исходя из эвристических соображений.

Целью данной работы является создание метода, позволяющего охватить большую часть пространства возможных параметров внешней среды, при этом сохраняя приемлемую скорость работы генетического алгоритма.

При применении автоматов в программировании, как правило, используются не абстрактные модели автоматов, а модель автоматизированного объекта [1], объединяющая управляющий автомат и объект управления.

Важная черта устройств управления всех абстрактных машин – их конечность. Управляющий автомат не только имеет конечное число состояний, но, кроме того, реализуемые им функции переходов и выходов оперируют исключительно конечными множествами. Именно это свойство позволяет описывать логику поведения машины явно – в виде таблицы или графа переходов. Поэтому свойство конечности устройства управления необходимо сохранить при построении модели автоматизированного объекта. Более того, это свойство целесообразно усилить следующим неформальным требованием: число управляющих состояний, входных и выходных воздействий должно быть небольшим (обозримым). Управляющий автомат с тысячей состояний, безусловно, является конечным, однако изобразить его граф переходов практически невозможно, что сводит на нет преимущества явного выделения управляющих состояний.

Напротив, число состояний объекта управления может быть сколь угодно большим (при переходе от модели к программной реализации оно с необходимостью станет конечным, однако может остаться необозримым). В процессе работы управляющему автомату требуется получать информацию о состоянии объекта управления и изменять его. Однако в силу свойства конечности автомат не может напрямую считывать и записывать это состояние. Для этого и требуются операции объекта управления. Небольшое число запросов, возвращающих конечные значения, позволяет автомату получать информацию о состояниях объекта управления, которую он способен обработать. Небольшое число команд используется для «косвенного» изменения состояний объекта управления.

На рис. 1 сплошными стрелками обозначены типичные для программных реализаций виды взаимодействия между автоматом, объектом управления и внешней средой. Автомат получает входные воздействия как со стороны среды, так и от объекта управления.

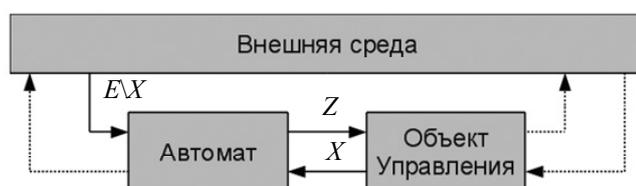


Рис. 1. Модель автоматизированного объекта

Пунктирными стрелками обозначены менее распространенные, хотя и возможные, варианты взаимодействия. Автомат может оказывать выходное воздействие и на внешнюю среду. Однако таких связей обычно можно избежать, включив все управляемые автоматом сущности в состав его объекта управления. Отметим, что в программировании, в общем случае, различие между объектом управления и внешней средой носит скорее концептуальный, а не формальный характер. Создавая модель системы со сложным поведением, разработчик производит ее декомпозицию на автоматизированные объекты, определяя тем самым объект управления каждого автомата. В целях минимизации связей между модулями программной системы целесообразно проводить декомпозицию таким образом, чтобы автомат оказывал выходные воздействия только на собственный объект управления. Кроме того, объект управления может взаимодействовать с внешней средой напрямую.

Предлагаемый метод

В рамках данной работы рассматривается ситуация, в которой поведение внешней среды автоматизированного объекта может быть представлено в виде конечного автомата. В этом случае внешняя среда сама может рассматриваться как автоматизированный объект, что порождает схему, изображенную на рис. 2.

Отметим, что полученная модель очень удобна ввиду естественности взаимодействия ее компонент. Обмен информацией между автоматизированным объектом и внешней средой ведется через входные и выходные воздействия управляющих автоматов, а также через взаимодействие объектов управления.

Также можно выделить класс задач, в которых подразумевается наличие нескольких автоматных программ с одинаковыми наборами входных и выходных воздействий, играющих в некую игру. В этом случае для каждого из игроков внешней средой будут являться его оппоненты, а входными воздействиями управляющего его автомата – выходные воздействия автоматов оппонентов, или их функция $f_z : Z^{N-1} \rightarrow X$, где N – число игроков; Z – множество выходных воздействий; X – множество входных воздействий.

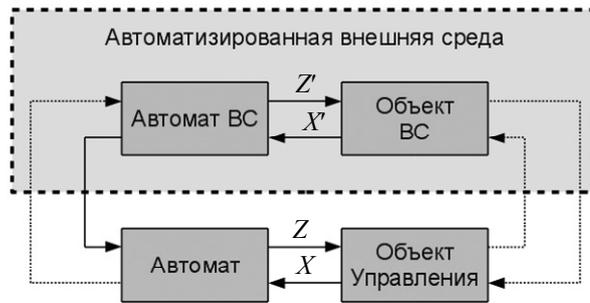


Рис. 2. Модель с автоматизированной внешней средой

Пусть задана функция приспособленности $f_t(i, e)$ для особи $i \in P_t$ и управляющего автомата внешней среды $e \in P_{env}$. Если известно максимальное возможное значение функции f_t для всех i и e , $f_{max} = \max_{(i, e)} [f_t(i, e)]$, то уровень отклонения особи i от оптимального поведения можно оценить как $f_{inv}(i, e) = f_{max} - f_t(i, e)$. Тогда качество теста с автоматом e для всей популяции P_t может вычисляться

$$\text{как среднее значение: } f_{inv}(e) = \frac{\sum_{i \in P_t} f_{inv}(i, e)}{|P_t|}.$$

Однако подбор одиночного управляющего автомата внешней среды не является эффективным способом всестороннего тестирования целевых особей. В связи с этим предлагается хранить множество таких автоматов, которые будут образовывать популяцию особей внешней среды P_{env} . К этой популяции можно применить метод генетического программирования, используя функцию $f_{inv}(e)$ в качестве функции приспособленности. Также необходимо определить модифицированную функцию приспособленности для особей из популяции P_t , вычисляемую как среднее значение:

$$f_t(i) = \frac{\sum_{e \in P_{env}} f_t(i, e)}{|P_{env}|}.$$

Если использовать генетический алгоритм для популяции P_{env} как составную часть основного алгоритма, то получится так называемый «мета-генетический» алгоритм [2]. Однако этот алгоритм работает достаточно длительное время. Для описываемой задачи более эффективно работает модификация классического алгоритма, в которой эволюция обеих популяций производится параллельно.

Сформулируем этапы модифицированного генетического алгоритма, автоматически подбирающего управляющие автоматы внешней среды.

1. **Инициализация.** Создаются две случайные начальные популяции целевых автоматов и управляющих автоматов внешней среды с соответствующими входными и выходными воздействиями. Размер популяций и параметры генерации случайных особей могут различаться.
2. **Тестирование.** На этапе тестирования происходит вычисление функций приспособленности $f_t(i)$ и $f_{inv}(e)$ для всех особей $i \in P_t$ и $e \in P_{env}$. Для этого производится моделирование поведения каждого целевого автоматизированного объекта i в каждой автоматизированной внешней среде e , результатом которой является функция $f_t(i, e)$. Далее искомые значения функций приспособленности вычисляются по приведенным выше формулам.
3. **Отбор.** Выбор особей для проведения генетических операций и формирования нового поколения производится независимо для каждой из популяций, поэтому алгоритм отбора не требует модификации.
4. **Кроссовер и мутация.** Способ проведения кроссоверов и мутаций также не отличается от классического алгоритма, однако можно сформулировать рекомендации по настройке его параметров. Как правило, при применении оператора мутации задается ее «сила», определяющая, насколько большая доля хромосомы будет подвергнута случайной модификации. Данный параметр, в совокупности с общим количеством производимых мутаций, определяет, насколько быстро изменяется генотип популяции в ходе эволюционного процесса. Для обеспечения большей стабильности работы модифицированного генетического алгоритма предлагается устанавливать этот параметр для популяции управляющих автоматов внешней среды ниже, чем для популяции целевых автоматов.

Принципиальная схема модифицированного генетического алгоритма изображена на рис. 3. На ней явно обозначены части алгоритма, выполняющиеся независимо для обеих популяций, и точка, в которой происходит их взаимодействие.

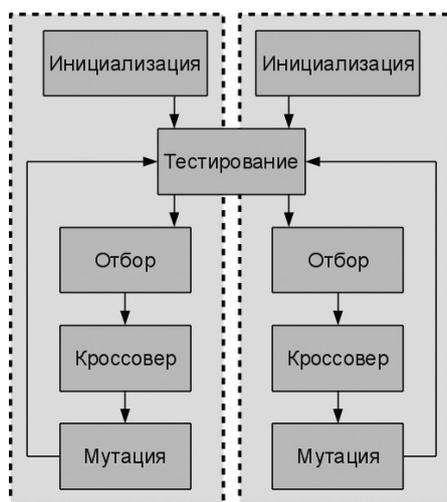


Рис. 3. Схема модифицированного генетического алгоритма

Экспериментальное исследование

Экспериментальное исследование построенного алгоритма было проведено на примере некооперативной матричной игры с ненулевой суммой, обычно называемой «Итерированная дилемма узника» («Iterated prisoner’s dilemma») [3].

В статье [4] описывается эксперимент с применением генетического алгоритма для генерации стратегий управления узником. В качестве хромосомы использовалась битовая строка, кодирующая управляющую функцию $\{S, P, R, T\}^3 \rightarrow \{Cooperate, Defect\}$. На вход функции предоставляются исходы трех предыдущих раундов игры, которые могут быть следующими:

- $S(Sucker)$ – игрок отрицал, когда другой сознался;
- $P(Punishment)$ – оба игрока сознались;
- $R(Reward)$ – оба игрока отрицали;
- $T(Temptation)$ – игрок сознался, когда другой отрицал.

Функция приспособленности особей вычислялась путем турнира между особями популяции. Стратегии, полученные в результате работы построенного алгоритма, имели поведение, близкое к стратегии «Око за око». В турнире многие особи набирали даже большее количество баллов, чем «Око за око», за счет того, что они успешнее противостояли особям из той же популяции.

Эксперимент для изучения работы модифицированного алгоритма был проведен в двух вариациях. Общими настройками алгоритма были:

- размер популяции – 300;
- число состояний автомата – 5;
- вероятность скрещивания – 0,5;
- вероятность мутации – 0,5;
- доля элитизма – 0,125.

Для оценки получаемого результата также подсчитывался выигрыш лучшей особи из популяции против стратегии «Око за око».

В первом эксперименте функция приспособленности для обеих популяций вычислялась по одинаковому алгоритму, который далее будем называть «прямым». В этом случае приспособленность особи определяется как ее средний выигрыш при игре со всеми особями из противоположной популяции. Графики, отражающие ход эксперимента, приведены на рис. 4 и 5.

Три коррелирующие линии (1, 2 и 4) обозначают соответственно минимальное, максимальное и среднее значение функции приспособленности в популяции. Линия 3 обозначает приспособленность лучшей особи из популяции при игре против стратегии «Око за око».

Во втором эксперименте функция приспособленности одной из популяций была заменена на обратную, которая вычисляется не как средний выигрыш игрока, а как средний проигрыш оппонента.

Таким образом, одна из популяций берет на себя четко обозначенную роль контроля качества особей из противоположной популяции. По графикам на рис. 6 и 7, полученным в ходе эксперимента, можно судить о пользе такого подхода, так как получаемые решения лучше ведут себя в игре против неизвестной им стратегии «Око за око».

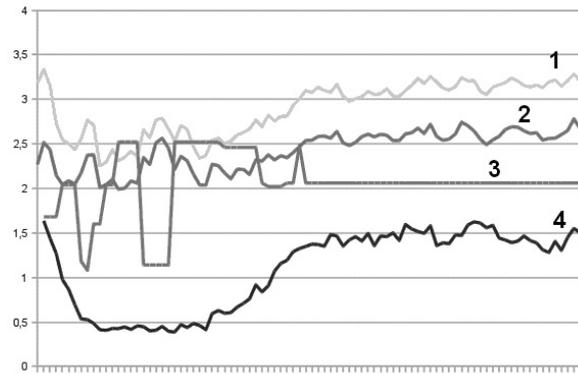


Рис. 4. Показания первой популяции с симметричной функцией приспособленности

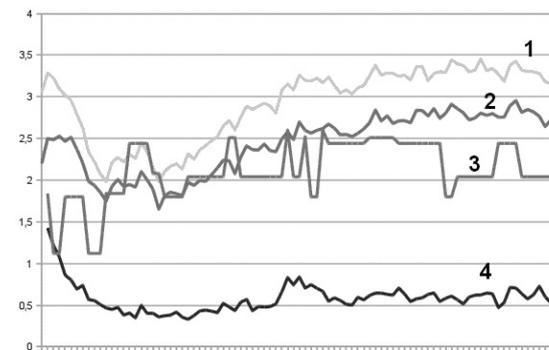


Рис. 5. Показания второй популяции с симметричной функцией приспособленности

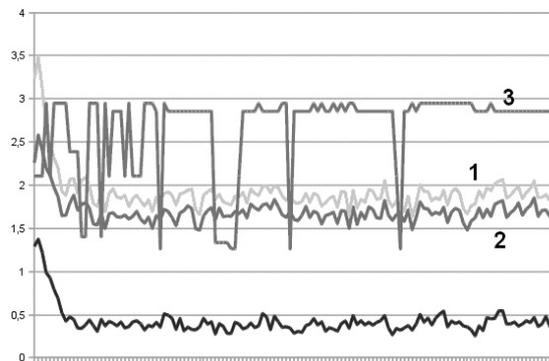


Рис. 6. Показания популяции с «прямой» функцией приспособленности

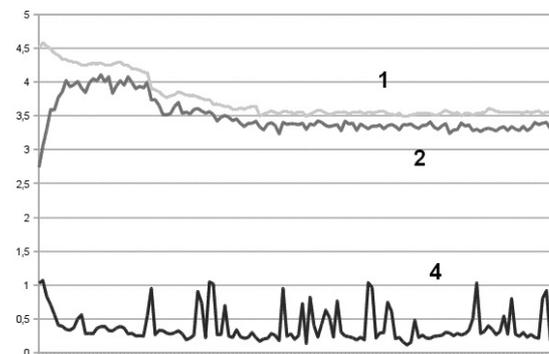


Рис. 7. Показания популяции с «обратной» функцией приспособленности

Заключение

Таким образом, применение модифицированного генетического алгоритма к задаче об «итерированной дилемме узника» позволило получить особей, способных успешно соревноваться не только со своими собратьями, но и с неизвестными им противниками, что было показано на примере стратегии «Око за око». Исходя из этих результатов, можно сделать вывод о том, что предложенный метод является эффективным решением проблемы, описанной в начале статьи.

Литература

1. Поликарпова Н.И., Точилин В.Н., Шальто А.А. Разработка библиотеки для генерации автоматов методом генетического программирования // Сборник докладов X международной конференции по мягким вычислениям и измерениям. – СПб: СПбГЭТУ, 2007. – Т. 2. – С. 84–87.
2. Bäck T. Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. – Oxford University Press, 1996.
3. Axelrod R., Hamilton W.D. The evolution of cooperation // Science. – 1981. – V. 211. – № 4489. – P. 1390.
4. Axelrod R. The evolution of strategies in the iterated prisoner's dilemma. – Cambridge university press, 1987. – P. 1–16.

Кулев Владимир Анатольевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, me@lightoze.net

УДК 004.4'242

**ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО ПОДХОДА ДЛЯ ГЕНЕРАЦИИ
КЛЕТОЧНЫХ АВТОМАТОВ**

А.В. Тихомиров, А.А. Шальто

Рассматривается метод генерации произвольных клеточных автоматов на основе тестовых наборов при помощи генетических алгоритмов. Описаны основные проблемы при использовании стандартного генетического алгоритма для решения поставленной задачи. Предложены модифицированные генетические операторы для устранения данных недостатков. Произведена апробация на нескольких обучающих примерах.

Ключевые слова: клеточные автоматы, генетические алгоритмы.

Введение

В настоящее время широко распространено использование клеточных автоматов для симуляции многих физических процессов, таких как диффузия энергии, разнообразные химические реакции, рост кристаллов и т.д. [1–4]. Однако использование клеточных автоматов затрудняется тогда, когда физическая система известна, а описывающий ее клеточный автомат – нет, или построение его обычными эвристическими методами затруднительно, так как автомат может иметь большое число состояний, переходов, условий и действий на переход [5].

Цель настоящей работы – устранить этот недостаток, используя генетическое программирование с использованием обучающих наборов.

Постановка задачи

Задана двумерная плоскость определенной размерности с координатной сеткой, которая делит плоскость на квадраты [6, 7]. При этом задаются данные для различных временных шагов, т.е. состояние системы в начале, несколько промежуточных состояний и конечное состояние системы.

Каждая ячейка плоскости управляется одинаковым автоматом.

Цель поставленной задачи – вырастить клеточный автомат, который наиболее точно описывает заданную физическую систему.

Строение хромосомы клеточного автомата

Обычно при использовании генетического подхода используется подход, при котором данные в хромосоме хранятся в виде строки [5]. Однако в рассматриваемой задаче длина хромосомы может варьироваться в зависимости от количества переходов между состояниями. Для более удобного описания данных ген представляется не строкой, а специализированной структурой, что отличается от стандартного метода генетического программирования. При этом некоторые виды генов являются составными, т.е. содержат другие, более простые составляющие.

Количество базовых генов соответствует количеству состояний в клеточном автомате. Таким образом, базовый ген под названием «StateGene» описывает определенное состояние автомата. «StateGene» состоит из генов переходов клеточного автомата, которые называются «TransitionGene» (рис. 1).

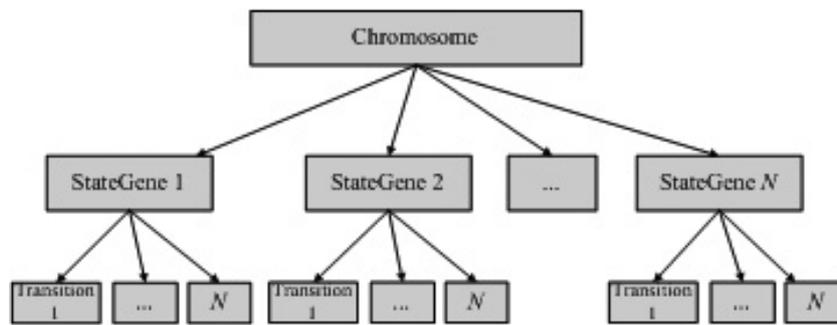


Рис. 1. Структура хромосомы

Каждый ген перехода состоит из списка генов условий на переход и генов действий, которые будут осуществляться в случае этого перехода. В случае отсутствия гена перехода из состояния $N1$ в $N2$, это означает, что в клеточном автомате, описываемом заданным геном нет перехода из состояния $N1$ в $N2$ (рис. 2). Иначе говоря, длина хромосомы зависит от количества состояний в клеточном автомате, который описывает данная хромосома.

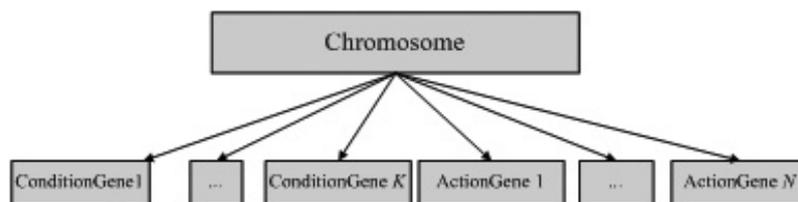


Рис. 2. Структура гена переходов

Условия переходов бывают двух видов:

1. Условия наличия рядом с текущей клеткой определенного числа N клеток с определенными состояниями. Этот вид условия задается в виде специальной строки;
2. Условия соотношения данных между текущей клеткой и определенной из окружающих клеток. Этот вид условия задается при помощи специального дерева, в верхнем узле которого находится операция сравнения, а ниже по дереву – операции вычисления условия.

В случае отсутствия генов условий подразумевается, что условие на данный переход всегда является верным.

Начальное поколение состоит из фиксированного числа случайно генерированных автоматов. Все автоматы в поколении имеют одинаковое наперед заданное количество состояний, но количество переходов между состояниями может варьироваться.

Проблема вырождения популяции

Одной из проблем в применении генетического подхода для генерации клеточных автоматов стало вырождение популяции, т.е. такой ситуации, когда выделяется один-единственный генотип, который представляет собой локальный максимум, а затем все элементы популяции проигрывают ему отбор, и вся популяция «забывается» копиями этой особи.

Из-за схожего строения хромосом операция скрещивания становится малоэффективной, и вероятность получить хромосому, которая лучше решает поставленную задачу, сильно падает. Следовательно, для сокращения времени работы алгоритма надо как-то бороться с этой проблемой. В данной работе рассматриваются несколько способов преодолеть этот недостаток:

- модифицированный алгоритм скрещивания хромосом;
- алгоритм каскадной фитнес-функции;
- специальный алгоритм отбора нового поколения.

Все эти модификации позволяют улучшить «сходимость» генетического подхода при генерации клеточных автоматов.

Операции скрещивания

В данной работе операция скрещивания была реализована в двух различных вариантах, чтобы избежать вырождения популяции.

Для решения этой проблемы при генерации начального поколения создается отдельная пустая популяция «Graveyard», которая не участвует в генетических операциях. В эту популяцию добавляются

особи, которые не попали в следующее поколение эволюции, при этом размер контролируется путем выбора произвольных n особей. Такой подход позволяет всегда иметь разнообразие в особях.

Операции скрещивания выполняется в двух вариантах (рис. 3):

- стандартная операция скрещивания, действие которой заключается в обмене генами между особями в текущей популяции;
- операция скрещивания между особью из текущей популяции и из популяции «Graveyard», причем право на участие в отборе имеет особь, основа которой была в популяции, участвующей в эволюции.

Каждая операция скрещивания состоит из двух вариантов:

1. Обмен несколькими «State» генами между особями, т.е. обмен состояниями;
2. Обмен одним из составных генов между особями, т.е. обмен переходами, условиями и действиями на переход. Причем возможен обмен генами, которые находятся в разных «State» генах.

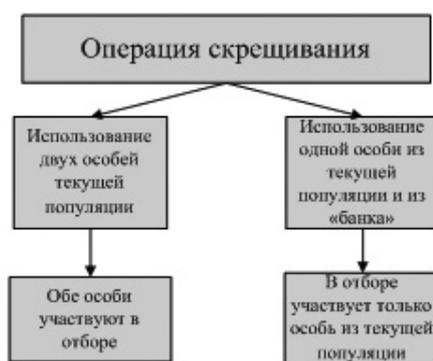


Рис. 3. Схема операции скрещивания

Если гены для скрещивания у особей являются идентичными, то операция скрещивания не совершается.

Для оптимизации операции скрещивания применяется анализ хромосом на совпадающие гены, т.е. из каждой пары хромосом, участвующих в скрещивании, выделяется список уникальных генов. Именно среди этих генов и производится поиск для обмена. Такая операция позволяет совершать меньшее число мутаций и заранее исключать мутации, которые приведут к хромосомам, которые уже есть в популяции.

Операция мутации

Операция мутации состоит в том, что происходит изменение в произвольном гене копии текущей особи.

При мутации случайно выбирается один из четырех равновероятных вариантов:

1. Добавление нового перехода или полное замещение уже существующего;
2. Удаление уже существующего перехода;
3. Мутация существующего перехода

Мутация действий

- изменение действия на переходе;
- добавление случайного действия;
- удаление уже существующего действия.

Мутация условий

- изменения условия на переходе;
- добавление случайного условия;
- удаление уже существующего условия;

4. Обмен двух переходов местами.

Фитнесс-функция

Функция приспособленности особи, т.е. клеточного автомата, состоит из нескольких значений. Первое значение характеризует автомат с точки зрения решения поставленной задачи, при моделировании сравниваются эталонные и полученные результаты для каждого шага моделирования. Второе значение характеризует автомат с точки зрения избыточности, в нем хранится «длина автомата» и количество неиспользуемых переходов и условий. Функция приспособленности особи вычисляется при моделировании и запоминается, таким образом, для каждой особи она вычисляется один раз. Для борьбы с вырождением популяции была применена каскадная фитнес-функция. Основная идея заключается в том, что

на начальных этапах выращивания клеточных автоматов функция приспособленности учитывает только несколько первых временных срезов входных данных. Когда в популяции есть клеточные автоматы, которые в какой-то степени удовлетворяют текущей фитнес-функции, то происходит смена на функцию с учетом большего количества слоев.

Ниже рассмотрены плюсы и минусы при использовании каскадной фитнес-функции:

Минусы:

- дополнительные затраты при смене функции;
- сложность сравнения эффективности особей (особенно это проявляется при сравнении особей из разных «островов»).

Плюсы:

- прирост производительности (на ранних этапах уменьшается количество этапов проверки клеточного автомата);
- после каждого этапа полученный автомат частично решает поставленную задачу.

Прирост скорости генерации (количества поколений, необходимых для получения решения) сильно зависит от начальных настроек при этом подходе.

Отбор нового поколения

В результате операций скрещивания и мутации возможно получение уже содержащихся хромосом, что приводит к вырождению популяции. Для решения этой проблемы авторами предложена дополнительная операция, которая удаляет из претендентов на попадание в новый этап эволюции определенные клеточные автоматы.

В данной работе были рассмотрены два варианта:

1. Одинаковые клеточные автоматы. Для этого производится приведение всех клеточных автоматов к единой форме;
2. Клеточные автоматы с одинаковым значением фитнес-функции.

Чтобы избежать локальных минимумов функции приспособленности и вырождения популяции, применяются два алгоритма отбора в новое поколение.

1. Происходит отбор первых N особей с лучшей функцией приспособленностью. Все они попадают в новое поколение.
2. Если на протяжении достаточно большого числа поколений не происходит улучшения значения функции приспособленности, то при обработке текущего поколения отбрасываются все особи, кроме нескольких, которые имеют наилучшее значение, т.е. наиболее приспособленных. Оставшееся место в новом поколении занимают особи, полученные из особей текущего поколения путем скрещивания и мутаций. Также изменяется механизм мутации поколения. Мутация может происходить не в единственном гене хромосомы, как происходит при обычных условиях, а во всех генах хромосомы с некоторой вероятностью.

Таким образом, в случае успешного процесса генерации используется первый алгоритм, а в случае долгих «простоев», когда генетический алгоритм не может сгенерировать хромосому, которая лучше решает поставленную задачу, в работу вступает второй алгоритм отбора. Он благодаря большому числу мутаций и скрещиваний имеет больше шансов выйти из «простоя».

Также стоит отметить, что при отборе первых N особей при одинаковой функции приспособленности выбираются более молодые особи.

Полученные результаты

Для тестового примера была поставлена задача: вырастить клеточный автомат, зная только состояния клеток на каждом конкретном шаге расчетов. Эталонный клеточный автомат, сделанный для проверки, содержит 5 состояний, 8 переходов и 8 условий на переходах. Система в среднем выращивает автомат, который является эквивалентным искомому за 700–1000 поколений, при этом среднее количество перебранных клеточных автоматов составляет 30000–45000.

Для ускорения процесса генерации было сделано допущение, что автоматы, у которых из какого-либо состояния истинно условие у нескольких переходов, являются корректными. На них было наложено условие: при нескольких вариантах перехода переход осуществляется в состояние с меньшим индексом. После упрощения полученного клеточного автомата и сравнения его с эталоном было установлено, что эти автоматы являются идентичными.

Как видно из примера, получившийся клеточный автомат содержит избыточные переходы, по которым никогда не происходит перехода. Однако при исследовании было установлено, что если задавать условие минимизации в генетических операторах и при выборе нового поколения, то время работы системы сильно увеличивается.

Быстродействие предложенного способа сильно зависит от начальных настроек, таких как порции между генетическими операциями, величина начальной популяции и многое другое.

Заключение

В работе предложен метод генерации клеточных автоматов произвольного количества состояний. Он позволяет автоматически получать клеточные автоматы, которые довольно точно описывают данные о физической системе на входе. Также этот метод позволяет получать клеточные автоматы, которые получить эвристическими методами затруднительно.

Литература

1. Тоффли Т., Марголус Н. Машины клеточных автоматов. – М.: Мир, 1991.
2. Frish U. Lattice gas hydrodynamics in two and three dimensions // Complex Systems. – 1987. – V. 1. – P. 649–707.
3. Wolfram S. Cellular automation Fluids // J.Stat.Phys. – 1986. – V. 45. – P. 471–526.
4. Фон Нейман Дж. Теория самовоспроизводящихся автоматов: Пер. с англ. – М.: Мир, 1971.
5. Царев Ф.Н., Шалыто А.А. Применение генетического программирования для генерации автомата в задаче об «Умном муравье» // Сборник трудов IV-ой Международной научно-практической конференции. – М.: Физматлит, 2007. – Т. 2. – С. 590–597.
6. Наумов Л.А. Метод введения обобщенных координат и инструментальное средство для автоматизации проектирования программного обеспечения вычислительных экспериментов с использованием клеточных автоматов. Дис...канд. техн. наук: 05.13.12. – СПб, 2007. – 283 с.
7. Скаков П.С. Классификация поведения одномерных клеточных автоматов [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/papers/_skakov_master.pdf, своб.
8. Wolfram S. A New Kind of Science. – Wolfram Media, Inc., 2002.

- Тихомиров Андрей Владимирович* – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, and.tikhomirov@gmail.com
- Шалыто Анатолий Абрамович* – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, зав. кафедрой, shalyto@mail.ifmo.ru

УДК 004.4*242

**ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ К ГЕНЕРАЦИИ ТЕСТОВ
ДЛЯ АВТОМАТНЫХ ПРОГРАММ**

А.Ю. Законов, А.А. Шалыто

Описан подход к автоматизации тестирования автоматных программ. Для формализации требований спецификации к модели и объектам управления предлагается использовать контракты. Тест описывается как последовательность переходов в модели. Для автоматизации процесса создания кода теста предложен генетический алгоритм, который позволяет находить значения переменных, удовлетворяющие условиям на переходах.

Ключевые слова: тестирование, автоматное программирование, контракты, генетические алгоритмы.

Введение

Автоматная программа состоит из конечных автоматов и набора объектов управления, с которыми взаимодействуют автоматы [1]. Наиболее распространенным способом проверки автоматных программ является Model Checking [2], так как для этого класса программ уровень автоматизации процесса верификации может быть повышен по сравнению с традиционными программами. Это объясняется тем, что в автоматных программах поведенческая модель задается априори, а не строится по программе, как это делается обычно. Однако проверка моделей позволяет верифицировать только автоматы, но не систему в целом. Поведение объектов управления и их взаимодействие с автоматами не проверяются при указанном подходе. Таким образом, в автоматной программе могут остаться невыявленные ошибки, даже если система автоматов была успешно верифицирована относительно данной спецификации.

В данной работе используется тестирование для проверки автоматных программ в целом. Тестирование является трудоемкой и ресурсоемкой задачей. Около половины всего времени разработки проекта часто тратится на тестирование [3]. В последнее время многие исследования посвящены теме автоматизации процессов создания и запуска тестов [4]. Несмотря на то, что успешное тестирование не гарантирует отсутствие ошибок в программе, большой набор тестов может существенно помочь в обнаружении ошибок в логике работы программы и в ее реализации. Это может обеспечить повышение вероятности того, что программа будет успешно решать поставленные задачи.

В данной работе предлагается подход к тестированию автоматных программ и способ автоматизации процесса создания тестов на основе использования генетических алгоритмов.

В предлагаемом подходе тестирование применяется для проверки соответствия спецификации системы ее реализации. Спецификация, заданная на естественном языке, пригодна только при ручном тестировании. Для автоматизации процесса тестирования спецификация должна быть записана на формальном языке. Поэтому предлагается внести максимально возможное число требований спецификации в описание автомата. Это позволит расширить программу так, что она будет содержать в себе требования для проверки. Для описания моделей автоматных программ естественно использование конечных автоматов. Однако в реактивных системах в конечных автоматах в качестве входных воздействий обычно применяются события. Для моделирования поведения сложных систем, взаимодействующих со средой, в работе предлагается использовать расширенные конечные автоматы, которые позволяют использовать переменные в модели и с их помощью задавать охраняемые условия на переходах.

Зависимость поведения системы от значений этих переменных позволяет описать сложную логику работы и тем самым внести значительную часть спецификации в модель. В работе [5] предлагается использовать контракты [6] для внесения дополнительных требований спецификации в модель автоматных программ. Наличие исполняемой спецификации внутри автоматной модели позволяет при запуске программы автоматически проверять выполнение указанных условий. Более того, формальное описание условий позволит реализовать автоматическую генерацию тестов и искать значения, которые помогут выявить несоответствия реализации и спецификации.

Создание теста для автоматной программы

Большая часть автоматных программ разрабатывается для взаимодействия с внешней средой: программа получает события и входные переменные, а автомат реагирует на эти воздействия и реализует выходные воздействия. В автоматных системах для реализации выходных воздействий используются объекты управления. Они получают из среды события и входные воздействия, которые используются в автомате в охраняемых условиях на переходах или передаются как аргументы вызовов методов других объектов управления. В расширенном конечном автомате все эти значения можно представить переменными. Таким образом, переменные, использованные в модели, разделяются на два типа: внутренние, заданные и определенные внутри модели, и внешние, полученные из среды от объектов управления. Во время тестирования значения внешних переменных необходимо генерировать вместе с кодом теста.

Учитывая приведенное выше описание модели и ее спецификации, определим понятие теста для автоматной программы. В случае традиционного подхода к разработке программного обеспечения создание тестов осуществляется путем представления требований спецификации в виде программного кода. В предложенном подходе тест определяется не в виде кода, а в терминах автоматной модели и требований спецификации. Тест автоматной программы – последовательность переходов автомата. Тест также содержит требования спецификации к задействованным переходам, состояниям и объектам управления. Такое описание теста значительно понятнее традиционного описания теста, так как не требует изучения исходного кода и позволяет описать важные тестовые сценарии уже на этапе написания спецификации, а не при написании кода программы. Это помогает преодолеть семантический разрыв между реализацией программы и ее спецификацией.

Предложенное определение теста удобно для его описания, но не пригодно для автоматического создания кода этого теста. Чтобы автомат выполнил описанный в тесте сценарий, необходимо определить последовательность событий и набор значений внешних переменных, которые приводят к выполнению автоматом заданной последовательности переходов. Для поиска значений внешних переменных в данной работе предлагается использовать генетические алгоритмы.

Существующие инструменты для тестирования моделей

При тестировании моделей, построенных с использованием расширенных конечных автоматов, выделяется две подзадачи: выбор или генерация выполнимых путей (feasible paths) для тестирования и подбор входных значений для выполнения этих путей.

Выбор выполнимых путей для тестирования изучен во многих работах [7, 8]. Также пути для тестирования можно задавать вручную, выбирая наиболее интересные сценарии для проверки. Для путей, полученных автоматически или вручную, необходимо найти значения переменных для их прохождения с учетом всех охраняемых условий расширенного конечного автомата. Задача поиска этих значений менее изучена, но не менее актуальна, так как без этих значений невозможно сгенерировать код теста, который проверит заданную последовательность действий.

Существует ряд инструментов для работы с моделями и создания тестов на их основе. В работе [9] рассмотрены инструменты для тестирования на основе моделей программ, написанных на языке C#. При этом рассматривается инструмент NModel, который позволяет создавать модели на текстовом языке, автоматически строить по модели конечные автоматы, описывающие переходы в системе, и генерировать тестовые сценарии для этой системы.

В работе [10] приведен обзор инструментов для тестирования на основе моделей и предложен инструмент ModelJUnit, который позволяет описывать расширенные конечные автоматы как классы на

языке Java и автоматически генерировать для них тесты. Инструмент Tigris MBT позволяет графически описывать расширенные конечные автоматы и, используя случайные значения, пытается найти последовательность событий и набор переменных для полного покрытия переходов или для достижения определенного состояния.

Институт системного программирования РАН предлагает семейство инструментов разработки тестов на основе моделей UniTesK [11], которые предлагают использовать «спецификации ограничений» для построения модели системы, используя ее интерфейсы. Для написания тестов используется SeC – спецификационное расширение языка C.

Перечисленные инструменты не решают проблему тестирования автоматных программ. Во-первых, эти инструменты предполагают ручное создание требований для проверки отдельно от модели, что делает процесс разработки более сложным. Во-вторых, принципиальное отличие автоматных программ от моделей, используемых в существующих инструментах, состоит в том, что автоматная программа включает в себя, кроме автомата, также и объекты управления. Тестирование только автомата не позволяет найти ошибки в программе в целом.

Включение спецификации в модель

Спецификация, записанная на естественном языке, пригодна только для ручного тестирования. Предлагаемый подход развивает идею, изложенную в работе [5], и предлагает использовать контракты для записи требований не только к модели, но и к объектам управления. Наличие исполняемой спецификации внутри автоматной модели позволяет при запуске программы автоматически проверять выполнение описанных условий как моделью, так и объектами управления.

Объекты управления, с которыми взаимодействует автомат, имеют спецификацию на входные воздействия, результаты их работы и особенности взаимодействия с автоматом. Все эти требования спецификации должны быть выполнены во время корректной работы программы. В предложенном подходе спецификация объектов управления включается в описание автоматной модели. Благодаря этому особенности реализации конкретных объектов управления не важны на этапе создания тестов, так как достаточно проверять, что на вход им подаются корректные значения и что автомат корректно обрабатывает любые значения, полученные из объектов управления, которые не противоречат их спецификации.

В предложенном подходе для интеграции требований спецификации в описание автоматной модели используются контракты, записанные на языке JML [6]. В случае автоматной модели предлагается задавать переходам пред- и постусловия, а для состояний записывать инварианты:

- предусловия перехода определяют ожидания системы относительно значений переменных при вызове методов объектов управления, которые используются на переходе;
- постусловия перехода задают требования к значениям переменных модели при завершении вызова методов объектов управления;
- инварианты, заданные для состояния, содержат требования, которые должны выполняться на протяжении всего времени пребывания системы в указанном состоянии.

Создание сценариев для тестирования и поиск значений входных переменных

Проанализировав спецификацию, заданную на естественном языке, необходимо выбрать сценарии работы, интересные для тестирования. Выбранные сценарии далее записываются в автоматных терминах как последовательность переходов автомата. Такое представление теста должно быть интуитивно понятно как автору спецификации, работающему с требованиями, записанными на естественном языке, так и разработчику системы, работающему с автоматной моделью и кодом объектов управления.

Существует ряд исследований [7], в которых изучается задача генерации пути в расширенном конечном автомате, который обеспечивал бы максимальное покрытие переходов и состояний. Описанные технологии автоматического создания сценариев можно успешно сочетать с подходом, представленным в настоящей работе, для создания эффективных наборов тестов.

Автомат реагирует на события и выполняет переходы в зависимости от значений переменных автоматной модели, которые используются в охранных условиях на переходах. Таким образом, возникает задача поиска последовательности событий и набора значений внешних переменных, соответствующих заданной последовательности переходов в модели. В данной работе предложен алгоритм для автоматизации решения этой задачи.

Последовательность событий однозначно определяется по последовательности переходов. Сложнее подобрать значения переменных – они должны удовлетворять ряду требований. Во-первых, охранные условия на всех переходах в описанном пути должны быть выполнены. Во-вторых, все требования спецификации объектов управления должны выполняться, так как при реальном использовании значения этих переменных будут приходиться из объектов управления с данными спецификациями.

Применение генетического алгоритма для поиска значений переменных

Генетические алгоритмы считаются успешными для решения оптимизационных задач, в том числе и задач генерации тестов [4, 7, 12]. Задачу поиска набора значений, при котором будет выполнен заданный путь в расширенном конечном автомате, можно свести к задаче оптимизации.

Набор внешних переменных, задействованных на выбранной последовательности переходов, можно рассматривать как вектор значений $\langle x_1, x_2, \dots, x_n \rangle$, где x_i – значение внешней переменной, а n – число внешних переменных для этого пути. Для генетического алгоритма набор внешних переменных является хромосомой. Определим для него функцию приспособленности – функцию, которая оценивает, насколько хромосома решает задачу, и позволяет выбирать лучшие решения из всего поколения.

В рассматриваемой задаче функция приспособленности на вход принимает вектор значений и выдает число, характеризующее приспособленность этого вектора значений для выполнения заданного пути. Чем меньше значение этого числа, тем больше подходит данный вектор значений. Если же функция приспособленности равна нулю, то требуемый путь выполняется. Таким образом, задача поиска подходящего набора значений внешних переменных сводится к задаче оптимизации, где требуется найти вектор, которому соответствует минимальное значение функции приспособленности.

Один ген хромосомы – это значение одной из внешних переменных для заданного пути. В работе используется одноточечное (классическое) скрещивание (one-point crossover), в котором две хромосомы разрезаются один раз в соответствующей точке и производится обмен полученными частями.

Оператор мутации (mutation operator) необходим для защиты генетического алгоритма от преждевременной сходимости. В данной работе мутация производит замену произвольного гена на случайно выбранное число из области допустимых значений.

Функция приспособленности

Функция приспособленности должна удовлетворять следующему условию: чем лучше особь подходит для поставленной задачи, тем меньше значение функции. Таким образом, проблема сводится к задаче оптимизации – найти решение с нулевым значением функции приспособленности.

Однозначного способа определения функции приспособленности не существует. В работе [12], описывающей применение генетических алгоритмов для тестирования структурных программ, для определения приспособленности хромосом используется такой критерий, как расстояние до условия (branch distance). Расстояние до условия позволяет оценить, насколько близка была данная хромосома к выполнению конкретного условия, которое на практике не было выполнено. Например, для условия $A=B$ расстояние до условия будет вычисляться по формуле $|A-B|$. Чем меньше значение $|A-B|$, тем ближе значение A к B и тем ближе хромосома к тому, чтобы это условие было выполнено. Если условие выполнено, то расстояние до условия равно нулю.

Для вычисления приспособленности хромосомы относительно корректного выполнения заданного пути в автоматной модели следует учитывать два типа условий:

1. охранные условия на переходах автомата;
2. требования спецификации методов объектов управления, которые задействованы на переходах.

Все эти условия обязательны для выполнения. Хромосомы, которые нарушают любое из указанных условий, не подходят для создания тестов, так как система в соответствии со своей спецификацией не должна поддерживать работу с этими значениями. В этом случае функция приспособленности должна оценить, насколько близка хромосома к тому, чтобы выполнить нарушенные условия. Это делается при помощи вычисления расстояний до условий.

Для последовательности переходов может быть задано большое число условий, поэтому расстояние до условия всего пути необходимо вычислять по отдельности, рассматривая условия на каждом переходе этого пути. Каждый переход описывается набором параметров:

- событие, по которому этот переход может произойти;
- охранное условие, которое должно быть выполнено для совершения перехода;
- действия на переходе: вызов методов объектов управления, получение значений внешних переменных из среды или изменение значений переменных модели;
- предусловия перехода;
- постусловия перехода.

Следовательно, даже в рамках одного перехода может быть задействовано большое число условий. Для более точного вычисления расстояния до условия перехода в работе каждый переход разбивается на несколько меньших шагов:

1. получение события, поиск возможных переходов и проверка их охранных условий;
2. проверка предусловий перехода, выполнение перехода и заданных действий на переходе;
3. проверка постусловий перехода.

При выполнении каждого из этих шагов могут быть обнаружены ошибки. Переход считается выполненным успешно, если все три шага выполнены успешно. Таким образом, сценарий теста разбивается на переходы, а каждый переход разбивается на три шага. После этого исходная последовательность переходов рассматривается как последовательность шагов. Оценка приспособленности всей последовательности шагов вычисляется как сумма оценок для каждого шага по отдельности. Каждый шаг оценивается по формуле вычисления расстояния для условия.

Следует заметить, что шаги выполняются последовательно и что выполнение шагов в начале пути важнее, чем в конце пути. Например, если выполнены условия всех шагов, кроме первого, то сумма расстояний до условия будет маленькой, так как для всех шагов, кроме первого, это значение будет равно нулю. На практике эта хромосома не позволяет пройти ни одного шага, так как для того, чтобы успешно пройти второй шаг, необходимо выполнить все условия на первом шаге. В предложенном подходе расстояния до условия шагов суммируются с учетом местоположения этих шагов в пути – используется взвешенная сумма. Если для одного шага задано несколько условий, то расстояние до условия этого шага вычисляется как сумма расстояний до условия каждого из этих условий.

Разработанные и используемые инструментальные средства

Для создания автоматной модели предлагается использовать инструмент уEd [13], который позволяет удобно графически описывать расширенные конечные автоматы, содержащие переменные и охраняемые условия. Возможность задания дополнительных меток к переходам и состояниям можно использовать для записи JML-контрактов.

Для поиска значений внешних переменных для заданного пути разработан инструмент GenValueFinder, который принимает на вход последовательность переходов автоматной модели и, используя описанный генетический алгоритм, осуществляет поиск значений внешних переменных для прохождения этого пути. Генерация кода теста, пригодного для запуска, происходит при помощи разработанного инструмента TestGenerator. На вход подаются файлы, полученные в результате работы программы GenValueFinder.

Проверка выполнения контрактов выполняется при помощи существующего инструмента JML Runtime Assertion Checker [14], который входит в стандартный набор утилит при установке JML. Инструмент в режиме реального времени проверяет, что все контракты, добавленные в Java-код в виде аннотаций, выполнены.

Пример

Проиллюстрируем предложенный подход на примере разработки банкомата, соответствующего приведенной спецификации на естественном языке.

1. Банкомат позволяет пользователям снимать деньги с определенного счета.
2. Сначала на счету находится сумма от 0 до 100 000.
3. Пользователь может снимать деньги произвольное число раз, пока на счету неотрицательная сумма.
4. Ввод суммы для снятия происходит с клавиатуры. Пользователь может ввести число от 1000 до 15 000.
5. В день со счета можно снять не более 50 000.

Спецификация в таком виде удобна для разработчика и заказчика, но такую спецификацию можно проверять и тестировать только вручную.

Выделение объектов управления и построение автоматной модели. Для формализации спецификации необходимо выделить спецификацию объектов управления и требования к логике работы системы, которые будут описаны при помощи автоматов. Как видно из текста спецификации, в системе задействованы два объекта управления:

1. *Счет* отвечает за взаимодействие с банковским счетом (вернуть число денег на счету, снять деньги со счета);
2. *Клавиатура* отвечает за взаимодействие с пользователем (ввод суммы для снятия на клавиатуре устройства).

Требования, которые описывают логику работы системы и также должны быть учтены при построении автоматной модели, состоят в следующем:

- Счет: вначале на счету находится сумма от 0 до 100 000.
- Клавиатура: пользователь может ввести число от 1000 до 15 000.

В данной модели будут использованы две внешние переменные:

1. *ext_sum* – количество денег на счету.
2. *ext_x* – сумма для снятия во время транзакции, которую вводит пользователь.

Внутренняя переменная *today* используется для суммирования снятых за сеанс работы денег. Добавление этой переменной позволяет записать условие, которое ограничивает число снятых денег за один сеанс.

На рисунке приведена модель, представленная в виде расширенного конечного автомата и требований, записанных при помощи контрактов.

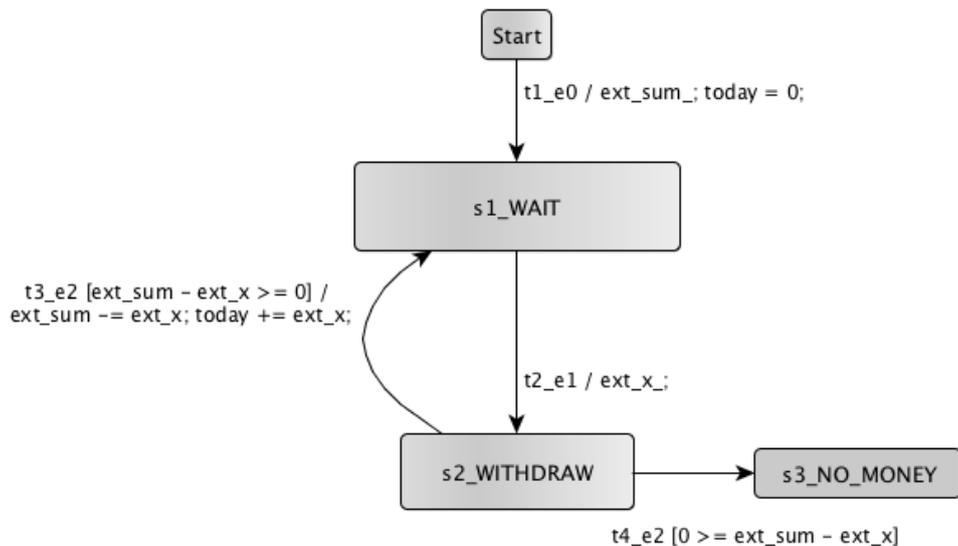


Рисунок. Расширенный конечный автомат, содержащий контракты

В модель включены требования спецификации к объектам управления. Клавиатура: @ensures $ext_x \geq 1000 \ \&\& \ ext_x \leq 15000$. Счет: @ensures $ext_sum \geq 0 \ \&\& \ ext_sum \leq 100000$. Требование спецификации, ограничивающее число снятых денег за сеанс работы, добавлено как контракт к состоянию: @invariant $today \leq 50000$.

Пример тестовых сценариев и создания тестов. Проанализировав текстовую спецификацию системы, предложим сценарий использования и создадим тест на его основе. Сначала запишем сценарий на естественном языке: три раза снимаются деньги со счета и на счету заканчиваются средства на четвертой попытке.

Следующий шаг – формальная запись сценария в терминах автоматной модели. Тестовый сценарий, записанный как последовательность переходов, имеет вид

– path.txt: t1, t2, t3, t2, t3, t2, t3, t2, t4.

В тестовом сценарии задействовано пять внешних переменных: ext_sum – изначальная сумма на счету; ext_x1 – пользователь ввел первый раз; ... ext_x4 – пользователь ввел четвертый раз.

Для поиска подходящих значений использован разработанный инструмент ./GenValueFinder model.xml path.txt. Значения найдены за 10 секунд:

– файл events.txt: e0 e1 e2 e1 e2 e1 e2 e1 e2;
 – файл variables.txt: 33177 13115 14485 4382 8513.

Код теста генерируется запуском инструмента TestGenerator. Далее необходимо скомпилировать тест командой jmlc и запустить при помощи команды jmlrac. Запуск теста выдает следующий результат: Exception in thread "main" JMLInvariantError: by method GeneratedTest.transition5 File "GeneratedTest.java", line 34

Данное описание ошибки позволяет понять, что условие invariant было нарушено после выполнения пятого перехода. Таким образом, при найденных значениях внешних переменных на тестируемом сценарии обнаружено несоответствие спецификации и реализации.

Заключение

Использование предложенного подхода к тестированию автоматных программ позволит повысить качество разрабатываемых систем и проверять соответствие реализации системы заданной спецификации. В результате работы получены следующие результаты:

- предложен способ проверки автоматных программ в целом;
- разработан метод нахождения входных параметров для выполнения заданного сценария в автоматной модели при помощи использования генетических алгоритмов;
- разработан инструмент для автоматизации предложенного подхода.

Исследование выполнено по Федеральной целевой программе «Научные и научно-педагогические кадры инновационной России на 2009–2013 годы» в рамках государственного контракта П2373 от 18 ноября 2009 года.

Литература

1. Поликарпова Н.И., Шальто А.А. Автоматное программирование. – СПб: Питер, 2010. – 176 с.
2. Кларк Э.М., Грамберг О., Пелед Д. Верификация моделей программ. Model Checking. – М.: МЦНМО, 2002. – 416 с.
3. Myers G. The Art of Software Testing. – John Wiley & Son. Inc, 2004.
4. McMinn P. Search-based software test data generation: a survey: Research Articles // Software Testing, Verification & Reliability. – 2004. – № 14 (2). – P. 105–156.
5. Степанов О.Г. Методы реализации автоматных объектно-ориентированных программ. Диссертация на соискание ученой степени кандидата технических наук. – СПб: СПбГУ ИТМО, 2009. – Режим доступа: http://is.ifmo.ru/disser/stepanov_disser.pdf, своб.
6. Meyer B. Applying design by contract // Computer. – 1992. – 25(10). – P. 40–51.
7. Kalaji A.S., Hierons R.M. and S. Swift. Generating Feasible Transition Paths for Testing from an Extended Finite State Machine (EFSM) // Software Testing, Verification, and Validation (ICST). 2nd International IEEE Conference. – 2009. Denver, Colorado: IEEE.
8. Duale Y., Ümit Uyar M. A Method Enabling Feasible Conformance Test Sequence Generation for EFSM Models // IEEE Transactions on Computers. – 2004. – Vol. 53. – № 5. – P.614–627.
9. Jacky J., Veanes M., Campbell C., Schulte W. Model-Based Software Testing and Analysis with C#. – Cambridge University Press, 2008.
10. Mark U., Legeard B. Practical Model-Based Testing: A Tools Approach. – Morgan–Kaufmann, 2007.
11. Bourdonov I., Kossatchev A., Kuliain V., Petrenko A. UniTesK Test Suite Architecture // Proc. of FME 2002. LNCS 2391. – Springer-Verlag, 2002. – P. 77–88.
12. Wegener J., Buhr K., Pohlheim H. Automatic test data generation for structural testing of embedded software systems by evolutionary testing // Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002). – NY. – 2002. – P. 1233–1240.
13. Веб-сайт yEd – Graph Editor [Электронный ресурс]. – Режим доступа: www.yworks.com/en/products_yed_about.html, своб.
14. Cheon Y., Leavens G.T. A Runtime Assertion Checker for the Java Modeling Language (JML) // Proceedings of the SERP '02, Las Vegas, Nevada, USA. – CSREA Press, June 2002. – P. 322–328.

Законов Андрей Юрьевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, andrew.zakonov@gmail.com

Шальто Анатолий Абрамович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, зав. кафедрой, shalyto@mail.ifmo.ru

УДК 004.415.53:004.832.23

ГЕНЕРАЦИЯ ТЕСТОВ ДЛЯ ОЛИМПИАДНЫХ ЗАДАЧ ПО ПРОГРАММИРОВАНИЮ С ИСПОЛЬЗОВАНИЕМ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

М.В. Буздалов

Предлагается метод автоматизированной генерации тестов для олимпиадных задач по программированию, предназначенный для выявления неэффективных решений. Этот метод основан на использовании генетических алгоритмов. Описывается использование предлагаемого метода для генерации новых тестов к олимпиадной задаче из Интернет-архива acm.timus.ru, при этом ни одно из имевшихся решений не прошло построенный набор тестов.

Ключевые слова: генетические алгоритмы, олимпиады по программированию, олимпиадные задачи, тестирование.

Введение

В мире проводится большое число олимпиад по программированию. Они способствуют выявлению талантливых программистов среди школьников и студентов. Среди них можно отметить международную студенческую олимпиаду по программированию International Collegiate Programming Contest [1], проводимую Association for Computing Machinery (далее олимпиада будет упоминаться как ACM ICPC), с развитой сетью отборочных соревнований, международную олимпиаду школьников по информатике [2], соревнования, проводимые компанией TopCoder [3], Интернет-олимпиады по информатике и программированию [4] и многие другие.

На олимпиадах по программированию предлагается решить одну или несколько задач. Формулировка задачи предполагает чтение входных данных, удовлетворяющих условию задачи, получение требуемых результатов на основе этих данных и вывод результатов в формате, указанном в условии задачи. Решением задачи является программа, написанная на одном из языков программирования (например, в соревнованиях ACM ICPC используются языки C, C++ и Java [5]).

Программа тестируется на наборе тестов, не известных участникам. На работу программы накладываются определенные ограничения, такие как максимальное время выполнения и максимальный объем используемой памяти.

Решение считается прошедшим определенный тест, если оно при работе с ним не нарушило ограничений, завершилось корректно (без ошибок времени выполнения) и его ответ признан правильным. О конкретных видах задач и ограничениях можно прочитать, например, на сайте олимпиады [5]. В статье [6] изложен процесс решения отдельно взятой задачи при одиночной работе участника, а в статье [7] описана работа в команде.

Подготовка задач для олимпиад по программированию

Проведение соревнований на высоком уровне предполагает качественную подготовку задач. Этот процесс включает в себя выбор интересных идей для будущих задач, написание условий и решений, а также составление тестов. В настоящей работе предлагается новый метод автоматизированного составления тестов против неэффективных решений – таких решений, которые всегда выдают верный ответ, но на некоторых тестах не укладываются в ограничения по времени или памяти.

Тесты для олимпиадных задач

В условии задачи на входные данные накладываются некоторые ограничения. Тем не менее, для большинства задач тестирование решения на всех возможных тестах, удовлетворяющих этим ограничениям, не представляется возможным, так как число таких тестов чрезмерно велико. В силу этого из всех возможных тестов необходимо выбрать только те, которые позволят установить, является ли некоторое решение корректным. Однако, согласно теореме Райса [8], данная задача в общем случае является алгоритмически неразрешимой. Следовательно, возможно выбрать набор тестов, который лишь с некоторой долей уверенности позволяет утверждать о корректности решения. Цель подготовки тестов состоит в том, чтобы сделать эту уверенность как можно большей.

Подготовка тестов

Подготовка тестов к олимпиадной задаче является творческим процессом. Для каждой задачи в обязательном порядке пишутся несколько решений, называемых *решениями жюри*. Среди них должны быть как корректные, так и неверные и неэффективные решения. Цель решений жюри – проверка тестов: любое корректное решение должно пройти все тесты, для любого некорректного решения должно существовать определенное ненулевое число тестов, которое оно не проходит.

Часть тестов пишется вручную. Такие тесты проверяют решения на корректность разбора случаев, встречающихся в задаче. К таким тестам относятся и так называемые «минимальные» тесты, которые проверяют корректность работы решений около нижних границ ограничений.

Тесты большого размера, как правило, генерируются согласно некоторому шаблону. Так, например, если в условии задачи фигурирует некоторый граф, то можно сгенерировать двоичное дерево, полный двудольный граф и другие виды графов [9]. Для покрытия тех ошибок, которые могут быть не найдены с помощью описанных выше тестов, создаются «случайные» тесты. В общем случае большие тесты создаются программами, написанными членами жюри.

Для некоторых задач может быть написано некорректное решение, которое не было предусмотрено жюри. В этом случае в наборе тестов может и не оказаться такого теста, на котором это решение не работает. Авторы задач стремятся предотвратить такое развитие событий, для чего реализуют эвристические решения и генерируют тесты против них. Однако не все идеи эвристических решений могут быть найдены или реализованы, а в отдельных случаях поиск тестов против них может затянуться на неопределенное время.

По указанным причинам на олимпиадах иногда встречаются задачи со «слабым» набором тестов, которые пропускают некорректные решения. Это приводит к тому, что наиболее подготовленные участники олимпиад, ищущие корректные решения, не решают такие задачи, в то время как менее подготовленные участники успешно сдают некорректные решения, что противоречит самой идее олимпиады.

Описание предлагаемого подхода

В настоящей работе рассматривается генерация тестов, на которых неэффективные решения работают как можно дольше. Качество таких тестов может характеризоваться величинами с большим диапазоном значений, что упрощает поиск требуемого теста.

При генерации теста против неэффективного решения предполагается, что такое решение уже имеется. Это допущение в некотором смысле противоречит цели получаемых тестов – выявлять различные неэффективные решения, реализуемые участниками с применением различных, зачастую непредсказуемых и нестандартных идей. Тем не менее, тест, сгенерированный против некоторого решения, обычно оказывается «фатальным» для достаточно большого множества неэффективных решений. При разумном выборе решений, против которых генерируются тесты, возможно «покрыть» почти все неэффективные решения.

При описываемом подходе тест кодируется в виде особи генетического алгоритма [10, 11]. Способ кодирования зависит от задачи, для которой генерируются тесты. Кроме этого, кодирование должно эффективно учитывать ограничения, данные в условии задачи. Для повышения эффективности подхода следует проанализировать возможные способы кодирования и выбрать тот из них, который обеспечивает большую долю потенциально эффективных тестов.

Большую трудность составляет выбор функции приспособленности. В качестве функции приспособленности нельзя выбирать время работы решения на тесте, поскольку это время может случайным образом изменяться в зависимости от загруженности различных узлов компьютера. Также измеренное время работы, как правило, пропорционально некоторому минимальному интервалу – «кванту» времени, выделяемому операционной системой для работы программы, поэтому число различных значений функции приспособленности снижается.

Выбор в качестве функции приспособленности числа выполненных инструкций кода решает описанные выше проблемы. Однако во многих случаях то, что функция приспособленности пропорциональна времени работы программы, может привести к тому, что алгоритм оптимизации сделает выбор в пользу «больших» тестов, игнорируя «качественные», иными словами, предпочтет количество качеству.

В связи с этим предлагается проанализировать решение, против которого требуется сгенерировать тест, и модифицировать его исходный код так, чтобы в процессе работы решения вычислялась та характеристическая величина, которая будет использована в качестве функции приспособленности. Эта величина может лучше выражать качество теста, что ускоряет процесс поиска.

Применение подхода к олимпиадной задаче «Ships. Version 2»

Для апробации описываемого подхода в условиях реальных олимпиадных задач была выбрана задача «Ships. Version 2». С текстом условия этой задачи можно ознакомиться на сайте *Timus Online Judge* [12], где она размещена под номером 1394 [13]. Условие задачи состоит в следующем.

Дано N ($2 \leq N \leq 99$) предметов с весами w_i , $1 \leq w_i \leq 100$. Также дано M ($2 \leq M \leq 9$) рюкзаков с вместимостями $c_j \geq 1$. Известно, что $\sum_{i=1}^N w_i = \sum_{j=1}^M c_j$. Требуется поместить все данные предметы в данные рюкзаки. Гарантируется, что искомое размещение существует. Ограничение по времени – одна секунда, ограничение по памяти – 16 Мбайт.

Из изложенного следует, что рассматриваемая задача является частным случаем задачи о мультирюкзаке [14] с дополнительными ограничениями: для всех предметов их вес равен стоимости, и все рюкзаки должны быть заполнены.

Задача о мультирюкзаке NP-полна, так как она принадлежит классу NP и к ней сводится NP-полная задача о сумме подмножеств. Кроме того, она является NP-трудной в сильном смысле [14], т.е. для нее неизвестны решения, работающие с полиномиальной оценкой от размерности задачи и ограничений на веса и стоимости предметов. Вполне естественно ожидать от рассматриваемой задачи такого же свойства. Из этого следует, что при данных ограничениях задачи маловероятно существование решения, которое укладывалось бы в ограничения по времени и памяти на всех возможных тестах. Однако существует большое число различных эвристических решений, для которых трудно составить такой тест. По состоянию на 15 июня 2009 года из 3100 посланных на проверку решений было принято 260.

Для генерации тестов к этой задаче был применен генетический алгоритм.

Представление теста в виде особи генетического алгоритма

В условии задачи сказано, что сумма весов предметов должна равняться сумме вместимостей рюкзаков, и существует способ разместить все предметы по рюкзакам. Для случайно сгенерированного теста это условие с большой вероятностью не выполняется. Целесообразно закодировать тестовые данные таким образом, чтобы эти условия выполнялись по построению.

Предлагается следующий способ построения теста по последовательности чисел. Будем считать, что последовательность состоит из целых чисел из диапазона $[0; 100]$. При этом положительным числам соответствуют предметы с весом, равным соответствующему числу. Нули же являются разделителями последовательности на непрерывные группы положительных чисел. Каждой такой группе сопоставлен рюкзак с вместимостью, равной сумме чисел этой группы. На рис. 1 проиллюстрирован пример последовательности и соответствующего ей теста.

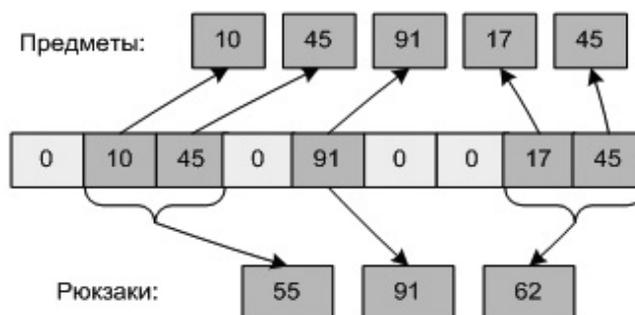


Рис. 1. Числовая последовательность и генерируемый ей тест

Любой тест, удовлетворяющий ограничениям, может быть закодирован в виде последовательности чисел длиной не более $N + M - 1$. Напротив, не любая возможная последовательность генерирует тест, удовлетворяющий ограничениям: число предметов и число рюкзаков могут быть как слишком малыми, так и слишком большими. В связи с этим предлагается обнулять значение приспособленности тех последовательностей, которые генерируют тесты, не удовлетворяющие ограничениям.

Согласно теореме о схемах [10], в классическом генетическом алгоритме, работающем со строками, «выживают» короткие по длине, малые по числу элементов схемы с большой приспособленностью. В данной же задаче после предварительного анализа было выяснено, что общую приспособленность определяют большие группы элементов последовательности.

Чтобы иметь возможность группировать блоки элементов последовательности, предлагается использовать новый вид кодирования последовательности – древовидные генераторы последовательностей (рис. 2).

Каждое поддерево генерирует некоторую последовательность, определяемую следующими правилами. Лист дерева генерирует последовательность из одного целого числа, хранящегося в нем. Внутренняя вершина дерева генерирует последовательность, образованную конкатенацией результатов генерации детей этой вершины.

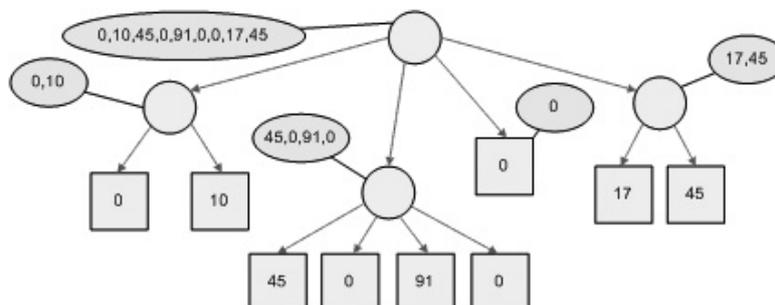


Рис. 2. Пример древовидного генератора

Древовидные генераторы напоминают деревья разбора, используемые в генетическом программировании [15], с той разницей, что результатом «выполнения» поддерева является не число, а последовательность.

Функция приспособленности

Для любого решения в качестве функции приспособленности можно выбрать число выполнений «узких мест» алгоритма. При таком подходе счетчик числа операций увеличивается внутри самого глубокого уровня вложенности некоторого числа циклов, встречающихся в программе. Итоговое значение счетчика в этом случае почти точно пропорционально времени работы программы. Однако этот общий подход работает не для всех решений.

Для рекурсивных алгоритмов решения в качестве функции приспособленности можно выбрать число вызовов рекурсивной функции (или одной из таких функций в случае, если их несколько) в течение времени работы алгоритма.

Часто также встречаются решения, запускающие один и тот же алгоритм на разных вариантах упорядочения входных данных. Так, многие эвристические решения перебирают случайные перестановки предметов и (или) рюкзаков и для каждой такой перестановки пытаются произвести поиск ответа. В таких случаях в качестве функции приспособленности разумно выбрать число таких запусков в процессе работы решения.

В некоторых случаях целесообразно использовать комбинации описанных выше подходов. Пусть F_1 – функция приспособленности, соответствующая одному подходу, F_2 – другому, F_n – n -му. Тогда функция приспособленности для комбинации подхода будет представлять собой вектор (F_1, F_2, \dots, F_n) . Значения этой функции сравниваются лексикографически.

В качестве оператора селекции используется турнирный отбор [10], в котором с вероятностью 0,9 выбирается более приспособленная особь.

Операторы скрещивания и мутации

В качестве оператора скрещивания используется стандартный для генетического программирования оператор обмена поддеревьями. Выбор поддерева описывается следующим образом: если в данный момент алгоритм рассматривает лист дерева, то он и будет выбран в качестве поддерева. Если же алгоритм рассматривает узел дерева, то с вероятностью 0,5 выбирается поддерево с корнем в этом узле, иначе равновероятно выбирается один из потомков, и процедура выбора продолжается. Оператор мутации заменяет случайным образом выбранное поддерево на сгенерированное случайным образом дерево той же величины.

Описание и результаты эксперимента

Целью эксперимента было сгенерировать такие тесты, чтобы максимально возможное число решений, прошедших уже имеющиеся тесты, не прошло хотя бы один из таких тестов. Для этого из имеющихся на сервере зачетных решений было выбрано 25 решений. Некоторые из них были выбраны для генерации тестов против них, остальные решения использовались для оценки эффективности полученных тестов. Генерация каждого теста производилась против одного решения за время от одного часа до суток.

Всего было сгенерировано 28 тестов. Три из них генерировались на основе трех решений, показавших на ранее существовавших тестах наихудшее время работы. Остальные тесты генерировались против семи других решений, против каждого из них было сгенерировано от одного до трех тестов. Решения, против которых тесты не генерировались, использовались для оценки эффективности получаемых тестов.

В результате перетестирования решений из имевшихся на момент начала тестирования принятых решений ни одно не прошло сгенерированный набор тестов. По результатам тестирования среди сгенерированных тестов было отобрано 11 лучших. Они получили номера с 48 по 58 в порядке уменьшения сложности.

На настоящее время набор тестов, в который входят и сгенерированные генетическим алгоритмом тесты, прошло всего 8 решений, что говорит о высоком качестве набора тестов.

Заключение

В работе описан метод, позволяющий автоматически генерировать тесты для олимпиадных задач по программированию с использованием генетических алгоритмов. Данный метод был применен для генерации тестов для реальной олимпиадной задачи, где показал высокое качество тестов, создаваемых с его помощью. Полученные результаты позволяют утверждать, что описанный метод является достаточно перспективным в плане его применения при подготовке тестов для задач по олимпиадному программированию в целях повышения качества олимпиад.

Литература

1. ACM International Collegiate Programming Contest [Электронный ресурс]. – Режим доступа: http://en.wikipedia.org/wiki/ACM_ICPC, свободный. Яз. англ. (дата обращения 21.09.2010).
2. International Olympiad in Informatics [Электронный ресурс]. – Режим доступа: <http://www.ioinformatics.org>, свободный. Яз. англ. (дата обращения 21.09.2010).
3. TopCoder [Электронный ресурс]. – Режим доступа: <http://www.topcoder.com/tc>, свободный. Яз. англ. (дата обращения 21.09.2010).
4. Интернет-олимпиады по информатике [Электронный ресурс]. – Режим доступа: <http://neerc.ifmo.ru/school/io/>, свободный. Яз. рус. (дата обращения 21.09.2010).
5. Правила проведения полуфинала NEERC [Электронный ресурс]. – Режим доступа: <http://neerc.ifmo.ru/information/contest-rules.html>, свободный. Яз. рус., англ. (дата обращения 21.09.2010).
6. Оршанский С.А. О решении олимпиадных задач по программированию формата ACM ICPC // Мир ПК. – 2005. – № 9.
7. Акишев И.Р. Об опыте участия в командных соревнованиях по программированию формата ACM // Методическая газета для учителей «Информатика». – 2008. – № 19. – С. 20–28.
8. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. – М.: Мир, 1982.

9. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ / Пер. с англ. – 2-е изд. – М.: Вильямс, 2005. – 1296 с.
10. Holland J.P. Adaptation in Natural and Artificial Systems. – The University of Michigan Press, 1975.
11. Mitchell M. An Introduction to Genetic Algorithms. – MA: MIT Press, 1996.
12. Timus Online Judge. Архив задач с проверяющей системой [Электронный ресурс]. – Режим доступа: <http://acm.timus.ru>, свободный. Яз. рус., англ. (дата обращения 21.09.2010).
13. Задача «Ships. Version 2» [Электронный ресурс]. – Режим доступа: <http://acm.timus.ru/problem.aspx?space=1&num=1394>, свободный. Яз. рус., англ. (дата обращения 21.09.2010).
14. Pisinger D. Algorithms for Knapsack Problems: PhD. Thesis. – University of Copenhagen, 1995.
15. Koza J.R. Genetic programming: On the Programming of Computers by Means of Natural Selection. – MA: The MIT Press, 1998.

Буздалов Максим Викторович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, mbuzdalov@gmail.com

УДК 004.05

СОВМЕСТНОЕ ПРИМЕНЕНИЕ КОНТРАКТОВ И ВЕРИФИКАЦИИ ДЛЯ ПОВЫШЕНИЯ КАЧЕСТВА АВТОМАТНЫХ ПРОГРАММ

А.А. Борисенко, В.Г. Парфенов

При создании систем со сложным поведением важную роль играет контроль качества разрабатываемых программ. Цена ошибки в таких системах может быть слишком велика, поэтому важно не просто проверить соответствие создаваемой программы всем предъявленным к ней требованиям, но и сделать этот процесс эффективным, максимально автоматизировав его. На практике этого можно добиться, формализовав все требования к программе и храня полученную исполнимую спецификацию непосредственно вместе с кодом программы.

Рассмотрены существующие методы контроля качества современных программных систем и автоматных программ, а также описан процесс создания среды, позволяющей поддержать сразу три подхода к проверке качества программ с явным выделением состояний: проверку на модели, модульное тестирование и контракты. Предложенный подход позволяет сохранить корректность записи сформулированных требований при изменении самой программы, а также интерактивно контролировать ее качество.

Ключевые слова: контроль качества, соответствие спецификации.

Введение

Качественное программное обеспечение (ПО) – это, прежде всего, надежное программное обеспечение. Зачастую системы, требующие высокого уровня надежности, представляют собой системы со сложным поведением [1], а цена ошибки в таких проектах может быть слишком высокой [2]. При разработке систем со сложным поведением важное место занимает стадия тестирования, а одним из распространенных методов разработки таких систем является автоматное программирование [1].

Другой важной чертой современных программных проектов является их частое изменение: модифицируются требования к системе, находятся и исправляются ошибки. Для контроля качества ПО, соответствия его реализации и спецификации, в современных проектах используется ряд методов: ручное и автоматизированное тестирование, контрактное программирование и верификация [3].

Контроль качества автоматных программ

К автоматным программам могут быть успешно применены следующие методы анализа корректности [4]: тестирование, верификация (проверка на модели и доказательная верификация), контракты. Рассмотрим последовательно каждый из них, выделяя при этом характерные для данного метода преимущества и недостатки.

Тестирование. Тестирование – процесс выявления ошибок в ПО. Запуск программы на определенных входных данных, а также проверка различных сценариев выполнения позволяют достаточно быстро (по сравнению с другими методами поиска ошибок) убедиться в корректности обработки заданных сценариев [5].

Тестирование, применяемое после окончательного написания программы, не способно найти все ошибки. Как заметил Э. Дейкстра, если при тестировании ошибки в программе не найдены, это еще не означает, что их там нет.

Верификация. *Артефактами* жизненного цикла ПО называются различные информационные сущности, документы и модели, создаваемые или используемые в ходе разработки и сопровождения ПО [6].

Верификация проверяет соответствие одних создаваемых в ходе разработки и сопровождения ПО артефактов другим, ранее созданным или используемым в качестве исходных данных, а также соответствие этих артефактов и процессов их разработки правилам и стандартам. В частности, верификация проверяет соответствие между нормами стандартов, описанием требований (техническим заданием) к ПО, проектными решениями, исходным кодом, пользовательской документацией и функционированием самого ПО.

Формальная верификация. Формальная верификация представляет собой процесс доказательства с помощью формальных методов корректности или некорректности алгоритмов, программ и систем в соответствии с заданным описанием их свойств. Она требует высококвалифицированных специалистов в области формальных доказательств и логического вывода. В общем случае задача, решаемая в рамках этого подхода, является алгоритмически неразрешимой. При этом весь процесс формального доказательства связан с огромной ручной работой, что делает его малоприменимым на практике [3].

Верификация на модели. Под верификацией на модели понимают метод формальной верификации, позволяющий проверить, удовлетворяет ли заданная модель системы формальным спецификациям. Применение данного подхода позволяет для заданной модели поведения системы с конечным (возможно, очень большим) числом состояний проверить выполнимость некоторого требования (спецификации), которое обычно формулируется в терминах языка темпоральной логики (LTL, CTL и т.д.). Таким образом, можно проверить не только условия в определенном состоянии системы, но и историю развития во времени [7].

Метод верификации на модели хорошо подходит для проверки поведения автоматных программ. Это связано с тем, что при верификации на модели, обычно, по программе строится модель Крипке [8], которая фактически является специальным видом автомата.

Ощутимым неудобством верификации на модели является необходимость ручного ввода темпоральных спецификаций и описания модели программы на языке, понятном программе-верификатору.

Контракты. Под контрактами обычно понимают совокупность способов формализации и проверки требований к программе, в которую входят инварианты, постуловия и предусловия [9].

С помощью инвариантов удобно отслеживать выполнение требований к работе программы, которые должны исполняться на непрерывном отрезке ее жизненного цикла. При этом сами инварианты остаются недоступными для людей, не участвовавших в разработке кода, а тем более для тех, кто будет заниматься поиском ошибок в программе [10]. Важно отметить, что в современных средах разработки отсутствует поддержка автоматической проверки контрактов для создаваемой автоматной программы. Это усложняет применение данного подхода на практике.

Формализация требований к автоматной программе

Требования, предъявляемые к разрабатываемым программам, обычно формируются словесно. Для автоматической проверки их необходимо формализовать. В качестве примера, наглядно иллюстрирующего проблему формализации вербальных требований к форме, пригодной для автоматической проверки, рассмотрим модель холодильника. Зададим список предъявляемых к ней требований:

1. при закрытой дверце внутренняя лампа не должна гореть;
2. при выходе показаний датчика напряжения за пределы допустимого диапазона управляющее реле выключит питание;
3. в момент отключения охлаждающего элемента показания термостатов не должны превышать допустимые;
4. пока дверца не будет открыта, лампа не будет включена;
5. если открыть дверцу холодильника, прогреть его основной объем и закрыть дверь, то: будет включена лампа; при достижении критической температуры будет включен охлаждающий элемент; лампа будет выключена.

Приведенный список требований можно условно разделить на три группы. В первую группу отнесем требования 1–3. Они подходят для записи контрактов. Первое из них, «При закрытой дверце внутренняя лампочка не должна гореть», является инвариантом – условием, которое должно соблюдаться в течение всего процесса выполнения программы. В требовании 2 проверяется выполнимость некоторого условия при выходе из заданного состояния, а в требовании 3 – при входе в заданное состояние. Формализуем их при помощи пост- и предусловий, оперирующих показаниями термодатчиков. Ко второй группе можно отнести требование 4: «Пока дверца не будет открыта, лампочка не будет включена». Оно содержит выражение, зависящее от последовательности выполнения программы во времени, и его можно компактно записать в качестве темпоральной спецификации: (*Лампа не включена U Дверь открыта*).

Ее можно использовать для верификации на модели. Занесем полученные результаты в таблицу, используя в формальной записи инварианты (**Inv**), постуловия (**Post**) и предусловия (**Pred**).

Заметим, что не всякую темпоральную спецификацию удастся легко верифицировать. Предположим, что проверке подлежит требование «Лампа в холодильнике зажигается строго после открытия двери». В таком случае придется исключить в темпоральной спецификации все возможные переходы, до-

пустимые заданной автоматной моделью при открытии двери за исключением собственно зажигания лампы. Полученная формула стала бы громоздкой и перестала бы быть удобочитаемой. Более того, полученная таким образом спецификация полностью становилась бы привязанной к конкретной реализации модели, а это обстоятельство затруднило бы процесс контроля качества программы при изменении ее реализации.

1. При закрытой дверце внутренняя лампочка не должна гореть	Inv [DoorClosed]: <i>lamp.isTurnedOff</i>
2. При выходе показаний датчика напряжения из допустимого диапазона управляющее реле выключит питание	Post [VoltageOutOfRange]: <i>powerAdapter.isTurnedOff</i>
3. В момент отключения охлаждающего элемента показания термостатов не должны превышать допустимые значения	Pred [FreezerTurnedOff]: <i>thermoSensor.valuesInRange</i>
4. Пока дверца не будет открыта, лампочка не будет включена	<i>lamp.isTurnedOff</i> U DoorOpened

Таблица. Примеры записи формальных спецификаций

Для описанного случая хорошо подходит тестирование. Именно к нему целесообразно прибегнуть и при проверке последнего, 5-го требования. Оно представляет собой сценарий исполнения. Его легко записать и проверить, используя, например, модульное тестирование.

Контроль качества автоматных программ

В настоящей работе предлагается объединить достоинства сразу трех подходов для проверки качества программ с явным выделением состояний:

- проверка спецификаций (проверка на модели);
- проверка предусловий и постусловий, инвариантов (контракты);
- unit testing (модульное тестирование).

В результате процесс создания автоматных программ (с внедренными в него указанными подходами) становится эффективным, благодаря сочетанию возможностей современной среды разработки (статические проверки, рефакторинг, автодополнение и т.д.) и семантической проверки автоматного кода.

Был предложен новый подход к разработке автоматных программ [11]. Он использует мультязыковую среду MPS (<http://www.jetbrains.com/mps>), для создания новых языков и расширения уже существующих, созданных с ее помощью. Языки, разрабатываемые с помощью MPS, не являются текстовыми в традиционном понимании, так как пользователь не пишет текст программы, а вводит ее в виде абстрактного синтаксического дерева (АСД). Такой подход позволяет обойтись без создания лексических и синтаксических анализаторов при создании новых языков, а также настроить преобразования АСД в код на конкретном языке программирования и задать удобную среду для его редактирования.

Отдельно отметим наличие в MPS языка stateMachine, позволяющего для Java-класса, реализующего сложное поведение, добавить автоматную модель, указав набор состояний и переходов между ними.

Реализация предложенного подхода

Для добавления верификации и проверки контрактов в процесс разработки автоматных программ потребуется:

- создать в среде MPS язык темпоральных спецификаций и контрактов, который был назван stateSpec, и описать операторы языка темпоральной логики LTL;
- настроить систему типов темпоральных операторов;
- внедрить секцию со спецификацией в описание автоматных программ на языке stateMachine;
- разработать плагин для запуска внешнего верификатора NuSMV (<http://nusmv.irst.itc.it/>);
- указать сочетания клавиш, запускающие верификацию и проверку контрактов;
- преобразовать автоматную модель к форме, пригодной для автоматической верификации, и настроить обработку полученных данных верификации;
- выделить сообщение о результатах верификации;
- в случае обнаружения контрпримера заменить исходные названия состояний и событий в цепочке исполнения программы, приводящей к нарушению спецификации или контракта.

Поддержка контрактов

С помощью языка stateSpec можно переформулировать заданные к автоматной программе контракты на языке LTL, а затем провести их статическую проверку, используя все тот же верификатор NuSMV. Таким образом, пользователь сможет записывать и проверять темпоральные спецификации или контракты в тех случаях, когда это будет удобнее.

Следует отметить, что верификатор NuSMV работает со своим языком описания автоматных программ. При этом, как уже отмечалось выше, преобразование модели к этому языку вручную чревато появлением привнесенных ошибок. Поэтому преобразование автоматной модели, созданной в среде MPS к форме, понятной верификатору NuSMV, необходимо автоматизировать.

Внедрение полученных результатов

Идея интеграции сразу трех подходов к контролю качества автоматных программ, предложенная в данной работе, была реализована в мультязыковой среде MPS в виде плагина (дополнительной функциональности, доступной при нажатии сочетания клавиш). Для практического внедрения разработанного в рамках данной работы инструментального средства была выбрана программа учета дефектов YouTrack, разработанная в компании ООО «ИнтеллиДжей Лабс» (работает на мировом рынке под брендом JetBrains).

Программа YouTrack представляет собой Интернет-приложение для работы с базами дефектов. Она реализуется на базе системы JetBrains MPS. При генерации автоматов, работающих на сервере используется Java, в то время как пользовательский интерфейс для работы через браузер основан на JavaScript. Это обстоятельство привело к тому, что язык stateSpec получил отдельную реализацию для работы с программами на JavaScript. С его помощью были формализованы и проверены требования, описывающие логику поведения интерфейса программы YouTrack.

Заключение

В современных проектах вся сопутствующая информация должна обновляться в процессе разработки программы. Это относится и к спецификации. Хранить ее «рядом» с кодом недостаточно. Необходимо сделать ее исполнимой, т.е. внедрить автоматический контроль качества программы в процесс ее разработки. Спецификация, которая не отвечает данному требованию, не является эффективной еще и потому, что она никак не реагирует на изменение модели программы и быстро устаревает.

В данной работе предложен метод, позволяющий разрабатывать надежные объектно-ориентированные системы с явным выделением состояний. При этом, в частности, получены следующие результаты:

- предложен подход к программированию по контрактам (контрактному программированию) с явным выделением состояний. При этом отдельно рассмотрены внешние и внутренние контракты;
- разработан метод, позволяющий выбирать оптимальный способ формализации спецификаций к автоматным системам в зависимости от характера спецификации и особенностей автоматной системы;
- предложен способ интеграции действий по обеспечению соответствия реализованной автоматной системы спецификации в процесс разработки программного обеспечения;
- реализована часть функциональности инструмента, позволяющего проводить верификацию автоматной системы во время разработки.

В целях создания среды разработки надежных программ, реализующих системы со сложным поведением, в работе предложено совместить сразу несколько подходов к проверке качества ПО. Тестирование и верификация на модели были внедрены в процесс разработки автоматных программ в среде MPS. Также была изучена роль контрактов в преобразовании требований к программе, сформулированных словесно, к форме, пригодной для автоматической проверки.

В качестве направления дальнейших исследований можно предложить итеративную верификацию, позволяющую существенно ускорить получение результата, опираясь на данные предыдущих процессов верификации.

Исследование выполнено по Федеральной целевой программе «Научные и научно-педагогические кадры инновационной России на 2009–2013 годы» в рамках государственного контракта П2373 от 18 ноября 2009 года.

Литература

1. Поликарпова Н.И., Шальто А.А. Автоматное программирование. – СПб: Питер, 2009 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/books/_book.pdf, своб.
2. Риган П., Хемилтон С. NASA: миссия надежна // Открытые системы. – 2004. – № 3. – С. 12–17. [Электронный ресурс]. – Режим доступа: <http://www.osp.ru/text/302/184060.html>, своб.

3. Отчет по государственному контракту о верификации автоматных программ. Второй этап. – СПбГУ ИТМО, 2007 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/verification/_2007_02_report-verification.pdf, своб.
4. Поликарпова Н.И. Объектно-ориентированный подход к моделированию и спецификации сущностей со сложным поведением. – СПб: СПбГУ ИТМО, 2006 [Электронный ресурс]. – Режим доступа: <http://is.ifmo.ru/papers/oosuch>, своб.
5. Веденеев В.В. Автоматизация тестирования использования программных интерфейсов приложений на основе моделирования конечными автоматами. – СПбГУ ИТМО, 2007 [Электронный ресурс]. – Режим доступа: <http://is.ifmo.ru/testing/vedeneev/>, своб.
6. Винниченко И.В. Автоматизация процессов тестирования. – СПб: Питер, 2005.
7. Курбацкий Е.А., Шалыто А.А. Верификация программ, построенных при помощи автоматного подхода // Материалы международной научно-методической конференции «Высокие интеллектуальные технологии и инновации в образовании и науке». – СПбГПУ. 2008. – С. 293–296 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/download/2008-02-25_politech_verification_kurb.pdf, своб.
8. Кларк Э.М., Грамберг О., Пелед Д. Верификация моделей программ. Model Checking. – М.: МЦНМО, 2002.
9. Мейер Б. Объектно-ориентированное конструирование программных систем. – М.: Интернет-университет информационных технологий, 2005.
10. Мейер Б. Семь принципов тестирования программ // Открытые системы. – 2008. – № 7. – С. 13–29 [Электронный ресурс]. – Режим доступа: <http://www.osp.ru/os/2008/07/5478839/>, своб.
11. Кузьмин Е.В., Соколов В.А. Моделирование спецификация и верификация «автоматных» программ // Программирование. – 2008. – № 1. – С. 2–5 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/download/2008-03-12_verification.pdf, своб.
12. Гуров В.С., Мазин М.А., Шалыто А.А. Текстовый язык автоматного программирования // Тезисы докладов Международной научной конференции, посвященной памяти профессора А.М. Богомолова «Компьютерные науки и технологии». – Саратов: СГУ. 2007. – С. 66–69 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/works/_2007_10_05_mps_textual_language.pdf, своб.

Борисенко Андрей Андреевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, Andrey.Borisenko@gmail.com

Парфенов Владимир Глебович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, декан, parfenov@mail.ifmo.ru

УДК 004.4*242

ВИРТУАЛЬНАЯ ЛАБОРАТОРИЯ ОБУЧЕНИЯ МЕТОДАМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ГЕНЕРАЦИИ УПРАВЛЯЮЩИХ КОНЕЧНЫХ АВТОМАТОВ

А.С. Тяхти

Описывается структура и возможности виртуальной лаборатории для обучения генетическому и автоматному программированию, реализованной на языке С#. Описываются основные этапы создания собственных подключаемых модулей лаборатории.

Ключевые слова: автоматное программирование, виртуальная лаборатория.

Введение

Парадигма автоматного программирования [1, 2] была предложена в 1991 году в России. В рамках данной концепции программа рассматривается как набор конечных автоматов, объектов управления и поставщиков событий.

Зачастую построение конечных автоматов эвристическими методами является затруднительным. В связи с этим для их генерации можно использовать автоматизированные методы, в том числе генетические алгоритмы и генетическое программирование [3].

Для обучения генетическому программированию ранее была создана виртуальная лаборатория для языка программирования Java [4]. Однако в указанной виртуальной лаборатории отсутствовала возможность применения других методов искусственного интеллекта, таких, как, например, метод имитации отжига [5–9]. Эта техника оптимизации использует случайный поиск на основе аналогии с процессом образования в веществе кристаллической структуры с минимальной энергией, происходящем при охлаждении этого вещества.

На основании изложенного можно сделать вывод о том, что актуальной является разработка виртуальной лаборатории для обучения методам искусственного интеллекта. При этом важным требованием

для такой виртуальной лаборатории является возможность не только самостоятельно разрабатывать методы и создавать новые задачи для их тестирования, но и сравнивать различные алгоритмы между собой применительно к конкретным задачам, а также наглядно визуализировать полученные решения.

Основные положения

Виртуальная лаборатория GLOpt (сокращение от Global Optimization) реализована на языке программирования C# на платформе Microsoft.NET. Она состоит из ядра и подключаемых модулей (плагинов), которые реализуют конкретные задачи и методы их решения.

Ядро представлено классом Brain, который отвечает за загрузку основных сущностей программного комплекса – задач, методов искусственного интеллекта и визуализаторов, а также выполняет функции распределения ресурсов между работающими алгоритмами.

Для описания задач используются абстрактные классы Problem и Individual, от которых наследуются классы, описывающие конкретные задачи, например, задачу об «Умном муравье» [10].

Класс Problem содержит метод OptimizationDirection, возвращающий информацию о направлениях оптимизации (min, max) и метод EvaluateIndividual, служащий для вычисления функции приспособленности у конкретной особи. Наследники этого класса должны реализовывать эти методы, а также методы, обеспечивающие доступ к базовой информации о задаче – названии, ее кратком описании. Класс Individual реализует представление индивидуальной для каждой задачи оценочной характеристики Fitness, описывает метод сравнения индивидов между собой.

Алгоритмы решения описываются классами Algorithm и SearchOperator, от которых, в свою очередь, наследуются классы, реализующие конкретные методы искусственного интеллекта, например, метод имитации отжига. Класс SearchOperator необходим для организации требуемых в задаче операций над объектами класса Individual. Он обеспечивает универсальность в применении методов решения к различным задачам. Так, например, для применения уже реализованного алгоритма к задаче об «Умном муравье» требуется определить методы мутации и скрещивания индивидов, а для метода имитации отжига – изменение уже найденного решения в соответствии с выбранным вероятностным распределением.

Общая структура виртуальной лаборатории приведена на рис. 1.

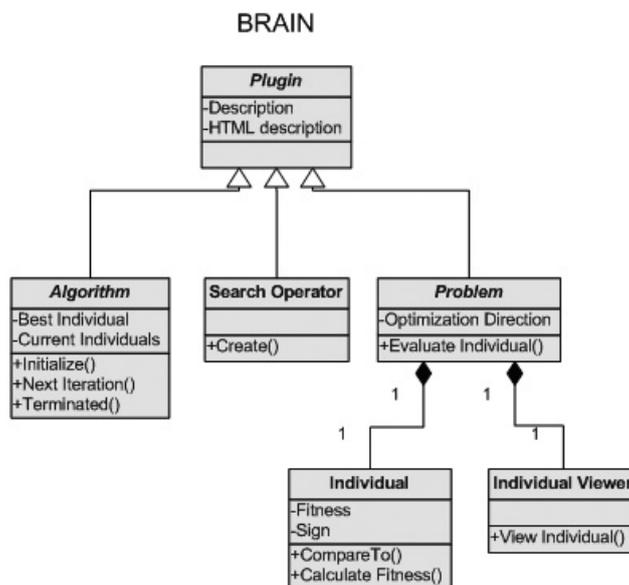


Рис. 1. Структура виртуальной лаборатории

Таким образом, в виртуальной лаборатории существует пять типов плагинов:

1. рассматриваемые задачи (наследуется от абстрактного класса Problem);
2. алгоритмы и методы искусственного интеллекта (наследуется от абстрактного класса Algorithm);
3. методы, адаптирующие алгоритмы к конкретным задачам (наследуется от абстрактного класса SearchOperator);
4. визуализаторы для задач (пример внешнего вида визуализатора приведен на рис. 2);
5. текстовые описания задач и алгоритмов.

Каждый плагин представляет собой dll-библиотеку. Особо отметим, что виртуальная лаборатория спроектирована таким образом, что любую задачу можно решать, используя каждый из реализованных методов искусственного интеллекта.

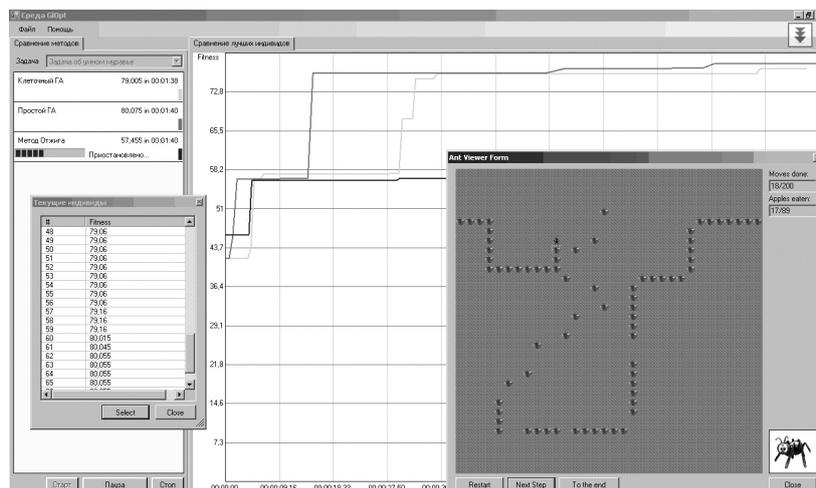


Рис. 2. Визуализатор для задачи об «Умном муравье»

В настоящее время программный комплекс виртуальной лаборатории включает в себя следующие плагины:

- задачи:
 - об «Умном муравье»;
 - об «Умном муравье-3»;
 - о роботе, огибающем препятствия;
 - о расстановке N ферзей на шахматной доске (рис. 3);
- методы искусственного интеллекта:
 - генетические алгоритмы – классический, клеточный, островной;
 - методы имитации отжига – больцмановский, Коши, тушения;
 - эволюционные стратегии;
 - метод оптимизации роем частиц;
 - пчелиный алгоритм;
- визуализаторы для указанных задач;
- графики, позволяющие оценивать эффективность методов применительно к конкретным задачам;
- документация, в том числе html-описания решаемых задач и используемых для их решения методов.

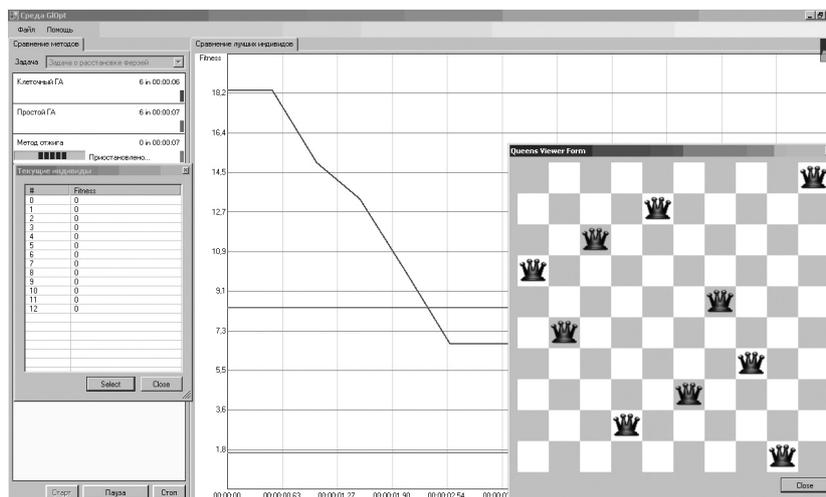


Рис. 3. Задача о расстановке N ферзей на шахматной доске

Создание плагинов

Проектирование модулей предполагает изучение структуры уже реализованных методов оптимизации и построение на их основе собственных. Ниже приведены базовые рекомендации, которым необходимо следовать при разработке собственных модулей, определяющих алгоритм решения. Процесс создания плагинов описывается на примере создания плагина-задачи и алгоритма решения.

Создание плагина-задачи. Приведем основные этапы при реализации новой задачи в виртуальной лаборатории GLOpt.

- Реализация класса-наследника от абстрактного класса Individual, определяющего объект к которому в дальнейшем применяется алгоритм оптимизации. Так, например, в задаче о расстановке ферзей таким объектом выступает шахматная доска с расставленными на ней ферзями.
- Реализация класса-наследника от абстрактного класса Problem. В данном классе должен быть определен метод OptimizationDirection. Также необходимо определить метод EvaluateIndividual, возвращающий значение типа double – значение целевой функции для данного индивида.
- Для реализации компоненты визуализации текущего решения требуется определить класс Viewer, наследуемый от класса IndividualViewer, и, в частности, определить метод ViewIndividual, вызывающий графическую форму, либо представляющий данные о решении в любом другом удобном для пользователя виде.
- Библиотека (dll-файл) откомпилированного модуля должна находиться в папке plugins лаборатории.

Создание плагина-алгоритма. Приведем основные этапы разработки собственного подключаемого модуля, определяющего алгоритм.

- Реализация алгоритма поиска оптимального решения – класса, наследованного от абстрактного класса Algorithm. Данный класс должен определять следующие методы:
 - OptimizationDirection – возвращает одну из двух именованных констант: OptimizationDirection.Minimize или OptimizationDirection.Maximize;
 - Initialize – описывает начальное состояние в алгоритме поиска решения;
 - NextIteration – описывает действия, происходящие на очередной итерации алгоритма;
 - В переменной BestIndividual типа Individual должна содержаться актуальная информация об объекте типа Individual с лучшей на данной итерации алгоритма целевой функцией.
- Реализация интерфейса SearchOperator, наследованного от интерфейса ICreateOperator. В данном интерфейсе должен быть определен метод Create, возвращающий объект типа Individual (абстрактный класс, для каждой задачи используется своя реализация). Также в SearchOperator реализуются все необходимые методы для работы с объектами Individual – например, операции мутации и скрещивания.
- Библиотеки *.dll откомпилированного модуля должны находиться в папке plugins лаборатории.

Сравнение виртуальных лабораторий

В таблице приведены основные сравнительные характеристики настоящей виртуальной лаборатории с ранее созданной лабораторией на языке Java [4].

Критерий сравнения	Лаборатория GLOpt	Лаборатория на Java
Язык программирования	C#	Java
Поддержка задач	Задачи об «Умном муравье», о расстановке ферзей Поддержка добавления новых задач в качестве плагинов	Только задача об «Умном муравье»
Возможность одновременного запуска алгоритмов	Поддерживается Результат работы отображается в виде сводного графика	Не поддерживается
Встроенная документация	В формате html	Отсутствует

Таблица. Сравнительные характеристики лабораторий

Заключение

Важной особенностью виртуальной лаборатории GLOpt является возможность применения реализуемых алгоритмов и методов для любой из рассматриваемых задач. Наглядное сравнение результатов работы алгоритмов, удобство анализа их эффективности при влиянии тех или иных факторов настройки обеспечивается наличием сводных графиков работы методов, средств визуализации, доступа к базовой информации о задачах и алгоритмах непосредственно из самой виртуальной лаборатории.

Гибкая система плагинов, позволяющая реализовывать весь комплекс необходимой функциональности – от новых задач до визуализаторов к ним, призвана максимально упростить реализацию новых модулей, что облегчает понимание основ методов искусственного интеллекта и, в частности, методов имитации отжига, генетических алгоритмов.

Параллельно с выполнением настоящей работы А. Цветковым [11] велась работа по созданию виртуальной web-лаборатории, с помощью которой стало возможным проводить исследование реализованных методов и задач на удаленном сервере, вне зависимости от наличия на конкретной локальной машине того или иного программного обеспечения.

Исследование выполнено по Федеральной целевой программе «Научные и научно-педагогические кадры инновационной России на 2009–2013 годы» в рамках государственного контракта П2236 от 11 ноября 2009 года.

Литература

1. Шалыто А.А. Switch-технология. Алгоритмизация и программирование задач логического управления. – СПб: Наука, 1998. – 628 с.
2. Поликарпова Н.И., Шалыто А.А. Автоматное программирование. – СПб: Питер, 2009 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/books/_book.pdf, своб.
3. Koza J.R. Genetic programming: On the Programming of Computers by Means of Natural Selection. – Cambridge: MIT Press, 1992.
4. Соколов Д.О., Давыдов А.А., Царев Ф.Н., Шалыто А.А. Виртуальная лаборатория обучения генетическому программированию для генерации управляющих конечных автоматов / Сборник трудов третьей Международной научно-практической конференции «Современные информационные технологии и ИТ-образование». – М.: МАКС Пресс, 2008. – С. 179–183 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/works/_2_93_davidov_sokolov.pdf, своб.
5. Ingber A.L. Simulating Annealing: Practice versus theory. – Mathl. Comput. Modelling, 1993.
6. Ёлкин Д.И. Тяхти А.С. Метод отжига. – СПб: СПбГУ ИТМО, 2008 [Электронный ресурс]. – Режим доступа: <http://rain.ifmo.ru/cat/view.php/theory/unordered/ai-annealing-2008/article.pdf>, своб.
7. Джонс М.Т. Программирование искусственного интеллекта в приложениях – М.: ДМК-Пресс, 2004.
8. Лопатин А. Методы отжига. Электронный конспект – Крысталь Б. [Электронный ресурс]. – Режим доступа: www.cs-seminar.spb.ru/reports/52.pdf, своб.
9. Орлянская И.В. Современные подходы к построению методов глобальной оптимизации [Электронный ресурс]. – Режим доступа: <http://zhurnal.ape.relarn.ru/articles/2002/189.pdf>, своб.
10. Бедный Ю.Д., Шалыто А.А. Применение генетических алгоритмов для построения автоматов в задаче «Умный муравей». – СПбГУ ИТМО, 2007 [Электронный ресурс]. – Режим доступа: <http://is.ifmo.ru/works/ant>, свободный. Яз. рус. (дата обращения 09.02.2011).
11. Цветков А.А. Сравнение поведенческих и эволюционных алгоритмов построения управляющих конечных автоматов. Бакалаврская работа. – СПб: СПбГУ ИТМО, 2010.

Тяхти Александр Сергеевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, tyahti@gmail.com

УДК 535.37, 535.343.2

ОБЕСЦВЕЧИВАНИЕ ПРИРОДНЫХ САПФИРОВ

В.А. Асеев, Я.А. Некрасова, К.В. Хомченко

Проведены исследования спектрально-люминесцентных свойств ограненных кристаллов природных сапфиров. В ходе работы образцы подвергались различным методам технологической обработки, которые позволили селективно изменять оптическую плотность сапфиров в определенных спектральных областях. На основе полученных данных были определены наиболее эффективные методики обесцвечивания сапфиров.

Ключевые слова: природный сапфир, облагораживание, спектрально-люминесцентные свойства.

Введение

Окраска является визуально самой заметной характеристикой драгоценных камней и во многом определяет их стоимость. Среди сапфиров наиболее дорогими являются кристаллы василькового цвета и кашмирского синего, а темно-синие камни не представляют большой ценности в ювелирной промышленности. Сегодня внимание разработчиков технологий облагораживания драгоценных камней концентрируется в основном на «улучшении» цвета и повышении прозрачности, что непосредственно связано с изменением оптической плотности кристаллов в определенных спектральных областях. Существует ряд методов облагораживания камней, например, термообработка или воздействие излучения различного спектрального состава и интенсивности, а также их комбинации для получения более значимого результата. Традиционно в ювелирной промышленности определение насыщенности окраски кристалла осуществляется визуально, что зачастую осложняется большим количеством и малыми размерами образцов. Подобная методика приносит значительную субъективность в оценку окраски сапфиров и не дает точной информации о ее изменении. Более объективным для экспресс-оценки является спектрально-люминесцентный анализ, так как окраска, как известно, однозначно связана с положением и интенсивностью полос поглощения в видимой области спектра. И поэтому контроль изменения окраски кристаллов осуществлялся на основе измерения и анализа спектров поглощения.

Также важную роль для эффективности процесса обесцвечивания сапфиров играет определение наличия следовых количеств примесей переходных металлов в составе образца, которые могут привести к получению обратного результата, т.е. потемнению сапфира и как следствие, снижению его стоимости. Традиционные методы неразрушающего контроля примесей, например, рентгенофлуоресцентный анализ, не способны оперативно осуществлять контроль примесей, а также не дают информацию о валентном состоянии переходных металлов. Наиболее чувствительным для таких измерений является люминесцентный метод, который и использовался в данной работе.

Описание эксперимента

В работе исследовались ограненные кристаллы природного сапфира, представляющие собой корунд Al_2O_3 , активированный ионами титана и железа, а также содержащий примесь хрома. Примеси в образцах сапфиров перед данными исследованиями были определены при помощи рентгенофлуоресцентного метода. Данный метод имеет ограничения по чувствительности порядка 0,01% вес. По полученным данным все образцы содержат Cr, V менее 0,01%, титана 0,01–0,02 % вес.

Измерение спектров поглощения проводилось на спектрофотометре Lambda 900 (Perkin Elmer) с интегрирующей сферой PELA-1000 в диапазоне 350–800 нм с шагом 0,2 нм. Интегрирующая сфера позволяет учитывать как проходящий через сапфир свет, так и отраженный от его граней. Образец был закреплен в диафрагме для унификации измерений. По изменению формы и интенсивности полос поглощения в определенных областях видимой области спектра можно судить об изменении окраски сапфиров.

Синяя окраска кристаллов обусловлена высокой оптической плотностью в зеленой и красной областях спектра. Типичный график поглощения природного сапфира представлен на рис. 1. Сильное поглощение у синих сапфиров в области 700–800 нм характерно для ионов титана, полосы на 380 и 450 нм – полосы поглощения ионов Fe^{3+} . Широкие полосы в области 580 нм и 750 нм могут соответствовать поглощению не только титана, но и ионов Fe^{2+} [1, 2]. В рамках поставленной задачи важным является уменьшение поглощения в синей области спектра, т.е. уменьшение содержания трехвалентного железа в составе образца.

Основными методами обесцвечивания природных сапфиров являются термообработка («внутренняя диффузия»), облучение УФ излучением и лазерным излучением при малой мощности.

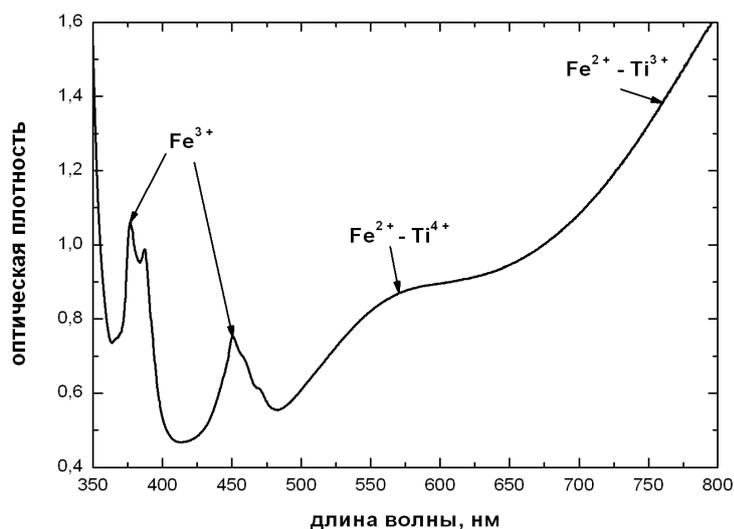


Рис. 1. Спектр поглощения темно-синего сапфира

Результаты и обсуждения

В основе обесцвечивающей термообработки лежит преобразование центров окраски в минералах под действием повышенных температур, а именно изменение их валентности, что приводит к изменению окраски кристаллов. Варьируя параметры отжига, можно разрушать одни и сохранять другие центры окраски. Обычно, чем дольше воздействие рабочей температуры, тем эффективнее облагораживание, что обусловлено диффузионным характером процессов изменения состояния примесей (валентность, комплексы), для прохождения которых необходима достаточная временная выдержка.

В данной работе температурная обработка проводилась при 450°C в течение 18 ч и 900°C в течение 15 ч. В результате отжига изменилась разность пиков, соответствующих длинам волн 380 и 450 нм, что вызвано диффузионными процессами внутри сапфира (рис. 2).

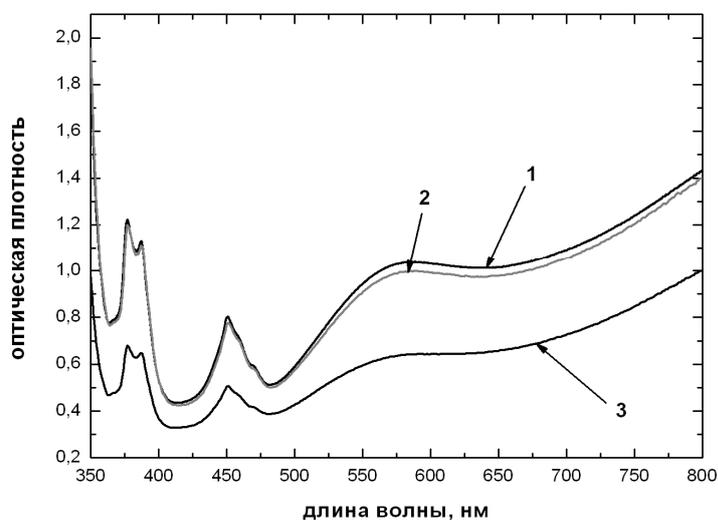


Рис. 2. Спектры поглощения сапфира до и после термообработки: 1 – исходный образец; 2 – термообработка 450°C, 18 ч; 3 – термообработка 900°C, 15 ч

Из графика видно, что при температуре 450°C особых изменений не наблюдается. Однако термообработка при 900°C привела к уменьшению интенсивности поглощения во всем диапазоне измерений. Были выбраны относительно низкие температуры для того, чтобы наличие температурной обработки нельзя было идентифицировать с помощью известных методов диагностики. Для перестройки внутренних структур сапфира, а именно кристаллической структуры, температур 450°C и 900°C недостаточно. Использование отжига при больших температурах позволяет добиться более сильного эффекта, но о проведении термообработок такого типа должно быть указано в паспорте камня, что значительно (в 2–3 раза) снижает стоимость камня.

В случае использования УФ излучения накачка осуществлялась в области полосы поглощения 450 нм, связанной с трехвалентным железом. При воздействии излучением в этой области длин волн

предполагалось добиться уменьшения оптической плотности кристалла за счет различных процессов, например, фотоионизации, которая может вызвать изменение валентности ионов переходных металлов (Fe и Ti), что приводит к смещению спектров поглощения в УФ или ИК диапазон. На рис. 3 представлены спектры изменения оптической плотности в результате воздействия на образец УФ излучения.

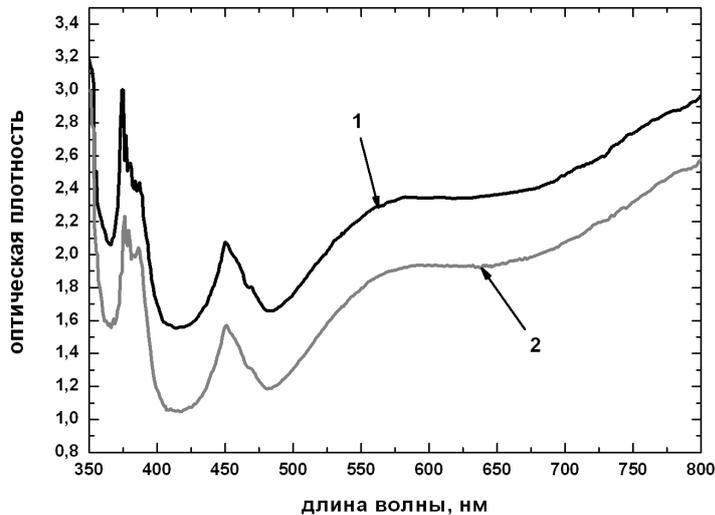


Рис. 3. Спектры поглощения сапфира: 1 – в исходном состоянии; 2 – после облучения УФ

Образец подвергался облучению ртутной лампой, у которой была выделена полоса с максимумом на 365 нм, в течение 40 мин и получил дозу облучения 40 кДж, что привело к увеличению интегрального пропускания по всему диапазону измерений 350–800 нм в результате возможного перехода $Fe^{3+} \rightarrow Fe^{2+}$. Дальнейшее облучение ультрафиолетовым светом не дало существенных изменений в видимой области спектра.

Также для воздействия на полосу поглощения трехвалентного железа в области 450 нм может применяться облучение когерентным излучением малой мощности. В этом случае на изменение интенсивности поглощения могут влиять эффект выжигания спектральных дыр и явление резонанса Фано.

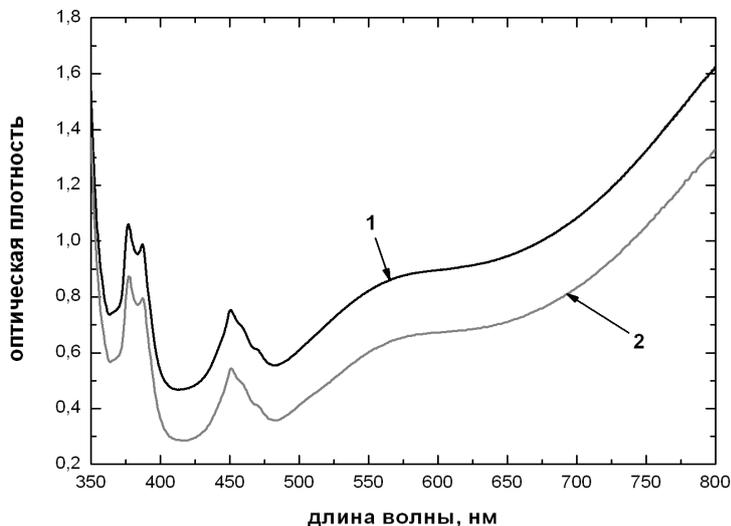


Рис. 4. Спектры поглощения образца: 1 – в исходном состоянии; 2 – после обработки He-Cd лазером (442 нм)

После воздействия излучением He-Cd лазера ($\lambda=442$ нм; 3 мВт) в течение 18 ч произошло увеличение пропускания света в области 380 и 450 нм (рис. 4). Неравномерное изменение спектра по всему диапазону измерений, в частности, в области 350–500 нм предположительно может объясниться частичной перезарядкой ионов железа в ходе облучения.

Для получения наилучшего результата при обесцвечивании сапфиров важно иметь сведения о наличии или отсутствии примесей в составе данного кристалла, которые могут повлиять на процесс облагораживания и привести к потемнению образца, а также об их валентном состоянии. Данная информация может быть получена по результатам люминесцентного метода, который в идеале способен определить наличие единичных атомов, например хрома. Однако существенным ограничением является то, что оп-

ределить концентрацию таких частиц очень сложно, поскольку не все ионы хрома могут люминесцировать. Также спектры люминесценции ионов переходных металлов (их положение и вид контура) зависят от валентности металлов.

В данной работе для определения присутствия следовых количеств примесей, а также изменения их валентного состояния были измерены спектры люминесценции образцов при возбуждении второй гармоникой неодимового лазера (532 нм) в диапазоне 600–1600 нм.

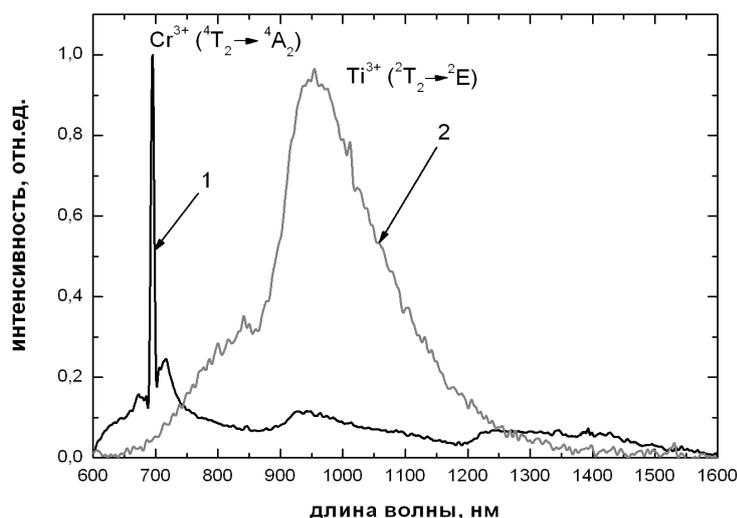


Рис. 5. Спектры люминесценции образца при возбуждении на длине волны 532 нм: 1 – в исходном состоянии; 2 – после облучения He-Cd лазером

На рис. 5 представлены графики люминесценции кристалла сапфира до и после облучения He-Cd лазером. Видно, что в исходном состоянии в основном люминесцируют ионы Cr^{3+} 695 нм (${}^4T_2 \rightarrow {}^4A_2$), также присутствуют полосы, характерные для ионов Ti^{3+} и Cr^{4+} . Можно предположить, что в процессе облучения происходит перезарядка ионов хрома и титана, в результате чего спектр обработанного образца представляет собой типичную широкую полосу люминесценции титана с максимумом в области 950 нм (${}^2T_2 \rightarrow {}^2E$).

Заключение

В ходе исследования были применены различные технологические методы обработки ограненных кристаллов природных сапфиров, которые привели к уменьшению поглощения в видимой области. Наиболее значимый результат дала обработка при температуре 900°C в течение 15 ч: оптическая плотность уменьшилась в среднем на 0,6 в области 380 нм и на 0,2–0,4 в остальном диапазоне измерений. Было показано, что при использовании люминесцентного метода анализа природных сапфиров возможно определение наличия следовых количеств примесей переходных металлов (Fe, Ti, Cr) в составе кристалла, которые влияют на эффективность процесса просветления. Также возможно определение изменения их валентного состояния в ходе обработки. Рассмотренные методики оптического облагораживания позволяют существенно изменять окраску природных сапфиров, повышая их потребительскую ценность.

Работа выполнена на базе Научно-исследовательского института нанофотоники и оптоинформатики при СПбГУ ИТМО.

Литература

1. Путилин Ю.М., Белякова Ю.А., Голенко В.П., Давыдченко А.Г. Синтез минералов. Т. 2. – М.: Недра, 1987. – С. 231–236.
2. Платонов А.Н., Таран М.Н., Балицкий В.С. Природа окраски самоцветов. – М.: Недра, 1984. – С. 28–42.
3. Gaft M., Reisfeld R., Panczer G. Modern luminescence spectroscopy of minerals and materials. – Springer, 2005. – P. 35.

Асеев Владимир Анатольевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, ассистент, Aseev@oi.ifmo.ru

Некрасова Яна Андреевна

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, Nekrasova@oi.ifmo.ru

Хомченко Константин Витальевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, Delibash69@mail.ru

УДК 004.021

ПРЕДСТАВЛЕНИЕ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ ОБ ОПТИЧЕСКИХ ПАРАМЕТРАХ АТМОСФЕРЫ АНАЛИТИЧЕСКИМИ ЗАВИСИМОСТЯМИ

А.В. Демин, М.И. Монсева

Предлагается метод получения эмпирического выражения для расчета коэффициента пропускания атмосферы на основе экспериментальных данных для двух спектральных диапазонов. Описан алгоритм разбиения множества известных экспериментальных данных на группы, представимые в виде различных аналитических выражений. На основании этого определена эмпирическая формула вычисления коэффициента пропускания атмосферы.

Ключевые слова: коэффициент пропускания атмосферы, алгоритм, спектральный диапазон, дистанционное зондирование Земли.

Введение

Дистанционное зондирование Земли из космоса с помощью оптико-цифровых систем и комплексов сверхвысокого разрешения открыло новые возможности по повышению эффективности управления сферами человеческой жизнедеятельности и научного познания. Дистанционное зондирование Земли из космоса (ДЗЗ) в оптическом диапазоне длин волн обеспечивает получение информации как о географических и геофизических параметрах Земли, так и о техногенных процессах, происходящих на ее поверхности, что увеличивает ценность этой информации. ДЗЗ в зависимости от задач, для решения которых оно предназначено, осуществляется такими методами, как маршрутная, объектовая, стереоскопическая, статическая, динамическая, топографическая и спектрометрическая съемки. В оптическом диапазоне ДЗЗ осуществляется с помощью оптико-цифровых систем и комплексов, устанавливаемых на борт космических аппаратов.

Для повышения достоверности получаемых при ДЗЗ данных необходимо учитывать в процессе съемки состояние атмосферы, что накладывает дополнительную нагрузку на служебные системы и требует введения дополнительных систем, обеспечивающих учет параметрического состояния атмосферы в момент съемки. Имеются экспериментальные данные, однако условия их получения могут не соответствовать условиям текущей съемки. В этой связи актуальным является вопрос об аналитическом представлении состояния атмосферы в реальном масштабе времени с целью повышения эффективности и достоверности работы системы ДЗЗ.

Параметрическое состояние атмосферы в части ее пропускания определяется, помимо химического состава, климатической и географической привязкой. Искомая аналитическая зависимость $F(\tau_a)$ включает в себя кроме основной компоненты $\{F(\lambda)\}$, учитывающей спектральное пропускание атмосферы, географическую $\{F(\varphi, \xi)\}$ и климатическую $\{F(T)\}$ компоненты, уточняющие параметрическое состояние атмосферы при конкретных условиях съемки, а именно:

$$F(\tau_a) = \{F(\lambda) \circ F(\varphi, \xi) \circ F(T)\}. \tag{1}$$

Основным расчетным параметром, который необходимо учитывать при проведении ДЗЗ, является коэффициент пропускания атмосферы τ_a .

В [1, 2] имеются экспериментальные данные о значениях коэффициента пропускания атмосферы для диапазонов длин волн 3–5,2 мкм и 8–14 мкм с шагом 0,02 мкм. Экспериментальные данные получены для различных климатических широт и различных значений дальностей видимости (трасса не указана). Следует заметить, что исходные данные содержат множество локальных минимумов и максимумов, что затрудняет подбор единой эмпирической формулы для их представления в аналитической форме.

Таким образом, для учета значения τ_a при проведении ДЗЗ необходимо на основе имеющихся экспериментальных данных выявить функциональную зависимость, т.е. найти алгоритм для вычисления коэффициента пропускания атмосферы для произвольного значения длины волны λ из указанных выше спектральных диапазонов без хранения в памяти множества полученных ранее экспериментальных данных.

Целью данной работы является разработка алгоритма построения эмпирической функции $\tau_a(\lambda)$ на основе экспериментальных данных, полученных ранее.

Решение поставленной задачи

Известны лишь результаты измерений τ_a в некоторых дискретных точках рабочих диапазонов длин волн, полученных в экспериментах, условия проведения которых могут не совпадать с условиями конкретной съемки. Чтобы предсказать результат в любой произвольной точке рассматриваемого диапазона, необходимо построить график зависимости $\tau_a(\lambda)$. Иначе говоря, нужно получить эмпирическую формулу $\tau_a = F(\lambda)$.

Для построения эмпирической формулы необходимо решить две задачи [3]: выявить общий вид формулы и определить значения параметров для полученной формулы. Следует заметить, что формула, которая строится для расчета коэффициента пропускания атмосферы τ_a , является приближенной, поскольку основана на экспериментальных данных.

Для решения данной задачи выберем из [1, 2] в качестве исходных данных результаты измерений для диапазонов 3–5,2 мкм и 8–14 мкм при дальности видимости 5 км в климатической зоне тропиков. Будем предполагать, что искомая зависимость $\tau_a(\lambda)$ определена в каждой точке диапазона и может быть представлена всюду дифференцируемой аналитической функцией. Тогда эта зависимость может быть выражена целой функцией, т.е. функцией вида

$$F(\lambda) = c_0 + c_1 \cdot \lambda + \dots + c_n \cdot \lambda^n + \dots, \quad (2)$$

где c_i – константы, $i = 0, 1, 2 \dots$ [4].

Важно отметить, что искомая эмпирическая формула должна быть пригодна для проведения расчетов на борту космического аппарата, т.е. обеспечивать достаточную точность и скорость вычислений, а также быть удобной для хранения. Кроме того, как было отмечено выше, искомая функция не является монотонно возрастающей или монотонно убывающей, а содержит множество экстремумов. С учетом сказанного представим искомую функцию в виде совокупности элементарных функций. К классу элементарных функций относятся полиномы, показательные функции, логарифмические, тригонометрические и обратные тригонометрические функции, а также функции, получаемые из перечисленных выше с помощью четырех арифметических действий и суперпозиций, примененных конечное число раз [4].

Для представления искомой зависимости совокупностью элементарных функций разобьем исходный диапазон данных на несколько поддиапазонов и на каждом поддиапазоне будем искать свою эмпирическую формулу. При этом возникает проблема нахождения компромисса между выбором минимального числа интервалов разбиения и обеспечением достаточной точности и простоты получаемых для них формул.

Одним из простейших вариантов разбиения диапазона является разбиение по точкам экстремумов. При этом, поскольку в качестве исходных данных используется конечный массив дискретных данных, минимумы и максимумы искомой функции можно определить путем последовательного сравнения значений соседних элементов этого массива.

Другим вариантом разбиения может быть разбиение по точкам перегиба. В случае аналитической функции точки перегиба можно найти путем вычисления нулей второй производной функции. В данном же случае, когда известны лишь дискретные значения функции, вместо вычисления производной функции можно найти приближенные значения точек перегиба через вычисления приращений функции в ее соседних точках. Например, приближенное значение производной можно вычислить по формуле

$$f'(x_i) = \frac{1}{2} \cdot \left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i} + \frac{y_i - y_{i-1}}{x_i - x_{i-1}} \right). \quad (3)$$

Однако исходные экспериментальные данные имеют множество экстремумов и точек перегиба, а число интервалов разбиения должно быть минимально, как было сказано выше. Для решения поставленной задачи будем стремиться при разбиении выбрать один или два наиболее выдающихся экстремума. Также заметим, что каждая точка разбиения входит только в один из разделяемых ею поддиапазонов.

Данные из диапазона 8–14 мкм для поиска аналитических зависимостей разобьем на два участка (рис. 1).

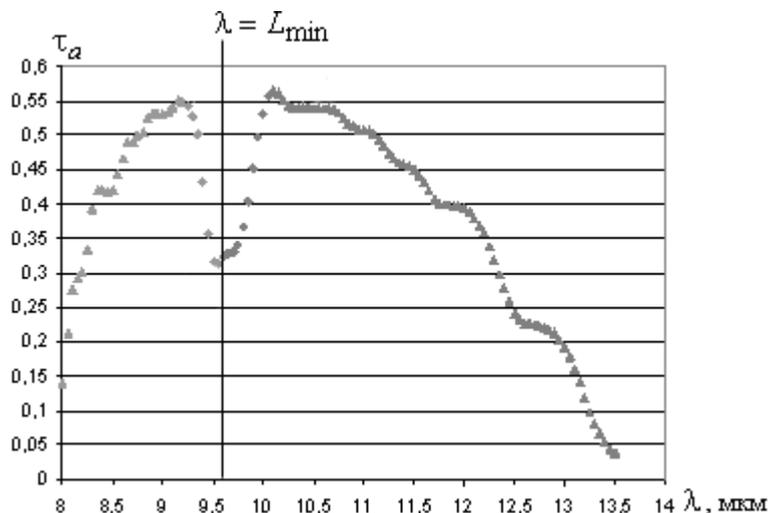


Рис. 1. Разбиение диапазона 8–14 мкм на участки для поиска зависимостей

Алгоритм разбиения – следующий:

1. Вычисляется приближенно производная $\tau'_a(\lambda)$ по формуле (3);
2. Находится абсцисса ее наименьшего минимума $\lambda_{d\min}$ (рис. 2);

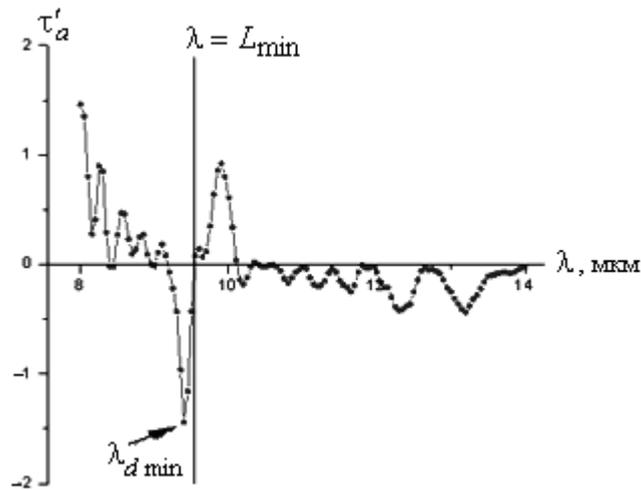


Рис. 2. График производной $\tau'_a(\lambda)$

3. Определяются экстремумы исходной функции путем поэлементного сравнения массива исходных данных;
4. Выбирается точка локального минимума исходной функции, чья абсцисса расположена наиболее близко к $\lambda_{d\min}$;
5. Абсцисса полученной точки обозначается как L_{\min} ;
6. Получаем два участка разбиения: $[\lambda_n; L_{\min}]$ и $(L_{\min}; \lambda_k]$. Здесь λ_n и λ_k – начальное и конечное значения диапазона, соответственно.

Данные из диапазона 3–5,2 мкм для поиска аналитических зависимостей разобьем на три участка (рис. 3) путем выделения наиболее выдающихся пиков. Алгоритм разбиения – следующий:

1. Находится «нулевая область» функции – наибольшая совокупность соседних абсцисс, в которых функция принимает нулевое значение. Абсцисса, ближайшая слева к нулевой области, обозначается как λ_{m2} ;
2. Находится λ_{m1} – абсцисса наименьшего минимума на интервале $[\lambda_n; \lambda_{m2}]$;
3. Приближение производится для трех участков: $[\lambda_n; \lambda_{m1})$, $[\lambda_{m1}; \lambda_{m2})$ и $(\lambda_{m2}; \lambda_k]$. Здесь λ_n и λ_k – начальное и конечное значения диапазона соответственно.

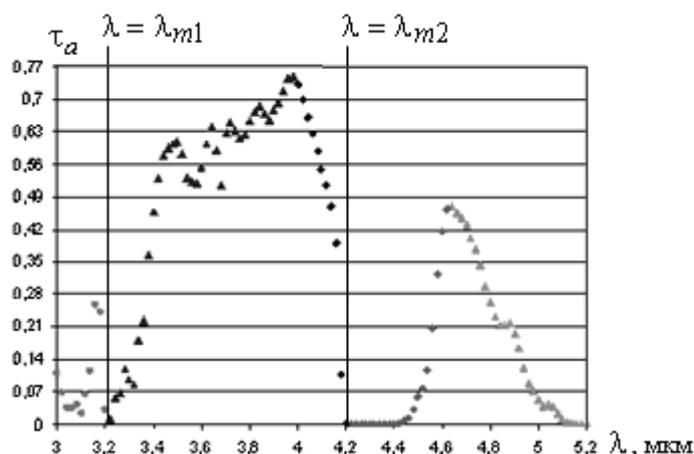


Рис. 3. Разбиение диапазона 3–5,2 мкм на участки для поиска зависимостей

Различия в разбиении диапазонов на участки обусловлены различиями в характере исходных экспериментальных данных. Также вследствие физических причин на каждом участке определяется свой вид кривой регрессии.

В результате перебора различных известных зависимостей (полиномиальной, экспоненциальной, логарифмической, кривой Гаусса и других) в качестве формулы приближения для всех участков диапазонов были выбраны полином пятой степени

$$F(\lambda) = c_0 + c_1 \cdot \lambda + c_2 \cdot \lambda^2 + c_3 \cdot \lambda^3 + c_4 \cdot \lambda^4 + c_5 \cdot \lambda^5 \quad (4)$$

с параметрами $c_0, c_1, c_2, c_3, c_4, c_5$ и функция вида

$$F(\lambda) = \tau_0 + A \cdot \frac{1}{1 + \exp\left(-\frac{\lambda - \lambda_c + \frac{1}{2} \cdot w_1}{w_2}\right)} \cdot \left(1 - \frac{1}{1 + \exp\left(-\frac{\lambda - \lambda_c - \frac{1}{2} \cdot w_1}{w_3}\right)} \right) \quad (5)$$

с параметрами $\tau_0, A, \lambda_c, w_1, w_2, w_3$.

Выбор зависимости проводился с использованием специальных математических пакетов, позволяющих провести аппроксимацию заданной функцией по методу наименьших квадратов и рассчитать коэффициент детерминации R^2 . Коэффициент детерминации вычисляется как отношение суммы квадратов регрессии к общей сумме квадратов. Таким образом, $0 < R^2 \leq 1$. $R^2 = 1$ в случае идеальной модели, когда все экспериментальные точки исходных данных лежат на полученной кривой регрессии. Для аппроксимации в итоге были выбраны функции, для которых коэффициент детерминации был наиболее близок к единице ($0,9 < R^2 \leq 1$).

После расчета значений параметров обеих формул приближения для каждого из участков выбирается та формула, для которой значение R^2 больше.

Определение наилучших параметров для полученных формул производится путем регрессионного анализа – с помощью метода наименьших квадратов. Поскольку аппроксимация производится нелинейными зависимостями, для минимизации величины отклонения воспользуемся итерационным алгоритмом. К итерационным методам относятся метод градиентного спуска, метод Ньютона, метод Ньютона–Гаусса и другие [5]. При решении поставленной задачи для нахождения параметров эмпирической формулы $\tau_a = F(\lambda)$ будем использовать алгоритм Левенберга–Маркуардта, представляющий собой «компромисс между методом линеаризации и методом наискорейшего спуска» [6].

Графики найденных зависимостей $\tau_a(\lambda)$ для дальности видимости 5 км в тропиках изображены на рис. 4. Для всех полученных кривых $R^2 > 0,94$.

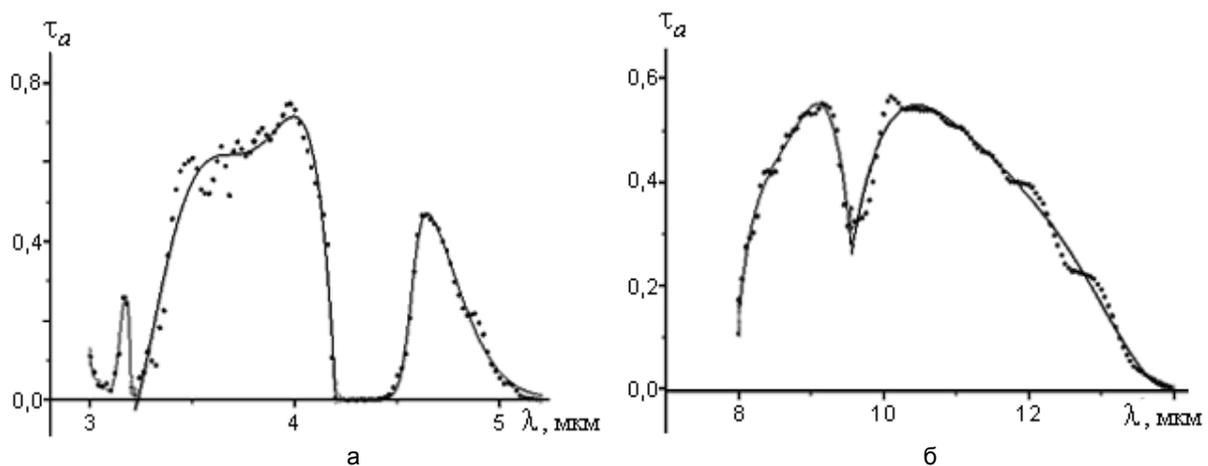


Рис. 4. Графики полученных аппроксимирующих зависимостей: диапазон 3–5,2 мкм (а); диапазон 8–14 мкм (б)

Для других исходных данных в этих же спектральных диапазонах общий вид эмпирических формул будет тот же. Различаться будут значения параметров формулы. Это обусловлено тем, что экспериментальные данные имеют географическую и климатическую привязку.

Точность приближения зависимости $\tau_a(\lambda)$, как было отмечено выше, оценивается коэффициентом детерминации R^2 . В настоящей задаче приемлемым значением R^2 будем считать значение, большее 0,9; оптимальным – большее 0,94.

Заключение

Предложен алгоритм разбиения множества известных экспериментальных данных для диапазонов 3–5,2 мкм и 8–14 мкм на группы для их представления аналитическими зависимостями. Объединение этих зависимостей позволит получить обобщенную эмпирическую формулу для расчета коэффициента пропускания атмосферы, что является предметом следующей работы.

Литература

1. Справочник по инфракрасной технике / Под ред. У. Волф, Г. Цисис. Т. 1. Физика ИК излучения: Пер. с англ. – М.: Мир, 1995. – 606 с.
2. Ллойд Дж. Системы тепловидения: Пер. с англ. – М.: Мир, 1978. – 417 с.
3. Демидович Б.П., Шувалова Э.З., Марон И.А. Численные методы анализа: Приближение функций, дифференциальные и интегральные уравнения / Под ред. Б.П. Демидовича. – 3-е изд. – М.: Наука, 1967. – С. 79–81.
4. Математическая энциклопедия в 5 томах / Под ред. И.М. Виноградова. – М.: Советская энциклопедия, 1985. – Т. 5. – С. 797–799.
5. Айвазян С.А., Енюков И.С., Мешалкин Л.Д. Прикладная статистика. Исследование зависимостей: Справочное издание. – М.: Финансы и статистика, 1985. – С. 301–306.
6. Дрейпер Н., Смит Г. Прикладной регрессионный анализ: в 2-х кн. Пер. с англ. – 2-е изд. – М.: Финансы и статистика, 1986. – Т. 2. – С. 206–207.

Демин Анатолий Владимирович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, dav_60@mail.ru

Моисеева Мария Игоревна

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, mim14@mail.ru

УДК 535.8

СОГЛАСОВАННАЯ СИСТЕМА КВАНТОВОЙ РАССЫЛКИ КРИПТОГРАФИЧЕСКОГО КЛЮЧА НА ПОДНЕСУЩЕЙ ЧАСТОТЕ МОДУЛИРОВАННОГО СВЕТА

А.В. Рупасов, А.В. Глейм, В.И. Егоров, Ю.Т. Мазуренко

Создана принципиальная схема согласованной (plug-and-play) системы распространения криптографического ключа на поднесущей частоте модулированного света, обеспечивающая, безусловно, безопасную передачу информации. Предложены механизмы компенсации поляризационной зависимости фазовых модуляторов и негативного влияния двулучепреломления в волокне, приведены экспериментальные результаты.

Ключевые слова: квантовая криптография, безопасное распределение ключа, поднесущие частоты

Введение

В работе [1] было показано, что коммуникационные методы, опирающиеся на квантовые свойства света (при использовании одиночных фотонов в технологии передачи), позволяют передавать по незащищенному каналу связи последовательность случайных символов таким образом, что вторжение злоумышленника в канал связи неизбежно обнаруживается легитимными пользователями (традиционно именуемыми Алиса и Боб). Тем самым, квантовая механика позволяет реализовать надежную передачу абсолютно стойкого ключа. Этот процесс передачи принято называть квантовой рассылкой ключа (КРК) [2]. В существующих системах КРК наиболее распространен метод кодирования состояний одиночных фотонов с помощью модуляции оптической фазы [2, 3].

Технология квантовой рассылки ключа на поднесущей частоте модулированного света (КРКПЧ) была предложена в работе [4] и развита в работах [5–9]. Использование поднесущих частот позволяет облегчить введение оптической фазы в рабочие сигналы [4]. Однако в системах КРКПЧ возникают те же проблемы, что и при разработке других устройств КРК, использующих модуляцию фазы. Можно выделить следующие проблемы устройств КРК с фазовой модуляцией: проблема синхронизации фазы и проблема двулучепреломления в системах волоконной связи. Первая из них связана с тем, что оптические фазы сигналов, вводимые Алисой и Бобом, должны быть согласованы с высокой точностью; одно из возможных решений приводится в [9]. Вторая проблема заключается в том, что электрооптические фазовые модуляторы, в частности, используемые в волоконных линиях связи, в большом числе случаев чувствительны к поляризации излучения. Вместе с тем стандартное оптическое волокно обладает двулучепреломлением, которое носит случайный характер, в том числе зависит случайным образом от времени. С учетом этого обстоятельства простейшая схема КРКПЧ [4] в реальности может обладать существенными недостатками, как и другие схемы КРК с фазовой модуляцией, и не являться, таким образом, согласованной (plug-and-play) [10]. Действительно, Алиса может однозначно ввести фазу в сигнал, непосредственно излучаемый ее лазером. Однако при передаче этого сигнала по длинному волокну к Бобу состояние поляризации этого сигнала может непредсказуемо измениться. Поскольку модулятор Боба также чувствителен к состоянию поляризации падающего излучения, результат модуляции сигнала Бобом может случайным образом зависеть от времени. Предлагаемое техническое решение данной проблемы описывается ниже.

Система компенсации двулучепреломления волокна и поляризационной чувствительности модуляторов в установках квантовой рассылки ключа

Рис. 1 иллюстрирует используемый в системах КРК принцип компенсации двулучепреломления волокна и поляризационной чувствительности модулятора [10]. Излучение лазера Л после его модуляции с помощью модулятора Алисы МА проходит сквозь 3-портовый циркулятор и направляется на модулятор Боба МБ. В предлагаемой схеме модулированное Бобом излучение направляется не на детектор фотонов, а на фарадеевское зеркало ФЗ. После отражения от фарадеевского зеркала вертикальная и горизонтальная компоненты поляризации меняются местами. Все изменения состояния поляризации при прохождении излучения через модулятор МБ компенсируются. Кроме того, при двойном проходе через модулятор МБ поляризационная чувствительность модуляции также компенсируется. Иными словами, модулятор МБ становится поляризационно-независимым. Излучение, выходящее из модулятора МБ и попадающее вновь в 3-портовый циркулятор, направляется циркулятором в другой канал, ведущий к детектору фотонов ДФ.

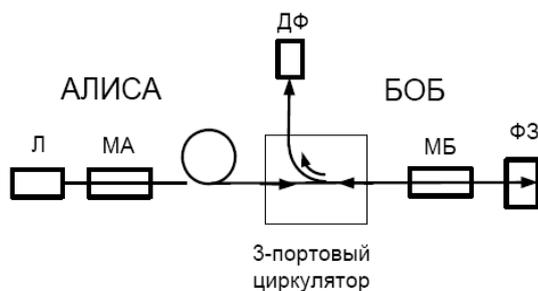


Рис. 1. Схема компенсации двулучепреломления в установках КРК

Схема (рис. 1) обладает следующим недостатком. При довольно высоких частотах модуляции (порядка ГГц), необходимых для реализации КРКПЧ, электрическое поле, накладываемое на электрооптический кристалл, формируется в виде бегущей волны. В этом случае распространение световой и электрической волн в кристалле модулятора может быть сделано синхронным, что приводит к существенному увеличению эффективности модуляции. Это означает, что электрооптические характеристики модулятора МБ в общем случае зависят от направления распространения света – слева направо или справа налево. Действительно, в одном случае распространение электрической и световой волны однонаправлено, а в другом случае – противоположно. Чтобы компенсировать такого рода невзаимность электрооптического модулятора, предлагается использовать вместо одного модулятора МБ на рис. 1 два идентичных модулятора, устанавливаемых последовательно таким образом, что направления распространения бегущих электрических волн в этих модуляторах противоположны. Полученная схема иллюстрируется рис. 2, на котором одинаковые модуляторы МБ1 и МБ2 установлены таким образом, что их бегущие электрические волны распространяются в противоположных направлениях (полые стрелки на рис. 2 указывают на направление распространения электрической волны).

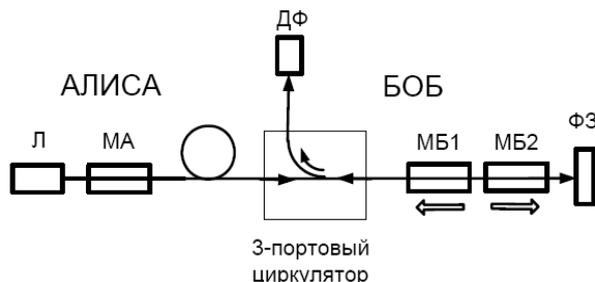


Рис. 2. Схема компенсации характеристик модулятора от направления распространения света в установках КРКПЧ

Схема согласованной установки квантовой рассылки криптографического ключа на поднесущей частоте модулированного света

Была разработана принципиальная схема самосогласованного устройства КРКПЧ с учетом описанных выше модификаций приемного узла. Эта схема изображена на рис. 3. Устройство КРКПЧ содержит терминалы Алисы и Боба, соединенные волоконно-оптической линией связи. Терминал Алисы содержит источник монохроматического излучения – лазерный диод ЛД и электрооптический модулятор Алисы МА. В результате периодической фазовой модуляции монохроматического излучения лазерного диода дополнительно к основной частоте формируются две боковые частоты, оптические фазы которых совпадают с фазой электрического сигнала, управляющего модулятором. Алиса вводит в боковые частоты оптическую фазу, в точности равную фазе модулирующего радиочастотного (РЧ) сигнала. Сигнал, сформированный Алисой, передается по волоконно-оптической линии связи в терминал Боба. В терминале Боба этот сигнал направляется прежде всего в 4-портовый волоконно-оптический циркулятор, который его направляет в группу двух модуляторов Боба МБ1 и МБ2. Необходимость применения двух одинаковых модуляторов с противоположным направлением электрической волны была объяснена выше. Излучение, прошедшее модуляторы МБ1 и МБ2, падает на фарадеевское зеркало ФЗ, отражается от него и вновь проходит через ту же пару модуляторов. При этом динамически компенсируется двулучепреломление оптического волокна, соединяющего терминалы Алисы и Боба, а также поляризационная зависимость модуляторов МБ1 и МБ2. Таким образом, предлагаемый комплекс двух зеркально расположенных модуляторов совместно с фарадеевским зеркалом фактически является модулятором, нечувствительным к поляризации излучения. Пользуясь одновременно модуляторами МБ1 и МБ2, Боб вводит в боковые частоты собственный сигнал, характеризуемый фазой Боба. Сигнал, содержащий фазы Алисы и Боба, направляется циркулятором в узел спектральной маршрутизации центральной частоты и боковых

частот передаваемого излучения. Этот узел состоит из двух волоконно-оптических коллиматоров и интерферометра Фабри–Перо. Коллиматоры необходимы для формирования широкого светового пучка на пластинах интерферометра. Следует отметить, что вместо традиционного интерферометра Фабри–Перо, состоящего из плоских зеркал, может быть использован конфокальный интерферометр. Конфокальный интерферометр может быть просто интегрирован в волоконно-оптическую систему. Центральная частота оптического излучения настраивается на резонанс пропускания интерферометра Фабри–Перо и поэтому распространяется в направлении фотодиода ФД. Фотодиод ФД обеспечивает детектирование центральной частоты, содержащей классический сигнал. Боковые частоты отражаются с хорошей эффективностью от интерферометра Фабри–Перо и вводятся в циркулятор, который их направляет на детектор одиночных фотонов ДФ. Для работы описываемого устройства необходима также синхронизация фазы РЧ сигналов, управляющих соответствующими модуляторами на терминалах Алисы и Боба. Синхронизация может быть реализована при передаче оптического синхронизирующего сигнала (в виде сигнала биений двух оптических частот) в том же волокне, но на частоте DWDM, соседней с частотой передачи квантовой информации [9].

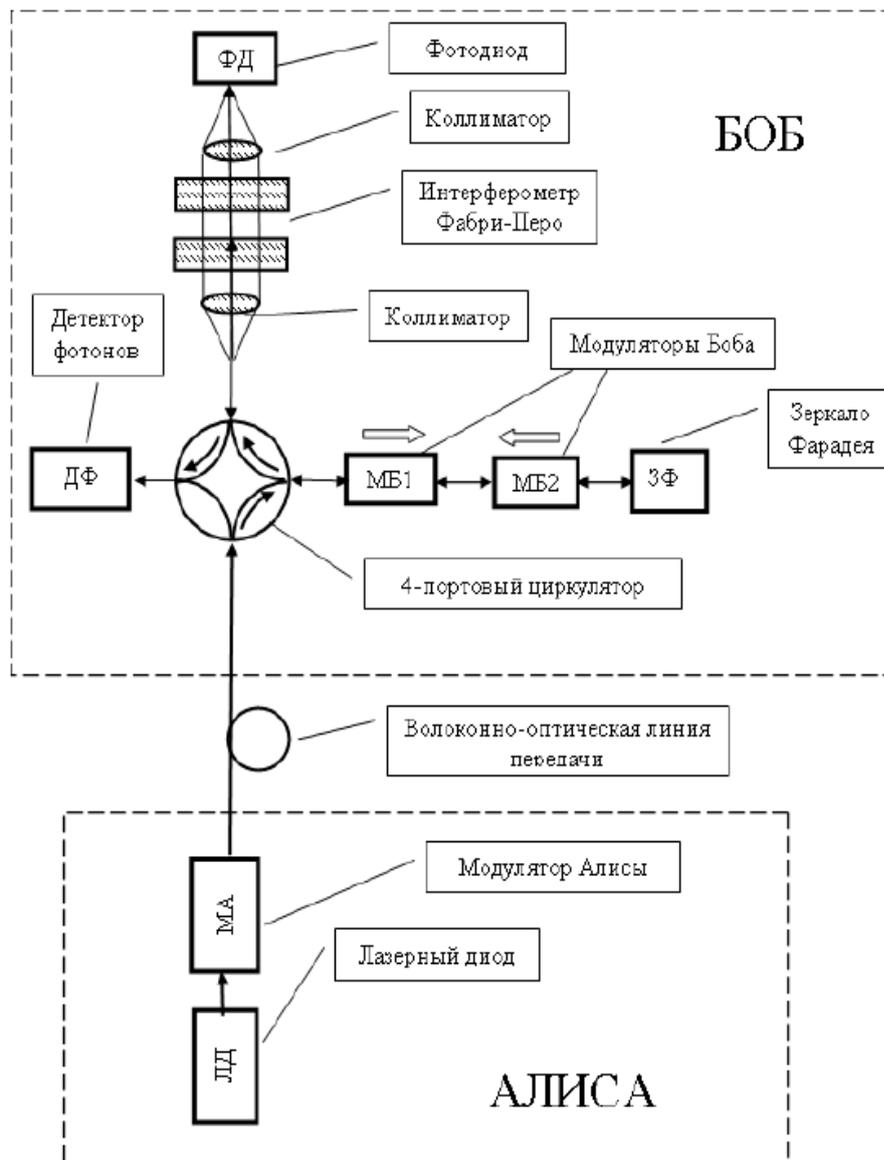


Рис. 3. Принципиальная схема согласованного устройства КРКПЧ

Экспериментальные результаты

Используемые в технологии безопасного распространения ключа явления конструктивной и деструктивной интерференции на поднесущих частотах были продемонстрированы (для классического режима) в экспериментах со сканированием частоты лазера при одновременной записи на осциллографе спектров излучения, прошедших через спектральный фильтр. Соответствующие осциллограммы приведены на рис. 4.

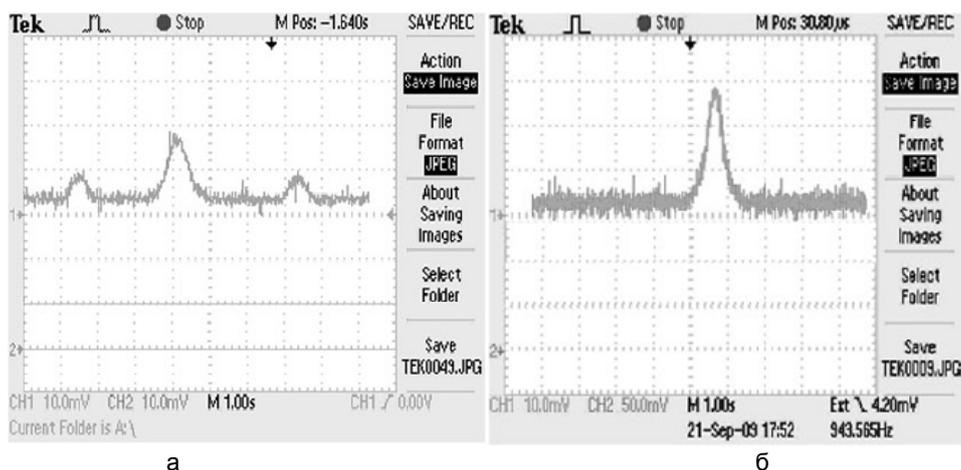


Рис. 4. Спектры сигнала при конструктивной (а) и деструктивной (б) интерференциях на поднесущих частотах

Для проверки эффективности компенсации поляризационной зависимости модуляторов был произведен следующий опыт. Оптическое волокно в области между Алисой и Бобом подвергалось механическому воздействию, в результате чего состояние поляризации передаваемого сигнала произвольно менялось, и в схеме без компенсации интерференционная картина должна была заметно исказиться. На рис. 5–6 приведены результаты этого опыта. Кривая Т соответствует прошедшему сквозь интерферометр Фабри–Перо сигналу, т.е. сигналу на основной частоте, а кривая R – сигналу на боковой частоте, отраженному от интерферометра.

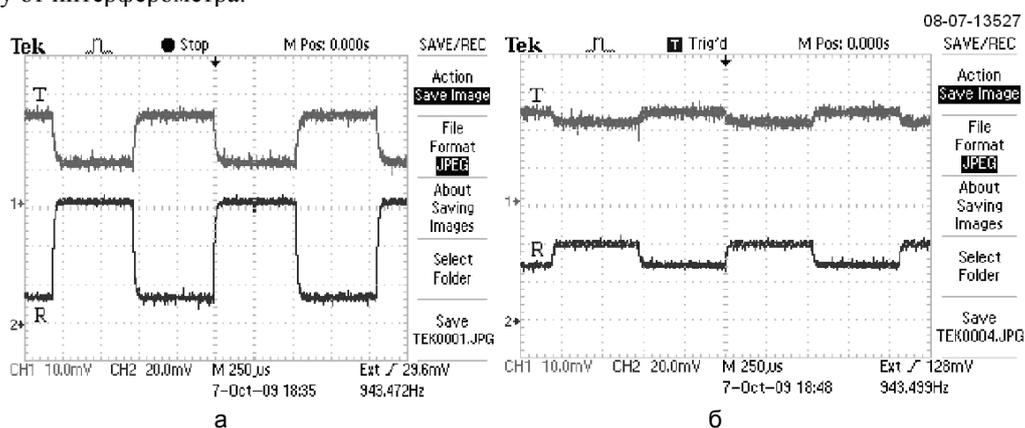


Рис. 5. Сравнение интерференционных картин при осцилляции разности фаз поднесущих частот в схеме с реализованным механизмом компенсации (а) и без компенсации (б). Исходное состояние

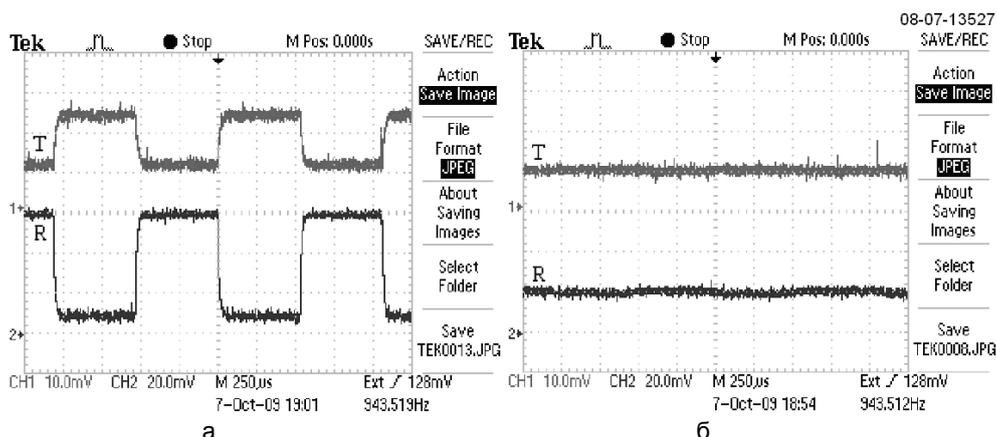


Рис. 6. Сравнение интерференционных картин при осцилляции разности фаз поднесущих частот в схеме с реализованным механизмом компенсации (а) и без компенсации (б). Результат механического воздействия

Из приведенных рисунков видно, что интерференционная картина при наличии компенсации не искажается, что позволяет утверждать, что эксперимент подтвердил практическую применимость описанного механизма компенсации.

Заключение

Предлагаемая модифицированная схема является согласованной (plug-and-play), поскольку в ней происходит автоматическая динамическая компенсация двулучепреломления волоконно-оптической линии связи и компенсация поляризационной зависимости электрооптических модуляторов, что было подтверждено экспериментально. Это делает данный класс систем более привлекательным для практического применения, чем более традиционные схемы, использующие классический импульс только в целях синхронизации сигналов.

Литература

1. Bennett C.H. and G. Brassard. Quantum Cryptography: Public Key Distribution and Coin Tossing // Proceedings of IEEE International Conference on Computers Systems and Signal Processing. – 1984. – P. 175–179.
2. Gisin N., Ribordy G., Tittel W. and H. Zbinden. Quantum cryptography // Rev. Mod. Phys. – 2002. – V. 74. – P. 145.
3. Bennett C.H. Quantum cryptography using any two nonorthogonal states // Phys. Rev. Lett. – 1992. – V. 68. – P. 3121.
4. Мазуренко Ю.Т., Меролла Ж.-М., Годжебюр Ж.-П. Квантовая передача информации с помощью поднесущей частоты. Применение к квантовой криптографии // Оптика и спектроскопия. – 1999. – Т. 86 – С. 181.
5. Merolla J.-M., Mazurenko Y., Goedgebuer J.-P., Porte H. and W.T. Rhodes. Phasemodulation transmission system for quantum cryptography // Optics Lett. – 1999. – V. 24. – P. 104.
6. Merolla J.-M., Mazurenko Y., Goedgebuer J.-P. and W.T. Rhodes Single photon interference in sidebands of phase-modulated light for quantum cryptography // Phys. Rev. Lett. – 1999. – V. 82 – P. 1656.
7. Merolla J.-M., Mazurenko Y., Goedgebuer J.-P., Duraffourg L., Porte L. and W.T. Rhodes. Quantum cryptographic device using single photon phase modulation // Physical Review A. – 1999. – V. 60. – P. 1899.
8. Duraffourg L., Merolla J.-M., Goedgebuer J.-P., Mazurenko Y. and W.T. Rhodes. Compact transmission system using single-sideband modulation of light for quantum cryptography // Optics Letters. – 2001. – V. 26 – P. 1427.
9. Guerreau O.L., Merolla J.-M., Soujaeff A., Patois F., Goedgebuer J.-P., Malassenet F.J. Long-distance QKD transmission using single-sideband detection scheme with WDM synchronization // Selected Topics in Quantum Electronics. – 2003. – V. 9 – P. 1533.
10. Muller A., Herzog T., Huttner B., Tittel W., Zbinden H., Gisin N. Plug and play' systems for quantum cryptography // Appl. Phys. Lett. – 1997. – V. 70 – P. 793.

Рупасов Андрей Викторович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, sadbender@yandex.ru

Глейм Артур Викторович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, aglejm@yandex.ru

Егоров Владимир Ильич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, egorovvl@gmail.com

Мазуренко Юрий Тарасович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор физ.-мат. наук, старший научный сотрудник, yurimaz@gmail.com

УДК 535.1

ФАЗОВАЯ САМОМОДУЛЯЦИЯ ОДНОПЕРИОДНЫХ ОПТИЧЕСКИХ ВОЛН

А.А. Дроздов, С.А. Козлов

Получены аналитические и численные решения уравнения динамики поля светового импульса в изотропной диэлектрической среде с нормальной групповой дисперсией и безынерционной кубической нелинейностью для граничного условия вида однопериодной оптической волны. Показано, что уширение спектра таких предельно коротких по числу колебаний импульсов в нелинейной среде происходит как в высоко-, так и в низкочастотную область. Максимум спектральной плотности смещается в низкочастотную область. Возникновения привычного в среде с кубической нелинейностью излучения утроенных частот не происходит, а генерируется волна с центральной частотой в примерно четыре с половиной раза большей исходной центральной частоты волны.

Ключевые слова: однопериодные импульсы, фазовая самомодуляция, нелинейные диэлектрические среды.

Введение

Фазовая самомодуляция интенсивных сверхкоротких световых импульсов в прозрачных оптических средах теоретически изучалась ранее в большом числе работ [1, 2]. В последнее десятилетие особо

большой интерес вызвали исследования особенностей этого нелинейного явления для импульсов, содержащих лишь несколько колебаний светового поля [3–5]. В работе [6], по-видимому, впервые явление фазовой самомодуляции экспериментально наблюдалось для предельно коротких по числу колебаний однопериодных импульсов (single cycle pulses), которые генерируются в терагерцовом спектральном диапазоне [7]. В настоящей работе особенности фазовой самомодуляции однопериодных волн в изотропных диэлектрических средах с нормальной групповой дисперсией и безынерционной кубической нелинейностью рассмотрены теоретически.

**Уравнение динамики поля излучения из малого числа колебаний
в нелинейной диэлектрической среде**

Уравнение, описывающее эволюцию линейно поляризованного излучения, спектр которого находится в области прозрачности нелинейной диэлектрической среды и в той ее части, где групповая дисперсия нормальная, можно записать в виде [8]

$$\frac{\partial E}{\partial z} + \frac{N_0}{c} \frac{\partial E}{\partial t} - a \frac{\partial^3 E}{\partial t^3} + gE^2 \frac{\partial E}{\partial t} = 0, \quad (1)$$

где E – электрическое поле светового импульса; z – пространственная координата, вдоль которой распространяется импульс; t – время; c – скорость света в вакууме; N_0 , a – параметры, характеризующие зависимость линейного показателя преломления n_0 среды от частоты ω излучения:

$$n_0(\omega) = N_0 + a\omega^2, \quad (2)$$

параметр g характеризует безынерционную кубическую по полю нелинейность поляризационного отклика среды и связан с коэффициентом ее нелинейного показателя преломления n_2 соотношением $g = 2n_2/c$ [9].

Для получения аналитических решений уравнения (1) целесообразно провести следующую его нормировку. Введем новые безразмерные переменные $\tilde{E} = E/E_0$, $\tilde{z} = z/\lambda_0$, $\tilde{t} = t/T_0$, где E_0 – максимальное значение напряженности электрического поля на входе в нелинейную среду, T_0 – центральный период колебаний поля входного излучения, $\lambda_0 = cT_0/N_0$ – центральная длина волны. В этих переменных уравнение (1) принимает следующий вид:

$$\frac{\partial \tilde{E}}{\partial \tilde{z}} + \frac{\partial \tilde{E}}{\partial \tilde{t}} - A \frac{\partial^3 \tilde{E}}{\partial \tilde{t}^3} + G \frac{\partial \tilde{E}^3}{\partial \tilde{t}} = 0, \quad (3)$$

где безразмерный коэффициент $A = ac/N_0T_0^2$ характеризует дисперсию линейного показателя преломления диэлектрической среды; $G = cgE_0^2/3N_0$ описывает вклад безынерционной кубической по полю нелинейности. При дальнейшем анализе уравнения (3) знак « \sim » будем опускать.

Аналитическое решение нелинейного уравнения динамики поля

Для получения аналитического решения уравнения (3) будем использовать метод последовательных приближений Пикара [10], в котором малым будем предполагать параметр G . В этом разделе работы ограничимся анализом решений нормированного уравнения (3) для случая, когда дисперсией линейного показателя преломления среды можно пренебречь, т.е. когда выполняется условие $G \gg A$. Например, это неравенство выполняется для кристалла ниобата лития ($N_0 = 4,7$, $a = 3,4 \times 10^{-38} \text{ с}^3/\text{см}$, $n_2 = 5,4 \cdot 10^{-12} \text{ см}^2/\text{Вт}$ [6]) в поле терагерцового излучения с интенсивностью $I = 1,0 \cdot 10^8 \text{ Вт}/\text{см}^2$ и центральным периодом $T_0 = 4,7 \cdot 10^{-12} \text{ с}$. Этим условиям, при которых в работе [6] экспериментально наблюдалась фазовая самомодуляция, соответствуют $G = 7,7 \cdot 10^{-5}$, $A = 9,8 \cdot 10^{-6}$. Приведенные параметры дисперсии ниобата лития N_0 и a были рассчитаны авторами методом наименьших квадратов при аппроксимации частотной зависимости показателя преломления стехиометрического кристалла $\text{MgO}:\text{LiNbO}_3$ [11].

В соответствии с выбранным методом решение уравнения (3) будем искать в виде

$$E = E^{(0)} + GE^{(1)} + G^2E^{(2)} + \dots, \quad (4)$$

в котором ограничимся только двумя первыми слагаемыми. Тогда в представлении (4) нормированное уравнение (3) будет иметь следующий вид:

$$\begin{cases} \frac{\partial E^{(0)}}{\partial z} + \frac{\partial E^{(0)}}{\partial t} = 0 \\ \frac{\partial E^{(1)}}{\partial z} + \frac{\partial E^{(1)}}{\partial t} + \frac{\partial E^{(0)3}}{\partial t} = 0. \end{cases} \quad (5)$$

Решение первого уравнения системы (5) имеет вид [12]:

$$E^{(0)}(z, t) = E^{(0)}(t - z). \quad (6)$$

Второе уравнение системы (5) также несложно решить в квадратурах, переписав его в новых переменных $z' = z$, $\tau = t - z$. Тогда его решение примет следующий вид [13]:

$$E^{(1)}(z', \tau) = - \int_{z'_0}^{z'} \frac{\partial}{\partial \tau} \left(E^{(0)}(\tau) \right)^3 dz'' \quad (7)$$

Временную эволюцию поля световой волны на входе в среду (при $z'_0 = 0$) будем рассматривать в виде однопериодного импульса

$$E^{(0)}(0, \tau) = E_0 \frac{\tau}{\tau_p} \exp\left(-\frac{\tau^2}{\tau_p^2}\right), \quad (8)$$

где E_0 – его амплитуда, τ_p – длительность. Волна (8) хорошо аппроксимирует, например, терагерцовое излучение фотопроводящих поверхностей полупроводников при их облучении фемтосекундными импульсами [14]. Соответственно спектр такого излучения имеет вид

$$G(\omega) = \int_{-\infty}^{\infty} E^{(0)}(0, \tau) e^{-i\omega\tau} d\tau = -\frac{\sqrt{\pi}}{2} E_0 \tau_p^2 i \omega e^{-\left(\frac{\tau_p \omega}{2}\right)^2}. \quad (9)$$

Из (9) следует, что максимум спектра приходится на центральную частоту $\omega_0 = \sqrt{2}/\tau_p$.

Решение (7) с учетом (8) может быть приведено к виду

$$E^{(1)}(z', \tau) = -3E_0 \left(\frac{\tau^2}{\tau_p^3}\right) \left(1 - 2\frac{\tau^2}{\tau_p^2}\right) \left(\exp\left(-\frac{3\tau^2}{\tau_p^2}\right)\right) (z' - z'_0), \quad (10)$$

спектр которого имеет вид:

$$G^{(1)}(\omega) = \frac{\sqrt{3\pi}}{36} E_0^3 \tau_p^2 \omega^2 e^{-\frac{1}{3}\left(\frac{\tau_p \omega}{2}\right)^2} \left(1 - \frac{1}{2}\left(\frac{\tau_p \omega}{3}\right)^2\right). \quad (11)$$

Из решения (11) следует, что спектральная плотность излучения, генерируемого в нелинейной среде, обращается в ноль на утроенной центральной частоте $\omega = 3\sqrt{2}/\tau_p = 3\omega_0$. Этот вывод чрезвычайно любопытен, так как генерация излучения на утроенных частотах – это классическое нелинейное оптическое явление в средах с кубической нелинейностью. Однако выяснилось, что для однопериодного импульса (8) для центральной частоты спектра исходного излучения это не так. Оценим теперь частоту, соответствующую максимуму спектральной плотности генерируемого излучения на новых высоких частотах. Взяв производную от выражения (11) по частоте и приравняв ее к нулю, находим, что

$$\omega_{\max}^{(1)} = \sqrt{42}/\tau_p \approx 4,6\omega_0. \quad (12)$$

Таким образом, итерационное решение (4) уравнения динамики светового поля (1) для однопериодной на входе в нелинейную среду волны (8) в бездисперсионном приближении имеет вид

$$E(z', \tau) = E^{(0)}(\tau) + GE^{(1)}(z', \tau) = E_0 \frac{\tau}{\tau_p} \exp\left(-\frac{\tau^2}{\tau_p^2}\right) \times \left[1 - 3G \left(\frac{\tau}{\tau_p}\right) \left(1 - 2\frac{\tau^2}{\tau_p^2}\right) \left(\exp\left(-\frac{2\tau^2}{\tau_p^2}\right)\right) (z' - z'_0)\right]. \quad (13)$$

Иллюстрации аналитического решения нелинейного уравнения динамики поля для граничного условия вида однопериодной волны

На рис. 1 приведена иллюстрация решения (13) при $G(z' - z'_0) = 0,3$. На рис. 1, а, изображена рассчитанная в первой итерации поправка (10) к электрическому полю волны, а на рис. 1, б, – поправка (11) к модулю спектра. Пунктиром показаны временная и спектральная структуры поля однопериодной волны на входе в среду. На рис. 1, в, приведено общее решение (13), которое описывает результат самовоздействия однопериодной оптической волны в нелинейной среде без учета ее дисперсии.

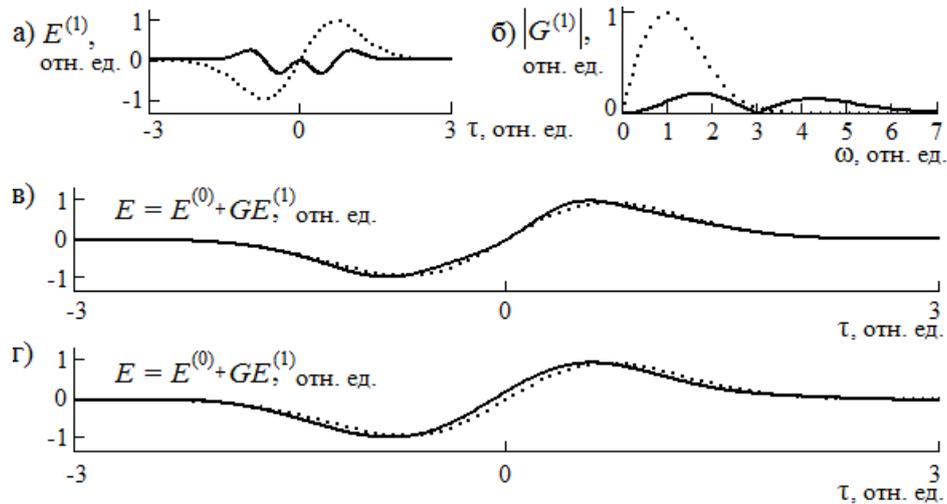


Рис. 1. Изменение временной и спектральной структур однопериодной оптической волны из-за ее самовоздействия в нелинейной диэлектрической среде: итерационная добавка к полю $E^{(1)}$ (а) и модулю спектра $G^{(1)}$ (б); результирующее поле с совместным действием эффектов фазовой само модуляции и генерации утроенных частот (в) и с учетом только эффекта фазовой само модуляции (г). Пунктиром показаны структуры поля и спектра начального импульса

Рис. 1, б, иллюстрирует обращение в ноль на утроенной центральной частоте модуля спектра $G^{(1)}$ поправки (10) к полю волны. Видно, что генерируемая в нелинейной среде добавка к основным частотам излучения простирается от трех до семи центральных частот. Ее максимум находится на частоте в примерно 4,5 раза большей исходной центральной частоты волны. Из рис. 1, в, видно, что максимум и минимум однопериодной волны из-за самовоздействия излучения в нелинейной среде начинают запаздывать во времени. Под самовоздействием здесь будем понимать эффекты фазовой само модуляции и генерации в поле начального излучения утроенных частот, которые в бездисперсионной среде не разделимы. Видно, что при совместном влиянии этих явлений происходит и искажение временного профиля импульса. Для сравнения на рис. 2, г, приведено изменение временной структуры волны, которое происходило бы без учета эффекта генерации утроенных частот. Оно, естественно, полностью соответствует хорошо известному характеру фазовой само модуляции для импульсов из большого числа колебаний.

Результаты численного моделирования эволюции исходно однопериодной оптической волны в нелинейной диэлектрической среде

Для численного моделирования динамики поля излучения в нелинейной оптической среде удобна другая нормировка уравнения (1). Введем новые переменные $\bar{E} = E/E_0$, $\bar{z} = a\omega_0^3 z$, $\bar{t} = \omega_0 t$, где E_0 – максимальное значение напряженности электрического поля на входе в нелинейную среду, ω_0 – центральная частота излучения, и перейдем к «запаздывающему» времени $\bar{\tau} = \bar{t} - N_0 \bar{z}/c$. В новых переменных уравнение (1) приводится к виду [3, 8]

$$\frac{\partial \bar{E}}{\partial \bar{z}} - \frac{\partial^3 \bar{E}}{\partial \bar{\tau}^3} + \bar{G} \bar{E}^2 \frac{\partial \bar{E}}{\partial \bar{\tau}} = 0, \tag{14}$$

где описывающий вклад безынерционной кубической по полю нелинейности безразмерный коэффициент $\bar{G} = cg E_0^2 / 3N_0 = 4n_2 I / ac\omega_0^2$. Например, рассмотренному выше кристаллу ниобата лития в поле терагерцового излучения с центральной частотой 1,0 ТГц и интенсивностью $I = 1,0 \cdot 10^8$ Вт/см² соответствует $\bar{G} = 5,4 \cdot 10^{-2}$. В качестве модели однопериодного импульса на входе среду (при $\bar{z} = 0$) будем по-прежнему брать зависимость (8), которая в новых переменных принимает вид

$$E^{(0)}(0, \bar{t}) = \frac{\bar{t}}{\omega_0 \tau_p} \exp\left(-\frac{\bar{t}^2}{\omega_0^2 \tau_p^2}\right), \tag{15}$$

На рис. 2 приведены результаты численного моделирования по уравнению (14) эволюции временной структуры поля однопериодной волны (15) в среде с параметром $\bar{G} = 5,4 \cdot 10^{-2}$.

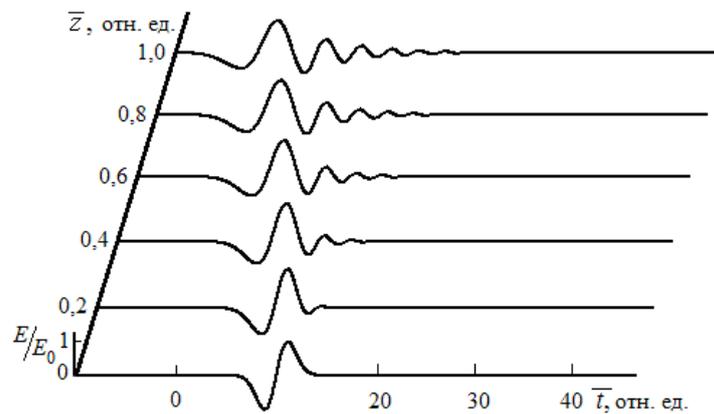


Рис. 2. Динамика электрического поля E , нормированного на максимальное значение входного поля E_0 , в среде с параметром $\bar{G} = 5,4 \cdot 10^{-2}$

Из рис. 2 видно, что уже на расстоянии $\bar{z} = 1,0$ ($z = 1,2$ мм) происходит увеличение длительности импульса, измеряемой по уровню половины максимальной амплитуды поля, за счет дисперсии линейного показателя преломления среды примерно в 4 раза. Расчеты показывают, что уширения спектра импульса при таком параметре \bar{G} практически не происходит, т.е. среда для излучения является линейной.

На рис. 3, 4 приведены аналогичные результаты численного моделирования по уравнению (14) эволюции временной и спектральной структур исходно однопериодной волны (15), распространяющейся в среде с параметром $\bar{G} = 5,4$. Из рис. 3 видно, что по мере генерации новых колебаний электрического поля на заднем фронте волны уже с расстояния $\bar{z} = 0,4$ ($z = 0,5$ мм) от основной волны отделяется импульс, максимальная амплитуда поля которого составляет величину порядка 2% амплитуды входного поля.

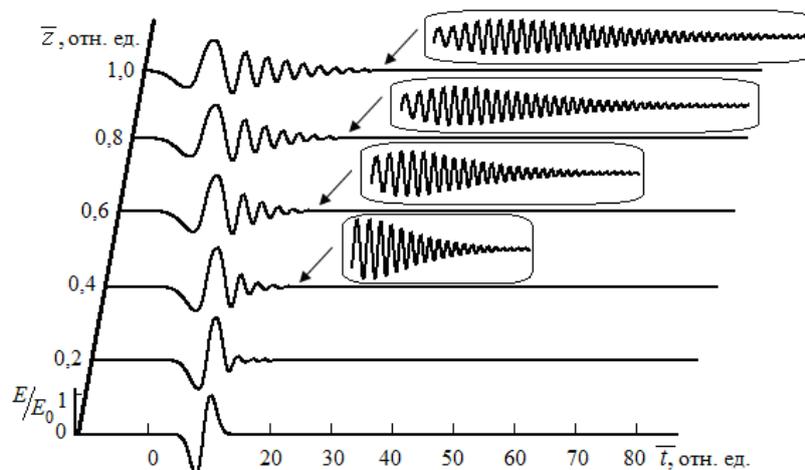


Рис. 3. Динамика электрического поля E , нормированного на максимальное значение входного поля E_0 , в среде с параметром $\bar{G} = 5,4$. Во вставке изображено электрическое поле образующейся новой временной структуры

На рис. 4 приведена иллюстрация эволюции спектральной плотности G однопериодной волны в среде с параметром $\bar{G} = 5,4$, нормированной на максимальную входную спектральную плотность G_0 . Из рисунка видно, что при распространении исходно однопериодной волны в среде с таким параметром \bar{G} происходит значительное смещение спектральной плотности как в высокочастотную область с характерным минимумом на частоте в примерно 3,5 раза большей начальной центральной частоты, так и в низкочастотную область с приблизительно 20% сдвигом максимума ее плотности. Видно, что происходит генерация волны с центральной частотой примерно в 4 раза большей центральной, спектральная плотность которой составляет величину порядка 3% от максимальной спектральной плотности.

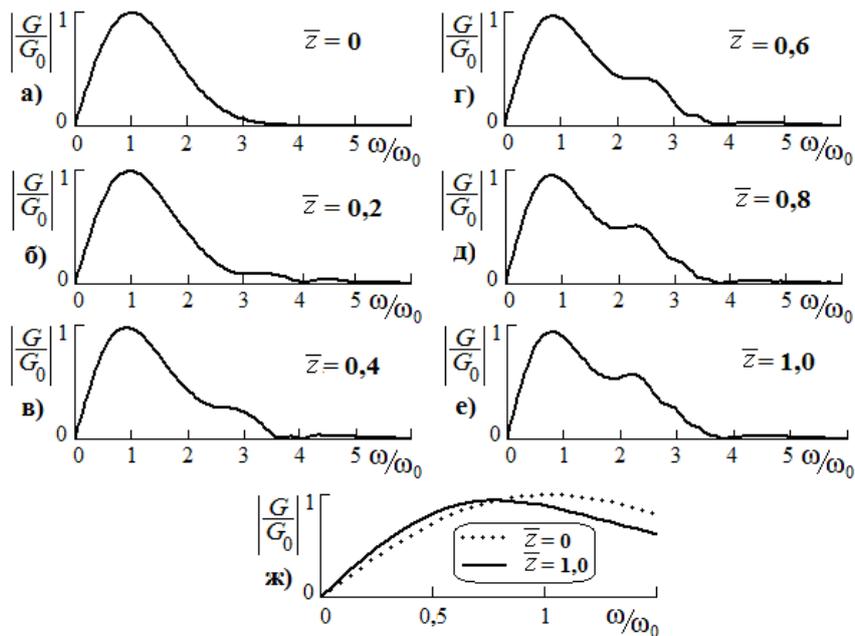


Рис. 4. Динамика модуля спектральной плотности G , нормированной на максимальную входную спектральную плотность G_0 , в среде с параметром $\bar{G} = 5,4$ (а–е). Смещение максимума спектральной плотности волны на выходе из среды в низкочастотную область (ж)

Заключение

В работе рассмотрены особенности фазовой самомодуляции световых волн, содержащих на входе в нелинейную среду лишь одно полное колебание светового поля. Показано, что уширение спектра однопериодных импульсов в изотропной диэлектрической среде с нормальной дисперсией и безынерционной кубической нелинейностью происходит как в высоко-, так и в низкочастотную область. Возникновения излучения утроенных частот не происходит, а генерируется волна с центральной частотой в примерно 4,5 раза большей исходной центральной частоты волны.

Работа поддержана грантами НШ-5707.2010.2 и РНП 2.1.1/4923.

Литература

1. Ахманов С.А., Выслоух В.А., Чиркин А.С. Оптика фемтосекундных лазерных импульсов. – М.: Наука, 1988. – 312 с.
2. Агравал Г. Нелинейная волоконная оптика. – М.: Мир, 1996. – 324 с.
3. Козлов С.А., Самарцев В.В. Основы фемтосекундной оптики. – М.: Физматлит, 2009. – 292 с.
4. Беспалов В.Г., Козлов С.А., Сутягин А.Н., Шполянский Ю.А. Сверхуширение спектра интенсивных фемтосекундных лазерных импульсов и их временное сжатие до одного колебания светового поля // Оптический журнал. – 1998. – Т. 65. – № 10. – С. 85–88.
5. Белашенков Н.Р., Дроздов А.А., Козлов С.А., Шполянский Ю.А., Цыпкин А.Н. Фазовая самомодуляция фемтосекундных световых импульсов, спектры которых сверхуширены в диэлектриках с нормальной групповой дисперсией // Оптический журнал. – 2008. – Т. 75. – № 10. – С. 3–8.
6. Hebling J., Hoffmann M.C., Ka-Lo Yeh, Tóth G. and Keith A.N. Nonlinear lattice response observed through terahertz SPM // Springer Series in Chemical Physics. – 2009. – V. 92. – Part 8. – P. 651–653.
7. Крюков П.Г. Фемтосекундные импульсы. – М.: Физматлит, 2008. – 208 с.
8. Козлов С.А., Сазонов С.В. Нелинейное распространение импульсов длительностью в несколько колебаний светового поля в диэлектрических средах // ЖЭТФ. – 1997. – Т. 111. – № 2. – С. 404–418.
9. Беспалов В.Г., Козлов С.А., Шполянский Ю.А. Метод анализа динамики распространения фемтосекундных импульсов с континуумным спектром в прозрачных оптических средах // Оптический журнал. – 2000. – Т. 67. – № 4. – С. 5–11.
10. Корн Г., Корн Т. Справочник по математике. – М.: Наука, 1973. – 832 с.
11. Palfalvi L., Hebling J., Kuhl J., Peter A. and Polgar K. Temperature dependence of the absorption and refraction of Mg-doped congruent and stoichiometric LiNbO₃ in the THz range // J. Appl. Phys. – 2005. – V. 97. – 123505.
12. Фарлоу С. Уравнения с частными производными для научных работников и инженеров. – М.: Мир, 1985. – 384 с.

13. Буяновская Е.М., Козлов С.А. Закономерности взаимодействия встречных световых волн из малого числа колебаний в нелинейных диэлектрических средах // Научно-технический вестник СПбГУ ИТМО. – 2010. – Т. 66. – № 2. – С. 23–29.
14. Беспалов В.Г., Крылов В.Н., Путилин С.Э., Стаселько Д.И. Генерация излучения в дальнем ИК диапазоне спектра при фемтосекундном оптическом возбуждении полупроводника InAs в магнитном поле // Оптика и спектроскопия. – 2002. – Т. 93. – № 1. – С. 158–162.

Дроздов Аркадий Анатольевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, arkadiy.drozlov@gmail.com

Козлов Сергей Аркадьевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор физ.-мат. наук, профессор, декан, kozlov@mail.ifmo.ru

УДК 535.361.1

ДИФФУЗИЯ СВЕТА В СРЕДЕ С НАНОРАЗМЕРНЫМИ НЕОДНОРОДНОСТЯМИ

А.В. Альфимов, Е.М. Арысланова, Д.Н. Вавулин, О.В. Андреева, Д.Д. Темнова, В.В. Лесничий, В.И. Альмяшев, С.А. Кириллова, С.А. Чивилихин, И.Ю. Попов, В.В. Гусаров

Предложен метод определения характерного размера наночастиц или других наноразмерных неоднородностей, основанный на сравнении экспериментальной и теоретической спектральной зависимости коэффициента ослабления света при его прохождении через неоднородную среду. Одним из возможных применений такого метода является анализ размеров наночастиц, получаемых в процессе гидротермального синтеза. В настоящей работе в качестве модельной среды выбрано пористое стекло с известным средним размером пор.

Ключевые слова: рассеяние света, диффузионное приближение, наночастицы, гидротермальный синтез, пористое стекло.

Введение

Описание распространения света в рассеивающих и поглощающих средах привлекает внимание многих исследователей [1–6]. Это связано как с широкой распространенностью такого рода систем, так и с их практической важностью.

Рассеяние света широко используется для определения размеров, морфологии и концентрации взвесей наночастиц в жидкости. Определение размеров и морфологии частиц позволяет, например, создавать современные биомолекулярные сенсоры, устройства контроля окружающей среды и т.д. В настоящей работе предложен метод оценки характерных размеров неоднородностей наноструктурированных сред, апробированный на примере нанопористого стекла.

Диффузия света в рассеивающей и поглощающей среде

Рассмотрим распространение света в случайно-неоднородной поглощающей и рассеивающей среде в предположении, что длина волны света много больше характерного размера неоднородностей [4]. Уравнение переноса излучения в этом случае имеет вид

$$\Omega \nabla I = -(\gamma_a + \gamma_s)I + \frac{\gamma_a}{4\pi} \int Id\omega, \quad (1)$$

где I – интенсивность излучения; Ω – единичный вектор в направлении луча; γ_a и γ_s – коэффициенты поглощения и рассеяния среды (которые считаются постоянными); $d\omega$ – элемент телесного угла. Величина $(\gamma_a + \gamma_s)^{-1}$ представляет собой характерный пространственный масштаб изменения интенсивности излучения. Будем считать, что эта величина мала по сравнению с характерными размерами рассматриваемой системы. Получим в этом предположении уравнение диффузии излучения. Введем суммарную интенсивность излучения, падающего на данный элемент среды,

$$Q = \int Id\omega, \quad (2)$$

и применим к (1) преобразование Фурье. Тогда имеем:

$$ik\Omega I_k = -(\gamma_a + \gamma_s)I_k + \frac{\gamma_a}{4\pi} Q_k,$$

где $I_k = \int I e^{-ikx} d^3x$. Отсюда

$$I_k = \frac{\gamma_a}{4\pi} \frac{Q_k}{\gamma_a + \gamma_s + ik\Omega}. \quad (3)$$

Интегрируя (3) по телесному углу и используя (2), получаем

$$Q_{\mathbf{k}} = \frac{\gamma_a}{4\pi} Q_{\mathbf{k}} \int \frac{d\omega}{\gamma_a + \gamma_s + i\mathbf{k}\Omega},$$

или

$$\left[1 - \frac{\gamma_s}{k} \operatorname{arctg} \left(\frac{k}{\gamma_a + \gamma_s} \right) \right] Q_{\mathbf{k}} = 0, \quad (4)$$

где $k = |\mathbf{k}|$. Выражение (4) представляет собой Фурье-представление псевдодифференциального уравнения [7] для величины Q . В коротковолновом приближении $\frac{k}{\gamma_a + \gamma_s} \ll 1$ это уравнение приобретает вид

$$\left[1 + \frac{k^2 \gamma_s}{3\gamma_a (\gamma_a + \gamma_s)^2} \right] Q_{\mathbf{k}} = 0. \quad (5)$$

Ограничимся рассмотрением сред, в которых коэффициент рассеяния γ_s существенно превышает коэффициент поглощения γ_a . Тогда, переводя (5) в координатное представление, получаем уравнение диффузии излучения [4]

$$[\Delta - 3\gamma_a \gamma_s] Q = 0. \quad (6)$$

В тех же предположениях может быть получено граничное условие к уравнению (6)

$$Q - \frac{1}{2\gamma_s} \frac{\partial Q}{\partial n} = 2Q_e, \quad (7)$$

где $\frac{\partial}{\partial n}$ – производная вдоль внешней нормали к границе области; Q_e – суммарная интенсивность излучения, падающая снаружи на границу области.

Нанопористые стекла

В качестве модельного объекта для экспериментальных исследований был взят образец пористого стекла НПС-17 в виде полированного диска диаметром 15 мм и толщиной 1 мм со средним размером пор 17 нм и относительным объемом пор 58%.

Перед проведением измерений поры были заполнены водой. Для этого образец был погружен на сутки в дистиллированную воду. Измерения проводились на стандартном спектрофотометре Evolution-300. Результаты измерения зависимости коэффициента пропускания образца от длины волны падающего света приведены на рисунке.

Нанопористые матрицы на основе силикатного стекла представляют собой совершенно особый, в своем роде уникальный, инструмент исследования физико-химических процессов в ограниченном объеме, соизмеримом с масштабом протекания процессов и размерами изучаемых объектов: ограниченность пространства и эффективный контакт со стенками пор обуславливают существенные особенности состояния вещества-наполнителя по сравнению со случаем его нахождения в свободном объеме.

Применение нанопористых стекол в настоящее время, главным образом, связано с прозрачностью таких стекол в видимой области спектра и возможностью получения образцов оптического качества [8, 9]. Оптическая плотность образцов в ближней УФ области спектра обусловлена рассеянием на структуре и поглощением материала. При рассмотрении оптических свойств пористых стекол в видимой области спектра и ближнем ИК-диапазоне поглощением в образцах можно пренебречь ввиду малости этой величины, а эффективные оптические постоянные образцов определяются только ослаблением излучения за счет рассеяния на пористой структуре и ее неоднородностях.

Уменьшение интенсивности света при его прохождении через нанопористое стекло и определение среднего размера неоднородностей

Рассмотрим плоскопараллельную пластину нанопористого стекла толщины h . Согласно (6)–(7), уравнение диффузии излучения и граничные условия к нему имеют вид

$$\frac{d^2 Q}{dx^2} - 3\gamma_a \gamma_s Q = 0; \quad (8)$$

$$\left[Q - \frac{1}{2\gamma_s} \frac{dQ}{dx} \right]_{x=0} = 2Q_e, \quad \left[Q + \frac{1}{2\gamma_s} \frac{dQ}{dx} \right]_{x=h}. \quad (9)$$

Интегрируя (8) с использованием (9), получаем

$$Q = A\text{ch}(\alpha x) + B\text{sh}(\alpha x), \quad (10)$$

где

$$A = \frac{\text{sh}(\alpha) + \beta\text{ch}(\alpha)}{(1 + \beta^2)\text{sh}(\alpha) + 2\beta\text{ch}(\alpha)} 2Q_e, \quad B = \frac{\text{ch}(\alpha) + \beta\text{sh}(\alpha)}{(1 + \beta^2)\text{sh}(\alpha) + 2\beta\text{ch}(\alpha)} 2Q_e, \quad \alpha = h\sqrt{3\gamma_a\gamma_s}, \quad \beta = 0,5\alpha/\gamma_s.$$

Зная величину Q , можно определить интенсивность излучения, распространяющегося под углом ϑ ,

$$I = \frac{1}{4\pi} \left(Q - \frac{\cos(\vartheta) dQ}{\gamma_s dx} \right) \quad (11)$$

и суммарную интенсивность излучения, распространяющегося в положительном направлении вдоль оси x ,

$$Q_+ = 2\pi \int_0^{\pi/2} I \sin(\vartheta) d\vartheta.$$

Используя (10)–(11), рассчитаем коэффициент пропускания $T = \frac{Q_+(h)}{Q_+(0)}$ пластины нанопористого стекла:

$$T = \frac{1}{\text{sh}(h\sqrt{3\gamma_a\gamma_s}) + \sqrt{\frac{\gamma_s}{3\gamma_a}} \text{ch}(h\sqrt{3\gamma_a\gamma_s})}. \quad (12)$$

Поскольку характерный размер D неоднородности рассматриваемой среды мал по сравнению с длиной волны падающего света, предполагаем, что рассеяние имеет релеевский характер, а коэффициент рассеяния обратно пропорционален четвертой степени длины волны падающего света, $\gamma_s = \frac{A}{\lambda^4}$. Коэффициент A и спектральная зависимость коэффициента поглощения γ_a подобраны так, чтобы обеспечить наилучшее согласование между теоретической и экспериментальной зависимостями коэффициента пропускания от длины волны света (рисунок). Полученное при этом значение коэффициента A позволяет приблизительно оценить характерный размер неоднородности среды $D \sim A^{1/3}$. Расчетное значение $D = 13,5$ нм близко к среднему размеру пор (17 нм) нанопористого стекла.

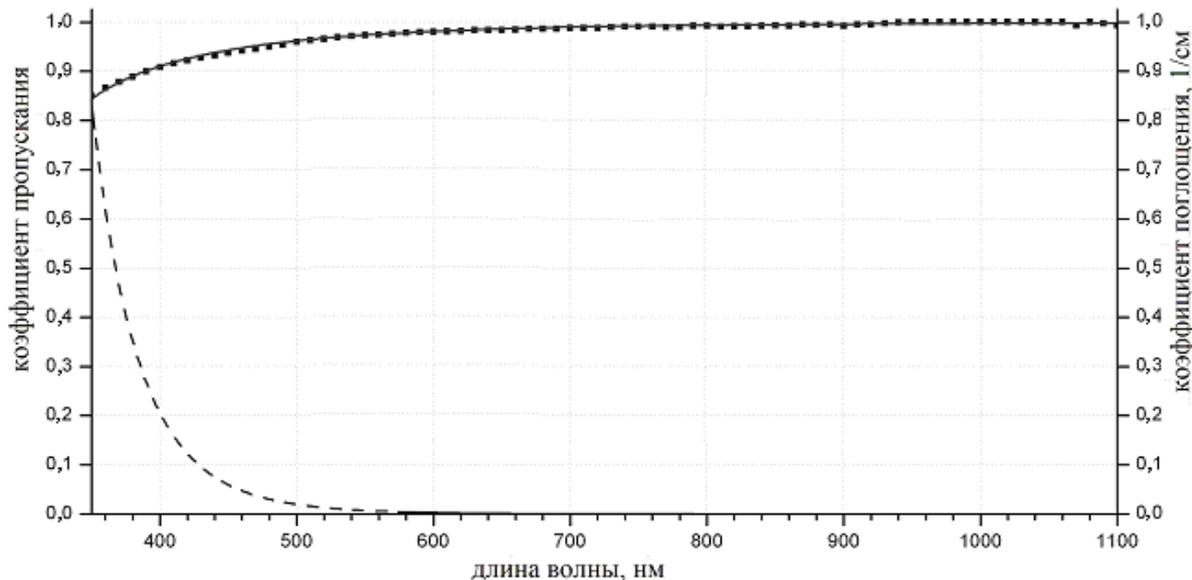


Рисунок. Экспериментальная и расчетная зависимости коэффициента пропускания, а также коэффициента поглощения нанопористого стекла от длины волны света: ■■■ коэффициент пропускания (эксперимент); — коэффициент пропускания (теория); --- коэффициент поглощения

Заключение

В работе предложен простой метод оценки среднего размера малых неоднородностей поглощающих и рассеивающих сред. Метод основан на сравнении экспериментальной и теоретической зависимостей коэффициента пропускания среды от длины волны. Метод опробован на образце нанопористого стекла и дал удовлетворительные результаты.

Литература

1. Ван де Хюлст. Рассеяние света малыми частицами. – М.: Издательство иностранной литературы, 1961.
2. Чандрасекар С. Перенос лучистой энергии. – М.: Издательство иностранной литературы, 1953.
3. Шифрин К.С. Рассеяние света в мутной среде. – М. –Л: ГИТТЛ, 1951.
4. Исимару А. Распространение и рассеяние волн в случайно-неоднородных средах. – М.: Мир, 1981. – Т. 1.
5. Зельдович Я.Б., Райзер Ю.П. Физика ударных волн и высокотемпературных гидродинамических явлений. – М.: Наука, 1966.
6. Schartl W. Light scattering from polymer solutions and nanoparticle dispersions. – Springer Laboratory, 2007. – 191 p.
7. Маслов В.П. Операторные методы. – М.: Наука, 1973.
8. Андреева О.В., Обыкновенная Е.И. Нанопористые матрицы – основа композитов различного назначения для оптических экспериментов // Труды Научно-исследовательского центра фотоники и оптоинформатики: сборник статей / Под редакцией И.П. Гурова и С.А. Козлова. – СПб: СПбГУ ИТМО, 2009. – С. 303–322.
9. Андреева О.В., Обыкновенная И.Е. Нанопористые матрицы НПС-7 и НПС-17 – возможность использования в оптическом эксперименте. Наносистемы: физика, химия, математика. – 2010. – Т. 1. – № 1. – С. 37–53.

- Альфимов Антон Владимирович* – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, Alfimov.anton@gmail.com
- Арысланова Елизавета Михайловна* – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, Elizabeth.aryslanova@gmail.com
- Вавулин Дмитрий Николаевич* – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, dima-vavulin@mail.ru
- Андреева Ольга Владимировна* – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат физ.-мат. наук, старший научный сотрудник, olga_andreeva@mail.ru
- Темнова Дарья Дмитриевна* – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, daren-ok@bk.ru
- Лесничий Василий Валерьевич* – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, kpnk@yandex.ru
- Альмяшев Вячеслав Исаакович* – Санкт-Петербургский государственный технологический институт (технический университет), ассистент, vac@mail.ru
- Кириллова Светлана Анатольевна* – Санкт-Петербургский государственный технологический институт (технический университет), ассистент, refractory-sveta@mail.ru
- Чивилихин Сергей Анатольевич* – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат физ.-мат. наук, доцент, sergey.chivilikhin@gmail.com
- Попов Игорь Юрьевич* – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор физ.-мат. наук, профессор, зав. кафедрой, popov@mail.ifmo.ru
- Гусаров Виктор Владимирович* – Санкт-Петербургский государственный технологический институт (технический университет), доктор химических наук, профессор, член-корреспондент РАН, зав. кафедрой, victor.v.gusarov@gmail.com

УДК 551.594:004.021:004.045:004.67:004.622

РАЗРАБОТКА МОДЕЛИ РЕШЕНИЯ ОБРАТНОЙ ЗАДАЧИ ВЕРТИКАЛЬНОГО ЗОНДИРОВАНИЯ ИОНОСФЕРЫ

А.Ю. Гришенцев, А.Г. Коробейников

Разработана модель решения обратной задачи вертикального зондирования ионосферы применительно к данным, полученным на ионосферных станциях типа «АИС-М». На основе разработанной модели реализовано программное приложение для автоматизированной обработки и анализа спектрограмм высотного зондирования ионосферы. Апробация программного продукта позволила выявить достаточно высокую эффективность предложенного метода, что позволяет позиционировать разработанную модель как эффективный способ решения обратной задачи высотного зондирования ионосферы.

Ключевые слова: ионосфера, вертикальное зондирование, спектрограмма, обратная задача, высотное распределение электронной плотности.

Введение

Исследование ионосферы является актуальной задачей для многих областей человеческой деятельности. Особенный интерес у исследователей вызывает изучение распространения электромагнитных волн в замагниченной плазме ионосферы [1] – как с точки зрения самих исследований процесса, так и с точки зрения применения в практической радиосвязи. Сущность обратной задачи радиочастотного зондирования состоит в нахождении ряда физических характеристик среды путем исследования влияния слоистой ионосферы на радиосигналы. Одной из ключевых характеристик ионосферы является высотное распределение электронной плотности. Наличие станций ионосферного зондирования по всему миру и их плотное сотрудничество через мировые центры данных позволяют получать информацию о глобальном распределении электронной плотности в ионосфере Земли. Наряду с методами радиочастотного зондирования в современных исследованиях активно применяется метод радиопросвечивания ионосферы [2]. Особенно актуальным (с точки зрения глобальных, регулярных наблюдений) метод радиопросвечивания стал после вывода на орбиты высокоорбитальных спутников GPS и ГЛОНАСС, а также низкоорбитальных GRACE, COSMIC, LEO и ряда других, с помощью которых на сегодня осуществляется более 5600 сеансов затменных измерений каждые сутки в более чем 2000 районах. Совокупность всех исследований позволяет получить практически в режиме реального времени данные о глобальном состоянии ионосферы Земли. Одной из проблем современных исследований при прогнозировании состояния ионосферы является недостаточность статистических наблюдений, что обусловлено относительной молодостью рассматриваемых методов. Таким образом, актуальной является задача рассмотрения архивных данных ионосферных исследований и их дополнительный анализ, позволяющий увеличить объемы статистической информации. В данной работе рассмотрена проблема моделирования высотного распределения электронной плотности на основании данных радиозондирования ионосферной станцией «АИС-М».

В рамках работы необходимо выполнить анализ спектрограмм вертикального зондирования ионосферы (СВЗИ), моделирование высотного распределения электронной плотности на основании данных радиозондирования ионосферной станцией «АИС-М». Задачами работы являются:

- разработка методов анализа СВЗИ;
- разработка обобщенной математической модели для решения обратной задачи с учетом особенностей СВЗИ;
- реализация разработанной математической модели в виде программы.

Анализ особенностей спектрограмм, полученных методом вертикального радиозондирования

Спектрограммы вертикального радиозондирования получаются в результате приема отраженного от ионосферы сигнала в диапазоне 1–20 МГц с минимальным шагом по частоте 1 кГц. Диапазон зондируемых высот составляет 50–1500 км, частота повторения импульсов 40 Гц. Мощность передатчика не менее 2,5 кВт. Полученные данные оцифровываются и сохраняются в сжатом виде.

Одна из сложностей, с которыми приходится сталкиваться при анализе ионограмм – зашумленность, вызванная локальными неоднородностями ионосферы, а в последнее время – лавинообразным ростом антропогенного заполнения радиочастотных диапазонов. Особенно заметным стало влияние радиозумов на СВЗИ с распространением цифровых каналов передачи информации, использующих сложные виды модуляции с большим числом низших гармоник. Таким образом, первой задачей для анализа СВЗИ является выделение высотных следов отражения. Решение данной задачи выполнялось с применением комплексных статистических методов обработки информации. Для предварительной локализации об-

ластей, в которых наиболее вероятно присутствие первых отражений (по простой траектории отражения в соответствии с [3] слои D, E, F_1) рассчитывалась высотная плотность распределения отраженных сигналов

$$p_j(h) = \frac{\sum_{i=0}^{i_{\max}-1} A_{i,j}}{S}, \quad (1)$$

где $A_{i,j}$ – амплитуда отраженного сигнала из матрицы СВЗИ \mathbf{A} с координатами i, j ; i – координата по горизонтали (соответствует частоте), j – координата по вертикали (соответствует высоте), S – число ненулевых элементов по горизонтали (т.е. из множества $A_{i,j}$, где $j = const, i = 0 \dots i_{\max} - 1$), причем для исключения деления на ноль полагаем, что $if(S = 0) S = 1$. Здесь необходимо отметить, что весь предварительный анализ производился в действующих высотах, это связано с тем, что расчет реальной высоты становится возможным только после определения высотного распределения электронной плотности в ионосфере. Выражение (1) хорошо локализует области с горизонтальным или близким к горизонтальному расположением отраженного следа, т.е. области, где величина $\frac{dh}{df}$ мала (f – частота, h – высота следа). Локализация областей с достаточно высокими значениями $\frac{dh}{df}$, обычно соответствующими критическим частотам слоя, где групповая скорость волны $t_{ГР} \rightarrow \infty$, по оценке величины $p_j(h)$ затруднительна. Введение величины горизонтальной плотности распределения позволяет решить задачу локализации следа отражений только на СВЗИ с относительно малым уровнем шума, что не представляет особенного интереса. Дальнейшая локализация отраженных следов в СВЗИ производилась с учетом значений $p_j(h)$ следующим образом (алгоритм 1):

- (а) осуществлялся проход по вертикальной координате матрицы \mathbf{A} ;
- (б) при встрече ненулевого линейного элемента $A_{i,jstart \dots jstop} \neq 0$ ($jstart \dots jstop$) определялось наличие ненулевых значений в матрице \mathbf{A} в элементах с координатами $A_{i \pm \Delta i, j \pm \Delta j}$, где $\Delta i, \Delta j$ – допуски по горизонтали и вертикали соответственно, значение Δi задается в виде константы, а Δj рассчитывается как функция $f(jstart, jstop)$;
- (с) если условие (б) выполнялось заданное число ns раз для элемента $A_{i,j} \neq 0$, то линейный элемент $A_{i,jstart \dots jstop} \neq 0$ считался следом отражения в СВЗИ.

В результате анализа СВЗИ по алгоритму 1 получается матрица \mathbf{M} , в которой ненулевые значения соответствуют различным компонентам отражений электромагнитной волны от ионосферы (рис. 1). Часть (б) указанного алгоритма была реализована в виде рекуррентной функции, что позволило учитывать особенности смещения следа отражения по высоте при движении по частоте, т.е. производить предварительную аппроксимацию паттерна следа отражения. Аппроксимация позволяет сделать частичное, предварительное выделение различных следов отражения. В общем виде решить задачу разделения на составляющие o - (обыкновенной), x - (необыкновенной) и z -компоненты (магнитоионной компоненты) на этом этапе анализа не представляется возможным.

В наиболее зашумленных СВЗИ перед выполнением указанного алгоритма производилась предварительная фильтрация путем выделения областей с высокими значениями энтропии. Оценку энтропии можно произвести следующим образом (алгоритм 2):

– матрица СВЗИ \mathbf{A} разбивается на несколько равных друг другу прямоугольных областей $B_{s,t}$ (для выравнивания ряд вертикальных векторов матрицы \mathbf{A} , соответствующих самым высоким частотам, и несколько горизонтальных векторов, соответствующих наибольшим значениям высоты, можно не рассматривать);

– каждая прямоугольная область $B_{s,t}$ разбивается на равные прямоугольники $X_{n,k}$, для которых рассчитывается суммарная амплитуда $X_{n,k} = \sum_i \sum_j (A_{i,j} : A_{i,j} \in X_{n,k})$, далее рассчитываются векторы плотности распределения для всех областей $B_{s,t}$, причем каждый элемент отдельного вектора плотности распределения для области $B_{s,t}$ рассчитывается для $X_{n,k} \in B_{s,t}$;

– для каждой области $B_{s,t}$ рассчитывается значение энтропии:

$$H = - \sum_l p B_l \log p B_l, \quad (2)$$

где pV_l – вектор плотности распределения амплитуд для области $B_{s,t}$.

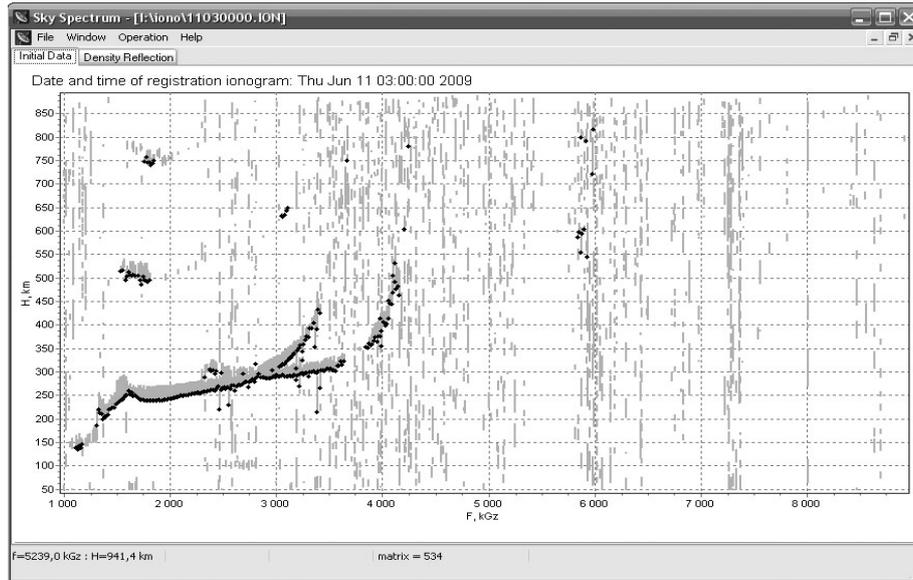


Рис. 1. Программная реализация фильтрации СВЗИ по алгоритму 1. Серым цветом обозначена исходная СВЗИ, черным – фильтрация (ненулевые элементы матрицы \mathbf{M})

На основании полученных значений энтропий для различных областей СВЗИ производится фильтрация путем отсеивания областей с наиболее высокими значениями энтропии. Рассмотренный метод фильтрации на основании оценки энтропии в связи с возможностью потери большого числа значимых данных применялся только в тех случаях, когда рассмотренные ранее методы локализации отраженных следов в СВЗИ не дают удовлетворительных результатов.

Использование фильтрации необходимо для начального определения локализации следов отражения в СВЗИ, конечное построение следов отражения наиболее целесообразно производить по исходной матрице СВЗИ \mathbf{A} , по алгоритму 1, с учетом зашумленности СВЗИ путем корректирования значений коэффициентов $\Delta i, \Delta j, ns$.

Определение высотных характеристик ионосферы по данным предварительного анализа СВЗИ

Наиболее интересными областями СВЗИ являются те участки, в которых $\left| \frac{dh}{df} \right|_{\max}$, причем области, в которых

$$\left| \frac{dh}{df} \right|_{\max}, \frac{dh}{df} > 0, \quad (3)$$

будут соответствовать критическим частотам слоя для f_o – o-компоненты, f_x – x-компоненты и f_z – z-компоненты электромагнитной волны (рис. 2). С учетом локализации области отражения и построения матрицы \mathbf{M} нахождение производной $\frac{dh}{df}$ не представляет сложностей. Кроме критических частот, по условию:

$$\left| \frac{dh}{df} \right|_{\max}, \frac{dh}{df} < 0 \quad (4)$$

можно определить f_B – гирочастоту слоя, которая, вообще говоря, может быть определена для конкретного географического места, исходя из значения B_0 в [Гц] у поверхности Земли [3]:

$$\begin{cases} f_B = 2,7993B(h), \\ B(h) = B_0 \left[1 - \frac{3h}{r_0} \right], \end{cases} \quad (5)$$

где h – высота над поверхностью Земли [км]; r_0 – радиус Земли в данном месте [км]; f_B – гирочастота [МГц]. Магнитную индукцию при ионосферных исследованиях принято выражать в [Гс], перевод в систему Си [Тл] может быть осуществлен с помощью следующего соотношения:

$$B[\text{Гс}] = 1 \times 10^{-4} B[\text{Тл}]. \quad (6)$$

Исходя из практического постоянства f_B для данного места (5), можно осуществить проверку полученных данных, применяя выражение, связывающее критические частоты обыкновенной и необыкновенной волны с гирочастотой [3]:

$$f_x^2 - f_x f_B = f_o^2. \quad (7)$$

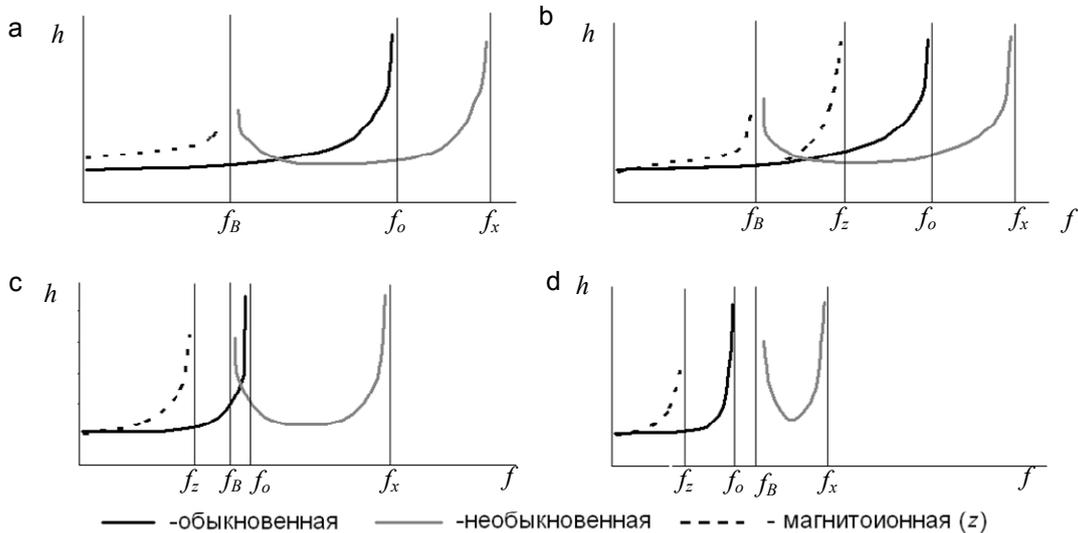


Рис. 2. Некоторые характерные виды СВЗИ при различных соотношениях частот:
 $b - f_o \gg f_B$; $c - f_o \approx f_B$; $d - f_o < f_B$

На основании полученных значений критических частот и следов отражения СВЗИ в виде матрицы \mathbf{M} вычисляется высотное распределение электронной плотности [м^{-3}]:

$$N_e(h) = 1,24 \times 10^{10} f_N, \quad (8)$$

где f_N – плазменная частота электронов [МГц]: для максимумов электронной плотности f_N будет соответствовать критической частоте f_o . По полученным точкам для высотного распределения электронной плотности с помощью стандартных методов аппроксимации можно получить гладкую кривую $N_e(h)$ [5].

Следующей задачей является расчет фактической высоты с учетом высотного распределения электронной плотности $N_e(h)$. Решить эту задачу можно по групповому времени распространения электромагнитной волны, с учетом прямого и обратного движения [4]:

$$t_{\text{ГР}} = \frac{2h_o}{c} + 2 \int_{h_o}^{h_{\text{и}}} \frac{dh}{\sqrt{1 - \frac{80,8 N_e(h)}{f^2}}}, \quad (9)$$

где h_o – высота расположения ионосферы над поверхностью Земли в данном месте (определяется по высоте отражения низших частот из матрицы \mathbf{M}); $h_{\text{и}}$ – высота ионосферы, для которой рассчитывается $t_{\text{ГР}}$; f – частота зондирующей электромагнитной волны; c – скорость света в вакууме.

Заключение

В ходе исследований произведен анализ СВЗИ с точки зрения особенностей шумовых составляющих, разработана математическая модель фильтрации СВЗИ. Полученные в результате фильтрации исходных СВЗИ данные позволяют производить расчет высотной характеристики $N_e(h)$.

Анализ данных ионосферной станции Воейково (Санкт-Петербург) за 2002–2003 г.г. показал, что разработанные методы фильтрации позволяют определить значение критической частоты $f_o \pm 0,16 f_o$ в 68% случаев, 23% СВЗИ были отсеяны на уровне фильтрации как неподдающиеся машинному анализу из-за высокого уровня шумов, в 9% СВЗИ критические частоты были определены с точностью менее 16%.

Решение проблем, возникающих при анализе СВЗИ ошибок и неопределенностей в силу высокого уровня шума, возможно с помощью взаимодействия мировой сети ионосферных станций, глобального радиозатменного мониторинга ионосферы и атмосферы Земли. Перспективным направлением зондирования ионосферы являются современные методы частотно-углового зондирования, позволяющие на стадии аппаратного анализа производить разложение отраженного сигнала на составляющие его компоненты, например, как это реализовано в Digisonde™ Portable Sounder (DPS) [6].

Литература

1. Гинзбург В.Л., Рухадзе А.А. Волны в магнитоактивной плазме. – 2-е изд. – М.: Наука, 1975. – 256 с.
2. Яковлев О.И., Павельев А.Г., Матюгов С.С. Спутниковый мониторинг Земли: Радиозатменный мониторинг атмосферы и ионосферы. – М.: Либроком, 2010. – 208 с.
3. Piggott W.R., K. Rawer. URSI handbook of Ionogram Interpretation and Reduction // INAG (Ionospheric Network Advisory Group) WORLD DATA CENTER A. National Academy of Sciences. – 2-d edition. – 1972. – 145 p.
4. Яковлев О.И., Якубов В.П., Урядов В.П., Павельев А.Г. Распространение радиоволн. – М.: ЛЕНАНД, 2009. – 496 с.
5. Демидович Б.П., Марон И.А. Основы вычислительной математики. – М.: Наука, 1970. – 664 с.
6. Background to Ionospheric Sounding [Электронный ресурс]. – Режим доступа: <http://ulcar.uml.edu/DPS.htm>, свободный. Яз. англ.

- Гришнцев Алексей Юрьевич* – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, tigerpost@ya.ru
- Коробейников Анатолий Григорьевич* – Санкт-Петербургский филиал учреждения Российской академии наук «Институт Земного магнетизма, ионосферы и распространения радиоволн им. Н.В. Пушкова РАН», доктор технических наук, профессор, зам. директора, Korobeynikov_A_G@mail.ru

УДК 681.5.11

ПРОГРАММИРУЕМЫЙ ФОРМИРОВАТЕЛЬ ТРАЕКТОРИИ ДВИЖЕНИЯ СЛЕДЯЩЕГО ЭЛЕКТРОПРИВОДА

С.Ю. Ловлин, М.Х. Цветкова, И.Н. Жданов

Рассматривается методика построения траектории движения следящего электропривода с ограничением скорости и ускорения. Особенность поставленной задачи – заранее неизвестна траектория задания, формирователь траектории (ФТ) реализуется в цифровом виде.

Ключевые слова: траектория движения, формирователь траектории, задатчик интенсивности, ограничение скорости, ограничение ускорения.

Введение

В системах регулирования положения важную роль играют законы управления движением, обеспечивающие оптимальное протекание процесса позиционирования. В ряде случаев классические законы управления следящими электроприводами, как, например, в установках лазерного раскроя листовых материалов или гравировки, не позволяют достичь требуемых динамических характеристик электромеханических систем – минимальной ошибки отработки заданной траектории движения при максимальном быстродействии, минимальном перерегулировании и ограничениях скорости и ускорения [1].

Данная проблема может быть решена посредством введения задатчика интенсивности (ЗИ) или ФТ на выходе канала задания. В общем случае задача формирования траектории решается при условии, что задание описывается кусочно-постоянной функцией и время между двумя скачкообразными изменениями задающего сигнала больше времени переходного процесса ФТ [2] (или ЗИ [1]). Другими словами, заранее рассчитывается траектория перехода из точки А в точку В, причем за все время переходного процесса точка В не меняется.

Такие ФТ удовлетворяют широкому кругу задач, но не могут быть использованы, например, в задачах слежения телескопов за небесными объектами. За время перехода телескопа от одного спутника к другому последний успевает изменить свое положение, причем его траектория заранее неизвестна. Отсюда возникает необходимость планирования траектории в реальном времени.

Постановка задачи

Задача состоит в разработке ФТ, обладающего следующими качествами: минимальное время переходного процесса, простота реализации, адаптация под изменяющиеся условия задания. Под простотой реализации понимается использование только таких функций, как сложение, умножение, деление. Это

необходимо для высокого быстродействия алгоритма ФТ. Также немаловажным является и цифровое исполнение ФТ.

В качестве входных данных ФТ использует:

z_n, z_{n-1}, z_{n-2} – текущее и два предыдущих значения задания;

$a_{n-1}, a_{n-2}, a_{n-3}$ – три предыдущих значения ФТ;

w_{\max} – максимальный прирост угла (произведение максимальной скорости на период дискретизации);

e_{\max} – максимальное дискретное ускорение (произведение максимального ускорения на период дискретизации в квадрате).

В расчетах, приведенных ниже, потребуются также следующие значения:

$\varepsilon = z_n - a_{n-1}$ – разница между заданием и предыдущим значением ЗТ;

$w_a = a_{n-1} - a_{n-2}$ – прирост выходного сигнала ЗТ;

$w_z = z_n - z_{n-1}$ – прирост сигнала задания;

$e_a = a_{n-1} - 2 \cdot a_{n-2} + a_{n-3}$ – дискретное ускорение выходного сигнала ЗТ;

$e_z = z_n - 2 \cdot z_{n-1} + z_{n-2}$ – дискретное ускорение сигнала задания;

$\Delta w = w_z - w_a$ – относительный прирост;

$\Delta e = e_z - e_a$ – относительное ускорение.

Все вышеприведенные величины измеряются в радианах (это не является обязательным, но необходимо для облегчения понимания нижеприведенных графиков).

Ограничение скорости и ускорения при скачке задания

Данная задача является упрощенной, так как задающий сигнал считается постоянным.

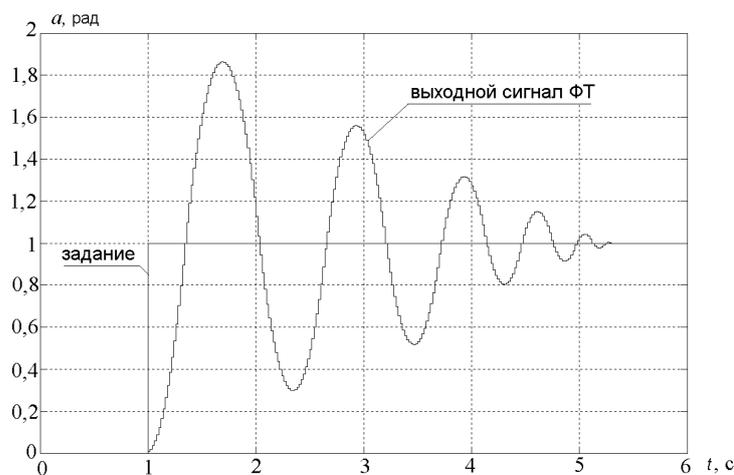
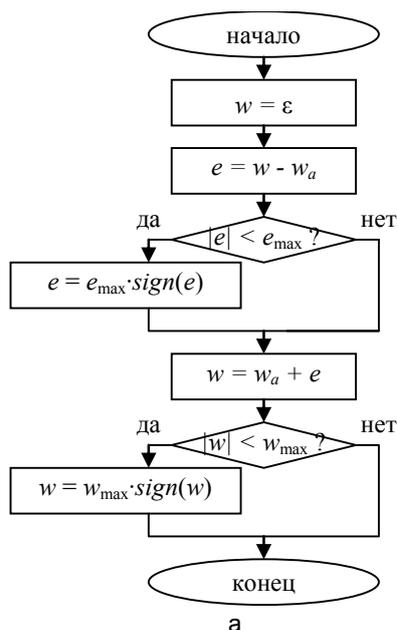


Рис. 1. Алгоритм ограничения скорости и ускорения (а) и результат его применения (б)

На рис. 1 изображен результат решения путем обычного ограничения скорости и ускорения. Анализируя этот график, видим, что переходный процесс носит колебательный характер, а это недопустимо. Причина такой реакции на скачок – ненулевое значение скорости сигнала ФТ при пересечении траектории задания.

Таким образом, необходимо добавить функцию торможения в ФТ, т.е. переходный процесс надо разделить на три этапа – разгон, движение с максимальной скоростью и торможение. Сложность заключается в поиске точки, в которой необходимо начать торможение. Сначала определим текущие значения угла и скорости через предыдущие:

$$w_n = w_{n-1} + e, \tag{1}$$

$$\alpha_n = \alpha_{n-1} + w_n. \tag{2}$$

Проанализировав (1) и (2), получим расчеты для вычисления скорости и угла в любой момент времени через начальные значения:

$$w_n = w_0 + n \cdot e; \quad (3)$$

$$\alpha_n = \alpha_0 + n \cdot w_0 + \frac{1}{2}n(n+1) \cdot e. \quad (4)$$

В конце торможения скорость изменения сигнала ФТ $w_n = 0$, а само значение ФТ α_n должно быть равно значению задания. Отсюда и вытекает условие начала торможения:

$$n = -\frac{w_0}{e}; \quad (5)$$

$$\alpha_0 = \alpha_n + \frac{1}{2} \left(\frac{w_0^2}{e} + w_0 \right),$$

где w_0 – скорость в данный момент времени; α_0 – предыдущее значение ЗТ; α_n – значение задания; e – тормозное ускорение; n – количество тактов вычислений, через которое скорость станет равной нулю.

Результат моделирования данного алгоритма ФТ изображен на рис. 2 (задание и выходной сигнал ФТ на участке от 0 до 2 с). Форма траектории напоминает латинскую букву *s*, откуда и название – S-образный ФТ.

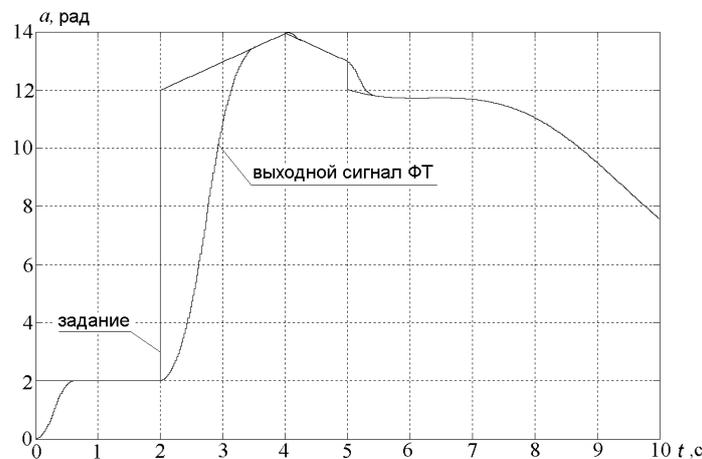


Рис. 2. Результат работы ФТ

Ограничение скорости и ускорения при любом виде задания

Выберем систему отсчета, в которой будет проводиться решение поставленной задачи. Совместим начало координат с текущим значением задания. Скорость и ускорение выбранной системы отсчета в абсолютной системе координат соответственно равны текущим скорости и ускорению задания. Таким образом, траектория задания преобразуется в ось абсцисс относительной системы координат. Нетрудно заметить, что решение такой задачи аналогично предыдущему, т.е. необходимо привести выходной сигнал ФТ из любой точки относительной системы отсчета в начало координат.

Проведем следующие замены:

$$\alpha_0 - \alpha_n = \varepsilon; \quad (6)$$

$$w_0 = \Delta w;$$

$$e = \Delta e.$$

Подставим (6) в (5):

$$n = -\frac{\Delta w}{\Delta e}; \quad (7)$$

$$\varepsilon = \frac{1}{2} \left(\frac{\Delta w^2}{\Delta e} + \Delta w \right).$$

Выражение (7) дает нам полное решение поставленной задачи о поиске точки начала торможения. Алгоритм цифрового ФТ приведен на рис. 3. Особенность цифрового исполнения ФТ заключается в том, что вычисления производятся в дискретные моменты времени. Таким образом, маловероятно, что тормозное ускорение будет приложено в точке начала торможения. Поэтому система прогнозирует ситуацию на следующем периоде дискретизации (переменные $\Delta w1$, $\Delta e1_{\max}$, $n1$), и если выполняется условие начала торможения, то ФТ формирует тормозное ускорение уже на текущем такте.

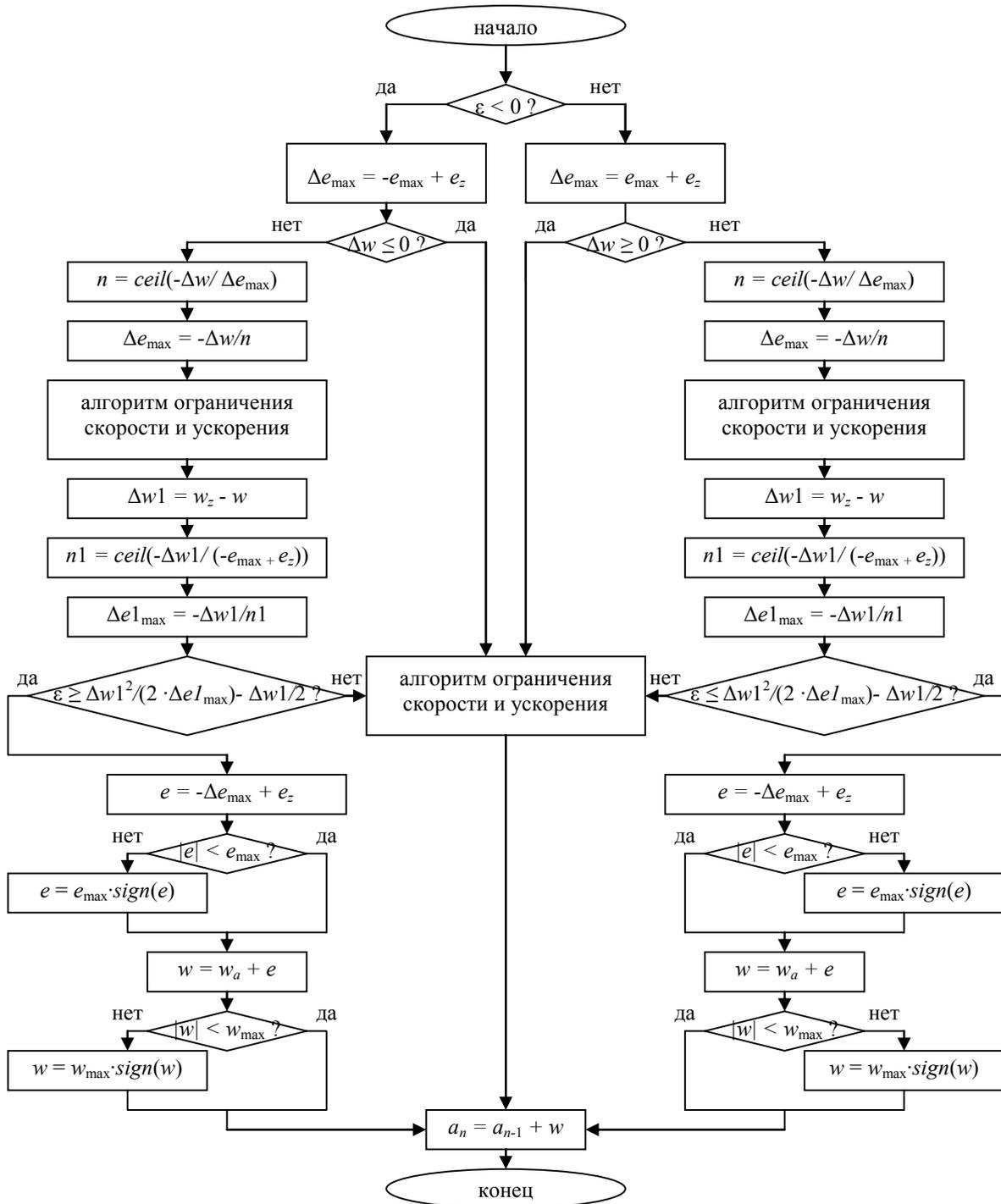


Рис. 3. Алгоритм формирователя траектории

В алгоритме используются следующие функции: $sign()$ – вычисление знака аргумента; $ceil()$ – округление аргумента вверх.

Выход на заданную траекторию должен быть выполнен за целое число тактов. Используя функцию округления $ceil()$, система ФТ вычисляет, с каким ускорением (близким к максимальному) необходимо проводить торможение.

В результате формирования траектории сложного задания (рис. 3) были получены скорости и ускорения, лежащие в пределах ограничения (рис. 4).

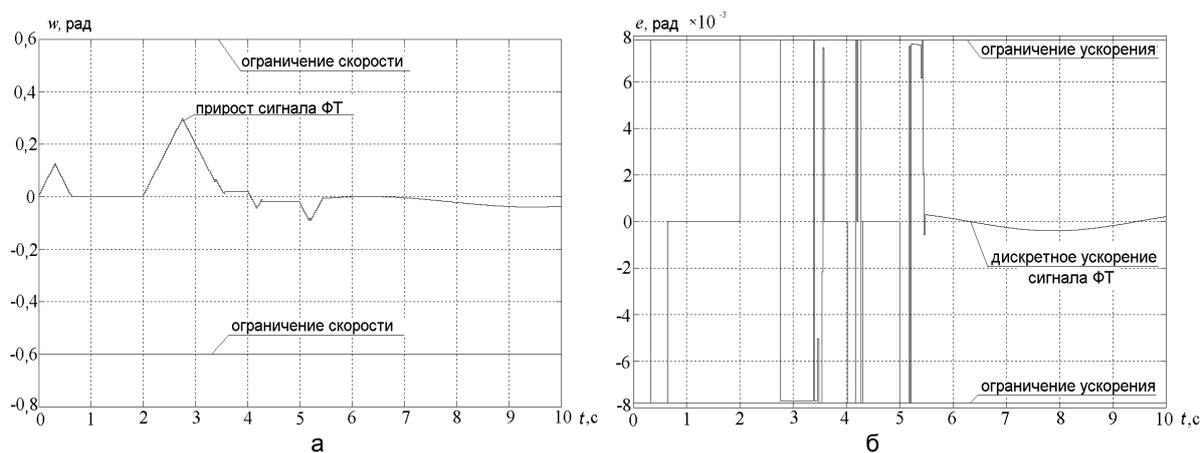


Рис. 4. Скорость (а) и ускорение (б) ФТ

Заключение

Разработанный алгоритм цифрового ФТ позволяет решать задачу ограничения скорости и ускорения в реальном времени при изменяющихся условиях задания с максимальными быстродействием и эффективностью. Данное решение, к примеру, подходит для следящих систем телескопов траекторных измерений. Но есть и один недостаток этого алгоритма – в некоторые моменты времени происходит скачкообразное изменение ускорения. Это требует скачкообразного изменения момента двигателя и приводит к перегрузке механической трансмиссии. При этом в нагрузке с упругой связью неизбежно будут возникать колебания. В некоторых случаях это недопустимо, и появляется необходимость повышения порядка гладкости траектории. В ближайшем будущем планируется решение данной проблемы.

Литература

1. Оптимальное управление движением при позиционировании и его моделирование в среде Matlab/Simulink / А.Г. Ильина, Д.В. Лукичев, А.А. Усольцев // Изв. вузов. Приборостроение. – 2008. – Т. 51. – № 6. – С. 63–67.
2. Использование параметрической аппроксимации при планировании траекторий движения аппаратов / Г.М. Довгоброд, Л.М. Клячко, А.В. Рогожников // Изв. вузов. Приборостроение. – 2009. – Т. 52. – № 9. – С. 11–17.

Ловлин Сергей Юрьевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, sergi-l@yandex.ru

Цветкова Мадина Хасановна

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, madina1986@bk.ru

Жданов Иван Николаевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, ассистент, zhdanov@ets.ifmo.ru

УДК 62.50

КОНТРОЛЬ ЗАТРАТ НА УПРАВЛЕНИЕ ПРИ ВОСПРОИЗВЕДЕНИИ ГАРМОНИЧЕСКИХ ЭКЗОГЕННЫХ ВОЗДЕЙСТВИЙ: ГРАМИАННЫЙ ПОДХОД

Д.С. Бирюков, А.В. Ушаков

Ставится задача контроля затрат управления при воспроизведении синтезируемой системой гармонических экзогенных воздействий с использованием грамианного подхода. Грамианный подход сформировался в рамках современной теории управления, опирающейся на векторно-матричный формализм метода пространства состояния. Подход аналитически устанавливает прямую связь установившейся составляющей движения технического объекта (ТО) в составе синтезируемой системы по выходу с вектором начального состояния источника гармонических экзогенных воздействий, причем эта связь осуществляется через матрицу подобия, являющуюся решением матричного уравнения Сильвестра. Задача получает прозрачное решение на сфере начальных состояний источника гармонических экзогенных воздействий в виде мажоранты и миноранты затрат управления как функции распределения мод, которое доставляется системе, образованной ТО и регулятором, при синтезе системы.

Ключевые слова: источник гармонического экзогенного воздействия, вектор начального состояния, объект управления, установившаяся составляющая, грамиан затрат управления, минорантная и мажорантная оценки.

Введение. Постановка задачи

Грамианные структуры в настоящее время активно используются отечественными и зарубежными специалистами для решения задач оптимального размещения управляющих органов и датчиков [1], оценки

межканальных связей [2], редуцирования моделей динамических объектов [3], синтеза управляющих воздействий [4]. Авторы данной работы решают проблему оценки затрат на управление с использованием грамианного подхода. Предлагаемая вниманию работа развивает научные положения, изложенные авторами в [5, 6]. Суть их состояла в том, что формирование желаемой структуры мод сопровождалось контролем затрат управления на сфере начальных состояний технического объекта (ТО) для множества возможных желаемых структур мод при условии выполнения отношения порядка применительно к длительности переходного процесса и возможного перерегулирования [7]. В работе [6] задача была проблемно расширена и сформулирована как задача контроля затрат управления при воспроизведении синтезируемой системой гармонических экзогенных воздействий с использованием грамианного подхода. В настоящей работе исследуется случай воспроизведения синтезируемой системой гармонических экзогенных воздействий. Грамианный подход сформировался в рамках современной теории управления, опирающейся на векторно-матричный формализм метода пространства состояния [8]. Подход устанавливает аналитическую связь компонентов движения ТО в составе синтезируемой системы по выходу и вектора начального состояния источника гармонического экзогенного воздействия. При этом наибольший практический интерес в решаемой задаче представляет установившаяся составляющая движения ТО. Алгоритмически указанная выше связь опирается на концепцию векторно-матричного подобия, матрица которого является решением матричного уравнения Сильвестра. Выход на оценку затрат управления при воспроизведении системой установившегося движения по выходу опирается на формирование закона управления, записанного в аддитивной форме, одна часть которого доставляет системе требуемое качество воспроизведения гармонического экзогенного воздействия, а вторая – требуемое качество выхода системы на установившуюся составляющую движения. Очевидно, в установившемся режиме в законе управления остается только первая его составляющая, что позволяет связать управление с начальным состоянием источника гармонического экзогенного воздействия. Финальный этап алгоритма решения задачи, вынесенной в название данной работы, состоит в формировании грамиана затрат управления с использованием матричного уравнения Ляпунова с последующим сингулярным разложением грамиана и выделением на алгебраическом спектре его сингулярных чисел минимального и максимального компонентов (миноранты и мажоранты). Выделение соответствующих этим компонентам алгебраического спектра сингулярных чисел левого сингулярного базиса завершает решение задачи, придавая ей геометрическую трактовку в виде портрета затрат управления.

Связь движения динамической системы с начальным состоянием источника гармонического экзогенного воздействия

Рассмотрим объект управления (ОУ)

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t), \quad x(t)|_{t=0} = x(0), \quad \varepsilon(t) = g(t) - y(t). \tag{1}$$

Источник конечномерного экзогенного воздействия описывается как

$$\dot{z}(t) = Ez(t), \quad z(t)|_{t=0} = z(0), \quad g(t) = Pz(t). \tag{2}$$

Для источника гармонического воздействия матрицы E , P и $z(0)$ имеют вид

$$E = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix}, \quad P = [1 \quad 0], \quad z(0) = [z_1(0) \quad z_2(0)]^T,$$

а выход источника становится равным $g(t) = z_1(0)\cos \omega t = z_2(0)\sin \omega t$.

Для ОУ требуется синтезировать закон управления (ЗУ), обеспечивающий в замкнутой системе требуемое распределение мод и соответствия выхода системы в установившемся режиме задающему воздействию

$$u(t) = u_\lambda(t) + u_g(t) = K_g g(t) - Kx(t). \tag{3}$$

Объединением ОУ и ЗУ получим систему управления (СУ):

$$\dot{x}(t) = Fx(t) + Gg(t), \quad y(t) = Cx(t), \tag{4}$$

где

$$F = A - BK, \quad G = BK_g. \tag{5}$$

Сконструируем агрегированную систему с вектором состояния $\tilde{x}(t) = [x^T(t) \quad z^T(t)]^T$, для которой справедливо представление

$$\dot{\tilde{x}}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} Fx(t) + GPz(t) \\ 0 \cdot x + Ez(t) \end{bmatrix} = \begin{bmatrix} F & GP \\ 0 & E \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} = \tilde{F}\tilde{x}(t), \quad \tilde{x}(0) = \begin{bmatrix} x(0) \\ z(0) \end{bmatrix}. \tag{6}$$

Решение системы (6) в явной форме принимает вид

$$\tilde{x}(t) = e^{\tilde{F}t} \tilde{x}(0). \tag{7}$$

Получим ряд выражений для компонентов исходной системы через компоненты агрегированной системы:

$$x(t) = Ix(t) + 0z(t) = [I \ 0]\tilde{x}(t), \quad x(t) = \tilde{C}_x\tilde{x}(t) = \tilde{C}_xe^{\tilde{F}t}\tilde{x}(0), \quad (8)$$

$$y(t) = Cx(t) + 0z(t) = [C \ 0]\tilde{x}(t), \quad x(t) = \tilde{C}_y\tilde{x}(t) = \tilde{C}_ye^{\tilde{F}t}\tilde{x}(0), \quad (9)$$

$$\varepsilon(t) = -Cx(t) + Pz(t) = [-C \ P]\tilde{x}(t), \quad x(t) = \tilde{C}_\varepsilon\tilde{x}(t) = \tilde{C}_\varepsilon e^{\tilde{F}t}\tilde{x}(0), \quad (10)$$

$$z(t) = 0x(t) + Iz(t) = [0 \ I]\tilde{x}(t), \quad x(t) = \tilde{C}_z\tilde{x}(t) = \tilde{C}_ze^{\tilde{F}t}\tilde{x}(0). \quad (11)$$

Для цели дальнейших исследований сформулируем утверждение.

Утверждение 1. Если матрицы F, E, G, P связаны уравнением Сильвестра

$$TE - FT = GP, \quad (12)$$

то матричная экспонента $e^{\tilde{F}t}$ представима в форме

$$e^{\tilde{F}t} = \begin{bmatrix} e^{Ft} & Te^{Et} - e^{Ft}T \\ 0 & e^{Et} \end{bmatrix}. \quad \square (13)$$

Доказательство утверждения приведено в [9]. ■

Пользуясь выражением (13), для движения агрегированной системы можно записать

$$\tilde{x}(t) = \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} e^{Ft} & Te^{Et} - e^{Ft}T \\ 0 & e^{Et} \end{bmatrix} \begin{bmatrix} x(0) \\ z(0) \end{bmatrix}. \quad (14)$$

Тогда уравнения движения замкнутой системы (4) приобретают вид

$$x(t) = e^{Ft}x(0) + (Te^{Et} - e^{Ft}T)z(0), \quad y(t) = Cx(t). \quad (15)$$

Представление (15) для вектора состояния $x(t)$ позволяет выделить в движении системы по этому вектору свободную, вынужденную, установившуюся и переходную составляющие:

$$x_{св}(t) = e^{Ft}x(0), \quad (16)$$

$$x_{в}(t) = (Te^{Et} - e^{Ft}T)z(0), \quad (17)$$

$$x_{у}(t) = \lim_{t \rightarrow \infty} x_{в}(t) = Te^{Et}z(0), \quad (18)$$

$$x_{пер}(t) = x_{у}(t) - x_{в}(t) = e^{Ft}Tz(0). \quad (19)$$

Для цели дальнейших исследований в силу постановки задачи сосредоточим внимание на установившейся составляющей движения. Очевидно, что движение в установившемся режиме системы (4) зависит от решения уравнения Сильвестра (12) относительно матрицы T , матрицы состояния E источника гармонического экзогенного воздействия и его вектора начального состояния $z(0)$.

С тем, чтобы наполнить содержанием матрицу T , формально являющуюся решением уравнения Сильвестра (12), сформулируем следующее утверждение.

Утверждение 2. Матрица T – решение уравнения Сильвестра (12) – представляет собой матрицу подобия, связывающую установившуюся составляющую $x_{у}(t)$ движения системы и свободное движение $z(t)$ источника гармонического экзогенного воздействия (2). □

Доказательство. Для свободного движения источника гармонического экзогенного воздействия в силу (2) можно записать:

$$z(t) = z(t), z(0) = e^{Et}z(0), \quad g(t) = Pz(t) = Pe^{Et}z(0). \quad (20)$$

Сопоставляя (18) и (20) для вектора $z(t)$, можно записать

$$x_{у}(t) = Tz(t). \quad \blacksquare (21)$$

Основной результат. Грамиан затрат на управление, минорантные и мажорантные эллипсоидные оценки

Для оценки затрат на управление рассмотрим вектор управления $u(t)$ (3) в установившемся режиме, так что $x(t) = x_{у}(t)$. Тогда с учетом (18) и (20) для вектора $g(t)$ вектор управления $u(t)$ (3) представим в форме

$$u(t) = K_g g(t) - Kx_{у}(t) = K_g Pe^{Et}z(0) - KTe^{Et}z(0) = (K_g P - KT)e^{Et}z(0). \quad (22)$$

Сформируем грамиан затрат на управление по схеме, изложенной в [5].

$$\|U_{[0,t]}\|^2 = \int_0^t U^T(\tau)U(\tau)d\tau = z^T(0)W_U(t)z(0), \quad (23)$$

где

$$W_U(t) = \int_0^t e^{E^T \tau} (K_g P - KT)^T (K_g P - KT) e^{E \tau} d\tau. \quad (24)$$

При $t \rightarrow \infty$ становится справедливым представление

$$\lim_{t \rightarrow \infty} W_U(t) = W_U. \quad (25)$$

Тогда для затрат на управление при $t \rightarrow \infty$ можно записать

$$\lim_{t \rightarrow \infty} \|U_{[0,t]}\|^2 = \|U_{[0,\infty]}\|^2 = z^T(0) W_U z(0). \quad (26)$$

В выражениях (25) и (26) матрица W_U именуется грамианом затрат на управление.

Для цели алгебраизации задачи сформулируем следующее утверждение.

Утверждение 3. Грамиан W_U затрат на управление удовлетворяет матричному уравнению Ляпунова

$$E^T W_U + W_U E = -(K_g P - KT)^T (K_g P - KT). \quad \square(27)$$

Доказательство утверждения строится по схеме, приведенной в [1], дополненной процедурой экспоненциальной регуляризации, состоящей в использовании приближенного равенства $s \cong s + \varepsilon$, где ε – бесконечно малая величина, максимально приближенная к «машинной ε », в соответствии с чем грамиан $W_U = \lim_{t \rightarrow \infty} (W_U(t))$ удовлетворяет уравнению Ляпунова

$$W_U E + E^T W_U = -(K_g P - KT)^T (K_g P - KT).$$

При наличии грамиана W_U в силу выражения (24) можно осуществить оценку затрат $\|U\|_\infty$ на управление установившимся движением системы на сфере начальных состояний источника гармонического экзогенного воздействия $\|z(0)\| = const$ в форме мажоранты $\|U_{[0,\infty]}\|_M$ и миноранты $\|U_{[0,\infty]}\|_m$ с использованием сингулярного разложения грамиана W_U в форме

$$\alpha_m^{1/2} \{W_U\} \|z(0)\| = \|U_{[0,\infty]}\|_m \leq \|U_{[0,\infty]}\| = (z^T(0) W_U z(0))^{1/2} \leq \|U_{[0,\infty]}\|_M = \alpha_M^{1/2} \{W_U\} \|z(0)\|, \quad (28)$$

где $\alpha_m \{W_U\}$, $\alpha_M \{W_U\}$ – соответственно минимальное и максимальное сингулярные числа грамиана W_U .

Таким образом, контроль затрат на управление при конечномерном экзогенном воздействии может быть осуществлен с использованием следующего алгоритма.

Алгоритм.

1. Сформировать векторно-матричное описание (ВМО) объекта управления в форме (1).
2. Задать источник конечномерного (гармонического) экзогенного воздействия с помощью ВМО в форме (2).
3. Сформировать требования к качеству процессов проектируемой системы в переходном и установившемся режиме для случая источника конечномерного экзогенного воздействия вида (2), отобразив их на структуру мод матрицы состояния системы, назначив их носителем матрицу Γ состояния модальной модели, задаваемой наблюдаемой парой матриц $(\Gamma, H : \dim H = \dim B^T)$.

4. Сформировать матрицу K обратной связи по вектору состояния ОУ закона управления (ЗУ) (3) методами модального управления в форме $K = HM^{-1}$, где M – матрица подобия отношения $M\Gamma = FM$ – ищется из уравнения Сильвестра

$$M\Gamma - AM = -BH. \quad (29)$$

5. Сформировать матрицу прямых связей K_g по вектору экзогенного воздействия $g(t)$ из условия единичного замыкания системы, приводящего к соотношению

$$K_g = \arg \{ \Phi(s) \}_{s=0} = C(sI - F)^{-1} B K_g = I \} = -(CF^{-1}B)^{-1}.$$

6. Решить уравнение Сильвестра (12) относительно матрицы T .
7. Решить уравнение Ляпунова (27) относительно грамиана W_U .
8. Построить оценки затрат на управление системой в установившемся режиме при заданном экзогенном воздействии в форме мажорант и минорант этих затрат с использованием соотношения (28).
9. Провести анализ полученных оценок. Вернуться в пункт 3 алгоритма с целью модификации структуры мод. Из процедуры выйти по достижении минимальных значений мажоранты $\|U_{[0,\infty]}\|_M$.

Примечание 1.

Пункт 6 алгоритма можно задать в явном виде с учетом вида матрицы E .

Запишем матрицу T в столбцовой форме:

$$T = [T_1 \ T_2].$$

Тогда уравнение (12) примет вид:

$$[-T_2\omega \ T_1\omega] - F[T_1 \ T_2] = [G \ 0],$$

а столбцы матрицы T можно задать в виде следующих выражений:

$$T_1 = -\frac{1}{\omega} F \left(WI + \frac{1}{\omega} F^2 \right)^{-1} G;$$

$$T_2 = -\left(WI + \frac{1}{\omega} F^2 \right)^{-1} G.$$

Примечание 2.

Нетрудно видеть, что матрица e^{Et} в силу своей структуры

$$e^{Et} = \begin{bmatrix} \cos \omega t & \sin \omega t \\ -\sin \omega t & \cos \omega t \end{bmatrix}$$

является ортогональной, и поэтому умножение на нее произвольных матриц слева и справа не меняет спектр сингулярных чисел сомножителя, а произведения $e^{E^T t} e^{Et}$ и $e^{Et} e^{E^T t}$ совпадают. Это наблюдение позволяет по-новому взглянуть на выражения (21)–(24) для вычисления грамиана затрат на управление.

Пример

Рассмотрим исходную неустойчивую систему:

$$A = \begin{bmatrix} 0 & 1 \\ -3 & 4 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; C = [1 \ 0].$$

Источник гармонического воздействия зададим следующим образом:

$$E = \begin{bmatrix} 0 & 2 \\ -2 & 0 \end{bmatrix}; P = [1 \ 0]. \text{ Таким образом, } \omega = 2.$$

Оценим затраты на перевод системы из начального состояния $x(0) = [1 \ 1]^T$ в начало координат с условием, что желаемые моды синтезируемой устойчивой системы находятся на единичной сфере. Изменяя структуру мод, можем найти оптимальный с точки зрения затрат на управление набор желаемых мод.

Моделирование представленной задачи в среде MATLAB дает результаты, представленные на рисунке.

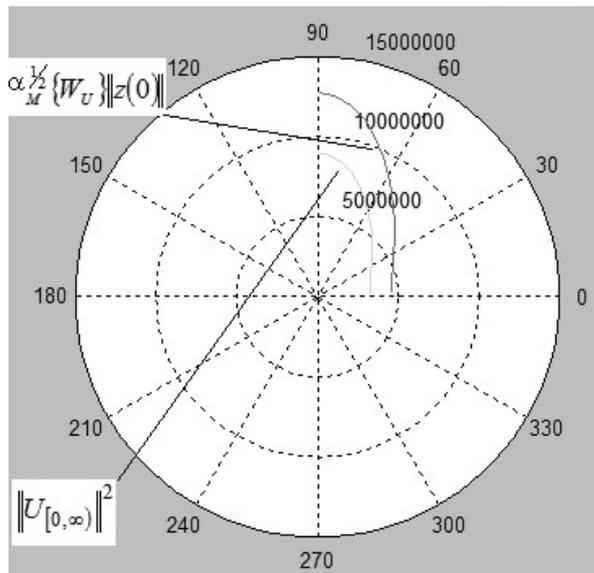


Рисунок. Зависимость затрат на управление от сектора локализации желаемых мод

На рисунке представлена зависимость затрат на управление от сектора локализации желаемых мод. Видно, что в данном случае оптимальным с точки зрения затрат на управление является выбор мод в соответствии со стандартным полиномом Ньютона (в нулевом секторе локализации).

Заключение

В работе предложены алгоритмы формирования затрат на управление установившимся движением системы, порождаемого гармоническим экзогенным воздействием. Решение задачи построено на использовании экстремальных элементов алгебраического спектра сингулярных чисел грамиана затрат на управление. Тем не менее, авторы считают, что «тонкое» решение этой проблемы требует также исследования структуры элементов геометрического спектра левого сингулярного базиса этого грамиана.

Литература

1. Schneiders M. Using Gramian Theory for Actuator and Sensor Placement. – Technische Universitet Eindhoven, 2004.
2. Conley A., Salgado M. Gramian based interaction measure // Proceedings of the 39th IEEE Conference on Decision and Control. – 2000. – V. 5. – P. 5020–5022.
3. Sorensen D., Antoulas A. The Sylvester equation and approximate balanced reduction // Linear Algebra and its Applications. – 2002. – P. 671–700.
4. Краснощеченко В.И. Синтез управления в задаче быстрогодействия с использованием метода модельного прогнозируемого управления // Мехатроника, автоматизация, управление. – 2008. – № 10. – С. 2–8.
5. Бирюков Д.С., Слита О.В., Ушаков А.В. Грамианские технологии оценки затрат на управление в задаче обеспечения желаемой структуры мод и их робастности // Изв. вузов. Приборостроение. – 2009. – Т. 52. – № 11. – С. 32–37.
6. Бирюков Д.С., Ушаков А.В. Контроль затрат на управление при воспроизведении полиномиальных экзогенных воздействий: грамианный подход // Материалы конференции «Управление в технических системах УТС-2010». – СПб: Концерн ЦНИИ «Электроприбор». – 2010. – С. 56–60.
7. Кострикин А.И. Введение в алгебру. Основы алгебры. – М.: Физматлит, 2004.
8. Гантмахер Ф.Р. Теория матриц. – М.: Наука, 1996.
9. Ушаков А.В. Модальные оценки качества процессов в линейных многомерных системах при внешних конечномерных воздействиях // Автоматика и телемеханика. – 1992. – № 10. – С. 72–82.

Бирюков Дмитрий Сергеевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, dbiryukov@list.ru

Ушаков Анатолий Владимирович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, ushakov-AVG@yandex.ru

УДК 004.423.4

**ОСНОВНЫЕ ПРИНЦИПЫ РЕШЕНИЯ ЗАДАЧИ ПРЕОБРАЗОВАНИЯ
ОБЪЕКТНО-ОРИЕНТИРОВАННОГО КОДА В ФОРМАТ RDF СРЕДСТВАМИ
СЕМАНТИЧЕСКОГО АНАЛИЗА**

А.В. Зараковский, С.В. Клименков, Н.И. Ткаченко, А.Е. Харитонова

Описано решение задачи преобразования исходного кода на объектно-ориентированном языке в формат RDF средствами семантического анализа. Рассмотрены основные принципы семантического анализа объектно-ориентированного кода и предложен архитектурный прототип программного продукта, осуществляющего преобразование кода в RDF.

Ключевые слова: код, RDF, семантический анализ, машинный анализ, триплет, грамматика, синтаксическое дерево, лексер, парсер, Java, ANTLR.

Введение

Одним из обязательных условий успешной разработки и сопровождения любого программного продукта является удобочитаемость его исходного кода. Стремясь добиться этого, программисты стараются делать код более логичным и понятным, именуя должным образом переменные и методы, используя правильно подобранные типы и классы, снабжая код комментариями. Все это говорит о том, что код является отличным источником информации о самом себе.

Программист, читая чужую программу с подробными комментариями, без труда поймет, что делает каждый метод, для чего предназначена любая переменная или же любой объект. Человек способен анализировать текст программы, распознавая в нем целостные куски информации. На самом деле в этот момент он осуществляет разбор кода на информационные блоки и сопоставление получаемых фактов друг с другом, так что, по сути, он строит зависимости в виде триплетов «объект – предикат – субъект» и оперирует уже ими. В качестве примеров можно привести обычные мысли вроде «переменная представляет целое число» или «метод запускается без параметров». В случае машинного анализа все не так просто. Машина не обладает разумом и способностью к анализу, а значит, не сможет самостоятельно провести разбор кода, как это делает человек. Таким образом, для успешной реализации машинного анализа кода необходимо выбрать формат представления данных и разработать специальную программу, которая осуществит разбор кода и приведет его к нужному программисту виду.

Язык описания ресурсов

В рамках данной работы в качестве формата, в который осуществляется преобразование программного кода, выбран Resource Description Framework (RDF). RDF – это разработанная консорциумом Всемирной паутины модель для представления данных, входящая в концепцию семантической паутины, а в особенности – метаданных. Метаданные – это информация о данных. Зачастую это структурированные данные, представляющие собой характеристики описываемых сущностей для целей их идентификации, поиска, оценки и управления ими. Важной особенностью формата RDF является то, что он описывает ресурсы в виде, пригодном (и достаточно удобным) для машинной обработки.

Ресурсом в RDF может быть любая сущность – информационная (сайт, программа, изображение) или неинформационная (человек, город, любое абстрактное понятие). Утверждение о ресурсе строится в виде триплетов и имеет вид «субъект – предикат – объект». Множество RDF-утверждений образует ориентированный граф, в котором вершинами являются субъекты и объекты, а предикаты являются ребрами.

RDF сам по себе представляется не каким-либо форматом файла, а абстрактной моделью. Для работы с RDF в основном используются несколько форматов, в число которых входят RDF/XML, RDFa, N3. Все они уже имеют определенные правила записи и структуру [1].

Семантический анализ программного кода

Основные принципы. Определившись с форматом хранения данных, можно перейти к основной задаче – собственно семантическому анализу программного кода. Для начала рассмотрим основные аспекты семантики любого объектно-ориентированного языка программирования.

Контекстно-свободная грамматика. Любой язык программирования имеет свой алфавит, который целиком состоит из терминальных и нетерминальных символов. Терминальный символ – это любой

имеющий конкретное известное значение символ, как, например, цифра или буква. Нетерминальный символ – это элемент конструкции языка, не имеющий заранее известного значения, как, например, формула или команда. Нетерминальные символы состоят из множества терминальных и в итоге к ним сводятся.

Язык программирования описывается набором правил, выделяющих некоторое подмножество известных символов из множества слов конечного алфавита языка. Такой набор правил называется грамматикой языка и может как задавать правила определения правильности построения слова языка, так и позволять построить любое новое слово языка. Первые являются распознающими (или аналитическими) грамматиками, вторые же являются порождающими.

Грамматика, левая часть правил которой состоит целиком из нетерминальных символов, называется контекстно-свободной [2].

Расширенная форма Бэкуса-Наура. Расширенная форма Бэкуса-Наура (РБНФ) – это формальная система определения синтаксиса, в которой одни синтаксические зависимости определяются через другие. Используется для описания контекстно-свободных грамматик.

Правило в РБНФ имеет вид «идентификатор = выражение.», где идентификатор – это имя нетерминального символа, а выражение – это соответствующая правилам РБНФ комбинация терминальных и нетерминальных символов. Точка в конце – это специальный символ, указывающий на завершение правила.

Набор возможных конструкций РБНФ очень невелик и состоит лишь из конкатенации символов, выбора, условного вхождения и повторения. Также при записи правила РБНФ можно использовать группирующие скобки для определения сложных правил [3].

Синтаксическое дерево. Объектно-ориентированный код, будучи правильно оформленным, сам по себе предоставляет отличные возможности для семантического анализа. При соблюдении правил именования сущностей в коде и его правильной структуры всегда существует возможность получить объектную модель, отражающую организацию кода. Каждый объект в программе можно рассматривать атомарно, не вдаваясь в подробности его реализации. Это называется абстрагированием. В свое же время каждый объект может включать в себя множество других объектов, которые также можно рассматривать как конечную атомарную сущность. Элементарные типы, которые помимо других объектов могут включаться в объект, всегда являются конечными и не имеют возможности инкапсулировать в себе что-либо. В данном случае при употреблении термина «объект» под объектом все же понимается не экземпляр класса, как можно было подумать, а некий атомарный блок информации – метод, класс, объявление переменной, блок комментариев или даже программа целиком. Каждый объект каким-то образом связан с другими, например, имеет некую вложенность в другой объект.

Рассмотрим следующий код:

```
public class Hello {
    public static void main(String[] args) {
        String str = "Hello, world!";
        System.out.println( str );
    }
}
```

Эта программа объявляет переменную *str*, присваивает ей значение, а затем вызывает метод вывода ее содержимого в поток *out*.

При анализе этого кода можно выделить класс *Hello*, в котором объявлен метод `void main(String[] args)`, в котором объявлена переменная *String str*, которой присваивается значение «Hello, world!», и вызывается метод `println()` с входным параметром *str* объекта *out* класса *System*. Соответственно, это можно и нужно отразить не словами, а в виде некой формализованной модели, которая покажет вложенность кода, значения, типы и сами сущности. Наилучшим образом для этого подходит двоичное дерево, такое, что каждый его узел может быть законченным, т.е. отражать некое детерминированное значение или поле любого примитивного типа, или расходиться вниз, демонстрируя, что элемент не является заключительным в иерархии. Примерное такое дерево, построенное по вышеприведенному коду, можно увидеть на рис. 1.

Таким образом, при семантическом анализе объектно-ориентированного кода можно составить иерархию объектов и элементарных типов, а также провести зависимости и связи между ними. Такую иерархию представляет абстрактное синтаксическое дерево (АСД) или *Abstract Syntax Tree (AST)*, листья которого представляют элементарные типы в коде, а различные ветви – объекты, представленные в коде. АСД – это конечное, помеченное, ориентированное дерево, в котором вершины сопоставлены с операторами языка программирования, а листья – с соответствующими операндами. Таким образом, листья являются лишь пустыми операторами и представляют только переменные и константы. Количество ветвей, как и количество листьев, не ограничено, однако дерево всегда имеет одну вершину – объект, представляющий программу целиком.

Применение контекстно-свободной грамматики при разборе кода приводит к получению абстрактного синтаксического дерева, в котором элементы не определяются конкретной грамматикой разбираемого языка. Классическим примером являются ограничительные скобки в языковых конструкциях –

группировка операндов в АСД явно задается структурой дерева, а ограничивающие скобки вообще отсутствуют в разборе, так как не влияют на АСД. АСД со свободной грамматикой отличается от дерева разбора, т.е. дерева с конкретной грамматикой, тем, что в нем отсутствуют ребра и узлы для тех синтаксических правил, которые никак не влияют на семантику программы. Для языка с контекстно-свободной грамматикой составление АСД является достаточно тривиальной задачей. Большинство правил грамматики создают вершину, а символы в правиле становятся ребрами. Правила, которые ничего не привносят в АСД, такие, как, например, группирующие, просто заменяются в вершине одним из своих символов. Кроме того, анализатор может создать полное дерево разбора и затем пройти по нему, удаляя узлы и ребра, которые не используются в абстрактном синтаксисе, чтобы получить АСД.

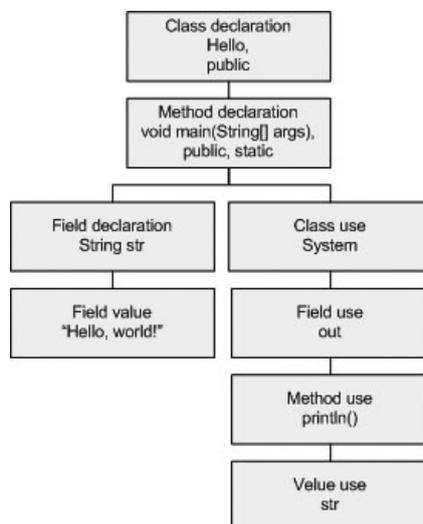


Рис. 1. Двоичное дерево разбора кода

В конечном итоге после разбора кода нас интересует, само собой, не АСД, а структура программы, откуда удалены все ненужные звенья дерева [4].

Решение задачи разбора кода

Построить семантическую модель кода и дерево разбора должны программа-лексер и программа-парсер, понимающие грамматику выбранного языка программирования, проходя от вершины к конечным операндам. Лексер – это программный модуль, осуществляющий разбор текста по заданным грамматическим правилам и генерирующий поток токенов, т.е. нетерминальных символов языка программирования. Например, токенами языка Java для полной грамматики языка будут являться класс, метод, блок комментариев, блок кода. Парсер – это программный модуль, который, основываясь на поступающих из потока лексера токенах, по заданным правилам строит абстрактное дерево разбора. Лексер должен выделять в коде следующие сущности, которые описываются лексическими правилами или грамматикой:

- функции, методы и процедуры, а также их вызовы;
- объекты и их вложенность;
- поля элементарных типов;
- комментарии, условные комментарии и документирующие комментарии;
- директивы;
- блоки кода и управляющие конструкции;
- различные атрибуты любой из перечисленных сущностей.

Дальнейший разбор дерева будет заключаться в выделении необходимых и важных для поставленной задачи узлов и листьев АСД. К примеру, не имеет смысла переводить в вид RDF различные терминальные символы, если они находятся в левой части грамматического правила. Таким образом, не имеет значения все то, что не укладывается в контекстно-свободную грамматику. В связи с тем, что RDF является языком описания ресурсов, а терминал вряд ли можно к таковым отнести, то парсер АСД в RDF должен такие листья дерева исключать из разбора.

На вход парсера в RDF поступают АСД и правила разбора в RDF. Работа парсера заключается в переводе поступающей информации в вид триплетов «субъект – предикат – объект», если эта информация имеет ценность для поставленной задачи. Генерируемый парсером RDF поток триплетов поступает на вход модуля, записывающего триплеты в выбранном формате.

Архитектурная модель создаваемой системы представлена на рис. 2.

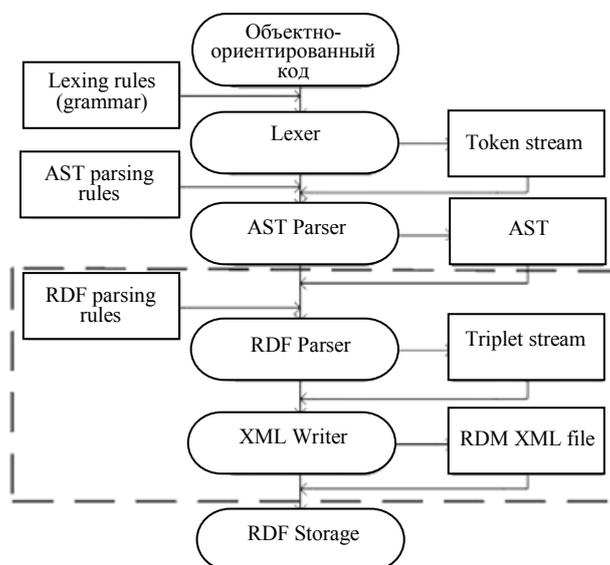


Рис. 2. Архитектура системы разбора кода в RDF

Программная реализация разбора кода

Существует два метода разбора кода – нисходящий и восходящий. Нисходящий разбор начинается с корня дерева, осуществляя рекурсивный проход до листьев. Восходящий же, наоборот, начинает проход с листьев, а заканчивает корнем. Алгоритмы восходящего метода разбора – более сложные, а в совокупности с тем фактом, что избыточность RDF не повлечет каких-то серьезных последствий, можно утверждать, что разумнее в задаче перевода кода в RDF использовать более простой и быстрый метод нисходящего разбора.

Существует множество программных библиотек, помогающих разобрать текст программы на контекстно-свободном языке программирования и получить его абстрактное представление в виде двоичного дерева. Каждое такое средство разбора оперирует грамматикой выбранного языка, написанной с соблюдением единой РБНФ, проводя лексический и синтаксический разборы.

Одно из подобных программных средств – это Another Tool for Language Recognition (ANTLR). ANTLR – это генератор парсеров на языке Java, который принимает на вход файл грамматики, написанный по единой форме Бэкуса–Наура, и файл с исходным кодом разбираемой программы, а на выходе создает абстрактное синтаксическое дерево. На самом деле провайдер ANTLR предоставляет утилиту ANTLRWorks, позволяющую описать грамматику и генерирующую классы лексера и парсера. Помимо утилиты предоставляется библиотека Java времени выполнения ANTLR, которая как раз и запускает в работу лексер и парсер, получает от них результат и строит дерево.

Готовых решений для перевода дерева в RDF не существует. Соответственно, встает задача разработки подобного решения. Поскольку к этому моменту уже имеется формальная форма представления кода, которая является результатом работы ANTLR, перед парсером RDF ставятся только следующие задачи:

- выборка важных и необходимых узлов АСД;
- сопоставление объектов, субъектов и предикатов на основе анализа ветвей АСД;
- формирование триплетов RDF;
- создание и запись XML-сущностей (или других, что зависит от выбранной формы записи RDF), отражающих суть каждого триплета.

Решение должно состоять из двух модулей – RDF Parser и XML Writer.

RDF Parser должен проходить по всем узлам дерева разбора, проверять каждый на соответствие правилам разбора в RDF и, если узел необходим для решения задачи, генерировать новый триплет в потоке триплетов и запоминать узел как проверенный.

XML Writer, получая триплеты из потока, генерирует XML-код, который описывает триплеты в нужном формате.

Алгоритмы работы RDF Parser и XML Writer представлены на рис. 3, а, б.

Весь программный комплекс, как «черный ящик», должен выглядеть так: на входе – исходный код программы, на выходе – требуемая форма записи RDF и, при необходимости, – отчет об ошибках. Отчет об ошибках служит для дополнительного анализа, если, к примеру, исходный код программы был написан с нарушениями общепринятых стандартов записи кода (например, соглашение о записи кода на языке Java).

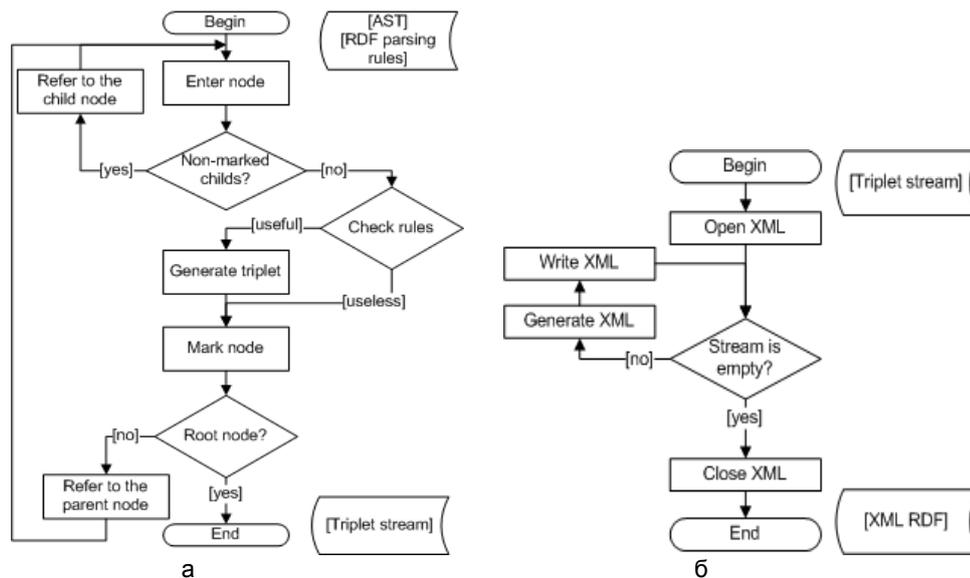


Рис. 3. Алгоритм RDF Parser (а); алгоритм XML Writer (б)

Заключение

В работе рассмотрены основные принципы решения задачи преобразования исходного кода на объектно-ориентированном языке программирования в формат RDF и предложен прототип архитектуры программного продукта решающего эту задачу. Программный продукт, основанный на предложенном архитектурном прототипе, может найти достаточно широкое применение при решении задач анализа кода, в экспертных системах и САПР.

Литература

1. Eric Miller. RDF Primer: W3C Recommendation // W3C [Электронный ресурс]. – Режим доступа: <http://www.w3.org/TR/rdf-primer/> (дата обращения: 20.01.10).
2. Context-free grammar // Wikipedia [Электронный ресурс]. – Режим доступа: http://en.wikipedia.org/wiki/Context-free_grammar (дата обращения: 04.02.10).
3. Extended Backus-Naur Form // Wikipedia [Электронный ресурс]. – Режим доступа: <http://en.wikipedia.org/wiki/EBNF> (дата обращения: 04.02.10).
4. Parr T. The definitive ANTLR reference: Building domain-specific languages / Terence Parr. – San Francisco; The pragmatic programmers, LLC, 2007. – 384 p.

- Зараковский Алексей Владимирович** – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, dbiryukov@list.ru
- Клименков Сергей Викторович** – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, ассистент, Serge.Klimenkov@Servicom.Ru
- Ткаченко Никита Иванович** – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, n.i.tkachenko@gmail.com
- Харитоновна Анастасия Евгеньевна** – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, ассистент, Anastassia.Kharitonova@Elcom.SPb.Ru

УДК 004.627

АНАЛИЗ МЕТОДОВ ПОСТРОЕНИЯ ТРАЕКТОРИИ ДВИЖУЩИХСЯ ОБЪЕКТОВ НА ОСНОВЕ СЕГМЕНТАЦИИ ВИДЕОДАНЫХ

И.С. Рубина

Проведено исследование алгоритмов, основанных на сегментном подходе к решению задачи прогнозирования и компенсации движения. Этот класс методов устраняет большую часть недостатков решений на основе попиксельного подхода с высокой трудоемкостью алгоритма и объектного подхода со сложностью в определении формы объекта прогнозирования. В связи с тем, что полный перебор возможных вариантов прогноза для блока имеет высокую вычислительную сложность, в рамках данной работы осуществлена выработка оптимальной схемы селекции блоков

для сравнения. Также предлагается ряд усовершенствований существующих схем с целью улучшения соотношения вычислительная сложность/качество.

Ключевые слова: сегментный подход, траектория, объект, компенсация движения, селекция блоков, сравнение подходов.

Введение

Задача построения траектории движущихся объектов является неотъемлемой частью эффективных схем сжатия видеопоследовательностей. Выбор ее решения всегда является компромиссом между точностью приближения и вычислительной сложностью алгоритма.

Сегментный подход к решению задачи построения траектории движущихся объектов устраняет большую часть недостатков решений на основе попиксельного подхода с высокой трудоемкостью алгоритма и объектного подхода со сложностью в определении формы объекта прогнозирования [1]. Сущностью компенсации в нем является прямоугольный блок. Сегментный подход использует предположение о том, что в рамках двух соседних кадров местоположение и форма объектов изменяются незначительно. Тогда это изменение можно скомпенсировать параллельным переносом сегмента на некоторый вектор так, как показано на рис. 1. Это допущение работает для подавляющего большинства кадров видеопоследовательности, за исключением участков полного изменения кадра при переключении сцены [2].

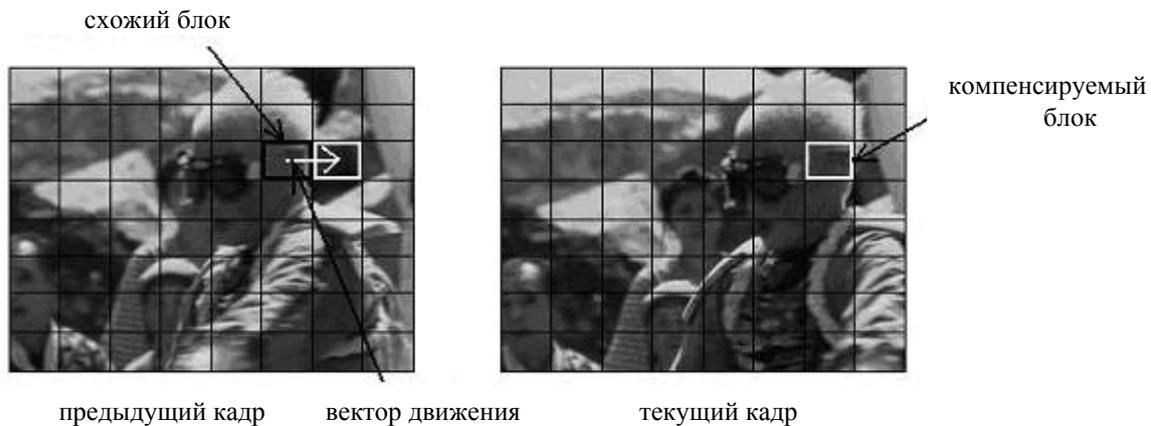


Рис. 1. Схема работы алгоритма

В связи с тем, что полный перебор возможных вариантов прогноза для блока (метод FS) имеет высокую вычислительную сложность [3], в рамках исследования необходима выработка схемы селекции блоков для сравнения.

Целью данной работы является анализ существующих методов построения траектории движущихся объектов на основе сегментации видеоданных и выработка способов их модификации.

Базовые положения исследования

Человеческое зрение обладает особенностью не отличать умеренно пониженное разрешение и менее точно представленное динамичное движение.

При поиске совпадений необходимо учитывать совпадения, найденные на более ранних стадиях решения задачи.

Необходимо осуществлять поиск глобального минимума критерия различия сущностей, препятствуя выявлению локального минимума вместо глобального.

Схемы алгоритмов в рамках исследования

Произведен анализ существующих схем построения траектории движущихся объектов на основе сегментации видеоданных. Были исследованы следующие схемы: полный перебор (FS), алгоритм «один за раз» (OTA), алгоритм ортогонального поиска (OSA), трехшаговый поиск (TSS), двумерный логарифмический поиск (TDL), четырехшаговый поиск (FSS), иерархический поиск – метод усредненной пирамиды (MP).

Алгоритм FS (алгоритм полного перебора). Алгоритм предполагает перебор всех возможных вариантов прогноза для блока. Данная схема имеет высокую вычислительную сложность при максимальном качестве прогнозирования и может использоваться в качестве эталона для сравнения.

Алгоритм OTA (алгоритм «один за раз»). Алгоритм прост в реализации, но эффективен в поиске положения оптимального блока. На протяжении начального этапа алгоритма по горизонтали находится блок с минимальным отклонением, затем, начиная с него, минимальное отклонение ищется в направлении вертикали. Обязательным условием расширения области поиска является уменьшение величины

отклонения, иначе один из этапов алгоритма заканчивается. Схема работы алгоритма представлена на рис. 2, где блок А – минимум на горизонтальной стадии, а В – искомый блок компенсации.

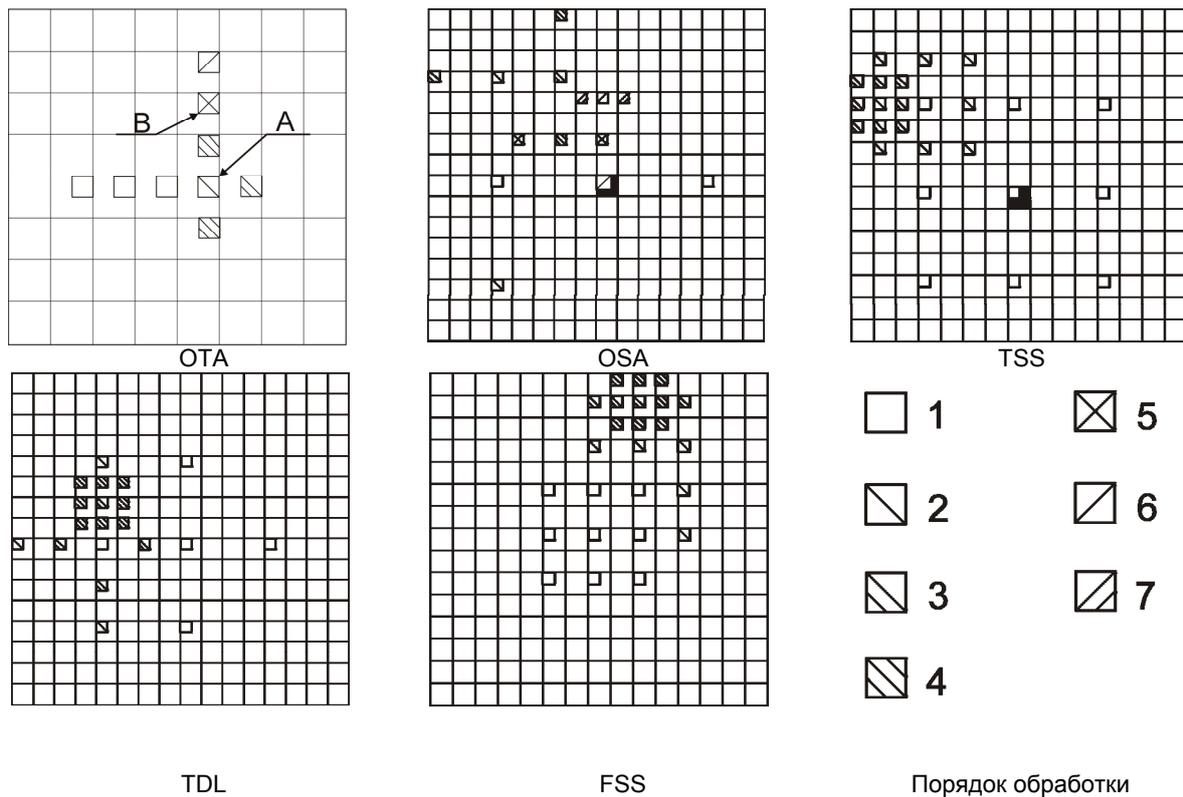


Рис. 2. Схемы работы рассматриваемых алгоритмов

Недостатком алгоритма является то, что в нем невозможно предсказать количество обрабатываемых блоков. Также необходимо учитывать тот факт, что функция ошибки компенсации почти никогда не бывает монотонной, часто имеет место множество ее локальных экстремумов, затрудняющих поиск глобального экстремума. Поэтому представляется целесообразным использовать алгоритмы с другими шаблонами.

Алгоритм OSA (алгоритм ортогонального поиска). Для уменьшения вероятности нахождения локального минимума вместо глобального используют алгоритм ортогонального поиска. Он решает задачу нахождения блока за конечное число шагов. Использует поочередно два шаблона, в процессе перемещая центр области в позицию блока с меньшим отклонением, с циклическим уменьшением плеча (рис. 2).

Ряд экспериментов показал, что использование этого шаблона также не устраняет вероятность случайных совпадений. Это приводит к необходимости использования алгоритмов многоточечных шаблонов.

Алгоритмы многоточечных шаблонов TSS, TDL, FSS.

- **Алгоритм TSS** (трехшаговый поиск) был разработан в 1981 году и до сих пор является популярным благодаря своей простоте, надежности и высокой производительности (рис. 2). Основная проблема алгоритма заключается в удаленности равномерно распределенных точек шаблона, что делает его неэффективным для областей малого движения.
- **Алгоритм TDL** (двухмерный логарифмический поиск) требует несколько большего количества шагов, но может быть более точным, особенно в случаях большого размера окна. В отличие от TSS, текущий центр области поиска включается в множество кандидатов. Условием уменьшения шага алгоритма является определение центра области блоком наименьшего отличия. На последней стадии алгоритма шаблон изменяется (рис. 2). Существует множество вариантов этого алгоритма, отличающихся условием смены шага шаблона, причем уменьшение шага вдвое не всегда является оптимальным решением.
- **Алгоритм FSS** (четырёхшаговый поиск) основан на таком свойстве большинства видеопоследовательностей, как ориентированность к центру кадра. На первой и на последней стадии алгоритма используется девятиточечный шаблон, на двух остальных в зависимости от локации блока с минимальной функцией критерия выбирается один из шести шаблонов (рис. 2). Как правило, данный алгоритм показывает большую надежность с сохранением эффективности для сложных

вариантов движения и операций масштабирования. Это делает FSS привлекательной стратегией для выборки блоков в схемах компенсации движения.

Алгоритм МР (иерархический поиск – метод усредненной пирамиды). Для уменьшения сложности схемы полного перебора также были предложены схемы грубого иерархического поиска. Это сокращение достигается за счет компенсации на кадре меньшего разрешения. Примером таких алгоритмов является МР [4].

Вначале для устранения шумового эффекта изображение меньшего разрешения получают с помощью фильтра низких частот. В дальнейшем для получения многоуровневой иерархии изображений меньшего разрешения используется простое усреднение пикселей изображения предыдущего уровня по принципу

$$g_L(p, q) = \left[\frac{1}{4} \left(\sum_{u=0}^1 \sum_{w=0}^1 g_{L-1}(2p+u, 2q+w) \right) \right],$$

где $g_L(p, q)$ – значение пикселя уровня L в позиции (p, q) . Таким образом, если использовать три уровня иерархии, один пиксель уровня 2 соответствует блоку 4×4 уровня 0 и блоку 2×2 уровня 1 соответственно. В то же время, блок размером 16×16 уровня 0 будет соответствовать блоку $(16/2L) \times (16/2L)$ уровня L .

После построения усредненной пирамиды осуществляется подсчет среднего абсолютного отклонения (MAD) и выбор вектора, имеющего наименьшее MAD , в качестве грубого вектора движения 2 уровня. Найденный вектор перемещается на уровень 1, и производится его уточнение. Тот же процесс повторяется для уровня 0, на котором получается искомым вектор. Для увеличения точности работы алгоритма данная процедура может выполняться для нескольких векторов уровня 2, имеющих близкие значения MAD , с использованием окон (областей поиска) уменьшенного размера (рис. 3).

Таким образом, были выявлены существенные достоинства и недостатки методов отбора блоков компенсации.

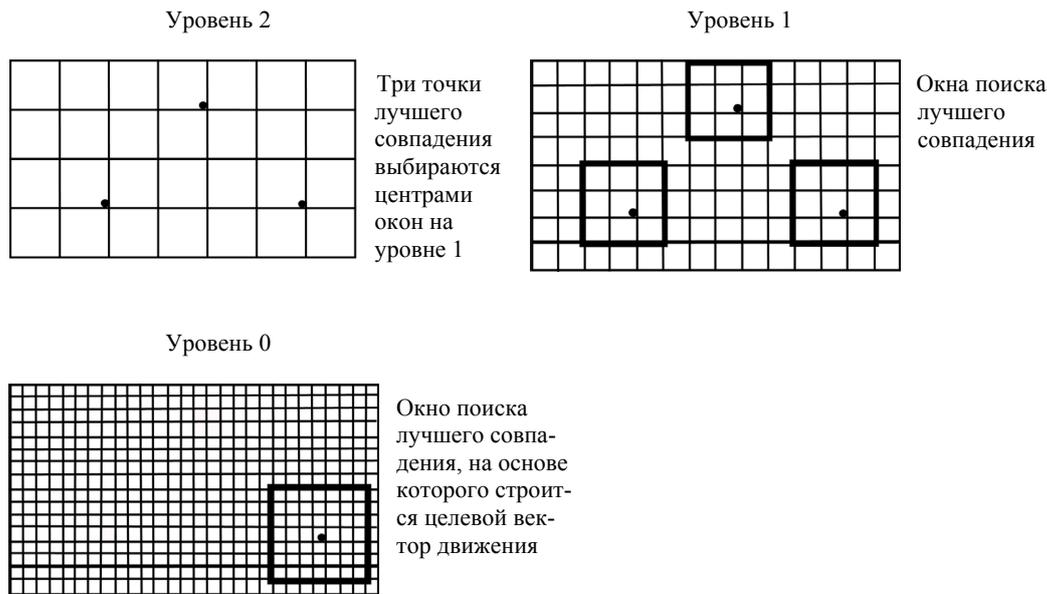


Рис. 3. Схема работы алгоритма иерархического поиска

Экспериментальные результаты

В рамках проведенного исследования для алгоритмов, за исключением МР, использовались размеры блока 16×16 . В ходе экспериментов были получены следующие зависимости.

1. Количество базовых операций на блок для перечня кадров (0–20) для последовательности «Береговая охрана». В данном случае под базовой операцией понимается количество операций сложения/вычитания.
2. Усредненное среднее абсолютное отклонение ($AMAD$) – среднее по изображению MAD для перечня рассматриваемых алгоритмов:

$$MAD = \frac{1}{mn} \sum |(F_{Orig}(p) - F_{Comp}(q))|,$$

где $m \times n$ – размеры блока, а $F_{Orig}(p)$ и $F_{Comp}(q)$ – значения интенсивности пикселя блока ссылочного кадра p и интенсивности соответствующего ему пикселя блока-прогноза q компенсируемого кадра.

3. Среднее время процессора для перечня рассматриваемых алгоритмов.

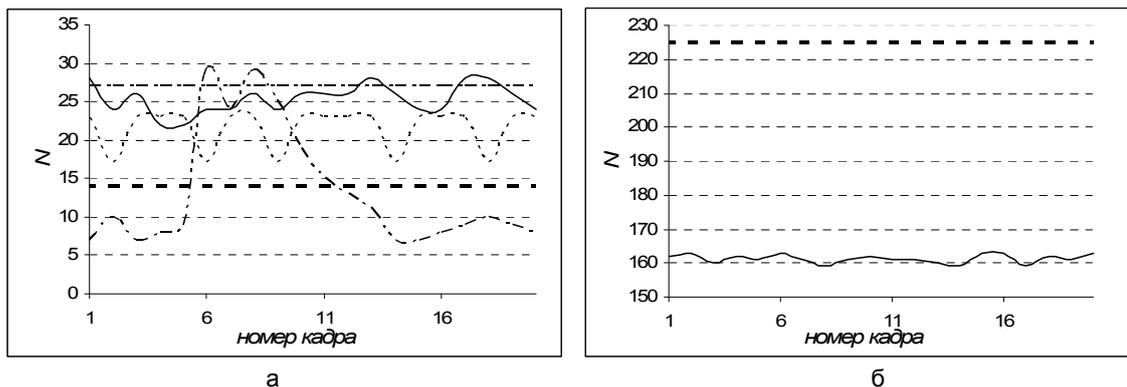


Рис. 4. Значения количества базовых операций (N) для ряда кадров последовательности «Береговая охрана» для перечня рассматриваемых алгоритмов: (а) — — TSS; - - - TDL; — FSS; - · - · - OTA; - - - OSA; (б) — — FS; — MP

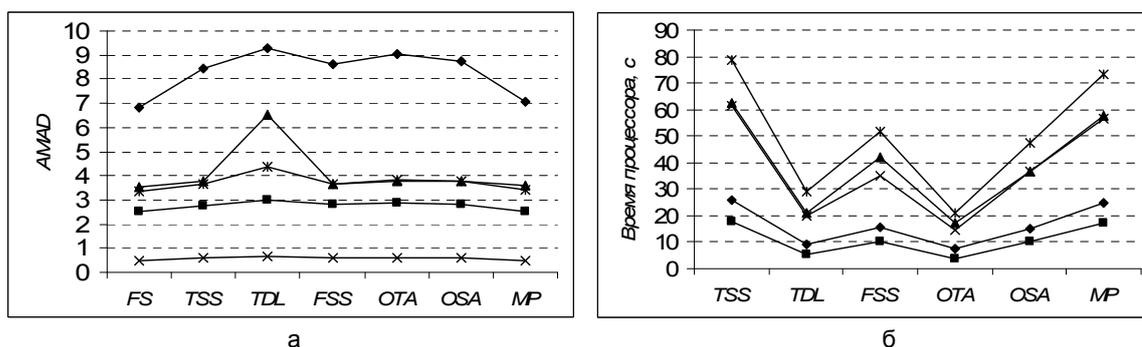


Рис. 5. Значения среднего по кадру усредненного абсолютного отклонения (а); значения времени процессора (б) для последовательностей: —◆— «Дети»; —■— «Погода»; —▲— «Береговая охрана»; —×— «Акийо»; —*— «Телефон в машине»

Исследование подтвердило предположение о том, что увеличение трудоемкости алгоритма сопровождается улучшением качества его работы. Таким образом, большинство алгоритмов с высокой производительностью страдает недостаточным качеством работы схем компенсации, и наоборот. По количеству базовых операций наиболее интересен в анализе алгоритм OTA, показавший хорошие значения критерия на областях плавного движения (кадры 1–5 и 12–20) и чуть худший в динамичных сценах (рис. 4, а).

Самые лучшие показатели процессорного времени показал алгоритм OTA (рис. 5, б) при одном из худших уровней качества работы схемы компенсации движения (рис. 5, а). TDL не дал оптимальной схемы выборки, уступив OTA как с точки зрения качества, так и по части трудоемкости. Популярный среди разработчиков алгоритмов компенсации движения TSS значительно уступил OTA в плане вычислительных затрат. В свою очередь, алгоритм FSS превзошел TSS по вычислительной сложности при схожем качестве получаемого прогноза.

В ходе исследования было выявлено, что единственный подход, сравнимый по качеству с FS, но снижающий его вычислительную сложность – MP-алгоритм иерархического поиска (рис. 4, б). MP избегает подмены глобального минимума локальным путем сглаживания поля векторов за счет анализа нескольких векторов-кандидатов (основной причины снижения производительности алгоритмов поиска по шаблону). Однако как преимущество алгоритмов поиска по шаблону можно отметить то, что поиск вектора движения для каждого блока не зависит от результатов поиска на более ранних стадиях алгоритма, что делает метод наиболее эффективным при сложных вариантах движения.

Заключение

В ходе исследования было выявлено, что для видеоследовательностей различной динамичности алгоритмы построения траектории движущихся объектов на основе сегментации видеоданных показывают различные результаты. Например, перебор по шаблону дает высокую точность на прогнозируемом количестве шагов в сценах высокой динамичности, в то время как иерархический поиск дает лучшее качество прогнозирования на последовательностях низкой динамичности.

В целях дальнейшей оптимизации схемы поиска необходимо объединение сегментов по направлению движения в рамках объекта – приближение к объектному подходу, а также сочетание с параметрическими алгоритмами.

Литература

1. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. – М.: Диалог-МИФИ, 2002. – 384 с.
2. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2006. – 1072 с.
3. Toivonen T. A New Algorithm for Fast Full Search Block Motion Estimation Based on Number Theoretic Transforms / J. Heikkilä, O. Silvén // Proc. 9th International Workshop on Systems: Signals and Image Processing: November 7–8: Manchester: United Kingdom, 2002. – P. 90–94.
4. Kwon Moon Nam. A Fast Hierarchical Motion Vector Estimation Algorithm Using Mean Pyramid / Joon-Seek Kim, Rae-Hong Park // IEEE Transactions on Circuits and Systems for Video technology. – 1995. – V. 5. – № 4. – P. 344–351.

Рубина Ирина Семеновна – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, rubren@mail.ru

УДК 004.912: 303.7

ПРОБЛЕМЫ ПОПОЛНЕНИЯ СЕМАНТИЧЕСКОГО СЛОВАРЯ

К.К. Боярский, Е.А. Каневский

Рассмотрены проблемы пополнения компьютерного семантического словаря новыми словами, встреченными в тексте при его анализе. Предлагаемая для этого система работает в полуавтоматическом диалоговом режиме. На первом этапе определяются морфологические характеристики нового слова, на втором – его синтактико-семантические параметры по аналогам, имеющимся в существующем словаре. Предлагаемые подходы обеспечивают высокий уровень точности. Впервые появилась возможность указания точной семантики новых слов с учетом не только семантических классов, но и аргументов, обеспечивающих связь с подсоединяемыми словами.

Ключевые слова: анализ текста, лексема, морфология, семантика, синтаксис, словарь, слово.

Введение

Задаче компьютерного анализа текста на естественном языке посвящено множество теоретических и практических работ. Эти задачи, а именно – поиск документов, рубрицирование и аннотирование документов, диалог с компьютером, машинный перевод и построение баз знаний, – решали и решают различными методами, используя или не используя ту или иную дополнительную информацию. Решение любых прикладных задач, связанных с анализом естественного текста, начинается с морфологического анализа. Такой анализ еще можно проводить без использования словаря [1]. Далее может проводиться синтаксический и семантический анализы, для которых словарь крайне необходим.

Используемый авторами семантический словарь В.А. Тузова [2] основан на расширенном морфологическом словаре А.А. Зализняка [3] и представляет собой список статей (лексем), каждая из которых соответствует одному слову русского языка. При этом одному слову может соответствовать несколько лексем, выражающих различный семантический смысл. Так, например, слову *коса* соответствуют три лексемы: *девичья коса*, *береговая коса* и *острая коса*. В настоящее время словарь насчитывает 165 тысяч лексем, соответствующих 145 тысячам слов общей нормативной лексики русского языка. Лексемы сгруппированы в 1650 классов, которые образуют иерархическую структуру, отражающую родовидовые отношения между лексемами [2]. Каждая из статей словаря содержит морфологическое, синтаксическое и семантическое описание лексемы. Так для лексемы «ЗОРИН» словарная статья имеет следующий вид:

ЗОРИН \$12413/03000(S1>Наб(S1>ЧЕЛОВЕК\$1241,S0:ФАМИЛИЯ\$1241/11)) {m1lo 298}

Морфологическое описание лексемы содержится в фигурных скобках, где «m1lo» (морфологический описатель, аналогичный описателю в [3] обозначает существительное мужского рода, 11-го класса, одушевленное, а число «298» – адрес соответствующих падежных окончаний в файле окончаний лексем. Идентификатор \$12413/03000 обозначает принадлежность к классу (ФО/Живой/Человек/Личность/ФИО/Фамилия). В круглых скобках расположено собственно синтаксическое и семантическое описание лексемы, которое в данном случае означает «человек имеет фамилию».

Проблема заключается в том, что какого бы объема ни был словарь, при анализе очередного текста всегда обнаруживаются новые слова (НС), в данном словаре отсутствующие. Это могут быть имена и фамилии, географические названия и образованные от них прилагательные, специальные термины и слова, употребленные автором в необычном значении, словоформы, противоречащие современным правилам грамматики (например, при передаче особенностей речи персонажей) и т.д. Так, по наблюдению Т.Ю. Кобзаревой, при анализе текстов Набокова, Мандельштама, Л.Н. Толстого, Гоголя часто встречались лексически продуктивные формы, неологизмы, «аномалии», не учтенные, например, в [3] и в компьютерном словаре пакета Word-2000 [4]. Рассмотрим подробнее структуру НС в романах Гончарова.

Новые слова

В процессе предварительного анализа текста романа авторами был получен полный перечень нераспознанной лексики по трем романам И.А. Гончарова. Ниже (таблица) приведены общие данные о словах, отсутствующих в семантическом словаре [5].

Романы И.А. Гончарова	Слово- форм	НС	Из них:				
			Имен	Сущ.	Глаг.	Прил.	Нареч.
Итого	467 тыс.	1300	545 39%	273 20%	188 14%	251 18%	84 6%

Таблица. Состав новых слов романов Гончарова

Как видно из таблицы, основную проблему представляют имена собственные. Количество их доходит до 40% от всех. Для данного списка характерно как раз наличие устаревших словоформ, использующих окончания с мягким знаком (*Артемий – Артемью, Василий – Васильем*) и уничижительных имен (*Аверка, Васька*).

Удельный вес новых существительных почти в два раза меньше. Часть из них использует окончания *-ье* вместо современного *-ие* (*вдохновенье, влечение*). Некоторые существительные использованы в устаревшем написании в корне (*бусурман, нумер*). В ряде существительных используются устаревшие словоизменительные формы (*крыло – крылами, чулки – чулков*). Особо следует отметить существительные, пишущиеся через дефис (*наденька-изменница, старец-классик*). Обычно в случае отсутствия таких слов в словаре программа разбивает их на два слова, что является ошибкой.

Доля новых глаголов несколько меньше. Некоторые глаголы использованы в устаревшем написании (*дотрогиваться, испужаться*). В ряде глаголов используются устаревшие словоформы (*воздвигнуть – воздвигнул, гулять – гуляючи*).

Что касается новых прилагательных, то сравнительно большая их группа, начинающаяся с приставки *не-*, в современном языке почти не употребляется (*непогрешительный, неупотребленный*). Небольшая группа прилагательных начинается с приставки *пре-* (*превеселенький, преглубокий*). Особо следует отметить прилагательные, пишущиеся через дефис (*безвинно-угнетенный, бледно-чернильный*).

Количество новых наречий невелико. Значительная их группа начинается с приставки *по-* (*повчерашнему, по-латыни*). Следует отметить небольшую группу наречий, пишущихся через дефис (*мало-мало, благородно-бесполезно*).

Во всех трех романах авторам встретилось 20 новых междометий (*м-м, тс, фу-фу*).

Вспомогательная система Adviser

В связи с вышеизложенным возникает задача пополнения словаря. Пополнение семантического словаря само по себе является сложной задачей, для решения которой предлагаются различные способы [6]. Простейший вариант, который может быть использован для этой цели, – использование образца [1]. Однако даже правильно указать морфологические параметры иногда оказывается весьма непростой задачей. Дело в том, что при сравнительно небольшом числе вариантов грамматических категорий, сопоставляемых определяемому слову, общее количество наборов окончаний приближается к тысяче. Описать же синтактико-семантические характеристики для неспециалиста по устройству данного конкретного словаря просто нереально. Авторами разработана система Adviser, позволяющая пополнять словарь НС в полувыводном диалоговом режиме.

На первом этапе определяются морфологические характеристики НС. Пользователь задает часть речи, к которой относится НС – существительное, прилагательное, глагол, наречие или междометие. Возможно также дополнительное указание ряда характеристик – одушевленность и род для существительных, совершенный/несовершенный вид глагола и т.д. После этого система позволяет подобрать из имеющихся в словаре такое слово, окончания словоформ которого совпадают с окончаниями словоформ НС.

Поскольку в систему подсказки заложены все известные варианты словоизменений, то нужный вариант обязательно найдется [3]. Исключение составляет архаическое или нарочито искаженное написание слова, например, деепричастие *завидя* от глагола *завидеть*. В этом случае используется файл замен.

После установления морфологических характеристик НС нужно задать его семантику (синтактико-семантические параметры). Прежде всего нужно определить класс по классификатору. Эта структура предъявляется пользователю в виде дерева классов. Для облегчения работы предусмотрена возможность ускоренного указания таких часто встречающихся для НС классов, как имя, отчество, фамилия, различные названия (географических объектов, фирм, документов и др.) – всего 82 класса. Кроме того, можно просто найти синоним НС. Например, к слову *вдохновенье* указать синоним *вдохновение* и сразу полу-

чить, что это слово относится к классу «Физический_объект/Живой Человек/Психика/Душа/Чувство/Депрессия-Вдохновение».

После этого пользователю предьявляется полный список слов, принадлежащих данному классу. Из них выбирается наиболее близкое по значению, и его семантика приписывается к НС. При необходимости эта семантика может быть уточнена вручную. Теперь НС с правильной морфологией и семантикой готово для занесения в словарь.

Предлагаемая система Adviser апробирована на массиве более 1000 слов и показала прекрасные результаты. Следует отметить, что без системы такого рода составление подсловаря на 1300 слов потребовало бы значительно больше времени.

Определение морфологии НС

На первом этапе определяются морфологические характеристики НС. Методика их определения основана на использовании обратного словаря [7]. Как известно, наиболее приемлемой в данном случае является почти полностью автоматизированная процедура склонения и спряжения, реализованная в виде диалога с пользователем. Считается, что достаточно ограничиться четырьмя знаменательными частями речи: существительными, прилагательными, глаголами и наречиями.

Система определения морфологических характеристик НС достаточно проста. Пользователю предлагается три окна и набор кнопок (рис. 1). Вручную или из заранее подготовленного файла новое слово вводится в среднее окно. Затем пользователь выбирает часть речи, к которой относится НС – существительное, прилагательное, глагол, наречие или междометие (по нашему мнению, междометие также заслуживает того, чтобы быть включенным в систему). Как показала практика, среди других частей речи НС практически не встречаются. Возможно также дополнительное указание ряда характеристик: одушевленность и род для существительных, совершенный или несовершенный вид глагола и т.д.

Программа осуществляет получение обратного отображения заданного слова и поиск статьи из соответствующего файла, в которой имеет место совпадение заданного слова с первым словом соответствующей статьи этого словаря по максимальному количеству букв, начиная с трех. Если необходимое трехбуквенное сочетание вообще отсутствует в файле, ищется двухбуквенное сочетание или одна буква. После нахождения подходящей статьи введенное слово, морфологический описатель и адрес падежных окончаний отобранной лексемы передаются в морфологический анализатор. Последний по исходной форме введенного слова (единственное число, именительный падеж – для склоняемых частей речи) выполняет генерацию всей его парадигмы. Для решения этой задачи используется файл окончания лексем. Результат генерации всегда выводится в правое окно.

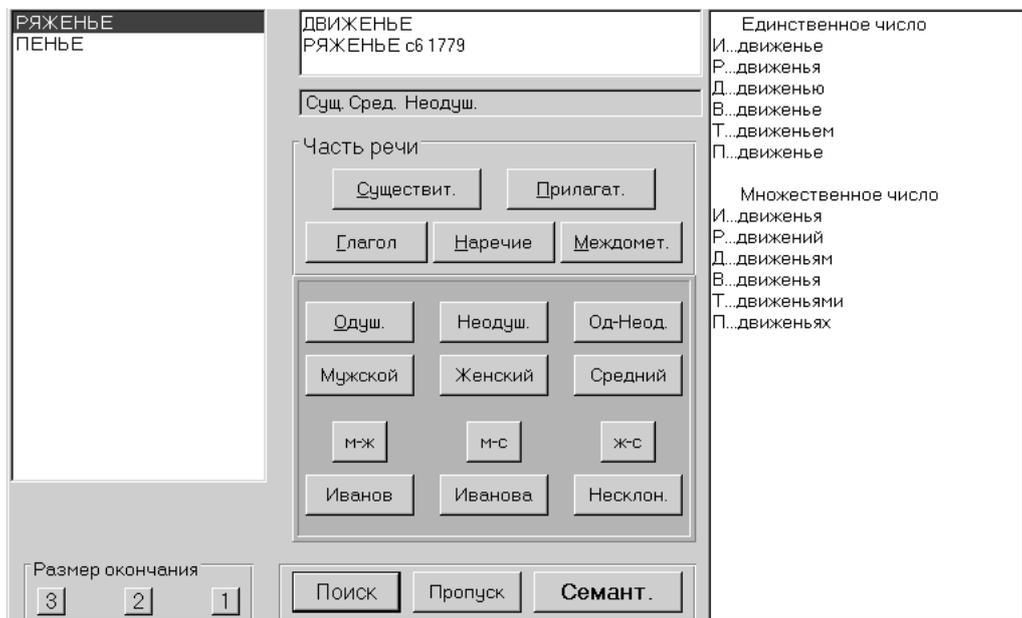


Рис. 1. Подбор морфологического аналога для существительного *двигенья*

Для имен существительных (рис. 1) выводится склонение по падежам для единственного и множественного чисел. Здесь следует отметить два обстоятельства. Во-первых, очень многие русские фамилии, особенно такие, которые оканчиваются на -ов, -ев, -ин, склоняются по типу слова ТОПТЫГИН, так что целесообразно для ускорения обработки таких фамилий ввести особую кнопку. Во-вторых, в исходном

словаре Зализняка [3] вообще не оказалось ни одного слова, которое бы склонялось по типу фамилий СИДОРОВА, ИВАНОВА, так что пришлось в основной словарь добавить статью

ИВАНОВА ж1 1о 27937

и также ввести особую кнопку для обработки подобных фамилий.

Для имен прилагательных выводится склонение по падежам для мужского и женского родов (единственное число) и множественного числа, мужской и женский род для краткой формы и сравнительная форма. Здесь следует отметить важность вывода информации о наличии кратких и сравнительных форм. Так, например, нужно уметь отличать склонение прилагательного *аляповатый* (*аляповат*, *аляповата*) от склонения прилагательного *бывалый* (формы *бывал* и *бывала* являются формами глагола *бывать*, а не краткими формами прилагательного *бывалый*).

Рис. 2. Подбор морфологического аналога для глагола *растопаться*

Для глаголов (рис. 2) выводится спряжение по лицам для настоящего или будущего времени (в зависимости от вида глагола), мужской и женский род для прошедшего времени, деепричастия настоящего и прошедшего времени, причастия действительного и страдательного залога и повелительное наклонение для единственного и множественного чисел. Здесь особую важность приобретает вид глагола, а также наличие соответствующих форм причастия, деепричастия и повелительного наклонения. Только учет всех этих параметров позволяет подобрать правильное морфологическое описание лексемы.

Если пользователя не устраивает предлагаемый ему вариант изменения введенного слова, то он может выбрать для образца какое-нибудь другое слово. Набор таких слов предлагается в левом окне. При необходимости возможна процедура отката к совпадению по двум или даже одной букве. После осуществления выбора введенному слову приписываются морфологический описатель и адрес падежных окончаний отобранной лексемы.

В ряде случаев (например, при архаическом или нарочито неправильном написании слова) НС отличается от лексемы, уже имеющейся в словаре одной или двумя формами. Так, например, И.А. Гончаров в романе «Обломов» использует имя *Артемя* вместо *Артемию*, деепричастие *завидя* вместо *завидев* и др. В этих случаях вместо пополнения словаря можно занести подобную словоформу в специальный файл исключений с тем, чтобы перед началом работы морфологического анализатора произвести необходимую замену (*Артемя* на *Артемию*).

Определение семантики НС

После установления морфологических характеристик НС нужно задать его семантику (синтактико-семантические параметры). Для этого, прежде всего, следует установить принадлежность этого слова к определенному классу. Затем необходимо задать возможные связи обрабатываемого НС с другими словами, по возможности описать смысл данного слова с помощью лексических функций и т.п.

Вначале определяется класс НС по классификатору. Используемый нами классификатор в настоящее время представляет собой иерархическую структуру из 1600 классов, являющихся основой описания формальной семантики понятий русского языка и отражающих родовидовые отношения между лексемами ([2], с. 101–128). Для облегчения работы предусмотрено несколько видов поиска:

- поиск классов, содержащих имена, отчества или фамилии;
- просмотр классов, содержащих различные названия (географических объектов, фирм, документов и др.);
- поиск классов, содержащих в своих названиях заданное слово;
- поиск класса, содержащего синоним НС.

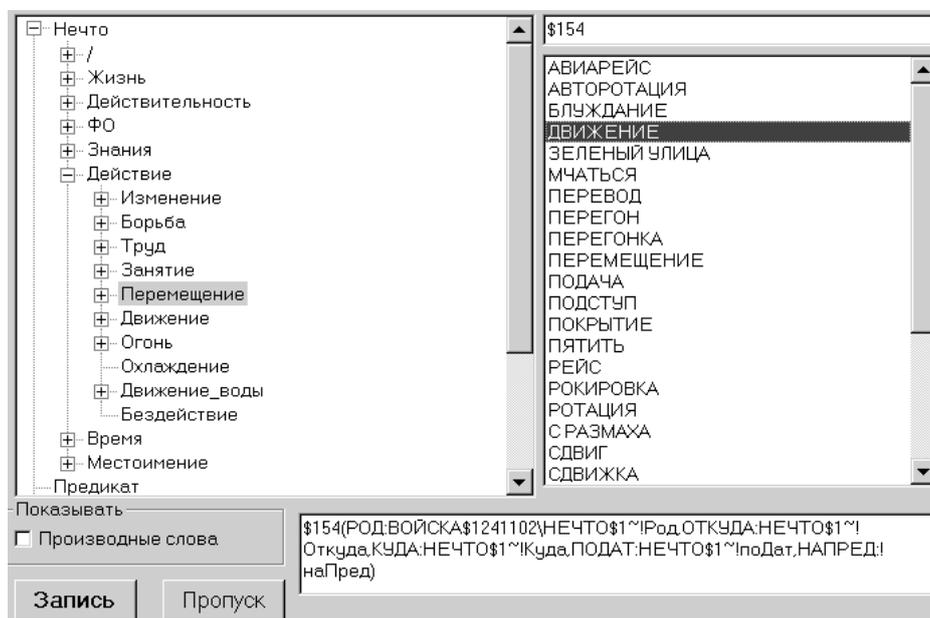


Рис. 3. Подбор семантического аналога

Например, для нового слова *движенье* достаточно в качестве синонима задать слово *движение* и сразу получить, что НС относится к классу \$154 «Действие/Перемещение» (рис. 3). При этом показывается дерево классов и полный список слов, принадлежащих данному классу. Из них выбирается наиболее близкое по значению, и его семантика приписывается к НС. В данном случае выбрана лексема, означающая дорожное движение. К ней могут быть подсоединены, например, слова в родительном падеже (движение колонны), откуда (из города), куда (в деревню), в дательном падеже с предлогом «по» (по дороге), и в предложном падеже с предлогом «на» (на машинах).

При необходимости семантика, полученная в нижнем окне, может быть уточнена вручную. Теперь НС с правильной морфологией и семантикой готово для занесения в словарь.

Заключение

Предлагаемая система Adviser апробирована на массивах более 1000 слов различной семантики из произведений И.А. Гончарова и около 5000 фамилий и географических названий и показала прекрасные результаты. Она показала себя достаточно удобной и адекватной задаче определения семантики и морфологии НС. Наибольшие трудности возникли в процессе описания прилагательных, начинающихся с приставки *не-* и образованных от причастий, например, *невысказанный*, *недочитанный*, *незаработанный*. Эти трудности связаны с тем, что причастия являются производной формой от глагола (а в словаре помещены только глаголы). Описание семантики этих прилагательных требует определенного ручного труда, связанного с преобразованием глагол–прилагательное.

Таким образом, описанная система позволяет достаточно быстро и просто пополнять семантический словарь НС, причем работать с ней может даже человек, не знакомый детально с языком описания компьютерного словаря. В отличие от традиционного ручного пополнения словаря, предлагаемые подходы обеспечивают гораздо более высокий уровень точности. Впервые появилась возможность указания точной семантики НС с учетом не только классов, но и аргументов, обеспечивающих связь с подсоединяемыми словами.

Очевидно, что подобные принципы организации системы пополнения семантического словаря с успехом могут быть использованы и при других типах семантических описаний, достаточно только наличия классов или аналогичного принципа построения словаря.

Литература

1. Леонтьева Н.Н. Автоматическое понимание текстов: системы, модели, ресурсы. – М.: Академия, 2006.
2. Тузов В.А. Компьютерная семантика русского языка. – СПб: Изд-во СПбГУ, 2004.

3. Зализняк А.А. Грамматический словарь русского языка. – М.: Русский язык, 1980.
4. Кобзарева Т.Ю. Морфанализ in vivo // Труды Международной конференции Диалог'2004. – М.: Наука, 2004. – С. 286–291.
5. Захаров В.П., Каневский Е.А. Язык И.А. Гончарова через призму современной грамматики // «Прикладна лінгвістика та лінгвістичні технології: MegaLing-2007». – Киев: Довіра, 2008. – С. 131–140.
6. Кожунова О. Опыт применения ДСМ-метода к пополнению семантического словаря // Прикладна лінгвістика та лінгвістичні технології: MegaLing-2006. – Киев: Довіра, 2007. – С. 149–161.
7. Каневский Е.А. Некоторые вопросы пополнения морфологического словаря терминами предметной области // Труды Международного семинара Диалог'2001 по компьютерной лингвистике и ее приложениям. – М.: РосНИИ искусственного интеллекта, 2001. – Т. 2. – С. 156–160.

Боярский Кирилл Кириллович – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат физ.-мат. наук, доцент, boyagin9@yandex.ru

Каневский Евгений Александрович – Санкт-Петербургский экономико-математический институт РАН, кандидат технических наук, ведущий научный сотрудник, kanev@emi.nw.ru

УДК 004.75, 004.772, 004.62

ОРГАНИЗАЦИЯ СЕТЕВОГО ВЗАИМОДЕЙСТВИЯ УЗЛОВ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ ХРАНЕНИЯ ДАННЫХ

Н.М. Лукьянов, А.М. Дергачев

Описывается порядок взаимодействия узлов распределенной системы хранения данных, входящей в состав Интернет сервисов, а также методы обработки информационных потоков, применяемые для организации взаимодействия локальных сетевых сервисов каждого узла. Рассмотрена последовательность движения пользовательских данных внутри системы, описан метод выбора оптимального узла хранения с использованием ранжирования весовыми коэффициентами, а также описана процедура восстановления данных узла после программного или аппаратного сбоя в работе системы.

Ключевые слова: хранилище, распределенные, сетевые, данные, алгоритмы, Интернет, сервис, модуль.

Введение

Данная работа является продолжением публикаций по результатам исследований в области обработки и хранения данных в распределенных системах, являющихся составной частью современных Интернет-сервисов. В предыдущих публикациях рассматривались результаты всесторонних исследований распределенных хранилищ данных, приводился анализ качественных и количественных показателей их функционирования. В ходе исследований, лежащих в основе работы, основное внимание уделялось вопросам производительности системы. Были построены имитационные модели, просчитаны необходимые объемы и структура хранилища, особое внимание уделено надежности данных. Просчитана вероятность выхода из строя носителей информации в процессе их эксплуатации, а также рассмотрены способы резервирования дискового пространства системы [1]. В результате исследований был построен и испытан прототип проектируемой распределенной системы хранения данных, которая была использована в открытой социальной сети студентов факультета Высшей школы менеджмента СПбГУ. Наблюдение за работой системы в реальных условиях позволило более тонко настроить алгоритмы работы с данными, а также предложить более продвинутые способы обработки информации.

Постановка задачи

Задачей распределенной системы хранения данных как составного элемента централизованного Интернет-сервиса является хранение программ и данных и предоставление доступа к ним по мере необходимости со стороны неограниченного количества сетевых пользовательских сервисов. Следовательно, подобные распределенные системы должны быть хорошо масштабируемыми и иметь открытую гибкую архитектуру, что, в свою очередь, требует единого подхода к организации работы с данными во всех узлах распределенной системы, начиная от приема и сохранения данных и заканчивая предоставлением необходимых данных потребителю. Применение вертикального масштабирования аппаратных средств обработки и хранения данных, а также применение систем хранения данных, используемых в корпоративных информационных системах, не дает приемлемого соотношения цены и производительности при использовании этих подходов для организации информационно-емких Интернет-сервисов [2]. В свою очередь, решения на базе многомашинных комплексов с применением несложных аппаратных компонентов могут обеспечить необходимый уровень открытости и гибкости, а также являются легко масштабируемыми как в рамках распределенной системы хранения данных, так и в рамках построения распределенной системы управления сетевыми сервисами в целом.

Архитектура системы

Любую распределенную систему хранения и обработки данных можно рассматривать как совокупность распределенных программно-аппаратных компонентов. В данном случае такими аппаратными компонентами являются отдельные серверы, объединенные локальной вычислительной сетью. Программная составляющая реализации системы является наиболее важной и несет в себе всю функциональную нагрузку системы. Организация взаимодействия отдельных узлов распределенной системы хранения данных опирается на сервис-ориентированную архитектуру. Каждый сетевой сервис состоит из ряда модулей (рис. 1), определяющих его функциональность, а также функциональное назначение узла распределенной системы хранения данных, на котором размещен данный сетевой сервис.

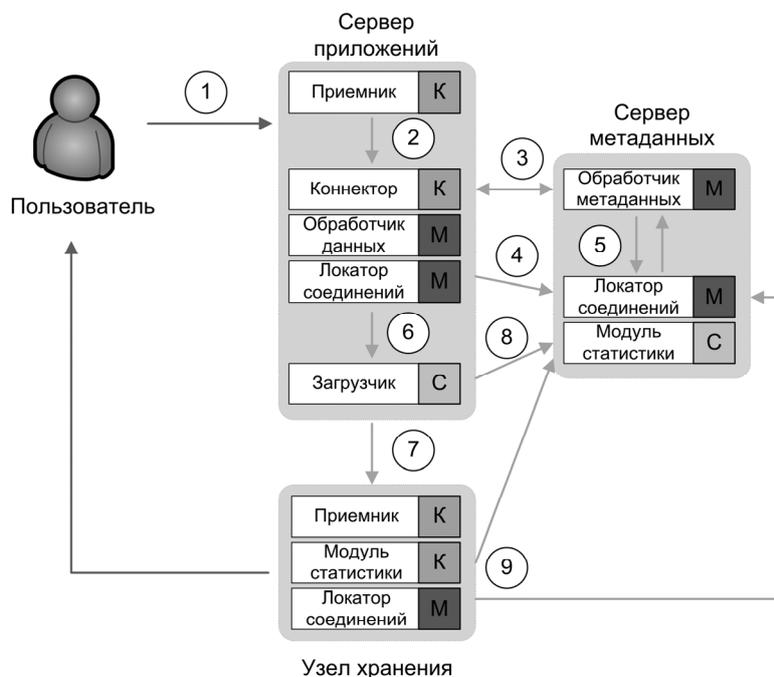


Рис. 1. Программные модули обработки данных

Условно всю распределенную систему хранения данных можно разделить на пять составляющих: пользователь, сервер приложений, сервер метаданных, узлы хранения и сервер производительности (на рисунке не показаны). Подобный подход является гибридным и использует составляющие всех трех разновидностей файловых систем: параллельных, симметричных и с API доступом [1]. Взаимодействие пользователя с системой происходит через сервер приложений, производящий прием и обработку данных для хранения, а также непосредственно с узлами хранения, предоставляющими пользователю ранее сохраненные данные. Алгоритмы и способы работы с информацией, применяемые в представляемой распределенной системе хранения данных, схематично изображены в виде модулей, которые подразделяются по своему типу:

- клиентские модули (К) являются сборщиками информации для ее сохранения в базе данных;
- серверные модули (С), основываясь на данных от клиентских модулей, выполняют заранее определенные действия;
- обособленные модули (М) похожи на серверные модули, но для работы не требуют клиентской части.

Отдельные алгоритмы, реализованные в виде модулей, позволяют обрабатывать разнородную информацию в зависимости от ее типа, среды передачи и условий внутри системы хранения. Происходит интеллектуальная, а не конвейерная обработка данных. Благодаря модульности, формализовав правила ввода/вывода данных, появляется возможность модернизировать последовательность обработки информации путем добавления новых модулей. Наиболее важными модулями являются модули статистики и загрузки/приема данных.

Модуль статистики

В процессе эксплуатации все узлы хранения периодически сообщают системе о своих нагрузочных параметрах. Их модуль статистики каждые три минуты передает в систему следующие характеристики: количество обработанных запросов, среднее время обработки запроса, нагрузка на центральный процессор с момента последнего опроса, загрузка канала передачи данных и объем свободного места на дисках. На стороне узла совершаются лишь простые действия по обработки статистических данных. Затем данные упаковываются и передаются серверу метаданных.

Серверная часть модуля статистики, выполняющая большую часть операций по обработке статистических данных, располагается на сервере метаданных. Основными параметрами, влияющими на быстродействие серверов, являются:

1. коэффициент нагрузки $k_l = \frac{q_{\text{запр}} \times t_{\text{ср}}}{t_{\text{пер}}}$, где $q_{\text{запр}}$ – количество запросов; $t_{\text{ср}}$ – среднее время обработки; $t_{\text{пер}}$ – период опроса;
2. нагрузка процессора;
3. нагрузка канала;
4. свободное место на диске.

Первые три параметра обрабатываются путем присвоения каждому из них весовых коэффициентов, с учетом которых далее рассчитывается общий индекс [3]. Весовые коэффициенты, применяемые в настоящей методике, приведены в таблице.

Параметр	Весовой коэффициент
Коэффициент нагрузки	0,75
Нагрузка процессора	0,1
Нагрузка канала	0,15

Таблица. Весовые коэффициенты параметров быстродействия

Таким образом, путем выставления рейтинга составляется упорядоченный список узлов, готовых к приему и предоставлению данных пользователю. Именно согласно этой таблице модуль загрузки на сервере приложений размещает данные на доступных узлах хранения. В случае отсутствия отчета от узла хранения за положенный промежуток времени считается, что сервер утратил связь с системой – вышел из строя или сильно загружен. Тогда этот узел изымается из списка до тех пор, пока не будет восстановлена его нормальная работа. В случае уменьшения лимита свободного пространства на дисках узла хранения до 10% узел автоматически переводится в режим «только чтение» и участвует в выборе подходящего для пользователя узла хранения только для чтения данных. Лимит в 10% введен для возможного использования сильно загруженных узлов в случаях крайней необходимости, например, при отсутствии свободного места на других узлах системы, а также для записи служебных данных при перемещении информации с одного узла хранения на другой.

Загрузка/прием данных

Загрузку данных в систему можно условно разделить на три этапа: прием данных от пользователя, обработка данных и определение узлов хранения и загрузка данных на узлы. Прием данных от пользователя осуществляется по HTTP протоколу через сервер приложений. Затем с использованием данных о состоянии узлов хранения, предоставляемых сервером статистики, определяются три узла хранения, на которые будут загружены три копии данных, полученных от пользователя. Обработчик метаданных генерирует уникальный строковый идентификатор файла, по которому можно будет однозначно определять его независимо от того, на скольких серверах он находится. Модуль загрузки поочередно соединяется с узлами хранения. Для аутентификации на узлах хранения используются специальные ключи безопасности, которые позволяют однозначно определить сервер метаданных. После успешной аутентификации файл загружается, и загрузчик переходит к следующему узлу хранения. В случае отказа одной (или двух) из выбранных целей хранения выполнение задания все равно считается успешным. Информация о необходимости загрузки дополнительных копий файла передается серверу производительности в модуль загрузки, который в дальнейшем, аналогично модулю загрузки сервера метаданных, обеспечивает необходимый уровень повторяемости данных.

Локаатор соединений

Локаторы соединений – это модули, реализующие интеллектуальный выбор узлов хранения, обеспечивающих наибольшее быстродействие обмена данными с конкретным пользователем. Локаторы соединений расположены на сервере приложений и узлах хранения. Статистические данные о скорости соединения пользователя с системой передаются во время загрузки данных на сервер приложений, а также при получении пользователем данных с узла хранения. Вначале для получения данных пользователь направляется на один из наименее загруженных узлов хранения. Данные о скорости загрузки направляются на сервер метаданных. В процессе работы пользователя с системой собирается статистика о скорости соединения его IP адреса с различными узлами хранения, составляется рейтинг скорости загрузки. В дальнейшем, при формировании ответа пользователю, происходит выборка не только наименее загруженного узла хранения, но и имеющего наилучшие скоростные характеристики обмена данными для данного пользователя [4].

Восстановление узла данных

Не менее важным моментом в работе всей системы является последовательность восстановления данных на узлах.

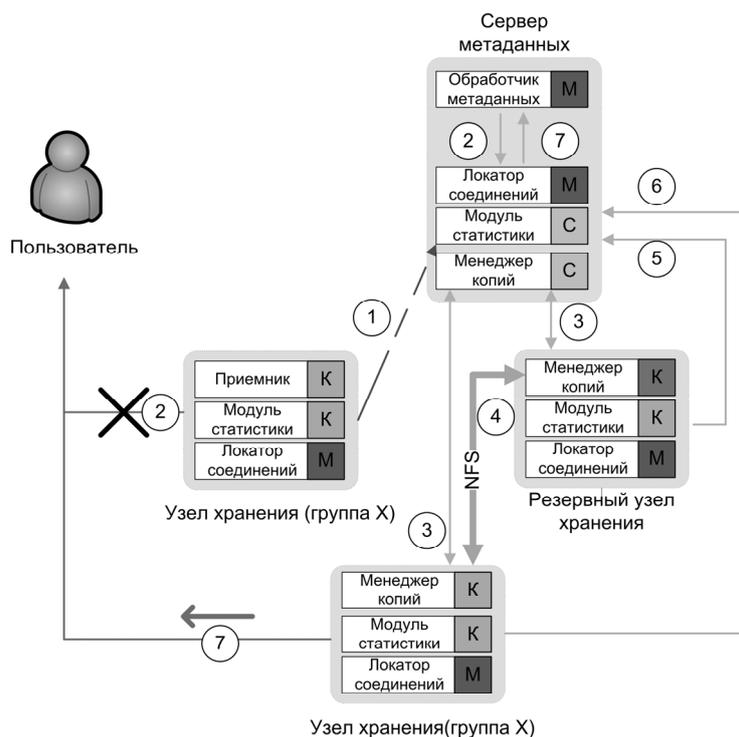


Рис. 2. Последовательность операций восстановления узла

В случае выхода из строя одного или двух узлов хранения данных, находящихся в одной группе, необходимо, чтобы система автоматически или с минимальным участием администратора восстановила свою нормальную работу. На рис. 2 представлена последовательность операций в системе в случае выхода из строя одного из узлов группы X, состоящей из трех узлов хранения с одинаковым набором данных.

В случае отсутствия отчетов модуля статистики от узла хранения принимается решение о его недееспособности, обработчик метаданных изымает узел из списка работоспособных, а затем инициируется процесс дублирования потерянного узла. Для этого менеджер копий на сервере метаданных выбирает из списка пустой резервный узел и запускает автономный процесс копирования данных между одним из узлов группы X и выбранным пустым узлом по протоколу NFS v4, что позволяет избежать участия сервера метаданных в этом процессе, однако требует высокоскоростного и защищенного соединения. По завершению копирования новый узел группы через модуль статистики сообщает о своей готовности и работоспособности, после чего узел заносится в список доступных для использования узлов [5].

Заключение

Применение сервис-ориентированной архитектуры в программной части реализации распределенной системы хранения данных позволяет отказаться от дорогостоящих аппаратных систем хранения данных при организации информационно-емких Интернет-сервисов. Кроме того, описанная выше идеология восстановления работоспособности узлов хранения данных позволяет легко масштабировать ресурсы системы, не уступая при этом в вопросах высокой доступности и надежности дорогостоящим многомашинным вычислительным комплексам. Гибкость и простота модульной системы позволяет наращивать функциональность как отдельных компонентов, так и всей системы в целом прямо в процессе эксплуатации. Так, уже на этапе эксплуатации распределенной системы хранения данных начал внедряться новый модуль – «локатор соединений», реализующий интеллектуальный выбор доступных зеркал данных на различных узлах хранения.

В ходе работы с помощью языков программирования были реализованы ранее недоступные аппаратные реализации алгоритмов резервирования данных. Разработан алгоритм выбора оптимального узла хранения, основанный на методе ранжирования с весовыми коэффициентами. Рассмотрена последовательность движения пользовательских данных внутри системы. Благодаря использованию внутреннего высокоскоростного соединения узлов хранения стало возможным переносить данные с узла на узел за минимально возможное время – 1500 Мбайт за 7,2 часа, что в три раза превосходит среднюю скорость восстановления RAID массива уровня 5.

В дальнейшем, по завершению всей запланированной функциональности, распределенная система хранения данных будет интегрирована с одной из популярных систем управления содержимым. Также в ближайшее время будет опубликован код системы, что позволит осуществлять совместную с открытым сообществом работу над улучшением системы, а также сделает доступным распределенную систему хранения данных любому желающему.

Литература

1. Лукьянов Н.М. Анализ факторов, влияющих на качественные и количественные показатели функционирования систем распределенного хранения данных // Научно-технический вестник СПбГУ ИТМО. – 2008. – № 56. – 9 с.
2. Hoff T. Google Architecture // HighScalability.com. – 2008. [Электронный ресурс] – URL: <http://highscalability.com/google-architecture> (дата обращения: 02.09.2010).
3. Корников В.В. Байесовская модель обработки нечисловой, неточной и неполной информации о весовых коэффициентах. – Санкт-Петербургский государственный университет, 2000 [Электронный ресурс] – URL: <http://www.inftech.webservis.ru/it/conference/scm/2000/session3/kornikov.htm> (дата обращения: 02.09.2010).
4. Vogt M., Troup J. Dynamic Mirror Decisions // OOO Canonical – 2006 [Электронный ресурс] – URL: <https://wiki.ubuntu.com/DynamicMirrorDecisions> (дата обращения: 02.09.2010).
5. Yokota H. A proposal of DNS-based adaptive load balancing method for mirror server systems and its implementation // Конференция Advanced Information Networking and Applications – США: IEEE Computer Society, 2004. – 208 с.

Лукьянов Николай Михайлович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, nikolay.lukianov@gmail.com

Дергачев Андрей Михайлович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, ст. преподаватель, dam600@gmail.com

УДК 778.38:777.6

МЕТОД СЖАТИЯ ДИНАМИЧЕСКОГО ДИАПАЗОНА ЦИФРОВЫХ ГОЛОГРАММ ФУРЬЕ В ЗАДАЧЕ ВСТРАИВАНИЯ СКРЫТЫХ ВОДЯНЫХ ЗНАКОВ

А.П. Старченко, Ю.А. Гатчин

Предлагается методика уменьшения искажений в изображении-контейнере со встроенной голограммой водяного знака, основанная на изменении динамического диапазона голограммы методом логарифмического преобразования. Предложенная методика позволяет снизить искажения, возникающие при восстановлении изображения водяного знака, по сравнению с традиционным методом голограммы Фурье. Выявлена линейная зависимость между изменением амплитуды сигнала водяного знака и средним квадратичным отклонением интенсивности цифровой голограммы Фурье.

Ключевые слова: водяной знак, голограмма, преобразование интенсивности.

Введение

Водяные знаки на основе голограмм устойчивы к различным преобразованиям изображения – масштабированию, геометрическим искажениям и др. При встраивании голограммы желательно уменьшить ее динамический диапазон для снижения искажений в изображении-контейнере.

В оптике для снижения динамического диапазона голограммы Фурье перед предметом устанавливают рассеиватель [1]. Рассеиватель не изменяет амплитуды, но изменяет случайным образом фазу в каждой точке, падающей на предмет волны. Случайная фаза не оказывает влияния на изображение предмета, получаемое при восстановлении амплитуды электромагнитной волны. В цифровой голографии тот же эффект достигается умножением исходного изображения водяного знака на произвольный фазовый множитель $\exp[jz(u, v)]$, где u, v – пространственные частоты. Случайная фаза $z(u, v)$ приводит к расширению спектра в частотной области голограммы и уменьшению диапазона изменения составляющих спектра [1], что значительно снижает требования к динамическому диапазону регистрирующего устройства.

Эффективность применения голограмм с рассеивателем бывает недостаточной в случае повышения требований к устойчивости скрытых водяных знаков, что на практике сводится к необходимости повышения отношения сигнал/шум за счет увеличения амплитуды сигнала водяного знака. При этом в изображении-носителе могут возникать сильные искажения, обусловленные внедрением голограммы водяного знака [2, 3].

Цель работы – показать возможность снижения искажений в изображении-контейнере при встраивании голограммы водяного знака за счет применения логарифмического преобразования интенсивности голограммы.

Анализ зависимости динамического диапазона интенсивности цифровой голограммы со случайной фазой от амплитуды сигнала водяного знака

Для улучшения устойчивости скрытых водяных знаков требуется повышение амплитуды сигнала водяного знака, что неминуемо отражается на величине искажений в изображении-носителе. Для оценки зависимости величины искажений в изображении-носителе от повышения амплитуды сигнала водяного знака рассмотрим изменение интенсивности цифровой голограммы со случайной фазой $z(u, v)$ от величины амплитуды сигнала водяного знака.

Распределение интенсивности в цифровой голограмме в рассматриваемом случае определяется соотношением

$$h(x, y) = F^{-1}\{bW(u - N, v - M) \exp(jz(u, v))\}, \quad (1)$$

где b – коэффициент, определяющий амплитуду сигнала водяного знака; $W(u-N, v-M)$ – функция изображения водяного знака; N и M – пространственные несущие в плоскости частот (u, v) ; F^{-1} – оператор обратного преобразования Фурье. Формула (1) описывает голограмму Фурье с одной боковой полосой [3].

Были проведены экспериментальные исследования при встраивании голограммы водяного знака в изображение для двух значений амплитуды сигнала водяного знака $b=50$ и $b=150$ для значений интенсивности в 256 градаций. В качестве примера водяного знака был использован знак ©. Выполнены расчеты среднего квадратичного отклонения (СКО) σ и дисперсии D интенсивности голограммы Фурье водяного знака при изменении значений амплитуды сигнала водяного знака b в диапазоне от 50 до 150. Полученные расчеты представлены в табл. 1.

b	σ	D
50	2,67	7,13
60	3,21	10,27
70	3,74	13,98
80	4,27	18,26
90	4,81	23,12
100	5,34	28,54
110	5,88	34,53
120	6,41	41,10
130	6,94	48,23
140	7,48	55,93
150	8,01	64,21

Таблица 1. Результаты расчета СКО и дисперсии интенсивности голограммы Фурье водяного знака

Из данных табл. 1 следует наличие линейной зависимости СКО интенсивности голограммы Фурье от амплитуды сигнала водяного знака в довольно широком диапазоне изменений значений амплитуды, что позволяет устанавливать требуемое снижение динамического диапазона в голограмме.

Исследование логарифмического преобразования интенсивности цифровой голограммы Фурье со случайной фазой

Для снижения динамического диапазона в голограмме предлагается использовать метод логарифмического преобразования, которое используется в задачах гомоморфной фильтрации [4] изображений. В ряде случаев используются нелинейные преобразования, которые подчиняются обобщенному принципу суперпозиции и получили название гомоморфных систем обработки сигналов. Применение гомоморфной обработки изображений позволяет, с одной стороны, повысить контрастность, а с другой стороны – сжать динамический диапазон [4].

Применительно к задаче уменьшения динамического диапазона интенсивности цифровых голограмм важной операцией при гомоморфной обработке является операция логарифмирования или вычис-

ления степенной функции вида $x^{p/q}$, где $p/q < 1$. Как известно, эти операции имеют схожие характеристики: как логарифмическая функция, так и степенная функция при $p/q < 1$ обеспечивают снижение уровня сигналов с большей амплитудой по сравнению с сигналами незначительной амплитуды.

Для оценки зависимости логарифмического преобразования интенсивности цифровой голограммы Фурье от преобразования интенсивности без логарифмирования исследуем изменение СКО интенсивности цифровой голограммы со случайной фазой и изменение отношения сигнал/шум (С/Ш) при восстановлении водяного знака от величины амплитуды сигнала водяного знака. Для описания процедуры встраивания водяных знаков $W(u, v)$ в предметной области будем использовать формулу

$$s(x, y) = g(x, y) + f\{h(x, y)\}, \quad (2)$$

где $s(x, y)$ – функция изображения со встроенной голограммой водяного знака; $g(x, y)$ – функция изображения-контейнера; $h(x, y)$ – функция цифровой голограммы Фурье, рассчитанная по формуле (1), которая имеет явную зависимость от амплитуды сигнала водяного знака b ; f – функция логарифмического преобразования интенсивности голограммы.

Для оценки отношения С/Ш воспользуемся формулой

$$C/Ш = \sum \sum [W(u, v) - a][W_1(u, v) - a_1] dudv / \sum \sum s^2(x, y) dx dy, \quad (3)$$

где $W(u, v)$ и $W_1(u, v)$ – сравниваемые сигналы водяного знака; a и a_1 – средние значения сигналов W и W_1 , соответственно. Ковариация (3) вычисляется при полном совмещении исходного изображения водяного знака W и восстановленного изображения W_1 .

В табл. 2 представлены результаты оценки зависимости СКО распределения интенсивности в голограммах со случайной фазой и зависимости отношения С/Ш при восстановлении водяных знаков от величины изменения амплитуды сигнала водяного знака b .

b	СКО ₁	СКО ₂	С/Ш ₁	С/Ш ₂
50	2,67	3,55	2,11	2,26
60	3,21	3,88	2,21	2,31
70	3,74	4,46	2,27	2,38
80	4,27	4,62	2,36	2,40
90	4,81	5,00	2,42	2,44
100	5,34	5,21	2,47	2,46
110	5,88	5,75	2,52	2,51
120	6,41	6,21	2,56	2,55
130	6,94	6,62	2,60	2,57
140	7,48	6,81	2,62	2,59
150	8,01	6,97	2,65	2,60

Таблица 2. Оценки СКО интенсивности голограмм Фурье со случайной фазой и значения отношения С/Ш при восстановлении водяных знаков

В табл. 2 столбцы значений СКО₁ и СКО₂ являются средними значениями СКО интенсивности голограмм для двух случаев: без применения амплитудного преобразования и с применением степенного преобразования вида

$$h_m(x, y) = y_m h^{p/q}(x, y) \quad (4)$$

соответственно. Параметр y_m определяется по формуле $y_m = B_m / B_m^{p/q}$ и представляет собой, как будет рассмотрено ниже, угол наклона искомой зависимости, где B_m – максимальное значение интенсивности голограммы, $h_m(x, y)$ – голограмма после нелинейного амплитудного преобразования. Столбцы значений С/Ш₁ и С/Ш₂ являются средними значениями отношения С/Ш для этих двух случаев, соответственно. Результаты экспериментов в табл. 2 получены при $p=5$, $q=8$, $y_m=2,15$, размере водяного знака © 32×32 пикселей. В качестве контейнера использовалось тестовое изображение «Леппа» форматом 256×256 пикселей.

На рис. 1, а, и рис. 1, б, представлены изображение-контейнер со встроенной голограммой водяного знака без предварительного преобразования интенсивности и восстановленный водяной знак соответственно, а на рис. 2, а, и рис. 2, б, показаны изображение после нелинейного преобразования интенсивности голограммы и восстановленный водяной знак соответственно. В обоих случаях значение амплитуды сигнала водяного знака составляло $b=150$. Задание такого значения амплитуды сигнала водяного знака позволяет оценить полученный эффект снижения динамического диапазона интенсивности голограммы.

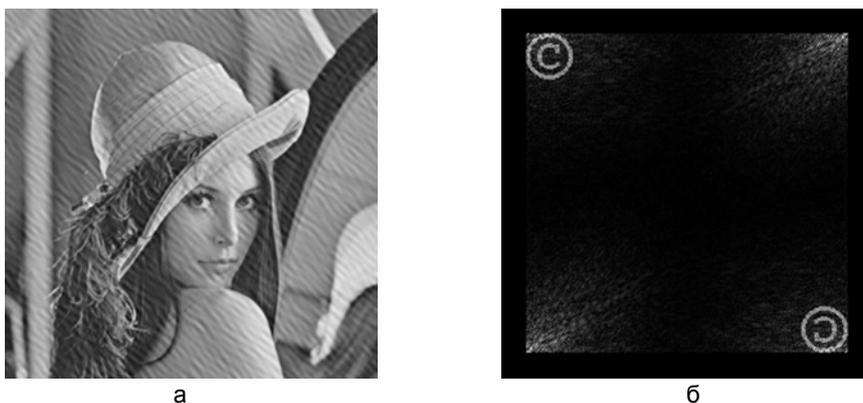


Рис. 1. Изображение-контейнер со встроенной голограммой водяного знака без предварительного преобразования интенсивности голограммы (а) и восстановленный водяной знак (б)

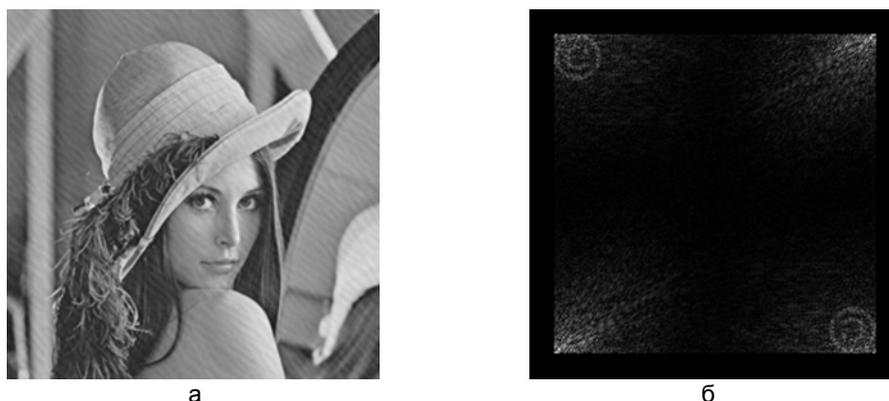


Рис. 2. Изображение-контейнер после нелинейного преобразования интенсивности голограммы (а) и восстановленный водяной знак (б)

На рис. 3 представлен график зависимости СКО интенсивности голограммы Фурье от амплитуды сигнала водяного знака (на графике 1 – результат степенного преобразования, 2 – без логарифмического преобразования). Из анализа полученных зависимостей на рис. 3 видно, что результаты логарифмического преобразования интенсивности голограммы имеют как положительный, так и отрицательный эффект. Результаты, связанные со снижением динамического диапазона голограммы, встраиваемой в контейнер, начинают проявляться с порогового значения амплитуды сигнала b_m водяного знака. При относительно небольших значениях амплитуды сигнала водяного знака СКО интенсивности голограммы, после нелинейного амплитудного преобразования, превышает соответствующее значение СКО голограммы без преобразования интенсивности. Однако при этом отношение С/Ш для голограммы с амплитудным нелинейным преобразованием превышает соответствующие значения для голограммы без нелинейного преобразования интенсивности.

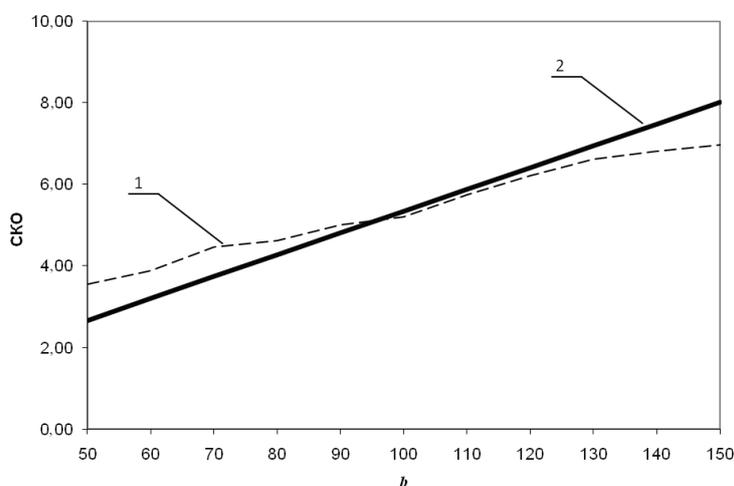


Рис. 3. Зависимости СКО интенсивности голограммы Фурье от амплитуды сигнала водяного знака: 1 – результат степенного преобразования; 2 – без логарифмического преобразования

При увеличении значения амплитуды сигнала водяного знака ситуация меняется на противоположную, а именно, при больших значениях амплитуды сигнала водяного знака динамический диапазон интенсивности голограммы снижается, а СКО достигает 12% при значении амплитуды сигнала водяного знака $b=150$. Для выбранных параметров формирования голограммы и параметров степенного преобразования $p/q=5/8$, $y_m=2,15$ положительный эффект сжатия интенсивности голограммы начинает наблюдаться со значений амплитуды сигнала водяного знака, примерно равных $b_m=95$.

Из графика на рис. 3 следует, что коэффициент пропорциональности y_m определяет угол наклона рассматриваемых зависимостей для случая применения нелинейного преобразования интенсивности голограммы. На рис. 4 представлены графики степенной функции уровней квантования b при $p/q=5/8$ для трех вариантов угла наклона y_m : 1,0; 2,15 и 5,0. При этом входное изменение уровней квантования $B1$ задавалось в диапазоне 0–255. Крутизна выходных характеристик $B2$ изменяется в зависимости от параметра y_m и может регулироваться в широких пределах.

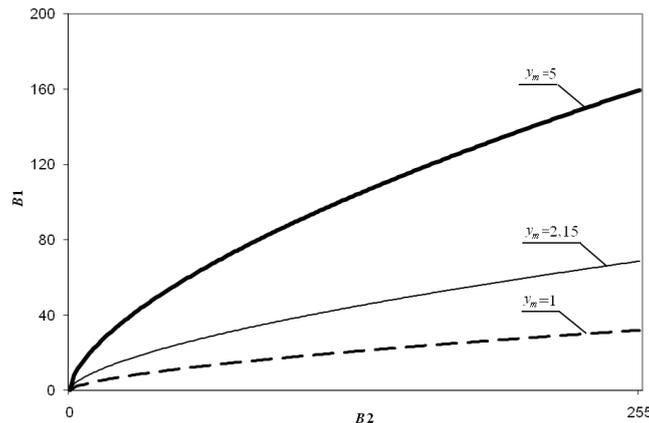


Рис. 4. График степенной функции уровней квантования b при $p/q=5/8$ для трех вариантов угла наклона y_m : 1,0; 2,15 и 5,0

Таким образом, в процессе экспериментов по применению логарифмического сжатия динамического диапазона интенсивности голограмм Фурье со случайной фазой установлено, что положительный эффект начинает выполняться с порогового значения амплитуды b_m водяного знака, величина которого определяется параметрами степенного преобразования $x^{p/q}$ ($p/q < 1$) и параметром наклона y_m , который влияет на крутизну характеристики преобразования.

Заключение

Установлена линейная зависимость между изменением амплитуды сигнала водяного знака и средним квадратичным отклонением интенсивности цифровой голограммы Фурье.

Проведенные эксперименты логарифмического сжатия интенсивности голограмм водяных знаков, встраиваемых в изображение-контейнер, показали существенное снижение динамического диапазона голограммы в области больших значений амплитуды сигнала водяного знака. Для значения амплитуды сигнала водяного знака $b=150$ и параметров степенного преобразования (4) $p=5$, $q=8$, $y_m=2,15$, получено уменьшение динамического диапазона ΔB на 37%, при этом уменьшение среднего квадратичного отклонения интенсивности голограммы составило около 12%.

Установлено, что эффект логарифмического преобразования интенсивности голограммы не является однозначным по мере увеличения амплитуды сигнала водяного знака. В области небольших значений амплитуды сигнала водяного знака СКО превышает соответствующее значение для голограммы без преобразования интенсивности. При увеличении амплитуды сигнала водяного знака СКО интенсивности нелинейно преобразованных голограмм снижается и становится меньше соответствующей характеристики голограмм без этого преобразования.

Литература

1. Короленко П.В. Оптика когерентного излучения. – М.: Изд-во МГУ, 1997. – 222 с.
2. Гендин В.Г. Формирование и идентификация изображений со встроенными водяными знаками методами цифровой голографии // Труды научно-исследовательского центра фотоники и оптоинформатики. – СПб: СПбГУ ИТМО. – 2009. – С. 406–415.

3. Смирнов М.В. Голографический подход к встраиванию скрытых водяных знаков в фотоизображение // Оптический журнал. – 2005. – Т. 72. – № 6. – С. 51–56.
4. Оппенгеймер А.В., Шафер Р.В. Цифровая обработка сигналов. – М.: Связь, 1979. – 416 с.

- Старченко Алексей Петрович* – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, lexis_spb84@mail.ru
- Гатчин Юрий Арменакович* – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, зав. кафедрой, gatchin@mail.ifmo.ru

УДК 004.89

ОЦЕНКА КОЛИЧЕСТВА ИНФОРМАЦИИ В БАЗАХ ЗНАНИЙ

И.А. Бессмертный

Обсуждается проблема построения систем искусственного интеллекта (ИИ) в части измерения количества информации, содержащейся в формализованных знаниях в виде фактов. Рассматривается метод измерения информативности знаний, одинаково пригодный для оценки естественного и искусственного интеллекта. В целях устранения неопределенности пространства понятий в работе предлагается оценивать информативность понятий и фактов в рамках контекста.

Ключевые слова: искусственный интеллект, количество информации, контекст.

Введение

Задача оценки уровня ИИ возникла одновременно с введением данного понятия, и наиболее важный вклад в понимание этой проблемы внес А. Тьюринг [1]. Тест Тьюринга, однако, является сугубо качественной оценкой ИИ в сравнении с естественным интеллектом (ЕИ) и принципиально не может дать ответа на вопрос, каким объемом базы знаний должен располагать ИИ для решения тех или иных задач и в каких единицах этот объем следует измерять.

Существующие методы оценки объемов знаний, в основном, базируются либо на объемах текстовых документов в символах или словах, либо на измерении времени, затраченного на приобретение этих знаний. Эти методы широко применяются, например, в образовательной деятельности, хотя и обладают очевидными недостатками: объем текста далеко не всегда отражает его информативность, а время обучения характеризует лишь предоставленную возможность приобретения знаний, но не результат обучения. Кроме того, эти методы не могут применяться к оценке ИИ.

В качестве единицы знаний (атома знания) иногда выбирают факт, соответствующий триплету субъект–предикат–объект (СПО), используемому в семантических сетях. В работе [2] делается анализ производительности образовательного процесса в терминах усваиваемых фактов, что позволяет связать время обучения с объемом приобретаемых знаний. В [3] приведены результаты оценки информативности слабо формализованных знаний, содержащихся в энциклопедиях и словарях, в тех же единицах СПО.

Очевидно, однако, что информативность разных фактов не является одинаковой. Кроме того, данный подход не согласуется с теорией информации, созданной К. Шенноном. Таким образом, представляется целесообразным разработать систему метрик и методику измерения количества информации, содержащейся в отдельных высказываниях и в базах знаний в целом.

Количество информации в слове как смысловой единице

В текстах на естественном языке минимальной смысловой единицей является слово. Очевидно, что каждое слово несет определенную информацию, количество которой должно поддаваться измерению. Согласно Шеннону, единицей информации является бит как величина энтропии для одного из двух равновероятных событий. Общая формула энтропии $H(x)$ для n состояний случайной величины x имеет вид

$$H(x) = -\sum_{i=1}^n p(x_i) \log_2 p(x_i).$$

Для двоичной пары (0, 1) энтропия равна единице. Таким образом, число двоичных разрядов в точности соответствует числу битов информации при условии, что для всех разрядов вероятности нулей и единиц равны. Количество информации I определяется как двоичный логарифм числа состояний n :

$$I = \log_2 n.$$

Слова (исключая иероглифическое письмо) состоят из букв. Количество информации в одной букве русского алфавита (в предположении, что вероятности всех букв равны) приблизительно равно пяти битам, что соответствует пяти двоичным разрядам, которыми русский алфавит как раз и может быть закодирован (если отбросить букву «ё»).

Следует признать, что механическое перенесение энтропии алфавита на составляемые с его помощью слова лишено смысла, иначе длинные слова всегда будут более информативными, а английский

текст – менее информативным, чем русский, поскольку не только латинский алфавит содержит меньше знаков, но и английские слова в среднем короче. Слово – это код для обозначения понятия, причем код заведомо избыточный. Пусть средняя длина слова равна шести знакам, каждый из которых кодируется пятью битами, т.е. имеет $32^6 \approx 1$ млрд. слов, следовательно, избыточность составляет не менее двух порядков. Количество информации в слове можно определять как логарифм объема словаря, однако в таком случае ситуация не лучше, так как объем словаря определить однозначно невозможно.

Другое соображение, не позволяющее механически вычислить информативность фразы как произведение числа слов на количество информации в слове, исходит из невозможности вычислить информативность числа по Шеннону, не зная его разрядности, т.е. возможного числа состояний. Таким образом, приходим к невозможности строгого вычисления количества информации в текстах на естественных языках.

Сказанное не означает, что число слов в тексте не может служить приблизительной оценкой его информативности. В качестве примера рассмотрим два выражения: «британец» и «подданный ее Величества Королевы Елизаветы II». С одной стороны, эти понятия эквивалентны, с другой – различаются по количеству слов, а значит, содержащейся в них информации, в шесть раз. Однако при внимательном рассмотрении тождественность этих выражений оказывается относительной. На самом деле второе выражение более информативно, поскольку британец может быть подданным короля Ричарда Львиное Сердце или любого другого британского монарха. Выражение «современный британец» уже более приближено ко второму выражению, но опять-таки не идентично, поскольку привязано к настоящему времени. Любые попытки привести первое выражение по смыслу ко второму неизбежно увеличат число содержащихся в нем слов.

Информативность понятий в базах данных и знаний

Попытаемся определить, каким образом можно измерить количество информации, если не в произвольных текстах, то, по меньшей мере, в понятиях, используемых в науке и технике. Отметим следующее.

- Любые понятия и термины существуют в контекстном окружении. Выражение «характеристики поля» имеет совершенно разные значения в журналах по физике, агрономии или футбольном обозрении.
- Информативность разных понятий разная. Слово «монголоид» означает принадлежность к одной из четырех рас, «китаец» – представитель одной из примерно двухсот наций, «Конфуций» – с одной стороны, один из миллиардов людей, живших на Земле, а с другой стороны – один из немногих величайших мыслителей всех времен и народов.
- В каждом контексте все понятия определяются на некотором уровне абстракции. Если речь идет о студенческой группе, то понятие «студент» нас интересует как объект преподавания или член студенческого сообщества, но не как биологический индивид. Мы абстрагируемся от его анатомии или метаболизма.

Пусть понятие e_i определено на множестве $E_i = \{e_i\}$, имеющем мощность n_i . Тогда количество информации $I(e_i)$, содержащегося в e_i , будет равно

$$I(e_i) = \log_2 n_i.$$

В свою очередь, само множество E_i является понятием более высокого уровня, определенного на множестве E_{i+1} . Аналогично, понятие e_i является понятием более высокого уровня для множества E_{i-1} . Таким образом, получаем иерархию понятий, информативность которых на каждом уровне можно вычислить. Обобщающее множество E_i будем называть контекстом (рисунок).

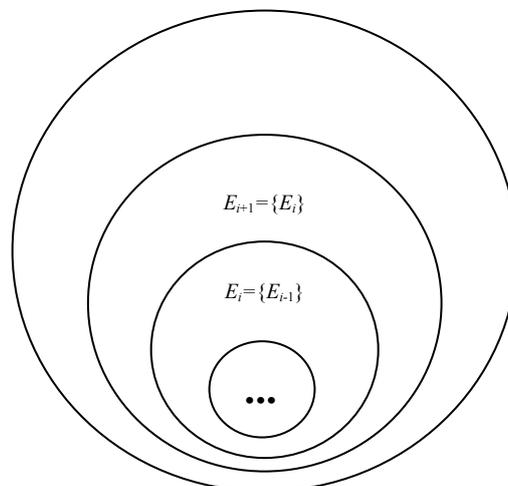


Рисунок. Вложенность контекста

При объединении нескольких уровней контекста информативность понятия нижнего уровня определяется как

$$I(e_i) = \sum_{k=i}^j \log_2 n_k, \tag{1}$$

где j – верхний уровень контекста.

Рассмотрим этот подход на примере университета. В контексте студенческой группы, состоящей из 25 человек, информативность выбора конкретного студента равна $\log_2 25 = 4,644$. Пусть 4 группы образуют поток, 6 потоков (по годам поступления) – кафедру, факультет объединяет пять кафедр, а университет насчитывает 12 факультетов. Тогда информативность выбора студента из всего потока будет равна 6,644, из всех студентов кафедры – 9,234 и т.д. (табл. 1).

Уровень контекста	Мощность множества состояний n	Количество информации $I(n)$	
		Для контекста	Для студента
Группа	25 студентов	4,644	4,644
Поток	4 групп	2,0	6,644
Кафедра	6 потоков	2,59	9,234
Факультет	5 кафедр	2,32	11,554
Университет	12 факультетов	3,58	15,134
Страна	600 университетов	9,23	...
Земля

Таблица 1. Информативность выбора студента на разных уровнях контекста

Таким образом, устанавливая уровень контекста, можно определить количество информации, которое несет тот или иной выбор, а при необходимости – расширить контекст, например, как в табл. 1, до уровня страны или в мировом масштабе.

Следует отметить, что мощность множеств в разных контекстах на одном уровне может быть разной. В таком случае информативность однотипных событий также будет отличаться, что вполне логично. Однако при объединении общим контекстом информативность, вычисляемая по формуле (1), также будет разной.

Группа 1		Группа 2		Группа 3		Группа 4		Группа 5		Группа 6	
n	$I(n)$	n	$I(n)$	n	$I(n)$	n	$I(n)$	n	$I(n)$	n	$I(n)$
16	4	10	3,32	18	4,17	22	4,46	19	4,25	20	4,32
Поток 1 (4 группы: $n=4$; $I(n)=2$)						Поток 2 ($n=2$; $I(n)=1$)					
	6		5,32		6,17		6,46		5,25		5,32
Кафедра А (2 потока: $n=2$; $I(n)=1$)											
	7		6,32		7,17		7,46		6,25		6,32

Таблица 2. Информативность выбора студента при различных мощностях множеств

Из табл. 2 видно, что информативность выбора студента зависит от численности его группы, числа групп в потоке, числа потоков, образующих кафедру. В результате информативность понятия «студент кафедры А» не является одинаковой для всех студентов, обучающихся на данной кафедре.

Решение может заключаться в том, чтобы на каждом уровне контекста вычислялась суммарная мощность множества всех сущностей:

$$I(e_i, E_j) = \log_2 \sum_{n_i \in E_j} n_i.$$

Если применить эту формулу к рассматриваемому примеру, информативность выбора студента из потока будет вычисляться, исходя из численности всех студентов потока, из кафедры – исходя из числа всех студентов, обучающихся на кафедре, и т.д. (табл. 2, 3).

Группа 1		Группа 2		Группа 3		Группа 4		Группа 5		Группа 6	
n	$I(n)$	n	$I(n)$	n	$I(n)$	n	$I(n)$	n	$I(n)$	n	$I(n)$
16	4	10	3,32	18	4,17	22	4,46	19	4,25	20	4,32
Поток 1 (66 студентов: $n=66$; $I(n)=6,044$)						Поток 2 ($n=39$; $I(n)=5,28$)					
Кафедра А (105 студентов: $n=105$; $I(n)=6,71$)											

Таблица 3. Информативность выбора студента

Таким образом, информативность выбора некоторой сущности может быть вычислена на разных уровнях контекста.

Контекст и идентификация понятий

С информативностью понятий связана также проблема их идентификации в базах данных и знаний. В работе [4] показано, что визуализация знаний возможна только в определенном контексте. Точно так же любой текст или человеческий диалог всегда строится с учетом контекста, благодаря чему достигается компактность при сохранении точности передаваемого смысла высказываний. В естественном языке говорящий (или пишущий) должен всегда быть уверен в том, что слушатель или читатель правильно понимает контекст, для чего делаются соответствующие вводные предложения в начале текста или диалога. Если же требуется использовать один и тот же объект в разных контекстах, то к идентификатору объекта добавляется идентификатор контекста. В рассмотренном выше примере студенческого сообщества, если высказывание делается в контексте группы, то для идентификации экземпляра достаточно фамилии; в контекстах более высокого уровня к фамилии добавляется номер курса, факультета, университета и т.д.

В последнее время идет активный процесс формализации знаний в стандарте Глобальной семантической сети (Semantic Web, SW). Для каждой предметной области в SW формируются онтологии, т.е. документы, подготовленные на OWL или других языках и содержащие описание классов объектов, отношений между ними и свойств. С использованием онтологий создаются документы, содержащие факты, относящиеся к заданной предметной области. Информативность таких документов можно оценивать с применением данного подхода. В частности, может быть разработан модуль в среде популярного редактора онтологий Protégé (<http://protege.stanford.edu>), вычисляющий информативность понятий с целью контроля создаваемых онтологий и выявления ошибок. Еще одно применение данного подхода возможно при создании и поддержке крупных баз данных, в которых иногда происходят нарушения целостности данных вследствие переполнения индексов. Разрядность индекса не должна быть меньше информативности соответствующей ему сущности.

Заключение

Предлагаемый подход позволяет измерять количество информации, содержащейся в отдельных понятиях, лежащих в основе баз знаний, в общепринятых единицах теории информации. Использование понятия контекста позволяет устранить проблемы полисемии и других неоднозначностей, а также естественным образом ограничить пространство состояний предметной области. Следующий этап исследований, который может базироваться на предложенном подходе, – оценка информативности сложных высказываний, состоящих из понятий, количество информации в каждом из которых известно.

Литература

1. Рассел С., Норвиг П. Искусственный интеллект: Современный подход. – 2-е изд. Пер. с англ. – М.: Вильямс, 2006. – 1408 с.
2. Богданов И.В. Учебная информация и единицы ее измерения // Труды СГУ. – Вып. 44. – Гуманитарные науки. Психология и социология образования. – М.: Изд-во СГУ, 2002.
3. Бессмертный И.А. Семантическая паутина и искусственный интеллект // Научно-технический вестник СПбГУ ИТМО. – 2009. – № 6(64). – С. 77–82.
4. Bessmertny I.A. Knowledge Visualization Based on Semantic Network // Programming and Computer Software. – 2010. – V. 36. – № 4. – P. 197–204.

Бессмертный Игорь Александрович – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, igor_bessmertny@hotmail.com

УДК 004.056

УЯЗВИМОСТИ МОНИТОРОВ ВИРТУАЛЬНЫХ МАШИН

Ю.Е. Евелев, Г.М. Чернокнижный

Рассмотрены проблемы безопасности систем виртуализации. Приведены уязвимости известных мониторов виртуальных машин (ВМ). Даны примеры эксплойтов. Рассмотрена технология VMsafe и ее возможности по увеличению эффективности системы защиты с учетом использования вендоров безопасности.

Ключевые слова: виртуализация, уязвимость, вендоры безопасности, эксплойт.

Введение

Несмотря на то, что виртуальные серверы нематериальны, они точно так же уязвимы, как и обычные физические серверы. Деление физического сервера на отдельные ВМ никоим образом не делает эти машины безопаснее – виртуализация естественным образом передает им хорошо известные врожденные

слабости стандартных серверов. Ставится задача: рассмотреть основные угрозы, возникающие в виртуальной вычислительной среде, проанализировать недостатки ВМ с точки зрения безопасности в плане общего подхода и конкретно для известных гипервизоров. Анализ современных технологий повышения безопасности ВМ позволяет сформулировать рекомендации по их эффективной защите.

Основные угрозы при использовании виртуализации

Угрозы при использовании виртуальных машин можно разделить на «старые» и «новые». Под старыми угрозами понимаются традиционные угрозы, присущие любой ИТ-инфраструктуре. Это уязвимости операционных систем, уязвимости исполняемых программ и др. Под новыми угрозами понимаются уязвимости гипервизора (программного или аппаратного) и уязвимости менеджмента (управления) сложной виртуальной инфраструктурой:

- захват гостевой машины;
- вывод из строя виртуальной машины;
- компрометация управления ВМ;
- запуск неавторизованных ВМ;
- изменение разделения ресурсов – процессор, память, приоритеты;
- выключение машин;
- компрометация VMM/Hypervisor;
- организационные проблемы – разделение полномочий, политики информационной безопасности (ИБ), разграничение доступа.

Таким образом, при использовании виртуализации количество возможных уязвимостей возрастает, а традиционные угрозы (вирусы, DoS / DDoS атаки, переполнение буферов, SQL-injection, XSS, утечки информации) остаются. Основные недостатки виртуальных машин с точки зрения ИБ сводятся к следующему:

- Необходимость обновлять пакеты (программы) системы, что не всегда возможно и удобно. Обновление необходимо для устранения текущих уязвимостей и установки патчей. Обновить виртуальную машину довольно сложно, так как не всегда известно, на каком сервере она находится (Live Migration), в случае неграмотного менеджмента есть вероятность запутаться. В связи с этим необходимо четко задать каталогизацию виртуальных машин;
- Snapshots / Suspend затрудняют процесс управления обновлениями. Snapshot (снимок) – моментальный снимок, копия файлов и директорий файловой системы на определенный момент времени. Это довольно полезная функция виртуальных машин, так как, например, в случае падения сервера восстановить его из снимка – дело нескольких десятков минут. С другой стороны, снимки занимают довольно большие объемы винчестеров, само создание снимка – довольно длительный процесс. Чаще всего в момент создания снимка ВМ недоступна для конечных пользователей. Suspend – это пауза системы. Она может требоваться, например, для снятия нагрузки с энергосети на ночь. В случае, если той же ночью необходимо сделать резервную копию, это будет невозможно;
- Возможность разрастания неизвестных ВМ. В ходе работы с виртуальной инфраструктурой постоянно создаются новые ВМ, удаляются старые, некоторые ВМ теряются по тем или иным причинам (например, Live Migration). В этом случае происходит разрастание ВМ. Дабы оградиться от этого недостатка, необходим грамотный менеджмент виртуальной сети;
- Сложно контролировать ВМ, возможно появление неуправляемых, неизвестных ВМ;
- Динамическое перемещение (Live Migration);
- Перемещение ВМ в менее защищенные машины, сети;
- Атаки методом повтора операций и повторного использования данных;
- Проведение повторяющихся операций внутри ВМ может привести к повторяющимся криптографическим атакам;
- Воровство: ВМ – это файлы, очень просто украсть всю систему или несколько систем;
- Атаки на гипервизор. Гипервизор – это программа. Уязвимости программ были и будут всегда. В случае ограниченного финансирования необходимо четко расставить приоритеты между дополнительными функциями гипервизора и его безопасностью. Дополнительные функции увеличивают функционал гипервизора, но это и новые строки кода, а значит, и новые уязвимости [1].

Проблема безопасности виртуальной инфраструктуры может быть условно разделена на две составляющие:

1. безопасность виртуальных машин;
2. безопасность платформы виртуализации.

В первом случае, так же, как и на физической платформе, необходима установка средств защиты в гостевой операционной системе (антивирусы, сетевые экраны и прочее), а также правильная настройка виртуальных машин и виртуальных сетей. Например, в некоторых продуктах VMware (Workstation, Server) виртуальные коммутаторы являются по своей природе «хабами», что может открыть возможно-

сти по прослушиванию незащищенного трафика другими ВМ, контролируемые злоумышленниками, и ввести в заблуждение системных администраторов [2].

В отношении безопасности платформы виртуализации необходимо проанализировать отчеты об уязвимостях платформ и учитывать этот фактор при выборе конкретной платформы. Частью обслуживания виртуальной инфраструктуры должна стать своевременная установка обновлений системы виртуализации и отслеживание критических уязвимостей. При этом необходимо также учитывать возможность внутреннего несанкционированного доступа в пределах хостовой системы. Сейчас атаки на виртуальные системы достаточно редки, ввиду не такого широкого распространения платформ виртуализации, поэтому истинная защищенность платформ виртуализации неизвестна. Аналитики предсказывают, что к 2015 году каждая вторая операционная система будет виртуализована, и со временем вопрос безопасности будет становиться все более серьезным.

Аппаратная виртуализация значительно упрощает написание программного обеспечения (ПО) с использованием технологий виртуализации, что значительно увеличит в ближайшее время объем подобного вредоносного ПО. О реальной опасности говорить еще рано, поскольку трудозатраты на его написание все еще достаточно велики. Компания Microsoft, разработавшая SubVirt, неоднократно заявляла, что затраты на его реализацию сравнимы с затратами на создание операционной системы. Тем не менее, эту опасность стоит иметь в виду, а производители антивирусного ПО в ближайшее время должны включить в свои системы возможности обнаружения руткитов, использующих виртуализацию.

Уязвимости программных средств виртуализации

Уязвимости продуктов VMware перечислены ниже.

1. Уязвимость из-за ошибки в проверке входных данных в драйвере «HGFS.sys» из пакета VMware Tools. Атакующий может передать специально сформированные данные, что приведет к выполнению произвольного кода с повышенными привилегиями на гостевой Windows-системе.
2. Уязвимость из-за ошибки в проверке входных данных в «vmware-authd». Атакующий может передать специально сформированные данные, что приведет к выполнению произвольного кода с повышенными привилегиями на гостевой Linux-системе.
3. Уязвимость из-за ошибки в проверке входных данных в сервисе управления Openwsman при обработке «Content-Length» заголовков. Атакующий может получить привилегии «root» на целевой системе.
4. Уязвимость из-за ошибки в проверке входных данных в VMware VIX API. Атакующий может передать специально сформированные данные, что приведет к выполнению произвольного кода на хостовой системе.

Уязвимости продуктов SUN/ORACLE Virtual Box. В Virtual Box, как и в любом другом программном продукте, есть множество уязвимостей. Выделенная в данной работе уязвимость связана с уровнем привилегий в системе и позволяет локальному злоумышленнику выполнить вредоносные действия с повышенными привилегиями на целевой системе. Уязвимость существует из-за ошибки в проверке входных данных в драйвере VBoxDrv.sys при обработке определенных IOCTL. Атакующий может передать специально сформированные данные, что приведет к выполнению произвольного кода с привилегиями ядра.

Пример эксплойта:

```
#include <windows.h/>
#include <stdio.h/>
int main(int argc, char **argv)
{
    HANDLE hDevice;
    DWORD cb;
    char szDevice[] = "\\.\VBoxDrv";
    if ( (hDevice = CreateFileA(szDevice,
        GENERIC_READ|GENERIC_WRITE,
        0,
        0,
        OPEN_EXISTING,
        0,
        NULL) ) != INVALID_HANDLE_VALUE )
    {
        printf("Device %s succesfully opened!\n", szDevice);
    }
    else
    {
        printf("Error: Error opening device %s\n",szDevice);
    }
}
```

```

cb = 0;
if (!DeviceIoControl(hDevice,
0x228103,
(LPVOID)0x80808080,0,
(LPVOID)0x80808080,0x0,
&cb,
NULL))
{
printf("Error in DeviceIo ... bytes returned %#x\n",cb);
}
}
}

```

Уязвимости MS Virtual PC. Основной уязвимостью остается ошибка в управлении памятью на уровне монитора виртуальных машин (Virtual Machine Monitor). Она позволяет обойти такие защитные механизмы, как предотвращение выполнения данных (DEP), безопасная обработка исключений и рандомизация адресного пространства (ASLR) и может быть использована для повышения уровня привилегий или удаленно для выполнения кода. В своем блоге Microsoft отказывается признавать данную брешь уязвимостью, называя ее лишь способом более легкой эксплуатации уже имеющихся уязвимостей.

Уязвимости XEN. Одна из основных уязвимостей данного продукта позволяет локальному злоумышленнику обойти ограничения безопасности на целевой системе. Уязвимость существует из-за ошибки в проверке входных данных в Xen pygrub. Эксплуатирование уязвимости приведет к обходу механизма аутентификации на целевой системе.

Пример эксплойта:

```

xm create -c guest
press space bar to stop the grub count down
press e to edit
select the kernel line and press e
Append a "1" to the end of the kernel line and press retur
press "b" to boot

```

Другой, не менее важной проблемой является уязвимость, которая позволяет локальному злоумышленнику выполнить DoS атаку на целевую систему. Уязвимость существует по двум причинам:

1. Из-за ошибки в проверке входных данных при обработке DR7 регистра отладки. Атакующий из вне гостевой системы может установить определенные точки останова, что приведет к краху гипервизора. Успешное эксплуатирование уязвимости, возможно, потребует использование гипервизора HVM;
2. Из-за того, что доступ к регистру CR4 неправильно проверяется. Атакующий из вне гостевой системы может выполнить действия, приводящие к краху DomU или Dom0 доменов [3].

Вендоры безопасности и SECURITY API на примере VMSafe

В 2008 году компания VMware анонсировала технологию VMSafe. Реализацию VMSafe на данный момент можно увидеть только в программном пакете VMware vSphere, выпущенном в мае 2010 года. По своей сути, VMSafe – это технология, позволяющая сторонним разработчикам получить доступ к гипервизору VMware и фактически представляющая собой набор API интерфейсов. Эти интерфейсы доступны по умолчанию в VMware vSphere, использовать этот набор API могут только VM управляемые решением по ИБ. Появление этой технологии позволило VMware, не проходя самостоятельного пути по созданию комплекса средств защиты, привлечь к обеспечению ИБ инфраструктур виртуализации, построенных на VMware vSphere, лидеров рынка ИБ.

В виртуальной среде при помощи технологии VMSafe и решений производителей средств ИБ (не исключая при этом разработку внутренних стандартов/политик обеспечения безопасности и следование рекомендациям производителей) можно обеспечить более высокий уровень ИБ, чем на физическом уровне.

Согласно официальной информации компании VMware, преимущества решений с использованием технологии VMSafe включают:

- Better Security – улучшение возможностей по обеспечению безопасности, появление новых возможностей мониторинга инфраструктуры не имеющих аналогов в физической инфраструктуре;
- Better isolation – улучшение изоляции гостевых операционных систем;
- Better correlation – улучшение возможностей по корреляции событий;
- Better enforcement across the infrastructure – решения, которые поддерживают технологию VMSafe, могут быть легко введены в эксплуатацию, так как такие решения обычно поставляются в виде виртуальных устройств (virtual appliance) и позволяют потратить меньше времени на их внедрение;
- Better scalability – улучшение возможностей масштабируемости инфраструктуры, сохраняя все правила и политики.

Работа с VMsafe может осуществляться в 2 режимах:

1. Kernel-mode – обработка в гипервизоре.
2. VM-mode – обработка внутри ВМ.

Для соблюдения политики безопасности необходимо настраивать политики виртуальных коммутаторов (vSwitch) на хостах таким образом, чтобы они были одинаковы – VLAN ID, имена Port Groups и т.п. Все это нужно поддерживать в актуальном состоянии. Технология Distributed vSwitch позволяет объединить несколько хостов VMware ESX Server одним виртуальным коммутатором, что дает множество новых возможностей.

- Появляется возможность централизованного назначения политик для такого коммутатора, что устранит ошибки в конфигурации и обеспечит задание параметров VLAN, групп портов и Security из единой точки интерфейса;
- Distributed vSwitch позволяет корректно работать механизму VMware Fault Tolerance, чтобы «теневая» копия ВМ имела идентичные сетевые политики на другом сервере, и в случае отказа основной ВМ мгновенно заменяла ее в действующем сетевом окружении. Таким образом, получаем не просто отказоустойчивый кластер, а безопасный отказоустойчивый кластер;
- Появляется возможность перезагружать коммутатор, не перезагружая хост ESX. Это необходимо для того, чтобы при изменении числа портов у коммутатора, не перезагружать хост ESX.

В контексте безопасности VMware продукт VMsafe Distributed vSwitch занимает отдельное место. На каждом сервере VMware ESX появляется еще одна ВМ «Security VM», подключенная к распределенному виртуальному коммутатору и выполняющая функции обеспечения сетевой безопасности для виртуальных машин сервера. Это может быть IDS/IPS-система, Firewall и т.п. – а может и все вместе. При этом защищающая ВМ назначает политики защищаемой ВМ. При миграции работающей ВМ за счет технологии VMware VMotion эти политики «переезжают» на другой хост-сервер ESX.

Данная модель говорит о том, что в виртуальной инфраструктуре VMware отпадает необходимость в установке агентов ПО для безопасности внутри ВМ, а управление политиками безопасности данных систем становится гораздо проще [4].

Заключение

Проведенный анализ угроз виртуализации и уязвимостей гипервизоров позволяет сделать вывод о том, что правильно развернутая инфраструктура, использующая виртуализацию, может обеспечить дополнительный уровень безопасности машинам в сети. К операционным системам и приложениям, запускаемым на ВМ, должны применяться те же меры безопасности, что и на индивидуальных физических машинах. Безопасность ВМ – это вложение, которое нужно делать. Виртуализация должна быть всего лишь одним из многих инструментов в арсенале надежности.

Литература

1. Грам В. Технологии виртуализации и повышение эффективности функционирования корпоративных предложений // Копьютерра [Электронный ресурс]. – Режим доступа: <http://offline.computerra.ru/2009/06.html>, своб.
2. Черняк Л. Виртуализация серверов стандартной архитектуры // Открытые системы [Электронный ресурс]. – Режим доступа: <http://www.osp.ru/os/list/2009/04.html>, своб.
3. Keir T. Virtualization: The Executive Summary // [Электронный ресурс]. – Режим доступа: http://www.pcworld.com/businesscenter/article/215199/virtualization_the_executive_summary.html, своб.
4. Причины выбрать VMware // [Электронный ресурс]. – Режим доступа: <http://www.vmware.com/ru/virtualization/why-choose-vmware.html>, своб.

Евелев Юрий Евгеньевич – Санкт-Петербургский государственный инженерно-экономический университет (ИНЖЭКОН), студент, evelev@xaker.ru

Чернокижный Геннадий Михайлович – Санкт-Петербургский государственный инженерно-экономический университет (ИНЖЭКОН), кандидат технических наук, доцент, gmcher@pochta.ru

УДК 617.3

**ПРИМЕНЕНИЕ МЕТОДА ВЧ-БЛИЖНЕПОЛЬНОГО ЗОНДИРОВАНИЯ
В ДИАГНОСТИКЕ БИООБЪЕКТОВ****А.В. Арсеньев, А.Н. Волченко, Л.В. Лихачева, В.И. Печерский**

Рассмотрена возможность определения электродинамических параметров тканей биологических сред с использованием неконтактного метода. Показаны принципиальная возможность использования методики для неинвазивной диагностики биообъектов, а также результаты определения функциональной активности ростковых зон (РЗ) костей подростков в качестве иллюстрации возможностей методики. Целью настоящей работы является исследование возможностей метода высокочастотного (ВЧ) ближнепольного зондирования в диагностике биообъектов и его апробация на конкретных обследованиях опорно-двигательного аппарата подростков.

Ключевые слова: ВЧ-ближнепольное зондирование, ростковые зоны, диэлектрическая проницаемость.

Введение

Возникновение и развитие многих заболеваний в ортопедотравматологической практике связано с нарушением процесса роста костей. Рост может ускоряться, замедляться или извращаться. Это может носить как системный, так и локальный характер. Примером таких заболеваний является идиопатический сколиоз, разная длина конечностей, ряд наследственных заболеваний скелета.

Рост костей скелета представляет собой сложный процесс. Он зависит от многих факторов, таких как наследственные заболевания, состояние здоровья ребенка (наличие или отсутствие хронических заболеваний), питание, физическая активность и многое другое.

В свою очередь, все вышеперечисленное зависит от функционирования специальных структур в костях ребенка – так называемых РЗ. Они обеспечивают продольный рост каждого сегмента, а, следовательно, и всего скелета в целом.

На сегодняшний день оценивать функциональную активность РЗ костей ребенка в клинических условиях можно, лишь прибегнув к радиоизотопному исследованию [1], которое имеет ряд недостатков: неизбежно приводит к определенной лучевой нагрузке на организм, требует специального оборудования, на обследование одного пациента затрачивается несколько часов времени, процедура обследования связана с внутривенным введением фармакологического препарата, на который возможна аллергическая реакция. По этим причинам радиоизотопное исследование не находит широкого применения в детской ортопедии для оценки функционирования активности РЗ. Других методов оценки функциональной активности РЗ, альтернативных радиоизотопному методу, в современной медицине не существует.

Однако с появлением в клинической практике эффективных неоперативных способов влияния на РЗ при лечении заболеваний в детской ортопедии и травматологии, в первую очередь – магнитотерапии, необходимость разработки достаточно простых и информативных методов контроля РЗ возросла многократно.

Целью настоящей работы является изучение возможностей метода ВЧ-ближнепольного зондирования в диагностике состояния биообъектов (конкретно – оценка активности РЗ) и его апробация в клинических условиях на исследовании функциональной активности РЗ опорно-двигательного аппарата у подростков.

Материалы и методы

Основными предпосылками к разработке излагаемого здесь метода контроля РЗ послужили работы по электрической системе регуляции процессов жизнедеятельности сложных многоклеточных организмов, полученные в биофизике и медицине [2, 3]:

- определяющая роль электрической системы, регулирующей процессы жизнедеятельности любого организма;
- патологические и физиологические процессы в исследуемых тканях, как правило, сопровождаются изменением их электродинамических параметров, прежде всего диэлектрической проницаемости и проводимости.

Такой же эффект в исследуемых тканях возникает при воздействии на них различных физических полей (электрического, магнитного, электромагнитного, теплого), которые используются в лечебных методиках, например, физиотерапии. Первичным звеном в цепи событий, запускаемых воздействием слабого комбинированного магнитного поля на биосистему, является ион Ca^{2+} , специфически связанный с Ca^{2+} -связывающим центром белка. Этот белок обладает Ca^{2+} -зависимой ферментативной активностью или, альтернативно, способен модулировать активность других ферментов [4]. Другими словами, если говорить о РЗ до воздействия слабых магнитных полей (СМП), область РЗ, имеющая диэлектрическую

проницаемость ε_1 , после воздействия на РЗ характеризуется уже ε_2 . Итак, прямое или косвенное измерение величины ε_2 или величины $\Delta\varepsilon = \varepsilon_2 - \varepsilon_1$ может служить оценкой изменения электрической активности РЗ под влиянием СМП и, соответственно, качественной оценкой пролиферативной активности тканей РЗ [4]. Таким образом, физической составляющей предлагаемого метода оценки функциональной активности РЗ является измерение (оценка) величины диэлектрической проницаемости ε живой ткани или ее $\Delta\varepsilon$.

В технике физического эксперимента хорошо известны методы оценки измерений диэлектрической проницаемости веществ в различных агрегатных состояниях в постоянном и переменном электрических полях [5]. Некоторые из этих методов были использованы и в медицине в методе реографии, в котором электрические свойства биологических тканей определялись по величине и изменению их импеданса (комплексного сопротивления). Все эти методы позволяли оценивать диэлектрическую проницаемость ε через абсолютные или относительные измерения емкости (емкостного сопротивления), в которые включалось вещество с исследуемой ε . Для биологических объектов было установлено, что импеданс живых клеток включает в себя только активное (омическое) и емкостное сопротивления [2, 3]. На основании этого вывода были построены схемы эпидермиса (многослойная композиция), информационных каналов передачи электрических сигналов от различных органов биообъекта на эпидермис, которые включали в себя различные композиции активного и емкостного сопротивлений (RC-цепи). Эти результаты, с учетом условий исследования РЗ (исследования активности РЗ должны быть неинвазивными и безопасными), определяли метод реализации поставленной задачи. Таким методом был выбран метод резонансного ВЧ-ближнепольного зондирования.

Суть этого метода можно пояснить следующим образом. Датчик (рис. 1, 2), представляющий собой малую электромагнитную антенну (открытый конденсатор), включается в качестве нагрузки в резонансную систему (колебательный контур генератора частоты). При контакте датчика с исследуемой областью тела пациента (область РЗ или близлежащие ткани) импеданс этой области оказывает влияние на импеданс антенны через поле самой антенны. Это влияние приводит к изменению (увеличению или уменьшению) резонансной частоты генератора ω_p . По этому изменению частоты можно судить об электромагнитных параметрах исследуемой области вблизи антенны. Используя известную формулу Томпсона для собственной частоты контура генератора $\omega_0 = \frac{1}{\sqrt{LC}}$, можно говорить, что если $\omega_p \leq \omega_0$, то эквивалентная емкость в связи импедансом увеличилась, и в этой точке исследования возросла электрическая активность тканей за счет увеличения ее диэлектрической проницаемости ε . И, наоборот, если $\omega_p \geq \omega_0$, то активность уменьшается за счет уменьшения ε .



Рис. 1. Структура диагностического комплекса

Таким образом, окончательно в методе ВЧ-ближнепольного зондирования информационный сигнал, регистрирующий состояние функциональной активности РЗ или другой биологической ткани, формируется как величина $\Delta\omega = \omega_p - \omega_0$:

- если $\Delta\omega \leq 0$ – функциональная активность возрастает;
- если $\Delta\omega \geq 0$ – функциональная активность уменьшается.

В связи с тем, что РЗ представляет собой зону повышенной обменной активности, и электрические характеристики этой зоны будут отличаться от характеристик соседних зон. Следовательно, с учетом вышесказанного, логично предположить, что сигнал, получаемый от этих областей с помощью прибора ВЧ-ближнепольного зондирования, будет отличаться от сигнала соседних областей.

При этом метод абсолютно безопасен, так как при обследовании на организм ребенка не оказываются никаких негативных воздействий физического плана. Пациент чувствует только легкое прикосновение датчика, который «считывает» уровень сигнала от определенной зоны тела. Продолжительность диагностики занимает несколько минут. Датчик устанавливается над областью РЗ костей конечностей, что требует от специалиста лишь хороших знаний анатомии. Основным элементом – датчик диагностического комплекса, генератор основной частоты ($\omega_0 \approx 3,4 \times 10^6$), преобразователь частота–код с USB интерфейсом и индикатор патологий с PC или Pocket PC (рис. 2).



Рис. 2. Диагностический комплекс

Результаты исследований

На сегодняшний день совместно СПб ГУЗ ВЦДОиТ «Огонек» и ортопедическим санаторием ООО «Родник» (г. Пермь) были произведены независимые обследования с использованием предлагаемого метода и аппаратуры для определения уровня активности РЗ пациентов различных возрастных групп. Всего было обследовано 200 человек в Санкт-Петербурге и 150 человек в Перми (таблица) в возрасте от 4 до 17 лет. Пациенты (мальчики и девочки) выбирались по клиническим наблюдениям как группа условно здоровых, т.е. отбирались пациенты, не имеющие явно выраженных патологий в опорно-двигательном аппарате. Результаты этих исследований приведены ниже на рисунках. На рис. 3, 4 по оси абсцисс приведен возраст групп пациентов. Что касается оси ординат, то по этой оси откладывается относительное изменение частоты генератора датчика в %, где за 100% принимается сигнал, полученный от воздуха перед началом измерений. Целью этих исследований было определить так называемую «норму» сигнала, получаемого от РЗ в разных возрастных группах детей.

Дети	Санкт-Петербург	Пермь
мальчики	120 человек	90 человек
девочки	80 человек	60 человек
всего детей	200 человек	150 человек

Таблица. Общее количество обследованных подростков

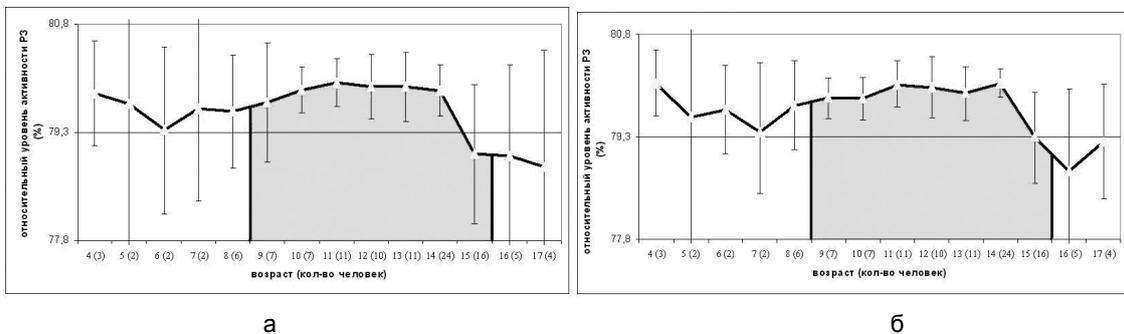


Рис. 3. Относительный уровень активности РЗ у мальчиков различных возрастных групп: правая нога (а); левая нога (б)

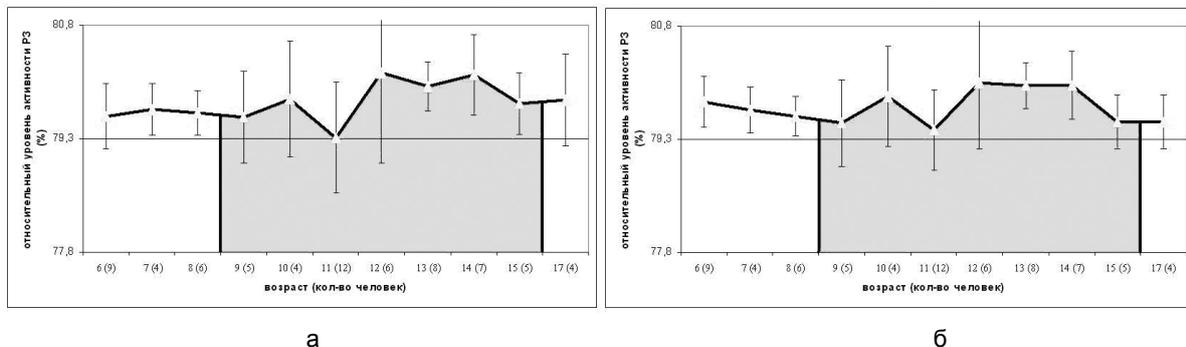


Рис. 4. Относительный уровень активности РЗ у девочек различных возрастных групп: правая нога (а), левая нога (б)

Эти исследования показали, что предложенный метод диагностики активности РЗ регистрирует показатели, коррелирующие с общими медицинскими представлениями о физиологической активности РЗ у подростков, а именно: наибольшая активность РЗ (рост и формирование скелета) у мальчиков и у девочек наблюдается в возрастном интервале 9–15 лет [6, 7].

Обсуждение результатов

Можно отметить следующее:

- у мальчиков область активности в интервале 9–15 лет выражена более явно (рис. 3);
- у девочек (рис. 4) эти отклонения в уровне активности РЗ по сравнению с мальчиками в интервале 11–15 лет могут быть объяснены перестройкой их организма.

Для большей очевидности области «нормы» для активности РЗ необходимо увеличить число наблюдений и уравнивать их по количеству пациентов в каждой возрастной группе. Из графиков на рисунках видно, что разброс измерений относительно среднего уровня определяется количеством пациентов в данной возрастной группе. Если количество пациентов невелико (2–4 человека), то ошибка измерений увеличивается (рис. 3). С увеличением количества пациентов (до 10 и более) наблюдается уменьшение разброса измерений. Таким образом, на основании вышеприведенных результатов (рис. 3, 4) можно говорить о том, что основная цель работы достигнута, а именно: метод ВЧ-ближнеполюсного зондирования дает достаточно простую и оперативную информацию о функциональном состоянии РЗ, совпадающую с физиологическими особенностями активности РЗ в зависимости от возраста и пола (возрастная физиология).

Заключение

Методика обследования достаточно проста, неинвазивна и безопасна. Аппаратно-диагностический комплекс портативен и мобилен.

В дальнейшем предполагается продолжить исследования с целью получения статистически достоверных показателей понятия «нормы» для активности РЗ. В свою очередь, выявленные показатели возрастов нормы активности РЗ могут быть использованы для совершенствования и разработки различных медицинских методик лечения заболеваний опорно-двигательного аппарата у детей и подростков.

Литература

1. Радионуклидная диагностика для практических врачей / Под ред. Ю.Б. Лишманова, В.И. Чернова. – Томск: STT, 2004. – 394 с.
2. Электрическая система регуляции процессов жизнедеятельности / Под ред. Г.Н. Зацепиной. – М.: Изд-во МГУ, 1992. – 160 с.
3. Самойлов В.О. Медицинская биофизика: Учебник для вузов. – СПб: СпецЛит, 2007. – 560 с.
4. Леднев В.В. Биоэффекты слабых комбинированных, постоянных и переменных магнитных полей // Биофизика, 1996. – Т. 41. – Вып. 1. – С. 224–231.
5. Браун В. Диэлектрики. – М.: Издательство иностранной литературы, 1961. – 327 с.
6. Вельтишев Ю.Е. Рост ребенка: закономерности, отклонения, патология и превентивная терапия: лекция №12 / Московский НИИ протезирования и детской хирургии МЗ РФ. – М., 1994. – 74 с.
7. Медведев В.П., Куликов А.М. Подростковая медицина. – СПб: СпецЛит, 1999. – 731 с.

Арсеньев Алексей Валентинович – СПб ГУЗ ВЦДОИТ «Огонёк», кандидат медицинских наук, зав. отделением, stivamat@ Rambler.ru

Волченко Александр Николаевич – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, wolf2684@mail.ru

Лихачева Людмила Валентиновна – ООО «Санаторий-профилакторий «Родник», гл. врач, rodnikperm@mail.ru

Печерский Виктор Иванович – ООО «Санаторий-профилакторий «Родник», врач-ортопед, rodnikperm@mail.ru

УДК 616-089.2(043.2)

НЕИНВАЗИВНЫЕ МЕТОДЫ ИЗМЕРЕНИЯ БИЛИРУБИНА, ГЕМОГЛОБИНА И ГЛЮКОЗЫ. ПРИБОР ГЕМОБИЛИГЛЮКОМЕТР

Е.В. Кожохина

Рассмотрены преимущества неинвазивных измерений параметров крови, предложено конструктивное исполнение прибора гемобилиглюкометра.

Ключевые слова: неинвазивный, гемоглобин, билирубин, глюкоза, поглощение, интерференционные фильтры, микроконтроллер, мониторинг.

Введение

Свет видимого диапазона длин волн является безопасным при малых уровнях плотности излучения. Свет этого диапазона длин волн хорошо проникает в биологическую ткань и несет информацию о структурных и динамических изменениях в тканях. С помощью такого физического явления как поглощение, можно диагностировать некоторые параметры крови, такие как гемоглобин, билирубин и глюкоза.

Лабораторные исследования, построенные на инвазивных методах, сопряжены с травмированием пациентов, возможностью инфицирования, а также с довольно длительной процедурой получения диагностического результата. Однако на сегодняшний день они являются определяющими, так как удовлетворяют требованиям допустимых пределов аналитической вариации: для гемоглобина $\pm 2\%$; для билирубина $\pm 10\%$ [1]. Тем не менее, для мониторинга состояния больных и терапии неинвазивные методы значительно превосходят лабораторные исследования как по оперативности, так и по эффективности и экономичности.

Приоритет безкожного определения билирубина оптико-электронным методом принадлежит фирме «Minolta» [2]. Усовершенствованием конструкции частично, при том же методе обработки информации, был АБЧ-01, ЦКБ завода «Арсенал» [3], г. Киев, а также «Билитест» фирмы «Техномед» г. Москва. Дальнейшим усовершенствованием является ИН-01Б, разработанный МП НПП «Оптамед-Интекон» г. Киев [4], измеряющий концентрацию билирубина в диапазоне 0–300 мкмоль/л.

Медицинское апробирование упомянутых приборов в ряде ведущих медицинских НИИ Украины и России выявило большую заинтересованность, так как приборы позволяют:

- в течение нескольких секунд бескровным методом поставить диагноз гипербилирубинемии, анемии или диабета и выявить методом скрининга пациентов группы риска;
- исключить инфицирование пациента во время взятия крови (СПИД, гепатит, кокковая инфекция и др.);
- проводить измерения любому медицинскому работнику;
- быстро окупаться, так как исключаются затраты на реагенты и т.д.

Постановка задачи

Из приведенного выше анализа следует, что неинвазивные оптико-электронные приборы весьма перспективны для клинической диагностики, однако по аналитической вариации они уступают традиционным биохимическим методам. К тому же, не существует комбинированных неинвазивных измерителей параметров крови. В данной работе рассмотрены свойства гемоглобина, билирубина и глюкозы, а также их спектры поглощения, особенности прохождения излучения через слои кожи, требования к созданию гемобилиглюкометра.

Взаимодействие излучения с кожным покровом

При биохимических измерениях концентрация исследуемого вещества определяется непосредственно в крови либо в сыворотке. В неинвазивных измерениях процесс измерения ведется в среде, которая отличается от крови по целому ряду параметров – коже. Кожа – это наружный покров тела человека, который выполняет разнообразные функции, защищая организм от разрушительных факторов. Кожа представляет собой трехкомпонентную тканевую систему, образованную эпидермисом, дермой и подкожной жировой клетчаткой [5].

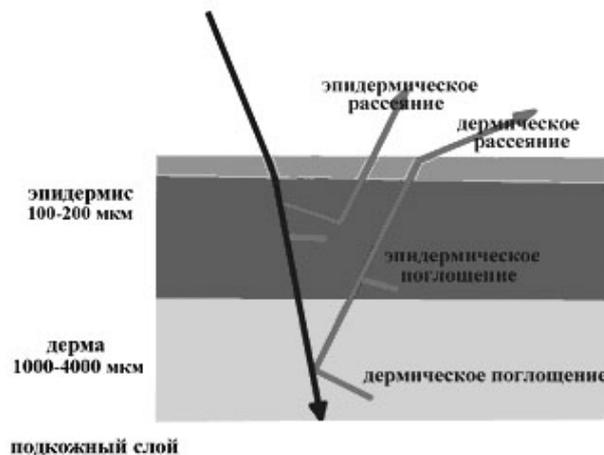


Рис. 1. Взаимодействие излучения видимого диапазона длин волн с кожными тканями

При взаимодействии излучения видимого диапазона длин волн с биологическими тканями имеют место обычные оптические эффекты, возникающие при прохождении света через неоднородную среду. Часть падающего на биоткань излучения отражается от ее поверхности. Проникающее в ткань излучение видимого диапазона длин волн подвергается многократному рассеиванию, поглощению различными биологическими структурами и частичному преобразованию во вторичное излучение [6]. На рис. 1 представлена схема взаимодействия лазерного луча с кожным покровом.

Величина поглощенной биотканями энергии ($\Phi_{\text{пол}}$) приблизительно исчисляется как разность между падающей ($\Phi_{\text{пад}}$) и отраженной ($\Phi_{\text{отр}}$) частями энергии светового потока [7]:

$$\Phi_{\text{погл}} = \Phi_{\text{пад}} - \Phi_{\text{отр}} = \Phi_{\text{пад}} \cdot [1 - \rho(\lambda)].$$

Кровеносные сосуды залегают на определенных глубинах в коже, для регистрации рассеянного и отраженного излучения необходима определенная интенсивность излучения для различных длин волн. При этом в разных участках тела количества кровеносных сосудов на единицу объема различные. В зависимости от длины волны разный объем излучения проникает сквозь различные слои кожи.

На рис. 2 приведена относительная спектральная чувствительность фотопреобразователя ФП33-1Л, используемого в гемобилиглокометре, и глубина проникновения излучения 0,4–1 мкм через слои кожи. Поскольку биологические ткани являются интенсивно рассеивающими средами, их толщина и структура сильно влияют на поглощение лазерного излучения, в диапазоне длин от 600 до 1400 нм кожа поглощает 25–40% излучения [8].

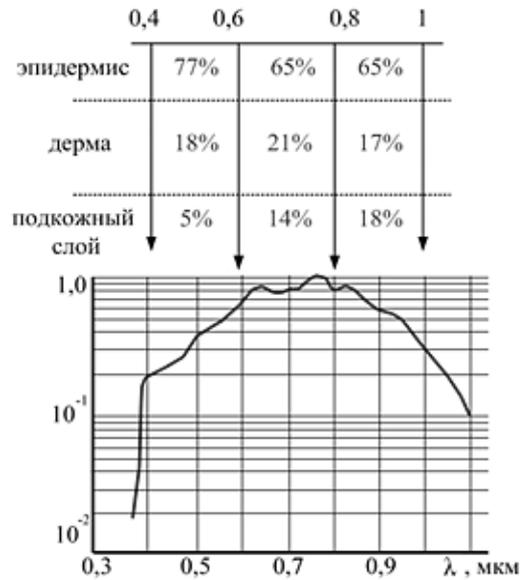


Рис. 2. Глубина проникновения излучения через кожную ткань и относительная спектральная чувствительность фотопреобразователя ФП33-1Л

Гемоглобин, билирубин и глюкоза

Гемоглобин (др.-греч. αἷμα – кровь и лат. globus – шар) – сложный железосодержащий белок животных и человека, способный обратимо связываться с кислородом, обеспечивая его перенос в ткани [9].

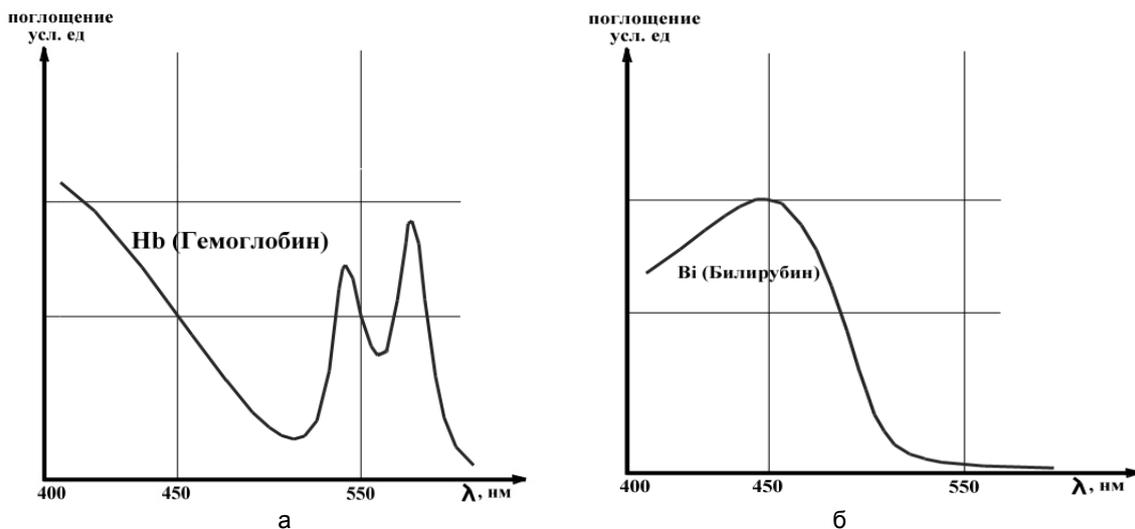


Рис. 3. Спектры поглощения: гемоглобина (а); билирубина (б)

При неинвазивном измерении гемоглобина проблемой для получения точных данных может стать эмоциональное состояние пациента. При волнении и возбуждении концентрация крови на единицу по-

верхности кожи может увеличиваться, человек краснеет, а это вызывает погрешность при определении уровня гемоглобина неинвазивно. На рис. 3, а, представлен спектр поглощения общего гемоглобина.

Билирубин (лат. bilis – желчь и tiber – красный) – один из желчных пигментов. Билирубин является одним из промежуточных продуктов распада гемоглобина, происходящего в макрофагах селезенки, печени и костном мозге (примерно 80%).

Пигментация от билирубина проявляется в верхних слоях кожи. Для неинвазивного измерения билирубина пациенту требуется обескровить участок кожной ткани, приложив давление порядка 10^5 Па, чтобы исключить погрешность, вызываемую наличием гемоглобина. Максимум поглощения билирубина – 460 нм (рис. 3, б).

Глюкоза (от греч. glykys – сладкий) (виноградный сахар) – углевод из группы моносахаридов. Глюкоза составляет основу субстрата, необходимого для энергообеспечения и жизнедеятельности клетки. Глюкоза поступает в кровь при всасывании в кишечнике после расщепления углеводов, содержащихся в продуктах питания.

Спектры поглощения глюкозы: $\lambda_1=0,83$ мкм, $\lambda_2=0,94$ мкм, $\lambda_3=1,03$ мкм (рис. 4).

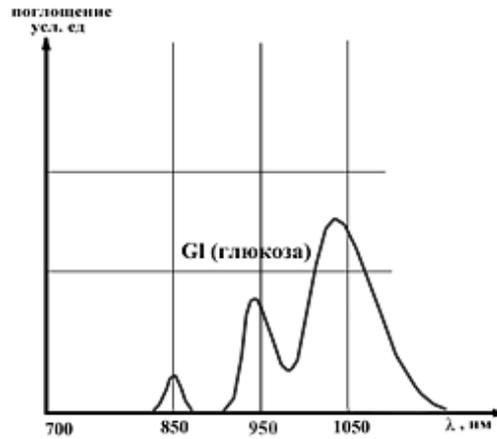


Рис. 4. Спектр поглощения глюкозы

Соединим спектры (рис. 3, 4) на одном рисунке, чтобы наглядно видеть оптимальные длины волн поглощения для каждого из рассматриваемых веществ, которые будут соответствовать светофильтрам 4, 5, 6 (рис. 6), а светофильтр 7 будет соответствовать опорному сигналу в 0,78 мкм.

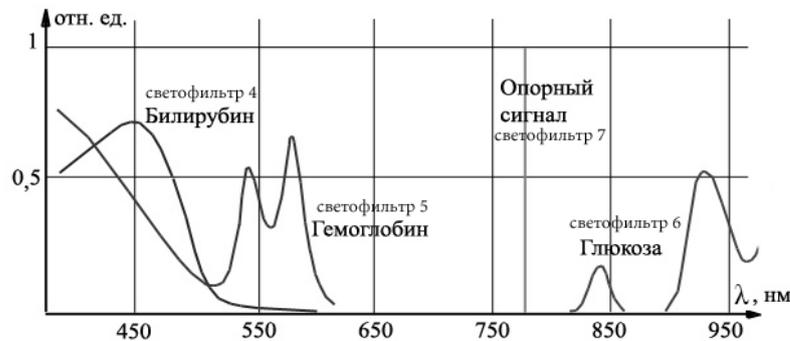


Рис. 5. Спектры поглощения билирубина (4), гемоглобина (5), глюкозы (6), а также опорный сигнал (7)

Разработка неинвазивного гемоглюкометра

Во всех неинвазивных оптико-электронных приборах используется источник широкополосного импульсного излучения с дальнейшим анализом поглощения света на определенной длине волны за счет узкополосных интерференционных фильтров и отдельных фотоприемников. Недостатком данного метода является разброс чувствительности фотоприемников, а потому необходимо перед каждым измерением проводить калибровку прибора.

Предложен комбинированный прибор, что позволяет одновременно измерять три параметра: билирубин ($\lambda = 0,46$ мкм), гемоглобин ($\lambda = 0,55$ мкм) и глюкоза ($\lambda = 0,94$ мкм), по отношению к коже $\lambda = 0,7$ мкм. С помощью одного пространственного линейного фоточувствительного прибора с зарядовой связью (ФПЗС) на рис. 6 приведена структурная схема прибора, в состав которого входят оптико-электронный блок (11), блок обработки информации (12), блок индикации на ЖК-индикаторе и блок пи-

тания со стабилизатором от пяти аккумуляторов. Оптико-электронный блок состоит из волоконного факона (3), интерференционных фильтров (4–7), линейного ФПЗС (9), светофильтра (8), который обрезает ультрафиолетовые и инфракрасные излучения, лампы-вспышка с отражателем (10). Внешний вид прибора показан на рис. 7.

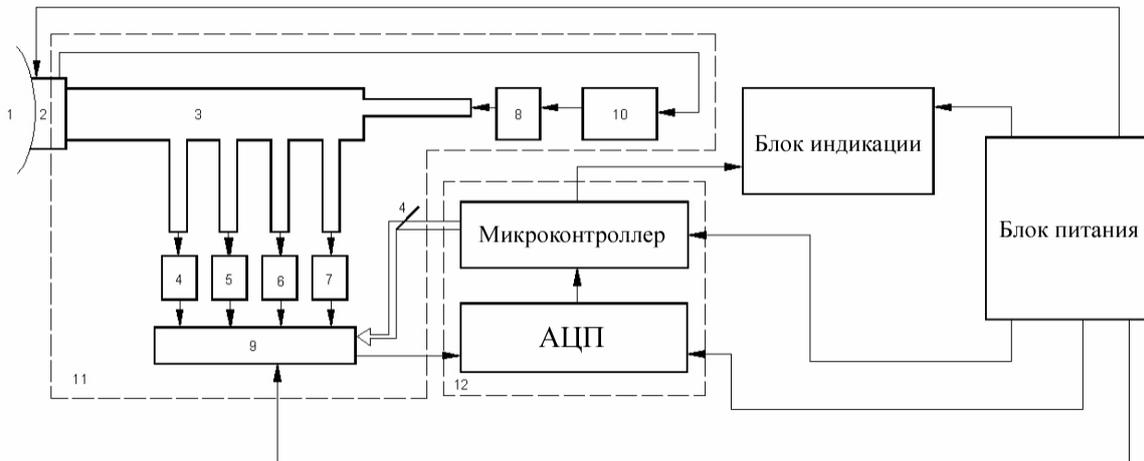


Рис. 6. Структурная схема гемобилиглюкометра: 1 – исследуемый участок кожи; 2 – подвижное приспособление; 3 – оптоволоконно; 4–8 – светофильтры; 9 – приемник излучения; 10 – источник излучения; 11 – оптический блок; 12 – полупроводниковый блок

Блок обработки информации состоит из микроконтроллера и аналогово-цифрового преобразователя. Микроконтроллер предназначен для управления линейным ПЗС, а также для проведения необходимых вычислений по результатам измерений.

Процесс измерения заключается в следующем: прибор приставляется волоконным датчиком к коже пациента, срабатывает лампа-вспышка, и широкополосное излучение через передаточный волоконный канал направляется в кожу пациента. Рассеянный и отраженный оптические сигналы через приемные каналы волоконного факона и интерференционные фильтры направляются на линейки ПЗС. Крепления волоконного факона, фильтров и линейного ПЗС осуществляется методом оптического склеивания.

Информация из выхода микроконтроллера поступает на ЖК-индикатор, где высвечивается значение измеряемых параметров в абсолютных величинах: билирубин 0–400 мкмоль/л, гемоглобин 0–300 г/л, глюкоза 0–40 моль/л.

В упомянутых ранее приборах в каждом приемном канале используется узкополосный интерференционный светофильтр с полосой пропускания ± 30 нм и своим фотоприемником. Так как чувствительность фотоприемников различна, то в этих приборах необходимо выравнивать чувствительность приемных каналов.

Авторами предложено в качестве фотоприемника использовать один пространственный линейный фотоприемник на приборах с зарядовой связью, к которому путем оптической склейки через интерференционные светофильтры подключены волоконные приемные каналы. В качестве источника света используется широкополосный источник света – лампа-вспышка с оптическим отражателем. Между факоном и лампой-вспышкой устанавливаются обрезачные светофильтры для подавления ближнего ультрафиолетового и инфракрасного излучения (эффект люминисценции кожи).

Как описано выше, кровеносные сосуды залегают на определенных глубинах в коже, поэтому для регистрации рассеянного и отраженного излучений необходима определенная интенсивность излучения для различных длин волн. Кроме того, пигментация от билирубина проявляется в верхних слоях кожи, а гемоглобин и глюкозу необходимо измерять там, где кровеносных сосудов максимум. На качество измерения, особенно при измерении билирубина, очень влияет цвет кожи (тип расы, загар). Таким образом, в процессе измерений надо вносить поправки в виде констант. Поэтому в блоке обработки информации используется микроконтроллер типа AVR с памятью типа EEPROM, куда можно заносить эти константы. Неинвазивные измерения дают хорошие результаты при измерениях билирубина и гемоглобина на новорожденных детях, т.е. в неонатологии. Если производить измерения на пациентах других возрастных групп, то необходимо учитывать сухость кожи, сальную прослойку и целый ряд других параметров, которые весьма трудно учесть.

В связи с этим авторами была предложена следующая корреляционная методика для старших возрастных групп: разработаны специальные отражающие микрокюветы, которые показали хорошие результаты при измерении билирубина. Иначе говоря, в начале прибором по сыворотке крови в микрокювете определяется концентрация билирубина, а после этого проводятся у того же пациента транскутанные измере-

ния. Устанавливается коэффициент корреляции для данного пациента и вводится поправка в микроконтроллер в виде константы в память EEPROM. Далее измерения у данного пациента проводят неизвазивно.

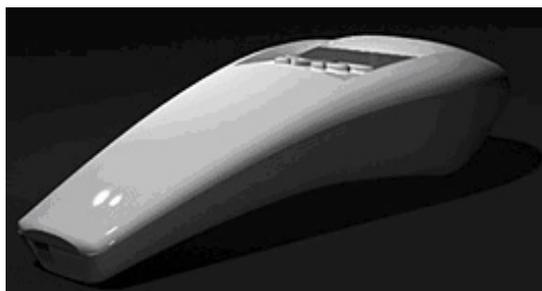


Рис. 7. Внешний вид гемобилиглюкометра

Заключение

Сформулируем основные рекомендации по применению прибора.

- При измерении билирубина необходимо устранить влияние гемоглобина за счет давления (обескровливать участок кожи).
- При измерениях гемоглобина и глюкозы необходимо увеличивать интенсивность излучателя, избегая механического давления на сосуды.
- При применении прибора для различных возрастных групп для получения достоверной информации необходимо для каждого пациента определять коэффициент корреляции «кровь-кожа» с последующим неизвазивным измерением. Коэффициент корреляции вводится в виде констант в вычислитель.

Литература

1. Лабораторные методы исследования в клинике / Справочник. – Под ред. В.В. Меньшикова. – М.: Медицина, 1987.
2. Патент США № 4267844, кл.А61В 5/00.1981.
3. Краснов В.Н., Яринич В.М., Макогон В.И. Анализатор билирубина черезкожный. Авт. свид. СССР, кл.А61В 5/00.1989.
4. Краснов В.Н. Измеритель билирубина неизвазивный ИБН-01 // Педиатрия, акушерство и гинекология (ПАГ). – Київ, 1993. – № 1. – 60 с.
5. Нормальная физиология: Учебник для студентов университетов / А.В. Коробков, А.А. Башкиров, К.Т. Ветчинкина / Под ред. А.В. Коробкова. – М.: Высшая школа, 1980. – 560 с.
6. Прикладная лазерная медицина: Учебное и справочное пособие / Под ред. Х.-П. Берлиена, Г.Й. Мюллера: Пер. с нем. – М.: Интерэксперт, 1997. – 356 с.
7. Козлов В.И. Взаимодействие лазерного излучения с биотканями // Сборник трудов «Применение низкоинтенсивных лазеров в клинической практике» / Под ред. О.К. Скобелкина. – М.: ГНЦ лазерной медицины, 1997. – С. 24–34.
8. Киселев Г.Л. Моделирование распространения света в биологических тканях // Биомедицинская радиоэлектроника. – 2001. – № 1. – С. 10–17.
9. Назаренко Г.И., Кишкун А.А. Клиническая оценка результатов лабораторных исследований. – М., 2005.

Кожохина Елена Владимировна – Национальный авиационный университет Украины, ассистент, miosend@mail.ru

УДК 004.021

ПРИМЕНЕНИЕ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ДЛЯ ОЦЕНКИ КАЧЕСТВА БИЗНЕС-ПЛАНОВ ИННОВАЦИОННЫХ ПРОЕКТОВ

А.Г. Кравец, А.С. Дроботов

В работе исследована возможность применения метода имитационного моделирования (метод Монте-Карло) для оценки качества инновационных проектов; предложен способ использования имитационного моделирования для принятия решения; программно реализован метод оценки рисков проекта на основе бизнес-плана.

Ключевые слова: имитационное моделирование, метод Монте-Карло, риск инновационного проекта, бизнес-план инновационного проекта.

Введение

Необходимым условием стабильного функционирования и развития экономики является эффективная инвестиционная политика. Но при разработке и анализе эффективности тех или иных инновационных проектов, часто приходится сталкиваться с тем, что рассматриваемые при их оценке потоки денежных средств относятся к будущим периодам и носят прогнозный характер. Неопределенность прогнозируемых результатов приводит к возникновению риска того, что цели, поставленные в проекте, могут быть не достигнуты полностью или частично.

Таким образом, целью данной работы является повышение эффективности процесса разработки бизнес-планов инновационных проектов за счет количественной оценки рисков на основе метода имитационного моделирования. Научная новизна работы заключается в разработке нового алгоритма оценки общего риска инвестиционного проекта и алгоритма идентификации рисков на основе метода имитационного моделирования Монте-Карло.

Метод оценки качества инновационных проектов на основе имитационного моделирования

Исходными данными для оценки качества инновационных проектов в работе служит макет бизнес-плана, отвечающий требованиям постановления Правительства РФ [1]. Структура бизнес-плана должна быть ориентирована на производство и продажу товаров (услуг).

Оценка качества инновационного проекта на основании предложенной структуры бизнес-плана основывается на модели денежных потоков и математической модели, определяемой формулой (1). Для оценки качества проектов была выбрана модель денежных потоков, включающих в себя потоки по финансовой, производственной и инвестиционной деятельности, представленные в дискретном виде [2]:

$$BP = \{AC_i, IA_i, FA_i\}, \quad (1)$$

где AC_i – поток денежных средств по операционной деятельности в i -ый квартал; IA_i – поток денежных средств по инвестиционной деятельности в i -ый квартал; FA_i – поток денежных средств по финансовой деятельности в i -ый квартал.

В качестве исходных величин (риск-переменные) для имитационного моделирования выступают [2] объем выпуска Q_i , цена P_i , переменные затраты V_i . В качестве закона изменения исходных величин используется треугольный закон распределения. В качестве интегрального показателя для поддержки принятия решения был выбран показатель – чистая современная стоимость проекта [3]:

$$NPV = \sum_{i=1}^N \frac{[Q_i \cdot (P_i - V_i) - F_i - A_i] \cdot (1 - T_i) + A_i}{(1 + r)^i} - I, \quad (2)$$

где I – начальные инвестиции; r – норма дисконта; N – срок проекта; Q_i – объем выпуска продукции (услуги); P_i – цена за штуку продукции (услуги); V_i – переменные затраты на выпуск продукции (услуги); F_i – постоянные затраты; A_i – амортизация; T_i – налог на прибыль. Анализ проекта по результатам имитационного моделирования для оценки качества предполагает два этапа [4]:

1. оценка общей эффективности проекта (принятие решения о целесообразности вложения инвестиций в проект);
2. идентификация рисков – определение того, какие риски могут повлиять на проект (рассматриваются только производственные риски по риск-переменным).

Оценка качества инновационного проекта в целом

После проведения имитационного моделирования (генерации возможных сценариев развития проекта по риск-переменным) следующим шагом является оценка эффективности проекта в целом на основе данных эксперимента с помощью следующих индикаторов [3]:

- вероятности реализации неэффективного проекта (3);
- индекса ожидаемых потерь (4);
- срока окупаемости проекта (среднее значение индикатора);
- чистой приведенной стоимости проекта (среднее значение индикатора);
- индекса доходности инвестиций (среднее значение индикатора).

$$P(\text{porog}) = \frac{m}{n}, \tag{3}$$

$$ELR = \frac{\left| \sum_{i=1}^{n_2} x_i^- \cdot p_i \right|}{\left| \sum_{i=1}^{n_1} x_i^+ \cdot p_i + \sum_{i=1}^{n_2} x_i^- \cdot p_i \right|}, \tag{4}$$

где m – число экспериментов со значением критериального показателя (2) ниже порогового уровня, задаваемого лицом, оценивающим риск; n – общее число имитационных экспериментов; porog – пороговый уровень критериального показателя (2); n_1 – количество экспериментов, для которых x_i принимает неотрицательный результат; x_i^+ – неотрицательный результат (величина NPV) при i -ом эксперименте; n_2 – количество экспериментов, для которых x_i принимает отрицательный результат; x_i^- – отрицательный результат (величина NPV) при i -ом эксперименте; p_i – вероятность получения результата x_i .

На основании полученных числовых значений индикаторов проекта делается вывод об эффективности проекта – решение об инвестировании проекта.

Для учета значений индикаторов (3) и (4) в эффективности проекта была разработана шкала оценки риска инновационного проекта (табл. 1–3) на основании [5].

Значение вероятности реализации неэффективного проекта	Балл
0–9% / уверенность в отсутствии риска	1
10–20% / скорее мнение об отсутствии риска, чем о его наличии	3
21–30% / позиция относительно риска неопределена	5
31–50% / скорее уверенность в наличии риска, чем в его отсутствии	7
51–100% / уверенность в высоком риске	9

Таблица 1. Определение баллов на основании значения вероятности реализации неэффективного проекта

Значение индекса ожидаемых потерь	Балл
0–0,08 / уверенность в отсутствии риска	1
0,09–0,19 / скорее мнение об отсутствии риска, чем о его наличии	3
0,20–0,29 / позиция относительно риска неопределенна	5
0,30–0,45 / скорее уверенность в наличии риска, чем в его отсутствии	7
0,46–1 / уверенность в высоком риске	9

Таблица 2. Определение баллов на основании значения индекса ожидаемых потерь

Уровень риска	Сумма баллов
Нерискованный проект	2
Минимальный риск	4–6
Средний риск	7–10
Высокий риск	11–14
Полный риск	15–18

Таблица 3. Оценка общего риска инновационного проекта

Определение рисков инновационного проекта

После оценки эффективности проекта в целом следующим шагом является идентификация возможных рисков проекта на основе проведенного имитационного моделирования по каждой риск-переменной [4]. Для каждого риска проекта рассчитывается сила риска и определяется план возможных действий, способствующих повышению благоприятных возможностей и снижению угроз для достижения целей проекта.

Сила риска определяется тремя показателями:

1. вероятность возникновения риска (5);
2. собственно силой риска, т.е. наиболее значимым экономическим убытком (6);
3. вероятностью восстановления развития проекта после наступления данного риска (7).

Вероятность возникновения риска определяется по следующей формуле [5]:

$$p_risk = \frac{N_1}{N_2}, \quad (5)$$

где N_1 – количество удовлетворяющих риску сценариев (число экспериментов); N_2 – общее число сценариев моделирования.

Сила риска, или экономический ущерб от возникновения риска, определяется по следующей формуле [4]:

$$f_risk = \min(NPV_i), \quad i = 1 \dots N_1, \quad (6)$$

где N_1 – количество удовлетворяющих риску сценариев (число экспериментов); $\min(NPV_i)$ – наименьшее значение показателя эффективности (NPV) проекта из удовлетворяющих риску сценариев.

Вероятность восстановления развития проекта после наступления данного риска определяется по следующей формуле [4]:

$$r_risk = \frac{R_1}{R_2}, \quad (7)$$

где R_1 – количество исходов, удовлетворяющих риску сценария, для которых $NPV_i > NPV_{POROG}$, R_2 – количество исходов, удовлетворяющих риску сценария, для которых $NPV_i < NPV_{POROG}$, NPV_{POROG} – пороговый уровень для показателя эффективности.

Далее для каждого риска происходит определение возможного сценария выхода из риска – разработка количественных показателей для риск-переменной, показывающих, на сколько нужно увеличить (уменьшить) переменную в следующие года развития и основанные на подсчете среднего значения:

$$RP = \frac{\sum_{i=1}^{N_1} rp_value_i}{N_1},$$

где N_1 – количество удовлетворяющих риску сценариев (число экспериментов); rp_value_i – i -е значение риск-переменной.

Заключение

В результате проведения имитационного моделирования определяется эффективность проекта (принятие решения о целесообразности вложения инвестиций в проект), а также определяется, какие риски могут повлиять на проект (например, падение объемов продаж товара на втором году реализации, повышение цен на необходимые товары для производства продукции на четвертом году реализации проекта), и план реагирования на риски.

В рамках работы была разработана автоматизированная система «Поддержка инновационных проектов «Эксперт» на языке C# (среда Microsoft Visual Studio 2005, СУБД – SQL Server 2005), которая состоит из модуля работы с бизнес-планами проектов, модуля проведения имитационного моделирования и оценки общего риска проекта, модуля идентификации рисков проекта и модуля формирования отчетов.

Система может использоваться в процессе разработки бизнес-плана проекта для анализа рисков и выработки стратегии их снижения, а также для принятия решения по инвестированию проектов.

Литература

1. Постановление Правительства РФ от 22 ноября 1997 г. № 1470 «Об утверждении Порядка предоставления государственных гарантий на конкурсной основе за счет средств Бюджета развития Российской Федерации и Положения об оценке эффективности инвестиционных проектов при размещении на конкурсной основе централизованных инвестиционных ресурсов Бюджета развития Российской Федерации» – [Электронный ресурс]. – Режим доступа: <http://base.garant.ru/176300/>, своб.
2. Макет бизнес-плана, представляемого претендентом в составе заявки [Электронный ресурс]. – Режим доступа: <http://www.siora.ru/businessplan/maket/>, своб.
3. Лукасевич И.Я. Анализ финансовых операций. Методы, модели, техника вычислений: Учеб. пособие / И.Я. Лукасевич. – М.: Финансы, ЮНИТИ, 1998. – 400 с.
4. Ларичев О.И. Теория и методы принятия решений: Учебник. – 2-е изд. – М.: Логос, 2002. – 392 с.
5. A Guide to the Project Management Body of Knowledge / Project Management Institute. – 3-d ed. – Project Management Institute, 2004. – 388 p.

Кравец Алла Григорьевна – Волгоградский государственный технический университет, доктор технических наук, доцент, al1212@rkm.ru

Дроботов Александр Сергеевич – Волгоградский государственный технический университет, аспирант, drobotov@rambler.ru

РОЛЬ ТЕХНОПАРКОВ В РАЗВИТИИ РОССИИ КАК ИННОВАЦИОННОГО ОБЩЕСТВА

Е.Ю. Абрамова

Реализация потенциала российских предприятий в сфере разработки инновационного продукта – одна из основных задач модернизации российской экономики, которую должны помочь решить российские технопарки. Однако попытки перехода к инновационному развитию наталкиваются на существенные препятствия: необходимость дополнения производителей знаний и технологий другими необходимыми звеньями инновационного процесса, такими как привлечение и размещения заказов на НИОКР, формирование спроса, продвижение инновации, сопровождение внедрения инновационных решений и их поддержка у заказчика. Нерешенность этих проблем усугубляется падением инновационной активности ведущих секторов экономики.

Ключевые слова: технопарки, инновации, высокие технологии, бизнес-инкубаторы, трансфер знаний, инновационная активность, инвестиции.

Введение

Развитие и распространение информационно-коммуникационных технологий являются важным фактором мирового экономического роста. Их широкое применение имеет решающее значение для повышения производительности и оптимизации деятельности предприятий и организаций практически всех отраслей экономики, а также модернизации и повышения эффективности основных институтов государственного управления. Для достижения этих целей необходимо решить следующие задачи:

- устранить административные барьеры для развития российских предприятий и их выхода на мировой рынок;
- создать специализированные технопарки в сфере информационно-коммуникационных технологий;
- обеспечить совершенствование системы профессиональной подготовки специалистов в сфере информационных технологий в соответствии с современными международными стандартами;
- обеспечить поддержку выхода российских предприятий, выпускающих продукцию с высоким экспортным потенциалом, на мировой рынок.

Основная часть

Ограниченность числа инструментов влияния на инновационную активность предприятий делает необходимым концентрацию интеллектуального капитала и формирование точек роста инновационной экономики по всему циклу – от возникновения идеи до промышленной реализации. Такой организационной формой являются технопарки. Технопарки создаются для предоставления широкого спектра услуг по поддержке инновационного предпринимательства путем развития материально-технической, социально-культурной, информационной и финансовой баз его становления.

10 марта 2006 года распоряжением Правительства Российской Федерации была утверждена государственная программа «Создание в Российской Федерации технопарков в сфере высоких технологий» [1], согласно которой основными задачами создания технопарков в сфере высоких технологий являются:

- развитие и поддержка российских высокотехнологичных производств;
- повышение инвестиционной привлекательности высокотехнологичных отраслей экономики;
- создание условий для размещения международными компаниями своих производств на территории России;
- увеличение объемов экспорта высокотехнологичной продукции, развитие малых и средних предприятий в сфере высоких технологий.

Создание технопарков в сфере высоких технологий в Российской Федерации обеспечивает территориальную концентрацию финансовых и интеллектуальных ресурсов для ускорения развития высокотехнологичных отраслей экономики.

Технопарки рассматриваются в качестве зон формирования кластеров разработки и производства конкурентоспособной высокотехнологичной продукции. Создание благоприятных условий для инноваций, максимальное снижение транзакционных издержек за счет расположения всех участников на одной компактной площадке с высокой концентрацией кадров разного профиля является основной целью технопарка как инновационной инфраструктуры. Технопарки объединят предприятия высокотехнологичных отраслей экономики, в том числе отраслей нано-, био-, информационных и других технологий, научные организации, учебные заведения, обеспечивающие научный и кадровый потенциал таких предприятий, а также иные предприятия и организации, деятельность которых технологически связана с организациями указанных отраслей или направлена на их обслуживание [2].

Технопарки могут поддерживать компании на разных стадиях развития. На их территории должны осуществлять свою деятельность как бизнес-инкубаторы, поддерживающие проекты только на начальном этапе, деятельность которых направлена на реализацию венчурных проектов в сфере высоких технологий,

так и коммерческие предприятия, оказывающие юридические, финансовые, информационно-технологические, маркетинговые и другие услуги. Таким образом, технопарки будут предоставлять набор всех необходимых услуг предприятиям, размещенным на их территории в сфере высоких технологий, что позволит этим предприятиям получить значительную экономию расходов и сконцентрироваться на своей основной деятельности.

Государственная программа должна была завершиться в 2010 году. К этому моменту девять высокотехнологичных площадок на территории восьми регионов (Калужской, Кемеровской, Московской, Нижегородской, Новосибирской, Тюменской областях, в республике Мордовия и Татарстане) должны были функционировать, принося прибыль.

По прогнозу общая выручка в 2010 году должна была составить около 26 млрд. руб., на территории технопарков должны были функционировать 677 компаний и созданы 14,5 тыс. рабочих мест. Общий объем инвестиций должен был составить 14 млрд. руб. [3].

Однако к настоящему времени всего три технопарка введены в эксплуатацию, а многие и вовсе находятся на стадии строительства.

Это происходит из-за того, что все инициативы, едва начавшись, гаснут в бюрократической возне, это касается буквально всего – законодательного регулирования, разработки механизма налогообложения, формирования производственной инфраструктуры, привлечения инвестиций и т.д.

На стадии функционирования технопарки в России также сталкиваются со многими проблемами.

Сегодня российские промышленные предприятия отличаются крайне низкой инновационной активностью. В их затратах доминируют расходы на новое оборудование при низком спросе на исследования и разработки.

Для повышения экономической отдачи от резидентов технопарков и реализации инноваций необходимо использование инструментов прямого регулирования инновационной активности государственного сектора экономики, национальных компаний и естественных монополий, стимулирование разработки и реализации инфраструктурных проектов. Существенное влияние на инновационную активность может оказать политика и практика закупок товаров и услуг для государственных нужд. Возможности влияния госзаказа на сферу инноваций пока не находят адекватного отражения в научно-технической политике Российской Федерации. Сегодня требуется усиление прямого регулирования и экономического стимулирования процессов технологической модернизации сырьевого сектора национальной экономики, а также организация государственного контроля реализации политических установок на переход к инновационному развитию.

Отсутствие спроса национальной экономики на инновации не позволяет запустить и поддерживать механизм перехода к инновационному развитию. Для активного использования инноваций в экономике России и формирования спроса на производимую в технопарках продукцию представляется необходимым создание емких внутренних рынков инновационной продукции, приоритетная государственная поддержка отечественных производителей, повышение доли инновационной составляющей в отечественном производстве и формирование инфраструктуры экономики знаний.

Уровень включения стратегических технологий в систему принятия практических решений по научно-техническому развитию и технологической модернизации российской экономики остается крайне низким. Пока не найдено способов их внедрения в практику инновационной деятельности. В этой связи принципиально важной задачей является приведение системы организации и деятельности технопарков в соответствие с приоритетными направлениями, концентрация их интеллектуальных и организационных ресурсов на стратегических технологиях, обладающих сильным потенциалом создания преимуществ во многих отраслях и видах коммерческой деятельности и имеющих международное значение.

Характер российской экономики и традиции финансирования инноваций таковы, что основная часть исследований осуществляется в организациях, контролируемых государством, а участие российского частного сектора в инновациях пока ограничено преимущественно освоением инноваций. В связи с этим на роль лидера инновационного процесса реально претендовать может пока только государство. Для осуществления поддержки предприятий-разработчиков в технопарках и запуске механизма инновационного развития необходимо создать в России национальный центр инновационной активности (подобный агентству перспективных исследований Министерства обороны в США), который будет брать на себя риски инновационного развития на этапе от реализации идеи до создания промышленного образца, когда становится достаточно очевидной коммерческая перспектива новшества.

Неразвитость научных коммуникаций в РФ ограничивает возможности российских исследователей и разработчиков эффективно участвовать в глобальном трансфере знаний и технологий, продвигать результаты интеллектуальной деятельности на отечественный и мировой рынки, привлекать внешние заказы на НИОКР. Создание и поддержка центротрансфера технологий, налаживание взаимодействия разработчиков инновационных продуктов с крупным консалтинговыми компаниями, способными обеспечить их связь с мировым рынком высоких технологий, представляется важным направлением развития инфраструктуры технопарков. Экономической реализации инноваций может способствовать развитие инструментов передачи прав на результаты бюджетных исследований резидентам технопарков для их коммерциализации [4].

Заключение

Для успешной реализации программы создания технопарков в России имеется целый ряд серьезных предпосылок. Разработка новой продукции в сфере информационно-коммуникационных технологий, особенно программного обеспечения, в целом соответствует основному профилю российского высшего образования, ориентированному на обучение естественным и точным наукам. В России создана достаточно эффективная система подготовки квалифицированных инженеров и специалистов в сфере прикладной математики, вычислительной техники и программирования, конкурентоспособных на мировом рынке труда. Студенты российских вузов неоднократно выигрывали и становились призерами всемирных олимпиад по программированию.

На пути перехода России к инновационной экономике создание технопарков в сфере высоких технологий играет очень важную роль, так как производить инновации без технопарков – это то же самое, что производить автомобили без заводов.

Литература

1. Распоряжение Правительства Российской Федерации от 10 марта 2006 г. № 328-р // Собрание законодательства РФ. – М.: Юридическая литература, 13 марта 2006. – № 11. – Ст. 1226.
2. Государственная программа «Создание в Российской Федерации технопарков в сфере высоких технологий» // Одобрена Правительством РФ 10 марта 2006 г. – № 328-р.
3. Министерство связи и массовых коммуникаций РФ. – 2001–2009. – Основные параметры и показатели технопарков // Информационный портал Технопарки России [Электронный ресурс]. – Режим доступа: <http://www.technoparks.ru/parameters/>, своб.
4. Рубанов В.А. Роль и место технопарков в переходе к инновационной стратегии развития России [Электронный ресурс] // В.А. Рубанов / Электронная библиотека Viperson / Технопарк, 2008. – Режим доступа: <http://viperson.ru/wind.php?ID=486996&soch=1>, своб.

Абрамова Елена Юрьевна – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, Abramova-e@bk.ru

УДК 681.51.015

НОВЫЙ ФУНКЦИОНАЛ ЛЯПУНОВА–КРАСОВСКОГО ДЛЯ ДОКАЗАТЕЛЬСТВА ЭКСПОНЕНЦИАЛЬНОЙ УСТОЙЧИВОСТИ НЕЛИНЕЙНОЙ СИСТЕМЫ С ЗАПАЗДЫВАНИЕМ

А.А. Бобцов, А.А. Пыркин

Рассматривается новый вид функционала Ляпунова–Красовского, доказывающий экспоненциальную устойчивость нелинейной параметрически и функционально неопределенной системы с запаздыванием. В качестве регулятора для стабилизации нелинейной системы был использован метод последовательного компенсатора, разработанный одним из авторов.

Ключевые слова: управление по выходу, системы с запаздыванием, экспоненциальная устойчивость.

На базе метода последовательного компенсатора [1] в статье [2] был рассмотрен алгоритм управления по выходу нелинейной параметрически и функционально неопределенной системы с запаздыванием. Была рассмотрена нелинейная система вида

$$a(p)y(t) = b(p)u(t) + c(p)\omega(t), \quad (1)$$

где $p = d/dt$ обозначает оператор дифференцирования; выходная переменная $y = y(t)$ измеряется, но ее производные недоступны для измерения; $b(p)$, $c(p)$ и $a(p)$ – полиномы с неизвестными коэффициентами; передаточная функция $b(p)/a(p)$ имеет относительную степень $\rho = n - m$; полином $b(p)$ – гурвицев; неизвестная нелинейная функция $\omega(t) = \varphi(y(t - \tau))$ удовлетворяет следующему допущению:

$$|\varphi(y(t - \tau))| \leq C_0 |y(t - \tau)| \quad \text{для всех } y(t - \tau), \quad (2)$$

где $\tau > 0$ – неизвестное постоянное запаздывание, $y(\theta) = \varphi(\theta)$ для $\forall \theta \in [-\tau, 0]$, число $C_0 > 0$ неизвестно. Было показано, что использование метода последовательного компенсатора [1] обеспечивает асимптотическую устойчивость рассматриваемой системе. Для доказательства асимптотической устойчивости был использован функционал вида

$$V(t) = x^T(t)Px(t) + \eta^T(t)N\eta(t) + \kappa \int_{t-\tau}^t y^2(\theta)d\theta \quad (3)$$

и получены достаточные условия, среди которых присутствует ограничение $\kappa \geq C_0^2(\kappa^{-1} + \delta^{-1})^2$, где κ – коэффициент при управляющем воздействии [2]. Целью данной работы является доказательство экспоненциальной устойчивости рассматриваемой системы при использовании того же метода. Вместо используемого в [1] функционала вида (3) рассмотрим новый функционал

$$V(t) = x^T(t)Px(t) + \eta^T(t)N\eta(t) + \kappa \int_{t-\tau}^t e^{-t+\theta} y^2(\theta)d\theta, \quad (4)$$

который отличается наличием экспоненциального члена в подынтегральной составляющей функционала. Дифференцирование (4) приводит к неравенству вида

$$\dot{V}(t) \leq -\gamma V(t), \quad \gamma > 0 \quad (5)$$

при ограничении $\kappa \geq e^\tau C_0^2(\kappa^{-1} + \delta^{-1})^2$. Именно неравенство (5) гарантирует экспоненциальную устойчивость.

Работа поддержана Российским фондом фундаментальных исследований (грант № 09-08-00139-а).

1. Бобцов А.А., Николаев Н.А. Синтез управления нелинейными системами с функциональными и параметрическими неопределенностями на основе теоремы Фрадкова // Автоматика и телемеханика. – 2005. – № 1. – С. 118–129.
2. Бобцов А.А. Стабилизация нелинейных систем по выходу в условиях запаздывания // Известия РАН. Теория и системы управления. – 2008. – № 2. – С. 21–28.

Бобцов Алексей Алексеевич – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, декан, bobtsov@mail.ifmo.ru

Пыркин Антон Александрович – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, ассистент, a.pyrkin@gmail.com

ПРИМЕНЕНИЕ ОНТОЛОГИЙ В СИСТЕМЕ УПРАВЛЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫМИ РЕСУРСАМИ

Д.И. Муромцев, Г.В. Варгин, И.А. Семерханов

Рассмотрена возможность применения онтологий при построении системы управления интеллектуальными ресурсами (СУИР) и преимущества от внедрения семантических сервисов.

Ключевые слова: интеллектуальные ресурсы, онтология, Dublin Core, SPARQL, RDF.

Необходимость поиска методов, обеспечивающих эффективное управление интеллектуальными ресурсами и получение максимальной пользы от интеллектуальной деятельности человека, привело к появлению широкого спектра систем для поддержки процессов управления знаниями организаций. Основная задача СУИР – это хранение и повторное использование накопленных интеллектуальных активов. Одна из таких систем – разрабатываемая авторами СУИР «ReferenceDB».

СУИР реализована в виде веб-приложения, включающего реляционную базу данных (БД), подсистему формирования отчетов по выборкам из БД и поисковую подсистему. Входные и выходные данные системы соответствуют ГОСТ 7.1-2003, ГОСТ 7.1-84, ГОСТ 7.32-2001.

Ключевой возможностью системы является поиск требуемых ресурсов. Применение онтологий может существенно улучшить поисковые возможности. Онтология состоит из классов сущностей предметной области, свойств этих классов, связей между этими классами и утверждений, построенных из этих классов, их свойств и связей между ними [1]. В качестве базового формата хранения данных может быть использован формат RDF [2]. RDF представляет собой абстрактную модель, обеспечивающую способ представления знаний в виде троек субъект–предикат–объект. Данные RDF хранятся в RDF-хранилище, использующем в качестве источника данных реляционную БД. Для извлечения информации из хранилища используются структурированные запросы SPARQL. SPARQL – язык запросов для RDF, он в общих чертах подобен SQL и является рекомендацией консорциума W3C и одной из технологий Semantic Web [3].

К настоящему моменту уже разработано большое число онтологий различных предметных областей. Самыми популярными являются FOAF, SIOC, SKOS и Dublin Core [4]. Благодаря таким элементам, как Title, Subject, Publisher онтология Dublin Core, или «Дублинское ядро», чаще всего используется для описания публикаций. Семантика Дублинского ядра была создана международной междисциплинарной группой профессионалов библиотечного дела, компьютерных наук, кодирования текстов, музейного дела и других смежных групп. Стандарт разделен на простой и компетентный уровни. Простой набор элементов метаданных состоит из 15 элементов, включая такие свойства, как «Title – название», «Creator – создатель», «Subject – тема», «Publisher – издатель», «Contributor – внесший вклад», «Rights – авторские права» и др. В компетентный набор, помимо 15 вышеперечисленных элементов, могут еще входить «Audience – аудитория», «Provenance – происхождение» и «RightsHolder – правообладатель». Каждый элемент опционален и может повторяться. Наличие таких терминов, как «название», «создатель», «издатель», делает стандарт «Дублинское ядро» удобной онтологией для описания публикаций в СУИР.

Интеграция интеллектуального поиска в СУИР позволяет расширить поисковые возможности системы. Помимо обычного поиска по названию и автору публикации, становится возможным выявление связей между семантически близкими материалами, относящимися к совместным проектам или, допустим, связанными с искомой предметной областью. Такой поиск будет осуществлен через SPARQL запросы к подсистеме логического вывода, которая, в свою очередь, будет осуществлять SQL запросы к БД.

Исследования авторов показали, что применение онтологий для реализации семантического поиска в СУИР, позволит расширить спектр решаемых ею задач и превратить в полноценную систему поддержки процессов управления знаниями организации.

1. Stader J., Macintosh A., Capability modelling and knowledge management, Applications and Innovations in Intelligent Systems VII. – Springer-Verlag, 1999. – P. 33–50.
2. RDF Working Group. Resource Description Framework (RDF) [Электронный ресурс]. – Режим доступа: <http://www.w3.org/RDF/>, свободный. Яз. рус. (дата обращения 02.10.2004).
3. Вехорев М.Н. Построение хранилищ онтологических баз знаний // Всероссийская конференция «Управление знаниями и технологиями Semantic-Web», 2010. – С. 165–170.
4. Karen Coyle, Thomas Baker. Guidelines for Dublin Core Application Profiles [Электронный ресурс]. – Режим доступа: / <http://dublincore.org/>, свободный. Яз. рус. (дата обращения 18.05.2009).

Муромцев Дмитрий Ильич – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, mouromtsev@mail.ifmo.ru

Варгин Герман Валерьевич – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, vargin.german@gmail.com

Семерханов Илья Александрович – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, semerhanov@gmail.com

УДК 681.3

ОРГАНИЗАЦИЯ МЕЖМАШИННОГО ОБМЕНА ПРИ РЕЗЕРВИРОВАНИИ МАГИСТРАЛЕЙ

В.А. Богатырев, А.В. Евлахова, Е.Ю. Котельникова, С.В. Богатырев, А.В. Осипов

Рассмотрены варианты межмашинного обмена по двум магистралям с учетом сбоев и повторных передач.

Ключевые слова: отказоустойчивость, резервирование магистралей, пакет, сбой.

Отказоустойчивость вычислительных систем и сетей достигается при резервировании узлов хранения, обработки и передачи данных, в том числе магистралей (коммутаторов). Резервирование магистрали связано с вариантностью организации межмашинного обмена. В работе рассмотрены следующие варианты организации обмена в сети с учетом ненадежности передачи:

А. Пакет передается по одной из магистралей, а в случае его недоставки (нарушение контрольной суммы) он повторно передается после освобождения магистрали;

В. Пакет передается по двум магистралям одновременно, а в случае недоставки обоих пакетов их передача повторяется после освобождения магистралей;

С. Пакет передается по одной из магистралей, а в случае неудачной попытки он передается повторно по той же магистрали без ее освобождения;

Д. Пакет передается по двум магистралям одновременно, а в случае недоставки обоих пакетов они повторно передаются по тем же магистралям (без их освобождения).

Будем считать, что время передачи пакетов распределено по показательному закону со средним значением v . Вероятность правильной доставки пакета по одной и по одной из двух магистралей определим как $p_1 = \exp(-\lambda_c v)$, $p_2 = 1 - (1 - \exp(-\lambda_c v))^2$, а среднее число попыток до успешной передачи пакета как $\alpha_1 = 1/p_1$, $\alpha_2 = 1/p_2$; λ_c – интенсивность сбоев при передаче; λ_0 – интенсивность запросов на передачу пакетов. Среднее время пребывания запросов при интерпретации процесса передачи системой массового обслуживания типа М/М/1 [1] для случаев А–Д по формуле Полячека–Хинчина [1] определим как

$$T_A = \alpha_1 \frac{v}{1 - (\alpha_1 \lambda_0 v / 2)}, T_B = \alpha_2 \frac{v}{1 - (\alpha_2 \lambda_0 v)}, T_C = V_1 + (\lambda_0 V_1 / 2) \frac{V_1(1 + D_1 / V_1^2)}{2(1 - (\lambda_0 V_1 / 2))}, T_D = V_2 + (\lambda_0 V_2) \frac{V_2(1 + D_2 / V_2^2)}{2(1 - (\lambda_0 V_2))},$$

где $V_1 = v/p_1$, и $V_2 = v/p_2$ – средние времена занятия магистрали с учетом нескольких попыток до успешной передачи. Дисперсию времени доставки пакета, когда число попыток до успешной передачи – случайная величина со средним $N_1 = 1/p_1$, $N_2 = 1/p_2$, определим как $D_1 = N_1 D_v + v^2 D_{N_1}$, $D_2 = N_2 D_v + v^2 D_{N_2}$, где дисперсия $D_v = v^2$, а дисперсия числа попыток до успешной передачи по одной D_{N_1} и по одной из двух D_{N_2}

$$D_1 = \frac{p_1}{(1 - p_1)} \sum_{i=1}^{\infty} (1 - p_1)^i (i - \frac{1}{p_1})^2, D_2 = \frac{p_2}{(1 - p_2)} \sum_{i=1}^{\infty} (1 - p_2)^i (i - \frac{1}{p_2})^2.$$

Зависимость среднего времени пребывания запросов в системе от интенсивности запросов λ_0 для вариантов А–Д представлена соответствующими кривыми на рисунке, когда $\lambda_c = 0,05$ 1/с и $v = 2$ с. Из графиков видно влияние рассмотренных вариантов организации обмена на ее эффективность. Влияние отказов сетевых средств на время доставки пакетов может быть учтено на основе моделей [2, 3].

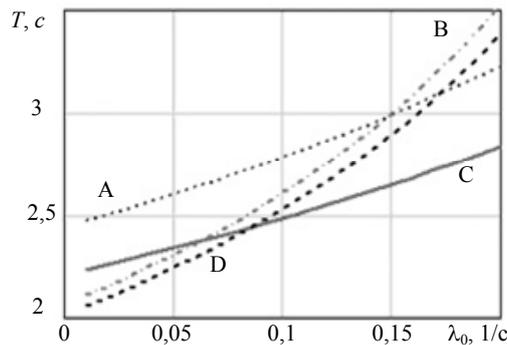


Рисунок. Зависимость времени пребывания запросов в системе от интенсивности запросов λ_0

Расчеты показывают предпочтительность повторной передачи без освобождения магистралей, причем при низкой нагрузке эффективней оказывается одновременная передача по двум магистралям (увеличивает вероятность доставки хотя бы по одной из них), а при высокой – по одной из магистралей (снижает нагрузку).

1. Клейнрок Л. Теория массового обслуживания. – М.: Машиностроение, 1979. – 432 с.
2. Богатырев В.А., Котельникова Е.Ю. Анализ влияния накопления отказов коммуникационных средств на снижение коммуникационных возможностей компьютерных систем с резервированием // Научно-технический вестник СПбГУ ИТМО. – 2010. – № 1. – С. 34–40.
3. Богатырев В.А. Надежность и эффективность резервированных компьютерных сетей // Информационные технологии. – 2006. – № 9. – С. 25–30.

Богатырев Владимир Анатольевич – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, Vladimir.bogatyrev@gmail.com

Евлахова Анастасия Владимировна – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, evlahova.anastasiya@mail.ru

Котельникова Елена Юрьевна – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, elka842@yandex.ru

Богатырев Станислав Владимирович – EMC², старший инженер, realloc@gmail.com

Осипов Андрей Владимирович – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, osipov-andrey@mail.ru

УДК: 339.137.22

ИННОВАЦИИ И ИХ РОЛЬ В СИСТЕМЕ УПРАВЛЕНИЯ КОНКУРЕНТОСПОСОБНОСТЬЮ ПРЕДПРИЯТИЯ В УСЛОВИЯХ СЕРВИСНО-ОРИЕНТИРОВАННОЙ ЭКОНОМИКИ

О.В. Чернышева

Представлены основные результаты исследования инноваций, внедряемых в хозяйственную деятельность предприятий в условиях сервисно-ориентированной экономики. Особое внимание уделено анализу влияния инноваций на конкурентоспособность предприятия в целом. Выработан комплекс направлений по совершенствованию процесса на современном этапе экономического развития.

Ключевые слова: инновации, конкуренция, качество, сервис, персонал, финансы.

Задачей исследования является анализ инновационных процессов, оказывающих наибольшее влияние на конкурентоспособность, рентабельность и эффективность деятельности предприятия, а также выработка предложений по совершенствованию процесса управления конкурентоспособностью предприятия в условиях динамично развивающейся рыночной среды.

По итогам проведенных исследований среди базовых составляющих, оказывающих влияние на конкурентоспособность предприятия, можно выделить инновации в системе управления персоналом, инновации в торгово-технологическом процессе, инновации в сфере рекламы и маркетинга, инновации в финансово-экономической деятельности. Проведем анализ приведенных составляющих, а также выявим их фактическое влияние на эффективность деятельности организации.

Инновационный подход к системе управления персоналом крайне важен, так как последние внедрения новинок управления позволяют повысить эффективность работы персонала в целом. На сегодняшний день активно развивается и внедряется методика «360 градусов». Сущность данной методики заключается в оценке единицы персонала всем окружением. Сотрудника оценивает целый ряд представителей «контактных» аудиторий (от высшего руководства до подчиненных). Оценку проводят также и клиенты, с которыми сотрудник общается в течение рабочего дня. Данная методика применяется сейчас и в ряде кредитных организаций, которые работают как с юридическими, так и с физическими лицами. Оценка производится с использованием анкетирования (опроса). Причем набор показателей для оценки должен различаться у сотрудников, имеющих разные должностные обязанности. При внешней инновационности и необычности, а также видимой справедливости данный подход имеет ряд выраженных недостатков. Во-первых, всегда присутствует фактор предвзятого отношения «коллег по работе», вызванный причинами личностного характера. К положительным сторонам данной методики можно отнести простоту ее применения и внешнюю объективность. При ее практическом внедрении часто возникают трудности в проведении. Если система «360 градусов» организована неграмотно, то это влечет за собой множество проблем, среди которых и возможность возникновения конфликтов межличностного и межгруппового характера. Не всегда можно обеспечить полную анонимность проводимого опроса, в связи с чем могут произойти сбои в работе как отдельно взятого структурного подразделения, так и предприятия в целом. Не всегда оценщики подходят серьезно к данной процедуре, хотя от ее результата во многом может зависеть карьерный рост сотрудника, а также уровень его заработной платы. По этой причине такая инновационность может и не принести объективного желаемого результата [1].

Немаловажное значение в деятельности предприятия играет и маркетинговая составляющая. Новацией в этой сфере является такое направление, как аромамаркетинг. Ароматические добавки, воспринимаемые человеком, могут целенаправленно воздействовать на психику личности. Запахи способны оказывать воздействие на интенсивность совершаемых покупок, их импульсивность и незапланированность. При этом аромат может исходить не только непосредственно от самих продуктов, но и от ароматизированного воздуха. По ароматам человек способен сделать вывод о качестве товара, целевом сегменте, мощности, надежности, долговечности товара. Для каждого типа магазина необходим свой индивидуальный аромат. Универсального аромата, который подходит для магазинов всех направлений, пока нет. Наиболее удачными считаются хвойные, кофейные ароматы, ароматы дыни, шоколада, а также цветочные композиции. Ароматы лаванды, ванили часто используются, например, в магазинах одежды. Популярностью пользуется также вся гамма морских ароматов. Для обувных магазинов для введения покупателей в заблуждение часто используют (реализуя продукцию из заменителей кожи) ароматы натуральной кожи. Большой популярностью пользуются запахи розы. Однако если объектом приложения инноваций является крупный гипермаркет, то не следует выбирать для ароматизации помещения универсальный

аромат. В этом случае следует перейти к процедуре зонирования. Каждая торговая зона должна иметь свой индивидуальный, неповторимый аромат, отвечающий самым взыскательным требованиям.

При организации торгово-технологического процесса предприятия также часто прибегают к использованию последних новинок техники. Однако следует помнить, что наличие только инновационного оборудования не обеспечит эффективного результата работы. При вводе в действие любого оборудования необходимо наличие на предприятии мощного кадрового потенциала. Специальным образом обученный персонал в связке с высокотехнологичным оборудованием предполагает эффективный результат работы компании. Следует также отметить, что грамотная организация покупательских потоков в магазине, а также правильный выбор форм обслуживания позволяет повысить показатели выручки и прибыли как основного показателя хозяйственной деятельности предприятия. Необходимо четко понимать, в каких отделах нужно оставить самообслуживание, а в каких от него отказаться. Зачастую выбор формы обслуживания зависит от вида реализуемого товара и от требований нормативно-правовой базы [2].

Одной из инноваций в сфере общественного питания последнего времени является направление fusion. Направление подразумевает под собой смешение несочетаемых продуктов, а также методов их обработки в одном блюде. Сейчас стиль fusion настолько распространен, что подразумевает слияние кухонь разных стран. Fusion позволяет повысить адаптационные характеристики блюд и сделать их более совершенными и универсальными. Блюда, приготовленные в этом стиле, отличаются своей неожиданностью и непредсказуемостью. Расширение возможностей для приобретения экзотических продуктов в России позволило развить этот стиль и сделать его привлекательным и привычным для российского потребителя. При использовании стиля fusion происходит адаптация уже устаревших рецептов кулинарных блюд к современности, слияние различных культур, экспериментирование в оформлении блюд. Таким образом, национальная кухня адаптируется к запросам местного населения. Однако в ресторанах низкого класса из-за невозможности в финансовом плане привлечения опытных поваров данный стиль используется для сокрытия неумений и невозможностей приготовления блюд национальной кухни. Постепенно мода на данную кухню проходит. На смену ей приходит забытая средиземноморская кухня.

Отметим, что для стабилизации текущей хозяйственной деятельности и развития современному предприятию необходимо использовать в текущей деятельности новейшие системы обработки финансово-экономической информации, в том числе программное обеспечение, которое позволяет провести финансовый анализ предприятия в кратчайшие сроки. Однако универсальные программы не всегда позволяют реализовать плановые цели организации, либо реализуют их, но не точно, что ведет к потерям и возникновению рисков упущенной выгоды. Данное явление вызвано тем, что программы позволяют оперативно проанализировать финансово-экономическое состояние хозяйствующего субъекта, оценить темпы, динамику и характер его развития, однако они выдают конкретный, как правило, числовой результат без его интерпретации с использованием устоявшегося терминологического аппарата. Отказ от использования специальных программ практически в ста процентах случаев приведет к кризисным явлениям или к снижению значений основных показателей хозяйственной деятельности предприятия. По этой причине только симбиоз инноваций и интеллектуального капитала позволит существенно повысить эффективность деятельности предприятия как субъекта рынка [3].

Таким образом, на современном этапе экономического развития в целях соответствия требованиям внешней среды предприятию необходимо активно использовать и внедрять в свою деятельность целый ряд инноваций как в сфере управления, так и в сфере организации торгово-технологического процесса, что в конечном счете позволит своевременно отвечать требованиям внешней среды, усиливать сильные и сводить к минимуму количество слабых сторон. Руководству организации следует своевременно проводить комплексный мониторинг текущего состояния предприятия в целях предотвращения фактов попадания предприятия в предкризисные ситуации, а также выработать комплекс альтернативных путей развития с целью дальнейшего повышения конкурентоспособности.

1. Чернышева О.В. Эффективное управление предприятием как элемент повышения его конкурентоспособности. Финансовые проблемы РФ и пути их решения: теория и практика // Сборник научных трудов 11-й международной научно-практической конференции. Ч.1. – СПб: Изд-во Политехнического университета, 2010. – 338 с.
2. Леви М., Бартон А.Вейтц. Основы розничной торговли. – СПб: Питер, 1999. – 448 с.
3. Чернышева О.В., Мурашова С.В. Банкротство торговых предприятий: Научное издание. – СПб: Изд-во СПбТЭИ, 2008. – 124 с.

Чернышева Ольга Владимировна – Санкт-Петербургский торгово-экономический институт, ст. преподаватель, ov49@mail.ru

SUMMARY

1. AUTOMATA-BASED PROGRAMMING AND ARTIFICIAL INTELLIGENCE TECHNOLOGIES

GENERATION OF FINITE STATE MACHINES FOR UNMANNED AIRPLANE CONTROLLING

A. Alexandrov, S. Kazakov, A. Sergushichev, F. Tsarev, A. Shalyto

A genetic programming method for induction of finite state machines with continuous and discrete output actions is suggested in the article. Instead of the known approach using modeling as a controlling automaton quality estimation and demanding much time we use method of automata behavior comparison with human controlling behavior. Special feature of this approach is the opportunity to use controlling objects not only with discrete but with continuous parameters. This approach is illustrated by the example of finite state machine creation for unmanned airplane controlling in the "loop" mode.

Keywords: finite state machines, genetic programming, unmanned aircraft.

Anton Alexandrov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, alantbox@gmail.com

Sergey Kazakov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, kazakov_sergey_v@mail.ru

Alexey Sergushichev – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, alsergbox@gmail.com

Fedor Tsarev – Saint Petersburg State University of Information Technologies, Mechanics and Optics, postgraduate, fedor.tsarev@gmail.com

Anatoly Shalyto – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Department Head, D.Sc., Professor, shalyto@mail.ifmo.ru

SMALL-SIZED HELICOPTER CONTROL SYSTEM DEVELOPMENT

V. Kleban, A. Shalyto

An example of automatic control system of unmanned small-sized helicopter with a coaxial location of the screws is given. Methods of control theory and digital signal processing are successfully supplemented with the automata-based programming paradigm, which allows effective building of systems with complex behavior.

Keywords: simulation, automata-based programming, coaxial helicopter.

Vitaliy Kleban – Saint Petersburg State University of Information Technologies, Mechanics and Optics, postgraduate, vk.developer@gmail.com

Anatoly Shalyto – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Department Head, D.Sc., Professor, shalyto@mail.ifmo.ru

DOUBLE-STEP GENETIC PROGRAMMING APPLICATION FOR TANK MODEL CREATION IN "ROBOCODE" GAME

D. Sokolov

The article deals with genetic programming application for creation of finite automata controlling complex behavior systems. Double-step genetic algorithm based on dynamic programming ideas is described. This method application for "Robocode" game is considered.

Keywords: automaton, parse tree, genetic programming.

Dmitry Sokolov - Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, dimoz_88@rambler.ru

MACHINE LEARNING APPLICATION FOR CONTROL AUTOMATA CREATION BY "ROBOCODE" GAME EXAMPLE

I. Chernyavskiy

The problem of finite state machine design based on machine learning application is studied and comparison with genetic programming method is given.

Keywords: machine learning, control finite state machines, Robocode.

Il'ya Chernyavskiy – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, chernyavsky@rain.ifmo.ru

AUTOMATIC SYNTHESIS OF MOBILE ROBOT CONTROL SYSTEM FOR SOLVING THE "KEGELRING" PROBLEM

S. Alexeev, A. Kalinichenko, V. Kleban, A. Shalyto

An example of automatic synthesis of mobile robot control system for solving the "Kegelring" problem is given. Automatic synthesis of the system is carried out by the genetic algorithm, which is used to determine the structure of the control automaton.

Keywords: automata-based programming, genetic algorithms, automatic synthesis of control systems.

Sergey Alexeev – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, alex.itmo@gmail.com

Alexander Kalinichenko – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, itrium@gmail.com

Vitaliy Kleban – Saint Petersburg State University of Information Technologies, Mechanics and Optics, postgraduate, vk.developer@gmail.com

Anatoly Shalyto – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Department Head, D.Sc., Professor, shalyto@mail.ifmo.ru

GPU EFFECTIVENESS ANALYSIS FOR AUTOMATIC SYNTHESIS OF ROBOT CONTROL SYSTEM

A. Sergeev, V. Kleban, A. Shalyto

Effectiveness of graphic co-processors (GPU) for the synthesis of automatic control systems (ACS) for a mobile robot is researched. The features of GPU are analyzed in relation to the problem. The parallel implementation of genetic algorithms is chosen as a model task for synthesis efficiency estimation by the example of the ACS synthesis for the “Sumo of Robots” competition.

Keywords: productivity, GPU, genetic algorithm, mobile robot, synthesis of control system, finite state machine, neuron.

Anton Sergeev – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, anton.a.sergeev@gmail.com

Vitaliy Kleban – Saint Petersburg State University of Information Technologies, Mechanics and Optics, postgraduate, vk.developer@gmail.com

Anatoly Shalyto – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Department Head, D.Sc., Professor, shalyto@mail.ifmo.ru

HARDWARE AND SOFTWARE COMPLEX FOR FINITE STATE MOBILE ROBOT CONTROL STUDY

S. Alexeev, V. Kleban, A. Shalyto

Robot control is the task of complex behavior systems control; several approaches are known nowadays for such problems, finite state programming in particular. The article deals with design and research system intended for development and study of finite state programs for mobile robot control.

Keywords: automata-based programming, mobile robots, visual design tool for automatic programs.

Sergey Alexeev – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, alex.itmo@gmail.com

Vitaliy Kleban – Saint Petersburg State University of Information Technologies, Mechanics and Optics, postgraduate, vk.developer@gmail.com

Anatoly Shalyto – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Department Head, D.Sc., Professor, shalyto@mail.ifmo.ru

FEATURE DETECTORS IN THE VIOLA-JONES METHOD ON THE BASIS OF FINITE AUTOMATA

P. Skorynin

Modification of the Viola-Jones method, one of the most effective methods for image classification, is considered. Feature detectors built on the base of finite automata are used instead of the rectangular features detectors in the proposed modification. The use of finite automata allows for reducing the number of levels in the cascade and the number of detectors on each level due to the fact that the automata-based detector can detect more complicated features than simple rectangles.

Keywords: images classification, finite automata, machine learning.

Pavel Skorynin – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, pavel.skorynin@rain.ifmo.ru

STRATEGY OPTIMIZATION METHODS FOR TWO-PLAYER GAMES WITH GENETIC ALGORITHMS

D. Trofimov, A. Shalyto

The article deals with genetic algorithm modification, allowing to solve the optimization problem without setting objective function evidently with the use of comparison function of two candidate solutions. This is achieved by applying the so-called tournament systems. Suggested algorithm enables to use the basic genetic programming principles in problems where it is impossible or too inefficient. For example, the task of an optimal strategy creation against a random opponent in two-player games refers to such problems class.

Keywords: genetic algorithms, two-player games.

Dmitriy Trofimov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, dmitriy.tref@gmail.com

Anatoly Shalyto – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Department Head, D.Sc., Professor, shalyto@mail.ifmo.ru

DEVELOPMENT OF FINITE AUTOMATA CREATION METHODS WITH ANNEALING SIMULATION ALGORITHM BY THE «WAR FOR RESOURCES» EXAMPLE

A. Zaikin

Annealing simulation algorithms are used in the given article to generate control automata of the defender in the "War for Resources" game. The article deals with the use of the annealing simulation algorithm in the problem of a finite-state machine creation that controls the defender in this game and subsequent analysis of the results.

Keywords: annealing simulation algorithm, finite-state machine, "War for Resources" game.

Alexander Zaikin – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, zaikin@rain.ifmo.ru

REPRESENTATION OF FINITE STATE AUTOMATA BY LINEAR BINARY GRAPHS IN GENETIC PROGRAMMING

V. Danilov, A. Shalyto

The article deals with representation of finite state automata by evolutionary algorithm individuals based on linear binary graphs. Comparison of the given method with known ones is done by the example. Suggested method is more efficient as compared with representation of transition function by complete charts. It is more efficient at some condition numbers than representation of transition function by decision trees.

Keywords: genetic programming, finite state automata, linear binary graphs.

Vladimir Danilov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Postgraduate, Vladimir.Daniloff@gmail.com

Anatoly Shalyto – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Department Head, D.Sc., Professor, shalyto@mail.ifmo.ru

AUTOMATIC SELECTION OF ENVIRONMENT PARAMETERS DURING AUTOMATA PROGRAMS GENERATION BY GENETIC ALGORITHMS

V. Kulev

The paper proposes a new method of genetic programming that improves the way of fitness function creation for genetic algorithms testing individuals by simulating the environment. This method makes it possible to select automatically the parameters of the external environment, which leads to quality improvement of generated individuals. It was shown, that the developed approach can be used to generate the automata programs in problems where the behavior of the environment can be described by a finite automaton, by the example of the "Prisoner's iterated dilemma" problem.

Keywords: genetic programming, finite automaton, finite state programming.

Vladimir Kulev – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Postgraduate, me@lightoze.net

GENETIC APPROACH FOR CELLULAR AUTOMATA GENERATION

A. Tikhomirov, A. Shalyto

The article discusses the method for arbitrary cellular automata generation based on test samples using genetic algorithms. Issues arising from applying a classic genetic algorithm to the problem are described. Modified genetic operators for elimination of defects are shown. These operators are tested by a few training examples.

Keywords: cellular automata, genetic algorithms.

Andrey Tikhomirov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, and.tikhomirov@gmail.com

Anatoly Shalyto – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Department Head, D.Sc., Professor, shalyto@mail.ifmo.ru

GENETIC ALGORITHMS APPLICATION TO TEST GENERATION FOR AUTOMATA PROGRAMS

A. Zakonov, A. Shalyto

The article deals with an approach for automated test generation of automata-based programs. Contract approach is suggested to formalize specification requirements to the model and control objects. Test is described as a sequence of transitions in a model. Genetic algorithm is proposed for automation of test creation to find set of values meeting transition conditions.

Keywords: testing, automata-based programming, contracts, genetic algorithms.

Andrey Zakonov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, postgraduate, andrew.zakonov@gmail.com

Anatoly Shalyto – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Department Head, D.Sc., Professor, shalyto@mail.ifmo.ru

TESTS GENERATION FOR OLYMPIAD PROGRAMMING TASKS USING GENETIC ALGORITHMS

M. Buzdalov

An automated method for tests generation against inefficient solutions for Olympiad programming tasks is given. This method is based on genetic algorithms. Method application in generation of new tests for an

Olympiad task from the Internet problem archive acm.timus.ru is described. None of submitted solutions succeeded in passing the resulting test set.

Keywords: genetic algorithms, programming Olympiads, Olympiad tasks, testing.

Maxim Buzdalov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, mbuzdalov@gmail.com

JOINT APPLICATION OF CONTRACTS AND VERIFICATION FOR AUTOMATA-BASED PROGRAMS QUALITY ENHANCEMENT

A. Borisenko, V. Parfenov

Quality assurance is an important aspect of development of software systems with complex behavior. The price of error in such systems may be too high, so it is important not just to check all program specifications, but also to make the process efficient and automated as much as possible. In practice, it can be achieved by formalizing the program requirements and storing the executable specification directly with the program code.

This paper presents the review of existing quality control methods applicable to software systems with complex behavior. The process of environment creation is described supporting three most common approaches to quality assurance of automata-based programs: model checking, unit testing, and contracts. The proposed approach helps to keep program specification up-to-date with availability of interactive quality control.

Keywords: quality assurance, specification compatibility.

Andrey Borisenko – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, Andrey.Borisenko@gmail.com

Vladimir Parfenov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Dean, D.Sc., Professor, parfenov@mail.ifmo.ru

VIRTUAL LABORATORY FOR ARTIFICIAL INTELLIGENCE TRAINING METHODS FOR CONTROLLING FINITE STATE AUTOMATA GENERATION

A. Tyakhti

The structure and capabilities of virtual laboratory on C# programming language are described. This virtual laboratory is intended for training in the area of genetic algorithms and automata-based programming. Main stages of plug-ins development for the laboratory are described.

Keywords: automata-based programming, virtual laboratory.

Alexander Tyakhti – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, tyakhti@gmail.com

2. OPTICAL AND OPTICAL ELECTRONIC SYSTEMS. OPTICAL TECHNOLOGIES

DISCOLORATION OF NATURAL SAPPHIRES

V. Aseev, Y. Nekrasova, K. Khomchenko

Spectral and luminescent characteristics of the faceted crystals of natural sapphires have been investigated. Samples were exposed to technological methods of processing that made it possible to modify selectively the optical density of sapphires in definite spectral regions. Most effective techniques of discoloration of sapphires have been showed.

Keywords: natural sapphire, elevation, spectral-luminescence characteristics.

Vladimir Aseev – Saint Petersburg State University of Information Technologies, Mechanics and Optics, assistant, Aseev@oi.ifmo.ru

Yana Nekrasova – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, Nekrasova@oi.ifmo.ru

Konstantin Khomchenko – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, Delibash69@mail.ru

REPRESENTATION OF EXPERIMENTAL DATA ON ATMOSPHERE OPTICAL PARAMETERS BY ANALYTIC RELATIONS

A. Demin, M. Moiseeva

A creation method of an analytic formula for the atmosphere transmitting efficiency calculation based on experimental data for two spectral regions is proposed. An algorithm of known empirical data sets partitioning into groups according to various analytic forms is described. On this basis an empirical formula for the atmosphere transmitting efficiency calculation is determined.

Keywords: atmosphere transmitting efficiency, algorithm, spectral range, the Earth remote sensing.

Anatoly Demin – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Professor, D.Sc., Professor, dav_60@mail.ru

Maria Moiseeva – Saint Petersburg State University of Information Technologies, Mechanics and Optics, postgraduate, mim14@mail.ru

3. PHOTONICS AND OPTICAL INFORMATICS

CONSISTENT SYSTEM OF UNIQUE KEY QUANTUM DISTRIBUTION ON CHOPPED LIGHT SIDE BAND

A. Rupasov, A. Gleim, V. Egorov, Y. Mazurenko

Functional diagram of unique key quantum distribution system on chopped light sideband with unconditional security is suggested. Sidebands application allows simplifying optical phase insertion into operational signals. Compensation mechanisms of optical modulator polarization dependence and negative influence of birefringence are discussed and proved by experimental results.

Keywords: quantum cryptography, secure key distribution, sidebands.

Andrey Rupasov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, sadbender@yandex.ru

Artur Gleim – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, aglejm@yandex.ru

Vladimir Egorov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, egorovvl@gmail.com

Yuri Mazurenko – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Professor, D.Sc., Senior scientific researcher, yurimaz@gmail.com

PHASE SELF-MODULATION OF SINGLE-CYCLE OPTICAL WAVES

A. Drozdov, S. Kozlov

Analytical and numerical solutions of the equation describing the dynamics of the light pulse field in isotropic dielectric medium with normal group dispersion and cubic delayless nonlinearity are obtained for the boundary condition of single-cycle optical wave. It is shown that spectral broadening of such extremely short pulses by the number of oscillations occurs both in a high- and low-frequency region in nonlinear medium. Spectral density maximum of radiation shifts to low-frequency region. Formation of triple frequency radiation which is customary in the medium with cubic nonlinearity doesn't occur but the wave is generated with the central frequency approximately 4.5 times more than initial wave frequency.

Keywords: single-cycle pulses, phase self-modulation, nonlinear dielectric media.

Arkadiy Drozdov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Postgraduate, arkadiy.drozdov@gmail.com

Sergei Kozlov – Saint-Petersburg State University of Information Technologies, Mechanics and Optics, Dean, D.Sc., Professor, kozlov@mail.ifmo.ru

LIGHT DIFFUSION IN A MEDIUM WITH NANOSCALE HETEROGENEITIES

A. Alfimov, E. Aryslanova, D. Vavulin, O. Andreeva, D. Temnova, V. Lesnichiy, V. Almyashev, S. Kirillova, S. Chivilikhin, I. Popov, V. Gusarov

A method for determining the characteristic size of nanoparticles or other nanoscale heterogeneities is proposed. The method is based on comparing of experimental and theoretical spectral dependences of light attenuation coefficient during the light propagation through heterogeneous medium. One of the possible applications of this method is the analysis of nanoparticle sizes produced in the process of hydrothermal synthesis. The porous glass with known size of pores was chosen as a model of medium.

Keywords: light scattering, diffusion approximation, nanoparticles, hydrothermal synthesis, porous glass.

Anton Alfimov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, Alfimov.anton@gmail.com

Elizaveta Aryslanova – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, Elizabeth.aryslanova@gmail.com

Dmitry Vavulin – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, dima-vavulin@mail.ru

Olga Andreeva – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Associate professor, PhD, senior scientist, olga_andreeva@mail.ru

Daria Temnova – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, daren-ok@bk.ru

Vasilij Lesnichiy – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, kpnk@yandex.ru

Vyacheslav Almyashev – Saint Petersburg State Institute of Technology (Technical University), Assistant, vac@mail.ru

Svetlana Kirillova – Saint Petersburg State Institute of Technology (Technical University), Assistant, refractory-sveta@mail.ru

Sergey Chivilikhin – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Associate professor, PhD, senior scientist, sergey.chivilikhin@gmail.com

Igor Popov – Saint Petersburg State University of Information Technologies Mechanics and Optics, Department Head, D.Sc., Professor, popov@mail.ifmo.ru

Victor Gusarov – Saint Petersburg State Institute of Technology (Technical University), Department Head, D.Sc., Professor, Corresponding member of the RAS, victor.v.gusarov@gmail.com

4. ANALYSIS AND SYNTHESIS OF COMPLEX SYSTEMS

SOLUTION MODEL OF INVERSE PROBLEM OF IONOSPHERE VERTICAL SOUNDING

A. Grishentsev, A. Korobeynikov

Solution model of the inverse problem of ionosphere vertical sounding with reference to the data received at ionospheric "AIS-M" stations is developed. Program application for the automated processing and analysis of spectrograms of high-rise ionosphere sounding is realized on the basis of the developed model. Software product approbation has allowed revealing high enough efficiency of the offered method and positioning the developed model as an effective way of an inverse problem solution for ionosphere high-rise sounding.

Keywords: ionosphere, vertical sounding, spectrogram, inverse problem, high-rise distribution of electronic density.

Alexey Grishentsev – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Associate professor, PhD, tigerpost@ya.ru

Anatoly Korobeynikov – Saint Petersburg Scientific Center of the Russian Academy of Sciences (SPbRC RAS), Deputy director for science, Professor, D.Sc., Professor, Korobeynikov_A_G@mail.ru

PROGRAMMED SERVO DRIVE TRAJECTORY SHAPER

S. Lovlin, M. Tsvetkova, I. Zhdanov

The article describes digital trajectory shaping algorithm used for servo drive directing. There are velocity and acceleration limitation included in this algorithm. The algorithm is intended for working in advance unknown input signal conditions; trajectory shaper is realized in the digital form.

Keywords: motion path, trajectory shaper, intensity shaper, velocity limitation, acceleration limitation.

Sergey Lovlin – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, seri-l@yandex.ru

Madina Tsvetkova – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, madina1986@bk.ru

Ivan Zhdanov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Assistant, zhdanov@ets.ifmo.ru

COST CONTROL ESTIMATION FOR HARMONIC EXOGENOUS ACTIONS: GRAMIAN APPROACH

D. Biryukov, A. Ushakov

The problem of cost control estimation for the harmonic exogenous actions (HEA) is solved based on gramian approach. This approach was formed within the limits of modern control theory and gives the opportunity to calculate the steady output of the system by means of initial state vector and conformity matrix. This approach analytically determines direct connection between steady output of technical object (TO) motion as a part of produced system with source initial state vector, and this connection is realized by similarity matrix which is the solution of Sylvester equation at that. The problem gets transparent solution on the initial states sphere of harmonic exogenous action source, as minor and major values of control costs, a function of mode distribution that is delivered to the system formed by TO and regulator during system synthesis.

Keywords: source of harmonic exogenous action, initial state vector, control object, steady output, cost control gramian, minor and major values.

Dmitriy Biryukov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Postgraduate, dbiryukov@list.ru

Anatoly Ushakov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Professor, D.Sc., Professor, ushakov-AVG@yandex.ru

5. COMPUTER SYSTEMS AND INFORMATION TECHNOLOGIES

BASIC PRINCIPLES FOR SOLVING THE PROBLEM OF TRANSFORMING AN OBJECT-ORIENTED SOURCE CODE TO RDF FORMAT USING SEMANTIC ANALYSIS

A. Zarakovsky, S. Klimenkov, N. Tkachenko, A. Kharitonova

The article describes ways of solving the task of object-oriented source code conversion to RDF format using semantic analysis. Basic principles of semantic analysis of object-oriented code are reviewed and architectural prototype of software is proposed which converts code to RDF format.

Keywords: source code, RDF, semantic analysis, computerized analysis, triplet, grammar, syntax tree, lexer, parser, Java, ANTLR.

Alexey Zarakovsky – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, akkulinoa@gmail.com

Sergey Klimenkov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Assistant, Serge.Klimenkov@Servicom.Ru

Nikita Tkachenko – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, n.i.tkachenko@gmail.com

Anastassia Kharitonova – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Assistant, Anastassia.Kharitonova@Elcom.SPb.Ru

CREATION METHODS ANALYSIS OF MOVING OBJECTS TRAJECTORY ON THE BASIS OF VIDEO DATA SEGMENTATION

I. Rubina

Algorithms research of the segment approach to the solution of the prediction and compensation task of motion is carried out. This class of methods eliminates the most part of disadvantages of solutions on a basis of the pixel approach with high computational complexity of algorithm and the objective approach with complexity in definition of the prediction object form. Because of high computational complexity of possible prediction variants exhaustive search for the block, the optimal selection scheme of blocks for comparison is offered in the article. The number of perfections of existing approaches for the purpose of improvement of relation computing complexity / quality is offered.

Keywords: segment approach, trajectory, object, motion compensation, selection of blocks, comparison of approaches.

Irina Rubina – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Postgraduate, rubren@mail.ru

PROBLEMS OF SEMANTIC DICTIONARY REPLENISHMENT

K. Boyarsky, E. Kanevsky

Problems of computer semantic dictionary replenishment by the new words met in the text during its analysis are considered. The offered system works in a semi-automatic dialogue mode. At the first stage, morphological characteristics of a new word are defined, at the second – we specify its syntactic-semantic parameters by the analogues that are available in the existing dictionary. Suggested approaches provide high precision level. The possibility of new words exact semantic indication in relation to both semantic classes and arguments providing connection with attached words appeared for the first time.

Keywords: text analysis, lexeme, morphology, semantics, syntax, dictionary, word.

Kirill Boyarsky – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Associate professor, PhD, Associate professor, boyarin9@yandex.ru

Evgeniy Kanevskiy – Institute for Economics and Mathematics. RAS, leading scientific researcher, PhD, kanev@emi.nw.ru

ORGANIZATION OF NODES NETWORK INTERACTION IN DISTRIBUTED REPOSITORIES

N. Luk'yanov, A. Dergachev

The article represents information streams processing methods and organization of nodes network interaction used for local network services of each node in distributed data storehouse system for Internet services. The article explains user data movement sequence inside the system, describes optimum node choice method using weight factors ranging and also describes the node data recovery procedure in case of hardware or software system failure.

Keywords: storage, repository, distributed, network, data, algorithms, Internet, service, module.

Nikolay Luk'yanov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Postgraduate, nikolay.lukianov@gmail.com

Andrey Dergachev – Saint Petersburg State University of Information Technologies, Mechanics and Optics, senior teacher, dam600@gmail.com

COMPRESSION METHOD FOR DYNAMIC RANGE OF DIGITAL FOURIER HOLOGRAM WHILE EMBEDDING HIDDEN WATERMARKS

A. Starchenko, Yu. Gatchin

The method of distortion decrease in image-container with embedded hologram of watermark based on hologram dynamic range changing by logarithmic transform is proposed. The proposed method provides decrease of distortions arising at image watermark restoration in comparison with conventional Fourier hologram method. Linear dependence has been exposed between changing of watermark signal amplitude and intensity average quadratic deviation of digital Fourier hologram.

Keywords: watermark, hologram, intensity transform.

Alexei Starchenko – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Postgraduate, ail.lexus_spb84@mail.ru

Yuri Gatchin – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Department Head, D.Sc., Professor, gatchin@mail.ifmo.ru

QUANTITATIVE ASSESSMENT OF INFORMATION IN KNOWLEDGE BASES

I. Bessmertny

The paper concerns the problem of artificial intelligence systems creation concerning information measurement containing in formalized knowledge bases. An approach of information assessment applicable for both natural

and artificial intelligence is proposed. To avoid uncertainty of concepts space we suggest evaluating information capacity within a context.

Keywords: artificial intelligence, quantity of information, context.

Igor Bessmertny – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Associate professor, PhD, Associate professor, igor_bessmertny@hotmail.com

ON THE VULNERABILITY OF VIRTUAL MACHINE MONITORS

Yu. Evelev, G. Chernoknizhny

The article is devoted to virtualization problems security. Vulnerabilities of well-known virtual machine monitors are shown. Examples of exploits are given. VMsafe technology and its abilities to increase the protection system effectiveness in view of security vendors are considered.

Keywords: virtualization, vulnerability, security vendors, exploits.

Yuri Evelev – Saint Petersburg State Engineering and Economic University (ENGECON), student, evelev@xakep.ru

Gennadiy Chernoknizhny – Saint Petersburg State Engineering and Economic University (ENGECON), Associate professor, PhD, Associate professor, gmcher@pochta.ru

6. BIOMEDICAL TECHNOLOGIES

CLOSE-FIELD HIGH FREQUENCY PROBING METHOD IN BIOLOGICAL DIAGNOSTICS

A. Arsen'yev, A. Volchenko, L. Likhacheva, V. Pecherskiy

The article deals with the possibility to define the electrodynamic parameters of biological environment tissues using non-contact method. The possibility of method application for noninvasive biological objects diagnostics is shown. Results of functional activity determination concerning teenagers' locomotor apparatus are given as an example of the method abilities. The main purpose of this work is research of high frequency near-field biological objects diagnostics probing method and its approbation on particular surveys of teenagers' locomotor apparatus.

Keywords: close-field high frequency probing method, growing zones, capacitivity.

Alexey Arsen'ev – Saint Petersburg Children Rehabilitation Orthopaedic and Traumatological Center “Ogonyok”, Head of the clinical department, PhD, stivamat@rambler.ru

Alexander Volchenko – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Postgraduate, wolf2684@mail.ru

Lyudmila Likhacheva – Sanatorium Preventive Clinic “Rodnik”, Head doctor, rodnikperm@mail.ru

Viktor Pecherskiy – Sanatorium Preventive Clinic “Rodnik”, orthopaedist, rodnikperm@mail.ru

NON-INVASIVE MEASURING METHODS OF BILIRUBIN, HAEMOGLOBIN AND GLUCOSE. HAEMOBILIGLUCOMETER DEVICE

E. Kozhokhina

Advantages of non-invasive measuring methods of blood parameters are considered. Haemobiliglucometer device design is proposed.

Keywords: non-invasive, haemoglobin, bilirubin, glucose, absorption, interference filters, microcontroller, monitoring

Elena Kozhokhina – Ukraine National Aviation University, Assistant, miosend@mail.ru

7. ECONOMICS AND FINANCES. MANAGEMENT

SIMULATION MODELING APPLICATION FOR THE QUALITY ASSESSMENT OF INNOVATIVE PROJECTS BUSINESS PLANS

A. Kravets, A. Drobotov

The article deals with the possibility of using simulation modeling (Monte-Carlo method) to assess the quality of innovative projects based on business plan; a method of simulation application for decision making is proposed; soft implementation of method for project risks evaluation based on business plans is done.

Keywords: simulation modeling, Monte-Carlo method, innovation project risk, business plan for an innovative project, risk assessment.

Alla Kravets – Volgograd State Technical University, Associate professor, D.Sc., Associate professor, al1212@rkm.ru

Alexander Drobotov – Volgograd State Technical University, Postgraduate, drobotov@rambler.ru

THE ROLE OF INDUSTRIAL PARKS IN DEVELOPMENT OF RUSSIA AS INNOVATIVE SOCIETY

E. Abramova

Potential realization of the Russian enterprises in the innovative product creation sphere is one of the primary goals of the Russian economy modernization, and Russian industrial parks are to help solving it. However, transition attempts to innovative development come upon essential obstacles: necessity to add other necessary links of innovative process to manufacturers of knowledge and technologies, such as attraction and placing of

research orders for development, formations of demand, innovation advancement, introduction maintenance of innovative decisions and their support with the customer. Indetermination of these problems is aggravated with falling of innovative activity of leading economy sectors.

Keywords: industrial parks, innovations, high technologies, business incubators, knowledge transfer, innovative activity, investments.

Elena Abramova – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Postgraduate, Abramova-e@bk.ru

BRIEF REPORTS

NEW LYAPUNOV-KRASOVSKIY FUNCTIONAL FOR EXPONENTIAL STABILITY PROOF OF NONLINEAR SYSTEM WITH DELAY

A. Bobtsov, A. Pyrkin

The new type of a Lyapunov-Krasovskiy functional is considered to prove the exponential stability of nonlinear system with parametric and functional uncertainties and delay. The method of consecutive compensator developed by one of the authors has been used as a regulator for stabilization of nonlinear system.

Keywords: output control, system with delay, exponential stability.

Alexey Bobtsov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Dean, D.Sc., Professor, bobtsov@mail.ifmo.ru

Anton Pyrkin – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Assistant, PhD, a.pyrkin@gmail.com

APPLICATION OF ONTOLOGY IN THE INTELLIGENT RESOURCE MANAGEMENT SYSTEM

D. Mouromtsev, G. Vargin, I. Semerkhanov

The article considers the possibility of ontology application in the design of the intelligent resource management system and advantages of semantic services introduction.

Keywords: intelligent resources, ontology, Dublin Core, SPARQL, RDF.

Dmitry Mouromtsev – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Associate Professor, PhD, Associate Professor, mouromtsev@mail.ifmo.ru

German Vargin – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, vargin.german@gmail.com

Il'ya Semerkhanov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, i.semerhanov@gmail.com

ARRANGEMENT OF INTER-COMPUTER EXCHANGE LINKS WITH RESERVATION

V. Bogatyrev, A. Evlahova, E. Kotelnikova, S. Bogatyrev, A. Osipov

Variants of an inter-machine exchange on two highways taking into account failures and repeated transfers are considered.

Keywords: fault tolerance, redundancy of exchange links, packet, failures.

Vladimir Bogatyrev – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Professor, D.Sc., Vladimir.bogatyrev@gmail.com

Anastasiya Evlahova – Saint Petersburg State University of Information Technologies, Mechanics and Optics, student, evlahova.anastasiya@mail.ru

Elena Kotelnikova – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Postgraduate, elka842@yandex.ru

Stanislav Bogatyrev – EMC², senior engineer, realloc@gmail.com

Andrey Osipov – Saint Petersburg State University of Information Technologies, Mechanics and Optics, Postgraduate, osipov-andrey@mail.ru

INNOVATIONS AND THEIR ROLE IN THE ENTERPRISE COMPETITIVENESS MANAGEMENT IN THE SERVICE-ORIENTED ECONOMICS

O. Tchernysheva

The article presents the key findings of innovations introduced in the activities of enterprises in service-oriented economics. Special attention is paid to analysis of the innovations impact on the company's competitiveness, on the whole. A set of directions to improve the innovation process at the present stage of economic development is produced.

Keywords: innovations, competition, quality, service, personnel, finance.

Olga Tchernysheva – Saint Petersburg Institute of Commerce and Economics, senior lecturer, ov49@mail.ru