

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

СЕМЬДЕСЯТ ЧЕТВЕРТАЯ
МЕЖДУНАРОДНАЯ СТУДЕНЧЕСКАЯ
НАУЧНАЯ КОНФЕРЕНЦИЯ ГУАП
(19–23 апреля 2021 г.)

Часть 2

ТЕХНИЧЕСКИЕ НАУКИ

Сборник докладов



УДК 001
ББК 72
С30

С30 Семьдесят четвертая Международная студенческая научная конференция ГУАП (19–23 апреля 2021 г.): сб. докл.: в 4 ч. Ч. 2: Технические науки. – СПб.: ГУАП, 2021. – 361 с.
ISBN 978-5-8088-1611-4 (Ч. 2)
ISBN 978-5-8088-1609-1

Представлены доклады Семьдесят четвертой Международной студенческой научной конференции ГУАП, прошедшей в апреле 2021 г.

Научные работы студентов ГУАП посвящены актуальным проблемам авиационного и аэрокосмического приборостроения, проблемам развития новых информационных технологий, радиотехники, электроники и связи, современным проблемам экономики, философии и права.

Под общей редакцией
доктора технических наук,
профессора *В. Ф. Шишлакова*

Редакционная коллегия конференции

Председатель редакционной коллегии:

В. Ф. Шишлаков, доктор технических наук, профессор

Члены редакционной коллегии:

А. Р. Бестугин, доктор технических наук, профессор

А. В. Бобович, заместитель председателя совета УНИДС

В. М. Боер, доктор юридических наук, профессор

А. С. Будагов, доктор экономических наук, доцент

Д. В. Ворновских, директор Ивангородского филиала ГУАП

К. В. Лосев, доктор экономических наук, профессор

Н. Н. Майоров, доктор технических наук, доцент

М. О. Самоловов, директор РИЦ

М. Б. Сергеев, доктор технических наук, профессор

В. В. Скуратов, начальник Военного учебного центра при ГУАП

А. М. Тюрликов, доктор технических наук, профессор

Е. А. Фролова, доктор технических наук, доцент

Н. А. Чернова, кандидат экономических наук

Ответственный секретарь редакционной коллегии *О. Ю. Багиева*

ISBN 978-5-8088-1611-4 (Ч. 2)
ISBN 978-5-8088-1609-1

© Санкт-Петербургский государственный
университет аэрокосмического
приборостроения, 2021

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ

Проблемно-ориентированные вычислительные комплексы

УДК 004.932.72

В. В. Асокова, А. И. Крот

студенты Санкт-Петербургского национального исследовательского
университета информационных технологий, механики и оптики

К. Б. Гурнов – кандидат технических наук, доцент – научный руководитель

АНАЛИЗ АЛГОРИТМОВ ОБРАБОТКИ ИЗОБРАЖЕНИЙ РАЗЛИЧНОЙ ФИЗИЧЕСКОЙ ПРИРОДЫ

Введение

С бурным развитием такого направления, как техническое зрение (ТЗ) различных спектральных диапазонов, стало возможно повысить безопасность полетов и обеспечить экипажи летательных аппаратов высокоточной информацией об окружающей обстановке. Системы ТЗ позволяют вести круглосуточное и метеонезависимое наблюдение, обеспечивают обнаружение и распознавание объектов интереса на различных дальностях и в любое время дня и ночи [1, 2].

Повышение безопасности летательных аппаратов и их эксплуатационных качеств при неблагоприятных погодных условиях может быть достигнуто путем применения так называемых систем комбинированного зрения (мультиспектральное), которые создаются по принципу комбинации данных от различных датчиков, отражающих фактическую информацию о закабинном пространстве, и дополнительной синтезированной информации. Фактически новые технологии направлены на улучшение ситуационной осведомленности экипажа кабины, особенно при посадке, взлете и маневрировании. В этом контексте информация, полученная путем слияния изображений (моно- или многоспектральных бортовых сенсоров) с синтезированной информацией, поддерживаемая данными о состоянии ЛА (координатами, ориентацией, скоростью и т. д.), станет важным и полезным инструментом при управлении в различных режимах пилотирования [3–5].

Разработка подобных систем – это сложная и многоитерационная задача, включающая большое количество информационных технологий, таких как передача данных, базы данных для хранения, специальные средства обработки и визуализации, использование нового поколения мультиспектральных сенсоров.

Корреляционно-экстремальная система навигации (КЭСН) с мультиспектральными датчиками

В современных КЭСН, работающих совместно с системами технического зрения, остро встает проблема уменьшения времени обработки мультиспектральных изображений, что требует оптимизации уже существующих алгоритмов обработки и создания новых. Наибольшие трудности возникают при совмещении разнородных по физической природе изображений. К ним могут быть отнесены радиолокационные изображения, полученные высокоточной съемкой с помощью SAR, и изображения видимого диапазона, созданные с помощью оптических систем.

Наиболее известные методы совмещения – корреляционные, которые, как правило, дают довольно точный результат. Но он достигается при условии близости физической природы изображений, а также при отсутствии различного рода искажений (аффинных) между текущим изображением и эталонным. Аффинные искажения могут быть скорректированы при наличии данных от датчиков, установленных на носителе, и специальных методов предобработки изображений.

Относительно разнородности физической природы возникает проблема дополнительной обработки изображений и приведения их к некоторому «единому» формату. После этого в силу возможных ошибок навигации ЛА для получения хорошего результата потребуются построение большого количества эталонных изображений под различными ракурсами наблюдения и в дальнейшем посредством специальных методов поиска их сравнение. Для систем реального времени это практически неприменимо в силу больших затрат машинного времени.

С целью значительного снижения затрат на подготовку эталонов и проведении вычислений предлагается для мультиспектральных КЭСН использовать метод контурного анализа или метод, основанный на анализе критериальных функций.

Суть такого подхода заключается в следующем:

- выделение контуров объектов на текущем изображении;
- аппроксимация контуров на текущем и эталонном изображении;
- выбор критериальной функции и совмещение изображений;
- оценка качества совмещения.

В рамках данной работы рассмотрена современная КЭСН и ее особенности при определении координат на базе сопоставления текущего и эталонного изображений, а также разработано ПО, реализующее обработку мультиспектральных изображений в ней.

Разработка моделирующей программы

Алгоритм работы программы моделирования состоит из четырех пунктов:

- ввод данных – «текущее изображение» и «эталон», а также искажения масштаба и угла поворота;
- выбор метода предварительной обработки изображений;
- выбор метода сопоставления и запуск алгоритма сопоставления;
- вывод результатов работы.

На основе перечисленных пунктов был разработан интерфейс программы, созданный при помощи среды разработки программ с графическим интерфейсом GUIDE пакета Matlab (рис. 1).

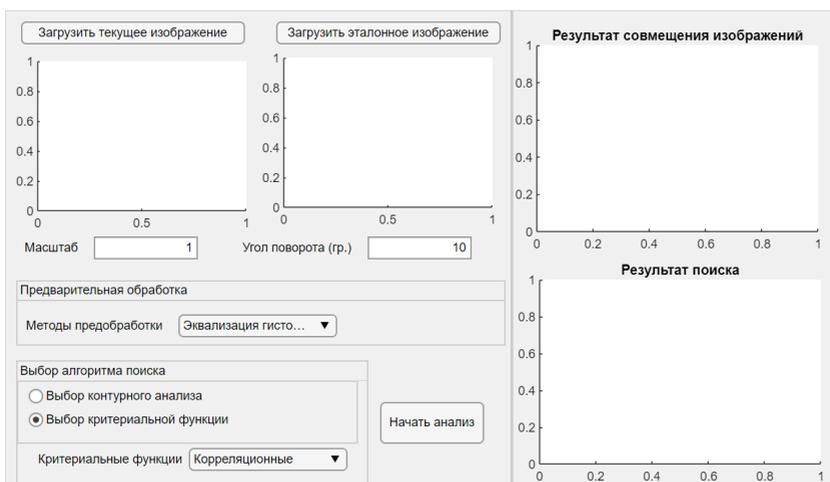


Рис. 1. Интерфейс программы поиска объектов в мультиспектральных КЭСН

При нажатии кнопки «Загрузить текущее изображение» или «Загрузить эталонное изображение» откроется окно проводника, в котором требуется выбрать изображение формата .jpeg, .png, .bmp, .tif и др. Далее с помощью строк ввода «масштаб» и «угол поворота» оператор может задать интересующее его начальное значение этих параметров. Масштаб может быть задан в диапазоне 0...2, а угол поворота 0...360° с произвольным шагом. Это позволяет симитировать аффинные искажения между текущим и эталонным изображениями, которые часто встречаются в КЭСН. Когда текущее изображение и эталон будут загружены в ОЗУ, в блоке ввода данных в графических окнах будут отображены загруженные изображения с учетом введенных данных «масштаб» и «угол поворота» (рис. 2).

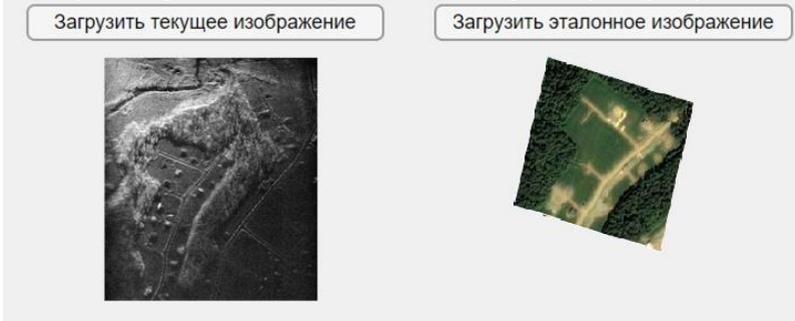


Рис. 2. Загрузка текущего и эталонного изображения

Программа позволяет выбрать методы предварительной обработки изображений с помощью выпадающего списка (рис. 3): преобразование яркости, эквализация гистограммы яркости, пространственная фильтрация и обработка в частотной области.

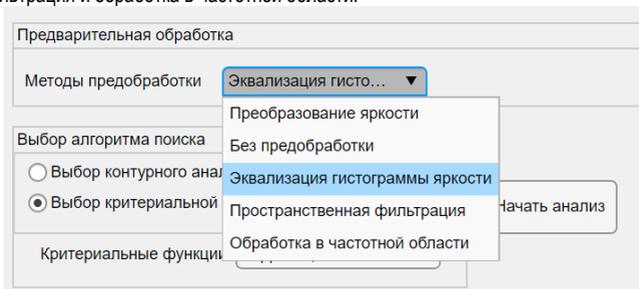


Рис. 3. Выбор методов предварительной обработки

Далее необходимо выбрать метод поиска объекта: критериальная функция или контурный анализ (рис. 4).

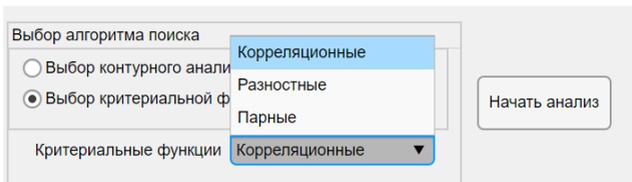


Рис. 4. Выбор метода обнаружения

При нажатии «Выбор критериальной функции» появится выпадающий список и необходимо будет выбрать одну из трех функций из него. После нажатия кнопки «Начать анализ» в графических окнах будет отображен результат работы программы. В зависимости от выбора метода поиска информация будет представлена по-разному для более удобного анализа результатов.

При использовании критериальных функций в первом окне будет отображено исследуемое изображение с выделенной областью, центр которой соответствует пику выбранной критериальной функции, т. е. результат поиска эталона на текущем изображении (рис. 5).

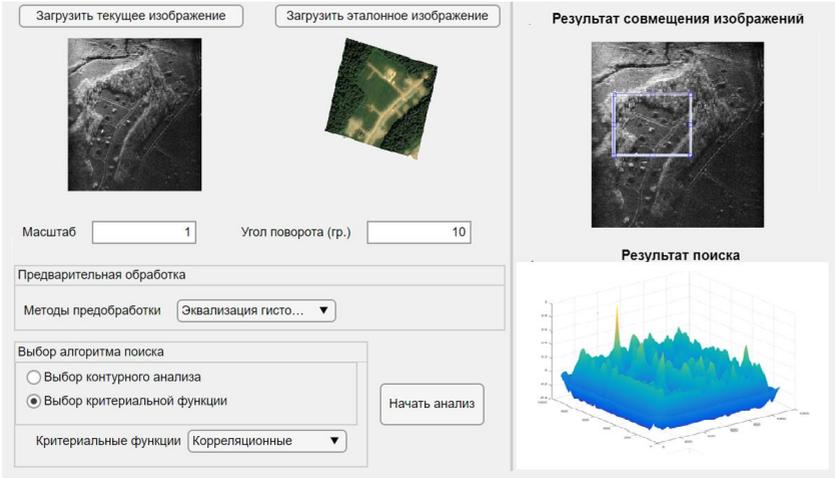


Рис. 5. Результат работы программы с использованием корреляционного анализа

При выборе контурного анализа результат будет отображаться на карте контуров текущего изображения, соответствующего найденному контуру эталона. В соседнем окне будет представлен контур эталона (рис. 6).

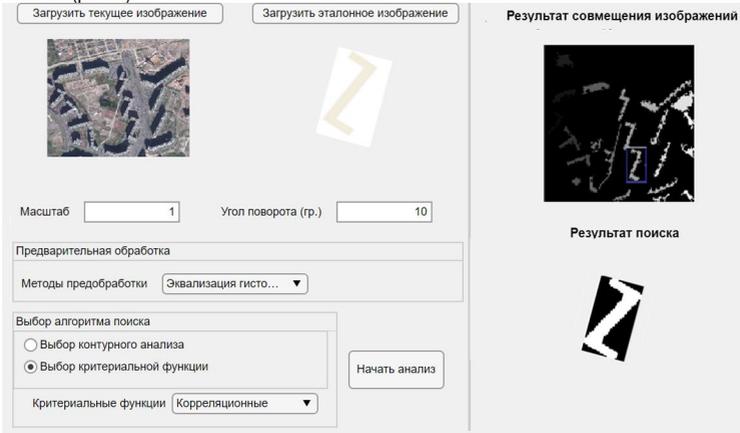


Рис. 6. Результат работы программы с использованием контурного анализа

Заключение

Был найден алгоритм поиска объектов на мультиспектральных изображениях на базе корреляционного и контурного анализов. На этой основе было разработано программное обеспечение, использующее указанные алгоритмы, благодаря чему можно проводить исследовательскую деятельность в различных условиях и для разнообразных типов изображений.

Библиографический список

1. Методы автоматического обнаружения и сопровождения объектов. Обработка изображений и управление / Б. А. Алпатов, П. В. Бабаян, О. Е. Балашов [и др.]. М.: Радиотехника, 2008. 176 с.
2. Методы компьютерной обработки изображений / под ред. В. А. Сойфера. 2-е изд., испр. М.: Физматлит, 2003. 784 с.
3. Злобин В. К., Еремеев В. В., Кузнецов А. Е. Обработка изображений в геоинформационных системах: учеб. пособие / Рязан. гос. радиотехн. ун-т. Рязань, 2006. 264 с.
4. Злобин В. К., Еремеев В. В. Обработка аэрокосмических изображений. М.: Физматлит, 2006. 288 с.
5. Обработка и анализ изображений в задачах машинного зрения: курс лекций и практ. занятий / Ю. В. Визильтер, С. Ю. Желтов, А. В. Бондаренко [и др.]. М.: Физматкнига, 2010. 672 с.

УДК 528.851

В. А. Афанасьев, Д. В. Пенкин

студенты кафедры проблемно-ориентированных вычислительных комплексов

Е. П. Виноградова – старший преподаватель – научный руководитель

ПРОЕКТИРОВАНИЕ БЕСПИЛОТНОЙ МАШИНЫ ПАТРУЛИРОВАНИЯ ЛЕСНЫХ НАСАЖДЕНИЙ

Развитие современных беспилотных аппаратов (в том числе летательных), усовершенствование систем навигации и улучшение силовых установок позволяют многие сложные и дорогостоящие операции проводить без необходимости непосредственного нахождения человека в зоне выполнения задач.

Беспилотная разведывательно-дозорная машина предназначена для разведки отдаленных от населенных пунктов или точек базирования территорий, охране и патрулированию обширных территорий в любую погоду или время суток. Проектируемая беспилотная разведывательно-дозорная машина предназначена для решения задач по патрулированию и охране заповедников, тушению пожаров в труднодоступных участках местности. Также она может выполнять задачи по разведке труднодоступных территорий, в том числе в условиях плохой видимости, например оценку очага лесного пожара. Помимо указанных задач, машина может быть использована как транспорт, легкий тягач или мобильная электростанция.

Общие характеристики машины:

- длина – 4 614 мм;
- ширина – 2 000 мм;
- высота – 2 050 мм;
- количество колес – 8;
- диаметр колеса 700×200 мм;
- тип подвески – торсионная;
- размеры башни: длина – 3 020 мм; ширина – 1 250 мм; высота – 790 мм.

Пусковые системы и системы обзора

Для обзора и обнаружения целей машина оснащена лидаром, РЛС «Фара», ОЛС и системой расположенных по всему корпусу камер, а также приборами теплового и ночного видения. Это позволяет обнаруживать как наземные цели, так и низколетящие, а также отслеживать их передвижение в пределах видимости РЛС и систем наблюдения. Также данное оснащение дает возможность свободно ориентироваться в темное время суток или в условиях плохой видимости, например при задымлении местности [1].

Машина оборудована системами пуска дымовых гранат и осветительных снарядов. С помощью дыма она может отметить какой-либо участок местности, например точку сброса необходимых припасов, неподготовленную вертолетную площадку или место обнаружения посторонних на закрытой территории. Система пуска осветительных снарядов позволяет обеспечивать приемлемый уровень освещения в темное время суток, что улучшает взаимодействие машины с наземными группами. Также осветительные снаряды необходимы для подачи различных сигналов [2].

В башне установлен водомет диаметром 30 мм. Он предназначен для тушения и локализации пожаров, их профилактики путем увлажнения местности, а также противодействия посторонним личностям, нарушившим границы закрытой территории или совершающим противоправные действия на территории заповедника. Для питания водомета в подбашенной корзине установлен бак на 500 л воды. Машина оснащена насосом, и с помощью разъема в кормовой плите можно осуществить дозаправку.

Корпус машины состоит из 20-миллиметровых катаных бронеплит. Он изготавливается двухслойным – на расстоянии от внешней плиты с помощью шпангоутов и лонжеронов крепится плита внутренней части корпуса. Такая конструкция благодаря воздушному зазору позволяет работать машине в условиях высоких температур без риска перегрева внутренних модулей, а силовые элементы

конструкции между плитами увеличивают живучесть машины при столкновениях, падениях или ударах тяжелыми предметами. Также корпус покрыт огнестойкой краской.

Оборудование управления машиной основано на отечественных образцах вычислительных машин. Управление производится оператором на расстоянии со специализированного рабочего места. Комплекс обработки и обмена информацией должен обеспечивать выдачу целеуказания на основе данных от систем обнаружения и навигации, а также иметь быструю обратную связь с управляющими сигналами от рабочего места оператора. При запуске системы управления производится автономное тестирование систем, которое можно пропустить при экстренном запуске. В случае потери сигнала или наличия активных помех машина автономно на основании данных навигационной системы о последних точках с устойчивой связью последовательно возвращается в каждую из них до тех пор, пока соединение с пунктом управления не будет восстановлено [3].

В качестве силовой установки выбран двигатель ISB6.7E5310 фирмы Cummins [4]. Двигатель расположен поперечно в задней части. Он имеет высокий крутящий момент 1 200 Н·м и достаточную мощность 226 кВт, обеспечивающие хороший разгон и высокую максимальную скорость, что позволяет оперативно направить или эвакуировать машину. Для обслуживания машины в ее задней части расположены технические люки, а также имеется разъем от внутреннего генератора для использования машины в качестве мобильной электростанции. Топливный бак расположен в лобовой проекции, его объем 100 л. Для использования машины в качестве тягача на кормовой плите имеется фаркоп.

Библиографический список

1. Шипунов А. Г. Унифицированная система запуска дымовых гранат. URL: <http://www.russianarms.ru/forum/index.php?topic=5486.04> (дата обращения: 23.03.2020).
2. Модернизированная портативная радиолокационная станция разведки наземных движущихся целей с панорамным индикатором. URL: <https://roe.ru/catalog/sukhoputnyevosyka/vozdushnye-misheni/fara-vr/> (дата обращения: 01.04.2020).
3. Лидар. URL: <https://gistroy.ru/article/lidar/> (дата обращения: 05.04.2020).
4. Двигатели Cummins серии B6.7. URL: <https://www.cumminskama.ru/rus/produkcija/dvigateli-cummins-serii-isb-6-7/> (дата обращения: 10.04.2020).

УДК 004.457

А. Р. Веселов, С. В. Золотарев

магистранты кафедры проблемно-ориентированных вычислительных комплексов

Е. Л. Турнецкая – кандидат технических наук – научный руководитель

ВЫБОР ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ИНВЕНТАРИЗАЦИИ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА ПРЕДПРИЯТИЯ

Учет вычислительной техники, установленного программного обеспечения (ПО) и компонентов интегрированной информационной среды предприятия позволяет получить оперативную информацию о текущем состоянии IT-инфраструктуры и повысить эффективность применения имеющихся ресурсов.

Цель настоящей работы – выбор ПО для автоматизированной инвентаризации программно-аппаратного комплекса предприятия. Для ее достижения требуется решить следующие задачи: провести обзор рынка ПО учета компонентов IT-инфраструктуры на основе самостоятельно разработанных критериев, выбрать технические средства для проведения инвентаризации, проанализировать возможность внедрения выбранных программных и технических средств в информационную среду предприятия.

Рассмотрим перспективные программные решения в области инвентаризации компьютеров, учета офисной техники, принтеров и картриджей, оргтехники, программного обеспечения, комплектующих, расходных материалов и т. д. Для сравнительного анализа между ПО выделим следующие критерии: тип сервера, на который будет инсталлирована программа; количество рабочих мест пользователей; максимальное количество учетных записей в базе данных; возможность проведения инвентаризации программного обеспечения; возможность ведения учета других видов объектов, которыми владеет предприятие, и их стоимость. Результаты сравнения приведены в табл. 1.

Таблица 1

Сравнительный анализ ПО для инвентаризации

Критерии сравнения	Hardware Inspector	IT Invent	OCS Inventory NG	Сетевой монитор Dude	NetXMS
Установка на сервер	Windows	Windows	LINUX	Windows	Windows и MacOS
Количество рабочих мест	∞	∞	∞	∞	∞
Количество записей каждого вида учетных единиц	∞	3 000	∞	∞	∞
Инвентаризация программного обеспечения	+	+	+	+	+
Возможность добавлять любое оборудование	+	+	–	–	–
Стоимость	26 000 руб. (6 500×4)	48 000 руб.	Бесплатно	Бесплатно	Бесплатно

Охарактеризуем дополнительные параметры некоторых из рассмотренных программных средств.

Сетевой монитор Dude (<https://mikrotik.com/thedude>) – это приложение от компании MikroTik, которое автоматически сканирует все устройства в пределах заданных подсетей, рисует карту заданных сетей, контролирует сервисы устройств и предупреждает в случае возникновения проблем с некоторыми сервисами [1]. Обладает нерусифицированным интерфейсом.

NetXMS (<https://www.netxms.org/>) – это ПО с открытым кодом для мониторинга компьютерных систем и сетей. Может быть использовано для мониторинга всей IT-инфраструктуры: коммутаторов и

маршрутизаторов, программного обеспечения на серверах и т. д. Система имеет трехуровневую архитектуру: информация собирается агентами мониторинга и предоставляется мониторинг серверу для обработки и хранения. Сетевой администратор получает доступ к собранным данным с помощью клиентского приложения или веб-интерфейса. Отсутствует возможность добавления стороннего оборудования [2].

Программа Hardware Inspector Client/Server (<https://dist.1c.ru/products/item/hardware-inspector-client-server/>) предназначена для учета компьютеров на предприятии. Ее уникальность заключается в возможности вести учет «всей истории жизни» каждого вычислительного устройства или другого оборудования [3].

Программное средство Hardware Inspector (<https://www.hwinspector.com/>) проводит низкоуровневый опрос оборудования, что позволяет учесть все компоненты вычислительных устройств. Параллельно автоматическому вводу информации предусмотрен ручной ввод данных с помощью сканера. При этом формируется полноценная база данных с отслеживанием всех перемещений и обслуживаний. ПО предлагает два метода контроля за изменениями конфигураций: с помощью механизма оперативного регулярного контроля и процедуры строгой инвентаризации с помощью штрих-кодов [4].

Таким образом, в качестве программного средства для инвентаризации выбрано ПО Hardware Inspector. На этапе интеграции программы в информационную среду предприятия внесены требуемые изменения в конфигурацию. Для проверки функционирования ПО проведено автоматическое и ручное сканирование ИТ-инфраструктуры с добавлением компьютеров, серверов, сетевого оборудования, периферийных устройств, программного обеспечения разных версий и других материальных ценностей в систему учета.

Библиографический список

1. The Dude: официальный сайт. URL: <https://www.mikrotik.com/thedude> (дата обращения: 05.04.2021).
2. NetXMS: официальный сайт. URL: <https://www.netxms.org/> (дата обращения: 05.04.2021).
3. Краткое руководство по настройке базы данных Hardware Inspector и Hardware Inspector Client/Server. URL: <https://docplayer.ru/30277633-Kratkoe-rukovodstvo-po-nastroyke-bazy-dannyh-hardware-inspector-i-hardware-inspector-client-server.html> (дата обращения: 05.04.2021).
4. Hwinspector: официальный сайт. URL: <http://www.hwinspector.com> (дата обращения: 05.04.2021).

УДК 004.624

В. В. Вихров

магистрант кафедры проблемно-ориентированных вычислительных комплексов

Е. А. Бакин – кандидат технических наук, доцент – научный руководитель

АВТОМАТИЗАЦИЯ СБОРА ИНФОРМАЦИИ О СТОИМОСТИ НЕДВИЖИМОСТИ ИЗ ОТКРЫТЫХ ИСТОЧНИКОВ ДАННЫХ

Построение систем предсказания стоимости недвижимости осложняется отсутствием в открытом доступе подходящих по объему данных для анализа, которые также часто неполны, что приводит к значительной ошибке предсказания даже для наиболее эффективных алгоритмов. В связи с этим в исследовании будет представлен механизм сбора данных о недвижимости для дальнейшего построения более точного предсказания цены.

При взятии данных из готовых датасетов наблюдаются такие проблемы, как недостаточный объем данных, ограниченность одним городом, а также отсутствие важных факторов, например вида ремонта, количества комнат и пр.

В России основными агрегаторами недвижимости являются «Яндекс Недвижимость», «Авито», «Циан» и др. Все они подходят для сбора данных, так как содержат большой их объем для большинства городов России. При «общении» с сайтами вся информация проходит по некоторому пути, который представлен на рис. 1 [1].

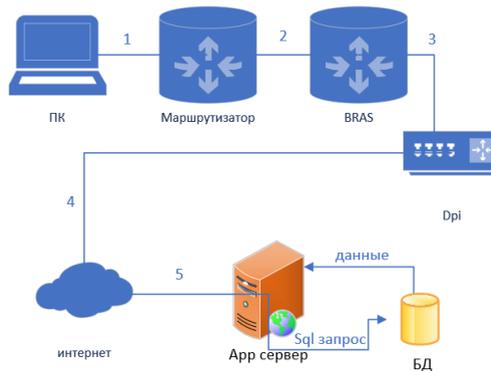


Рис. 1. Упрощенная схема взаимодействия персонального компьютера и интернет-ресурса

Проходя по этому пути, мы получаем на ПК некоторую часть базы данных, в нашем случае объекты недвижимости. Одной из технических сложностей при реализации парсера может стать взаимодействие с *app*-сервером, на котором могут быть реализованы различные алгоритмы по регуляции автоматической выгрузки данных. Остальные этапы движения данных для парсера в большинстве случаев не предоставляют принципиальной сложности. Потенциальная проблема DPI, как правило, нивелируется тем, что глубокая инспекция не охватывает всех возможных паттернов и, вероятнее всего, для создаваемого парсера будут отсутствовать соответствующие алгоритмы [2].

Для создания парсера был использован высокоуровневый язык программирования Python, так как он достаточно универсален для многих задач, поскольку содержит большое количество готовых библиотек для различных прикладных задач. Основными библиотеками для создания парсера были *requests* и *bs4*. Детали использования основных, а также вспомогательных библиотек можно увидеть в табл. 1.

Используемые библиотеки и их функционал

Библиотека	Версия	Используемый функционал
Bs4	0.0.1	Получение данных из html-страницы
Requests	2.25.1	Получение html-страницы
Time	1.0.0	Замедление работы программы для защиты от блокировки сервером
Numpy	1.18.2	Генерация случайного количества секунд сна программы
Fake_useragent	0.1.11	Выбор случайного параметра useragent при каждом запросе для защиты от блокировки сервером
Csv	1.0	Запись полученных данных в csv-файл
Socks	1.0	Выбор порта, с которого будет работать программа
Socket	3.7	В связке с socks настраивает работу через прокси

Среди основных проблем, возникших в процессе разработки, можно выделить следующие.

1. Заголовок useragent, используемый библиотекой *requests*, презентировал себя серверу как парсер, в ответ на что сервер выдавал ошибку 503. В связи с этим была использована библиотека *fake_useragent* для изменения этого заголовка при каждом новом запросе страницы [3].

2. При запросе страниц чаще, чем один раз в секунду, сервер тут же блокирует передачу с ошибкой 403. В связи с этим с помощью библиотек *numpy* и *time* была введена случайная пауза между запросами от 1 до 2 с. Однако даже при задержке между запросами сервер все равно периодически осуществлял блокировку в силу частого запрашивания страницы за страницей. Эта проблема может быть решена двумя способами. Первый – смена IP 1 раз в 10 с. Второй – использование мобильного телефона в качестве WiFi-роутера. По всей видимости, это объясняется наличием «белого списка» IP мобильных операторов

3. При выборе любого фильтра максимальное количество страниц с квартирами ограничивается 54. На них помещалось около 1 500 квартир, что достаточно много, а при использовании различных фильтров (таких как стоимость, количество комнат, дата объявления) можно выбирать непересекающиеся множества квартир.

После решения проблем работы парсера он был запущен и, обрабатывая порядка 28 квартир в минуту, выгрузил данные о 1 500 квартир приблизительно за час. За двое суток было получена информация о 20 000 квартирах, располагающихся в Санкт-Петербурге. Планируется обработать большинство городов-миллионников страны и набрать достаточное количество данных для более точного анализа цен на недвижимость по разным городам России.

Выводы

В ходе исследования были получены следующие результаты.

1. Была рассмотрена схема информационного обмена между пользователем и сервером.
2. Рассмотрены проблемы при написании парсера, а также их возможные решения.
3. Была осуществлена оценка временных характеристик работы парсера – 28 квартир в минуту.
4. На данный момент собраны сведения приблизительно по 20 000 квартирам в Санкт-Петербурге.

Библиографический список

1. Хазов В. Российские производители DPI и их платформы // Блог компании VASExperts. URL: <https://vasexperts.ru/blog/rossijskie-proizvoditeli-dpi-i-ih-platfo/> (дата обращения: 03.04.2021).
2. Краткий обзор технологии DPI. URL: <https://habr.com/ru/post/111054/inspection> (дата обращения: 11.04.2021).
3. Документация использования библиотеки *fake_useragent*. URL: <https://pypi.org/project/fake-useragent/> (дата обращения: 8.04.2021).

УДК 519.246.2

А. С. Волкова, А. С. Раскопина

студенты кафедры проблемно-ориентированных вычислительных комплексов

М. С. Иванова – ассистент – научный руководитель

СРАВНЕНИЕ АЛГОРИТМОВ ПО СКОРОСТИ ФОРМИРОВАНИЯ ВЫБОРКИ СЛУЧАЙНЫХ ЧИСЕЛ

Сравниваются алгоритмы формирования выборки случайных чисел по скорости. Рассматриваются алгоритмы формирования по плотности распределения Симпсона и функции распределения закона Симпсона методом обратной функции.

Алгоритм плотности вероятности закона Симпсона

Функция распределения закона Симпсона [1]

$$F(x) = \begin{cases} 0, & x < 0 \\ \frac{x^2}{2}, & 0 \leq x \leq 1 \\ 1 - \frac{(2-x)^2}{2}, & 1 < x \leq 2 \\ 1, & x > 2 \end{cases} \quad (1)$$

Дифференцируя эти выражения, получим

$$f(z) = \begin{cases} 0, & z < 0 \\ z, & 0 < z \leq 1 \\ 2-z, & 1 < z \leq 2 \\ 0, & z > 2 \end{cases} \quad (2)$$

где $z = x + y$.

К распределению по закону Симпсона приводит сложение двух случайных величин, подчиненных закону равной вероятности при одинаковых параметрах рассеяния. Кривая рассеяния имеет вид равнобедренного треугольника, из-за чего этот закон часто называют законом треугольника (рис. 1) [1].

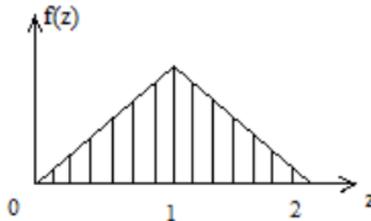


Рис. 1. Плотность вероятности закона Симпсона

Для получения случайных чисел, распределенных по закону Симпсона, генерируем случайные числа x и y , распределенные по равномерному закону в интервале от 0 до 1, складываем их и получаем искомую величину z . Находим интервал, в который попадает число z , по выражению (2), и если z попадает в интервал от 0 до 2, то делим его на равные отрезки по 0,1 [2].

Для $z \in (0;1]$ вероятность попадания случайной величины в данный интервал равна площади прямоугольной трапеции $S = \frac{h(a+b)}{2}$, где h – высота равная длине отрезка, a и b будут равны $z(i)$ и $z(i+1)$, где i – номер отрезка. Тогда $S = 0,05 * (z(i) + z(i+1))$.

Для $z \in (1;2]$ a и b будут равны $2 - z(i)$ и $2 - z(i+1)$, тогда $S = 0,05 \cdot ((2 - z(i)) + (2 - z(i+1)))$.

Для $z > 2$ площадь S равна нулю.

Для $z < 0$ площадь S равна нулю.

На рис. 2 приведен алгоритм программы измерения времени формирования выборки случайной величины в среде MATLAB по плотности распределения Симпсона.

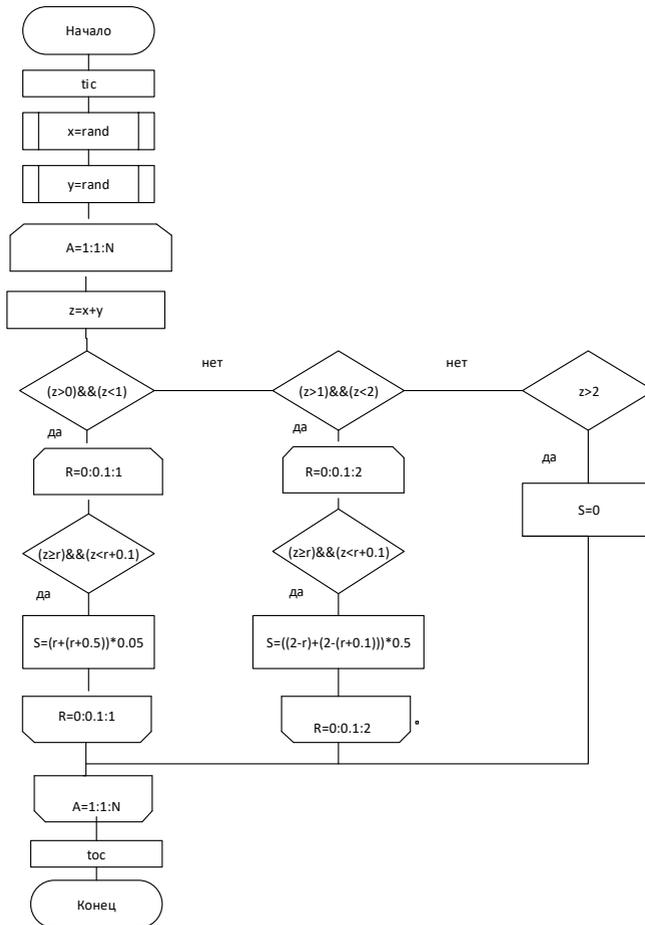


Рис. 2. Алгоритм программы измерения времени формирования выборки случайно величины в среде MATLAB по плотности распределения Симпсона

Алгоритм функции распределения закона Симпсона методом обратной функции

Пусть $y = F(x)$ – функция распределения закона Симпсона (1), тогда обратная ей функция $F^{-1}(y)$ получается из выражения $x = F^{-1}(y)$ и имеет вид

$$x = \begin{cases} 0, & F(x) = 0 \\ \sqrt{2F(x)}, & 0 < F(x) \leq 0,5 \\ 2 - \sqrt{2 - 2F(x)}, & 0,5 < F(x) \leq 1 \end{cases} \quad (3)$$

На рис. 3 приведен закон распределения Симпсона.

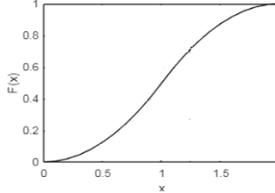


Рис. 3. Закон распределения Симпсона

Генерируем случайное число y , равномерно распределенное в интервале от 0 до 1, имеющее значение функции распределения. Находим, к какому интервалу принадлежит y (3), и вычисляем x методом обратной функции.

На рис. 4 приведен алгоритм программы измерения времени формирования выборки случайной величины в среде MATLAB по функции распределения закона Симпсона методом обратной функции.

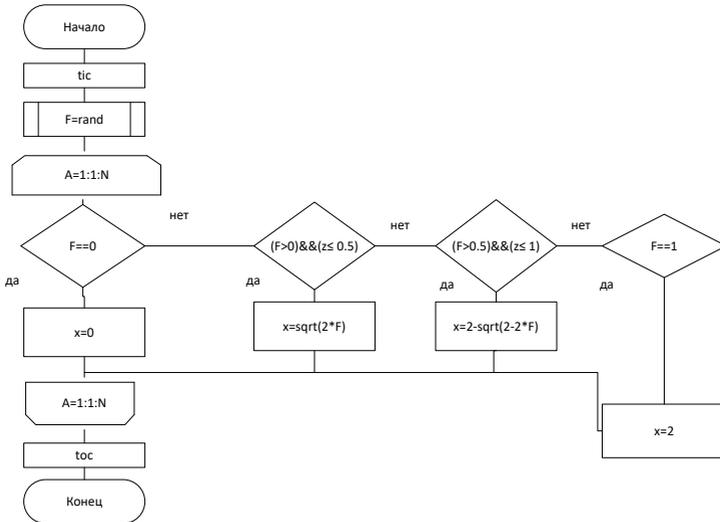


Рис. 4. Алгоритм программы измерения времени формирования выборки случайной величины в среде MATLAB по функции распределения закона Симпсона методом обратной функции

Для сравнения алгоритмов по быстродействию была произведена 1 000 опытов для выборки N случайных чисел от 1 до 1 000 000 (табл. 1, рис. 5) и посчитаны средние значения времени работы программ.

Таблица 1

Время работы алгоритмов для разных величин выборки, с

Номер выборки	1-й метод	2-й метод
1	$6,335 \cdot 10^{-6}$	$4,179 \cdot 10^{-6}$
10	$1,992 \cdot 10^{-5}$	$4,375 \cdot 10^{-6}$
50	$3,310 \cdot 10^{-5}$	$6,720 \cdot 10^{-6}$
100	$5,788 \cdot 10^{-5}$	$1,011 \cdot 10^{-5}$
500	$2,680 \cdot 10^{-4}$	$3,926 \cdot 10^{-5}$
1 000	$5,496 \cdot 10^{-4}$	$9,041 \cdot 10^{-5}$
5 000	0,002 6	$3,322 \cdot 10^{-4}$
10 000	0,005 0	$6,472 \cdot 10^{-4}$
50 000	0,025 0	0,003 2
100 000	0,049 2	0,006 3
500 000	0,251 1	0,031 3
1 000 000	0,595 6	0,066 8

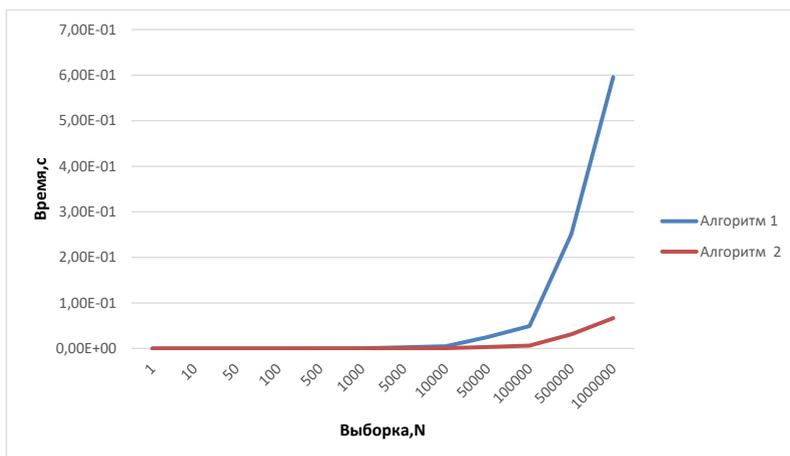


Рис. 5. Графики зависимостей времени работ программ от выборки N

Из рис. 5 видно, что при малых значениях объема выборки время работы программ совпадает, но при значениях больше 10 000 оно начинает расходиться и алгоритм 2 работает значительно быстрее, т. е. при моделировании случайных величин следует использовать алгоритм функции распределения закона Симпсона методом обратной функции.

Библиографический список

1. Studopedia.ru: сайт. URL: https://studopedia.ru/9_203977_treugolnyy-zakon-raspredeleniya-zakon-simpsona.html (дата обращения: 26.01.2021).
2. Вентцель Е. С. Теория вероятностей: учебник. 2-е изд., стер. М.: ЮСТИЦИЯ, 2018. 575 с.

УДК 629.039.58

А. М. Горбунова

студент кафедры проблемно-ориентированных вычислительных комплексов

Е. П. Виногорабова – старший преподаватель – научный руководитель

НЕЗАВИСИМАЯ МЕХАНИЧЕСКАЯ СИСТЕМА БЕЗОПАСНОСТИ АЭС

Сегодня во многих странах мира существуют АЭС. В истории отмечены случаи, когда из-за аварий на станциях погибло большое количество людей, система безопасности не обеспечивала полной надежности. Из этого можно сделать вывод, что существующая система требует доработок.

В основе предложенной работы лежит применение физических (механических) законов, которые обеспечивают независимую систему безопасности для атомных электростанций.

Принципы системы безопасности в аварийных случаях: 1) сведение ядерных реакций к минимуму; 2) охлаждение реактора; 3) обеспечение возможности восстановительных работ реактора или создание «могильника».

Цель работы: 1) повысить безопасность эксплуатации АЭС в случае аварий; 2) уменьшить затраты на восстановительные работы; 3) максимально обеспечить экологичность (уменьшить количество выбросов при аварии).

Задачи: 1) разработать схему аварийной остановки атомного реактора; 2) разработать схемы узлов безопасности.

Для современного уровня развития общества характерен рост потребления энергии. Ее природные источники когда-то закончатся, возобновляемые – имеют малое КПД. Атомная энергетика будет развиваться, но система безопасности не обеспечивает 100% гарантий. Поэтому происходят такие аварии, как в Чернобыле и Фукусиме.

Идея заключается в том, что в случае аварийной ситуации (отключение основного и резервного электропитания) реактор опускается в шахту. Необходимо обеспечить остановку реактора, его охлаждение, управление при восстановлении электроэнергии и проведение впоследствии восстановительных работ или консервации реактора [1].

Рассмотрим рис. 1. В верхней части шахты установлен реактор, который прикреплен к стенке шахты аварийными замками (3). В случае повышения давления замки открываются и реактор опускается в шахту. В начале движения блока реактора перекрываются каналы подачи воды (1) и выхода пара (2). Вода перенаправляется в аварийный канал подачи.

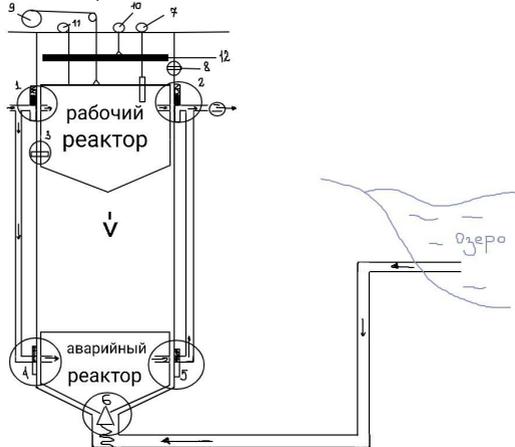


Рис. 1. Общая схема реактора

Условные обозначения на рис. 1: 1 – узел подачи воды в реактор, 2 – узел выхода пара на генератор, 3 – аварийный замок реактора, 4 – аварийный узел подачи воды, 5 – аварийный узел выхода пара, 6 – аварийный клапан подачи воды в шахту, 7 – механизм управления стержнями, 8 – аварийный замок плиты, 9 – механизм подъема реактора, 10 – механизм подъема плиты, 11 – кабели управления и контроля реактора, 12 – бетонная плита.

Реактор, опустившись на дно шахты, открывает аварийный узел подачи воды (4), аварийный узел выхода пара (5) и клапан (6) подачи воды из водохранилища. Шахта заполняется водой, что обеспечивает дополнительное охлаждение реактора.

При опускании реактора происходит:

- открытие стопора плиты (8) синхронно с аварийным замком реактора (3);
- при опускании плиты (12) утапливание стержней, что обеспечивает минимальную работу реактора;
- разматывание аварийного троса (7) стержней;
- разматывание тросов механизма (9) аварийного подъема реактора;
- разматывание электрических кабелей управления и контроля реактора.

Перед началом восстановительных работ проводится анализ состояния реактора и принимается решение о его подъеме/захоронении.

Стопор атомного реактора

Стопор (рис. 2) удерживает реактор в верхней части шахты. Подпружиненный стержень 1 одной частью входит в стенку шахты, другой – в корпус реактора.

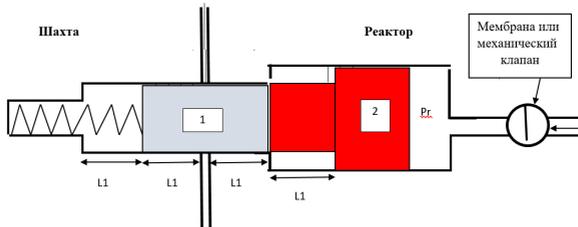


Рис. 2. Стопор атомного реактора (штатная ситуация)

В случае аварийной ситуации и превышения давления в реакторе > 160 атмосфер срабатывает мембрана или механический клапан. Давление давит на поршень 2, он выдавливает стержень 1. Они занимают положение (рис. 3), при котором реактор не удерживается, следовательно, он опускается вниз шахты.

Мембрана или механический клапан предназначен для того, чтобы при изменении давления в реакторе, стопор не двигался, и не происходил его износ.

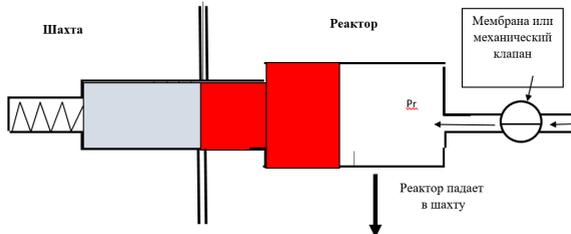


Рис. 3. Стопор атомного реактора (аварийная ситуация)

Узел подачи воды в реактор

В рабочем положении (рис. 4) вода 1-го контура поступает в реактор через узел подачи 1. Клапан обратного давления реактора открыт, вода поступает в реактор, и пар уходит в теплообменник 2-го контура. Заслонка удерживается подпружиненным стопором заслонки, что не дает ей перекрыть подачу воды. Клапан обратного давления открыт. При движении реактора в шахту опускается заслонка, стопор заслонки утапливается, и канал подачи воды перекрывается. Вода направляется по аварийному каналу. Клапан обратного реактора закрывается за счет внутреннего давления < 160 атмосфер (рис. 5) [2].

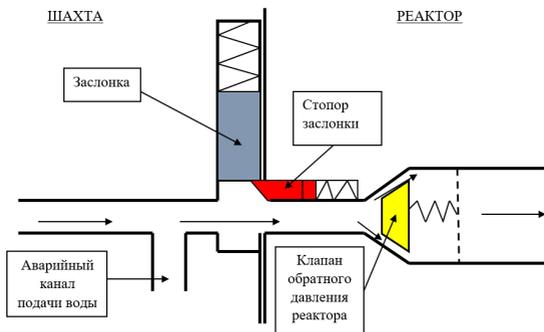


Рис. 4. Узел подачи воды (штатная ситуация)

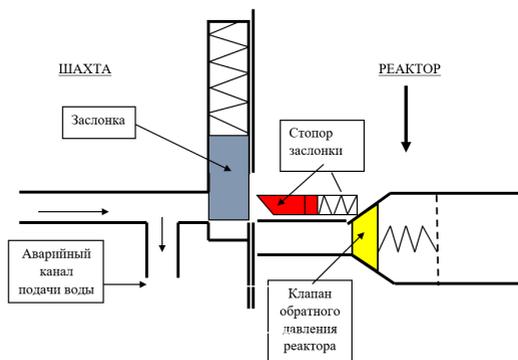


Рис. 5. Узел подачи воды (аварийная ситуация) [2]

Аварийный узел подачи воды в реактор

При нормальной работе реактора заслонка перекрывает подачу воды (рис. 6). При опускании реактора (рис. 7) «стопор заслонки» опускает заслонку и открывает канал подачи воды в реактор.

Аналогично устроен узел вывода пара из реактора.

В таком положении реактор практически не работает. Стержни замедляют его работу, циркуляция 1-го контура осуществляется в нормальном (аварийном) режиме.

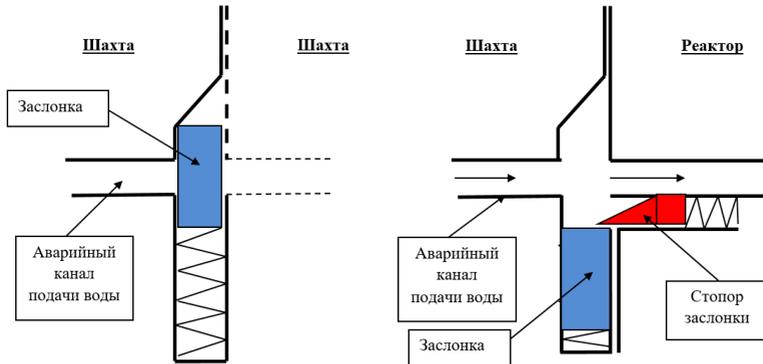


Рис. 6. Штатная ситуация

Рис. 7. Аварийная ситуация

Если использовать данную концепцию, то мы можем защитить экологию и спасти людей, уменьшить затраты (не надо строить «саркофаг», так как при взрыве шахту засыпают свинцом и заливают бетоном), и при незначительных разрушениях можно восстановить реактор или утилизировать ядерное топливо.

В работе рассмотрен основной принцип построения системы безопасности АЭС. Здесь нужно разработать и применить еще множество механизмов и узлов. Например, к «стопору механизма заслонки» нужно механически соединить поворотный клапан, который обеспечит герметичность реактора на время опускания в шахту; установить дополнительные кремневые стержни на реактор, используемые только в случае аварии; применить конусные уплотнительные механизмы между реактором и шахтой в «узлах подачи воды» и «узлах выхода пара» и т. д.

Рассмотрена система безопасности АЭС, построенная на механических законах физики. Это позволяет минимизировать отрицательные последствия. Но в случае аварии необходима подробная информация о состоянии срабатывания систем безопасности и само состояние реактора, это большой объем информации. Значит, надо иметь электронный блок сбора информации, запитанный независимым источником питания (АКБ). Собранная информация должна постоянно передаваться через спутник в общий центр контроля.

Помимо штатной информации с реактора (давление, температура и т. д.), поступающей по проводам, мы должны получать сведения:

- о положении и состоянии графитовых стержней;
- опускании аварийных графитовых стержней;
- срабатывании аварийных заслонок;
- подаче воды в реактор по аварийному каналу;
- выходе пара (воды) из реактора на охлаждение;
- наличии электропитания основного или резервного;
- открытии клапана затопления шахты и т. д.

Имея данную информацию, мы можем более оперативно принять решение о восстановлении или «захоронении» реактора.

Библиографический список

1. История атомной энергетики Советского Союза и России / Рос. науч. центр «Курчатовский ин-т»; под ред. В. А. Сидоренко. 2-е изд. М.: ИздАТ, 2009. Вып. 2: История ВВЭР. 427 с.
2. Аварии и инциденты на атомных электростанциях / под ред. С. П. Соловьева. Обнинск: ИАТЭ, 1992. 298 с.

УДК 692.6

А. М. Горбунова

студент кафедры проблемно-ориентированных вычислительных комплексов

Е. П. Виноградова – старший преподаватель – научный руководитель

**ОПТИМИЗАЦИЯ ЭНЕРГОПОТРЕБЛЕНИЯ ЛИФТОВ И ЭСКАЛАТОРОВ.
СИСТЕМА БЕЗОПАСНОСТИ**

Для современного уровня развития общества характерен быстрый рост потребления энергии. Однако увеличение ее генерации с использованием традиционных источников ведет к их истощению и нередко к серьезным экологическим проблемам. В проекте рассмотрена система генерации энергии, использующая работу силы тяжести при перемещении кабины лифта и спуске/поднятии эскалатора.

Основная идея – использование работы силы тяжести при движении эскалатора и кабины лифта вниз с выработкой электроэнергии, которая накапливается на аккумуляторе, являющемся частью противовеса лифта. Выработанная энергия может быть использована для аварийного освещения кабины лифта и систем безопасности, в случае эскалатора – для бесконтактной зарядки мобильных устройств на поручнях. Главная составляющая – сила тяжести. Масса всех людей, находящихся в лифте, опускает кабину. Для достижения поставленной цели создана условная модель, имитирующая принцип работы лифта, на примере которой рассмотрена идея оптимизации энергопотребления.

Модель работает от аккумуляторных батарей (источник питания). При включении трехпозиционного двойного выключателя ток подается на привод, состоящий из редуктора, двигателя и приводного вала. На валу закреплен трос (в случае модели нить), перемещающий кабину лифта. В шахте расположены нижний и верхний концевые выключатели. При движении лифта вверх кабину поднимает привод, который работает благодаря аккумуляторным батареям в штатном режиме. При переключении трехпозиционного двойного выключателя в противоположную позицию кабина перемещается вниз, в это время аккумулятор не успевает зарядиться, поэтому будет наблюдаться плавное включение и выключение светодиода, а не постоянное его свечение. В цепи вырабатывается переменный ток, который не подходит для непрерывной работы светодиода, поэтому в цепь включен диодный мост из четырех диодов. Он преобразует переменный ток в постоянный. Новое в данной цепи – внедрение генератора, который механически соединен с двигателем и к которому последовательно присоединен диодный мост, к нему крепится конденсатор для сглаживания пиков напряжения, далее аккумулятор, являющийся накопителем электроэнергии. Благодаря ему питается потребитель, в модели – светодиод.

На рис. 1 представлена электрическая цепь модели лифта. «Усовершенствованный» лифт будет иметь такую же схему, т. е. в уже существующую цепь (изображена в правой части цепи (до механического соединения), является условной) надо добавить механическое соединение генератора с двигателем, далее будут присоединены диодный мост, конденсатор, аккумулятор и потребитель [1].

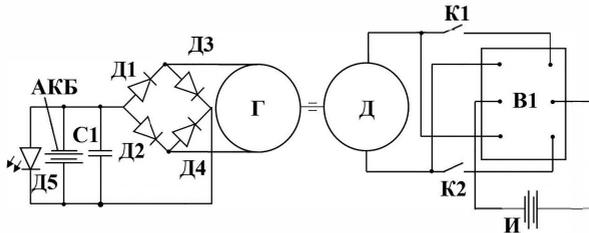


Рис. 1. Электрическая цепь:

В1 – трехпозиционный двойной переключатель, С1 – электрический конденсатор, Д1–4 – диодный мост, Д5 – светодиод (потребитель), И – источник питания, АКБ – накопительный аккумулятор, Г – генератор, Д – двигатель, К1–К2 – концевые выключатели

Механическая цепь состоит из последовательно соединенных деталей. В уже существующей цепи используются штатный привод лифта и приводной вал.

Идея – внедрение новых запчастей, благодаря которым можно вырабатывать энергию при спуске. Энергия будет вырабатываться по закону силы тяжести, поэтому в цепь будет внедрен динамометр, на основе его измерений будут включаться/выключаться муфты приводов лифта или генератора. Если в кабину зайдет достаточное количество человек (чтобы их масса смогла перемещать кабину вниз под силой тяжести), то при опускании одна муфта разблокирует левую часть цепи, генератор начнет вращаться и будет вырабатываться электроэнергия, в этот момент электрическая муфта заблокирует свою часть цепи, т. е. муфты работают попеременно, включая генератор или двигатель. После одной муфты стоит тахометр, который управляет оборотами приводного вала через тормозную систему (обеспечивая скорость лифта 0,5 м/с). Далее расположен повышающий редуктор и потом генератор. Выработанная энергия поступает на аккумулятор, который является накопителем энергии.

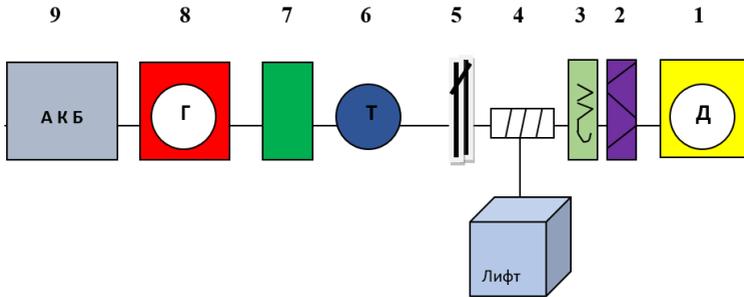


Рис. 2. Механическая цепь лифта:

- 1 – штатный привод лифта, 2 – электрическая муфта, 3 – динамометр, 4 – приводной вал, 5 – одна муфта, 6 – тахометр, 7 – редуктор, 8 – генератор, 9 – аккумулятор

Система включает независимый источник энергии, обеспечивающий аварийное освещение кабины лифта, что существенно снижает психологическую нагрузку на застрявших в кабине людей, и систему безопасного спуска до ближайшего этажа, что позволяет осуществить самостоятельный выход.

Рассмотрим систему безопасности. На рис. 3, а изображена схема системы безопасности. На каждом этаже на четырех направляющих установлены механические стопора 2. Данный механизм удерживается замком 1, который соединен с электромагнитом 1. В случае аварийной ситуации или отключения электроэнергии лифт под своей массой и массой пассажиров начинает падать. В это время подается команда на электромагнит 1, и на всех этажах механически открываются замки 2. Питание на срабатывание электромагнита 1 берется с АКБ. Энергия АКБ тратится на тормозную систему, чтобы опустить лифт с постоянной скоростью (0,5 м/с) до ближайшего нижнего этажа. Кабина, встав на механический стопор, включает концевой выключатель (штатный) на разблокировку дверей лифта и площадки этажа. Именно это позволяет открыть двери лифта и эвакуировать пассажиров, не дожидаясь спасателей или ремонтных бригад (рис. 3, б).

В случае попадания стопорного механизма между верхним и нижним роликовыми башмаками лифта стопор упирается в стенку кабины. У верхнего роликового башмака установлен «отбойник», который утапливает стопор и пропускает верхний башмак. Это позволяет не застрять кабине лифта между этажами и остановиться напротив ближайшего нижнего этажа.

По окончании аварийной ситуации и восстановительных работ электромагнитом 2 с диспетчерского пункта система безопасности приводится в дежурное состояние.

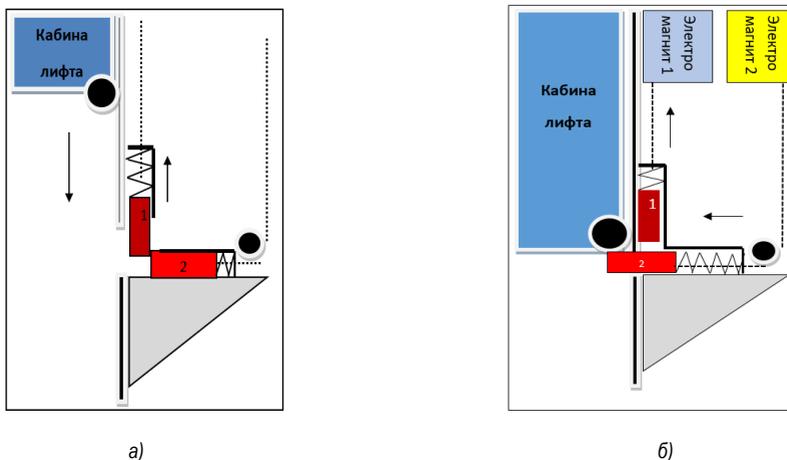


Рис. 3. Спуск (а) и остановка (б) лифта

В случае эскалатора энергия вырабатывается только при движении вниз под силой тяжести пассажиров.

Идея – внедрение односторонней муфты (принцип работы идентичен работе муфты, которая находится в цепи лифта) (рис. 4). При ее разблокировке начинает вращаться вал, на генераторе будет вырабатываться энергия, которая в дальнейшем поступает на накопитель – аккумулятор. В механизме управления появляется динамометр (такой же, как в лифте).

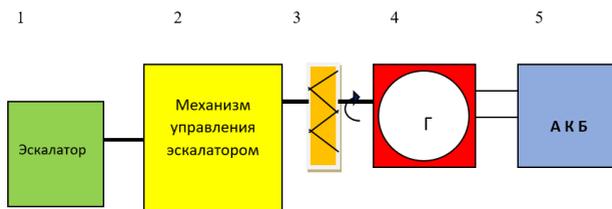


Рис. 4. Механическая цепь эскалатора:

1 – лента эскалатора, 2 – механизм управления, 3 – односторонняя муфта, 4 – генератор, 5 – АКБ [2]

При положительных значениях динамометра эскалатор будет работать в штатном режиме. При увеличении массы людей нагрузка на двигатель уменьшается, динамометр показывает отрицательное значение, выдается команда на муфты, отключение двигателя и включение генератора.

Генератор подключается механически к силовому двигателю эскалатора. Когда масса людей начинает «разгонять» эскалатор, то включается штатная система торможения (1 м/с), в этот момент генератор и будет забирать часть энергии и накапливать ее. Нагрузка на тормозную систему уменьшится, что приводит к ее меньшему износу.

Алгоритм работы системы в случае лифта

Мы рассмотрели два вопроса по модернизации лифтов – повышение системы безопасности и экономии электроэнергии. Но в свете современных требований необходимо получать максимум обрат-

ной информации от различных систем и желательно автоматического управления ими по заданным алгоритмам [2].

Зная расход накопленной энергии в случае аварийной ситуации (из расчетов на определенные лифты) на открывание дверей и аварийное освещение мы можем определить алгоритм. Например, АКБ должна иметь 20 % заряда для обеспечения аварийного электропитания. Значит, при зарядке АКБ до 95 % мы можем накопленную энергию пустить на освещение кабины лифта. При разряде до 25 % АКБ вновь включается на подзарядку.

Также мы можем установить датчики контроля работы стопорных механизмов, отдельный для каждого механизма. Это позволит ремонтной бригаде не искать по этажам неисправный стопор.

Устанавливаются фотодатчики контроля тросов лифта.

Вся информация поступает на пульт диспетчера, и при любом «сбое» он принимает решение по устранению недостатков.

Алгоритм работы системы в случае эскалатора

Выработка энергии на эскалаторах метрополитена может быть большой. Поэтому алгоритм ее использования можно заложить не только на освещение, но и на работу турникетов. Система будет работать автоматически без вмешательства человека, но под его контролем.

Мы можем также установить датчики контроля состояния ленты эскалатора. Данные динамометра, данные с турникетов и вся информация поступают на центральный пост управления и контроля, что позволит автоматизировать частоту прохождения поездов, обеспечив оптимальное количество людей на станциях.

Библиографический список

1. Об утверждении технического регламента о безопасности лифтов: постановление Правительства РФ от 2 окт. 2009 г. № 782. URL: <https://www.garant.ru/products/ipo/prime/doc/12070101/> (дата обращения: 08.02.2021).
2. Павлов Н. Г. Лифты и подъемники. Основы конструирования и расчета. М.: Машиностроение, 1965. 203 с.

УДК 621.391.82

И. Р. Гордеев

студент кафедры проблемно-ориентированных вычислительных комплексов

Е. К. Григорьев – ассистент – научный руководитель

ИССЛЕДОВАНИЕ И АНАЛИЗ КОРРЕЛЯЦИОННЫХ ХАРАКТЕРИСТИК КОДОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ МАКСИМАЛЬНОЙ ДЛИНЫ

В настоящее время наиболее серьезные проблемы радиолокации, передачи информации и связи решаются применением сложных широкополосных сигналов. Особенно активно применяются дискретно-кодированные сигналы (ДКС), в которых закон изменения манипулируемого параметра задается дискретно-кодированными последовательностями, определяющими свойства ДКС [1]. Поэтому поиск новых кодовых последовательностей в целом и с хорошими корреляционными свойствами в частности актуален.

В последнее время активно развивается теория квазиортогональных матриц [3–5], к которым относятся матрицы Мерсенна, Рагхавараро и др. Матрицы Мерсенна существуют на всех порядках $N = 4 * n - 1$, где n – натуральное число. Указанные матрицы, являющиеся ядром матриц Адамара и обобщающие их, могут иметь как циклическую, так и симметричную конструкции. В работе исследуется возможность использования строк таких матриц в качестве кодовых последовательностей.

В первую очередь необходимо дать определение ортогональной и квазиортогональной матрицы.

Ортогональная матрица – матрица A , удовлетворяющая условию

$$A^{-1} * A = A^T * A = I,$$

где A^{-1} – обратная матрица, A^T – транспонированная матрица, I – единичная матрица.

Квазиортогональной матрицей [3] будем называть квадратную матрицу A порядка n с приведенными к единице максимумами модулей элементов каждого из столбцов, которая удовлетворяет условию

$$A^T * A = \omega * I,$$

где ω – вес матрицы.

В качестве примера на рис. 1, приведен портрет квазиортогональной матрицы Мерсенна 15-го порядка. Здесь белый квадрат соответствует элементу «1», а черный – «-b», где значение $-b$ определяется из теории квазиортогональных матриц [3–5].

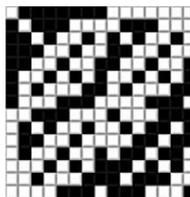


Рис. 1. Портрет квазиортогональной матрицы Мерсенна 15-го порядка

Модификация кодовых последовательностей

Поскольку в работе исследуются корреляционные характеристики кодовых последовательностей с алфавитом $(1, -b)$, рассмотрим процедуру преобразования исходной последовательности с алфавитом $(1, -1)$ к алфавиту $(1, -b)$. Будем называть ее модификацией, структурная схема представлена на рис. 2.



Рис. 2. Процедура генерации последовательности с алфавитом (1, -b)

Исследование проводилось следующим образом.

1. Генерировалась матрица, где в качестве первой строки выбиралась одна из последовательностей: последовательность Якоби, Лежандра и М-последовательностей.
2. Строки матрицы изучались на предмет максимального уровня бокового лепестка автокорреляционной функции (АКФ).
3. Из всей матрицы выбиралась строка с наименьшим уровнем бокового лепестка АКФ.

Обзор исследуемых последовательностей и соответствующие результаты исследования

Сигналы, сформированные на основе последовательностей максимальной длины или М-последовательностей занимают особое место. В силу своих свойств дискретные сигналы на основе М-последовательностей чрезвычайно популярны в современных информационных системах.

В табл. 1 представлены результаты исследования последовательностей с алфавитом (1,-1) и (1,-b) на предмет максимального уровня бокового лепестка АКФ. В качестве примера на рис. 3, а приведена АКФ кодовой последовательности с алфавитом (1,-b), которая превзошла алфавит (1,-1).

Таблица 1

Результаты исследования АКФ М-последовательностей

Порядок	Значение уровня бокового лепестка АКФ	
	Алфавит (1,-1)	Алфавит (1,-b)
15	0,133	0,11
31	0,096 8	0,108 6
63	0,079 4	0,080 2
127	0,063	0,070 2
255	0,047 1	0,050 3
511	0,039 1	0,038 1

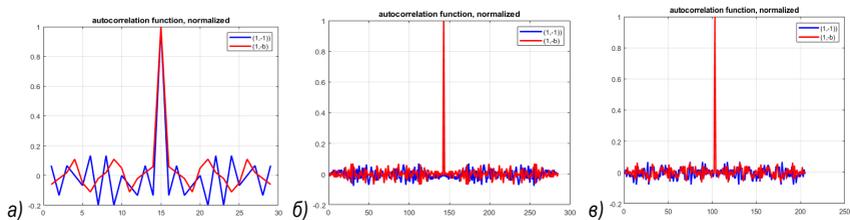


Рис. 3. Автокорреляционные функции исследуемых последовательностей для двух алфавитов

В то же время множество длин, при которых данные последовательности существуют, достаточно разрежено [6]. По этой причине целесообразно исследовать еще некоторые классы бинарных минимаксных последовательностей, например последовательность символов Якоби, формирующуюся, соответственно, на основе символов Якоби [2].

В табл. 2 представлены результаты исследования последовательностей Якоби с алфавитом (1,-1) и (1,-b) на предмет максимального уровня бокового лепестка АКФ. В качестве примера на рис. 3, б приведена АКФ кодовой последовательности с алфавитом (1,-b), которая превзошла алфавит (1,-1).

Таблица 2

Результаты исследования последовательностей Якоби

Порядок	Значение уровня бокового лепестка АКФ	
	Алфавит (1,-1)	Алфавит (1,-b)
15	0,133 3	0,138 9
35	0,085 7	0,106 6
143	0,069 9	0,068 6
323	0,046 4	0,049 5

Последовательность символов Лежандра, или последовательность квадратичных вычетов, формируется на основе символов Лежандра [2]. В табл. 3 представлены результаты ее исследования с алфавитом (1,-1) и (1,-b) на предмет максимального уровня бокового лепестка АКФ. В качестве примера на рис. 3, в приведена АКФ кодовой последовательности с алфавитом (1,-b), которая превзошла алфавит (1,-1).

Таблица 3

Результаты исследования последовательностей Лежандра

Порядок	Значение уровня бокового лепестка АКФ	
	Алфавит (1,-1)	Алфавит (1,-b)
7	0	0,082 4
15	0,133 3	0,209 1
31	0,096 8	0,916 3
43	0,069 8	0,091 4
51	0,235 3	0,297 1
71	0,056 3	0,073 3

Порядок	Значение уровня бокового лепестка АКФ	
	Алфавит (1,-1)	Алфавит (1,-b)
79	0,063 3	0,066 2
91	0,329 7	0,378 1
103	0,068	0,063 3
107	0,560 8	0,058 3
151	0,046 4	0,050 2
159	0,308 2	0,343 6
163	0,049 1	0,053 9

Заключение

Рассмотрен механизм формирования модифицированных кодовых последовательностей. Проведено их изучение на предмет максимального уровня бокового лепестка АКФ. Исследование показало превосходство алфавита (1,-b) над (1,-1) в следующих случаях: для M-последовательности на порядках 15 и 511, для последовательности Якоби – на 143, для последовательности Лейбнера – на 31 и 103.

На некоторых порядках строки квазиортогональных матриц на основе модифицированных кодовых последовательностей имеют хорошие автокорреляционные свойства, значит, можно говорить о перспективности их применения в системах локации, навигации и обмена информации.

Библиографический список

1. Грантмахер В. Е., Быстров Н. Е., Чеботарев Д. В. Шумоподобные сигналы. Анализ, синтез, обработка. СПб.: Наука и техника, 2005. 400 с.
2. Варакин Л. Е. Системы связи с шумоподобными сигналами. М.: Радио и связь, 1985. 384 с.
3. Сергеев А. М. Методы преобразования изображений и кодирования сигналов в каналах распределенных систем на основе использования специальных квазиортогональных матриц: дис. ... канд. техн. наук. СПб, 2019. 153 с.
4. Балонин Н. А., Сергеев М. Б. Матрицы локального максимума детерминанта // Информ.-управляющие системы. 2014. № 1 (68). С. 2–15.
5. Balonin N., Sergeev M. Quasi-Orthogonal Local Maximum Determinant Matrices // Appl. Math. Sciences. 2015. № 8. С. 285–293.

УДК 004.921

А. В. Душеба

студент кафедры проблемно-ориентированных вычислительных комплексов

Е. П. Виноградова – старший преподаватель – научный руководитель

ПОСТРОЕНИЕ ТРАЕКТОРИИ ДВИЖЕНИЯ НЕБЕСНЫХ ТЕЛ

Описание задачи

Гравитационная задача N -тел описывает поведение системы небесных тел, между которыми возникает гравитационное взаимодействие.

Пусть даны N -тел с заданными начальными положениями x_i и скоростями $v_i, 1 \leq i \leq N$. Сила F_{ij} , возникающая в результате гравитационного взаимодействия тел i и j , задается формулой

$$F_{ij} = Gm_i m_j \frac{(x_j - x_i)}{|x_j - x_i|^3},$$

где m_i, m_j – массы, x_i, x_j – координаты тел i, j соответственно, G – гравитационная постоянная [1].

Чтобы найти положения тела x_i и его скорость v_i , необходимо знать его ускорение $a_i = \frac{F_i}{m_i}$.

Подставив F_i , получим следующий вид уравнения ускорения:

$$a_i = G \sum_{j \neq i}^N m_j \frac{x_j - x_i}{(x_j - x_i)^3}.$$

Получаем систему дифференциальных уравнений:

$$\begin{cases} \frac{dx_i}{dt} = v_i \\ \frac{dv_i}{dt} = G \sum_{j \neq i}^N m_j \frac{x_j - x_i}{(x_j - x_i)^3} \end{cases}.$$

Случай, когда $N = 1$, не рассматривается. Для $N = 2$ существует решение в общем виде, при $N \geq 3$ решения в общем виде быть не может [2].

Описание алгоритма

Алгоритм производит вычисление каждой силы F_{ij} , действующей на тело i со стороны тела j путем перебора всех возможных комбинаций. Сначала для всех тел вызывается метод UpdateVelocity, который рассчитывает ускорение для каждого тела и применяет его к текущей скорости.

```
def UpdateVelocity(planets, planet, delta_time):
```

```
  for p in planets:
```

```
    if (p != planet):
```

```
      deltaDist = vec2(p.pos.x - planet.pos.x, p.pos.y - planet.pos.y)
```

```
      sqrDst = deltaDist.x ** 2 + deltaDist.y ** 2
```

```
      forceDir = vec2(deltaDist.x/(sqrDst ** 0.5), deltaDist.y/(sqrDst ** 0.5))
```

```
      acceleration = vec2(G * forceDir.x * p.mass / sqrDst, G * forceDir.y * p.mass / sqrDst)
```

```
      planet.velocity += vec2(acceleration.x * delta_time, acceleration.y * delta_time)
```

Так как полученная система уравнений решается с помощью перебора, временная сложность работы такого алгоритма составляет $O(N^2)$. Однако существуют более эффективные способы реали-

зации данного алгоритма, позволяющие распараллелить вычисления, во много раз увеличивая его скорость работы. Это позволит производить вычисления для большого количества тел в реальном времени и визуализировать, например, столкновения галактик.

Метод `UpdatePosition` изменяет положения тела согласно его скорости. Важная деталь, на которую следует обратить внимание в данном алгоритме, это то, что для достижения большей точности результата важно сначала рассчитать и применить ускорение для каждого из тел и лишь затем обновлять их положение.

```
def UpdatePosition (planet, delta_time):  
    planet.pos.x += planet.velocity.x * delta_time  
    planet.pos.y += planet.velocity.y * delta_time
```

Визуализация задачи

Для визуализации была использована библиотека `Arcade`. На рис. 1 представлена периодическая орбита для трех тел равной массы.

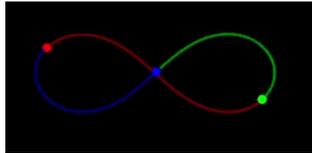


Рис. 1. Стабильная орбита для трех тел

На рис. 2 представлена траектория движения четырех тел с фокусировкой на тело в центре, на рис. 3 – эволюция системы трех тел со случайно заданными начальными параметрами.

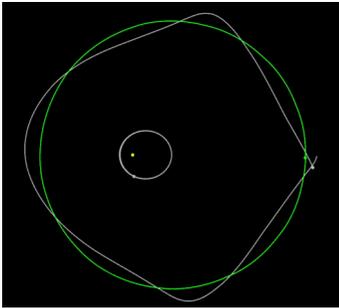


Рис. 2. Фокусировка на желтое тело

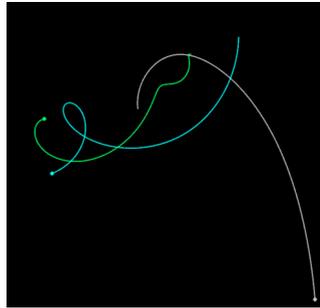


Рис. 3. Случайные начальные параметры

Заключение

Визуализация задачи N тел позволяет оценить эволюцию системы с течением времени. Представленный алгоритм может быть оптимизирован при помощи более эффективных способов решения дифференциальных уравнений, что позволит производить вычисления для большого количества тел в реальном времени.

Библиографический список

1. Moore C. Braids in Classical Dynamics // *Phys. Rev. Letters*. 1992. Vol. 70. P. 3675–3679
2. Alain Chenciner, Richard Montgomery A remarkable periodic solution of the three-body problem in the case of equal masses. *Annals of Mathematics* 152, 2000. С. 881–901

УДК 004.921

А. В. Душеба

студент кафедры проблемно-ориентированных вычислительных комплексов

Е. П. Виноградова – старший преподаватель – научный руководитель

ПРОЕКТИРОВАНИЕ МОДЕЛЕЙ ТРЕХМЕРНЫХ ОБЪЕКТОВ С ПОМОЩЬЮ АЛГОРИТМА RAY MARCHING

Описание задачи

Задача алгоритма объемного рендеринга – получение проекции изображения на экране из трехмерного дискретного набора данных.

Постановка задачи

Пусть [1] задана сцена, в которой все объекты представлены в виде формул, позволяющих найти расстояние до этого объекта. Расстояние d из точки p с координатами (x, y, z) до сферы S с координатами (x_s, y_s, z_s) и радиусом r определяется по формуле

$$d = \sqrt{(x^2 - x_s^2) + (y^2 - y_s^2) + (z^2 - z_s^2)} - r.$$

Реализация на языке GLSL:

```
float sdSphere(vec3 s, vec3 p float r) {
    return length(p - s) - r;
}
```

Для каждого пикселя на экране выпустим луч, при попадании которого в объект пиксель окрашивается в цвет объекта (рис. 1). Расстояние, которое проходит луч за один шаг, – это расстояние для ближайшего объекта в сцене (рис. 2). Это позволяет сократить количество шагов и добиться лучшей производительности.

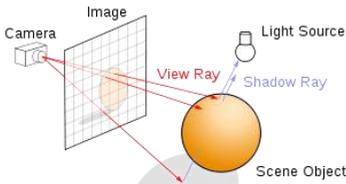


Рис. 1. Проекция на экран

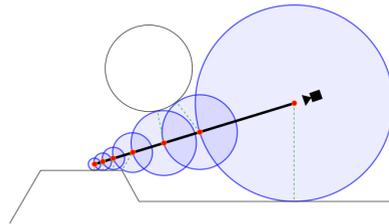


Рис. 2. Ход луча

Будем считать, что луч попал в объект, если расстояние d до ближайшего объекта будет меньше некоторой константы SURF_DIST. Если же луч прошел заданное максимальное количество шагов MAX_STEPS или его длина превысила максимальное значение MAX_DIST, будем считать, что луч ни во что не попал [1].

Реализация метода RayMarch:

```
float RayMarch(vec3 ro, vec3 rd) {
    float dO=0;
    for(int i=0; i<MAX_STEPS; i++) {
        vec3 p = ro + rd*dO;
        float dS = GetDist(p);
        dO += dS;
        if(dO>MAX_DIST || dS<SURF_DIST) break;
    }
}
```

```

    }
    return dO;
}

```

Для того чтобы добавить тени, необходимо:

- 1) найти скалярное произведение между направлением к источнику света и нормалью к поверхности;
- 2) проверить, не попадает ли луч в другой объект по пути к источнику света [2].

Чтобы найти угол между направлением к источнику света \vec{l} и нормалью к поверхности \vec{n} в точке p , необходимо вычислить градиент ∇p .

$$\nabla p = \left(\frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}, \frac{\partial p}{\partial z} \right).$$

Для этого достаточно рассмотреть три точки в окрестности ϵ , тогда

$$\vec{n} = \begin{bmatrix} p(x+\epsilon, y, z) - p(x+\epsilon, y, z) \\ p(x, y+\epsilon, z) - p(x, y+\epsilon, z) \\ p(x, y, z+\epsilon) - p(x, y, z+\epsilon) \end{bmatrix}.$$

Реализация на языке GLSL:

```

vec3 GetNormal(vec3 p) {
    float d = GetDist(p);
    vec2 e = vec2(0.01, 0);
    vec3 n = d - vec3(
        GetDist(p-e.xxy),
        GetDist(p-e.yxy),
        GetDist(p-e.yyx));

    return normalize(n);
}

```

Далее найдем скалярное произведение между векторами \vec{l} , \vec{n} и проверим, не попал ли световой луч в другой объект [3]:

```

float GetLight(vec3 p, vec3 lightPos) {
    vec3 l = normalize(lightPos-p);
    vec3 n = GetNormal(p);

    float dif = clamp(dot(n, l), 0.0, 1.0);
    float d = RayMarch(p+n*SURF_DIST*2.0, l);
    if(d<length(lightPos-p)) dif *= 0.1;

    return dif;
}

```

Выпустим луч для каждого пикселя на экране:

```

void mainImage( out vec4 fragColor, in vec2 fragCoord )
{
    vec2 uv = (fragCoord-0.5*iResolution.xy)/iResolution.y;
    vec3 lightPos = vec3(1, 5, 1);

    vec3 ro = vec3(1, 2, -5);
    vec3 rd = normalize(vec3(uv.x, uv.y, 1.0));
    float d = RayMarch(ro, rd);
}

```

```
vec3 p = ro + rd * d;  
float dif = GetLight(p, lightPos);  
vec3 col = vec3(dif);  
  
fragColor = vec4(col, 1.0);  
}
```

Визуализация

Программа запускается на GPU в параллельном режиме, что позволяет получать изображение в реальном времени. На рис. 3 представлен рендер сцены, состоящей из сферы, плоскости и источника света.

Если в функции расстояния до сферы добавить модуль, то получится бесконечная сетка из сфер (рис. 4). Также можно брать максимум между двумя функциями расстояния и «вырезать» один объект из другого (рис. 5).

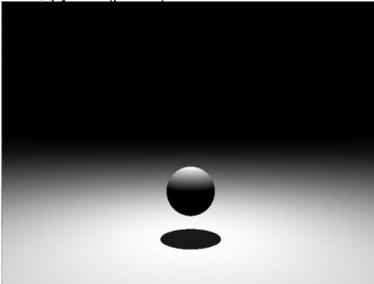


Рис. 3. Сфера и плоскость

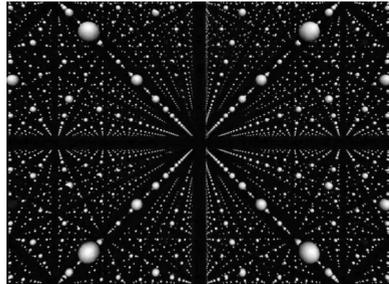


Рис. 4. Бесконечные сферы

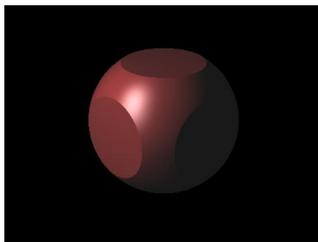


Рис. 5. Пересечение объектов

Заключение

Алгоритм Ray Marching позволяет получить проекцию трехмерных объектов, заданных при помощи уравнений расстояния в реальном времени.

Библиографический список

1. Jamie W. Ray Marching and Signed Distance Functions. URL: <http://jamie-wong.com/2016/07/15/ray-marching-signed-distance-functions/> (дата обращения: 03.05.2021).
2. Hart C. J. Sphere Tracing: A Geometric Method for the Antialiased Ray Tracing of Implicit Surfaces // The Visual Computer. 1995. Vol. 12. P. 527–545.
3. Hart C. J., Sandin D., Kauffman L. Ray Tracing Deterministic 3-D Fractals // ACM SIGGRAPH Computer Graphics. 1989. Vol. 23 (3). P. 289–296.

УДК 004.418

И. Ю. Класс

магистрант кафедры проблемно-ориентированных вычислительных комплексов

В. С. Павлов – доктор технических наук, профессор – научный руководитель

РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ МОНИТОРИНГА ТЕХНИЧЕСКОГО СОСТОЯНИЯ АВТОТРАНСПОРТНОГО СРЕДСТВА

Своевременная диагностика технического состояния тормозной системы автомобиля необходима для обеспечения безопасности во время движения. Диагностические операции проводят в сервисных центрах методами визуального контроля, ходовых испытаний или на специализированных стендах, которые сопряжены с программно-аппаратным комплексом по измерению и интерпретации состояния элементов тормозной системы. Плановую проверку автотранспортного средства проводят в соответствии с технической документацией завода-изготовителя. Однако при планировании передвижений необходимо самостоятельно оценивать уровень технического состояния элементов тормозной системы.

Цель работы: провести обоснование методических приемов автоматизированной диагностики автотранспортных средств в интерактивном режиме и разработать практический пример реализации такой диагностики для оценки технического состояния тормозной системы легкого автомобиля. Эта цель достигается за счет последовательного решения ряда задач, отражающих этапы разработки.

В настоящее время перед автомобилестроителями стоит одна из важнейших задач повышения технического уровня, безопасности и надежности автотранспортного средства и конкурентоспособности выпускаемой техники. Вследствие постоянного усложнения технических характеристик автотранспортного средства и предъявления жестких требований к его безопасности с помощью традиционных методов диагностики нельзя быстро оценить техническое состояние агрегатов, узлов автомобиля и одну из главных систем, которая имеет самый большой приоритет – тормозную [1]. Также все усложняется спецификой транспортного средства, условием ее эксплуатации, качеством технического обслуживания и проведенных ранее ремонтных работ. Для решения перечисленных проблем разработана интеллектуальная система мониторинга технического состояния автотранспортного средства, которая оперативно реагирует на поломки автомобиля и работу узлов и агрегатов в штатном режиме. Система может проводить текущее техническое обслуживание и на основе результатов диагностики исключить эксплуатацию неисправного автотранспортного средства, а также избавить водителя от преждевременных затрат на замену узлов и агрегатов, состояние которых может быть некорректно определено при визуальном осмотре оператором на станции технического обслуживания.

Рассмотрим интеллектуальную систему мониторинга технического состояния автотранспортного средства, основная задача которой заключается в диалоге между водителем и автомобилем (рис. 1).



Рис. 1. Структурная схема интеллектуальной системы мониторинга технического состояния автотранспортного средства

Ядром системы служит микроЭВМ, которая представляет собой совокупность нескольких интегральных схем с большой степенью интеграции (БИС), взаимодействующих между собой и с внешними устройствами через информационные каналы. МикроЭВМ содержит ОЗУ и ПЗУ. Различные постоянные и вспомогательные данные, стандартные программы хранятся в ПЗУ, в то время как ОЗУ содержат параметры и фрагменты программ, которые могут в процессе работы измениться. В нашем случае в ПЗУ хранится программа диагностирования автотранспортного средства. Одно из главных устройств в работе автомобиля – ЭБУ (электронный блок управления), который служит для сбора значений внешних параметров, их преобразования, обработки, формирования управляющих сигналов [2]. Например, получение необходимой информации для определения износа фрикционного слоя тормозной колодки происходит при помощи датчиков тормозного момента, датчика угловой скорости колеса (ABS), датчика износа тормозных колодок, датчика распределения тормозных усилий (EBD) [3]. С целью соединения микроЭВМ с ЭБУ автотранспортного средства используется диагностический сканер (OBD), предназначенный для предварительной фильтрации входных информационных сигналов и преобразования их в стандартную для микроЭВМ форму.

Срединяющим звеном между интеллектуальной системой мониторинга технического состояния автотранспортного средства и водителем является графический интерфейс ввода-вывода. Его цепи передают данные от микропроцессора к периферийным устройствам и обратно и, кроме того, преобразуют внешнюю информацию в форму, пригодную для обработки в микроЭВМ [4]. Данный графический интерфейс составлен на высоком языке программирования Python, на котором также работает микроЭВМ [5]. Источником питания всех узлов системы служит бортовая сеть автотранспортного средства, имеющая постоянное напряжение питания номиналом в 12В.

Уникальность системы также заключается в передаче и хранении всей информации об автомобиле и его техническом состоянии в текущий момент на сервере базы данных, реализованной на сервере MySQL [6].

Заключение

Предлагаемая интеллектуальная система мониторинга технического состояния автотранспортного средства является уникальным продуктом в сфере диагностики транспортного средства, так как впервые происходит переход с механического способа диагностики автотранспортного средства на интеллектуальную систему мониторинга.

Библиографический список

1. Место и роль диагностики в системе ТО и ремонта автомобилей. URL: <https://studfiles.net/preview/1790598/page/4/> (дата обращения: 01.05.2021).
2. Электронный блок управления двигателем: назначение и устройство. URL: <http://krutimotor.ru/bloki-upravleniya-dvigatелеm-vidy-ustrojstvo-remont> (дата обращения: 01.05.2021).
3. Как работает система EBD в автомобиле? URL: <https://seite1.ru/zapchasti/kak-rabotaet-sistema-ebd-v-avtomobile/> (дата обращения: 01.05.2021).
4. Графический интерфейс пользователя с применением микроконтроллеров Microchip. URL: <http://www.gamma.spb.ru/novosti-proizvoditelej/microchip/stati-microchip/79-graficheskij-interfejs-polzovatelya> (дата обращения: 01.05.2021).
5. Язык программирования Python. URL: <http://www.gamma.spb.ru/novosti-proizvoditelej/microchip/stati-microchip/79-graficheskij-interfejs-polzovatelya> (дата обращения: 01.05.2021).
6. Bringing MySQL to the web. URL: <http://www.phpmyadmin.net/> (дата обращения: 11.05.2021).

УДК 681.58; 796.08

С. А. Матренинский

студент кафедры проблемно-ориентированных вычислительных комплексов

Н. Н. Григорьева – старший преподаватель – научный руководитель

ЭЛЕКТРОННЫЕ УСТРОЙСТВА КАК ЧАСТЬ ЭКИПИРОВКИ ДЛЯ ВОЕННО-ТАКТИЧЕСКОЙ ИГРЫ

Лазертаг (бой с применением лазерной системы) – это военизированная игра, которая предполагает имитацию перестрелки между участниками в реальном пространстве с помощью лазерного луча. Игроки снаряжаются специальной формой в виде жилетов и повязок с датчиками и тагерами (лучевыми имитаторами боевого оружия). Идея лазерных боев изначально не была игровой, лучевые имитаторы применялись для тактической и огневой подготовки военных еще в начале 1960-х гг. Вместо лазерного использовалось и инфракрасное излучение, что повышало реалистичность применяемой системы имитации боя. Современная игра «Лазертаг» берет начало в США в 80-х гг. прошлого столетия. В настоящее время она достаточно популярна, чему способствует то, что физический контакт в ней отсутствует, чего нельзя сказать о пейнтболе или страйкболе.

Первые варианты снаряжения для игры Photon состояли из тагера, датчиков излучения, размещенных на голове и жилете игрока, и вычислительного устройства. Структура снаряжения представлена на рис. 1.

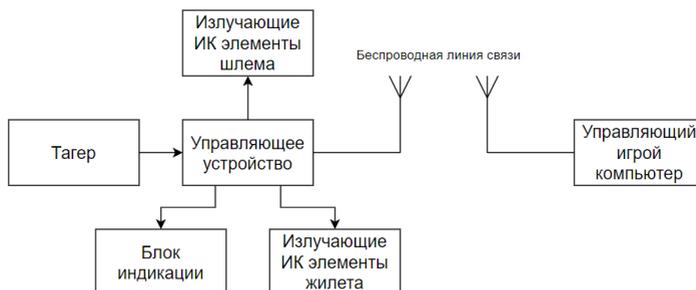


Рис. 1. Структурная схема снаряжения Photon

В снаряжении Photon использовалась обратная инфракрасная технология, т. е. жилет и шлем испускали ИК-луч, а тагер был приемником. На специальном жилете размещались батареи, устройство индикации попадания и вычислительное устройство для создания задержки стрельбы, фиксации попаданий и связи с управляющим игрой компьютером. Шлем и тагер были связаны с жилетом проводом [1].

В настоящее время преимущественно используют прямую инфракрасную технологию, где тагер выступает в качестве излучателя, а датчики расположены на теле игрока. Современная структура основного снаряжения не сильно отличается от классической, но появилось множество модификаций: для связи между устройствами используется Wi-Fi или Bluetooth, вычислительное устройство чаще расположено в тагере, добавлены возможности программной настройки режима игры, виброотклик и звуковой эффект от стрельбы и попадания [2, 3]. Обобщенная структурная схема современного снаряжения представлена на рис. 2.

Дополнительное оборудование делится на три категории: устройства управления игрой, дополнения игры и связи всех игроков.

Устройства управления игрой (УУ) позволяют менять режим игры, восстанавливать здоровье, боезапас и воскрешать игроков. Все УУ состоят из вычислительного устройства, блока индикации для вывода статистики об игроке или игре, блока взаимодействия и антенного модуля для связи [4].

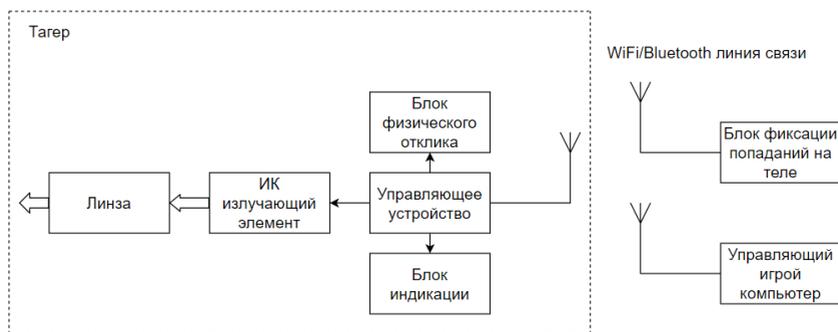


Рис. 2. Общая структурная схема современного снаряжения

Устройства дополнения игры (УД) расширяют возможности игроков или режимы игры. Дополнениями могут быть как элементы вооружения (гранаты, мины, бомбы), так и точки захвата или контроля. Структура устройств дополнения разнообразна, например у мины есть датчик присутствия игрока, а у точки захвата – нет. Но в составе почти всех УД есть ИК-излучатель, управляющее устройство, блок фиксации попаданий, индикации, настройки и антенного модуля (рис. 3) [5].

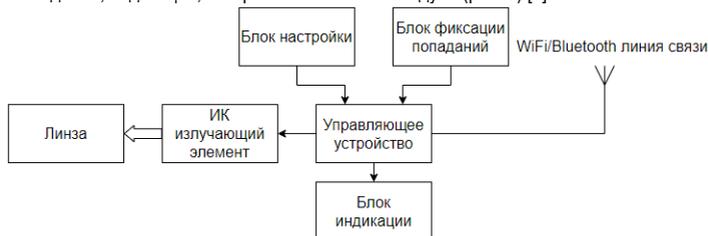


Рис. 3. Структурная схема устройства дополнения игры

Устройство связи всех игроков (УС) включает управляющий игровой компьютер, блок индикации, блок настройки и антенный модуль, подключенный к компьютеру [6]. Может использоваться стационарный компьютер как с предустановленным пользователем управляющим ПО, так и с готовым установленным и настроенным заводом ПО.

В последнее время в связи с доступностью микроконтроллеров все чаще игроки создают собственные устройства для игры. В данный момент разрабатывается дополнительное устройство, выступающее в роли стационарного наводимого тагера. С его помощью игрок сможет на сенсорном экране выбрать цель посредством нажатия, после чего двигатели направят на нее ИК-излучатель (рис. 4).

С помощью микроконтроллера вычисляются отклонения по горизонтальным и вертикальным осям между направлением на цель и лучом ИК-излучателя, который сонаправлен с центром видеоизображения камеры. Далее микроконтроллер отправляет на драйверы шаговых двигателей управляющие сигналы, позволяющие повернуть тагер. Выстрел производит игрок. В разрабатываемой системе предусмотрен блок фиксации попаданий. При нескольких его срабатываниях установка блокирует возможность наведения и выстрела до сброса системы. Алгоритм работы разрабатываемого устройства представлен на рис. 5, структурная схема – на рис. 6.

Для реализации устройства наведения используются: одноплатный компьютер Raspberry Pi 4 Model B на Raspberry Pi OS [7], шаговые двигатели FL42STH – 0.9° NEMA 17 [8], драйверы шаговых двигателей A4988, 7-дюймовый сенсорный экран от Waveshare, инфракрасный излучающий диод TSLC 6100, оптическая система Parallax, инфракрасный TSOP-сенсор, веб-камера A4Tech PK-925H.

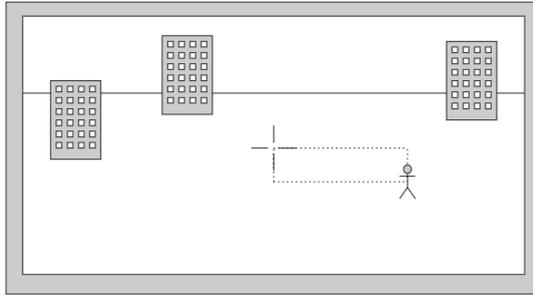


Рис. 4. Схематическое изображение на сенсорном экране устройства

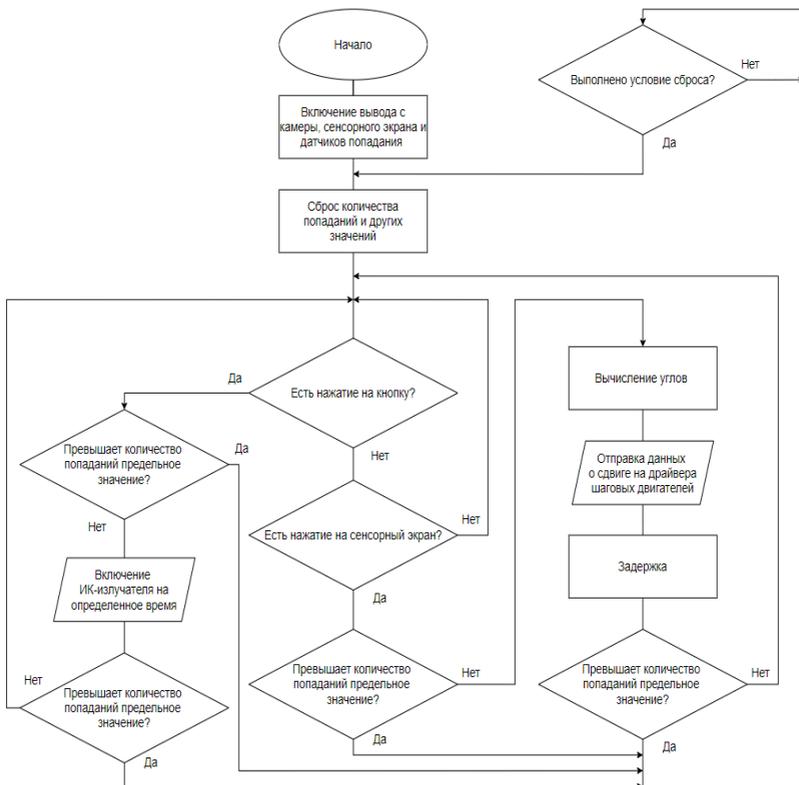


Рис. 5. Алгоритм работы устройства

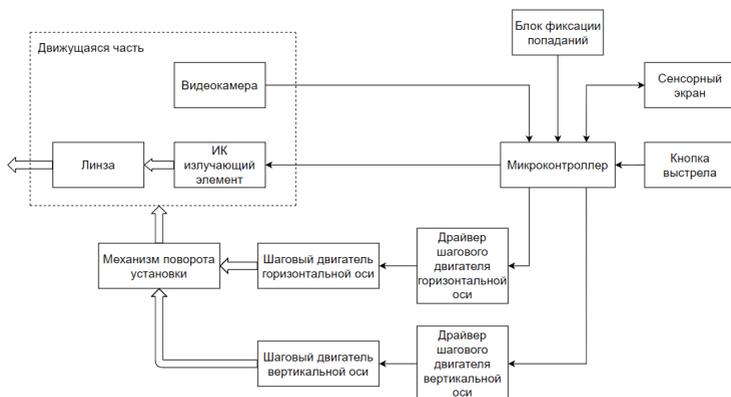


Рис. 6. Структурная схема разрабатываемого устройства

Библиографический список

1. Laser Tag Museum, Photon. URL: <http://www.lasertagmuseum.com/indoor-laser-tag/indoor-company/n-u/photon> (дата обращения: 11.04.21).
2. Комплект электроники v.10 X-GEN. URL: <https://laserwar.ru/shop/elektronika-kit/komplekt-elektroniki-x-gen/> (дата обращения: 11.04.21).
3. Комплект электроники «Лазертаг-жилет». URL: <https://laserwar.ru/shop/elektronika-kit/lasertag-vest-electronics-kit-1/> (дата обращения: 12.04.21).
4. Комплект электроники «Аптечка». URL: <https://laserwar.ru/shop/elektronika-kit/komplekt-elektroniki-apteчка-1/> (дата обращения: 12.04.21).
5. Универсальная точка «Хамелеон». URL: <https://laserwar.ru/shop/dopolnitelnye-ustroystva/dopolnitelnoe-oborudovanie/chameleon-universal-point-1/> (дата обращения: 12.04.21).
6. Роутер ELITE для online-статистики Laserwar. URL: <https://laserwar.ru/shop/dopolnitelnye-ustroystva/specialnoe-oborudovanie/laserwar-online-statistic-elite-router/> (дата обращения: 16.04.21).
7. Documentation Raspberry Pi OS. URL: <https://www.raspberrypi.org/documentation/raspbian/> (дата обращения: 12.04.21).
8. Шаговые двигатели FL42STH – 0,9°, NEMA 17 (42 мм), характеристики, размеры. URL: http://www.npoatom.ru/katalog/step_motor/fl42sthm/ (дата обращения: 16.04.21).

УДК 621.3

Н. И. Мирошниченко

студент кафедры проблемно-ориентированных вычислительных комплексов

А. В. Сорокин – ассистент – научный руководитель

МЕТОДЫ МОДЕРНИЗАЦИИ ЭЛЕКТРИЧЕСКИХ СЕТЕЙ. ВНЕДРЕНИЕ SMART GRID

Одна из главных проблем современной электроэнергетики – продолжающийся рост износа основного оборудования электрических сетей и станций. Актуальность вопроса определения состояния электрических сетей, распределительного оборудования и, в частности, комплексных распределительных устройств возрастает с каждым днем. Процент распределительных электрических сетей, выработавших нормативный срок службы, составляет около 50. Общий износ распределительных сетей достиг 70 %, магистральных – 50 %, в то время как износ электросетевых активов в промышленно развитых странах составляет 27–44 %.

В целом по России в эксплуатации находится порядка 1,2 млн комплексных распределительных устройств с высоковольтными выключателями, имеющих срок службы более 20–30 лет. Из них около 30 % подлежат экстренной замене из-за частых сбоев и отказов в работе [1].

Основными показателями для потребителя электроэнергии выступают качество электроэнергии и ее надежность, которые характеризуются напряжением синусоидальной формы, стабильной по величине, частоте и форме. Эти параметры могут отклоняться в небольших пределах. Приведем пример (рис. 1).

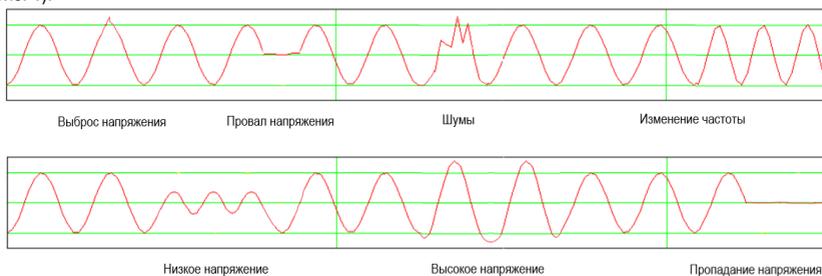


Рис. 1. Типовые проблемы электроснабжения

В большинстве случаев системы производства электроэнергии и ее распределения не могут полностью обеспечить достаточный уровень надежности системы электроснабжения. Например, передача электроэнергии происходит на длительных участках линий электропередачи от электростанций через трансформаторные подстанции и коммутационные щиты к конечному потребителю. Чем дальше от источника, тем больше риск возникновения проблем с качеством, надежностью электроснабжения и неполадок в работе электрооборудования.

Проблемы с электроснабжением возникают по разным причинам:

- 1) перегруженность линии электропередачи;
- 2) короткое замыкание;
- 3) низкокачественная электропроводка в здании;
- 4) выход из строя оборудования электроподстанций, неисправность;
- 5) обрыв линии электропередачи.

Рассмотрим два метода внедрения интеллектуальной сети Smart Grid.

Первый заключается в усовершенствовании энергетических узлов подстанции 380/220 кВ.

Вторичная система подстанции выполняет функции контроля состояния и защиты первичной цепи от короткого замыкания. Две эти задачи решаются при помощи интеллектуальных электронных устройств, у которых входы и выходы соединены с оборудованием первичной цепи для контроля всех

или части ячеек распределительного пункта. На подстанции целесообразно внедрение интеллектуальных электронных устройств, которые должны быть присоединены к каналу передачи информации. Стандарт определяет файл, который содержит все необходимые данные о конфигурации подстанции: от топологии основного оборудования до данных о вторичных цепях.

Второй метод заключается в том, чтобы внедрить умные счетчики в квартирах, частных домах и офисах. Предполагается, что эти приборы будут вести учет расхода электроэнергии по дифференцированным тарифам в зависимости от времени суток. Предусматривается возможность потребителей уменьшать или увеличивать потребление электроэнергии с тем, чтобы оно было максимальным во время суточного спада потребления и минимальным в пиковые периоды.

Внедрение интеллектуальной сети Smart Grid поможет противостоять физическим и информационным негативным воздействиям без тотальных отключений или высоких затрат на восстановительные работы, максимально быстрое восстановление (самовосстановление) [2].

Сами по себе интеллектуальные сети Smart Grid представляют собой двусторонний обмен, при котором электричеством можно обмениваться в обоих направлениях – между энергокомпаниями и потребителями. Эта растущая сеть связи, автоматизации, компьютеров и управления помогает сделать сеть более эффективной, надежной, безопасной и «зеленой» [3].

Технология Smart Grid имеет ряд исключительных преимуществ, например:

- 1) эффективное использование электроэнергии;
- 2) обеспечение синхронной работы источников генерации и узлов хранения электроэнергии;
- 3) способность сети к самовосстановлению;
- 4) повышение эффективности работы энергосистемы;
- 5) снижение операционных затрат;
- 6) точность и прозрачность расчетов электроэнергии;
- 7) повышение точности расчета фактических и коммерческих потерь в сетях;
- 8) охват всей технологической цепочки энергетической системы;
- 9) увеличение доли распределенных систем производства энергии и возобновляемых источников энергии;
- 10) повышение гибкости подачи питания;
- 11) увеличение стабильности и качества подачи электричества;
- 12) увеличение безопасности энергосистем [4].

Библиографический список

1. Вороничкий В. Э. Ключевые проблемы электроэнергетики России и нормативно-правовые пути их решения. URL: <https://clck.ru/Ughzh> (дата обращения: 25.03.2021).
2. Воропай Н. И. Интеллектуальные электроэнергетические системы: концепция, состояние, перспективы. URL: <https://transform.ru/articles/html/12reforma/ref00090.article> (дата обращения: 25.03.2021).
3. Davis N. The Smart Grid: What's «the grid» and how is it «smart?». URL: <https://www.powerelectronicsnews.com/the-smart-grid-whats-the-grid-and-how-is-it-smart/> (дата обращения: 25.03.2021).
4. Гуревич В. И. Интеллектуальные сети: новые перспективы или новые проблемы? URL: http://www.gurevich-publications.com/articles_pdf/smart_grid_2.pdf (дата обращения: 25.03.2021).

УДК 004.4.275

В. К. Михайлов

студент кафедры проблемно-ориентированных вычислительных комплексов

Е. Л. Турецкая – кандидат технических наук – научный руководитель

АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ СРЕДСТВ ВИЗУАЛИЗАЦИИ ТРЕХМЕРНОЙ ГРАФИКИ В ВЕБ

В настоящее время трехмерная графика активно применяется в дизайне веб-приложений. Разработчики программного обеспечения (ПО) для работы с 3D-графикой и отображения ее в браузере используют кроссплатформенный программный интерфейс API WebGL (Web-based Graphics Library), основанный на специализированных библиотеках и фреймворках, таких как Babylon JS [1], Three JS [2] и Unity [3]. Повышения эффективности процесса разработки веб-приложений достигают применением фреймворка с требуемыми техническими характеристиками для решения конкретной задачи.

Цель работы – сравнение производительности перечисленных фреймворков для разработки трехмерных веб-приложений. Это востребованные фреймворки для добавления трехмерной графики в веб, поддерживаются браузерами Google Chrome, Safari, Mozilla, имеют справочную документацию, активно используются разработчиками и сообществом программистов.

Тестирование фреймворков проведено на персональном компьютере с техническими характеристиками, представленными в табл. 1.

Таблица 1

Характеристики вычислительной системы

Параметр	Показатель
CPU	Intel Core i7-4 720HQ 2.60GHz – 3.60GHz
GPU	Nvidia GTX 960M 2GB
RAM	12GB 1 600 MHz
OS	Windows 10
Монитор	1 920 × 1 080 × 60 Гц

Программные инструменты для проведения замеров параметров фреймворков.

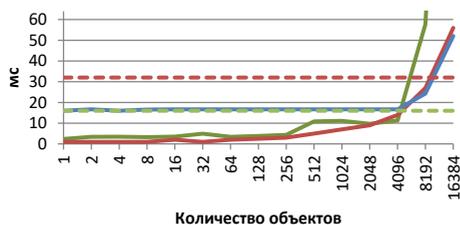
1. Google Chrome – браузер, поддерживающий все современные стандарты [4].
2. GPUZ – бесплатная программа для отображения информации о состоянии GPU [5].
3. Windows Task Manager – показывает состояние персонального компьютера и запущенных на нем процессов.

Для каждого фреймворка проведено пять серий тестов, в основе которых лежит изменение производительности ПК в зависимости от количества объектов на сцене, теней на сцене, полигонов на тестируемом объекте, производительности сцены или модели и производительности сглаживания. Следует отметить, что браузер ограничивает отображение графических элементов до частоты обновления экрана, поэтому все значения времени кадра менее 16 мс не имеют критического значения. Основным показателем при прохождении теста было среднее время отображения кадра. Время от 0 до 16 мс считается хорошим, что составляет 60 кадров в секунду и больше; от 17 до 32 мс – нормальным (от 60 до 30 кадров в секунду); время 32 мс неприемлемо для большинства задач (< 30 кадров в секунду).

Результаты тестов представлены в табл. 2–6. Наглядное представление результатов можно увидеть на графиках (рис. 1–7), где результаты тестирования фреймворка BABYLON JS отображены зеленым цветом, Three JS – красным, Unity – синим.

Тест 1. Загрузка основных элементов ПК в зависимости от количества объектов на сцене

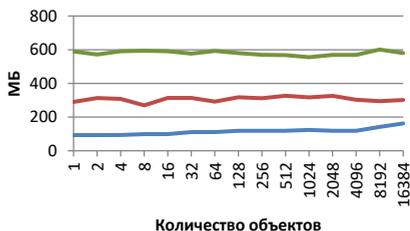
Количество объектов	Среднее время кадра (мс)	Загрузка GPU (%)	Загрузка CPU (%)	VRAM (МБ)
BABYLON JS				
256	4,34	47	9,9	570
4 096	11,25	57	24,40	569
8 192	57,7	48	27,5	601
16 384	301,3	45	21	580
Three JS				
256	3	35	7,2	312
4 096	14	62	23,7	303
8 192	27	71	23,8	295
16 384	56	78	24,4	302
Unity				
256	16,7	47	7,7	119
4 096	16,7	56	21,3	119
8 192	24,3	55	24,1	142
16 384	52,1	56	22,5	162



Длительность кадра

Загрузка GPU

Рис. 1. Результаты теста 1 по длительности кадра и загрузке GPU



Загрузка CPU

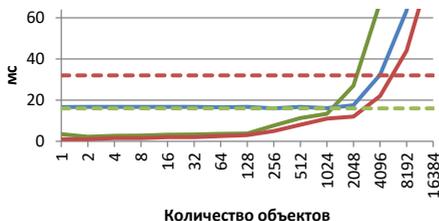
Загрузка VRAM

Рис. 2. Результаты теста 1 по загрузке CPU и VRAM

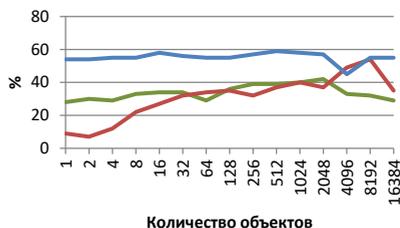
Тест 2. Загрузка основных элементов ПК в зависимости от количества теней объектов на сцене

Количество объектов	Среднее время кадра (мс)	Загрузка GPU (%)	Загрузка CPU (%)	VRAM (МБ)
Babylon JS				
256	7,64	39	8,1	393
4 096	67,5	33	15,9	396
8 192	207	22	19,7	508
16 384	1 176	15	20,1	502
Three JS				
256	5	32	7,7	521
4 096	22	49	24,1	497
8 192	44	54	23,9	498
16 384	88	35	23,5	500
Unity				
256	16	57	11,1	237
4 096	32,3	45	24,1	224
8 192	63,5	55	23,8	224
16 384	127,3	55	23,7	224

Тест 2. Время кадра



Тест 2. GPU

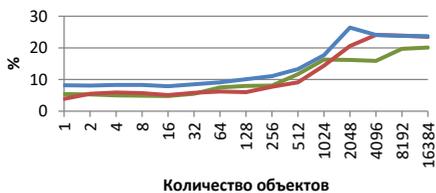


Длительность кадра

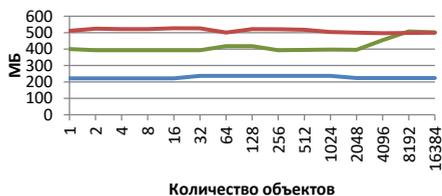
Загрузка GPU

Рис. 3. Тест 2. Время кадра и Загрузка GPU

Тест 2. CPU



Тест 2. VRAM



Загрузка CPU

Загрузка VRAM

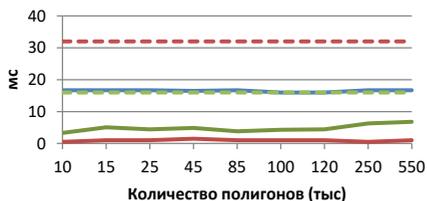
Рис. 4. Результаты теста 2 по загрузке CPU и VRAM

Таблица 4

Тест 3. Загрузка основных элементов ПК в зависимости от количества полигонов на модели

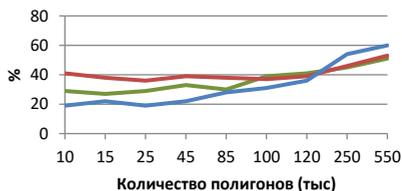
Количество (×10 ³)	Среднее время кадра (мс)	Загрузка GPU (%)	Загрузка CPU (%)	VRAM (МБ)
Babylon JS				
10	3,3	29	9,1	316
15	5,06	27	8,4	315
25	4,43	29	8,6	339
45	4,88	33	9,8	337
85	3,8	30	10,3	338
100	4,3	39	7,6	336
120	4,41	41	8,7	368
250	6,3	45	9,8	339
550	6,8	51	8,6	371
Three JS				
10	< 1	41	5	300
15	1	38	5,1	301
25	1	36	4,8	303
45	1,5	39	5,1	301
85	1	38	9,8	318
100	1	37	10	329
120	1	39	10,2	308
250	0,5	46	10,9	334
550	1	53	12,5	310
Unity				
10	16,7	19	9,7	167
15	16,7	22	9,5	167
25	16	19	9,3	168
45	16	22	9,4	170
85	16,5	28	12,5	173
100	16,7	31	11,4	178
120	16	36	11,7	183
250	16	54	11,9	193
550	16,7	60	12,1	217

Тест 3. Время кадра

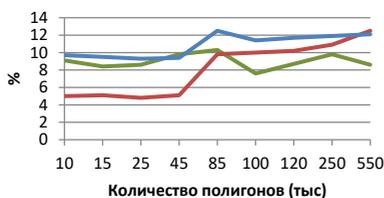


Длительность кадра

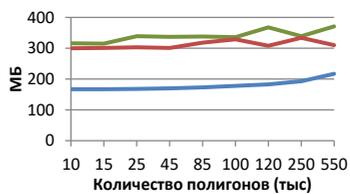
Тест 3. GPU



Загрузка GPU



Загрузка CPU



Загрузка VRAM

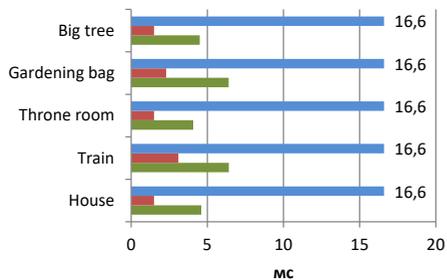
Рис. 5. Результаты теста 3

При проведении тестов пятой серии были использованы низкополигональная сцена с домом [6], высокополигональные сцены с вагоном [7] и тронным залом [8], фотограмметрические сцены с мешком [9] и деревом [9]. Файлы сцен и моделей распространяют по лицензии CC BY 4.0. Они загружены из онлайн-галереи 3D-моделей Sketchfab (<https://sketchfab.com/>).

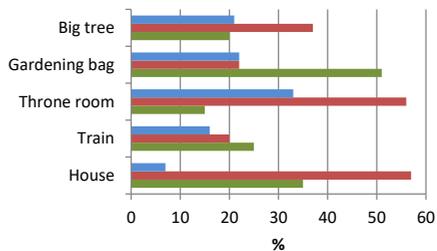
Таблица 5

Тест 4. Загрузка основных элементов ПК при отображении реальных сцен

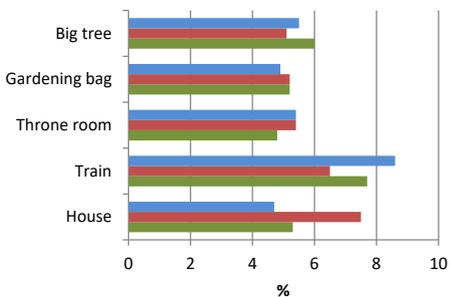
Сцена	Среднее время кадра (мс)	Загрузка GPU (%)	Загрузка CPU (%)	VRAM (МБ)
Babylon JS				
House	4,6	35	5,3	406
Train	6,41	25	7,7	421
Throne Room	4,07	15	4,8	396
Gardening Bag	6,4	51	5,2	1 074
Big tree	4,5	20	6	730
Three JS				
House	1,5	57	7,5	291
Train	3,1	20	6,5	277
Throne room	1,5	56	5,4	260
Gardening bag	2,3	22	5,2	937
Big tree	1,5	37	5,1	596
Unity				
House	16,6	7	4,7	864
Train	-	16	8,6	912
Throne room	-	33	5,4	912
Gardening bag	-	22	4,9	912
Big tree	-	21	5,5	868



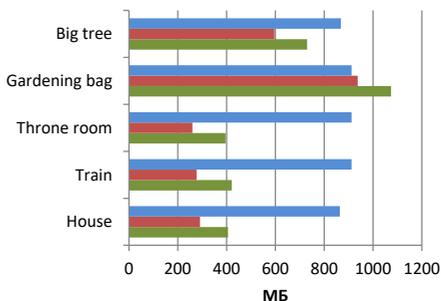
Длительность кадра



Загрузка GPU



Загрузка CPU



Загрузка VRAM

Рис. 6. Результаты теста 4

Таблица 6

Тест 5. Загрузка основных элементов ПК при использовании сглаживания

Среднее время кадра (мс)	Загрузка GPU (%)	Загрузка CPU (%)	VRAM (МБ)
Babylon JS			
7,4	38	5,3	1 181
Three JS			
2,1	31	5,1	1 050
Unity			
16	50	7,2	923

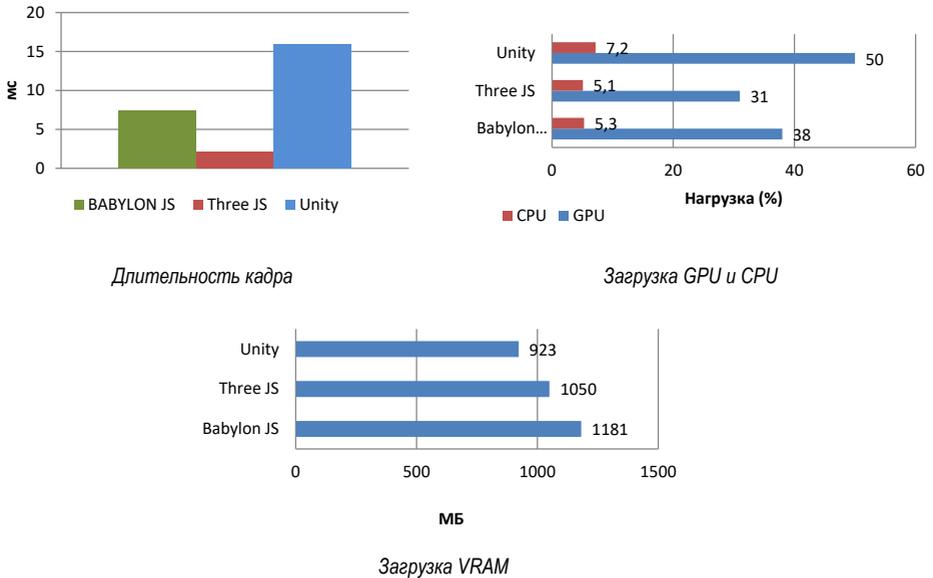


Рис. 7. Результаты теста 5

Результаты тестов показывают, что ни один из исследуемых фреймворков не использовал предоставленные вычислительные ресурсы на 100 %. Различие в производительности между ними проявляется при больших нагрузках, при низких и средних разница не существенна. Можно сделать вывод, что разработчикам игровых приложений предоставлены многовариантные программные инструменты. Поэтому для каждого проекта потребуется выбрать тот фреймворк, на основе которого возможно эффективнее реализовать идею приложения.

Библиографический список

1. Babylon JS. URL: <https://www.babylonjs.com/> (дата обращения: 20.04.2021).
2. Three JS. URL: <https://threejs.org/> (дата обращения: 26.04.2021).
3. Unity. URL: <https://unity.com/> (дата обращения: 20.04.2021).
4. Google Chrome. https://www.google.com/intl/ru_ru/chrome/ (дата обращения: 20.04.2021).
5. GPUZ. <https://www.techpowerup.com/gpuz/> (дата обращения: 20.04.2021).
6. Korshunova U. House and brewery. URL: <https://skfb.ly/onqF1> (дата обращения: 20.04.2021).
7. Tumblewee British Rail Class 55 Swallow Livery. URL: <https://skfb.ly/onr9P> (дата обращения: 20.04.2021).
8. Stashko M. Throne room. URL: <https://skfb.ly/on8tT> (дата обращения: 20.04.2021).
9. Matousekfoto Gardening garbage collection bag. URL: <https://skfb.ly/ontMY> (дата обращения: 20.04.2021).
10. B_nealie Moreton Bay Fig Tree 2 Rescan. URL: <https://skfb.ly/on8RU> (дата обращения: 20.04.2021).

УДК 621.3.056

А. С. Раскопина

студент кафедры проблемно-ориентированных вычислительных комплексов

А. В. Сорокин – ассистент – научный руководитель

СИСТЕМА МОНИТОРИНГА ВЫСОКОВОЛЬТНЫХ СЕТЕЙ ДЛЯ ПОВЫШЕНИЯ ИХ ПРОПУСКНОЙ СПОСОБНОСТИ

Поддержание в эффективном состоянии и своевременный ремонт инфраструктуры – ресурсо-емкая задача. Актуальна разработка принципиально нового подхода и методов обеспечения безопасности, а также мониторинга потенциально опасных воздействий на объекты высоковольтных сетей.

На пропускную способность высоковольтных сетей, среди прочего, влияет уязвимость воздушных линий электропередачи к механическим повреждениям. При понижении температуры увеличивается риск образования изморози и ледяных отложений на них, что при фиксированных точках закрепления повышает механические напряжения. Повышение температуры проводов приводит к снижению механической прочности. Кроме того, в этом случае провода удлиняются и увеличиваются стрелы провеса. В результате могут быть нарушены габариты воздушной линии и изоляционные расстояния.

Также гололедные образования приводят к появлению значительной механической нагрузки на провода, тросы и опоры в виде дополнительных вертикальных сил. В результате гололедных образований возникают обрывы проводов и поломки опор, сближения и схлестывания проводов с перекрытием изоляционных промежутков.

В настоящее время во всем мире находят широкое применение различные системы мониторинга воздушных ЛЭП, обеспечивающие системного оператора подробными сведениями о текущем состоянии воздушных кабельных сетей электроснабжения. Комплекс датчиков мониторинга тока, температуры и силы сжатия и растяжения поможет поддерживать в эффективном состоянии инфраструктуру, что повысит пропускную способность высоковольтных сетей. Контролируя потребление электроэнергии в режиме реального времени, можно предотвратить перегрузку и принять необходимые меры для предотвращения любой крупной аварии в инфраструктуре распределения электроэнергии.

Для реализации системы целесообразно использование датчика [1] тока, который должен обладать высокой чувствительностью и надежностью. Такие системы могут использовать датчик с большой величиной магнитоимпеданса (ГМИ) для измерения тока, однопортовую линию задержки на поверхностной акустической волне (ПАВ) для бесчиповой и безбатарейной системы приемопередатчика, элемент согласования импеданса для обеспечения высокой чувствительности, две антенны, зажимную упаковочную платформу и сетевой анализатор.

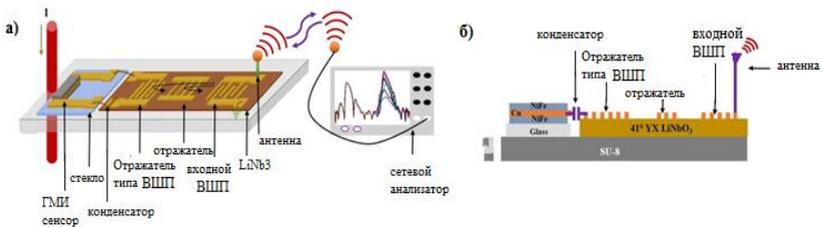


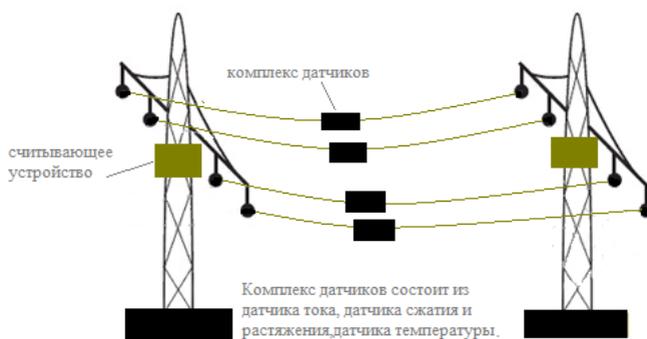
Рис.1. Предлагаемая беспроводная система датчиков тока без чипа: а – общий вид; б – вид в разрезе на прикрепляемой упаковочной платформе

Когда встречно-штыревой преобразователь (ВШП) на однопортовой линии задержки отражения на ПАВ принимает опросный сигнал через антенну, ПАВ, генерируемая на пьезоэлектрической подложке, распространяется к отражателю, расположенным на подложке.

Часть энергии ПАВ отражается отражателями, возвращается на вход ВШП и преобразуется преобразователем в электромагнитную волну. Эта волна передается в удаленную измерительную систему через антенну. Магнитный датчик с ГМИ подключается ко второму отражателю типа ВШП на однопортовой линии задержки отражения на ПАВ. Проводящая линия, по которой проходит ток, создает магнитное поле, которое влияет на изменение импеданса ГМИ, что приводит к вариациям второго пика отражения считывающего устройства. Оценивая изменения амплитуды второго пика отражения, можно оценить данные, относящиеся к величине тока в проводнике.

Предлагаемая система превосходит по ряду характеристик существующие в настоящее время системы для мониторинга беспроводных датчиков тока, так как используемые датчики не требуют источников энергии, долговечны и устойчивы к внешним воздействиям [2].

На рис. 2 изображена система мониторинга состояния ЛЭП при помощи комплекса пассивных датчиков. Их использование оправдано из-за их уникальных свойств, таких как повышенная стойкость к внешним воздействиям (повышенная температура, высокое напряжение), отсутствие необходимости в источниках питания. Передача информации на подстанцию может происходить через коммуникационные технологии Zigbee или через сотовые GSM.



Высоковольтная ЛЭП

Рис. 2. Комплекс датчиков для мониторинга состояния системы

Комплекс, показанный на рис. 2, предназначен для использования в составе системы автоматического контроля состояния провода воздушных линий электропередачи с передачей информации в систему диспетчерского управления и сбора данных (SCADA) потребителя. SCADA-система представляет собой пакет программ, предназначенный для обеспечения работы в реальном времени систем мониторинга, обработки, отображения и архивирования информации о контролируемом объекте [3].

Библиографический список

1. Койгеров А. С., Дмитриев В. Ф. Радиомаркер на поверхностных акустических волнах с мехоустойчивым частотно-манипулированным кодом // Информ.-управляющие системы. 2010. № 4. С. 22–28.
2. Sorokin A. V., Shepeta A. P. Anti-collision radio-frequency identification system using passive SAW tags // Proc. SPIE, Smart Sensors, Actuators, and MEMS VIII. Barcelona, Spain. 2017. Vol. 10246. P. 1024613.
3. Воропай Н. И. Интеллектуальные электроэнергетические системы: концепция, состояние, перспективы. URL: <https://transform.ru/articles/html/12reforma/ref00090.article> (дата обращения 25.03.2021).

УДК 614.8.084

П. А. Рогозин

студент кафедры проблемно-ориентированных вычислительных комплексов

О. О. Жаринов – кандидат технических наук, доцент – научный руководитель

АНАЛИЗ ПОДХОДОВ К РАЗРАБОТКЕ УСТРОЙСТВ ДЛЯ ОЦЕНИВАНИЯ ФУНКЦИОНАЛЬНОГО СОСТОЯНИЯ ЧЕЛОВЕКА-ОПЕРАТОРА

Введение

Существенный промежуток своей жизни человек проводит на работе, выполняя служебные обязанности. При этом постоянно возрастает роль роботов на производстве, набирает обороты автоматизация, а с ней получает большее распространение должность «человек-оператор», на которого возлагается обязанность следить за работой компонентов автоматизированных производств [1]. Но, помимо положительных сторон данных процессов, есть одна отрицательная – возрастание травматичности из-за переутомления монотонной деятельностью. Таким образом, необходимо следить за функциональным состоянием оператора, чтобы оценить возможность выполнения им своих обязанностей, предотвращая несчастные случаи, остановку производства и т. п. Для этого требуются системы, надежно оценивающие функциональное состояние человека-оператора.

Функциональные показатели человека-оператора и существующие устройства их оценивания

Функциональные показатели, по которым можно провести анализ состояния человека, можно разделить на несколько категорий [2].

Поведенческие показатели: анализируется поза и мимика человека, характер общения с другими людьми [2]. Данные признаки очень сложно «оцифровать», так как их оценка субъективна и результаты интерпретации неоднозначны. Для контроля таких показателей потребуются камера и специальное программное обеспечение, задача которых будет сводиться к визуальному осмотру и анализу внешнего вида оператора.

С подобной задачей на данный момент способна справиться лишь нейросеть, обрабатывающая видеоизображение с камеры, следящей за оператором. Нейросети, под управлением специалистов, сначала учится анализировать фотографии, на которых изображены лица людей с разными эмоциями, а затем программисты настраивают ее на работу с видео. Существует вариант с обработкой фотографий – камера не снимает оператора постоянно, а делает фото с определенной частотой. Стоит также упомянуть о возможности анализа не всего лица, а отдельно области глаз. При прочих одинаковых условиях (например, уровень освещенности) размер зрачка глаза человека может изменяться в зависимости от уровня усталости, учитываются и характеристики моргания.

Другой распространенный вариант – анализ характеристик голоса оператора. Данный подход интересен, но проблематичен в реализации, так как, во-первых, посторонний шум от промышленного оборудования может быть интенсивным и затруднять распознавание голоса, во-вторых, необходимо отделить речь исследуемого оператора, от других людей, разговаривающих поблизости.

К поведенческим показателям можно отнести также параметры двигательной активности оператора, в частности изменение наклона головы. Например, если он наклоняет голову слишком низко на более-менее продолжительное время, то это можно считать признаком утомления. Для контроля данного показателя сравнительно несложно использовать акселерометрический датчик.

Физиологические показатели: исследуются параметры функционирования сердечно-сосудистой, дыхательной, центральной нервной систем, изменения биопотенциалов сердца, мозга, работающих мышц, сопротивления кожи [2]. Упор делается на данные, которые можно получить не в лабораторных условиях, не только в специализированном медицинском учреждении, а при помощи приборов, прикрепленных к телу человека снаружи. Технически съем таких показателей реализуется несложно, но полученные данные также могут быть неоднозначно интерпретированы. Для полноценного анализа необходимо тщательно подбирать датчики, проводить предварительные исследования зависимостей тех или иных физиологических показателей в различных условиях.

Устройств, работающих с указанными показателями, достаточно много, перечислим основные, наиболее известные. *Пульсоксиметр* измеряет насыщение артериального гемоглобина кислородом. Используемая технология основана на оценке поглощения гемоглобином света, кроме того, световой сигнал, проходя через ткани, становится пульсирующим из-за изменения объема артериального русла при каждом сокращении сердца. Этот компонент может быть отделен микропроцессором от неп пульсирующего, идущего от вен, капилляров и тканей. *Электрэнцефалограф* (ЭЭГ) позволяет регистрировать электрическую активность головного мозга. Для оценки активности и бодрствования достаточно анализа электрических потенциалов на поверхности головы в нескольких точках. Устройство получается достаточно компактным, крепится к затылку. Однако уровень потенциалов ЭЭГ весьма мал, и распознавание этих потенциалов затрудняют помехи.

Существуют методы, основанные на измерении значения критической частоты слияния мельканий, но в результате исследований выявлено, что этот параметр зависит от большого числа факторов: предварительной адаптации, условий проведения измерения и побочных слуховых, температурных, обонятельных раздражителей; существенное влияние оказывают характеристики оборудования и методики измерения [3]. Таким образом, использование данного метода малоперспективно из-за неоднозначности результатов.

Психологические показатели основываются на результатах выполнения оператором некоторых тестов на внимание и мышление [2]. Помимо оценки правильности результатов выполнения тестовых заданий, оценивается время, потраченное на его прохождение. Тесты могут быть представлены как в электронном, так и в бумажном виде. Суть от этого не меняется: оператору задаются вопросы, предъявляются различные задачи. В идеальных условиях он пройдет тестирование с максимальной скоростью, процент правильных ответов будет наибольшим. При повышении усталости показатели будут падать.

Биохимические показатели: анализируется обмен веществ, эндокринные и ферментные реакции [2]. Результаты соответствующих анализов (например, анализа крови) могут многое сказать о состоянии оператора, но при этом получение таких данных в условиях трудовой деятельности оператора в реальном времени практически невозможно.

Субъективные показатели: оценка функционального состояния человека-оператора им самим [2]. Как ясно из названия, данный метод опирается на субъективные оценки, когда оператор сам отвечает на вопрос о своем состоянии, например по десятибалльной шкале. Данные показатели не могут объективно передать всю картину и служат лишь вспомогательным средством верификации функционального состояния оператора.

Подходы к разработке устройств

Универсального устройства и аппаратно-программного комплекса для оценки функционального состояния человека-оператора, обеспечивающих необходимую достоверность анализа, еще не придумано. Связано это с объективно высокой сложностью оценивания: разработчик должен уметь предвидеть, какие результаты ожидаемы во всех возможных условиях в процессе реальной работы. Обычно решаются частные задачи, создаются отдельные приборы, устройства, требования к функционированию которых можно более-менее надежно формализовать.

Уже существуют приборы, способные анализировать сразу несколько каналов информации: видео-, аудио- и фото – и обеспечивающие высокую достоверность анализа физиологического состояния обследуемого. Например, полиграф «Диана-07». Но, во-первых, подобные приборы не являются носимыми, во-вторых, их стоимость велика.

Более простые приборы для решения поставленной задачи основаны на анализе физиологических показателей. Обзор показывает, что чаще всего производители основывают свои приборы либо на методах пульсоксиметрии (совместно с датчиком пульса), либо на измерении параметров движения с использованием акселерометра, либо на измерении параметров кожно-гальванической реакции (КГР). Одним из основных факторов, который учитывают все разработчики, является минимум требований к экипировке оператора и к условиям выполнения оператором своих функций на рабочем месте.

Одна из главных особенностей всех подходов состоит в том, что для однозначной обработки полученных данных, повышения достоверности анализа состояния человека-оператора необходимо

заранее провести ряд «эталонных» измерений, настраивая систему под индивидуальные особенности конкретного человека. Как показывает опыт, в настоящее время надежность результатов, получаемых с использованием имеющихся носимых приборов, настолько низка, что операторы часто вынуждены отключать их, чтобы избежать эксцессов на рабочем месте.

Перспективы развития

С каждым годом значимость искусственного интеллекта растет. Компьютер учится распознавать человеческие эмоции, анализировать видеозаписи. В будущем, вероятно, самой надежной и простой системой может стать система на основе видеофиксации. Во-первых, единственное, что необходимо для ее работы со стороны оператора – доступное для наблюдения лицо. Во-вторых, видеофиксация в наименьшей степени зависима от индивидуальных физиологических особенностей оператора.

Одним из направлений развития систем на основе видеофиксации являются технологии, использующие умные очки. Уже сейчас данные устройства применяют в промышленности. Добавление камеры, сканирующей лицо работника, может позволить решить проблему постоянного слежения за лицом человека. На данный момент сложности возникают в обеспечении компактного размещения камеры и передатчика регистрируемых данных так, чтобы не усложнять экипировку оператора. В идеале необходимо отказаться от проводов, добиться полной автономности устройства, возможности установить его компактно так, чтобы не сковывать движений оператора (например, на его голове).

В результате анализа существующих устройств, а также с учетом требований к данным приборам был сделан вывод, что при разработке прибора целесообразнее опираться на технологии умных очков [4] и измерение параметров движения оператора с применением акселерометрических датчиков. Показания акселерометра минимально зависят от индивидуальных особенностей человека. Умные очки, оснащенные специальной малогабаритной камерой, позволяют надежно определять, закрыты или открыты глаза, в реальном времени.

Реализацию опытного прототипа устройства, основанного на измерении параметров движения оператора, можно осуществить на базе существующих готовых модулей для линейки Arduino: акселерометр, микроконтроллер и Bluetooth-модуль смогут обеспечить решение всех задач по сбору, обработке и передаче необходимой информации на центральный модуль сбора данных. На данный момент открытым остается вопрос о том, будет ли точности акселерометра достаточно для анализа поведения человека с учетом специфики поставленной задачи.

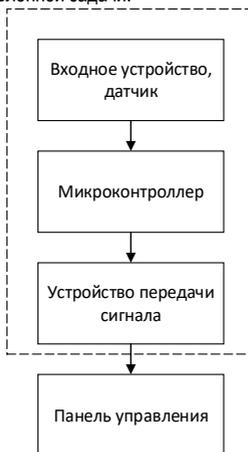


Рис. 1. Функциональная схема устройства (пунктирной линией обведена часть, соответствующая разрабатываемому носимому устройству)

Немаловажным фактором выступает уровень энергопотребления. Аккумулятор должен быть выбран, исходя из необходимости непрерывной работы устройства не менее 8 ч. Для удобства необходимо сделать аккумуляторный отсек прибора съёмным, чтобы была возможность быстрой замены отработанного аккумулятора на другой, предварительно заряженный.

Также необходимо учитывать, что на производствах часто наблюдается повышенный уровень электромагнитных помех, поэтому интенсивность обмена данными между носимым прибором и центральным модулем сбора данных должна быть минимальной, чтобы исключить ложные срабатывания аппаратуры. Поэтому вся обработка данных и принятие решение о степени усталости оператора должны производиться на вычислительной платформе, установленной на носимом приборе; на модуль сбора данных (панель управления) передается небольшой набор служебной информации и оцененный уровень усталости оператора.

Заключение

Универсальность вычислительного модуля позволит при необходимости усовершенствовать устройство с целью повышения достоверности анализа. Например, несложно добавить в такую систему датчик ЭЭГ и пульсовой датчик. Параллельная обработка данных с датчика ЭЭГ и акселерометра позволит идентифицировать «артефакты» в канале съема ЭЭГ, большинство из которых возникают вследствие резких движений человека, таким образом можно будет снизить вероятность ошибочной интерпретации регистрируемых данных.

Библиографический список

1. Автоматизация производства. URL: <https://www.expocentr.ru/ru/articles-of-exhibitions/2016/avtomatizaciya-proizvodstva/> (дата обращения 08.02.2021).
2. Функциональные состояния оператора. URL: <https://lektcii.org/12-55434.html> (дата обращения 08.02.2021).
3. Критическая частота слияния мельканий. URL: <https://eduherald.ru/ru/article/view?id=12282> (дата обращения 08.02.2021).
4. Промышленные умные очки Toshiba. URL: <https://nplus1.ru/turbopages.org/nplus1.ru/s/news/2018/03/13/toshiba-ar-glasses> (дата обращения 08.02.2021).

УДК 004.921

М. В. Русанов

студент кафедры проблемно-ориентированных вычислительных комплексов

Е. П. Виноградова – старший преподаватель – научный руководитель

РЕАЛИЗАЦИЯ ГРАФИЧЕСКОГО МОДУЛЯ TURTLE С ПОМОЩЬЮ OPENGL

Цель работы – изучение принципов работы Turtle Graphics и их реализация на базе OpenGL. Мотивацией разработки послужила необходимость в получении базы для создания аналогичного Turtle графического модуля, но в 3D-пространстве для моделирования L-систем. Принцип работы L-систем вкратце можно описать следующим образом. Имеются аксиома A – начальная строка символов и набор так называемых правил R , которые преобразуют строку аксиомы. Например, $A: f; R: f \rightarrow fx + f, x \rightarrow f - f$. Затем порожденное данной грамматикой слово интерпретируется как действия некоего объекта «художника»: ' f ' – пойти вперед; '+' – поворот направо; '-' – поворот налево [1].

OpenGL расшифровывается, как Open Graphics Library и представляет собой платформо-независимую библиотеку рисования 2D- и 3D-графики, а также сцен, с помощью простых примитивов. С точки зрения программирования OpenGL – это программный интерфейс, состоящий из набора функций, которые предоставляют возможность использования графики в приложениях. OpenGL имеет очень широкую сферу применения – от создания компьютерных игр до визуализации в научных исследованиях [2].

Прежде чем приступить к описанию работы программы, определим, какие принципы должны выполняться, чтобы рисование графики осуществлялось по Turtle Graphics.

Все рисование графики должно выполняться неким объектом (или несколькими), который находится на холсте. Причем данные объекты должны реализовывать следующие базовые действия:

- 2) повернуть на указанный угол налево/направо;
- 3) переместиться вперед и оставить след;
- 4) переместиться вперед, не оставляя след;
- 5) положить текущее положение в стек;
- 6) встать на место из верхушки стека.

Описание программы

Состав программы.

1. Структура $TDrawer(x, y, \alpha)$, где x, y – координаты объекта на холсте; α – угол поворота против часовой стрелки относительно оси Ox , т. е. то направление, куда «смотрит» объект. При $\alpha = 0$ объект смотрит вдоль оси Ox .

2. Структура $Tstack_Drawer(MAX, stack, tos)$, где MAX – вместимость стека; $stack$ – массив структур $TDrawer$ размера MAX ; tos – индекс верхнего элемента стека в массиве $stack$.

3. Метод объекта $InitDrawer(obj, x, y, \alpha)$, где obj – экземпляр структуры $TDrawer$; x, y – координаты на холсте; α – угол поворота относительно оси Ox , т. е. то направление, куда «смотрит» объект. Данный метод, используя аналогию с объектно-ориентированным программированием (ООП), реализует функционал конструктора для типа $TDrawer$, т. е. присваивает значения x, y, α экземпляру obj структуры $TDrawer$.

4. Метод объекта $setD(obj1, obj2)$, где $obj1$ и $obj2$ – экземпляры (скорее всего различные) структуры $TDrawer$. Метод реализует присваивание атрибутов $obj1$ к $obj2$.

5. Метод объекта $putToD(x, y, \alpha, obj)$, где x, y, α, obj обозначают те же понятия, что и выше. Метод перемещает объект в указанную точку (x, y) на холсте с заданным поворотом α относительно оси Ox .

6. Метод объекта $forwardD(f, obj)$ – аналогичный прошлому метод, за исключением того, что объект obj при перемещении оставляет след. То есть рисуется линия длины f от начальной точки, в которой находится obj , до конечной точки, которая вычисляется через проекцию f (так как известен угол

поворота obj) на оси O_x и O_y и прибавлением значений проекций к текущим координатам obj соответственно.

7. Метод объекта moveToD(f, obj), где f – вещественное число. Метод реализует перемещение объекта obj на f пикселей вперед, не оставляя след. Конечные координаты вычисляются по аналогии с методом forwardD().

8. Метод объекта rightD(α, obj) – поворот объекта obj направо на угол α, т. е. по часовой стрелке, относительно оси O_x.

9. Метод объекта leftD(α, obj) – поворот объекта obj налево на угол α, т. е. против часовой стрелке, относительно оси O_x.

10. Метод стека InitDrawerStack(stack_obj, tos, MAX), где stack_obj – экземпляр структуры Tstack_Drawer; tos – верхушка стека; MAX – максимальное количество элементов в стеке. Выполняет инициализацию экземпляра структуры Tstack_Drawer. По аналогии с ООП конструктор структуры Tstack_Drawer.

11. Метод стека push_Drawer(stackD, obj), где stackD – стек экземпляров структуры TDrawer; obj – некоторый объект TDrawer. Метод кладет объект obj в верхушку стека. То есть obj как бы «запоминает», где он находится.

12. Метод стека pop_Drawer(stackD, obj) – метод достает из верхушки стека stackD экземпляр TDrawer и присваивает значения его атрибутов объекту obj. То есть obj возвращается на место, где был последний элемент стека.

Наличие стека очень полезно, когда требуется, например, нарисовать дерево

Примеры работы программы

1. Рисование буквы Y.

Перед основным циклом OpenGL создается объект d структуры TDrawer с начальными координатами (0, 0) и поворотом относительно оси O_x на 90° (вдоль оси O_y) и его стек stackD структуры Tstack_Drawer. Далее в основном цикле прорисовки графики OpenGL пишется следующий текст программы:

```
forwardD(200, &d);
push_Drawer(&stackD, &d);
rightD(45, &d);
forwardD(100, &d);
pop_Drawer(&stackD, &d);
leftD(45, &d);
forwardD(100, &d);
```

Здесь объект d двигается на 200 пикселей вперед, запоминает текущее положение, поворачивает на 45° вправо и двигается на 100 пикселей вперед. Далее методом pop_Drawer возвращается на предыдущее положение, поворачивает на 45° влево и двигается на 100 пикселей вперед.

В результате получается изображение (рис. 1).



Рис. 1. Рисунок буквы Y.

2. Рисование различных фигур.

В данном случае создаются два объекта перед основным циклом OpenGL d и d2 структуры TDrawer без стека. Объект d находится на координатах (300, 300), d2 – на координатах (–100, –350).

Оба объекта направлены вдоль оси x. Как и в предыдущем примере, прорисовка выполняется в основном цикле OpenGL.

```

for (int i = 0; i < 50; i++)
{ // Рисование окружности
  forwardD(20, &d);
  leftD(7.2, &d);
}

float n = 5;
putToD(0,0, 360/n, &d);
for (int i = 0; i < n; i++)
{ // Рисование пятиугольника
  forwardD(200, &d);
  rightD(360/n, &d);
}

for (int i = 0; i < 100; i++)
{ // Рисование пунктирной кривой
  if (i % 2 == 0)
    forwardD(i/2, &d2);
  else
    moveToD(i/2, &d2);
  rightD(7.2/2, &d2);
}
for (int i = 0; i < 101; i++)
{ // Рисование гладкой кривой
  forwardD(i/2, &d2);
  leftD(7.2/2, &d2);
}

```

В результате выполнения представленного кода получим изображение (рис. 2).

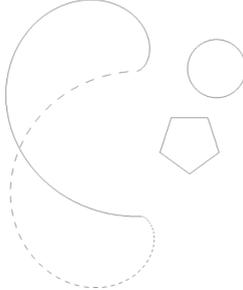


Рис. 2. Результат работы объектов d и d2.

3. Рисование растения по построенной L-системе.

В качестве L-системы было выбрано «дерево Пифагора», так как на его примере видно, чем именно полезен стек в рисовании L-систем и в принципе Turtle Graphics.

```

for (int i = 0; i < len(axiom); i++)
{
  if (axiom[i] == '0')
  {
    glColor3f(0,0,1);

```

```

forwardD(10 + p, &dL);
}
if (axiom[i] == '1')
{
glColor3f(0,1,0);
forwardD(15 + p, &dL);
}
if (axiom[i] == '[')
{
push_Drawer(&stackDL, &dL);
leftD(deg, &dL);
}
if (axiom[i] == ']')
{
pop_Drawer(&stackDL, &dL);
rightD(deg, &dL);
}
glColor3f(1,0,0);
forwardD(6, &dL);
}

```

Если описать принцип рисования дерева, не вдаваясь в подробности написания алгоритма построения строки, по которой оно рисуется, то это работает следующим образом.

Имеется алфавит переменных и констант – { 0, 1, [,] }. Аксиома с которой начинается построение строки, в данном случае 0. И правила преобразования строк в данном случае: (1 → 11), (0 → 1[0]0) [1].

С рисованием графики это связано следующим образом:

- 0: рисуй отрезок, кончающийся листом;
- 1: рисуй отрезок;
- [: положи в стек положение и угол рисования, поверни влево на 45°;
-]: выбери из стека положение и угол, поверни вправо на 45°.

В результате получим изображение (рис. 3). В данном случае поворот в стороны выполняется не на 45°, а на 15°.

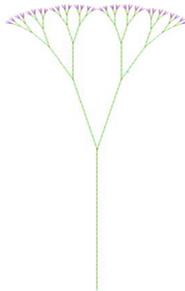


Рис. 3. Рисунок «дерева Пифагора»

Библиографический список

1. L-система // Википедия. URL: https://ru.wikipedia.org/wiki/L-система#Пример_2:_Дерево_Пифагора (дата обращения: 15.05.2021).
2. OpenGL (Open Graphics Library) // Национальная библиотека им. Н. Э. Баумана. URL: [https://ru.bmstu.wiki/OpenGL_\(Open_Graphics_Library\)](https://ru.bmstu.wiki/OpenGL_(Open_Graphics_Library)) (дата обращения: 14.05.2021).

УДК 004.9; 069.02:929

Е. И. Сергеева

студент кафедры проблемно-ориентированных вычислительных комплексов

Н. Н. Григорьева – старший преподаватель – научный руководитель

ВИРТУАЛЬНЫЙ МУЗЕЙ КАК ИНФОРМАЦИОННАЯ СИСТЕМА

Виртуальные музеи (ВМ) обязаны своим появлением стремительно развивающимся информационным технологиям. Один из ранних и наиболее проработанных виртуальных музеев – ресурс WebLouvre, созданный французским студентом Николя Пишем в 1994 г.

Музейные сайты, представленные сегодня в сети, можно разделить на две группы: представляющие реально существующих музеев и виртуальные музеи. Первый вид информационного ресурса выполняет ознакомительную, рекламную и популяризаторскую функции, дополняя реальный музей. Сайты реальных музеев обычно делают больший акцент на внешние эффекты, создаются по образу рекламного проспекта, путеводителя, каталога.

Второй вид – это оригинальные ресурсы, представляющие любую тематику, по которой существуют реальные материалы. Иногда объекты культуры и искусства по разным причинам оказываются безвозвратно утраченными или сильно видоизмененными. В этом случае незаменимым методом становится виртуальная реконструкция, результаты которой в дальнейшем могут стать основой для проектирования виртуального музея. «Виртуальный музей – это информационная система (ИС), содержащая концептуально единую электронную коллекцию или совокупность коллекций предметов (экспонатов) с метаданными, имеющая характеристики музея и позволяющая осуществлять научную, просветительскую, экспозиционную и экскурсионную деятельность в виртуальном пространстве» [1].

Созданию виртуального музея предшествует проработка научной концепции и дизайнерского решения. Необходимо решать следующие задачи:

- сформулировать концепцию отбора информации (музейных экспонатов и материалов) и принципы формирования виртуального музейного фонда;
- определить набор интерфейсов доступа к документам музея;
- сформулировать требования к лингвистическому процессору системы, определить порядок формирования онтологии музея, словарей, тезаурусов, классификаторов, каталогов и других информационных структур;
- сформировать базовые информационные структуры для представления документов и музейных материалов (экспонатов);
- разработать архитектуру распределенной ИС, способной устойчиво и непрерывно функционировать в глобальной сети, и компоненты программного обеспечения для поддержки функционирования распределенной системы;
- осуществить наполнение баз данных документами (музейными материалами).

Необходимо учитывать, что электронные документы (экспонаты) ВМ несут разноплановый характер: это базы (таблицы) данных и идеографическая информация (описания, книги и статьи), изображения, видео-, аудиоинформация, библиографическая информация, списки данных, модели и модельные данные [2].

Виртуальный музей, будучи распределенной ИС, должен обладать свойствами, обеспечивающими ее устойчивое функционирование. Узлы системы должны обеспечивать автономное функционирование, система должна обладать свойством «прозрачности расположения», ее архитектура должна обеспечивать независимость системы от типов используемых СУБД, операционных систем, аппаратных платформ и сетевых протоколов [3].

Наша задача состоит в разработке виртуального музея, посвященного И. А. Бродскому. Создание экспозиции в виде простых статических веб-страниц в данном случае не подходит, так как они не обеспечивают нужной интерактивности – одного из ключевых свойств виртуального музея.

Содержание информационной системы разрабатываемого музея можно подразделить на несколько основных модулей:

- 1) модуль музейных объектов – основной модуль, отвечающий за различные типы музейных объектов в виртуальном музее;
- 2) модуль групп, позволяющий распределять музейные объекты по группам для удобной классификации;
- 3) лекционный модуль – структура для создания лекций и экскурсий по виртуальному музею;
- 4) модуль поиска – осуществляет поиск по музейным объектам, лекциям, экспозициям и другим объектам;
- 5) модуль справки – содержит определения и справочные статьи.

Технологически информационная система ВМ состоит из двух частей – серверной и клиентской (для сотрудников и посетителей). В разрабатываемой ИС предусмотрены три роли – посетителя музея, работника музея (администратора), управляющего его содержимым, и системного администратора.

Главная функция разрабатываемой системы состоит в генерации веб-страниц по образцу (шаблону), предусмотренному для каждого экспоната музея. Шаблон может быть использован каждым объектом БД, он встраивает в свою структуру содержимое выбранного объекта. Основные элементы ИС ВМ – экспонат и экспозиция, содержащие данные о шаблоне, по правилам которого они отображаются, и о наполняющем их содержимом. Макет карточки объекта представлен на рис. 1.

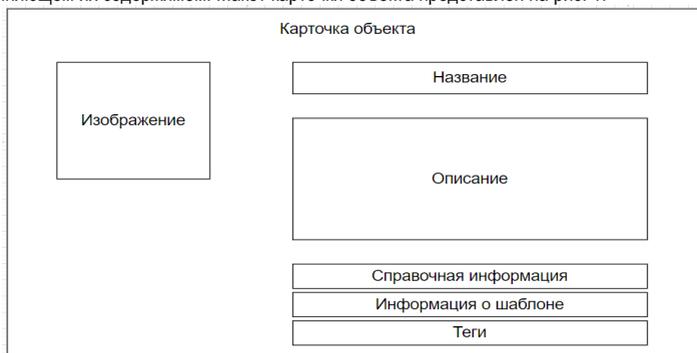


Рис. 1. Макет карточки объекта

В системе обеспечены возможности удаления, редактирования и создания экспонатов/экспозиций; добавления новых шаблонов; назначения определенного шаблона каждому объекту; автоматической генерации веб-страниц музея в соответствии с заданным для каждого объекта шаблоном.

Посетитель музея просматривает веб-страницы, которые генерирует система. Стартовой точкой выступает начальная страница музея, с которой пользователь может начать просмотр выбранной им экспозиции, а работник имеет возможность перейти в панель администрирования. Возможно сортировать экспозиции по дате публикации, либо выбирать интересующие пользователя разделы (лекции, видео, тексты, фото, относящиеся к определенному периоду жизни или творчества и т. д.).

Наиболее важным функционалом подсистемы является способность генерирования содержимого по определенному шаблону. На рис. 2 представлена так называемая разводная страница ВМ, с которой посетитель может попасть в одну из экспозиций (или войти в специальную панель администрирования, предназначенную для управления содержимым виртуального музея). Каждая экспозиция представлена в виде плитки с соответствующим изображением. Плитки не зафиксированы на странице, что дает возможность динамической генерации содержимого разводной страницы – сортировки экспозиций по критериям, по новизне и т. д.



Рис. 2. Страница экспозиций виртуального музея (динамическая часть)

Пользователь – музейный работник имеет доступ к хранилищу данных, содержимое которого и представлено для просмотра на страницах виртуального музея. Управляющие элементы данной формы авторизации создаются самим браузером. За управление учетными записями пользователей отвечает системный администратор.

Работая с панелью экспозиции, посетитель музея попадает на выбранную страницу, которая каждый раз генерируется заново с помощью шаблона, определяющего разметку страницы (рис. 3).

Шаблоном является текстовый файл с тегом frame с атрибутом name, обозначающим место для вставки реализации конкретной страницы. Столкнувшись с таким тегом, подсистема запоминает его и все атрибуты, и когда шаблон будет применяться к какому-либо объекту (экспозиции или экспонату), она осуществит просмотр фреймов этого объекта и вставит их в соответствующие места в шаблоне. Вставка будет происходить при условии совпадения атрибутов «name» и объекта.

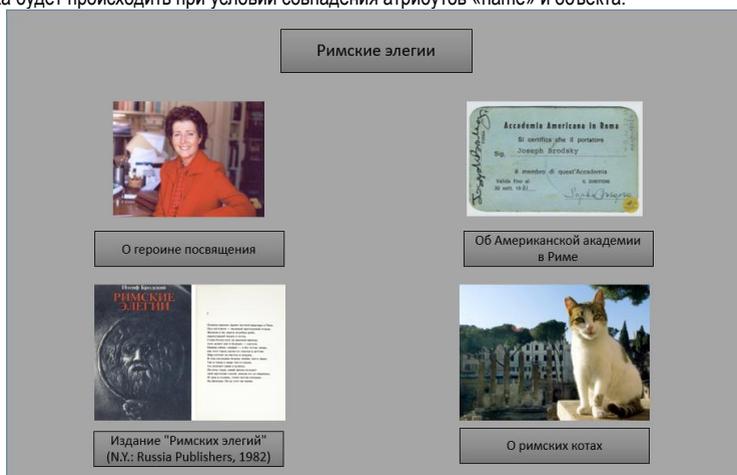


Рис. 3. Страница экспозиции «Римские элегии»

Результатом работы подсистемы будет сгенерированная страница, заполненная содержимым объекта. Пример страницы, соответствующей конкретному экспонату, приведен на рис. 4.

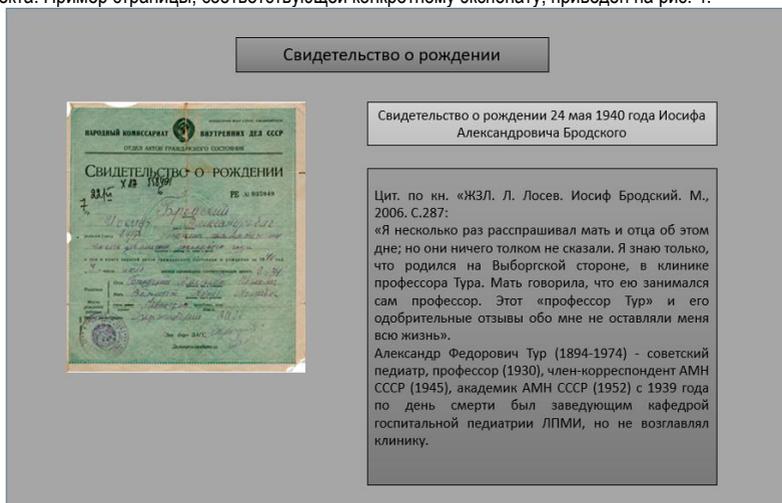


Рис. 4. Экспонат «Свидетельство о рождении»

Библиографический список

1. Поврозник Н. Г. Виртуальный музей: сохранение и репрезентация историко-культурного наследия. URL: https://www.researchgate.net/publication/329309359_Virtualnyj_muzej_sohranenie_i_reprezentacia_istoriko-kulturnogo_nasledia (дата обращения: 25.03.2021).
2. Сидоров В. А. Виртуальное музейное пространство в контексте культурно- образовательной деятельности. URL: <http://www.vestnik.vsu.ru/pdf/educ/2017/02/2017-02-17.pdf> (дата обращения: 18.04.2021).
3. Бурлакова Ю. В., Быкова Н. С. Виртуальный музей: технология создания и реализации. URL: <https://docplayer.ru/39354194-Virtualny-muзей-tehnologiya-sozdaniya-i-realizacii.html> (дата обращения: 30.03.2021).

УДК 621.3.087.44

М. Д. Синкин

студент кафедры проблемно-ориентированных вычислительных комплексов

А. В. Сорокин – ассистент – научный руководитель

ДИАГНОСТИКА НЕИСПРАВНОСТЕЙ СЕТЕЙ ЭЛЕКТРОСНАБЖЕНИЯ

Введение

Для потребителей электроэнергии важны непрерывность и надежность энергоснабжения. Электроэнергетическое оборудование – сложное техническое устройство, подверженное риску возникновения аварийных ситуаций, дефектов и отказов. Наиболее частыми причинами его выхода из строя являются износ и ухудшение состояния контактных соединений. Самый явный показатель состояния электрооборудования – температура токоведущих элементов электрических цепей. Такие цепи имеют высокую нагрузку по току, вследствие чего возникает риск перегрева контактных соединений. Подобные неисправности могут иметь скрытый характер, что затрудняет их обнаружение в момент осмотра [1].

В первую очередь нагреваются изоляция и переходные контакты электрических цепей, а также электрическое оборудование, механические части, пружины и контакты. Из основных причин возникновения перегрева можно выделить плохой контакт соединения, вследствие чего возникает повышенное переходное сопротивление, окисление компонентов, их коррозия и иные неисправности. Как правило, локальный перегрев носит длительный характер, что со временем может привести к повреждению основных узлов и агрегатов и распространению аварийной ситуации на соседние ячейки и высоковольтное оборудование [2].

Неисправное состояние наблюдаемых контактных соединений, их повышенный нагрев могут быть обнаружены с помощью термоиндикаторов либо внешнего осмотра контакта или шины. На поверхности контакта могут появиться: цвета побежалости, потемнение, выгорание краски, при сильном его нагреве контакта может возникнуть свечение. Контроль нагрева может осуществляться с помощью тепловизора. Однако в настоящий момент он подразумевает использование ремонтной бригадой инфракрасных камер с сотрудником-оператором, такая процедура трудозатратна и дорогостояща.

Проблему перегрева электрооборудования можно решить с помощью установки систем мониторинга. Выделяют три их вида систем:

- 1) термосенсорная система непрерывного контроля температурных соединений;
- 2) пирометрическая система непрерывного контроля температуры шин;
- 3) радиоканальные системы контроля температуры.

Термосенсорная система непрерывного контроля температурных контактных соединений состоит из газогенерирующих наклеек, газового датчика и контрольно-приемного устройства. Наклейки размещают на контактах. При нагревании наклейки изменяют окраску и выделяют сигнальный газ-маркер, который фиксируется датчиком.

Пирометрическая система непрерывного контроля температуры шин – это встраиваемая система измерения температуры шин бесконтактным способом. Измеряется мощность теплового излучения, после чего полученные данные обрабатываются и выводятся в систему визуализации.

Наиболее перспективны для контроля температуры высоковольтного оборудования беспроводные системы контроля температуры. Целесообразно рассматривать системы на базе пассивных акустозлектронных датчиков на поверхностных акустических волнах (ПАВ) с функцией радиочастотной идентификации. Применение в таких системах пассивных датчиков на ПАВ обусловлено их повышенной устойчивостью к электромагнитным помехам, высокой стабильностью характеристик. Средний срок службы датчиков до 20 лет.

Беспроводные системы контроля нагрева, использующие пассивные датчики

Схема работы системы контроля температуры состоит из четырех основных элементов: датчика температуры, антенны считывателя, считывателя температуры и автоматизированной системы управления (АСУ) (рис. 1).

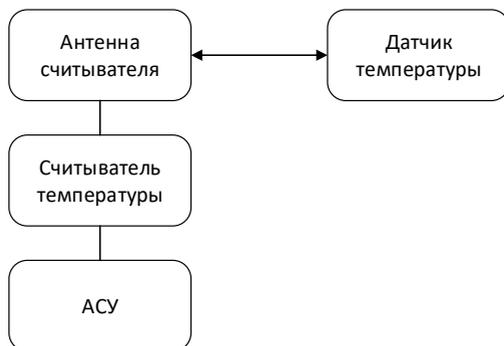


Рис. 1. Схема системы контроля температуры

Как правило, пассивный акустоэлектронный датчик температуры представляет собой акусто-электронное устройство – линию задержки, состоящую из встречно-штыревого преобразователя, подключенного к антенне, и нескольких отражающих кодовых рефлекторов, пространственно-временное положение которых зависит от температуры контролируемой среды. Эти элементы расположены на подложке из пьезоэлектрического материала. Встречно-штыревой преобразователь преобразует электромагнитную волну в акустическую, которая распространяется вдоль подложки [3].

Любые изменения, внесенные в механическую волну, будут отражены в выходном электрическом сигнале. Выходной сигнал передается обратно на считывающее устройство, радиоэлектронное устройство, определяющее пространственно-временное положение кодовых рефлекторов в ПАВ-датчике радиолокационными методами, после чего обрабатывается и выносится решение о состоянии контролируемого электрооборудования [4].

Заключение

Представленная система может обеспечить безопасную работу электрооборудования. Весомые преимущества: высокая помехозащищенность, устойчивость к воздействию электромагнитных помех, большая дальность считывания по радиоканалу, долговечность (неизменность характеристик в процессе эксплуатации), отсутствие необходимости сервисного обслуживания – делают радиоканальные системы контроля температуры подходящими для применения их в диагностике неисправностей высоковольтных сетей электроснабжения.

Библиографический список

1. Рожкова Л. Д. Нагревание электрооборудования. URL: <https://forca.com.ua/knigi/navchannya/nagrevanie-elektrooborudovaniya.html> (дата обращения 03.02.2021).
2. Мечин Д. Ю. Современные методы диагностики температуры электрооборудования. URL: <http://szte.ru/ru/article/view?slug=sovremennyye-metody-diagnostiki-temperatury-elektrooborudovaniya> (дата обращения 14.02.2021).
3. Кэмби Г. Датчик поверхностных акустических волн. URL: https://wikichi.ru/wiki/Surface_acoustic_wave_sensor (дата обращения 18.02.2021).
4. Козлов А. Н. Диагностика электрооборудования высокого напряжения: учеб. пособие. Благовещенск, 2013. 44 с. URL: https://irbis.amursu.ru/DigitalLibrary/AmurSU_Edition/6925.pdf (дата обращения 03.02.2021).

УДК 004.42

М. И. Судьин, В. А. Кадывкин

студенты кафедры проблемно-ориентированных вычислительных комплексов

К. Б. Гурнов – доцент, кандидат технических наук – научный руководитель

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ПЛИС – QUARTUS

В учебном процессе студенты выполняют лабораторные работы. С целью подготовки методических указаний к дисциплине «Электронные промышленные устройства» были изучены методы программирования на языке Verilog HDL и метод схемного моделирования в программируемой интегральной логической среде программирования (ПЛИС) Quartus.

Среда Quartus и структура ПЛИС используются для создания конфигурируемых цифровых электронных схем. В отличие от обычных цифровых микросхем, логика работы ПЛИС не определяется при изготовлении, а задается посредством программирования. Программируется ПЛИС с помощью таких языков, как Verilog, VHDL, AHDL, и символьного моделирования.

Язык Verilog – это язык описания аппаратуры, используемый для описания цифровых элементов, входящих в состав более сложной системы, определяющей логику работы ПЛИС. Разработчики Verilog HDL сделали его синтаксис очень похожим на синтаксис языка C, что упрощает его освоение [1].

Схемное моделирование подразумевает, что элементы для схемы уже созданы и их можно применить из каталога библиотеки. Программный пакет Quartus позволяет преобразовать код программы в элемент для графического представления.

В разрабатываемых лабораторных работах будут представлены этапы моделирования схем: • логических элементов (И, ИЛИ, стрелка Пирса, штрих Шеффера, исключающее ИЛИ, равнозначность); • комбинационной схемы; • триггеров (D-триггер, T-триггер, RS-триггер, JK-триггер); • устройств на основе уже изученных элементов (регистры, шифраторы, дешифраторы, мультиплексоры, демультиплексоры, счетчики). Будут присутствовать элементы программирования на языке Verilog и анализ работы программы при помощи временных диаграмм. Студентам будет предложено ознакомиться с программным пакетом Quartus и изучить его интерфейс с дальнейшим применением для моделирования схем и построения временных диаграмм.

Для примера была реализована разработка мультиплексор с помощью языка Verilog HDL на основе выполнения поставленных задач (рис. 1).

```

1  module mixel_f
2  (
3      clk,
4      data_in1,
5      data_in0,
6      data_out,
7      es1
8  );
9
10     input  clk;
11     input  es1;
12     input  [2:0] data_in0;
13     input  [2:0] data_in1;
14     output reg [2:0] data_out;
15
16     always @(posedge clk)
17     begin
18         if (es1==1'b1) data_out <= data_in1;
19         else data_out <= data_in0;
20     end
21
22 endmodule

```

Рис. 1. Код программы на языке Verilog HDL для реализации мультиплексора

Для графического представления код был преобразован в элемент схемы (рис. 2).

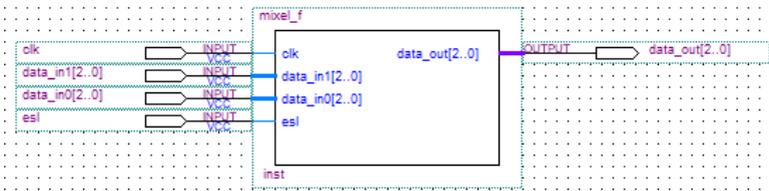


Рис. 2. Код, преобразованный в элемент для графического представления

После преобразования кода программы в элемент схемы был произведен анализ работы написанной программы (рис. 3).

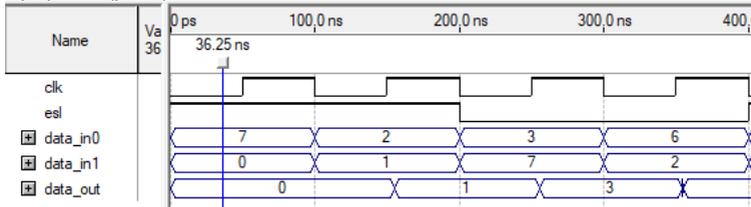


Рис. 3. Временная диаграмма работы мультиплексора

На основе описанной процедуры разрабатываются методические указания для выполнения лабораторных работ по дисциплине «Электронные промышленные устройства».

Библиографический список

1. Хабр. Зачем язык Verilog программисту микроконтроллеров. URL: <https://habr.com/ru/post/212507/> (дата обращения: 27.01.2021).

УДК 621.38

Е. О. Фаталиева

студент кафедры проблемно-ориентированных вычислительных комплексов

В. А. Килимник – кандидат технических наук, начальник научно-исследовательского отдела биотехнических проблем – научный руководитель

УСТРОЙСТВО ДЛЯ ОЦЕНКИ ОТНОСИТЕЛЬНОЙ ВЛАЖНОСТИ КОЖНОГО ПОКРОВА

В статье представлена разработка устройства для оценки относительной влажности кожного покрова, которое позволит уточнить диагноз и выбрать метод лечения больных с острой сердечной недостаточностью. Устройство также позволяет регистрировать относительную влажность, температуру и атмосферное давление окружающего воздуха.

В настоящее время для ФГБУ «Всероссийский центр экстренной и радиационной медицины имени А. М. Никифорова» МЧС России, для действующего врача Владимира Николаевича Хирманова, заведующего кардиологическим отделением, особую актуальность приобрела проблема снятия показаний влажности кожи. В связи с этим возникла потребность в устройстве, которое способно фиксировать эти значения.

Острая сердечная недостаточность (далее СН) вызывает катастрофическое ухудшение кровообращения, нередко – повышение влажности кожного покрова. Такое патологическое состояние в момент заболевания требует экстренного диагностирования, оценки причин и принятия срочных мер. При несвоевременном и недостаточном оказании помощи высока вероятность летального исхода и необратимого течения заболевания – инвалидности, связанной с частичной или полной потерей функций центральной нервной системы и периферийных органов. Проблемой является однозначное определение и классификации патологий СН.

В 1970–1980-е гг. использовался инвазивный метод оценки с помощью катетера, который вводится в систему кровообращения, практически в сердце или крупные сосуды, чтобы оценивать параметры кровообращения. Но он устарел по причинам отсутствия возможности в нужный момент катетеризировать сердце. Требуется ускоренный метод, который на практике возможен только в специальных условиях, когда больной уже находится в операционной или в хорошо оснащенном отделении интенсивной терапии, а таких не много. Экстренная ситуация может возникнуть где и когда угодно: дома, на улице, на работе, в машине скорой помощи, где нет условий для замеров внутрисердечных показателей. Классификация острой СН предусматривает определение ее тяжести в зависимости от состояния периферических тканей (ладоней, лица, ушей, стоп).

Классификация СН необходима для обеспечения индивидуального подхода к каждому пациенту, для наблюдения за ним в процессе оказания помощи, для полного понимания причин возникновения физиологических изменений. С целью диагностики и определения лечения требуется устройство контроля влажности. Его общая структурная схема представлена на рис. 1.

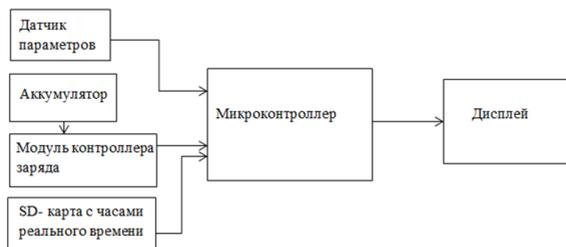


Рис. 1. Структурная схема устройства для оценки относительной влажности в изолированном объеме

Принцип действия устройства можно представить следующим образом: датчик параметров, подключенный к микроконтроллеру (МК), определяет уровень заложенных параметров (относительная влажность, температура, атмосферное давление), после чего на дисплее отображается информация с действительными значениями показателей. Все данные записываются (с указанием времени) на SD-карту. Аккумулятор обеспечивает требуемую длительность работы системы. Модуль контроллера заряда позволяет пользователю обеспечить зарядку аккумулятора.

Датчик корпусируется таким образом, что при прикосновении к коже пациента создается замкнутый воздушный объем. Как только влажность и температура этого объема перестают меняться, можно считать, что условия для оценки относительной влажности обеспечены. Соответственно, относительную влажность замкнутого воздушного объема можно считать равной относительной влажности кожи. Модуль датчика BME280 фирмы Bosch Sensortec [1] (рис. 2) предназначен для измерения влажности, температуры и атмосферного давления.

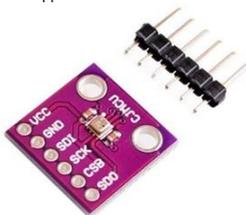


Рис. 2. Датчик влажности, температуры и давления BME280

Основные характеристики измеряемых показателей датчика: диапазон измерения влажности 20...95 %, пределы измеряемого давления: 225...825 мм рт. ст., температурный диапазон 0...65 °С. Точность измерения всех параметров не ниже 5 %.

В качестве управляющего модуля используется МК фирмы Atmel типа Atmega 328P [2]. Для его программирования существует большое количество библиотек, что ускоряет процесс разработки макета устройства. При разработке использовалась плата Arduino Nano, построенная на базе данного МК и позволяющая его программировать по интерфейсу USB. Визуализация данных обеспечивается применением OLED символьного индикатора (16 символов на две строки). Выбранный индикатор дает хорошую видимость при любой освещенности и под любым углом обзора [3]. В качестве источника питания используется Li-Pol аккумулятор LP103450-PCM емкостью 2 000 мАч с рабочим напряжением 4,2...3,6 В [4]. Контроль заряда и защита аккумулятора осуществляются с помощью модуля TP4056 [5]. Для дальнейшей обработки параметров с функцией сохранения данных и их привязки ко времени применяется модуль часов реального времени (ЧРВ) с интерфейсом для SD-карты [6]. На рис. 3 представлен макет устройства.



Рис. 3. Устройство для оценки относительной влажности кожного покрова

Процедура состоит из двух этапов. На первом измеряются параметры окружающей среды (температура, относительная влажность, атмосферное давление). После их фиксации переходят к этапу измерения влажности кожного покрова. Датчик находится в корпусе из биосовместимого материала. Измерение происходит путем наложения датчика на поверхность участка кожи, относительную влажность которой необходимо замерить. После установления показаний влажности среды внутри датчика осуществляется их регистрация на SD-карту с привязкой ко времени.

Вопрос стерильности корпуса датчика при измерении у разных пациентов решается путем применения одноразовых чехлов.

Таким образом, разработанное устройство соответствует требованиям медицинских сотрудников и после технических испытаний может быть применено для научно-исследовательских работ в клинической практике. Использование устройства позволит уточнить поставленный диагноз больным с острой сердечной недостаточностью и скорректировать лечебный процесс.

Библиографический список

1. Датчик BME280. URL: <https://3ddiy.ru/wiki/arduino-datchiki/datchik-bme280/> (дата обращения: 23.03.2021).
2. Микроконтроллер Arduino Nano. URL: <http://arduino.ru/Hardware/ArduinoBoardNano> (дата обращения: 23.03.2021).
3. Дисплей OLED. URL: <https://www.winstar.com.tw/ru/products/oled-module/oled-character-display/weh001602a.html> (дата обращения: 23.03.2021).
4. Аккумулятор LP103450. URL: <https://www.chipdip.ru/product/lp-103450-pcm> (дата обращения: 23.03.2021).
5. Контроль заряда TP4056. URL: <https://mysku.ru/blog/china-stores/65087.html> (дата обращения: 23.03.2021).
6. Модуль регистрации и хранения данных. URL: <https://arduino-kit.ru/product/modul-registratsii-i-hraneniya-dannyih-plata-data-loggera-dlya-arduino> (дата обращения: 23.03.2021).

УДК 004.934

М. А. Цыганова

студент кафедры медицинской радиоэлектроники

Е. П. Виноградова – старший преподаватель – научный руководитель

СОВРЕМЕННЫЕ СИСТЕМЫ РАСПОЗНАВАНИЯ РЕЧИ

Системы распознавания речи – востребованное направление исследований с учетом широкого спектра их применения. Это может пригодиться при наборе текста диктовкой, голосовом поисковом запросе, управлении техническим средством через команды, работе пользователей со специальными возможностями, управлении умными приборами. Одна из возможностей – интеграция с медицинской информационной системой, например в автоматизированное рабочее место терапевта. Это позволит сократить время приема, автоматизировать рабочие процессы и облегчить выполнение врачом профессиональных обязанностей, что положительно скажется на его психоэмоциональном состоянии.

Во многих проектах по всему миру задействованы системы распознавания речи, но чаще всего они дорогостоящи, представлены коммерческими компаниями и плохо поддерживают русский язык. Проблему можно решить созданием собственного проекта по обработке голосовых данных, пользуясь системами с открытым кодом или приобретая готовые коммерческие решения. В данной работе была поставлена цель оценить крупнейшие системы распознавания речи для выбора оптимальной, на основании которой можно создать полноценное рабочее место врача с голосовым вводом. Язык системы – русский, профиль – медицинский.

Были выбраны четыре системы с открытым кодом – CMU Sphinx (pocketsphinx/sphinx4) [1], Kaldi [2], Julius [3], iAros [4], пять с закрытым – Dragon Mobile SDK [5], Mozilla DeepSpeech [6], Google Speech Recognition [7], Yandex SpeechKit Cloud [8], Microsoft Speech API [9]. Системы были проанализированы по двум группам параметров: технические критерии и общие критерии. Технические включали точность и скорость распознавания речи, общие – возможность работать офлайн, язык системы, доступ и ограничения. Технические критерии были рассмотрены только для систем с открытым кодом. Для оценки точности использовались метрики Word Recognition Rate (WRR), Word Error Rate (WER), для оценки скорости – Speed Factor (SF).

WER – производная от расстояния Левенштейна; рассчитывается по формуле

$$WER = \frac{(S + D + I)}{N},$$

где S – число операций замены слов, D – число операций удаления слов, I – число операций вставки слов, N – общее количество слов.

WRR описывает точность распознавания слова и находится по формуле

$$WRR = 1 - WER.$$

SF – показатель отношения времени распознавания к длительности распознаваемого сигнала:

$$SF = \frac{T_{расп}}{T},$$

где $T_{расп}$ – время распознавания сигнала, T – длительность, измеряется в долях реального времени. Чем показатель меньше, тем выше скорость.

Результаты сравнения систем представлены в табл. 1.

Таблица 1

Результаты сравнения

Система	WER, %	WRR, %	SF
CMU Sphinx	22,7	78,6	0,5

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ

Система	WER, %	WRR, %	SF
Kaldi	6,5	93,5	0,6
Julius	23,1	76,9	1,3
iAtrios	16,1	83,9	2,1

Sphinx, Kaldi показали отличные технические результаты: первая система имеет лучшую скорость распознавания речи, но среднюю точность, вторая – лучшую точность распознавания и второй результат по скорости. Однако Kaldi поддерживает только английский язык [1, 2].

Julius имеет средние технические показатели и поддерживает только японский язык. Также система использует инструментарий Hidden Markov Model Toolkit (HTK), принадлежащий Microsoft, поэтому применять ее можно только в личных целях [3].

Система iAtrios подходит для распознавания как речи, так и рукописного варианта текста, что выгодно ее отличает, но не подходит для исследования, поддерживает английский и испанский языки. Показала посредственные технические результаты [4].

Mozilla DeepSpeech – перспективный проект, который был закрыт в 2021 г. Mozilla свернула разработку DeepSpeech и объявила о программе грантов для финансирования ряда инициатив, демонстрирующих способности приложений для DeepSpeech [5, 6].

Google Speech Recognition плохо поддерживает офлайн, но содержит несколько языков. Ценовая политика: до 60 мин в месяц бесплатно, далее 0,006 \$ за 15 с речи. Каждый запрос округляется до числа, кратного 15 [7].

Система Microsoft Speech API содержит большое количество подробно разобранных примеров и шаблонов, технология легко внедряется и поддерживается Microsoft, но полноценно может использоваться только на Windows Server [9].

Dragon Mobile SDK, Yandex SpeechKit Cloud – коммерческие проекты с существенными ограничениями. Yandex SpeechKit Cloud – отечественная разработка, рассчитана на русский язык (также поддерживает турецкий), обладает большим словарем и технологией распознавания непрерывной речи. Система умеет распознавать только следующие типы данных: короткие запросы – фразы (3–5 слов) на различные темы, адреса, имена, даты, числа, заказы и короткие тексты длиной не более 2–3 предложений; ценовая политика: 400 руб. за 1 000 запросов, первый месяц бесплатно.

Таким образом, на данный момент готовой системы распознавания речи для создания автоматизированного места врача на русском языке не существует. Имеется несколько подходов, которые помогут реализовать эту идею. Первый – с помощью системы с открытым кодом создать собственный проект, добавлять туда русский язык и словарь медицинских терминов. Для такой работы подойдут системы с высокими техническими результатами, например Sphinx, Kaldi. Второй подход заключается в освоении коммерческих решений. Его плюсом будет работа с русскоязычными системами, например Yandex SpeechKit Cloud, минусом – ограничения по словам, доступу, стоимости и трудности внедрения в специализированную медицинскую сферу.

Библиографический список

1. CMU Sphinx Open Source Speech Recognition. URL: <https://cmusphinx.github.io> (дата обращения: 01.02.2021).
2. Kaldi ASR. URL: <https://kaldi-asr.org> (дата обращения: 01.02.2021).
3. GitHub. URL: <https://github.com/julius-speech/julius> (дата обращения: 01.02.2021).
4. iATROS. URL: <https://www.prlht.upv.es/wp/> (дата обращения: 23.03.2021).
5. Dragon Mobile SDK. URL: <https://developer.nuance.com/public> (дата обращения: 02.02.2021).
6. GitHub. URL: <https://github.com/mozilla/DeepSpeech> (дата обращения: 17.03.2021).
7. Google Speech Recognition. URL: <https://cloud.google.com/speech-to-text?hl=ru> (дата обращения: 05.02.2021).
8. Yandex SpeechKit Cloud. URL: <https://cloud.yandex.ru/services/speechkit> (дата обращения: 20.02.2021).
9. Microsoft Speech API (SAPI) 5.4. URL: [https://docs.microsoft.com/en-us/previous-versions/windows/desktop/ee125663\(v=vs.85\)](https://docs.microsoft.com/en-us/previous-versions/windows/desktop/ee125663(v=vs.85)) (дата обращения: 20.02.2021).

УДК 621.38

А. А. Чеkmенева

студент кафедры проблемно-ориентированных вычислительных комплексов

В. А. Килимник – кандидат технических наук, начальник научно-исследовательского отдела биотехнических проблем – научный руководитель

УСТРОЙСТВО ДЛЯ ОЦЕНКИ КИСЛОРОДА И УГЛЕКИСЛОГО ГАЗА В ЗАМКНУТОМ ОБЪЕМЕ С ЭКСПЕРИМЕНТАЛЬНЫМ ЖИВОТНЫМ

Один из важнейших факторов существования живых организмов – их газообмен с окружающей средой. Его основу составляют поглощение кислорода и выделение углекислого газа, происходящие при дыхании [1, 2]. Эти процессы настолько интенсивны, что изменения концентрации кислорода и углекислого газа вследствие дыхания достигают нескольких процентов ($> 3\%$) от суммарного состава выдыхаемого воздуха [2]. Представления о выделении живыми организмами некоторых газообразных веществ на уровне микроконцентраций активно развиваются в настоящее время благодаря прогрессу в создании высокочувствительных и высокоточных аналитических методов. При исследовании газообмена человека наиболее актуален анализ следов газообразных веществ в выдыхаемом воздухе, так как анатомическое строение легких специально приспособлено для высокоэффективного обмена газами между воздухом и кровью [3].

Устройство для оценки кислорода и углекислого газа было разработано для проведения научно-исследовательских работ по физиологии дыхания, его общая структурная схема представлена на рис. 1.

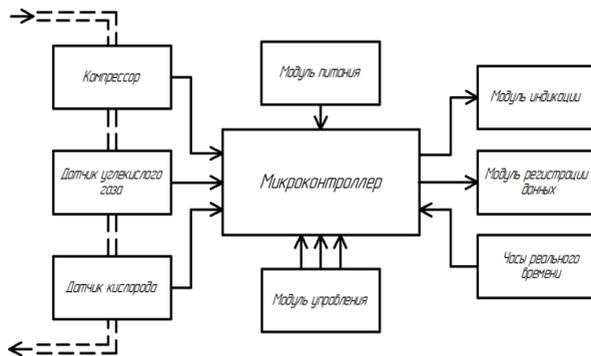


Рис. 1. Структурная схема устройства для оценки кислорода и углекислого газа в замкнутом объеме с экспериментальным животным

Для оценки содержания углекислого газа используется датчик (рис. 2), принцип работы которого основан на поглощении газом инфракрасных лучей. Анализируемый воздух, находясь в небольшой камере, подвергается облучению инфракрасным лучом. Сначала осуществляется замер интенсивности без оптического устройства. Затем луч, проходя через смесь газов и светофильтр, доходит до считывающего датчика. После определения двух значений, прибор по разнице определяет концентрацию углекислого газа в воздухе. В данный модуль встроен датчик температуры и влажности. Благодаря двухканальному принципу измерения концентрации углекислого газа он автоматически компенсирует долгосрочную девиацию.

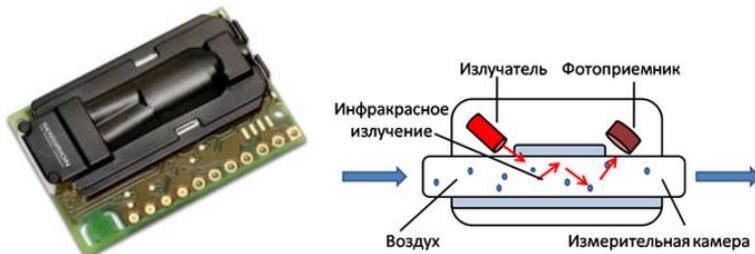


Рис. 2. Датчик углекислого газа и принцип его работы

Для оценки содержания кислорода используется датчик Oksik-11 (рис. 3). В основу функционирования электрохимического сенсора заложен амперометрический метод определения концентрации, при поступлении кислорода через поры мембраны на поверхность измерительного (рабочего) электрода происходит регенерация подвергающегося анализу газа (кислорода). Вследствие отдачи электронов катодом и принятия их анодом – преобразователем кислорода вырабатывается выходной сигнал постоянного напряжения.

Электрохимический преобразователь концентрации кислорода Oksik-11 является измерительным элементом, применяющимся только с принудительным отбором пробы (требуется внешний побудитель потока газа).



Рис. 3. Датчик кислорода

Требования к микроконтроллеру (далее МК) определялись возможностью реализации алгоритма оценки значений параметров окружающей среды, их обработки, индикации и регистрации. Для комплексного устройства был выбран МК типа Atmega328P. Данный процессор широко применяется в устройствах серии Arduino, что позволяет существенно ускорить разработку и изготовление макета устройства.

Для индикации оцениваемых параметров служит графический OLED индикатор 128×64, который был выбран исходя из габаритов, малого энергопотребления и наличия программных драйверов для подключения к МК [5].

Для создания анализируемого потока воздуха (газа) применяется компрессор типа SC3101PM, отличающийся малыми габаритами, требуемой производительностью и низким напряжением питания [6].

В качестве источника питания используется Li-Pol аккумулятор LP103450-PCM емкостью 2 000 мАч с рабочим напряжением 4,2–3,6 В. Контроль заряда и защита аккумулятора осуществляется с помощью модуля TP4056.

Целесообразно наличие возможности регистрации получаемых данных о газовом составе с привязкой их ко времени и факту внешнего воздействия. Для регистрации временных параметров углекислого газа и кислорода применяется плата, содержащая часы реального времени (ЧРВ) и интерфейс для подключения SD-карты. Для включения/выключения прибора и создания метки используются мембранные кнопки.

Устройство применяется для лабораторных исследований, на рис. 4 представлен его внешний вид.



Рис. 4. Внешний вид устройства для оценки кислорода и углекислого газа в замкнутом объеме с экспериментальным животным

На рис. 5 представлена установка, которая позволяет оценивать пробу выдыхаемого воздуха у экспериментального животного. На ней проводились исследования по поведению крысы в условиях гипоксии, при этом содержание кислорода уменьшалось от 21 до 10 %. Показания датчика углекислого газа в тех же условиях менялись до 1 000 до 4 000 ppm. В перспективе планируется провести исследование с уровнем углекислого газа до 10^4 ppm и наблюдать физиологические показатели крыс при введении различных дополнительных газов.

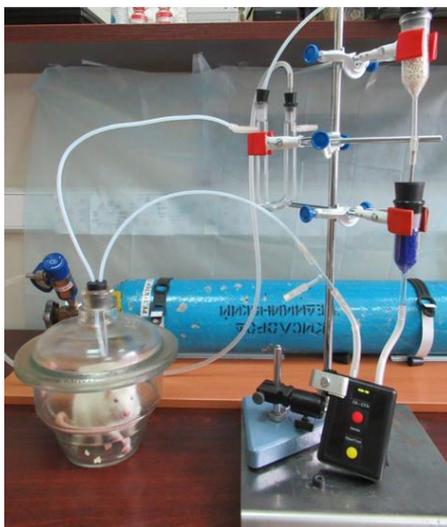


Рис. 5. Экспериментальная установка для оценки пробы выдыхаемого воздуха у экспериментального животного

Библиографический список

1. Шмидт Р., Тевс Г. Физиология человека. М.: Мир, 1996. 237 с.
2. Особенности внешнего дыхания крыс при развитии острой гипоксии с гиперкапнией и ее коррекции новым антигипоксантом металлокомплексной природы / Д. В. Сосин, А. В. Евсеев, В. А. Правдивцев, П. Д. Шабанов // Обзоры клин. фармакологии и лекарств. терапии. 2014. Т. 12. С. 36–42.
3. Степанов Е. В. Методы высокочувствительного газового анализа молекул-биомаркеров в исследованиях выдыхаемого воздуха // Тр. Ин-та общей физики им. А. М. Прохорова. 2005. № 61. С. 43.
4. Датчик кислорода. URL: <https://www.eksis.ru/catalog/sensors-and-mikrokompressory/product293.php> (дата обращения: 05.05.2021).
5. Sensirion SCD30 Sensor Module. <https://docs.rs-online.com/2d49/0900766b816b6f9d.pdf> (дата обращения: 05.05.2021).
6. Компрессор типа SC3101PM. URL: <https://iarduino.ru/lib/d4ecfd50802cb0e4431ae33301548ac1.pdf> (дата обращения: 05.05.2021).

УДК 621.38

А. А. Чекменева

студент кафедры проблемно-ориентированных вычислительных комплексов

В. А. Килимник – кандидат технических наук, начальник научно-исследовательского отдела биотехнических проблем – научный руководитель

УСТРОЙСТВО ДЛЯ ОЦЕНКИ ВОДОРОДА В ВЫДЫХАЕМОМ ВОЗДУХЕ

Вопрос диагностики патологии кишечника является важным и актуальным, учитывая значительную распространенность кишечной патологии, сложности в инструментальной диагностике заболеваний кишечника. Как известно, существует корреляция между продукцией водорода в кишечнике и выделением водорода в выдыхаемом воздухе [1, 2]. Таким образом, отмечается взаимосвязь между перевариванием пищи в тонком кишечнике и количеством выделяемого водорода.

Водородный дыхательный тест – неинвазивный метод исследования, позволяющий определить концентрацию водорода в выдыхаемом воздухе, который используется для выявления синдрома избыточного роста числа бактерий (СИБР) в тонкой кишке, для диагностики непереносимости лактозы и непереносимости фруктозы [3].

В рамках данной работы было проведено исследование двух разных по принципу работы и функционалу устройств (электрохимического и полупроводникового датчиков).

Устройство с электрохимическим сенсором исследовалось на примере анализатора «Лактофан-2» (рис. 1) [4]. В основе принципа измерения лежит процесс электролиза. Датчик состоит из трех электродов – рабочего (газопроницаемой пленки с нанесенным катализатором из драгоценного металла), референсного и интегрирующего, которые размещены в пластиковом корпусе с электролитом. В датчике используется потенциостатическая цепь, которая обеспечивает постоянное напряжение между рабочим и референсным электродами. Ток, возникающий в ходе химических реакций на рабочем и интегрирующем электродах, пропорционален концентрации измеряемого газа. Преимущества использования такого датчика: линейный выходной сигнал, высокая точность и хорошая воспроизводимость результатов. Основной недостаток – небольшой срок службы.



Рис. 1. Анализатор выдыхаемого водорода «Лактофан2»

Для исследования полупроводникового сенсора послужил портативный анализатор дыхания Food marble aire [5] (рис. 2) – персонализированное устройство, предназначенное для обнаружения и самоконтроля пищевой непереносимости.



Рис. 2. Портативный анализатор дыхания Food marble aire

В устройстве aire применяется полупроводниковый датчик газа. В данном типе датчиков используется полупроводник с металлоксидным напылением, сопротивление которого меняется при контакте с газом. Датчик состоит из нагревательной спирали и покрытия из оксида металла, нанесенного на трубку, по краям трубки находятся контакты из драгоценного металла, предназначенные для измерения сопротивления. При попадании газа на поверхность датчика он окисляется, что приводит к уменьшению электрического сопротивления, которое преобразуется в концентрацию газа. Преимущества использования таких датчиков: чувствительность к низким концентрациям, долговременная стабильность, недостаток – низкая селективность.

Для исследований выбран полупроводниковый датчик фирмы Figaro TGS8100 [6] – маломощный, малогабаритный со сроком службы больше, чем у других полупроводниковых датчиков.

Микросхема датчика TGS8100 содержит сенсорный элемент с нагревателем. Датчик требует два входа напряжения: напряжение нагревателя (V_H) и напряжение измерительной цепи (V_C). Напряжение нагревателя (V_H) подается на встроенный нагреватель для поддержания чувствительного элемента при температуре, оптимальной для измерения. Напряжение цепи (V_C) применяется для измерения напряжения (V_{OUT}) на нагрузочном резисторе (R_L), который соединен последовательно с датчиком. Напряжение постоянного тока требуется для питания измерительной цепи, так как для нее необходима определенная полярность. Значение нагрузочного резистора (R_L) должно быть выбрано исходя из требований, приводимых в техническом описании датчика. Сопротивление датчика (R_S) рассчитывается с измеренным значением $V_{OUT}(V_{RL})$ по следующей формуле:

$$R_S = \left(\frac{V_C}{V_{RL}} - 1 \right) \cdot R_L.$$

В экспериментальную установку входит микроконтроллер, в данном случае используется плата Arduino nano на базе микроконтроллера ATmega328P. На рис. 3 представлен стенд для исследований датчиков фирмы Figaro TGS8100.

Исследование проводилось на трех датчиках Figaro TGS8100 и медицинском приборе «Лактофан2». Измерения на «Лактофан2» служили опорными точками, по которым сравнивались показания датчиков TGS8100.

Полученный экспериментально график зависимости концентрации водорода, согласно прибору «Лактофан2», имеет линейную зависимость от показаний исследуемых датчиков. Для нормировки данных снятых с датчиков вводится коэффициент, который вычисляется по следующей формуле:

$$K = \frac{R_0}{R_S} - 1,$$

где R_0 – начальное сопротивление датчика (воздушная среда), R_S – сопротивление датчика при подаче водорода.

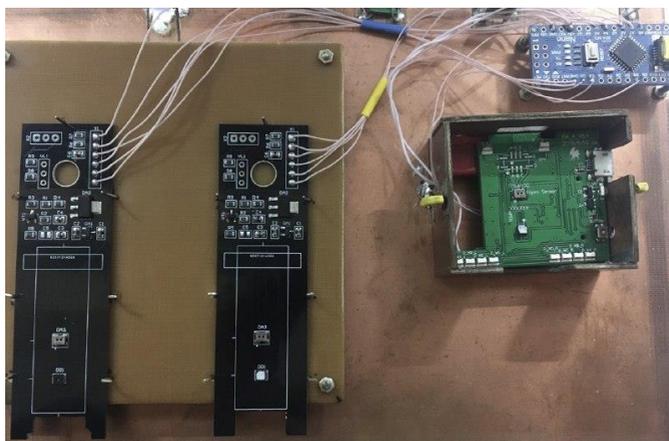


Рис. 3. Стенд для исследований датчиков фирмы Figaro TGS8100

На рис. 4 представлены результаты исследований для трех датчиков TGS8100.

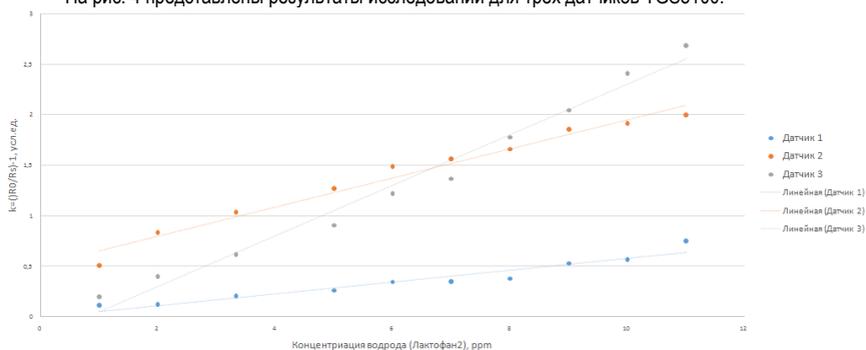


Рис. 4. Результаты исследований датчиков TGS8100

Библиографический список

1. Степанов Ю. М., Будзак Я., Коненко И. С. Дыхательный водородный тест в диагностике патологии желудочно-кишечного тракта. URL: <https://cyberleninka.ru/article/n/dyhatelnyy-vodorodnyy-test-v-diagnostike-patologii-zheludochno-kishechnogo-trakta/viewer> (дата обращения: 05.05.2021).
2. Актуальные вопросы проведения и оценки результатов водородных дыхательных тестов в положениях современных консенсусов / И. Л. Кляритская, В. В. Кривой, И. А. Иськова, Т. А. Цапьяк. URL: <https://cyberleninka.ru/article/n/aktualnye-voprosy-provedeniya-i-otsenki-rezultatov-vodorodnyh-dyhatelnyh-testov-v-polozheniyah-sovremennykh-konsensusov/viewer> (дата обращения: 05.05.2021).
3. Синдром избыточного бактериального роста в тонкой кишке: клиническое значение, критерии диагностики и терапевтическая тактика / И. В. Маев, Ю. А. Кучерявый, Д. Н. Андреев, Н. Ю. Ивашкина. URL: <https://cyberleninka.ru/article/n/sindrom-izbytochnogo-bakterialnogo-rosta-v-tonkoy-kishke-klinicheskoe-znachenie-kriterii-diagnostiki-i-terapevticheskaya-taktika/viewer> (дата обращения: 05.05.2021).

4. Анализатор выдыхаемого водорода Лактофан2. URL: <http://www.amamed.ru/index.php?i=34> (дата обращения: 05.05.2021).
5. Устройство для дыхательного теста Aire. URL: <https://www.foodmarble.com/> (дата обращения 05.05.2021)
6. Датчик TGS8100. URL: <https://www.figarosensor.com/product/docs/TGS8100%280914%29.pdf> (дата обращения: 05.05.2021).

Компьютерные технологии и программная инженерия

УДК 681.51

В. А. Блудов

студент кафедры компьютерных технологий и программной инженерии

Ю. А. Скобцов – доктор технических наук, профессор – научный руководитель

ПОСТРОЕНИЕ ПУТИ РОБОТА В СЛОЖНОЙ СРЕДЕ НА ОСНОВЕ ГИБРИДНОГО МУРАВЬИНОГО АЛГОРИТМА

Предложен муравьиный алгоритм (МА) планирования пути робота в сложной среде. Сначала на основе MakinK (теория графов) и алгоритма Дейкстры строится путь робота, проходящий примерно посередине между объектами. Далее проблема сводится к одномерной задаче определения смещений относительно срединных линий, которая решается с помощью МА. Определены основные формулы МА для поставленной задачи: вероятность выбора следующей позиции, коррекция концентрации искусственного феромона. Оптимизация пути робота проводится по длине пути. Компьютерные эксперименты по построению путей роботов в сложной среде показывают эффективность такого подхода, прежде всего быстродействия алгоритма

Для повышения эффективности, «интеллектуальности» и расширения функций технических систем широко применяются методы искусственного интеллекта (ИИ). МА представляют собой одно из наиболее перспективных направлений ИИ и хорошо себя показали прежде всего при решении задач комбинаторной оптимизации. В данной работе рассмотрены проблемы применения МА в робототехнике при планировании пути робота в сложной среде. Область применения роботов в настоящее время стремительно расширяется, они используются во многих отраслях промышленности, в опасной и агрессивной среде, в условиях военных действий, космических исследованиях и т. д. Применение роботов, особенно интеллектуальных, позволяет решать многие сложнейшие научно-технические задачи: от исследования далеких планет до диагностики и лечения заболеваний человеческого организма. Поэтому проблемам проектирования и обучения роботов посвящено огромное число исследований, выполняемых как в России, так и за рубежом

Цель работы – повышение эффективности планирования пути робота в сложной среде путем применения современных методов искусственного интеллекта. Для ее достижения предложен гибридный МА, который позволяет строить субоптимальный путь робота в сложной среде с препятствиями.

Существующие методы планирования путей и навигации роботов можно разделить на несколько классов [1].

- I. Методы на основе графов:
 - диаграмма видимости;
 - диаграмма Вороного;
 - вероятностная дорожная карта [2];
 - метод быстро исследующих случайных деревьев.
- II. Методы на основе клеточной декомпозиции:
 - алгоритм Дейкстры;
 - метод распространения волнового фронта.
- III. Методы потенциальных полей:
 - метод искусственных потенциалов;
 - метод виртуального силового поля;
 - метод гистограммы векторного поля.
- IV. Оптимизационные методы:
 - плоские модели;
 - смешанное целочисленное линейное программирование;

- двоично-целочисленное программирование.
- V. Методы на основе интеллектуальных технологий:

- муравьиный алгоритм;
- искусственная нейронная сеть;
- роевой алгоритм;
- реактивные методы.

Основные недостатки обозначенных классов.

- I. Методы на основе графов.

Методы на основе графов или деревьев удобно применять в статической окружающей среде, поскольку построение графа подразумевает использование полной информации об окружающей среде, которая должна быть известна заранее. Наличие подвижных препятствий, например другого движущегося робота, в этих методах трудно поддается интерпретации и не позволяет строить граф.

- II. Методы на основе клеточной декомпозиции.

Один из наиболее существенных недостатков алгоритмов на основе клеточной декомпозиции состоит в ограниченном количестве вариантов направления, вызванных структурой сетки (только восемь вариантов соединения в двумерном случае).

- III. Методы потенциальных полей.

Эти методы имеют существенный недостаток – возможное существование локальных минимумов и колебательные движения в полосообразной области.

- IV. Оптимизационные методы.

Классические методы оптимального управления базируются на нетривиальных аналитических вычислениях. Поэтому подход, основанный на них, эффективен для простых, в частности линейных, систем, а для сложных нелинейных его реализовать труднее

Рассматривается реализация гибридного муравьиного алгоритма построения пути робота в сложной среде [3]. Сначала на основе алгоритма Дейкстры и теории MAKLINK-графов строится путь робота, проходящий через середины MAKLINK-линий между объектами. Далее проблема сводится к одномерной задаче – определению смещений относительно срединных линий, которая решается с помощью МА, позволяющего найти субоптимальный путь за адекватное количество времени.

Укрупненный алгоритм.

1. Построить модель свободного пространства на основе MAKLINK-теории [2] и найти субоптимальный путь, свободный от столкновений, с применением алгоритма Дейкстры:

1.1) находим все линии, которые соединяют угол, принадлежащий многоугольному препятствию, с углами других препятствий (включая текущее препятствие и границы карты);

1.2) удаляем избыточные свободные линии, чтобы построить свободное пространство, у которого ребра – свободные линии, стороны препятствий и стены границы среды. Свободное пространство должно быть выпуклым многоугольником и иметь максимальную площадь;

1.3) находим средние точки оставшихся свободных линий, используем их в качестве вершин пути, нумеруем их. Соединения средних точек, которые принадлежат той же выпуклой области, образуют сеть;

1.4) будем искать оптимальный вектор параметров (смещений от срединных линий) $h_1^*, h_2^*, \dots, h_d^*$, который дает минимальную длину пути $L = \sum_{i=0}^d \text{length}\{P_i(h_i), P_{(i+1)}(h_{(i+1)})\}$, d – количество узлов графа;

1.5) при этом используем дискретизацию с некоторым шагом h_j . Тогда муравьиный алгоритм при пересечении каждой свободной линии должен выбрать величину (дискретную!) смещения h_j относительно средней точки свободной линии. То есть аналогом выбора следующей вершины в графе является выбор дискретного значения смещения h_j .

2. Определить параметры алгоритма:

- начальная точка, от которой муравьи начинают движение;
- конечная точка, которую муравьи должны достигнуть в процессе построения пути;
- ρ -коэффициент испарения феромона;

- τ_0 – начальное значение концентрации феромона в каждой вершине;
- V – коэффициент, определяющий степень влияния параметра «видимости»;
- m – размер популяции муравьев в одном поколении;
- NC – количество поколений;
- q_0 – регулируемый параметр.

Определить одномерный массив Path_k – путь с d элементами, в которых y -координаты d узлов (не включая точки S и T) такой, что k -й муравей проходит в каждой итерации в указанном порядке. Массив Path_k используется для обозначения пути k -го муравья.

3. Установить счетчик итераций (поколений) $t = 1$ и определить максимальное число итераций; поместить m муравьев в начальную точку S .

4. Выбрать узел на линии h_i для муравья k , используя формулу

$$j = \left\{ \begin{array}{l} \text{argmax}\{\tau_{iu}(t) \cdot [n_{iu}]^\beta\}, \text{если } q \leq q_0 \\ J, \text{если } q > q_0 \end{array} \right\},$$

$$P^k_j(t) = \left\{ \begin{array}{l} \frac{\tau_{ij}(t) * n_{ij}^\beta(t)}{\sum_{w=1}^m \tau_{iw}(t) * n_{iw}^\beta(t)} \\ 0 \end{array} \right\}.$$

Первое выражение выбирается, если данный сегмент реализуем, второе – если нет, $\tau_{ij}(t)$ – концентрация феромона в i -й вершине, $n_{iw}(t)$ – фактор «видимости» узла, определяемый как

$$\frac{1.1 - |y_{ij} - y^*_{ij}|}{1.1},$$

В уравнениях y_{ij} – y -координата узла, y^*_{ij} – значение, соответствующее построенному оптимальному пути в предыдущей итерации, α – коэффициент, определяющий степень влияния феромона, β – коэффициент, определяющий степень влияния параметра «видимости», m – количество особей в популяции.

Передвинуть муравья k в этот узел и запомнить y -координату этого узла в i -м элементе массива Path_k.

5. Выполнить локальную коррекцию концентрации феромона согласно формуле

$$\tau_{ij}(t) = (1 - \rho) * \tau_{ij}(t) + \rho * \tau_0.$$

6. Передвинуть каждого муравья от его текущего положения к глобальной цели T .

7. Для каждого муравья ($k = 1, 2, \dots, m$):

а) в соответствии с массивом Path_k получить значения параметров $\{h_1^k, h_2^k, \dots, h_d^k\}$;

б) в соответствии с полученными значениями $\{h_1^k, h_2^k, \dots, h_d^k\}$ определить положения точек пути $\{P_1^k, P_2^k, \dots, P_d^k\}$ на соответствующих МАКЛИНК линиях;

в) вычислить длину пути L_k , построенного k -м муравьем в соответствии с формулой

$$L = \sum_{i=0}^d \text{length}\{P_i(h_i), P_{(i+1)}(h_{(i+1)})\}.$$

8. Сравнить полученные m путей и найти самый короткий путь T^t на текущей итерации t . Сравнить T^t с T^+ – оптимальным путем, который был получен за предыдущие $(t - 1)$ итераций и обо-

значить лучший путь через T^+ ; запомнить значения параметров смещения h , соответствующие T^+ , в $\{h_1^k, h_2^k, \dots, h_d^k\}$.

9. Установить каждый элемент Path_k в 0 для $k = 1, 2, \dots, m$.

10. Выполнить глобальную коррекцию концентрации феромона в соответствии с формулой

$$\tau_{ij}(t) = (1 - \rho) * \tau_{ij}(t) + \rho * \Delta\tau_0,$$

$$\Delta\tau_{ij}(t) = \frac{1}{L^+},$$

где L^+ – длина лучшего пути, построенного лучшим муравьем в этом туре.

11. Установить $t = t + 1$.

12. Если $t < NC$, все m муравьев не построили одинаковый тур, поместить всех муравьев в начальную точку S , продолжить поиск пути. Если $t > NC$ и все m муравьев построили одинаковый тур или $T = NC$, то оптимальным будет путь T^+ , а оптимальными значениями параметров являются – $\{h_1^k, h_2^k, \dots, h_d^k\}$.

В результате написания реализации данного алгоритма и его исследования были подобраны следующие оптимальные параметры: $\tau_0 = 0,0002$; $\rho = 0,1$; $\beta = 2$; $m = 10$; $NC = 200$; $q_0 = 0,85$.

Полученные результаты представлены на рис. 1. Толстыми черными линиями описаны границы препятствий. В результате построен оптимальный путь – красный. Заметим, что построение началось с субоптимального пути (желтого), а точность решения зависит от величины дискретизации h_i (размера шага смещения).

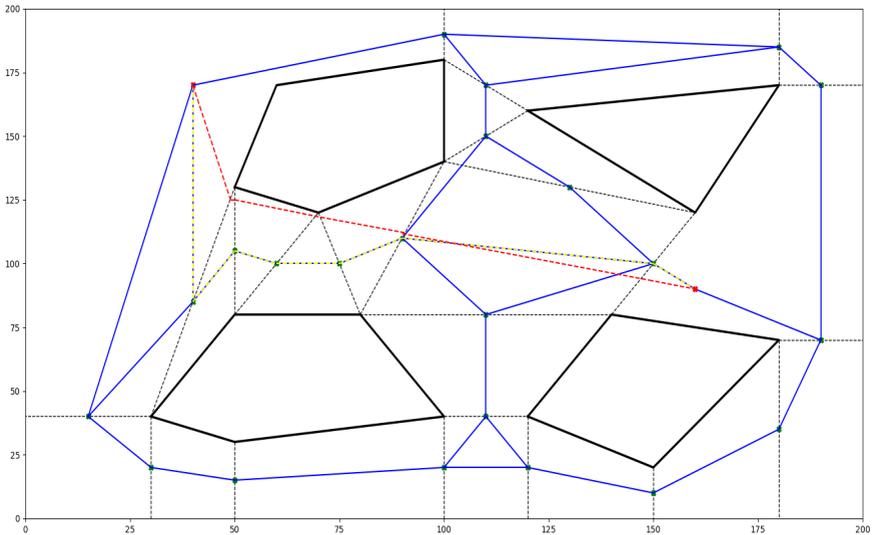


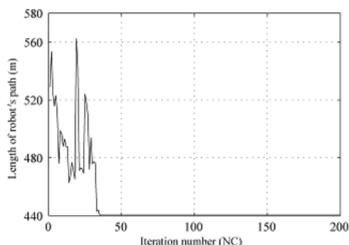
Рис. 1. Гибридный МА

Интересно сравнить результаты построения путей с применением генетических и муравьиных алгоритмов (табл. 1). Параметры ГА следующие: кодирование вещественное, отбор элитарный, размер популяции $m = 50$, вероятность кроссинговера $p_c = 0,6$, вероятность мутации $p_m = 0,05$.

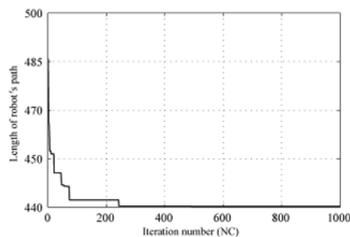
Построение пути с применением разных алгоритмов

Показатель	Муравьиный алгоритм	Генетический алгоритм
Время выполнения одной итерации, с	0,000 59	0,000 67
Количество итераций, необходимых для сходимости	175	912
Время поиска оптимального решения, с	0,103 3	0,611 0

Скорость сходимости алгоритмов представлена на рис. 2.



Муравьиный алгоритм



Генетический алгоритм

Рис. 2. Скорость сходимости алгоритмов

Реализован и исследован гибридный муравьиный алгоритм, построенный на основе алгоритма Дейкстры, классического муравьиного алгоритма и теории графов MAKLINK. В целом гибридный MA показал себя как эффективный алгоритм поиска, позволяющий найти оптимальный путь робота всего за несколько итераций, что сравнимо быстрее генетических аналогов.

Библиографический список

1. Лю В. Методы планирования пути в среде с препятствиями (обзор). URL: <https://doi.org/10.24108/mathm.0118.0000098> (дата обращения: 01.05.2021).
2. Janet J. A., Luo R. C., Kay M. G. The essential visibility graph: An approach to global motion planning for autonomous mobile robots // IEEE Intern. conf. on robotics and automation, Nagoya, Japan, May 21–27, 1995. Proc. Vol. 2. N. Y., 1995. P.1958–1963. URL: <https://ieeexplore.ieee.org/document/526023> (дата обращения: 01.05.2021).
3. Mohamad M. M., Dunnigan M. W., Taylor N. K. Ant colony robot motion planning // Computer as a tool: Intern. conf. on EUROCON 2005, Belgrade, Serbia, Nov. 21–24, 2005. N. Y., 2005. Vol. 1. P.213–216. URL: <https://ieeexplore.ieee.org/document/4129846> (дата обращения: 30.04.2021).

УДК 004.415

С. А. Гришин

магистрант кафедры компьютерных технологий и программной инженерии

А. В. Фомин – кандидат технических наук – научный руководитель

СОЗДАНИЕ ОТКАЗОУСТОЙЧИВОЙ ИНФРАСТРУКТУРЫ НА ОСНОВЕ KUBERNETES ДЛЯ СБОРА ИНФОРМАЦИИ С СЕТЕВЫХ УСТРОЙСТВ

Введение

В последнее время получает широкое распространение схема распространения приложения с помощью контейнеров. Использование контейнеризации вместо виртуальных машин или физических серверов позволяет увеличить скорость запуска, уменьшить размер программных компонентов. Дополнительно контейнеризация улучшает безопасность, потому что процессы работают в изолированных средах и изменения или обновление в соседнем контейнере не влияют на другие [1].

Контейнеризация решает проблемы изолированности процессов, но не проблемы конфигурации инфраструктуры для программных проектов. В компаниях существуют проблемы так называемых «снежинок», когда каждый программный проект имеет уникальные требования и конфигурацию. Из-за различия конфигураций устранение проблем требует времени и часто простоя системы.

Для автоматизации развертывания, масштабирования контейнеризированных приложений и управления им было создано открытое программное обеспечение Kubernetes. Название происходит от греческого «рулевой или штурман». Google открыл исходный код Kubernetes в 2014 г. Kubernetes основывается на десятилетнем опыте работе Google с масштабными рабочими нагрузками в сочетании с лучшими в своем классе идеями и практиками сообщества.

Основа работы Kubernetes – применение декларативного подхода. От разработчика требуется указать, чего необходимо достичь, а не способы достижения. Помимо этого, могут быть задействованы императивные команды (create, edit, delete), которые позволяют непосредственно создавать, модифицировать и удалять ресурсы. Однако их не рекомендуется использовать для критически важных задач.

Kubernetes содержит ряд абстракций, которые представляют состояние системы: развернутые контейнеризованные приложения и рабочие нагрузки, связанные с ними сетевые и дисковые ресурсы и другую информацию о том, что делает кластер. Эти абстракции представлены объектами в API Kubernetes.

Для развертывания программного обеспечения в Kubernetes применяется база Linux-контейнеров (например, Containerd или CRI-O) и описание, сколько потребуется контейнеров и какое количество ресурсов для них. Само развертывание контейнеров происходит на основе рабочих нод — виртуальных или физических машин.

Kubernetes – удобный инструмент оркестрации контейнеров. Однако это решение не работает само по себе, без подготовки и дополнительных настроек. Например, пользователям придется решать вопросы по миграции схем баз данных или разбираться с обратной совместимостью API [2].

Kubernetes предоставляет следующие возможности.

Мониторинг сервисов и распределение нагрузки: Kubernetes может обнаружить контейнер, используя имя DNS или собственный IP-адрес. Если трафик в контейнере высокий, Kubernetes может сбалансировать нагрузку и распределить сетевой трафик, чтобы развертывание было стабильным.

Оркестрация хранилища: Kubernetes позволяет автоматически смонтировать систему хранения по Вашему выбору, такую как локальное хранилище, провайдеры общедоступного облака и др. [3–5]

Автоматическое развертывание и откаты: используя Kubernetes, можно описать желаемое состояние развернутых контейнеров и изменить фактическое состояние на желаемое. Например, Вы можете автоматизировать Kubernetes на создание новых контейнеров для развертывания, удаления существующих и распределения всех их ресурсов в новый контейнер.

Автоматическое распределение нагрузки: Вы предоставляете Kubernetes кластер узлов, который он может использовать для запуска контейнерных задач, указываете, сколько ЦП и памяти (ОЗУ)

требуется каждому контейнеру. Kubernetes может разместить контейнеры на узлах так, чтобы наиболее эффективно использовать ресурсы.

Самоконтроль: Kubernetes перезапускает отказавшие контейнеры, заменяет и завершает работу тех, которые не проходят определенную пользователем проверку работоспособности, и не показывает их клиентам, пока они не будут готовы к обслуживанию.

Управление конфиденциальной информацией и конфигурацией: Kubernetes может хранить и управлять конфиденциальной информацией, такой как пароли, OAuth-токены и ключи SSH. Вы можете развертывать и обновлять конфиденциальную информацию и конфигурацию приложения без изменений образов контейнеров и не раскрывая конфиденциальную информацию в конфигурации стека.

При развертывании Kubernetes Вы имеете дело с кластером. Кластер Kubernetes cluster состоит из набора машин, так называемых узлов, которые запускают контейнеризированные приложения. Кластер имеет как минимум один рабочий узел.

В рабочих узлах размещены поды, являющиеся компонентами приложения. Плоскость управления управляет рабочими узлами и подами в кластере. В промышленных средах плоскость управления обычно запускается на нескольких компьютерах, а кластер, как правило, развертывается на нескольких узлах, гарантируя отказоустойчивость и высокую надежность.

Плоскость управления поддерживает запись всех объектов Kubernetes в системе и запускает непрерывные циклы управления для обработки их состояния. В любое время циклы управления панели управления будут реагировать на изменения в кластере и работать, чтобы фактическое состояние всех объектов в системе соответствовало желаемому состоянию, которое указали.

На рис. 1 показана диаграмма кластера Kubernetes с тремя рабочими узлами для запуска приложений.

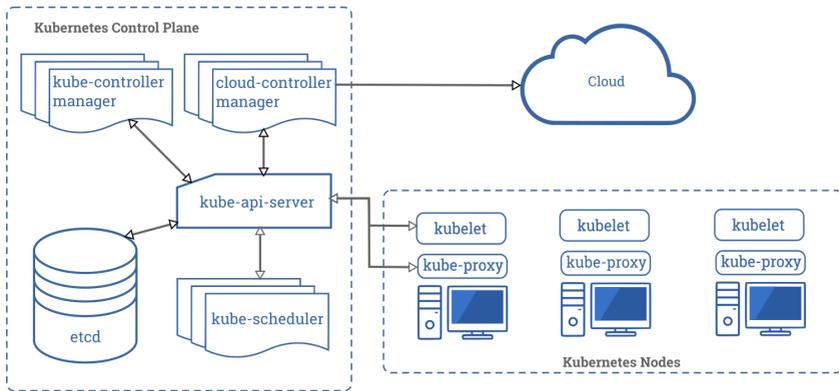


Рис. 1. Диаграмма кластера Kubernetes

Постановка задачи

Для решения проблемы по сбору параметров у сетевых устройств предполагается разработать архитектуру, показанную на рис. 2.

Слева на рис. 2 располагаются сетевые устройства двух мобильных операторов. Внутри рабочего контура (on-premises) располагаются рабочие узлы с модулями для сбора статистики с устройств (Collector). Общение с другими частями кластера происходит с помощью настройки Gateway на каждой из частей кластера.

Вся информация, полученная с устройств, передается с помощью системы сообщений (Message broker) к модулю сохранения информации в постоянное хранилище данных (Cloud Storage).

Далее пользователи системы (User 1, User 2) могут получать информацию о состоянии устройств и ее изменении во времени, используя браузер и REST API.

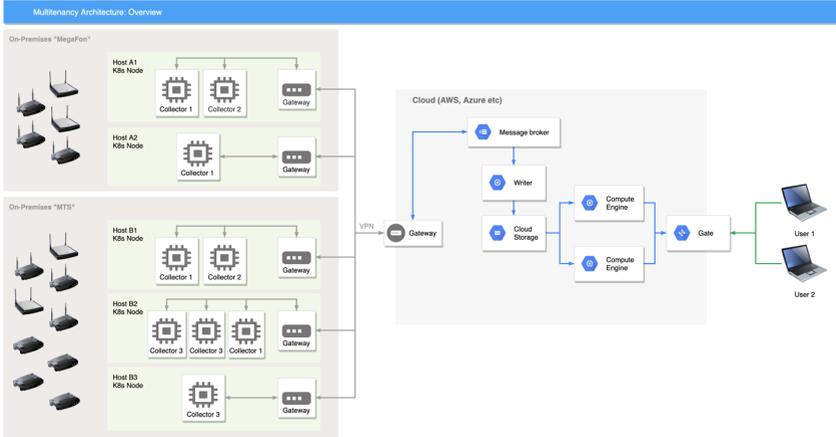


Рис. 2. Архитектура системы

Для реализации сервисов был выбран язык Go. Этот язык был разработан 2009 г. и начал применяться для написания интернет-приложений и инфраструктурных модулей. Docker и Kubernetes написаны на нем. Язык Go также набирает популярность относительно других языков. В 2017 г. по результатам Stack Overflow's 2017 developer survey он вошел в топ-5 в категории самый любимый и самый желанный для использования (рис. 3).

Most Loved, Dreaded, and Wanted Languages

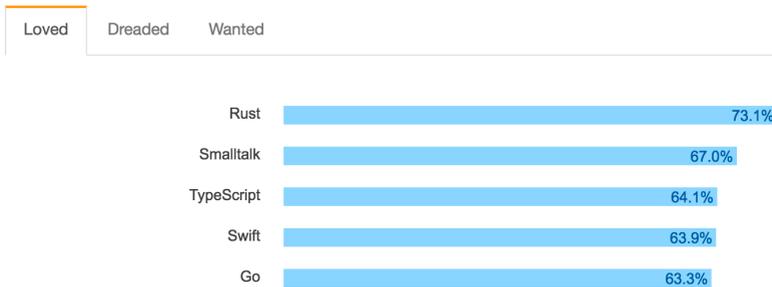


Рис. 3. Опрос на самый любимый язык Stack Overflow's 2017 developer survey

Заключение

Сбор информации с помощью применения Kubernetes позволяет решать проблемы отказоустойчивости по мониторингу инфраструктуры системы, динамически расширять архитектуру горизонтально за счет добавления новых рабочих узлов, не меняя архитектуру приложения. Данный подход для построения архитектуры является перспективным и уже широко применяется для решения проблем возрастающей нагрузки в больших IT-фирмах.

Библиографический список

1. Таненбаум Э., Узеролл Д. Компьютерные сети. СПб.: Питер, 2019. 960 с.

2. Ибрам Б., Хасс Р. Паттерны Kubernetes. Шаблоны разработки собственных облачных приложений. СПб.: Питер, 2020. 320 с.
3. Миллан И., Сейерс Э. Х. Docker на практике. М.: ДМК-Пресс, 2020. 516 с.
4. Донован А. А. А., Керниган Б. У. Язык программирования Go. М.: Диалектика, 2018. 432 с.
5. Дэвис К. Шаблоны проектирования для облачной среды. М.: ДМК Пресс, 2019. 388 с.

УДК 519.714

Д. В. Клин

магистрант кафедры компьютерных технологий и программной инженерии

А. В. Туманова – кандидат технических наук– научный руководитель

СИСТЕМА ПОМОЩИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ АНАЛИЗЕ ДВИЖЕНИЙ ФОНДОВОГО РЫНКА

Введение

Ранее мировой фондовый рынок был доступен ограниченному кругу лиц со специальным образованием. С развитием технологий банковского сектора торговля на мировых биржах открылась для каждого человека в любой точке мира. Она стала основной статьёй дохода большого количества людей, чему способствует внушительный финансовый оборот, достигающий триллионов долларов [1].

Фондовый рынок – одна из наиболее значимых частей современной рыночной экономики. Здесь путем разнообразных сделок покупаются и продаются акции, облигации и прочие ценные бумаги.

Основанный в 1896 г. Чарльзом Доу и Эдвардом Джонсом индекс DJIA является средневзвешенным по цене, что означает, что акции влияют на индекс пропорционально их цене за акцию. DJIA отслеживает 30 крупных публично торгуемых акций США. Он был разработан как средний показатель для экономики в целом.

S&P500, также называемый просто S&P, был введен Standard & Poor's в 1957 г. в качестве индекса фондового рынка для отслеживания стоимости 500 крупных корпораций, зарегистрированных на Нью-Йоркской фондовой бирже (NYSE) и NASDAQ Composite. Коллекция акций, составляющих S&P, предназначена для представления общей структуры экономики США. Его точное сочетание и веса различных секторов корректируются по мере изменения экономики, а некоторые акции добавлялись и убирались с течением времени. Сегодня S&P500 (рис. 1) – главный индекс Америки после NASDAQ и Dow Jones [2].

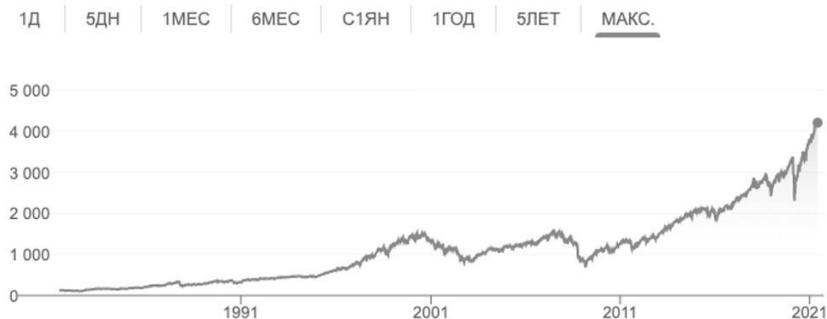


Рис. 1. Показатели индекса S&P500

Акция – один из самых популярных видов ценных бумаг. Каждая акция – это «кусочек» какой-либо компании. Например, покупая акцию «Татнефти», Вы приобретаете долю в компании. Компания зарабатывает деньги и нуждается в дополнительных средствах для развития новых проектов. С этой целью она приходит на биржу и продает часть своего бизнеса в виде акций. Люди или крупные фонды покупают акции, компания получает деньги на развитие.

Облигации – второй вид самых популярных ценных бумаг на фондовом рынке. Их покупка есть одалживание денег под процент. Одолжить деньги можно все тем же компаниям или государству. В первом случае это будут корпоративные облигации, во втором – государственные. Последние называются ОФЗ (облигации федерального займа). На фондовом рынке присутствует возможность приобрести ОФЗ РФ, тем самым дать в деньги в долг под процент Российской Федерации.

Тикер – это обозначение ценной бумаги, товара или валюты на бирже. Он нужен, чтобы найти любой финансовый продукт у брокера и купить. Тикер акции состоит из основных букв в названии компании: у «Газпрома» на Московской бирже – GAZP, у Apple на американской NASDAQ – AAPL. Тикеры, одной и той же компании могут различаться на разных биржах. Тикер облигации выглядит как 12-значный код из цифр и букв. Он обычно включает краткую информацию по конкретной облигации. Тикеры есть у всех финансовых инструментов, которые можно купить на бирже, например у валюты: RUR – российский рубль, USD – доллар США, EUR – евро. У драгоценных металлов и нефти тоже есть тикеры.

Треjder – человек, совершающий сделки покупки или продажи на фондовом рынке.

Очевидно, что акции продолжают свой рост, хотя едва ли настолько же бурный, как во второй половине XX в. Акции – хорошая защита долгосрочных вкладов от инфляции, но все же не полная гарантия победы над ней при коротком горизонте планирования около 5 лет. Первое десятилетие XXI в., отмеченное двумя сильными кризисами, показало, что не стоит формировать свой инвестиционный портфель исключительно на основе акций.

Постановка задачи

Торговля на фондовом рынке требует больших временных затрат на отслеживание текущих цен, отчетов компаний. Мониторинг большого количества событий влияет на психическое состояние человека, что влечет эмоциональные покупки или продажи. Возможности отвлечься нет, так как можно пропустить наступление какого-либо события, которое может сильно повлиять на текущий портфель акций и принести большой убыток. Решением проблемы может стать создание системы помощи в принятии решений о покупке или продаже на основе движений фондового рынка.

Торговая система – совокупность различных инструментов анализа фондового рынка, позволяющая принимать решения о покупке или продаже ценных бумаг.

Торговля на фондовом рынке по определенной системе дает ряд преимуществ:

- система исключает эмоции из процесса торговли, что позволяет инвестору избежать многих ошибок;

- система гарантирует последовательный подход к торговле, т. е. трейдер должен следовать всем сигналам, предписанным общим набором условий торговой системы. Только в этом случае использование системы будет иметь смысл;

- система обычно предоставляет трейдеру метод управления рисками. Это очень важно, так как в отсутствие плана по ограничению убытков даже одна неудачная сделка может привести к большим убыткам. Правильно выстроенная система либо будет оснащена точными правилами остановки при возникновении убытков, либо будет определять условия для открытия противоположной позиции при достаточно неблагоприятном движении цен.

В реальной торговле результатом работы торговой системы будет определение направления тренда и диапазона цен на определенный период. Торговая система также должна включать правила расположения стоп-заявок (stop-loss) для минимизации возможных убытков, так как ни одна система не может дать 100% гарантии.

Можно выделить ряд задач-этапов, которые по мере реализации программы необходимо решить (пройти):

- выбор рынка. Перед началом разработки торговой системы нужно определиться, на каких рынках она будет работать. Рынки должны обладать хорошей ликвидностью (достаточным количеством желающих, купить или продать бумагу в данный момент времени), чтобы можно было открывать и закрывать позицию в любой момент, когда Вы этого захотите, и высокой потенциальной доходностью;

- определение рыночных условий (восходящий, нисходящий тренд, отсутствие тренда) и временного периода, в котором осуществляется торговля (кратко-, средне- или долгосрочного);

- определение способа открытия позиции – вход. Открытие позиции должно происходить после выполнения трех различных условий, каждому из которых соответствует определенный компонент системы:

- идентификаторы направления (тренда) – показывают, в каком направлении движется рынок: вверх, вниз или вбок;

- идентификаторы установки – сообщают о том, что вскоре возможно открытие позиции;

- спусковые механизмы входа – непосредственно сообщают о том, что позиция должна быть открыта прямо сейчас, в настоящий момент;

— выбор способа закрытия позиции – выход. Процесс выхода также определяется несколькими условиями, каждому из которых соответствует определенный компонент системы. Для этого потребуются компоненты, относящиеся к группам, отвечающим за выход из позиции:

- выходы управления начальным риском. Обычно это фиксированное число или процент от капитала. Этот выход ограничивает максимально возможные потери и не позволяет потерять больше запланированной суммы;

- выходы смены тренда. Это сигналы для закрытия позиции в случае разворота тренда. Они закрывают позицию, когда становится ясно, что направление движения рынка было выбрано неправильно, и логика ее открытия больше недействительна;

- выходы защиты прибыли. Как только позиция стала достаточно прибыльной, эти выходы предохраняют ее от превращения в убыточную;

- выходы, сохраняющие основную прибыль. Как только прибыль стала существенно велика, начинают работать эти выходы с целью сохранения основной части прибыли.

При принятии решений используются также индикаторы – математически рассчитанная на основе цены и/или объема величина, которая применяется для прогнозирования ценовых изменений курса ценных бумаг.

У каждого человека, пришедшего на фондовый рынок, разные горизонты инвестирования, разный уровень дохода, цели инвестирования, отношение к риску. Универсальность будущей системы можно обеспечить динамической возможностью использования математического аппарата. Система будет отвечать за сбор данных, их хранение, а также отправку уведомлений пользователю о наступлении того или иного события. Решения о совершении покупки или продажи, а также о том, в каком случае получать уведомление, принимает пользователь.

Диаграмма бизнес-процесса приложения представлена на рис. 2 [3, 4].

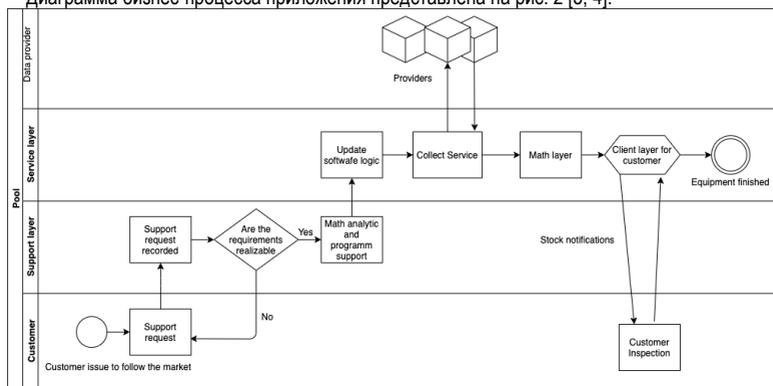


Рис. 2. Диаграмма итерационного процесса

Результаты архитектурного моделирования системы помощи

Взаимодействие пользователя с основными частями системы представлено в виде диаграммы последовательностей на рис. 3.

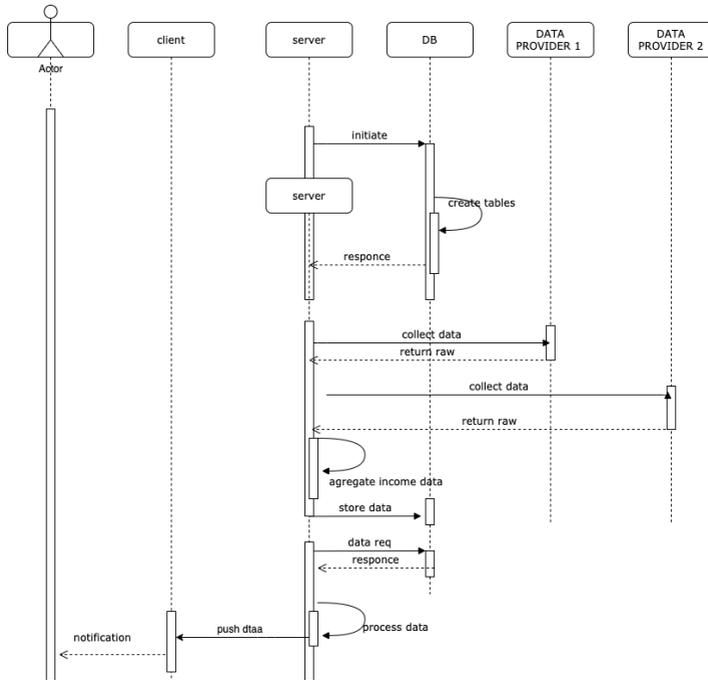


Рис. 3. Диаграмма последовательностей

Результаты тестового запуска системы управления

Результаты сбора данных с биржи Санкт-Петербург по тикеру TCS с уникальным идентификатором BBG005DXJS36 показаны в табл. 1.

Таблица 1

Результаты сбора данных с биржи

Время	Идентификатор	Цена	Объем торгов
2021-05-28 07:00:00.000000 +00:00	BBG005DXJS36	54,19	1 621
2021-05-28 07:01:00.000000 +00:00	BBG005DXJS36	55,00	886
2021-05-28 07:02:00.000000 +00:00	BBG005DXJS36	54,46	360
2021-05-28 07:03:00.000000 +00:00	BBG005DXJS36	54,31	332

Запуск тестового пользовательского алгоритма на основе полученных данных и получение соответствующих уведомлений показаны на рис. 4.

	helpMeTrade BUY-TCS-UP-5%	5  itsTimeToTrade 13:21
	helpMeTrade SKIP-BUY-TCS	5  itsTimeToTrade 13:21
	helpMeTrade BUY-NET-UP-3%	5  itsTimeToTrade 13:21
	helpMeTrade SKIP-BUY-NET	5  itsTimeToTrade 13:21
	helpMeTrade SELL-NET-DOWN-7%	5  itsTimeToTrade 13:21

Рис. 4. Получение уведомлений пользователем

Заключение

Анализ результатов работы программы, полученных в ходе тестирования построенной системы помощи принятия решений, свидетельствует о работоспособности сконструированного приложения. Основная проблема при составлении подобного рода систем заключается в подборе математических алгоритмов под текущую экономическую ситуацию и их последующей интеграции в систему [5, 6]. Тем не менее перспективность данного подхода к совершению сделок обусловлена исключением одного из главных факторов, влекущих потерю денег на фондовой бирже, – человеческих эмоций и возможностью охвата более широкого круга биржевых инструментов путем автоматизации сбора и анализа данных.

Библиографический список

1. Таненбаум Э., Уэзеролл Д. Компьютерные сети. СПб.: Питер, 2019. 960 с.
2. Далио Р. Большие долговые кризисы. Принципы преодоления. СПб.: Питер, 2020. 498 с.
3. Охтилев М. Ю. Системы искусственного интеллекта и их применение в автоматизированных системах мониторинга состояния сложных организационно-технических объектов: монография. СПб.: ГУАП, 2018. 261 с.
4. Донован А. А. А., Керниган Б. У. Язык программирования Go. М.: Диалектика, 2018. 432 с.
5. Бенджамин Г. Полное руководство по стоимостному инвестированию. СПб.: Питер, 2019. 568 с.
6. Бенджамин Г., Додд Д. Анализ ценных бумаг. СПб.: Питер, 2016. 880 с.

УДК 519.714

Г. С. Лёвин

магистрант кафедры компьютерных технологий и программной инженерии

С. И. Колесникова – профессор, доктор технических наук, доцент – научный руководитель

МОДЕЛИРОВАНИЕ НЕЛИНЕЙНОГО ВЕКТОРНОГО УПРАВЛЕНИЯ ПРОДОЛЬНЫМ ДВИЖЕНИЕМ САМОЛЕТОМ-АМФИБИЕЙ

Постановка задачи

Самолет как объект управления неустойчив в продольном движении при воздействии внешних возмущений. Для его стабилизации пилоту необходимо вручную контролировать и регулировать требуемые параметры. Часто из-за таких проблем возникают авиакатастрофы. Чтобы уменьшить риски, целесообразно пользоваться автопилотом. Но необходимы огромные усилия и труд, чтобы запрограммировать автопилот для различных режимов движения в условиях непредсказуемого воздействия окружающей природной среды. Поэтому моделирование полета летательных аппаратов до сих пор актуально.

На рис. 1 представлены возможные органы управления самолетом-амфибией (СА). Будем рассматривать только продольное движение.



Рис. 1. Органы управления самолетом-амфибией

Математическая модель объекта управления

Синтез нелинейного управления по-прежнему, несмотря на возрастающее число публикаций, задача нетривиальная и не вполне решенная в условиях возмущений, моделирующих не только неточность самой модели, но и наличие помех разной природы. Из результатов [1, 2] следует возможность аналитического конструирования системы управления, компенсирующей неслучайные возмущения по каналу управления. Дальнейшее изложение базовым для применяемого здесь алгоритма управления теоретического материала опирается на работы [3, 4].

Ставится задача определения векторного управления $u(x) = (u_1, u_2, u_3)$, являющегося основой собственно устройства автопилота, обеспечивающего стабилизацию управляемых переменных в окрестности заданных (целевых) значений. Для поиска управления будет рассматриваться следующая модель объекта управления [5] (табл. 1):

$$\begin{cases} \dot{x}_1(t) = b_1 x_2 x_3 - g \sin x_5 + a_1 u_1 + z_1; \\ \dot{x}_2(t) = -b_2 x_1 x_3 - g \cos x_5 + a_2 u_2 + z_2; \\ \dot{x}_3(t) = a_3 u_3 + z_3; \\ \dot{x}_4(t) = x_1 \sin x_5 + x_2 \cos x_5; \\ \dot{x}_5(t) = x_3; \\ \dot{x}_6(t) = x_1 \cos x_5 - x_2 \sin x_5. \end{cases} \quad (1)$$

Согласно методу нелинейной адаптации, на начальном этапе ставится задача расширения фазового пространства для преобразования в замкнутую систему на основе моделирования возмущений в виде [5, 6]:

$$\begin{cases} \dot{z}_1 = \eta_1(x_1 - V_0); \\ \dot{z}_2 = \eta_2(x_4 - H_0) \\ \dot{z}_3 = \eta_3(x_5 - \vartheta_0). \end{cases}$$

Таблица 1

Комментарии к переменным и параметрам объекта управления

Обозначение	Пояснение
x_1	горизонтальная скорость
x_2	вертикальная скорость
x_3	продольная угловая скорость
x_4	высота
x_5	угол тангажа
x_6	дальность полета
$a_j, b_j, \eta_j, \mu_j, i = \overline{1,3}, j = 1,2$	известные параметры
$u_k, k = \overline{1,3}$	выражения управления
$z_m, m = \overline{1,3}$	помехи

Пусть цель управления заключается в достижении заданных горизонтальной скорости (V_0), высоты (H_0) и угла тангажа (ϑ_0), соответственно:

$$\Psi^* = (\Psi_1^*, \Psi_2^*, \Psi_3^*), \text{ где } \Psi_1^* = x_1 - V_0 \rightarrow \min; \Psi_2^* = x_4 - H_0 \rightarrow \min; \Psi_3^* = x_5 - \vartheta_0 \rightarrow \min.$$

Метод аналитического конструирования агрегированных регуляторов (АКАР) и его обобщение – метод нелинейной адаптации

Рассмотрим алгоритм нелинейной адаптации на заданном многообразии.

Этап I – передача управления от координат x_2, x_3 к переменным φ_1, φ_2 .

Функционал качества будет следующим:

$$\Phi_1 = \int_0^\infty \sum_{i=1}^3 (\omega_i \Psi_i^l)^2 + (\Psi^l)^2 dt \rightarrow \min.$$

Функциональное уравнение метода АКАР:

$$\omega_i \dot{\Psi}_i^l + \Psi_i^l = 0, \quad \omega_i > 0, \quad i = \overline{1,3}. \tag{2}$$

В соответствии с процедурой метода АКАР введена первая совокупность макропеременных:

$$\Psi^l = (\Psi_1^l, \Psi_2^l, \Psi_3^l),$$

где

$$\Psi_1^l = x_1 - V_0 + \mu_1 z_1; \Psi_2^l = x_2 - \varphi_1(x_4, x_5, z_1, z_2, z_3); \Psi_3^l = x_3 - \varphi_2(x_4, x_5, z_1, z_2, z_3). \tag{3}$$

Решая систему основных функциональных уравнений с учетом (1)–(3), получим

$$\begin{aligned}\omega_1(b_1x_2x_3 - g \sin x_5 + a_1u_1 + z_1 + \mu_1\eta_1(x_1 - V_0)) + \Psi_1^I &= 0 \\ \omega_2(-b_2x_1x_3 - g \cos x_5 + a_2u_2 + z_2 - \dot{\varphi}_1(x_4, x_5, z_1, z_2, z_3)) + \Psi_2^I &= 0. \\ \omega_3(a_3u_3 + z_3 - \dot{\varphi}_2(x_4, x_5, z_1, z_2, z_3)) + \Psi_3^I &= 0\end{aligned}$$

Находим управляющие воздействия:

$$\begin{aligned}u_1 &= \left[-\Psi_1^I \omega_1^{-1} - b_1x_2x_3 + g \sin x_5 - z_1 - \mu_1\eta_1(x_1 - V_0) \right] a_1^{-1} \\ u_2 &= \left[-\Psi_2^I \omega_2^{-1} + b_2x_1x_3 + g \cos x_5 - z_2 + \dot{\varphi}_1(x_4, x_5, z_1, z_2, z_3) \right] a_2^{-1}. \\ u_3 &= \left[-\Psi_3^I \omega_3^{-1} - z_3 + \dot{\varphi}_2(x_4, x_5, z_1, z_2, z_3) \right] a_3^{-1}\end{aligned}$$

Этап II – декомпозиция.

Рассматриваем новую модель объекта управления:

$$\begin{cases} \dot{x}_4(t) = (V_0 - \mu_1z_1) \sin x_5 + \varphi_1 \cos x_5; \\ \dot{x}_5(t) = \varphi_2; \\ \dot{x}_6(t) = (V_0 - \mu_1z_1) \cos x_5 - \varphi_1 \sin x_5; \\ \dot{z}_1 = -\eta_1\mu_1z_1; \\ \dot{z}_2 = \eta_2(x_4 - H_0); \\ \dot{z}_3 = \eta_3(x_5 - \vartheta_0). \end{cases} \quad (4)$$

Функционал качества будет следующим:

$$\Phi_2 = \int_0^\infty \sum_{i=4}^5 (\omega_i \dot{\Psi}_i^II)^2 + (\Psi_i^II)^2 dt \rightarrow \min.$$

Функциональное уравнение метода АКАР

$$\omega_i \dot{\Psi}_i^II + \Psi_i^II = 0, \quad \omega_i > 0, \quad i = 4, 5. \quad (5)$$

Теперь необходимо задать вторую совокупность макропеременных:

$$\Psi^II = (\Psi_4^II, \Psi_5^II), \quad \text{где } \Psi_4^II = x_4 - H_0 + \mu_2z_2; \quad \Psi_5^II = x_5 - \vartheta_0 + \mu_3z_3. \quad (6)$$

Решая систему основных функциональных уравнений с учетом (4)–(6), получим:

$$\begin{aligned}(x_4 - H_0 + \mu_2z_2)' + (x_4 - H_0 + \mu_2z_2) &= 0 \\ (x_5 - \vartheta_0 + \mu_3z_3)' + (x_5 - \vartheta_0 + \mu_3z_3) &= 0\end{aligned}$$

Найдем «внутренние» управления φ_1, φ_2 :

$$\begin{aligned}\varphi_1 &= \left[-\Psi_4^II \omega_4^{-1} - (V_0 - \mu_1z_1) \sin x_5 - \mu_2\eta_2(x_4 - H_0) \right] (\cos x_5)^{-1} \\ \varphi_2 &= -\Psi_5^II \omega_5^{-1} - \mu_3\eta_3(x_5 - \vartheta_0)\end{aligned} \quad (7)$$

Находим частные производные с учетом (7) (табл. 2).

Вид частных производных переменной внутреннего управления

$\frac{\partial \varphi_1}{\partial x_4} = [-\omega_4^{-1} - \mu_2 \eta_2] (\cos x_5)^{-1}$	$\frac{\partial \varphi_2}{\partial x_4} = 0$
$\frac{\partial \varphi_1}{\partial x_5} = [-\Psi_4'' \omega_4^{-1} \sin x_5 - V_0 + \mu_1 z_1 - \mu_2 \eta_2 (x_4 - H_0) \sin x_5] (\cos^2 x_5)^{-1}$	$\frac{\partial \varphi_2}{\partial x_5} = -\omega_5^{-1} - \mu_3 \eta_3$
$\frac{\partial \varphi_1}{\partial z_1} = \mu_1 \operatorname{tg} x_5$	$\frac{\partial \varphi_2}{\partial z_1} = 0$
$\frac{\partial \varphi_1}{\partial z_2} = -\mu_2 \omega_4^{-1} (\cos x_5)^{-1}$	$\frac{\partial \varphi_2}{\partial z_2} = 0$
$\frac{\partial \varphi_1}{\partial z_3} = 0$	$\frac{\partial \varphi_2}{\partial z_3} = -\mu_3 \omega_5^{-1}$

При раскрытии φ_1, φ_2 управляющие воздействия принимают следующий вид:

$$u_1 = [-\Psi_1' \omega_1^{-1} - b_1 x_2 x_3 + g \sin x_5 - z_1 - \mu_1 \eta_1 (x_1 - V_0)] a_1^{-1}$$

$$u_2 = \left[-\Psi_2' \omega_2^{-1} + b_2 x_1 x_3 + g \cos x_5 - z_2 + \frac{\partial \varphi_1}{\partial x_4} \dot{x}_4 + \frac{\partial \varphi_1}{\partial x_5} \dot{x}_5 + \frac{\partial \varphi_1}{\partial z_1} \dot{z}_1 + \frac{\partial \varphi_1}{\partial z_2} \dot{z}_2 + \frac{\partial \varphi_1}{\partial z_3} \dot{z}_3 \right] a_2^{-1} \cdot (8)$$

$$u_3 = \left[-\Psi_3' \omega_3^{-1} - z_3 + \frac{\partial \varphi_2}{\partial x_4} \dot{x}_4 + \frac{\partial \varphi_2}{\partial x_5} \dot{x}_5 + \frac{\partial \varphi_2}{\partial z_1} \dot{z}_1 + \frac{\partial \varphi_2}{\partial z_2} \dot{z}_2 + \frac{\partial \varphi_2}{\partial z_3} \dot{z}_3 \right] a_3^{-1}$$

Подробнее с методом нелинейной адаптации можно ознакомиться в работе [6].

Результаты численного моделирования системы управления

Для моделирования полета СА, описание которого подчиняется системе обыкновенных дифференциальных уравнений (1) с синтезированными законами управления (8), разработана программа на основе MATLAB (версия R2018a), структура которой отображена на рис. 2.

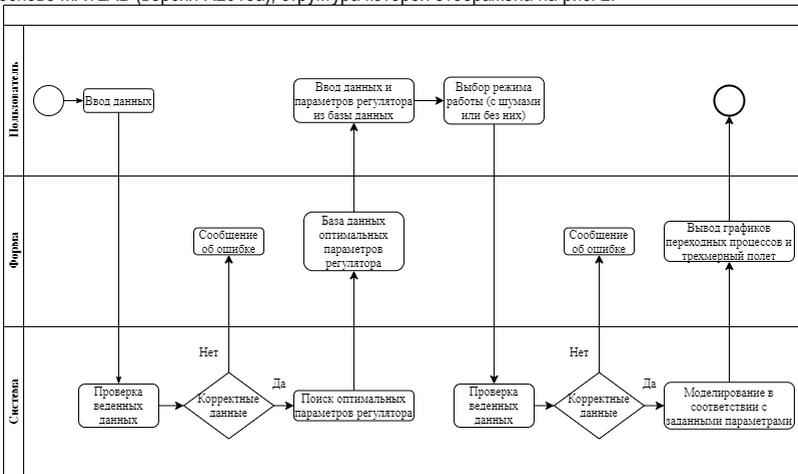


Рис. 2. Диаграмма сценария взаимодействия с программой

Диаграмма описывает поведение системы на запрос пользователя в различных условиях. На рис. 3, 4 приведены результаты моделирования.

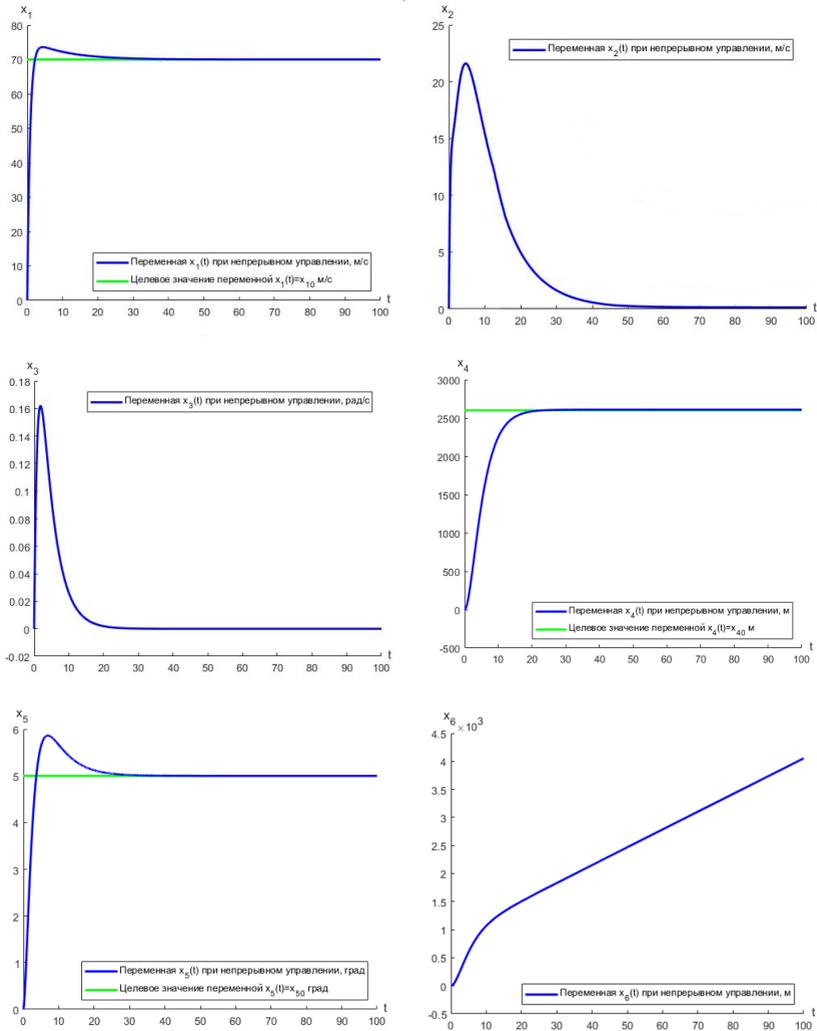


Рис. 3. Переходные процессы относительно горизонтальной, вертикальной, продольной угловой скорости, высоты, угла тангажа и дальности полета.

Из рис. 3 следует, что с течением времени обеспечивается затухание переходных процессов и происходит установление в минимум режимов целевых значений, что говорит об устойчивости системы управления.

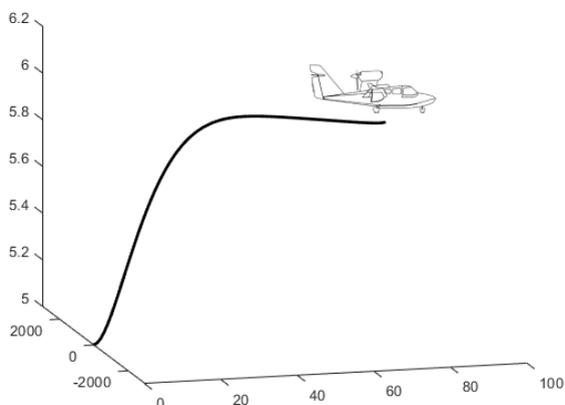


Рис. 4. Трехмерная модель полета СА

Из рис. 4 следует, что взлет и полет СА под воздействием построенного регулятора осуществляется с приемлемым качеством.

Заключение

Анализ результатов численного моделирования, полученных в ходе исследования, свидетельствует о работоспособности сконструированного регулятора. Основной проблемой при моделировании системы управления оказывается подбор параметров векторного регулятора $\omega_j, j = \overline{1,5}$, пропорциональных длительности переходного процесса и влияющих на его качество. Тем не менее перспективность подходов и методов теории управления, основанных на идеях нелинейной динамики и синергетики, которые позволяют эффективно решать задачи управления летательных аппаратов, вне сомнений и нуждается в исследовании для нелинейных многомерных объектов разной прикладной направленности.

Библиографический список

1. Колесников А. А. Синергетическая теория управления. М.: Энергоатомиздат, 1994. 344 с.
2. Колесников А. А. Синергетическая теория управления: концепции, методы, тенденции развития. // Изв. ЮФУ. Технические науки. 2001. № 5. С. 7–27.
3. Колесников А. А., Кобзев В. А. Динамика полета и управление: синергетический подход. Таганрог: Изд-во ТТИ ЮФУ, 2009. 198 с.
4. Колесников А. А., Кобзев В. А., Никитин А. И. Синергетический синтез законов векторного управления системы автоматической посадки самолета. // Изв. ЮФУ. Технические науки. 2011. № 6. С. 125–139.
5. Колесников А. А., Кобзев В. А., Нгуен Ф. Синергетический синтез системы управления движением самолетов-амфибий, функционирующих в экстремальных условиях // Изв. ЮФУ. Технические науки. 2010. № 5. С. 150–155.
6. Колесникова С. И. Синтез системы управления нелинейным объектом второго порядка с неполным описанием // Автоматика и телемеханика. 2018. № 9. С. 18–30.

УДК 004

И. А. Погодина

студент кафедры компьютерных технологий и программной инженерии

М. В. Фаттахова – кандидат физико-математических наук, доцент – научный руководитель

РЕШЕНИЕ ЗАДАЧИ CVRP С ПЛАВАЮЩИМИ ВРЕМЕННЫМИ ОКНАМИ

За последние несколько лет проблема экономии ресурсов при перевозке грузов стала важной задачей транспортной логистики. Разработка программных систем поддержки принятия решения, помогающих справиться с ней, требует серьезных исследований с целью изучения алгоритмов для выявления или разработки самого эффективного, пригодного для постоянного использования. Одна из задач в этой области – маршрутизация транспортного потока. Ее разрабатывают с 1959 г., добавляя различные ограничения, которые выведены как подклассы задачи. Сейчас существует несколько подклассов, например учитывающие грузоподъемность автомобилей или временные окна доставки товаров.

Задача маршрутизации транспортного потока представляет собой развитие задачи коммивояжера, отличие состоит в построении сразу нескольких замкнутых маршрутов, при этом необходимо посетить все вершины и вернуться в исходную, которую принято называть депо. За время исследования задачи маршрутизации транспорта (рис. 1) стало понятно, что она принадлежит к классу задач дискретной оптимизации, которые являются NP-трудными. У таких задач найти точное решение практически невозможно в силу ограничений мощностей технологий. Время нахождения решения зависит от входных данных полиномиально [1].



Рис. 1. Классическая ЗМТ

Рассмотрим следующую задачу маршрутизации: пусть имеется некоторое количество автомобильных заправочных станций (АЗС), топливо на которые поставляется с одной нефтебазы. Также имеется множество транспортных средств (ТС), расположенных в депо, при этом у каждого из них своя грузоподъемность. Необходимо построить такие маршруты для ТС, которые минимизируют суммарные транспортные издержки, при этом каждое ТС, посетив клиентов, возвращается в депо. Предположим, что составление маршрутов происходит утром, когда от каждого клиента (АЗС) предприятие (топливная база) получает данные об остатках топлива и его среднем расходе. Кроме того, АЗС формулируют важное требование – чтобы резервуары топлива не пересыхали. Соответственно, необходимо своевременно обеспечить АЗС топливом, для чего необходимо рассчитывать, к какому времени подвозить нефтепродукт.

Поставленная задача представляет собой объединение двух классов задач VRP (рис. 2):

- VRPTW – VRP with Time Windows (задача VRP с временными окнами) [2];
- CVRP – Capacitated VRP (задача VRP с учетом ограничения на вместительность машин)

[2].

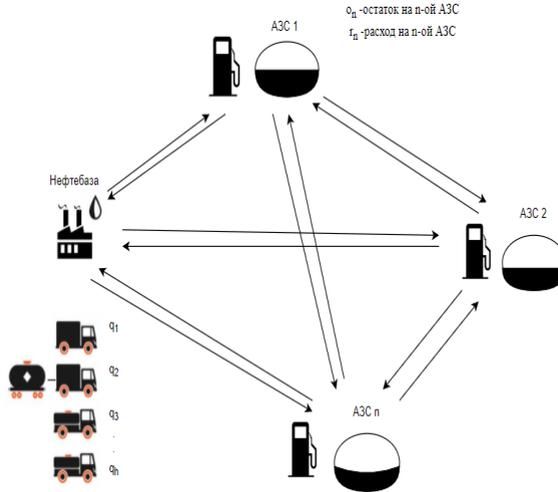


Рис. 2. Графическая интерпретация задачи SVRPFTW

Рассмотрим ориентированный граф $G=(N,E)$ с множеством узлов $N=D \cup \{0\}$, где $D=\{1,\dots,n\}$ – множество АЗС. Здесь через 0 обозначена начальная и конечная точка любого маршрута – нефтебаза (депо), через $E=\{(i,j):i,j \in D\} \cup \{(0,i):i \in D\} \cup \{(j,0):j \in D\}$ – множество маршрутов.

С каждой дугой $(i,j) \in E$, где $i \neq j$, связываем время транспортировки t_{ij} , которое может включать время обслуживания у АЗС и стоимость транспортировки c_{ij}^k (стоимость пройденной дистанции между двумя точками для ТС).

Обозначим через $V=\{1,\dots,k\}$ множество транспортных средств с ограниченной вместительностью (объем ТС): $Q=(q_1,\dots,q_k)$.

Целевой функцией является суммарная стоимость перевозок, которая подлежит минимизации:

$$\min \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} x_{ij}^k c_{ij}^k.$$

Минимизация проводится при следующих ограничениях:

- 1) транспортные средства могут посетить АЗС только один раз:

$$\sum_{k \in V} \sum_{j \in N} x_{ij}^k = 1 \quad \forall i, j \in \{1,\dots,n\};$$

- 2) количество топлива, размещаемое в ТС, не может превышать вместительности данного ТС:

$$\sum_{i=1}^n \sum_{j \in N} x_{ij}^k d_i \leq q_k \quad \forall k \in V;$$

- 3) ТС после завершения объезда клиентов должно вернуться в исходное депо:

$$\sum_{i \in N} x_{i,0}^k = 1 \quad \forall k \in V;$$

4) установим взаимосвязь между временем отправления транспортного средства от АЗС i до следующей АЗС j (включая время обслуживания на i -й АЗС). Должно выполняться следующее неравенство:

$$x_{ij}^k (s_j^k - s_i^k - t_{ij}^k - s_i) \geq 0 \quad \forall i, j \in N, \forall k \in V,$$

где s_i^k – время начала обслуживания i -й АЗС ТС, s_j – время обслуживания i -й АЗС, s_j^k – время начала обслуживания j -й АЗС ТС, t_{ij} – время проезда от i -й АЗС к j -й АЗС.

Поскольку одна из важнейших задач – обеспечение непересыхаемости резервуаров АЗС, необходимо рассчитать время, к которому нефтепродукт должен быть доставлен на АЗС. Расчет плавающих временных окон происходит на основе данных о среднем расходе и остатке топлива:

$$a_i = b_i - T_{ij}; \quad b_i = t_{oi} - \theta;$$

$$t_{oi} = \frac{O_i}{r_i};$$

5) таким образом, для обеспечения непересыхаемости резервуаров необходимо соблюдение временных окон для обслуживания АЗС:

$$a_i \leq s_i^k \leq b_i \quad \forall i \in N, \forall k \in V.$$

Для решения поставленной задачи сравнили несколько алгоритмов. Первым стал генетический алгоритм из-за его случайных процессов в начале работы, так как начальные маршруты создаются случайным образом и есть возможность со старта получить хороший результат. Вторым алгоритмом стал муравьиный. Это один из эффективных полиномиальных алгоритмов, популярен в использовании для решения задачи коммивояжера. Последним стал жадный алгоритм, который будет строить оптимальную дорогу для каждой ТС только в границах текущей итерации, т. е. выбор лучшего пути происходит на текущей вершине независимо от того, какое решение предстоит на следующем шаге.

Результат работы программы представлен на рис. 3–5.

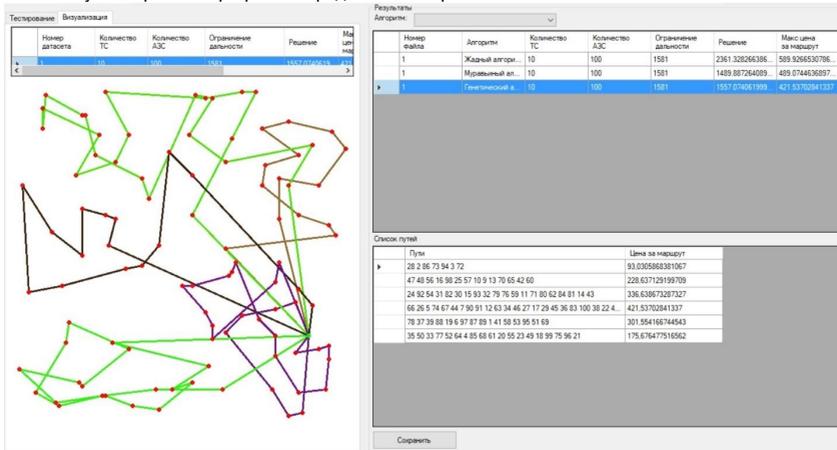


Рис. 3. Решение задачи с использованием генетического алгоритма

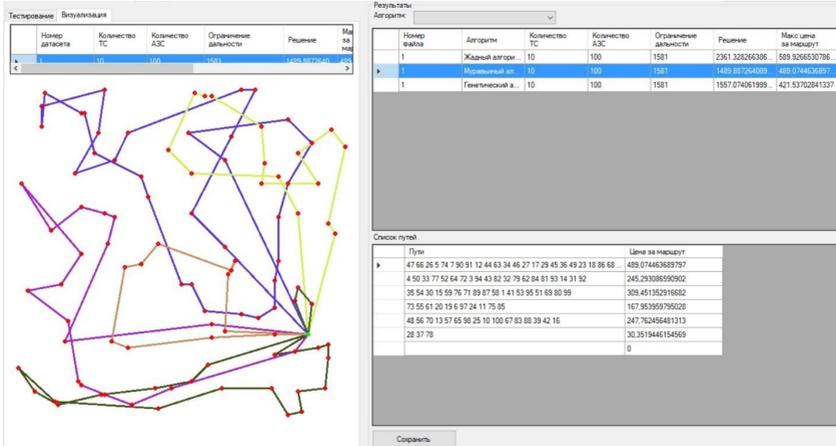


Рис. 4. Решение задачи с использованием муравьиного алгоритма

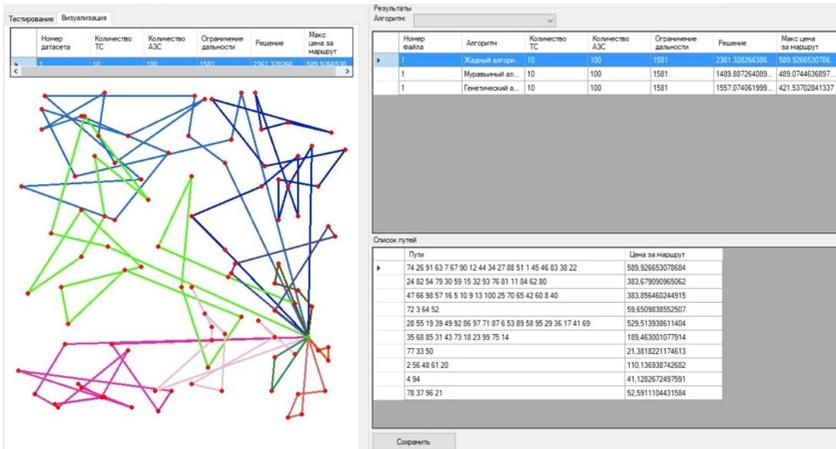


Рис. 5. Решение задачи с использованием жадного алгоритма

Разработанная математическая модель с учетом плавающих временных окон была проверена на практике с применением нескольких алгоритмов. На множественных запусках муравьиный алгоритм показал себя лучше двух других, обеспечивая меньшую стоимость перевозки. Результаты выполнения программы приближены к оптимальному решению.

Библиографический список

1. Sarikli D., Powell S. A heuristic method for the open vehicle routing problem // J. of the Operation Res. Soc. 2000. Vol. 51 (5). P. 564–573.
2. Toth P., Vigo D. The vehicle routing problem. SIAM Monographs on Discrete Mathematics and Applications. Philadelphia, 2002. 367 p.

УДК 004

Д. В. Самусенко

магистрант кафедры компьютерных технологий и программной инженерии

М. В. Фаттахова – кандидат физико-математических наук, доцент – научный руководитель

ОПТИМИЗАЦИЯ ДИСТРИБУТИВНОЙ СЕТИ ПОСТАВОК С ОДНИМ ДИСТРИБЬЮТЕРОМ И ДВУМЯ ПОСТАВЩИКАМИ

Постановка задачи

На рис. 1 представлена сеть поставок с дистрибьютером x_1^1 . Внутри каждого из узлов находятся несколько фирм, между которыми происходит конкурентная борьба за рынок. В узлах нулевого уровня – поставщики, у которых узел-дистрибьютор закупает продукцию и перепродает ее ритейлерам, расположенным на втором уровне. Последние сбывают продукцию на рынках.

Поскольку такая схема перепродаж встречается почти во всех экономических сферах, тема работы актуальна [1].

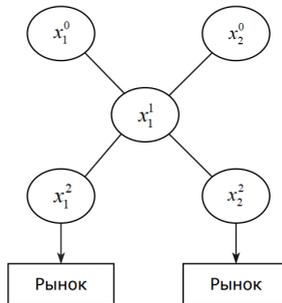


Рис. 1. Сеть поставок

Процедура принятия решения

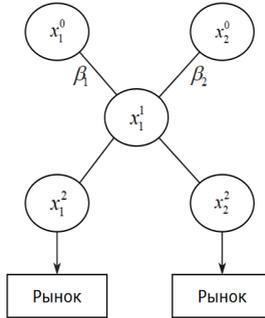
Рассматривается следующая процедура принятия решений в сети.

1. Узлы нулевого уровня определяют цену, по которым они отпускают свой товар дистрибьютеру.
2. Дистрибьютер получив цены, назначает отпускные цены для ритейлеров.
3. Ритейлеры на основе цен, полученных от дистрибьютера, и функции спроса определяют объем выпуска товара на рынок.
4. Происходит процедура распределения объемов между фирмами в каждой из вершин нижнего уровня.
5. Информация об объемах поступает на все верхлежащие уровни и внутри каждого узла идет их распределение.
6. Выполняется подсчет прибыли каждого из участников сети поставок.

Такая процедура может быть представлена многошаговой некооперативной иерархической игрой n лиц.

Теоретико-игровая модель трехуровневой игры с двумя поставщиками

Рассмотрим теоретико-игровую модель для сети поставок (рис. 2, табл. 1). Игроками являются фирмы, находящиеся в каждом из узлов. Стратегиями игроков нулевого и первого уровня выступают цены и объемы. Для второго уровня стратегии фирм – это выбор объемов производства.



$$p_{21} = 5000 - 0,25 \cdot Q_{21} \quad p_{22} = 6000 - 0,09 \cdot Q_{22}$$

Рис. 2. Сеть поставок с двумя поставщиками

Далее представлены функции прибыли для вычисления прибылей фирм нулевого, первого и второго уровней. Каждый игрок стремится максимизировать свою прибыль [2]:

$$S_{0jk} = q_{0jk} (p_{0j} - c_{0jk}). \quad (1)$$

Функции прибыли первого уровня:

$$S_{1jk} = q_{1jk} (p_{11} - c_{1jk} - \beta_1 p_{01} - \beta_2 p_{02}) \quad (2)$$

Функции прибыли второго уровня:

$$S_{2jk} = q_{2jk} (p_{2j} - p_{11} - c_{2jk}). \quad (3)$$

Таблица 1

Параметры сети поставок

Обозначение	Пояснение
c_{0jk}	Удельные затраты
β_i	Доля продукции i -го поставщика
q_{ijk}	Объем выпуска

Построение равновесия по Нэшу

Исходные данные для построения равновесия представлены в табл. 2.

Таблица 2

Исходные данные

Узел	x_1^0	x_2^0	x_1^1	x_1^2	x_2^2
Количество фирм в узле	$n_{01} = 2$	$n_{02} = 2$	$n_{11} = 2$	$n_{21} = 4$	$n_{22} = 2$

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ

Узел	x_1^0	x_2^0	x_1^1	x_1^2	x_1^2
Затраты	$c_{011} = 850$	$c_{021} = 1000$	$c_{111} = 1500$	$c_{211} = 342$	$c_{221} = 120$
	$c_{012} = 1100$	$c_{022} = 900$	$c_{112} = 1505$	$c_{212} = 340$ $c_{213} = 338$ $c_{214} = 345$	$c_{222} = 122$
Доля	$\beta_1 = 0,25$	$\beta_2 = 0,75$	-	-	-

Функции прибыли для всех фирм из узлов уровня 2 имеют вид:

$$\begin{cases} S_{211} = q_{211}(5000 - 0,25 \cdot Q_{21} - 342 - p_{11}) \\ S_{212} = q_{212}(5000 - 0,25 \cdot Q_{21} - 340 - p_{11}) \\ S_{213} = q_{213}(5000 - 0,25 \cdot Q_{21} - 338 - p_{11}) \\ S_{214} = q_{214}(5000 - 0,25 \cdot Q_{21} - 345 - p_{11}) \end{cases}, \quad (4)$$

$$\begin{cases} S_{221} = q_{221}(6000 - 0,09 \cdot Q_{22} - 120 - p_{11}) \\ S_{222} = q_{222}(6000 - 0,09 \cdot Q_{22} - 122 - p_{11}) \end{cases}. \quad (5)$$

Применяем ко всем функциям (4) и (5) необходимое условие максимума и получаем две системы уравнений [3]:

$$\begin{pmatrix} 0,5 & 0,25 & 0,25 & 0,25 \\ 0,25 & 0,5 & 0,25 & 0,25 \\ 0,25 & 0,25 & 0,5 & 0,25 \\ 0,25 & 0,25 & 0,25 & 0,5 \end{pmatrix} \begin{pmatrix} q_{211} \\ q_{212} \\ q_{213} \\ q_{214} \end{pmatrix} = \begin{pmatrix} 4658 - p_{11} \\ 4660 - p_{11} \\ 4662 - p_{11} \\ 4655 - p_{11} \end{pmatrix}, \quad (6)$$

$$\begin{pmatrix} 0,18 & 0,09 \\ 0,09 & 0,18 \end{pmatrix} \begin{pmatrix} q_{221} \\ q_{222} \end{pmatrix} = \begin{pmatrix} 5880 - p_{11} \\ 5878 - p_{11} \end{pmatrix}. \quad (7)$$

После решения систем (6) и (7) получаем выражения для q_{2ij} :

$$\begin{cases} q_{211} = 3724 - 0,8p_{11} \\ q_{212} = 3732 - 0,8p_{11} \\ q_{213} = 3740 - 0,8p_{11} \\ q_{214} = 3712 - 0,8p_{11} \end{cases}, \quad (8)$$

$$\begin{cases} q_{221} = 21785,2 - 3,7p_{11} \\ q_{222} = 21763,0 - 3,7p_{11} \end{cases}. \quad (9)$$

Из условия отсутствия дефицита и излишков получаем соотношение

$$\begin{aligned} Q_{21} &= q_{211} + q_{212} + q_{213} + q_{214} = 14908 - 3,2p_{11} \\ Q_{22} &= q_{221} + q_{222} = 43548,2 - 7,4p_{11} \\ Q_{11} &= Q_{21} + Q_{22} = 58465,2 - 10,6p_{11} \end{aligned}. \quad (10)$$

Из соотношения (10) можно выразить значение p_{11} :

$$p_{11} = 5510,9 - 0,094Q_{11}. \quad (11)$$

Фирмы в узле-дистрибьютере x_1^1 после подстановки в них функции цены (11) имеют такие функции прибыли:

$$\begin{cases} S_{111} = q_{111}(5510,9 - 0,094Q_{11} - 1500 - (\rho_{01}\beta_1 + \rho_{02}\beta_2)) \\ S_{112} = q_{112}(5510,9 - 0,094Q_{11} - 1505 - (\rho_{01}\beta_1 + \rho_{02}\beta_2)) \end{cases} \quad (12)$$

Применяем ко всем функциям (12) необходимое условие максимума и получаем систему уравнений:

$$\begin{pmatrix} 0,188 & 0,094 \\ 0,094 & 0,188 \end{pmatrix} \begin{pmatrix} q_{111} \\ q_{112} \end{pmatrix} = \begin{pmatrix} 4010,9 - (\rho_{01}\beta_1 + \rho_{02}\beta_2) \\ 4005,9 - (\rho_{01}\beta_1 + \rho_{02}\beta_2) \end{pmatrix}. \quad (13)$$

После решения системы (13) получаем выражения для q_{11j} :

$$\begin{cases} q_{111} = 14199,4 - 3,536(\rho_{01}\beta_1 + \rho_{02}\beta_2) \\ q_{112} = 14146,3 - 3,536(\rho_{01}\beta_1 + \rho_{02}\beta_2) \end{cases} \quad (14)$$

Из условия отсутствия дефицита и излишков, используя (14), получаем соотношение

$$\begin{aligned} Q_{11} &= q_{111} + q_{112} = 28345,7 - 7,07(\rho_{01}\beta_1 + \rho_{02}\beta_2) \\ Q_{11} &= Q_{01} + Q_{02} \\ Q_{01} &= q_{011} + q_{012} = \beta_1 Q_{11} = \beta_1(28345,7 - 7,07(\rho_{01}\beta_1 + \rho_{02}\beta_2)) \\ Q_{02} &= q_{021} + q_{022} = \beta_2 Q_{11} = \beta_2(28345,7 - 7,07(\rho_{01}\beta_1 + \rho_{02}\beta_2)) \end{aligned} \quad (15)$$

Из (15) выражаем значения функций ρ_{0j} через Q_{01}, Q_{02} :

$$\begin{aligned} \rho_{01} &= 16033,5 - 2,26Q_{01} - 3\rho_{02} \\ \rho_{02} &= 5344,5 - 0,25Q_{01} - 0,33\rho_{01} \end{aligned} \quad (16)$$

Функции прибыли для всех фирм из узлов нулевого уровня имеют вид:

$$\begin{cases} S_{011} = q_{011}(16033,5 - 2,26Q_{01} - 3\rho_{02} - 850) \\ S_{012} = q_{012}(16033,5 - 2,26Q_{01} - 3\rho_{02} - 1100) \end{cases}, \quad (17)$$

$$\begin{cases} S_{021} = q_{021}(5344,5 - 0,25Q_{01} - 0,33\rho_{01} - 1000) \\ S_{022} = q_{022}(5344,5 - 0,25Q_{01} - 0,33\rho_{01} - 900) \end{cases}. \quad (18)$$

Применяем ко всем функциям (17) и (18) необходимое условие максимума и получаем две системы уравнений:

$$\begin{pmatrix} 4,52 & 2,26 \\ 2,26 & 4,52 \end{pmatrix} \begin{pmatrix} q_{011} \\ q_{012} \end{pmatrix} = \begin{pmatrix} 15183,5 - 3\rho_{02} \\ 14933,5 - 3\rho_{02} \end{pmatrix}, \quad (19)$$

$$\begin{pmatrix} 0,5 & 0,25 \\ 0,25 & 0,5 \end{pmatrix} \begin{pmatrix} q_{021} \\ q_{022} \end{pmatrix} = \begin{pmatrix} 4344,5 - 0,33\rho_{01} \\ 4444,5 - 0,33\rho_{01} \end{pmatrix}. \quad (20)$$

После решения систем (19) и (20) получаем выражения для q_{0ij} :

$$\begin{cases} q_{011} = 2273,7 - 0,442p_{02} \\ q_{012} = 2163,3 - 0,442p_{02} \end{cases}, \quad (21)$$

$$\begin{cases} q_{021} = 5627,9 - 0,4376p_{01} \\ q_{022} = 6025,7 - 0,4376p_{01} \end{cases}. \quad (22)$$

Из условия отсутствия дефицита и излишков, используя (21) и (22), получаем соотношение

$$\begin{aligned} Q_{01} &= q_{011} + q_{012} = 4437 - 0,884p_{02} \\ Q_{02} &= q_{021} + q_{022} = 11653,6 - 0,8751p_{01} \end{aligned} \quad (23)$$

Приравниваем значения объемов (23) к значениям (15) соответственно:

$$\begin{cases} Q_{01} = 4437 - 0,884p_{02} = \beta_1(28345,7 - 7,07(p_{01}\beta_1 + p_{02}\beta_2)) \\ Q_{02} = 11653,6 - 0,8751p_{01} = \beta_2(28345,7 - 7,07(p_{01}\beta_1 + p_{02}\beta_2)) \end{cases}. \quad (24)$$

После решения системы получаем цены p_{01}, p_{02} :

$$\begin{cases} p_{01} = 5994,51 - p_{02} = 4027,13 \\ p_{02} = 1967,38 \end{cases}. \quad (25)$$

Используя полученные цены (25), делаем расчет всех параметров в обратном порядке и строим равновесие по Нэшу в данной игре (табл. 3)

Таблица 3

Оптимальное решение

Узел	x_1^0	x_2^0	x_1^1	x_1^2	x_1^2
Оптимальный объем выпуска	$q_{011} \approx 1404$ $q_{012} \approx 1294$	$q_{021} \approx 3848$ $q_{022} \approx 4246$	$q_{111} \approx 5422$ $q_{112} \approx 5369$	$q_{211} \approx 129$ $q_{212} \approx 137$ $q_{213} \approx 145$ $q_{214} \approx 117$	$q_{221} \approx 5143$ $q_{222} \approx 5120$
Оптимальная цена	$p_{01} \approx 4027,1$	$p_{02} \approx 1967,4$	$p_{11} \approx 4493,5$	$p_{21} \approx 4867,8$	$p_{22} \approx 5076,3$
Оптимальная прибыль участников	$S_{011} \approx 4461367$ $q_{012} \approx 3786884$	$S_{021} \approx 3722474$ $S_{022} \approx 4531854$	$S_{111} \approx 2771859$ $S_{112} \approx 2717900$	$S_{211} \approx 4173$ $S_{212} \approx 4706$ $S_{213} \approx 5271$ $S_{214} \approx 3434$	$S_{221} \approx 2380152$ $S_{222} \approx 2359626$

Библиографический список

1. Corbett C., Karmarkar U. S. Competition and structure in serial supply chains with deterministic demand // Management science, 2001. № 47. P. 966—978.
2. Петросян Л. А., Зенкевич Н. А., Шевкопляс Е. В. Теория игр. Изд. 2-е. СПб.: БХВ-Петербург, 2014. 432 с.
3. Лоягина Ю., Никольченко Н., Зенкевич Н. Конкурентное и кооперативное поведение в распределительных сетях // Вклад в теорию игр и менеджмент. 2018. Т. 11. С. 73–102.

УДК 004

А. Н. Скряпник

студент кафедры компьютерных технологий и программной инженерии

М. В. Фаттахова – кандидат физико-математических наук, доцент – научный руководитель

РЕШЕНИЕ ЗАДАЧИ CVRP С ДВУМЯ ДЕПО И N КЛИЕНТАМИ

За последние полвека множество компаний все чаще сталкиваются с логистикой и ее задачами – методами управления цепями поставок, оптимизации поставок и др. Однако центральной задачей прикладной математической логистики стала маршрутизация транспорта с ее обширными возможностями и разными ограничениями.

Задачи маршрутизации транспорта [1] (Vehicle Routing Problems, VRP (рис. 1)) – задачи комбинаторной оптимизации, в которых есть множество транспортных средств, расположенных в депо (одном или нескольких), имеется множество клиентов с различными спросами и требованиями (например, спрос на товар или временные окна для доставки), необходимо построить набор маршрутов для каждого транспортного средства с учетом посещения всех клиентов и возвращения в депо. Это задача целочисленного программирования, которая к тому же относится к классу NP-трудных задач, что означает, что построение оптимальных решений возможно только путем полного перебора всех вариантов. Вычислительная сложность задачи зависит от входных данных экспоненциально. Поэтому на практике часто строят и используют приближенные решения с заданными ограничениями на время поиска и точностью. Обычно это достигается разными эвристическими и метаэвристическими методами.

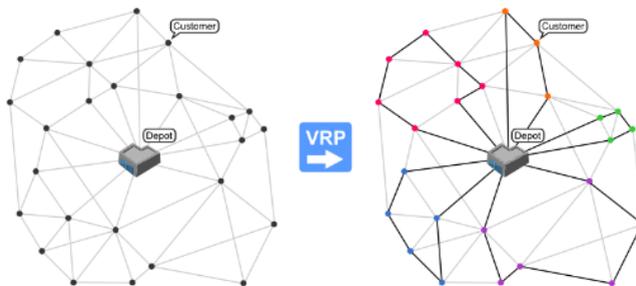


Рис. 1. Графическая интерпретация задачи VRP

Задачи VRP появились не сразу, а благодаря совмещению двух уже хорошо изученных задач:

- задачи коммивояжера;
- задачи об упаковке рюкзака.

Задача коммивояжера (Traveling Salesman Problem, TSP) имеет отличия от VRP в том, что грузоподъемность каждого транспортного средства изначально считается достаточной. VRP же сводится к множественной задаче коммивояжера.

Задача об упаковке рюкзака по сути схожа с задачей VRP при условии, что все расстояния незначительны.

Для решения поставленной задачи нужно понимать, из каких классов VRP она состоит. По сути, это объединение трех классов задач VRP в один:

- MDVRP – Multiple Depot VRP (задача VRP с несколькими депо, в рассматриваемой задаче депо два) [2];

- CVRP – Capacitated VRP (задача VRP с учетом ограничения на вместительность машин) [3];

• SVRP – Stochastic VRP (задача VRP со случайными параметрами, в рассматриваемой задаче – спрос клиентов на товары, производимые в депо) [4].

Рассмотрим следующую задачу. Имеется некоторое количество транспортных средств с заданной вместительностью и множество депо. Транспортные средства (ТС) изначально расположены в одном депо. Там же находится множество продуктов, спрос на которые у каждого клиента свой. Задача состоит в том, чтобы определить маршрут транспортных средств таким образом, чтобы минимизировать суммарную стоимость перевозок. При этом необходимо удовлетворить спрос клиентов по каждому из продуктов. После выезда любое ТС может прибыть в ближайшее от последнего клиента депо, возвращаться в исходное необязательно. Графическая интерпретация задачи приведена на рис. 2.

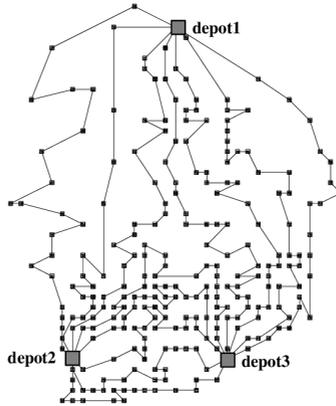


Рис. 2. Графическая интерпретация задачи SCMDVRP

Для решения задачи была разработана математическая модель.

Рассмотрим взвешенный граф $G=(Z,A)$. Здесь через A обозначено множество дуг, соединяющих вершины графа, а через $Z=(E,D)$ – множество всех вершин этого графа, а именно:

- $E=\{e_1,\dots,e_{N_{cust}}\}$ – множество клиентов, где N_{cust} – количество клиентов;
- $D=\{d_1,\dots,d_{N_{depo}}\}$ – множество депо, где N_{depo} – количество депо.

Множество всех транспортных средств обозначим через $V=\{1,\dots,K\}$. Для каждого задана грузоподъемность $Q_v=\{Q_1,\dots,Q_K\}$, а сами машины размещены в депо d_1 , там же находится множество товаров, наименования которых для удобства пронумеруем: $L=\{1,\dots,l\}$, где l – количество товаров.

Введем переменную X_{ij}^v : если в маршрутном плане клиент j посещается сразу за клиентом i машиной v , переменная будет равна 1, иначе – 0. Тогда под маршрутом транспортного средства $v \in V$ будем понимать такую последовательность дуг $(i_1, j_1), (i_2, j_2), \dots, (i_z, j_z)$, для каждой из которых выполняется условие $X_{i_t j_t}^v = 1, \forall (i_t, j_t), t=1, \dots, z$.

Целевой функцией является суммарная стоимость перевозок, которая подлежит минимизации:

$$\min(F) = \min \left(\sum_{v=1}^K \sum_{i \in V} \sum_{j \in V} c_{ij} X_{ij}^v \right).$$

При этом необходимо учесть следующие ограничения.

1. Сумма спроса клиентов не может превышать производственные возможности депо, что важно для обеспечения возможности покрыть спрос клиентов полностью. Это условие задается следующим неравенством:

$$\sum_{v=1}^K \sum_{i \in V} y_{il} Y_i^v - \sum_{d \in D} R_d \leq 0,$$

где y_{il} – спрос клиента i на продукт l , Y_i^v – логическая переменная, которая будет равна 1, если для ТС v есть плановое перемещение к вершине i , иначе – 0, R_d – вместимость депо.

2. Объем доставляемых товаров не может превышать грузоподъемность ТС $v \in V$. Данное ограничение является классическим ограничением на грузоподъемность:

$$\sum_{i \in V} y_{il} Y_i^v \leq Q_v,$$

где y_{il}^v – спрос клиента i на продукт l , удовлетворенный ТС v , Y_i^v – логическая переменная, которая будет равна 1, если для ТС v есть плановое перемещение к вершине i , иначе – 0, Q_v – грузоподъемность ТС v .

3. Спрос каждого клиента должен быть удовлетворен полностью:

$$\sum_{v=1}^K y_{il}^v Y_i^v = y_{il},$$

где y_{il}^v – спрос клиента i на продукт l , удовлетворенный ТС v , Y_i^v – логическая переменная, которая будет равна 1, если для ТС v есть плановое перемещение к вершине i , иначе – 0, y_{il} – спрос клиента i на продукт l .

4. Каждое использованное ТС $v \in V$ должно быть возвращено в депо, необязательно в исходное:

$$\sum_{d \in D} X_d^v \leq 1,$$

где X_d^v – логическая переменная, которая принимает значение 1, если ТС v назначено возвращение к депо d , иначе – 0.

5. Удовлетворенность клиента $i \in E$, ТС $v \in V$ не должна превышать спрос на продукт $l \in L$, данного клиента:

$$y_{il}^v \leq y_{il} \sum_{j \in V} X_{ij}^v,$$

где y_{il}^v – спрос клиента i на продукт l , удовлетворенный ТС v , y_{il} – спрос клиента i на продукт l .

6. Следующее ограничение гарантирует непрерывность маршрутов движения ТС:

$$\sum_{i \in V} X_{ij}^v = \sum_{i \in V} X_{ji}^v, \forall j \in Z;$$

7. Каждый клиент должен быть посещен не менее одного раза. Данное условие связано с необходимостью полного удовлетворения спроса клиентов:

$$\sum_{i \in V} X_{ij}^v \geq 1, \forall j \in Z.$$

Для проверки работоспособности модели была разработана программа с использованием генетического алгоритма со стандартными параметрами самого алгоритма. Результаты ее работы представлены на рис. 3-5.

Депо: 1
Машин: 5
Клиентов: 25
Товар: 1

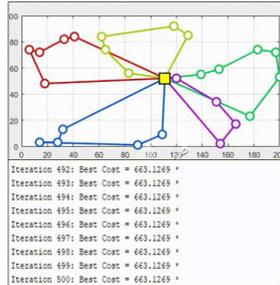


Рис. 3. Решение задачи с одним депо, двадцатью пятью клиентами, пятью транспортными средствами и спросом на один товар

Депо: 2
Машин: 3
Клиентов: 8
Товар: 1

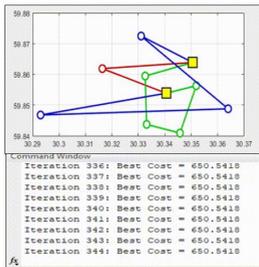


Рис. 4. Решение задачи с двумя депо, восемью клиентами, тремя транспортными средствами и спросом на один товар

Депо: 2
Машин: 3
Клиентов: 8
Товар: 3

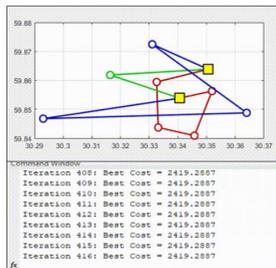


Рис. 5. Решение задачи с двумя депо, восемью клиентами, тремя транспортными средствами и спросом на три товара

Полученные решения не являются глобально-оптимальными. Все они приближены к оптимуму. Объединение трех классов задач VRP в один возможно без применения многоэтапных решений. Разработанная математическая модель была проверена на практике с применением генетического алгоритма. Результаты выполнения программы являются приближенными к оптимальному решению.

Библиографический список

1. Sarikli D., Powell S. A heuristic method for the open vehicle routing problem // J. of the Operation Res. Soc. 2000. Vol. 51 (5). P. 564–573.
2. Crevier B., Cordeau J. F., Laporte G. The multi-depot vehicle routing problem with inter-depot routes // Europ. J. of Operational Res. 2007. Vol. 176 (2). P. 756–773.
3. Toth P., Vigo D. The vehicle routing problem. SIAM Monographs on Discrete Mathematics and Applications. Philadelphia, 2002. 367 p.
4. Gendreau M., Laporte G., Seguin R. An exact algorithm for the vehicle routing problem with stochastic demands and customers // Transportation Science. 1995. Vol. 29. P. 143–155.

УДК 004.942

А. С. Ткаченко

студент кафедры компьютерных технологий и программной инженерии

М. Д. Поляк – старший преподаватель – научный руководитель

ИССЛЕДОВАНИЕ МОДЕЛИ «ХИЩНИК – ЖЕРТВА С ПИТАНИЕМ И УПРАВЛЕНИЕМ БЕЗ РАСШИРЕНИЯ ФАЗОВОГО ПРОСТРАНСТВА»

Модель Лотки – Вольтерра – это модель типа «хищник – жертва», описывающая изменение численности двух популяций. Названа в честь авторов, которые предложили модельные уравнения независимо друг от друга.

Рассматривается закрытый ареал, в котором обитают два вида – жертвы и хищники. Предполагается, что животные не иммигрируют и не эмигрируют и что еды для жертв имеется с избытком. Тогда уравнение изменения количества жертв (без учета хищников) принимает вид:

$$\frac{dx}{dt} = \alpha x, \tag{1}$$

где α – коэффициент рождаемости жертв, x – величина популяции жертв, $\frac{dx}{dt}$ – скорость прироста популяции жертв.

Пока хищники не охотятся, они вымирают, следовательно, уравнение для численности хищников (без учета численности жертв) принимает вид:

$$\frac{dy}{dt} = -\gamma y, \tag{2}$$

где γ – коэффициент убыли хищников, y – величина популяции хищников, $\frac{dy}{dt}$ – скорость прироста популяции хищников.

При встречах хищников и жертв (частота которых прямо пропорциональна величине xy) происходит убийство жертв с коэффициентом β , сытые хищники способны к воспроизводству с коэффициентом δ . С учетом этого система уравнений модели такова:

$$\begin{cases} \frac{dx}{dt} = \alpha x + \beta xy \\ \frac{dy}{dt} = -\gamma y + \delta xy \end{cases} \tag{3}$$

Система (3) описывает взаимодействие и изменение численности двух биологических видов. Предположим, что мы можем прогнозировать и управлять изменением численности хищников и жертв и добиваться заданного количества жертв.

Для этого будем использовать метод аналитического конструирования агрегированных регуляторов (АКАР) [1]. Рассмотрим постановку задачи АКАР – управления для модели «Хищник – жертва с питанием и управлением без расширения фазового пространства». Общий вид системы:

$$\begin{aligned} \frac{dx}{dt} &= \alpha(t) - \beta xy, \\ \frac{dy}{dt} &= -\gamma y + \delta xy, \\ u(t) &= \alpha(t). \end{aligned} \tag{4}$$

Для объекта (4) построить управление $U(t)$, выводящее изображающую точку (ИТ) $(x_1(t), x_2(t), \alpha(t))$ системы на целевое многообразие вида:

$$\psi(t) = x(t) - x_1^* \xrightarrow{t \rightarrow \infty} 0, \quad (5)$$

где x_1^* – заданная постоянная целевая (для $x_1(t)$) величина и доставляющая глобальный минимум функции качества управления:

$$\Phi = \int_0^{\infty} ((\dot{\psi}(t))^2 + \omega^2 \psi^2) dt \longrightarrow \min, \quad (6)$$

где постоянная ω интерпретируется как величина, пропорционально влияющая на длительность переходного процесса.

Рассмотрим алгоритм непрерывного управления без расширения фазового пространства [2].

Пусть

$$\dot{x}(t) = u(t)x - \beta xy, \psi(t) = x(t) - x_1^*.$$

Определение структуры АКАР-управления [3]:

$$T_1 \dot{\psi} + \psi(t) = 0 \Rightarrow T_1 \frac{dx}{dt} + \psi(t) = 0 \Rightarrow. \quad (7)$$

Используем $\dot{f}_1 = u(t)x - \beta xy$, подставив в уравнение (7), получим уравнение, определяющее структуру АКАР-управления:

$$\begin{aligned} T_1 \dot{f}_1 + \psi(t) &= u(t)T_1 x - T_1 \beta xy + \psi(t) = 0 \Rightarrow \\ u(t) &= (T_1 \beta xy - \psi(t))(T_1 x)^{-1} = \beta y - \psi(t)(T_1 x)^{-1}. \end{aligned} \quad (8)$$

Проверим алгоритм, для этого подставим управление в исходную систему:

$$\begin{aligned} \frac{dx}{dy} &= u(t)x - \beta xy = \beta xy - \psi(t)(T_1 x)^{-1}, \\ \frac{dy}{dt} &= -\gamma y + \delta xy \end{aligned} \quad (9)$$

Из первого уравнения в системе (9):

$$\dot{x}(t) = -\psi(t)T^{-1}, \quad (10)$$

т. е. производная $\dot{x}(t)$ стремится к нулю, значит, $x(t) \rightarrow const$ (а именно, из уравнения Эйлера – Лагранжа $x \rightarrow x_1^*$).

Рассмотрим второе уравнение в системе (9):

$$\dot{y}(t) = -\gamma y + \delta xy. \quad (11)$$

Из выражения (5) известно, что $x \xrightarrow{t \rightarrow \infty} x_1^*$, тогда уравнение (11) примет вид:

$$\dot{y} = -\lambda y + \delta x_1^* y = y(-\lambda + \delta x_1^*) \quad (12)$$

Для того чтобы рассмотреть поведение системы, решим уравнение (12) относительно численности хищников (переменной y):

$$\frac{dy}{dt} = -\gamma y + \delta x_1^* y = (-\gamma + \delta x_1^*) y$$

$$\int_0^{\infty} \frac{dy}{y} = \int_0^{\infty} (-\gamma + \delta x_1^*) dt$$

$$\ln|Cy| = (\delta x_1^* - \gamma)t + C, t \rightarrow \infty$$

Рассмотрим три случая:

$$\begin{cases} 0 < Cy < 1 \\ Cy > 1 \\ Cy = 1 \end{cases}$$

При $0 < Cy < 1$:

$$(\delta x_1^* - \gamma)t + C < 0$$

$$(\delta x_1^* - \gamma)t < -C$$

$$\delta x_1^* - \gamma < \frac{C}{t}, t \rightarrow \infty \Rightarrow C \rightarrow 0, \text{ тогда} \quad (13)$$

$$\delta x_1^* - \gamma < 0$$

$$x_1^* < \frac{\gamma}{\delta}$$

Из итогового неравенства (13) следует, что если $x_1^* < \frac{\gamma}{\delta}$, то $y \rightarrow 0$, численность хищников будет постоянно убывать, пока они не вымрут.

При $Cy > 1$

$$(\delta x_1^* - \gamma)t + C > 0$$

$$(\delta x_1^* - \gamma)t > -C$$

$$(\delta x_1^* - \gamma) > -\frac{C}{t}, t \rightarrow \infty \Rightarrow C \rightarrow 0. \quad (14)$$

$$(\delta x_1^* - \gamma) > 0$$

$$x_1^* > \frac{\gamma}{\delta}$$

Из неравенства (14) следует, что если $x_1^* > \frac{\gamma}{\delta}$, то $y \rightarrow \infty$, а значит, численность хищников будет экспоненциально возрастать.

При $Cy = 1$

$$(\delta x_1^* - \gamma)t + C = 1$$

$$\delta x_1^* - \gamma = \frac{1-C}{t}, t \rightarrow \infty \Rightarrow C \rightarrow 1. \quad (15)$$

$$x_1^* = \frac{\gamma}{\delta}$$

Из равенства (15) следует, что если $x_1^* = \frac{\gamma}{\delta}$, то $y \rightarrow const$, а значит, численность хищников будет стремиться к постоянному значению, причем устойчиво.

Система уравнений для непосредственного мультипликативного управления

$$\begin{aligned} \frac{dx}{dt} &= u(t)x - \beta xy, \\ \frac{dy}{dt} &= -\gamma y + \delta xy, \\ u(t) &= \beta y - \psi(t)(T_1 x)^{-1}, \psi(t) = x(t) - x_1^*, \\ -\gamma + \delta x_1^* &= 0. \end{aligned} \tag{16}$$

Рассмотрим на примере случай, когда $x_1^* > \frac{\gamma}{\delta}$, из этого следует, что $y \rightarrow \infty$, а значит, численность хищников будет экспоненциально расти.

Из рис. 1 видно, что цель $x_1^* = 10$ достигнута, однако количество хищников экспоненциально возрастает, что не соответствует физическому смыслу модели и нарушает устойчивость системы. Отсюда следует, что набор коэффициентов при $x_1^* > \frac{\gamma}{\delta}$ нарушает устойчивость.

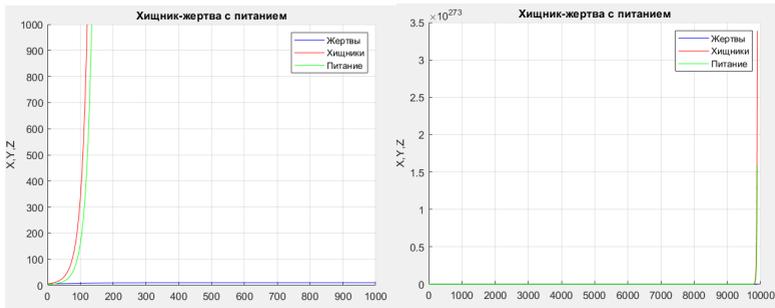


Рис. 1. График изменения численности жертв, хищников и питания с коэффициентами и начальными значениями $T_1 = 0.92, \beta = 0.50, \gamma = 1.40, \delta = 0.80, x(0) = 5.00, y(0) = 6.00, x_1^* = 10.00$

Рассмотрим на примере случай, когда $x_1^* < \frac{\gamma}{\delta}$, из этого следует, что $y \rightarrow 0$, а значит, численность хищников будет постоянно убывать.

Из рис. 2 видно, что цель $x_1^* = 10$ достигнута, однако количество хищников стремится к нулю и с течением времени они вымирают, это не соответствует физическому смыслу модели и нарушает устойчивость системы. Отсюда следует, что набор коэффициентов при $x_1^* < \frac{\gamma}{\delta}$ нарушает устойчивость.

Рассмотрим на примере случай, когда $x_1^* = \frac{\gamma}{\delta}$. Из рис. 3 видно, что цель $x_1^* = 10$ достигнута, при этом хищники не вымирают, их численность стремится к постоянной величине и со временем стано-

вится $y = 3$. Отсюда можно сделать вывод, что система устойчива при $x_1^* = \frac{\gamma}{\delta}$ и коэффициентах $T_1 = 0.69, \beta = 1.50, \gamma = 1.50, \delta = 1.00, x(0) = 5.00, y(0) = 6.00, x_1^* = 10.00$.

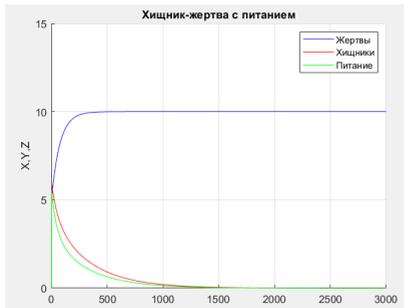


Рис. 2. График изменения численности жертв, хищников и питания с коэффициентами и начальными значениями

$$T_1 = 0.76, \beta = 0.70, \gamma = 1.30, \delta = 0.10, \\ x(0) = 5.00, y(0) = 6.00, x_1^* = 10.00$$

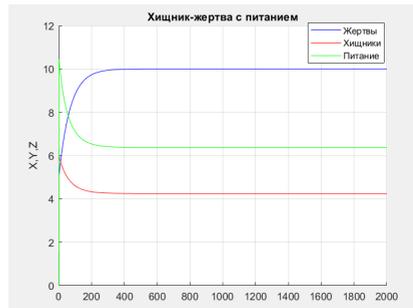


Рис. 3. График изменения численности жертв, хищников и питания с коэффициентами и начальными значениями

$$T_1 = 0.69, \beta = 1.50, \gamma = 1.50, \delta = 1.00, \\ x(0) = 5.00, y(0) = 6.00, x_1^* = 10.00$$

Была рассмотрена модификация модели Лотки – Вольтерра «Хищник – жертва с питанием и управлением без расширения фазового пространства». Эксперимент выявил, какие из коэффициентов модели влияют на изменение численности хищников и условия, при которых рассматриваемая система будет устойчива.

В дальнейшем планируется исследование влияния других коэффициентов системы на численность хищников и изучение иных модификаций модели Лотки – Вольтерра.

Библиографический список

1. Колесников А. А. Синергетика и проблемы теории управления: сб. науч. тр. / под ред. А. А. Колесникова. М.: ФИЗМАТЛИТ, 2004.
2. Kolesnikova S. I. Synthesis of the Control System for a Second Order Non-Linear Object with an Incomplete Description // Automation and Remote Control. 2018. Vol. 79, № 9. P. 1556–1566.
3. Колесникова С. И. Синтез системы управления нелинейным объектом второго порядка с неполным описанием // Автоматика и телемеханика. 2018. № 9. С. 18–30.

УДК 004.932.2

Ю. А. Хмельва

магистрант кафедры компьютерных технологий и программной инженерии

А. А. Попов – кандидат технических наук – научный руководитель

ПОДХОДЫ К РАЗРАБОТКЕ СИСТЕМЫ ВЫДЕЛЕНИЯ, КЛАССИФИКАЦИИ И РАСПОЗНАВАНИЯ ОБЪЕКТОВ ВИДЕОРЯДА

Постановка задачи

Для своевременного устранения нарушений железнодорожного подвижного состава в местах обслуживания необходима система предварительного контроля состояния вагонов поезда. Для его организации предлагается использовать камеры, установленные на ригеле жестких поперечин (рис. 1) [1].

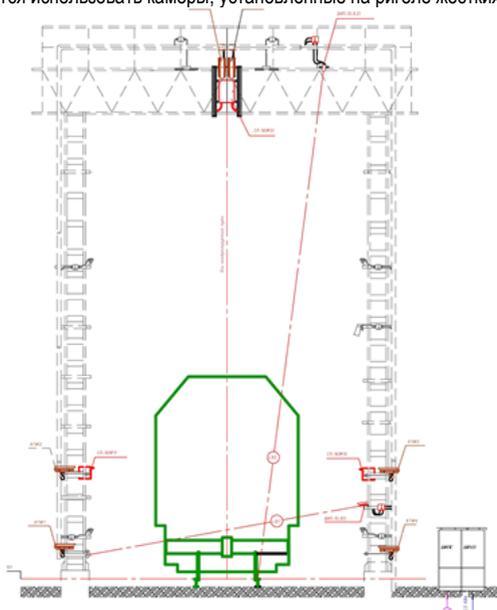


Рис. 1. Схема размещения камер

На рис. 1 красным цветом обозначены пять камер (по две на вертикальном ригеле для обзора боковых поверхностей состава и одна сверху на горизонтальном). На данный момент с помощью камер оператор тщательно проверяет состав и своевременно сообщает на станцию о выявленных нарушениях. Помимо осмотра вагонов, оператор также фиксирует номер поезда в общей системе контроля движения железнодорожных составов [2].

Автоматизация функций оператора возможна введением системы выделения, классификации и распознавания деталей поверхностей подвижного состава. На нее предлагается возложить следующие функции:

- 1) выделение на поверхности проезжающего состава заранее заданных контуров;
- 2) определение, к какому классу относятся вагоны;
- 3) распознавание состояния выделенной детали вагона;
- 4) сохранение фото детали для контроля оператором.

Для захвата номера состава нужна специальная камера, поскольку поезд движется с большой скоростью, на изображениях появятся эффекты «наклона» и «смазывания». На рис. 2 и 3 наиболее отчетливо виден эффект «наклона» изображения.

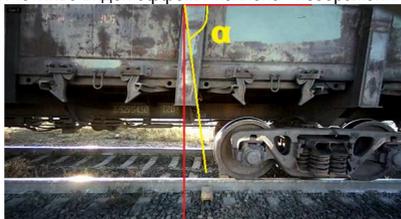


Рис. 2. Снимок вагона сбоку

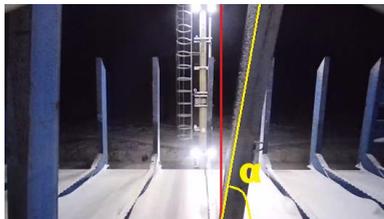


Рис. 3. Снимок вагона сбоку

Для съемки будет использоваться камера Basler ac800-200gm, которая уберет данные эффекты благодаря большой скорости затвора (короткая выдержка), частоте 240 кадров в секунду и глобальному затвору (Global shutter).

Скорость затвора

Использование меньшей скорости затвора (длинной выдержки) приводит к появлению эффекта смазывания. Если снимать с низкой скоростью затвора, лучше установить камеру на штатив и использовать стабилизацию изображения для снижения любых нежелательных перемещений камеры. В некоторых случаях смазывание движения желательны и позволяет создавать художественные фотографии.

Использование большой скорости затвора (например, выдержки 1/250 с и меньше) подходит для съемки быстро движущихся объектов с минимальным смазыванием движения или вовсе без него. Это позволяет создать неподвижное изображение, которое кажется замершим во времени без какого-либо смазывания, ассоциирующегося с быстрым движением объектов.

Единственный минус использования большой скорости затвора состоит в том, что должно быть достаточно яркое освещение. Так как затвор открывается на короткое время, это ограничивает поступающее внутрь камеры количество света. При медленном затворе такое освещение не требуется, так как затвор камеры открыт в течение более длительного времени, что позволяет большему количеству света попасть внутрь камеры.

Большинство камер имеет меньшую скорость затвора, что для решения задачи совершенно не подходит, так как поезд – быстро движущийся объект и нужно использовать специальные камеры с короткой выдержкой. На рис. 4 и 5 показаны различия большой и маленькой скорости затвора.



Рис. 4. Фотография, полученная при использовании большой скорости затвора



Рис. 5. Фотография, полученная при использовании низкой скорости затвора

Global shutter vs Rolling shutter

Режим скользящего затвора, или Rolling Shutter, по сути, означает, что смежные ряды массива экспонируются в несколько разное время, когда считываемые «волны» проходят через каждую половину датчика. То есть каждая строка будет начинать и заканчивать свою экспозицию, слегка смещенную по времени от соседней строки. С точки зрения показаний датчик разделен пополам по горизонтали, и каждый столбец считывается параллельно, строка за строкой. При съемке серии изображений с помощью Rolling Shutter можно работать в режиме непрерывного «перекрытия», в результате чего после считывания каждой строки он сразу входит в следующую экспозицию [3].

Потенциальный недостаток режима скользящего затвора состоит в искажении и артефактах на изображении. Артефакты появляются в сценах с быстрыми объектами или при быстром панорамировании. Очень заметен эффект при съемке транспортных средств типа автобусов или поездов, когда они проезжают мимо камеры наблюдения. В целом чем быстрее движение в кадре, тем более заметен эффект.

В режиме глобального затвора (Global Shutter) все пиксели массива экспонируются одновременно, что позволяет захватывать «стоп-кадр» быстро движущихся или быстро меняющихся событий. Глобальный затвор может быть сконфигурирован для работы в режиме непрерывного «перекрытия», при котором экспозиция может продолжаться, пока предыдущая считывается из узлов считывания каждого пикселя. В этом режиме датчик имеет полный рабочий цикл 100 %, что снова приводит к оптимальному временному разрешению и эффективности сбора фотонов.

На рис. 6 можно увидеть наглядно различия между глобальным и скользящим затвором.



Рис. 6. Изображение автомобиля с Global Shutter и Rolling shutter

Обработка изображения

Более ранние рамки обнаружения рассматривали разные части изображения несколько раз в разных масштабах и применяли технику классификации изображений для обнаружения объектов. Этот подход медленный и неэффективный, поэтому будет применен метод YOLO, основанный на совершенно другом подходе.

YOLO – это передовая сеть для распознавания объектов (object detection), разработанная Джоозефом Редмоном (Joseph Redmon). Главное, что отличает ее от других популярных архитектур, скорость [1]. Модели семейства YOLO действительно быстрые, намного быстрее R-CNN и др. Это значит, что можно распознавать объекты в реальном времени, так как метод просматривает все изображение только один раз, проходит через сеть один раз и обнаруживает объекты. Отсюда и название. Это очень быстро [4, 5].

Все изображение разбивается с помощью сетки на ячейки размером $S \times S$. После этого каждая ячейка отвечает за предсказание нескольких вещей (рис. 7, 8). Во-первых, за предсказание нескольких содержащих рамок и показателя уверенности (confidence) для каждой из них, другими словами, это вероятность того, что данная рамка содержит объект. Если в какой-то ячейке сетки объектов нет, то очень важно, чтобы confidence для этой ячейки был очень малым. Когда визуализируются все предсказания, получается карта всех объектов и набор содержащих рамок, ранжированных по их confidence.

Во-вторых, каждая ячейка отвечает за предсказание вероятностей классов. Это не значит, что какая-то ячейка содержит какой-то объект, это всего лишь вероятность. Таким образом, если ячейка сети предсказывает автомобиль, это не значит, что он там есть, но это значит, что если там есть какой-то объект, то это автомобиль.

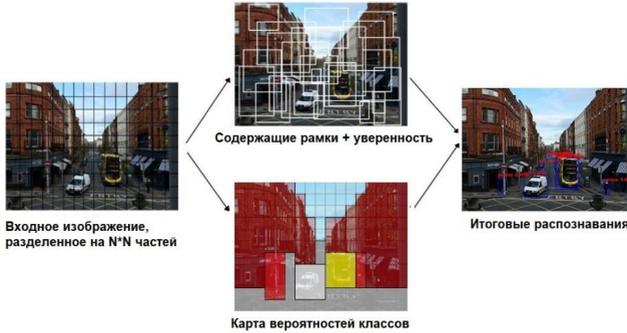


Рис. 7. Алгоритм предсказания YOLO

Для предсказания содержащих рамок используются якорные рамки (anchor boxes) (рис. 8). Их основная идея заключается в предопределении двух разных рамок, называемых якорными рамками или формой якорных рамок. Это позволяет ассоциировать два предсказания с этими якорными рамками. В общем, можно использовать и большее количество якорных рамок (пять или даже больше). Якоря были рассчитаны на Data set COCO с помощью k-means кластеризации.

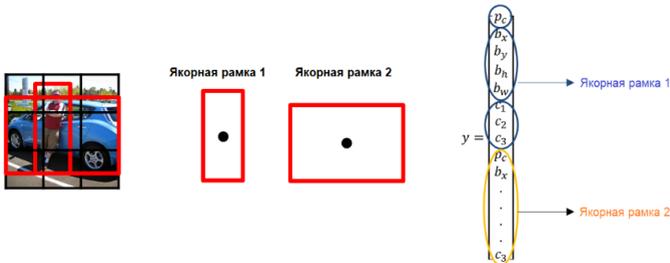


Рис. 8. Алгоритм предсказания содержащих рамок в YOLO

Благодаря алгоритму YOLO можно будет вычислить номер вагона. И для точного и корректного распознавания цифр в номере также будет использоваться фильтр Sanny edge detector, также известный как детектор границ Канни, чей алгоритм состоит в следующем:

- 1) сглаживание. Размытие изображения для удаления шума;
- 2) поиск градиентов. Границы отмечаются там, где градиент изображения приобретает максимальное значение.
- 3) подавление не-максимумов. Только локальные максимумы отмечаются как границы;
- 4) двойная пороговая фильтрация. Потенциальные границы определяются порогом;
- 5) трассировка области неоднозначности. Итоговые границы определяются путем подавления всех краев, не связанных с определенными (сильными) границами.

Перед применением детектора изображение преобразуется в оттенки серого, чтобы уменьшить вычислительные затраты. Этот этап характерен для многих методов обработки изображений. На рис. 9 представлен пример использования фильтра границ Канни на изображении.



Рис. 9. Использование фильтра Канны на изображении

Библиографический список

1. You Only Look Once: Unified, Real-Time Object Detection / J. Redmon, S. Divvala, A. Girshick, A. Farhadi. URL: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf (дата обращения: 22.04.2021).
2. Калинина М. О., Николаев П. Л. Детектирование книг на книжных полках при помощи глубоких нейронных сетей. URL: <http://oaji.net/articles/2021/2401-1609959855.pdf> (дата обращения: 22.04.2021).
3. Кулешов А., Сушкова Л., Шевченко Н. Методы и алгоритмы обработки изображений. М.: LAP Lambert Acad. Publ., 2012. 104 с.
4. Хуанг Т. С. Быстрые алгоритмы в цифровой обработке изображений. М: Книга по требованию, 2012. 222 с.
5. Зотов С. С., Яковлев А. А., Колчинцев Д. А. Обнаружение объектов в реальном времени с помощью алгоритмов распознавания YOLO. URL: <https://elibrary.ru/item.asp?id=35538888> (дата обращения: 22.04.2021).

УДК 004.942

А. А. Щеголева

студент кафедры компьютерных технологий и программной инженерии

М. Д. Поляк – старший преподаватель – научный руководитель

**АПРОБАЦИЯ МОДЕЛИ «ХИЩНИК – ЖЕРТВА С ПИТАНИЕМ»
НА ДАННЫХ ГИДРОБИОЛОГИЧЕСКОГО МОНИТОРИНГА БАЛТИЙСКОГО МОРЯ**

Введение

Рассматривается система «хищник – жертва с питанием», созданная на основе классической модели «хищник – жертва» с использованием управления по методу АКАР синергетической теории управления. Перспективная область применения данной системы – прогнозирование цветения сине-зеленых водорослей в акватории Балтийского моря. Для апробации использовались открытые данные гидробиологических показателей Невской губы и Финского залива (побережье Швеции).

Система «хищник – жертва с питанием»

Модель Лотки – Вольтерра [1, 2] описывает межвидовые отношения типа «хищник – жертва». Она представлена системой двух дифференциальных уравнений, где x_1 и x_2 – популяция жертв и хищников соответственно, $\alpha_1, \alpha_2, \beta_1, \beta_2$ – коэффициенты, влияющие на изменение популяций:

$$\begin{cases} \frac{dx_1}{dt} = f_1 = \alpha_1 x_1 - \beta_1 x_1 x_2 \\ \frac{dx_2}{dt} = f_2 = -\alpha_2 x_2 + \beta_2 x_1 x_2 \end{cases} \quad (1)$$

Система (1) решается методом Эйлера, поскольку не может быть решена аналитически в силу нелинейности [3].

Для того чтобы систему (1) можно было применять для моделирования «цветения», было спроектировано управление по методу АКАР синергетической теории управления [4–7]. Поскольку «цветение» характеризуется значительным увеличением популяции цианобактерий (сине-зеленых водорослей), то целевым значением x^* , относительно которого строится управление, является некоторое значение популяции жертв, многократно большее, чем начальное значение популяции, т. е. $x^* = k \cdot x_1(0)$.

$$\begin{cases} \frac{dx_1}{dt} = f_1 = \alpha_1(t)x_1 - \beta_1 x_1 x_2 \\ \frac{dx_2}{dt} = f_2 = -\alpha_2 x_2 + \beta_2 x_1 x_2 \\ \frac{d\alpha_1}{dt} = U(t) \\ U(t) = -\frac{\psi^{(1)}}{T_1} + \frac{d\varphi}{dt} \\ \psi^{(1)}(t) = \alpha_1(t) - \varphi(x_1(t), x_2(t)) \\ \varphi(x_1, x_2) = \beta_1 x_2 - \frac{1}{T_2} + \frac{x_1^*}{T_2 x_1} \\ \frac{d\varphi(x_1, x_2)}{dt} = -\frac{x_1^*}{T_2 x_1^2} f_1 + \beta_1 f_2 \end{cases} \quad (2)$$

Апробация на данных по Финскому заливу

В статье [8] описан алгоритм подбора коэффициентов $\alpha_1, \alpha_2, \beta_1, \beta_2, T_1, T_2$ для системы (2) по данным гидробиологического мониторинга. Будет апробироваться этот алгоритм на датасетах. В этой же работе описаны ограничения, накладываемые на решение системы (2), и критерии, по которым определяются устойчивость полученной модели и достижимость заданного целевого значения x^* .

Данные для первого датасета, на котором апробировался метод подбора коэффициентов, были взяты из ежегодника состояния вод Невской губы и восточной части Финского залива по гидробиологическим показателям в 2012 г. [9]. Поскольку данный датасет имеет существенный недостаток в виде малого количества измерений, было решено сгруппировать несколько станций по территориальному признаку. У каждой станции свои начальные данные, для каждой получается набор коэффициентов. Полученные коэффициенты усредняются для дальнейшего моделирования.

Для выявления возможного цветения на одной из станций был выполнен кластерный анализ методом k-средних по данным мониторинга, результаты которого представлены на рис. 1. Все точки разделились на два кластера. Синяя точка, единственная точка второго кластера, является выбросом, что вероятно означает, что ей соответствует «цветение». Исходя из результатов кластерного анализа, делается вывод, что «цветение» происходит на 10-й станции. Данные по этой станции будут в дальнейшем использованы для подбора коэффициентов системы (2).

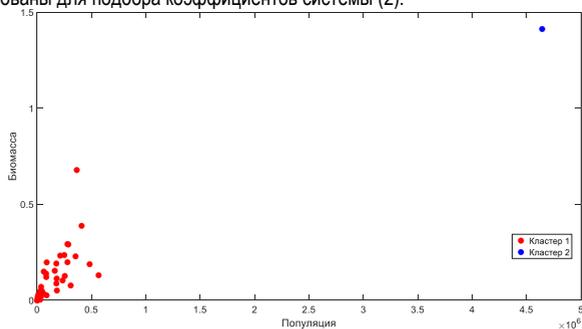


Рис. 1. Кластерный анализ

В данном датасете следует группировать станции по территориальному признаку (рис. 2), тогда в качестве группы для апробации были взяты станции № 6, 10 и 11 (выделены красным контуром).



Рис. 2. Карта расположения станций в Невской губе

Результаты применения метода подбора коэффициентов представлены в табл. 1.

Таблица 1

Коэффициенты для станций № 6, 10, 11

Станция № 6	0,12	0,05	0,11	0,02
Станция № 10	0,07	0,07	0,03	0,04
Станция № 11	0,07	0,06	0,02	0,02
Среднее значение	0,09	0,06	0,05	0,03

Полученные значения будут применяться при построении моделей на основе системы «хищник – жертва с питанием» (2) для прогнозирования возможного «цветения». Далее следует подобрать значения T_1, T_2 (коэффициенты, влияющие на время достижения цели) для целевого значения x^* . Для этого необходимо задать целевое значение. Предположим, что $x^* = 20 \cdot x_1(0)$. Модель, построенная на основе найденных коэффициентов и удовлетворяющая всем критериям, представлена на рис. 3.

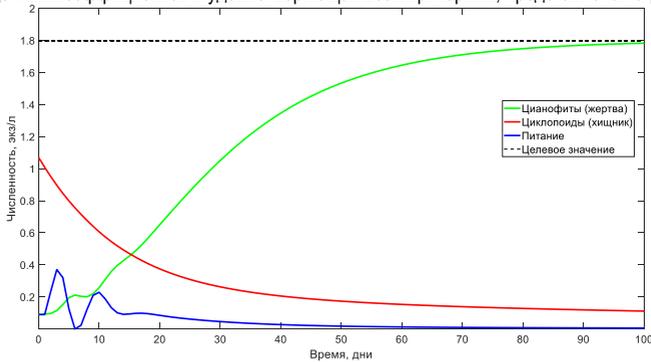


Рис. 3. Модель с управлением ($a_1(0) = 0.09, a_2 = 0.06, \beta_1 = 0.05, \beta_2 = 0.03, T_1 = 5.4, T_2 = 18$)

Апробация по данным Шведского института гидрологии и метеорологии

Как отмечалось ранее, датасет по Финскому заливу не содержит достаточного количества измерений гидробиологических показателей. Поэтому апробация метода подбора коэффициентов проводилась также по открытым данным Шведского института гидрологии и метеорологии за 2019 г. (рис. 4) [10].

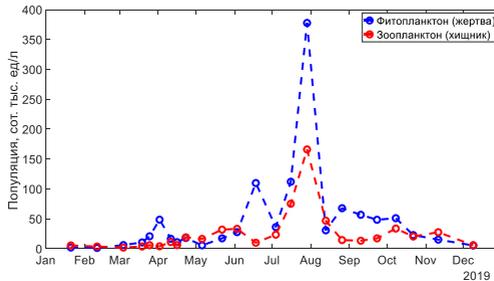


Рис. 4. Данные по популяциям фитопланктона и зоопланктона на станции B1

Преимущество этого датасета – большое количество измерений, сделанных на станциях в течение календарного года. В качестве апробации была выбрана станция с возможным «цветением». Например, станция В1 имеет заметный всплеск численности фитопланктона и зоопланктона в конце июля.

Далее были определены виды, чья численность повышается во время «цветения». Наибольшую популяцию имеет вид *Aphanocapsa* (рис. 5, а). Он относится к отделу цианобактерий (сине-зеленые водоросли) и, соответственно, может вызывать вредоносное цветение, поэтому был выбран в качестве жертвы [11]. Аналогичный анализ был проведен и для зоопланктона. Хотя наибольшую популяцию имеет вид *Tintinnida*, но поскольку по нему нет данных до июля, то в качестве хищника был выбран вид *Keratella quadrata* (рис. 5, б).

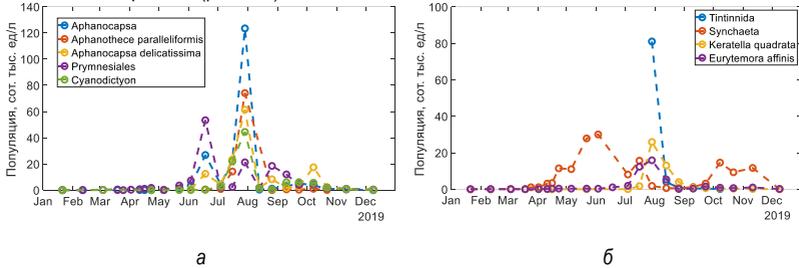


Рис. 5. Данные по видам фитопланктона и зоопланктона на станции В1:
а – фитопланктон, б – зоопланктон

После определения начальных данных был применен метод подбора коэффициентов. Следует отметить, что расчет «ошибки» производился исключительно по данным одной станции. Были получены следующие коэффициенты (табл. 2).

Таблица 2

Коэффициенты для станции В1

Станция В1	0,06	0,1	0,03	0,01
------------	------	-----	------	------

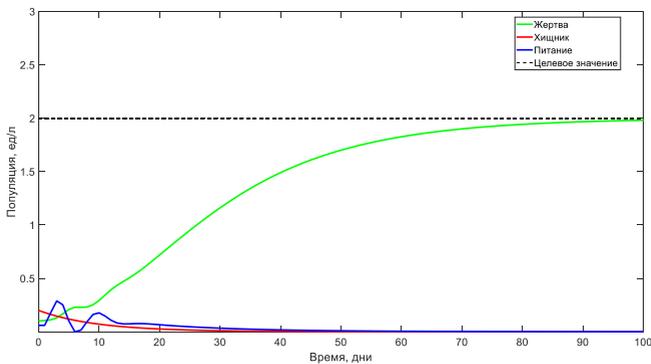


Рис. 6. Модель с управлением ($a_1(0) = 0.06, a_2 = 0.1, \beta_1 = 0.03, \beta_2 = 0.01, T_1 = 5.6, T_2 = 18$)

Полученные значения будут применяться при построении моделей на основе системы «хищник – жертва с питанием» (2) для прогнозирования возможного «цветения». Далее следует подобрать значения T_1, T_2 (коэффициенты, влияющие на время достижения цели) для целевого значения x^* . Чтобы подобрать их значения, необходимо задать целевое значение. В качестве примера предположим, что $x^* = 20 \cdot x_1(0)$. Модель, построенная на основе найденных коэффициентов и удовлетворяющая всем критериям, описанным выше, представлена на рис. 6.

Заключение

Была апробирована модифицированная система «хищник – жертва с питанием» на данных гидробиологического мониторинга Невской губы и Финского залива (побережье Швеции). Предложен метод выбора видов фитопланктона и зоопланктона для построения моделей. Модели, построенные по коэффициентам, подобранным с помощью алгоритма, являются устойчивыми, целевое значение в них достигается, что говорит об эффективности алгоритма. Также метод может быть модифицирован, если исходные данные имеют малое количество измерений.

Библиографический список

1. Lotka A. J. Contribution to the Theory of Periodic Reaction // The J. of Phys. Chemistry A. 1910. № 3. P. 271–274.
2. Volterra V. Variations and Fluctuations of the Number of Individuals in Animal Species living together // ICES J. of Marine Science. 1928. № 3. P. 3–51.
3. Shchegoleva A. The mathematical model of the biological system «predator-prey with food» // Bul. of the UNESCO dep. «Distance education in engineering» of the SUAI: Coll. of the papers. Saint-Petersburg, 2020. P. 112–117.
4. Щеголева А. А., Поляк М. Д. Модель «хищник – жертва с питанием» // Обработка, передача и защита информации в компьютерных системах '21: Междунар. науч. конф.: сб. докл., СПб., 14–22 апр. 2021 г. СПб., 2021. С. 86–91.
5. Колесников А. А. Синергетика и проблемы теории управления. М.: ФИЗМАТЛИТ, 2004. 504 с.
6. Колесников А. А., Колесников Ал. А., Кузьменко А. А. Методы АКАР и АКОР в задачах синтеза нелинейных систем управления // Мехатроника, автоматизация, управление. 2016. № 10. С. 657–669.
7. Колесникова С. И. Синтез управления нелинейным объектом второго порядка с неполным описанием // Автоматика и телемеханика. 2018 № 9. С. 18–30.
8. Shchegoleva A. Investigation of phytoplankton dynamics in the Neva bay with the «predator-prey with food» model // Bul. of the UNESCO dep. «Distance education in engineering» of the SUAI : Col. of the papers. Saint-Petersburg, 2021. P. 129–136.
9. Ежегодник состояния вод Невской губы и восточной части Финского залива по гидробиологическим показателям в 2012 г. Т. 2 / исполн.: И. В. Андреева [и др.]. СПб.: ФГБУ «Санкт-Петербургский ЦГМС-Р», 2013. 352 с.
10. Swedish Ocean Archive (SHARK), DATE (2021-05-29) / The Swedish Agency for Marine and Water Management and the Swedish Meteorological and Hydrological Institute. URL: www.smhi.se/en/services/open-data/national-archive-for-oceanographic-data (дата обращения: 29.05.2021).
11. Algal bloom and its economic impact / I. Sanseverino, D. Conduto, L. Pozzoli [et al.]; Europ. Commiss., Joint Res. Centre Inst. for Environment and Sustainability. Ispra, 2016. 48 p.

УДК 004.92

И. А. Александров

магистрант кафедры вычислительных систем и сетей

А. В. Никитин – кандидат технических наук, доцент – научный руководитель

РАЗРАБОТКА КООПЕРАТИВНОЙ ИГРЫ В СТИЛЕ ШУТЕР НА ИГРОВОМ ДВИЖКЕ UNREAL ENGINE 4

Разработка компьютерных игр в настоящее время – целая индустрия. В статье рассматривается один из видов многопользовательских компьютерных игр, который принято называть кооперативной игрой. Такая игра подразумевает, что два и более игрока будут играть на одной карте и выполнять вместе какие-либо задания (например, уничтожение монстров, зачистка территории и т. д.) в одной команде. Для создания игры используется игровой движок Unreal Engine 4, который появился в 1998 г. и уже тогда стал одним из лучших для написания игр [1].

Классический подход разработки игры заключается в том, что нужно писать объемный исходный код построчно, например на C++, для создания ядра (игровой платформы), внедрения 3D-моделей, создания игровой логики. В этом случае требуются высокий уровень знаний языка программирования и высококвалифицированные программисты. Уровень знаний у разработчика игры непосредственно влияет на время разработки. Сейчас компьютерные технологии и требования к качеству игр быстро развиваются. Одними из ключевых параметров стали скорость разработки и качество графики. Разработчик, который быстро и своевременно выпустил игру под требования игроков и обеспечил качество графики лучше, чем у конкурентов, становится востребованным на рынке.

Решением может стать использование готовой игровой платформы Unreal Engine 4, которая позволяет уменьшить время, затрачиваемое на разработку игры, и обеспечить требуемое качество. Unreal Engine 4 имеет стандартный набор инструментов и механик для создания игр, кроме того, разработчики игровой платформы выпускают обновления для улучшения качества самого движка. В движке Unreal Engine 4 используется визуальное программирование Blueprint. Оно требует менее высокого уровня у программиста, все функции работают за счет подключения друг к другу блоков (нод), одна нода может содержать несколько строк кода.

Этапы разработки кооперативной компьютерной игры на игровом движке UE4 [2].

Первый этап – проектирование. Был выбран жанр кооперативной компьютерной игры в стиле шутер. Придуман сюжет: на военной исследовательской базе произошла авария и команда военных спасателей приходит, чтобы узнать, что произошло. Нужно выполнить ряд простых заданий, таких как: подать питание, включить свет и отбиваться от монстров. Игра пишется на визуальном языке программирования blueprint, придуманном Unreal Engine 4. Создается десктопное приложение с возможностью в будущем перевести игру в виртуальную реальность.

Второй этап – непосредственно создание игры. Проектирование и создание уровня игры путем использования готовых 3D-моделей зданий, лестниц, крыш и т. д., нарисованных в графическом редакторе Blender и импортированных в игровой движок. На карте расставлены объекты, созданы материалы, прописаны взаимодействия с объектами. Также готово управление персонажем, создан примитивный ИИ для монстров, добавлены физические свойства объектам. Подготовлены анимации для 3D-объектов, монстров, добавлены различные звуки на карту, например фоновый звук, звук шагов. Создаются виджеты главного меню, настройки и т. д.

На последнем этапе происходит тестирование, ищутся и исправляются ошибки.

Создание проекта в UE4 от первого лица, в котором уже есть базовые настройки персонажа: он может ходить, прыгать, смотреть по сторонам, стрелять, показано на рис. 1.



Рис. 1. Создание игры от первого лица

В UE4 автоматически была создана базовая карта, измененная впоследствии. Перед началом разработки настроены папки для удобства. UE4 сразу предоставил стартовый контент, в котором есть материалы, текстуры, 3D-модели, аудио, анимация и т. п. для создания кооперативной игры.

Далее был построен игровой уровень (рис. 2), ускорению работы помогли готовые ресурсы (assets), в которых находятся готовые 3D-модели, скачанные из магазина UE4, также 3D-модели создавались в графическом редакторе Blender и импортировались в формате fbx непосредственно в игровой движок. Каждую модель в UE4 можно вращать, передвигать, изменять ее размер, что удобно для построения уровня и не зависит от размера 3D-модели.



Рис. 2. Построенная карта первого уровня игры

На каждую модель наложен материал. Материал импортировался вместе с моделью или был создан непосредственно в UE4.

На карте уже поставлены все объекты, сделана установка света. В UE4 освещенности сцены уделено много внимания для создания большей реалистичности при визуализации, здесь есть и общий свет, и направленный, также имеются карты света, динамический и статический свет. Свету было уделено особое внимание, так как в зависимости от того, насколько производительна система, сделаны настройки его параметров. В каких-то моментах поставлен статический свет, т. е. тень от объектов не будет отображаться в реальном времени, что избавляет от лишних расчетов, а также уменьшает нагрузку на систему. На место на карте, где нужно более детально показать какую-либо зону или объ-

ект, поставлен динамический свет, который обрабатывается каждый тик и приводит к более реалистичному виду (например, тень от персонажа).

Далее было создано главное меню (рис. 3). Для этого применялся виджет (Widget Blueprint), в котором был создан внешний вид с использованием базовых инструментов.



Рис. 3. Вид главного меню

Использовались вертикальные и горизонтальные боксы, кнопки, слайдеры. Также применялись изменение цвета кнопок, создание заднего фона, вставлена анимация, монстр в главном меню шевелится. После создания главного меню сформированы виджеты меню новой игры и все остальные, которые перечислены в главном меню (см. рис. 3).

Blueprint-программирование – это нодовая система, в которой каждая нода отвечает за какое-либо действие. Как и в других языках программирования, тут есть переменные, массивы, циклы и т. д.

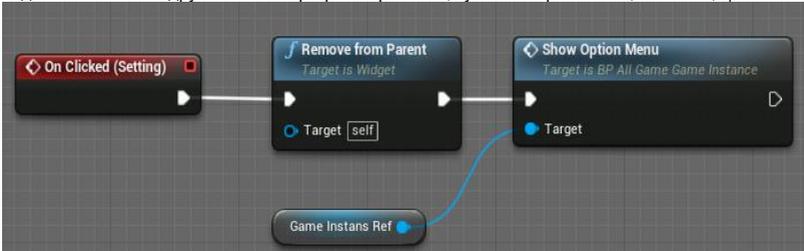


Рис. 4. Пример blueprint-нажатия кнопки

Для кооперативной игры необходимо создать сервер. В UE4 есть выбор.

Listen-сервер, когда игрок, который запускает у себя на компьютере игру и сервер в одном процессе, является хостом. В данном случае есть большой минус: если игрок, который является хостом, выйдет из игры по какой-либо причине, то у всех остальных игроков, которые к нему подключились, также произойдет выход из игры. Еще один немало важный минус в том, что тот, кто является хостом, имеет доступ к серверу и может мошенничать, по этой причине Listen-сервер используют только в играх, где нет нужды применять читерство (cheat) (получение преимущества за счет сторонних программ). Плюсы: Listen-сервер легко настраивается, легко запускается, и, если нужно поиграть в кооперативный шутер с друзьями, такой тип сервера более чем подходит.

Второй вид сервера – Dedicated (выделенный сервер) – запускается как отдельный процесс, как на компьютере клиента, так и на отдельной машине. Он лучше подходит для кибер-спорта или игр, где игроки не имеют доступа на сервер без прав администратора. Такой сервер сложнее настраивать, чем Listen-сервер, но так как это отдельный процесс, то игроки не зависят друг от друга.

Игровая карта создана, так же сделано меню, и можно зайти в игру, где поставлены монстры на сцену и создан примитивный искусственный интеллект, т. е. игровой персонаж, не управляемый игро-

ком: моб (mob) ходит по определенной территории, и если игрок попадает в зону действия атаки моба, то он нападает на игрока. Пример разработки искусственного интеллекта на blueprint приведен на рис. 5.

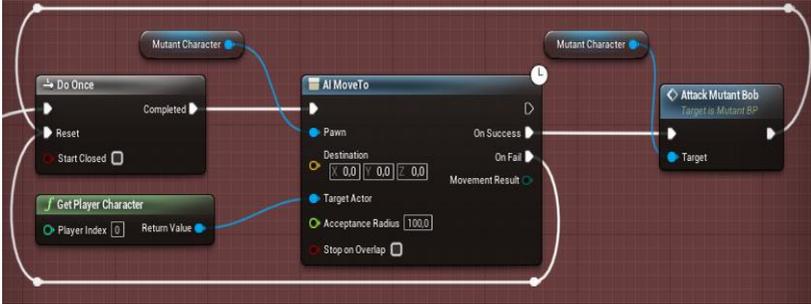


Рис. 5. Blueprint-нападение монстра на игрока

В любой игре есть процессы, которые не зависят от игрока и происходят только на сервере, например мигающая лампочка, вращающийся вентилятор, искрение провода, горение костра и т. д. Такие процессы прописываются с помощью blueprint и выполняются как отдельные маленькие программы.

Плюсы blueprint.

Программирование на blueprint удобно благодаря блочному синтаксису, который интуитивно понятен и работать с ним могут не только программисты, но и люди, которые только начинают знакомиться с программированием, а также те, кто знаком из других программ с блочным синтаксисом. Этот мощный инструмент используется для построения игровой логики, работы с анимацией и системой эффектов. Также хорошо использовать в проектах, где есть много ссылок на контент и логика больше работает на визуальную составляющую.

Минусы blueprint.

Blueprint бьет по производительности, потому что работает на встроенной в движок виртуальной машине. Также сложно совместить проекты, над которыми трудятся разные люди, по сравнению с написанием кода классическим способом. Наконец, водить мышкой по полю и выставлять блоки и соединять их получается дольше, чем при написании кода построчно, учитывая современные технологии, которые помогают программисту.

Таблица 1

Сравнение с аналогами

Показатель	Unreal Engine 4	Unity	Godot
Бесплатная	+	+	+
Мультиплатформенность	+	+	-
Выбор ресурсов (assets)	+	+	-
Удобный интерфейс	+	+/-	+/-
Качественная графика	+	+/-	-
C#	-	+	+
C++	+	-	+
Blueprint	+	-	-
Готовность к работе	+	+	+/-
Большое количество документации	+	+	-

Сравнение одной технологии с другой субъективно. Каждый популярный движок – это огромный пласт информации с большим сообществом, различиями в версиях, лучшими практиками применения и серьезным функционалом [3].

Методика разработки игры на игровом движке UE4 экспериментально проверена и научно обоснована. Использование визуального программирования blueprint в нем ускоряет процесс разработки игр. Качество графики отвечает современным требованиям.

Библиографический список

1. Русскоязычное сообщество Unreal Engine 4. URL: <https://uengine.ru/docs> (дата обращения: 23.03.2021).
2. Компьютерные игры как искусство // GamesIsArt.ru. URL: https://gamesisart.ru/game_dev_create.html (дата обращения: 23.03.2021).
3. Пекач. Блоги. URL: <https://sc2tv.ru/blogs/falcon/2019/07/23/sravnienie-igrovyykh-dvizhkov> (дата обращения: 24.03.2021).
4. Уроки Unreal Engine 4 // Уроки, документация, полезные ресурсы по Unreal Engine 4. URL: <https://uedev.blogspot.com/> (дата обращения: 23.03.2021).
5. Unreal Engine 4 documentation. URL: <https://docs.unrealengine.com/en-US/index.html> (дата обращения: 23.03.2021).

УДК 004.946

А. А. Антипова, Н. В. Салова

магистранты кафедры вычислительных систем и сетей

А. В. Никитин – кандидат технических наук, доцент – научный руководитель

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ ДЛЯ ОБУЧЕНИЯ ГИБКИМ НАВЫКАМ

В современном мире для достижения профессионального и личного успеха человеку необходимо обладать не только специфичными знаниями по специальности, но и уметь общаться с людьми, эффективно организовывать свое время, творчески мыслить, принимать решения и брать на себя ответственность, эти качества объединены в понятие «гибкие навыки» (soft skills). Для их приобретения различные организации готовят курсы, как в очном, так и в онлайн форматах, в том числе с использованием виртуальной реальности.

Виртуальная реальность (VR, VR) «относится к любой ситуации, когда искусственно создается ощущение пребывания человека в определенной среде», которое достигается при помощи компьютерных средств [1].

Основное преимущество обучающих VR-приложений перед другими методиками обучения – максимально глубокий уровень погружения в ситуацию. С помощью данных технологий любой заинтересованный темой человек может в разы эффективнее освоить навык или обучиться новому через активный процесс обучения и взаимодействия, а работодатели и преподаватели с большей вероятностью могут быть уверены, что формируют у своих специалистов и студентов одинаковое представление о рассматриваемых процессах [2].

В России технологии VR и AR для различных областей знаний стали популярны в последние несколько лет, и многие компании уже начали внедрять их в свои решения. В настоящее время, согласно данным аналитического агентства KPMG за 2019 г., VR/AR-технологии уже используют 21 % крупнейших отечественных компаний. Вместе с тем развитие рынка VR имеет ряд причин, препятствующих его полномасштабному распространению. К факторам, тормозящим внедрение VR в российских компаниях, относятся:

- недостаточная зрелость процессов или низкий уровень автоматизации;
- отсутствие компетенций;
- низкий уровень ИТ-грамотности сотрудников;
- отсутствие необходимой инфраструктуры;
- отсутствие достаточного бюджета [3].

Типичными причинами перехода предприятий к работе в виртуальной реальности являются необходимость внедрения передовых технологий и освоение новых инструментов, повышение производительности и эффективности процесса, сокращение трудозатрат и издержек.

Несоответствие между стремлением предприятий сократить различные расходы за счет использования технологий виртуальной реальности и отсутствием общепринятых стандартов для планирования и организации перехода к виртуализации процессов обуславливает возникновение вопросов о том, какие технологии виртуальной реальности необходимо использовать для достижения необходимых компаниям целей, как с их помощью обучать персонал гибким навыкам (а главное, какими конкретно им нужно овладеть) и как обосновать целесообразность такого решения. Для их решения необходима разработка научно обоснованной методики перехода от очного обучения гибким навыкам к получению их с использованием VR/AR-технологий, что определяет актуальность темы исследования.

Мы предлагаем решение объединения подхода к обучению гибким навыкам путем подачи информационной части и геймификации с помощью средств виртуальной реальности. Цель работы – исследовать возможность виртуальной среды для дальнейшего проектирования в ней игрового образовательного приложения по обучению гибким навыкам, с элементами геймификации и интерактивностью, с возможностью вовлечения пользователя в сюжет сценария, что позволит экономить компаниям время и ресурсы для приобретения необходимых знаний по гибким навыкам их сотрудниками.

Отсюда формируются задачи:

- 1) выделить и проанализировать основные аналоги приложений по обучению гибким навыкам;
- 2) выбрать наиболее актуальные направления для исследования. Исходя из анализа источников, наиболее популярные гибкие навыки – коммуникативные, связанные с умением общаться с людьми, а также тайм-менеджмент, направленный на самоорганизацию времени [4];
- 3) проанализировать источники и сформировать набор понятий для обучения гибким навыкам в VR;
- 4) спроектировать прототип предлагаемого приложения.

Для создания обучающего гибким навыкам геймифицированного приложения стоит опираться на социальные компетентности. Понятием «компетентность» определяют такие факторы, как способности, личностные качества, возможности личности. Специалисты по психологии под компетентностью подразумевают «владение определенными знаниями, навыками, жизненным опытом, позволяющими судить о чем-либо, делать или решать что-либо» [3]. С опорой на экспертное мнение под социальными компетентностями будем понимать такие качества, как «коммуникативные способности, социальная адаптивность, уверенность в себе, социальная мобильность, креативность, эмпатийность, критичность мышления» [5] и т. д. В работе рассматриваются два метода – коммуникативный и тайм-менеджмент, как одни из самых актуальных в современном мире [4]. Перейдем к рассмотрению признаков существующих решений, представленных в табл. 1.

Таблица 1

Сравнение аналогов по выбранным навыкам

Аналог № 1	Аналог № 2	Экспериментальный образец
Интерактивность	+	+
Контроль за выполнением заданий	–	+
		Погружение в среду
		Геймификация

1. Коммуникативный навык

Аналог № 1 – Lloyds Banking Group – Personal Vitality & Resilience [6].

Приложение, разработанное компанией Lloyds Banking Group, в VR обучает в основном навыку стрессоустойчивости: у игрока есть возможность проходить различные уровни, соответствующие определенным ситуациям, которые могут возникнуть в работе [6]. В качестве недостатков отметим небольшое количество взаимодействий, невозможность изменить сюжет и ограничение по времени – 5 мин.

Анализ других аналогов приложений по soft skills [6] показал, что на рынке не хватает приложений с использованием AR/VR, направленных на изучение нескольких гибких навыков с возможным влиянием пользователя на сюжет. Реализация итогового геймифицированного приложения обусловлена тем, в игровом формате подача информации будет более интересна и легко усваиваема, а влияние на сюжет позволит удерживать внимание пользователя. Прототип было решено делать в виде приложения для ПК с целью тестирования пригодности выбранной среды.

Цель приложения – обучение людей социальным взаимодействиям. Выбранный навык – информационно-коммуникативный, т. е. способность вступать в общение, ориентироваться в беседах и ситуациях, соотносить разные способы общения (вербальные/невербальные) [7], в прототипе это умение слушать другого человека и воспринимать информацию.

Сюжет игры развивается в виртуальном университете. Основное место действия – двор, также у игрока есть доступ в здание и аудитории. Для достижения правдоподобности у пользователя будет собственный аватар, от лица которого он будет общаться и взаимодействовать с другими персонажами в игре, идущей от первого лица. Диалоги отображаются на панелях. Управление действиями происходит с помощью мыши и клавиатуры. Задания, которые необходимо выполнить, отображаются в левой части экрана и исчезают по завершении, они не нагружают интерфейс, который ограничен панелью с диалогами и главным меню. Модели, содержащиеся в игре, – аватар куратора, деревья, скамейки, само здание, стол, стул и различные канцелярские предметы. Инструменты, использованные при создании самого прототипа, – движок Unity и язык программирования C#. Тесты проходили на Windows 10.

2. Навык тайм-менеджмента.

На сегодняшний день одно из самых популярных направлений по развитию гибких навыков – тайм-менеджмент [4], направленный на достижение максимального результата путем управления собственным временем с максимальной эффективностью.

Аналог № 2 – Helmenton – «VR-обучение для компании „Газпром Нефть“» [8].

«VR-обучение для компании „Газпром Нефть“» рассчитано на тренировку soft skills навыков для новых сотрудников и студентов; «Обучающее VR-приложение для ТФН» – на полный цикл обучения и тестирования сотрудников в виртуальной реальности на знание материала об опасных веществах.

Для исследования возможностей технологии виртуальной реальности с использованием интерактивного веб-приложения для развития soft skills навыка тайм-менеджмента, можно использовать когнитивное искажение «ошибки планирования» [10], связанное с излишним оптимизмом и недооценкой времени, требуемого для выполнения задачи.

Один из способов научить человека правильному планированию – поместить его в заранее смоделированную среду, выяснить, сколько, по его мнению, времени займет выполнение действия, а затем попросить выполнить это действие и посмотреть, сколько времени потребуется на самом деле. С использованием технологии виртуальной реальности возможно геймифицировать процесс обучения и разбить его на следующие этапы.

Первый этап будет состоять из набора жизненных ситуаций с препятствиями, которые помогут более объективно закладывать время в расписании на их выполнение.

Второй этап – избавление от отвлекающих факторов. Большинство людей подвержены чрезмерному потреблению контента из социальных сетей, а также постоянному отвлечению на сообщения из мессенджеров, что не учитывается при планировании времени. Решение – поместить человека в среду, где ему необходимо выполнить рутинную задачу, и добавить отвлекающие факторы. Завершение данного сценария – показать, сколько человек мог потратить на эту задачу времени и сколько действительно затрачено.

Третий этап – планирование собственного расписания. На данном этапе человек должен изменить все полученные знания для того, чтобы оценить и спланировать свою рабочую неделю. Таким образом, будет проведена оценка того, насколько эффективным оказался процесс обучения (все задачи будут заранее оценены и будет рассчитана разница между оценкой человека и предполагаемой, но с небольшой погрешностью).

Таким образом, выбранный подход является улучшенным по сравнению с существующими решениями за счет одновременной доступности приложения с различных мобильных и десктопных устройств, что позволяет использовать обучающие технологии более широкому кругу пользователей, а также за счет создания условий для обучения с погружением в рассматриваемый опыт.

В процессе работы были разработаны экспериментальные образцы по обоим навыкам. Скриншоты фрагментов прототипов представлены на рис. 1.



Рис. 1. Примеры скриншотов по коммуникативному навыку и тайм-менеджменту

Результатом исследования стал разработанный и реализованный первый базовый прототип технического решения геймифицированного интерактивного приложения с влиянием пользователя на сюжет по изучению гибких навыков. Были проанализированы основные материалы, касающиеся использования виртуальной и дополненной реальности для обучения soft skills, сделаны выводы о преимуществах внедрения технологий AR/VR для компаний и отдельных людей. Рассмотрены основные аналоги приложения, недостаток которых заключается в избытке информационной части и слабой отработке полученных знаний в интерактивном режиме. Были выбраны направления гибких навыков – информационно-коммуникативные навыки, являющиеся основополагающими при организации взаимодействия между людьми, и тайм-менеджмент. Следует отметить, что разработанные прототипы – это первый шаг к реализации задуманной системы, направленной на изучение и освоение этих навыков.

Библиографический список

1. Никитин А. В., Решетникова Н. Н., Ситников И. А. Цифровые реальности: основные понятия и применения: учеб. пособие, СПб.: ГУАП, 2020. 109 с.
2. Разработка обучающих приложений в VR для soft skills. URL: <https://helmeton.ru/blog/razrabotka-obuchayushchikh-prilozheniy-v-vr-dlya-soft-skills/> (дата обращения: 23.05.2021).
3. Цифровые технологии в российских компаниях. Результаты исследования. URL: <https://assets.kpmg/content/dam/kpmg/ru/pdf/2019/01/ru-ru-digital-technologies-in-russian-companies.pdf> (дата обращения: 04.03.2021).
4. Какие soft skills нужны в IT и как их развивать. URL: <https://issoft.by/blog/kakie-soft-skills-nuzhny-v-it-i-kak-ikh-razvivat/> (дата обращения: 07.05.2021).
5. Шарова Н. В. Социальная и конфликтная компетентность в процессе профессионализации личности. URL: <https://sites.google.com/site/konferp/Home/2-sekcia/sarova> (дата обращения: 17.03.2021).
6. 5 Best Uses of VR for Soft Skills Training. URL: <https://www.immersivelearning.news/2020/04/09/5-best-uses-of-vr-for-soft-skills-training/> (дата обращения: 17.03.2021).
7. Шарова Н. В. Социальная и конфликтная компетентность в процессе профессионализации личности. URL: <https://sites.google.com/site/konferp/Home/2-sekcia/sarova> (дата обращения: 26.03.2020).
8. Helmeton – VR-обучение для компании Газпром Нефть. URL: <https://helmeton.ru/portfolio/vr-obuchenie-dlya-kompanii-gazprom-neft/> (дата обращения: 04.04.21).
9. Modum – VR-симуляция КОВ (Комиссия по оценке вклада сотрудника). URL: <https://modumlub.com/portfolio/kov> (дата обращения: 04.04.21).
10. Ошибка планирования. URL: <https://nplus1.ru/blog/2019/01/14/planning-fallacy> (дата обращения: 23.05.2021).

УДК 004.94

А. А. Антонов, Г. В. Дук

магистранты кафедры вычислительных систем и сетей

С. А. Чернышев – кандидат технических наук – научный руководитель

ПРИМЕНЕНИЕ МУЛЬТИАГЕНТНОГО ПОДХОДА ПРИ РАЗРАБОТКЕ ИМИТАЦИОННЫХ МОДЕЛЕЙ

Прогнозирование вероятности наступления того или иного события востребовано во многих сферах: научной, коммерческой, медицинской и военной. Каждое из возможных событий зависит от множества факторов и их взаимосвязей. И из-за их постоянного роста просчет таких событий становится все более трудоемким. В последнее время большую популярность набирает имитационное моделирование при решении такого типа задач [1]. Имитационная модель – это компьютерная программа, которая описывает структуру и воспроизводит поведение реальной системы во времени [2]. Главная особенность такого подхода – решение проблем, которое нельзя получить в естественных условиях и на реальных объектах.

Существует пять основных подходов для разработки имитационных моделей: динамическое моделирование, системная динамика, метод Монте-Карло, дискретно-событийный и мультиагентный подход. Последний имеет наибольшую популярность, применяется в исследованиях искусственного интеллекта, сложных систем, теории игр и т. д. Агент – это некий объект, способный принимать решения на основе заложенных правил, взаимодействовать с другими сущностями системы, проявлять активность и самостоятельно действовать [3]. Важная отличительная характеристика данного подхода состоит в децентрализации агентов. Это значит, что глобальное поведение системы складывается из множества локальных поведений каждого агента.

Для исследования мультиагентного подхода необходимо выбрать такую область, которая будет соответствовать понятию сложной адаптивной системы [4], т. е. будет состоять из автономных элементов, вступающих в связи и разрывающих эти связи в результате взаимодействий и собственных решений. Такой областью являются грузоперевозки.

В описанной системе хорошо прослеживаются действующие лица, что позволяет разделить поведение системы на две категории:

- 1) локальное поведение, выделяется на основе индивидуальных решений и правил конкретных агентов;
- 2) глобальное поведение, формируется за счет совокупности действий каждого агента.

Действия всех агентов системы должны быть синхронизированы по времени и иметь четкий контроль над объектами имитационной модели. Для этого вводится дискретно-событийный подход [5], основанный на замене обычного времени на дискретное. Отсюда модель представляется как список последовательных событий, позволяющих переходить системе из одного состояния в другое.

При комбинировании двух подходов контроль модели осуществляется как по времени, так и по поведению агентов. Имея доскональный контроль системы, можно детально рассмотреть все влияющие факторы, а также после получения результата вернуться назад и внести корректировки.

На основе анализа предметной области были введены три действующие популяции агентов: магазины, склады и транспорт [6]. Например, магазин имеет различные типы (гипермаркет, супермаркет и т. д.), расположение и время работы. Агенты из популяции «Магазины» занимаются закупкой и продажей товаров, в результате чего высчитывается их популярность, на основе которой увеличивается влияние на образование цен на рынке.

Поставщиком товаров для магазина является склад. Он занимается хранением товаров, регулированием цен поставки и пополнением товаров от различных поставщиков.

Транспорт представляет интересы компаний по грузоперевозкам. Основная функция – перевозка товаров и расчет необходимого количества транспорта, его типа и габаритов для выполнения заказа.

Для гарантированного выполнения обязательств вводится система контрактов, которые заключаются между агентами. В них прописываются участники, сроки, цена и условия. Нарушение контракта или его расторжение влечет последствия для обеих сторон.

Один из возможных вариантов развития событий в системе – возникновение у магазина необходимости в пополнении товара. Он формирует запрос для поиска склада с наименьшей стоимостью товаров, не превышающую цену продажи. При формировании контракта между машиной и магазином выбирается наилучшая комбинация цены работы и длины дороги, а также учитывается максимальная грузоподъемность машины. После заключения контракта между всеми заинтересованными сторонами начинается пошаговое моделирование.

Для взаимодействия пользователя с моделью была разработана графическая оболочка, позволяющая наглядно отслеживать действия агентов, вносить корректировки и анализировать полученные результаты.

На рисунке изображен графический интерфейс приложения, благодаря которому можно точно управлять сложной адаптивной системой: на рис. 1 – несколько агентов из популяций транспорт, магазин и склад, проложены маршруты перевозки товаров; на рис. 2 – окна редактирования параметров магазина и склада, где можно задавать товары, цены и их количество, также видна его популярность.

По результатам экспериментов, каждый агент стремится к получению выгоды, а, следовательно, вся система стремится к выгоде. Это можно заметить по изменению рейтинга, который высчитывается на основе количества проданного товара. Была разработана тайлова картографическая библиотека на языке программирования Python, которая на текущий момент не имеет аналогов. Предложен подход к разработке сред мультиагентного моделирования с картографической основой на языке программирования Python, из-за ограничений которого основными языками программирования для мультиагентных систем выступают C++ и Java.

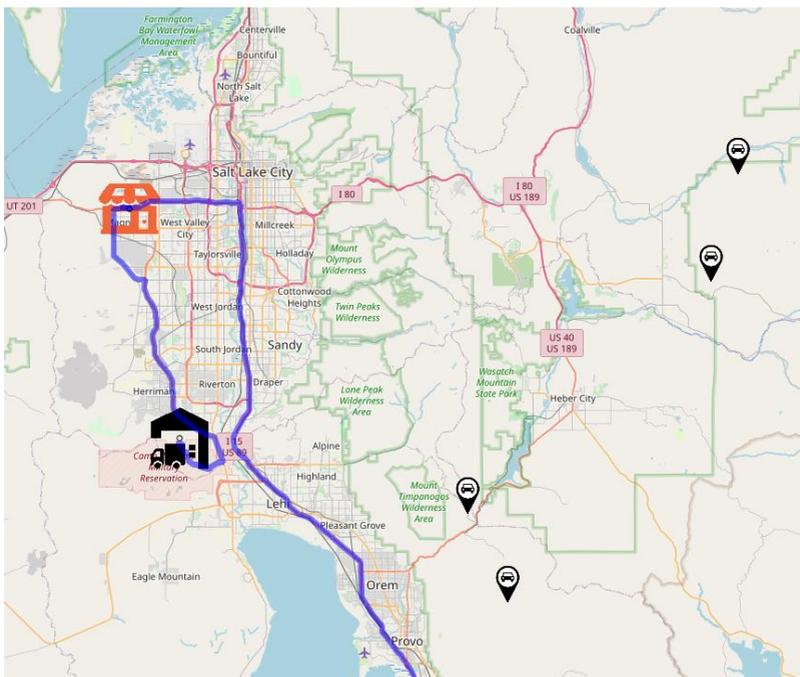


Рис. 1. Демонстрация работы приложения

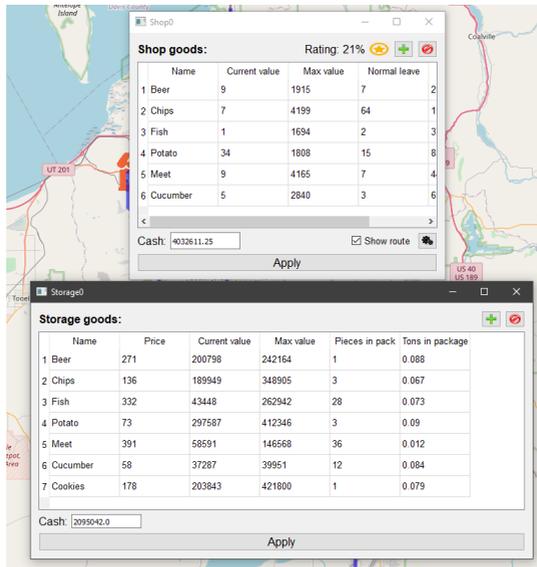


Рис. 2. Окна редактирования параметров магазина и склада

Законченная среда для мультиагентного моделирования грузоперевозок позволяет отслеживать и моделировать распределение грузопотоков по поступающим заявкам таких компаний, как агентство по грузоперевозкам.

Библиографический список

1. Laguna M., Marklund J. Business process modelling, simulation and design. L., N.-Y.: CRC Press, 2013. 523 p.
2. Эльберг М. С., Цыганков Н. С. Имитационное моделирование: учеб. пособие. Красноярск: Сиб. федер. ун-т, 2017. 128 с.
3. Дук Г. В., Антонов А. А., Чернышев С. А. Анализ существующих подходов к проектированию имитационных моделей // Междунар. журн. информ. технологий и энергоэффективности. 2020. № 18. С. 9–14.
4. Rzevski G., Skobelev P. Managing complexity. UK: WIT Press, 2015, 290 p.
5. Голубков А. С., Царев В. А. Методы и алгоритмы микроскопического дискретно-событийного моделирования транспортных потоков в улично-дорожных сетях. // Матем. Моделирование. 2010. Т. 22, № 8. С. 33–41.
6. Миротин Л. Б., Лебедев Е. А., Покровский А. К. Инновационные процессы в логистике. М.: Инфра-Инженерия, 2019. 392 с.

УДК 621.396.967

В. И. Афанасьева

студент кафедры вычислительных систем и сетей

В. А. Ненашев – кандидат технических наук – научный руководитель

КЛАССИФИКАЦИЯ НАБЛЮДАЕМЫХ ЗОН НА СФОРМИРОВАННЫХ СНИМКАХ ПОТОКА ВИДЕОКАДРОВ В БОРТОВЫХ СИСТЕМАХ МОНИТОРИНГА ЗЕМНОЙ ПОВЕРХНОСТИ

Реализуется метод классификации наблюдаемых зон земной поверхности по снятым кадрам видеопотока, сформированным оптико-электронной системой наблюдения ЛА. Используются методы глубокого обучения, цифровой обработки изображений и семантической сегментации. Результат работы – классифицированные кадры видеопотока, сегментированные по следующим классам: небо, растительность и строение.

Введение

Классификация наблюдаемых зон по снятым аппаратурой ЛА кадрам – в последнее время одна из важнейших задач анализа сцен в приложениях автоматизированного мониторинга за земной поверхностью [1].

Обработка кадров видеопотока с помощью семантической сегментации позволяет разделить обрабатываемую сцену земной поверхности на классы. Целесообразность подобной интерпретации кадра заключается в том, что все большее число приложений переводит компьютерные данные в графический вид, который более понятен человеку [1].

Данный вид сегментации применяется в различных сферах. К ним относятся: автономные системы управления воздушным транспортным средством, в том числе беспилотным, продвинутые способы взаимодействия человека с компьютером, виртуальная реальность, навигация бортовых беспилотных систем и т. д. Таким образом, можно говорить об актуальности темы.

Задача исследования состоит в разработке метода автоматизированного выделения различных зон с последующей их классификацией на основе использования соответствующей нейронной сети.

Особенности классификации и сегментации изображений

Существует несколько видов сегментации, например сегментация экземпляра и семантическая сегментация изображений.

Сегментация экземпляра – это разделение изображения на отдельные области или классы, соответствующие конкретному объекту или зоне. Например, выделение области «небо», «дорога», «растительность», «строения» и т. д. Сегментация изображений применяется в системах содействия оператору автономных воздушных беспилотных транспортных средств. Семантическая сегментация изображений – это разделение изображения на отдельные группы пикселей, области, соответствующие одному объекту с одновременным определением типа объекта каждой области, т. е. его классификации [2].

Семантическая сегментация изображений, сформированных на борту беспилотных транспортных средств для классификации различных атрибутов, довольно сложная задача, так как различия очень велики, местность не всегда одинакова и однородна, что затрудняет процесс ее классификации. Таким образом, существует потребность в реализации подобных систем, которые автоматически осуществляют процесс сегментации в режиме квазиреального времени, используя сверточную нейронную сеть.

Семантическая сегментация отличается от сегментации экземпляра тем, что разные объекты одного и того же класса будут иметь разные метки, например разные дома будут помечены одним цветом и соответственно одной меткой «Building». На рис. 1 продемонстрировано различие данных видов сегментации.

Сверточная нейронная сеть – это концепция глубокого обучения, которая в данном случае применяется для работы с видеокдрами (рис. 2).

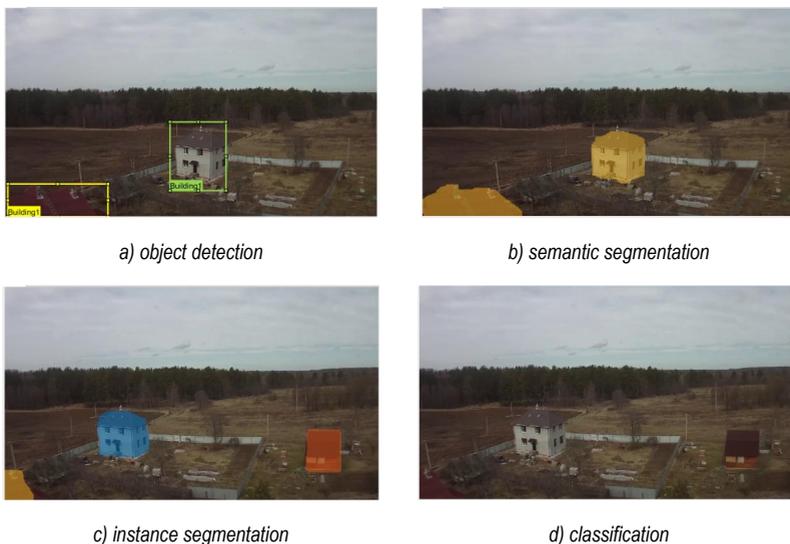


Рис. 1. Пример различных видов сегментации

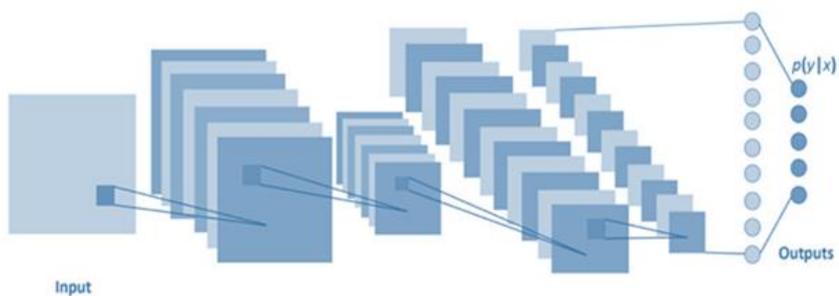


Рис. 2. Структурная схема распознавания объектов

На начальном этапе нейронная сеть является необученной (ненастроенной). В общем смысле под обучением понимают последовательное предъявление образа на вход нейросети из обучающего набора, затем полученный ответ сравнивается с желаемым выходом, полученная разница между ожидаемым ответом и полученным есть результат функции ошибки (дельта ошибки). Затем эту дельту ошибки необходимо распространить на все связанные нейроны сети (рис. 3) [3–5].

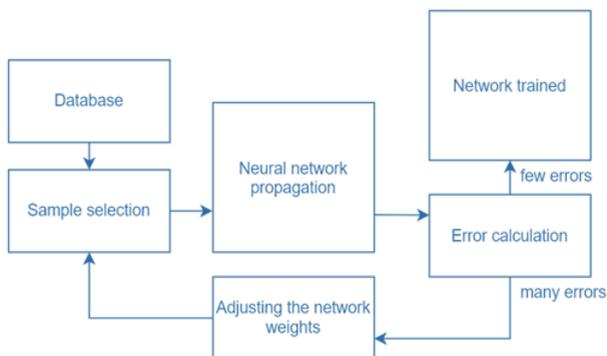


Рис. 3. Алгоритм обучения нейронной сети

В большинстве случаев наборы данных для обучения сети состоят не более чем из тысяч кадров. Обучение сети с нуля происходит, начиная со случайно инициализированных весов. Также уже предобученную сеть можно обучить на собственном наборе данных [6].

Таким образом, особенности рассматриваемого метода – эффективность и простота получения результата, высокая точность, легкость реализации.

Методы сегментации изображений

Первый шаг в процессе отслеживания объектов – их идентификация в видеопоследовательности и кластеризация их пикселей. Поскольку движущиеся объекты обычно являются основным источником информации, большинство методов основываются на обнаружении движущихся объектов. Основные методы обнаружения движущихся объектов:

- разность кадров;
- оптический поток;
- вычитание фона.

Метод разности кадров предполагает определение наличия движущихся частиц путем вычисления разности между двумя последовательными изображениями видеопоследовательности.

Метод оптического потока предназначен для вычисления поля оптического потока изображения и выполнения обработки собранной информации в соответствии с характеристиками распределения оптического потока изображения.

Первым шагом метода вычитания фона выступает фоновое моделирование, которое должно быть достаточно чувствительным, чтобы распознать движущиеся объекты. Фоновое моделирование – это создание эталонной модели. Она используется для вычитания фона, в котором каждая видеопоследовательность сравнивается с эталонной моделью, чтобы определить возможное изменение. Различия между текущими видеокдрами и неподвижными кадрами указывают на существование движущихся объектов. В настоящее время средний фильтр и медианный фильтр широко используются для реализации фонового моделирования. Метод вычитания фона может предоставить наиболее полную информацию об объекте в случае, если фон известен.

Классификация объектов

На рис. 4 представлен кадр видеопотока и этот же кадр после его сегментации и выделения на нем двух классов – неба и земли. Классификация данного кадра была получена с помощью средств MATLAB и нейронной сети resnet18, специально дообученной на множестве видеоданных, записанных в ходе полевых испытаний. Нейронная сеть хорошо отделила на рис. 4, б небо и растительность.



а) исходный



б) сегментированный

Рис. 4. Кадр видеопотока с двумя классами: небо и растительность

Также произведена сегментация кадров видеопотока (рис. 5, 6), где имеется более насыщенный ландшафт земной поверхности. На рис. 4 было выделено два класса, на других – три.



а) исходный



б) сегментированный

Рис. 5. Кадр видеопотока с тремя классами: небо, растительность и строения



а) исходный



б) сегментированный

Рис. 6. Кадр видеопотока с тремя классами: небо, растительность и строения

Таким образом, были проведены эксперименты по сегментации кадров видеопотока с последующей классификацией выделенных зон. Голубым цветом выделено небо, оранжевым – деревья и растительность, желтым – различные строения.

Заключение

Реализован и рассмотрен метод семантической сегментации с последующей классификацией выделенных в отдельные классы зон земной поверхности по снятым кадрам видеопотока, сформированных аппаратурой ЛА. При этом была обучена нейронная сеть, классифицирующая несколько зада-

ваемых классов. В рамках эксперимента были использованы реальные видеоданные, полученные с аппаратуры ЛА. Таким образом, была показана возможность классификации зон кадра на основе использования семантической сегментации и обучений нейронной сети.

Библиографический список

- 1 Podoplekin Yu. F., Shepeta D. A., Nenashev V. A. Modeling of on-board radar input signals caused by reflections of the sounding signal from the underlying surfaces of the earth and the sea // *The Sea Messenger*. 2016. № 4. P. 69-71.
- 2 Rybkin S. V., Nesterov A. Yu. Approaches in the recognition of road signs // *Problems of the formation of a unified scientific space: Coll. of art. of the Intern. Scientific and Practical Conf.: In 4 parts*. 2017. P. 16–19.
- 3 Convolutional neural network, part 2: learning by error back propagation algorithm. URL: <https://habr.com/ru/post/348028/> (дата обращения: 05.04.2021).
- 4 Nenashev V. A., Sentsov A. A., Kuyumchev G. V. Modeling of the process of forming a high-resolution radar image in an on-board radar. 2013. Vol. 2, № 3. P. 48–56.
- 5 Features of information processing in the on-Board dip small radar on the basis of the UAV / V. A. Nenashev, A. F. Kryachko, A. P. Shepeta, D. A. Burylev // *SPIE Future Sensing Technologies, Tokyo*, 2019. P. 111970X-1–111970X-7.
- 6 Distributed matrix compression techniques, masking and error-correcting coding of images in high-speed network of information exchange, data processing and information aggregation / E. A. Kapranov, V. A. Nenashev, A. M. Sergeev [et al.] // *SPIE Future Sensing Technologies, Tokyo*, 2019. P. 111970T-1–111970T-7.

УДК 004.921, 004.928, 004.946

Б. С. Гайков

магистрант кафедры вычислительных систем и сетей

Н. Н. Решетникова – кандидат технических наук, доцент – научный руководитель

МОДЕЛИРОВАНИЕ ТРЕХМЕРНЫХ ОБЪЕКТОВ ДЛЯ ИНТЕРАКТИВНЫХ ПРИЛОЖЕНИЙ. ТЕХНОЛОГИЯ МОДЕЛИРОВАНИЯ ТВЕРДЫХ ПОВЕРХНОСТЕЙ

Моделирование твердых поверхностей (Hard Surface Modelling) – это особая специализация 3D-моделирования. Как следует из названия, такое моделирование сосредоточено на создании неорганических или «твердых» объектов, например сделанных руками человека.

Использование технологии твердотельного моделирования дает возможность создавать сложные и детализированные объекты за несколько минут. Детализация позволяет получить более качественную итоговую модель, которая будет вписываться в окружение игрового мира и будет приятна глазу для конечного пользователя (рис. 1).

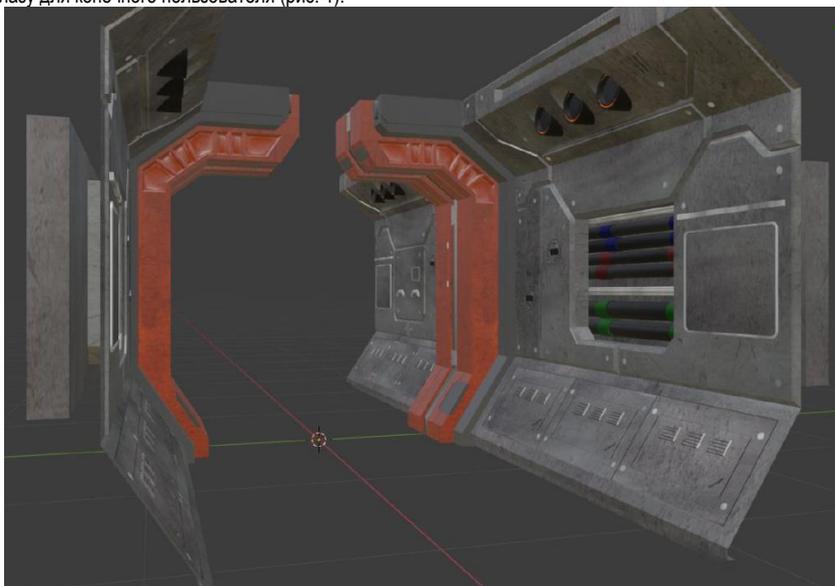


Рис. 1. Туннель космического корабля в стиле Sci-Fi

Индустрия твердотельного моделирования бурно развивается. Областей для его применения очень много: концепт-арты для фильмов, объекты для видеоигр, реальная разработка роботизированных устройств, 3D-печать и др.

Эта ниша моделирования требует специфических навыков. Точность, чистая топология, оптимизация – все это важные аспекты моделирования твердых поверхностей. В процессе создания модели, могут возникнуть следующие проблемы:

- треугольники и многоугольники (Tris и Ngon);
- булевы операции (Boolean);
- способы создания скошенных ребер;
- вырез отверстий в изогнутых поверхностях;

- проблемы с тенями (Shading);
- плохая топология.

Применение данной технологии в пакете Blender позволяет решить все без исключения проблемы и получить красивую и изящную модель.

При создании непосредственно самой модели достаточно использовать стандартные инструменты для разработки, предоставленные Blender. Однако это и не исключает использование сторонних модификаций (Addons), установленных дополнительно на программное обеспечение. Поскольку Blender является абсолютно бесплатной программой, его сообщество активно поддерживает и развивает идею модификаций. Их установка и последующее использование в процессе моделирования позволяют существенно облегчить работу дизайнеру и ускорить ее. На данный момент на рынке можно найти множество полезных инструментов для работы, как бесплатных, так и на коммерческой основе.

Процесс создания модели начинается с блокинга (рис. 2) [1] с использованием стандартных примитивов для получения желаемой формы и размера. По сути, это процесс применения ряда булевых операций – вычитание, объединение и пересечение, которые идут один за другим линейно и по порядку.



Рис. 2. Процесс блокинга, стек булевых операций

Поскольку в процессе создания базовой формы используются лишь примитивы, данная технология позволяет получать сложные геометрические трехмерные объекты, не нагружая компьютер и не используя вычислительную мощность (рис. 3), поскольку для расчета конечной модели нужны только начальные объекты и список операций, которые идут по порядку. Кроме того, операциями легко манипулировать – удалить, переставить местами, скрыть, дублировать [2].

Акцент при создании делается в первую очередь на дизайне, и если полигонов и точек становится слишком много, то впоследствии применяется ретопология [3].

В большинстве компьютерных игр присутствуют сильно детализированные 3D-объекты, созданные, чтобы удивлять реалистичностью и красотой. Но дизайнеры и создатели моделей в процессе проработки объектов не ограничивают себя строгими рамками и дают волю фантазии. Часто результатом их творчества становится проработанный до мельчайшей детали, узнаваемый или интересный

предмет либо персонаж, который просто невозможно перенести в игровой мир. Причина кроется в огромном количестве полигонов.

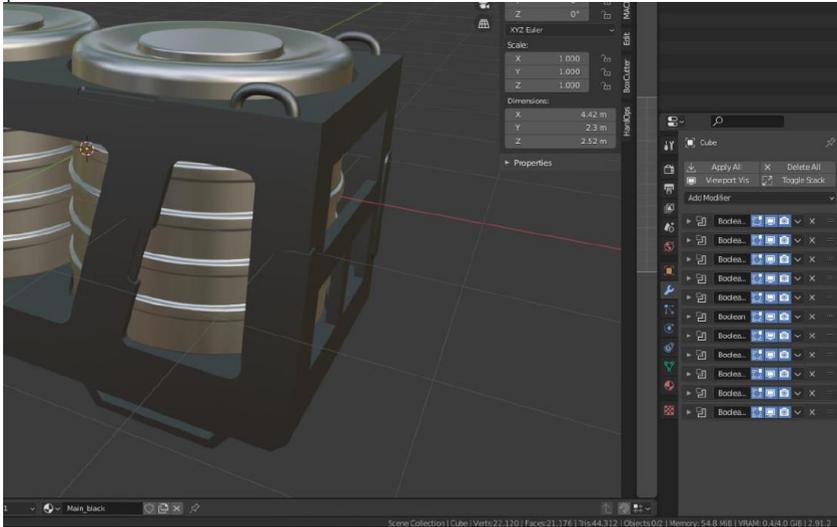


Рис. 3. Объем занимаемой памяти на количество полигонов

Моделирование – сложный процесс, вначале которого редко обращают внимание на то, как будут располагаться полигоны готового объекта. Поэтому за его созданием следует ретопология. Это общая часть рабочего процесса моделирования. В технологии моделирования твердых поверхностей объекты создаются с упором на форму и детализацию.

Часто топология объекта неэффективна: ребра не сходятся, имеются многоугольники, очень плотная сетка полигонов (полисетка) и т. д. Неэффективная топология занимает много памяти (из-за большого количества точек) или/и мешает дальнейшему моделированию. В таких случаях применяется ретопология – создание новой сетки полигонов с меньшим разрешением (рис. 4).

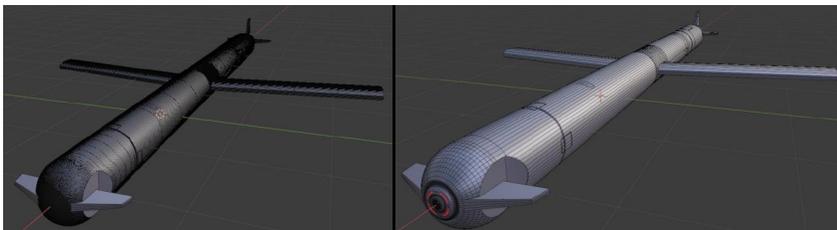


Рис. 4. Высокополигональная и низкополигональная модели

После переработки топологии гораздо проще развернуть объект на 2D-плоскость для создания текстуры (UV-развертки [4]). Пакет Blender включает редактор материалов, что позволяет создать процедурную текстуру [5] для дальнейшего запекания и экспорта в игровой мир. Любой объект с наложенными процедурными текстурами будет смотреться корректно, так как текстура генерируется внутри программы и накладывается прямо на объект (или часть объекта) (рис. 5).

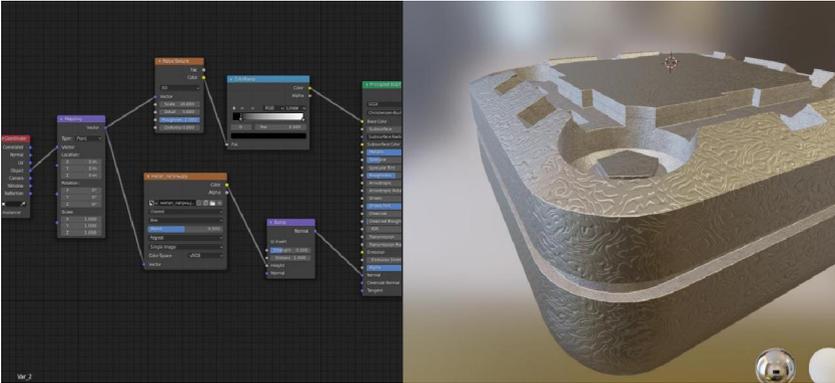


Рис. 5. Процедурные текстуры на объекте

В игровых движках, например Unity или Unreal, как и в Blender, имеется редактор материалов, что позволяет и в них создавать процедурные текстуры. Но делать это крайне не рекомендуется, поскольку их первоочередная цель – игра и игровая логика и тратить ресурсы на расчет материалов неэффективно. Вместо этого используется метод наложения материалов при помощи общей текстуры и карт нормалей на развертку модели и различных дополнительных карт, например карта освещения окружения (Ambient Occlusion map) или карта высот (Height map). Использование уже запеченных текстур и карт нормалей существенно снижает время на обработку модели в интерактивном приложении (рис. 6).

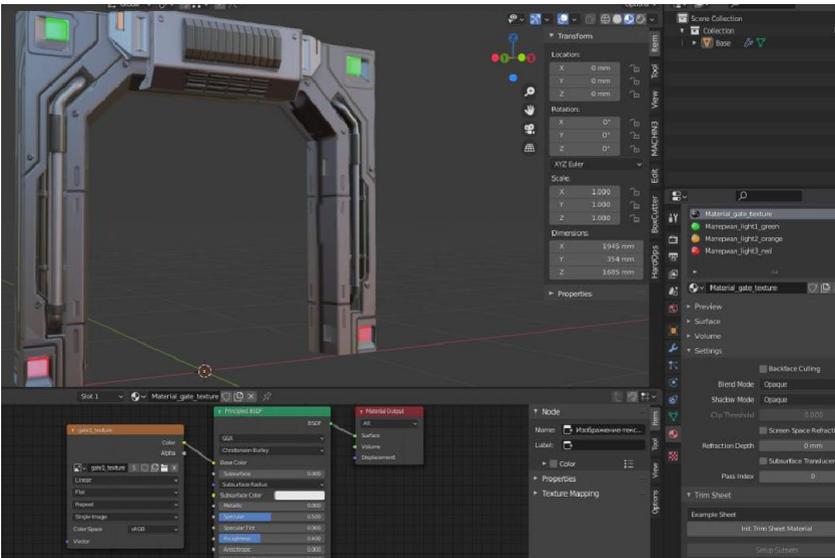


Рис. 6. Готовые текстуры на объекте

На данном этапе разработки уже можно приступить к созданию костей для последующей анимации объекта. Если анимация не требуется, то можно сразу экспортировать [6] в игровой движок.

Для использования модели в качестве концепт-арта [7] в программном обеспечении Blender можно сделать рендер анимации. Для этого следует настроить сцену: установить освещение, камеры, а для деталей объекта установить ключи – положения, вращения и масштабирования. Ключи также можно устанавливать и для параметров материалов, что позволяет изменять материал в процессе анимации.



Рис. 7. Рендер концептуальной модели с помощью Cycles

Создание объектов для интерактивных приложений по технологии моделирования твердых поверхностей существенно упрощает и ускоряет процесс работы для дизайнера. Конечный объект ничуть не уступает в визуальном уровне качества, при этом занимает гораздо меньше объема оперативной памяти, что существенно снижает нагрузку на процессор и видеокарту.

Библиографический список

1. Valsanen T. Hard-Surface моделирование и советы по материалам. 2019. URL: <https://3dpara.ru/hard-surface-modeling-material-tips/> (дата обращения: 16.02.2021).
2. Руководство пользователя Blender 3D. URL: https://programishka.ru/docs_manual/doc/blender/ (дата обращения: 18.02.2021).
3. LOOKINAIR, 2020. URL: <https://lookinair.com/ru/education-ru/chto-takoe-retopologyya/> (дата обращения: 19.02.2021).
4. Филенко Р. UV-развертка. 2015. URL: <https://blender.filenko.ru/visualization/uv-map.html> (дата обращения: 20.02.2021).
5. Корбут Д. Процедурные текстуры. 2016. URL: https://blender3d.org.ua/book/materials_and_textures/ (дата обращения: 21.02.2021).
6. Секреты экспорта из Blender в Unity. 2015. URL: <https://habr.com/ru/post/254937/> (дата обращения: 22.02.2021).
7. Слаква А. Hard-Surface моделирование в Blender. 2020. URL: <https://blender3d.com.ua/hard-surface-modeling-in-blender/> (дата обращения: 17.02.2021).

УДК 004.92

В. А. Карамышева

студент кафедры вычислительных систем и сетей

Н. Н. Решетникова – кандидат технических наук, доцент – научный руководитель

ПОСТРОЕНИЕ ВИРТУАЛЬНОГО 3D-ТУРА ПО УДАЛЕННЫМ ПОДРАЗДЕЛЕНИЯМ ПРЕДПРИЯТИЯ

Введение

В настоящее время технологии изготовления интерактивно-виртуальных проектов в виде презентаций с эффектом присутствия на основе панорамных фотографий 360° очень популярны и вошли в моду, стали хорошим тоном для современных компаний, учебных заведений, музеев, ресторанов, магазинов и др.

Виртуальный тур как мультимедийная технология, в зависимости от требований заказчика (руководителя предприятия, руководителя отдела по связям с общественностью и т.д.) и поставленных целей (например, рекламы или обучения), может объединять не только 3D-панорамы, но и звуковую дорожку, текстовую информацию, видео, переходы по ссылкам к другим объектам.

Современные программные решения для создания виртуальных туров позволяют не только собирать отснятые панорамы в тур, но и наполнять их интерактивностью при помощи встроенных решений для наилучшего вовлечения конечного пользователя в процесс получения информации благодаря изменению восприятия.

Одна из проблем при создании виртуального тура, которая стоит перед разработчиком, расстановка сферических панорам и наполнение тура интерактивностью таким образом, чтобы навигация по нему была не только интуитивно понятной для пользователя, но и соответствовала желаниям заказчика, так как с ростом числа отдельных панорам в виртуальном туре навигация по нему становится сложнее, и приходится тратить много времени для поиска необходимой информации. Улучшить навигацию по туру и выбрать подходящий под предпочтения заказчика маршрут предлагается путем внесения изменений в процесс создания виртуального тура, используя метод взвешенных графов с наиболее значимыми показателями.

Способы навигации по виртуальному туру

Виртуальный тур создает впечатление настоящего присутствия внутри объекта и позволяет пользователю самому решать, как составить свой маршрут, рассмотреть все, что его интересует, приблизившись к определенным предметам, или получить большой обзор, удаляясь при помощи визуальных кнопок. Можно сказать, что виртуальный тур – это набор переходов между отдельными 3D-панорамами. Маршрут виртуального тура представляет собой наиболее удобный путь следования, способствующий раскрытию темы. Он строится в зависимости от оптимальной последовательности осмотра объектов. Одна из задач маршрута – способствовать получению наиболее полного объема необходимой информации.

Рассмотрим три основных способа навигации по виртуальному туру.

1. Расстановка точек переходов непосредственно на 3D-панорамах (примером такого отображения может служить тур по СПбГУТ [1]).

Этот способ прост в реализации; точки переходов на отдельных 3D-панорамах расставляются стандартными средствами. Но при перемещении по туру возникают следующие проблемы:

– пользователь не видит границ виртуального тура, ему сложно понять, сколько еще необходимо времени, чтобы закончить осмотр объекта отображения;

– план виртуального тура отсутствует, поэтому пользователю приходится запоминать все переходы, а значит, возврат в точку начала обзора для пользователя может быть проблематичен;

– проблемы, указанные в предыдущих пунктах, приводят к тому, что пользователь теряется в виртуальном туре.

2. Навигация осуществляется при помощи «статической карты» (примером проведения тура по карте может быть виртуальный тур по Оксфордскому университету [2]). Для нее справедливы следующие утверждения:

- карта представляет собой план объекта отображения и набор точек переходов;
- для всех посетителей карта выглядит одинаково;
- изменять маршрут (набор точек переходов) может только разработчик тура.

Основное преимущество статических карт для разработчика – простота реализации, для пользователя важно, что ему понятны размер и план виртуального тура, в отличие от предыдущего метода. Но с повышением числа 3D-панорам и увеличением размера карты возникают следующие проблемы:

- тур становится недостаточно понятным;
- ухудшается ориентация по туру при выборе необходимой информации;
- пользователю следует просмотреть все точки перехода, что занимает много времени;
- происходит наложение точек перехода в зависимости от размера карты;
- уменьшается информативность тура, под которой будем понимать получение наиболее полной информации об объекте при наименьших затратах времени.

3. Комбинированный способ – сочетание первого и второго. Визуально пользователь может, путешествуя по туру, определить, где находится в данный момент. Данный способ был применен при создании проекта виртуального тура по ГУАП, где каждой группе панорам соответствует план здания, на котором расположены анимированные активные точки перехода с мест съемки панорам [3].

Способ навигации по виртуальному туру закладывается разработчиком тура при его создании. Поэтому для изменения процесса навигации по туру необходимо изменить процесс его создания.

Процесс построения виртуального тура

Процесс создания виртуальных туров складывается из нескольких этапов:

- 1) сбор информации об объекте, формирование сценария видео, уточнение требований заказчика;
- 2) проведение профессиональной фото- и видеосъемки;
- 3) обработка и ретушь панорамных снимков, монтаж видео;
- 4) создание виртуального тура: дизайн оболочки и внутренней среды, наполнение тура дополнительными элементами.

На первом этапе осуществляется сбор требований и информации. Объектом проведения фото- и видеосъемки для проекта виртуального тура стало ООО «ОМ3-ИТ». В процессе ознакомительной экскурсии по предприятию в рамках вводного инструктажа основанием будет инструкция по проведению вводного инструктажа, а наполнение тура интерактивными эффектами будет производиться с учетом специфики деятельности предприятия, которые определены уставом и иными нормативными документами. Дополнительно в тур добавлен проект видео 360°, для монтажа видео необходимо заранее сформировать и согласовать с заказчиком сценарий.

Определившись с объектами для проведения профессиональной видео- и фотосъемки, необходимо подобрать оборудование. Для выполнения поставленных задач наиболее целесообразно было использовать камеру Ricoh Theta S. Она оснащена двумя широкоугольными объективами с углом обзора 200°. После съемки получаются два изображения, которые программно конвертируются в бесконечную панораму, которую можно просматривать на компьютере, смартфоне или в очках виртуальной реальности. Данная камера имеется в наличии на кафедре № 44.

В камере Theta S установлен сенсор разрешением 12 мегапикселей, позволяющий снимать видео FullHD (30 к/с) или фото в разрешении 5 376 × 2 688 пикселей [4]. Недостаток данной камеры в том, что в ней есть только встроенные 8 Гб памяти, слота для дополнительной карты памяти нет. Но данный минус для проекта не критичен.

После того как все объекты сняты, переходим к третьему этапу работ и при помощи программных средств подготавливаем панорамные фотоснимки. Для редактирования панорам использовался программный продукт GIMP – свободный графический редактор, который поддерживает больше тридцати форматов изображений, умеет работать со слоями, масками, фильтрами и режимами смеси-

вания. В арсенале программы есть огромный спектр инструментов для цветокоррекции и обработки любых фотографий и изображений [5].

Монтаж видео 360° осуществляется при помощи программы Movavi 360 VideoEditor [6], просмотр возможен с видеохостинга YouTube.

Дальнейшее изготовление проекта виртуального тура по подразделениям предприятия, а также формирование дизайна обложки и внутренней среды, наполнение тура дополнительными элементами будет производиться в программе KolorPanotourPro (рис. 1) [7].

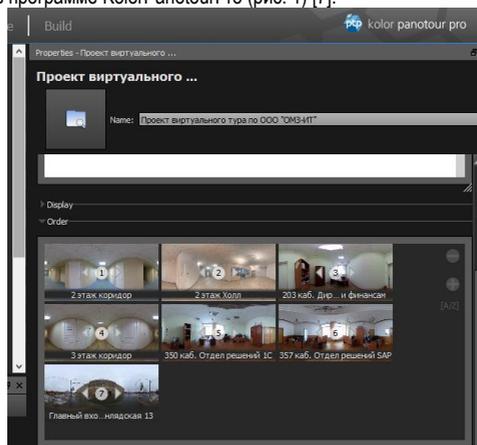


Рис. 1. Загруженные 3D-панорамы в программе KolorPanotourPro (версия 2.5)

Внесем изменения в процесс создания виртуального тура, дополним его, чтобы устранить упомянутые проблемы навигации. Теперь тур включает следующие подпроцессы:

1) «загрузка 3D-панорам» – в данном подпроцессе при помощи программного средства RICOH-THETA созданные 3D-панорамы загружаются в программу KolorPanotourPro и создается новый проект.

2) «формирование показателей маршрута» – исходя из предметной области тура, определенной заказчиком, разработчик выделяет набор показателей, которые будут использоваться заказчиком при построении маршрута;

3) «разработка шаблона» – разработчик загружает в KolorPanotourPro набор файлов (фото, документы, текстовую информацию, звуковые дорожки и др.), предоставленных заказчиком в рамках предметной области;

4) «заполнение значений показателей» – для каждой панорамы разработчик тура заполняет значения набора показателей;

5) «формирование анкеты для оценки наполненности тура в соответствии с показателями заказчика» – критик (в нашем случае заказчик) оставляет оценку о просмотренном туре, заполняя анкету, содержащую набор вопросов, по которым оценивается удовлетворенность;

6) «построение схемы тура» – разработчик строит графпереходы между панорамами;

7) «построение маршрута» – на сформированном графе строится маршрут с учетом значений показателей анкет.

Также предложены изменения процесса просмотра виртуального тура, результат которого представлен на рис. 2. Процесс содержит два новых подпроцесса – «Заполнение критериев поиска» и «Заполнение анкеты». В первом пользователь (в нашем случае заказчик) через устройство ввода информации смотрит интересующие его характеристики тура – предпочтения заказчика. Данные предпочтения служат основой при генерации переходов в виртуальном туре. Во втором подпроцессе пользователь по окончании просмотра виртуального тура заполняет анкету с оценкой.



Рис. 2. UML-диаграмма деятельности, отображающая процесс просмотра виртуального тура

Модель виртуального тура с переходами

Объектом моделирования является проект виртуального тура по предприятию, который состоит из множества отдельных 3D-панорам и активных точек-переходов между ними.

Представим проект виртуального тура по предприятию в виде кортежа [8]:

$$T = \langle G, P, W, Q, Z \rangle,$$

где $G = (V, E)$ – связной неориентированный граф;

$V = \{v_1, v_2, \dots, v_n\}$ – множество вершин графа, которым соответствуют 3D-панорамы в виртуальном туре;

$E = \{e_1, e_2, \dots, e_m\}$ – множество ребер в графе, которые соответствуют возможным переходам от одной панорамы к другой;

$P = \{p^1, p^2, \dots, p^k\}$ – показатели анкеты для оценки вершин графа, где каждый показатель имеет наименование и целочисленное значение в порядковой шкале от 0 до 5;

$W = \{w_1, w_2, \dots, w_k\}$ – весовые коэффициенты показателей ($0 \leq w_i \leq 1$ и сумма всех коэффициентов равна 1);

$Q = \{Q_1, Q_2, \dots, Q_n\}$ – оценки вершин графа G относительно показателей P ,

$$Q_i = \{p^1, p^2, \dots, p^k\},$$

где i – номер вершины графа G ;

$Z = \{z^1, z^2, \dots, z^k\}$ – предпочтения заказчика для просмотра тура (анкета для формирования маршрута).

Для каждой вершины i графа G определим степень близости d_i к предпочтениям заказчика Z для просмотра тура и назовем весом вершины i :

$$d_i = \sqrt{\sum_{j=1}^k w_j (p_j^i - z^j)^2}.$$

Полагаем, что вершина i предпочтительнее вершины j , если $d_i < d_j$.

Отметим, что компоненты G, P, W, Q строятся экспертом – разработчиком тура, а компонента Z задается заказчиком (например, руководителем предприятия, руководителем отдела по охране труда, специалиста по кадрам или председателем профсоюзного органа, в зависимости от того, на кого приказом генерального директора возлагаются функции по охране труда на предприятии).

Рассмотрим пример модели построения проекта виртуального тура по ООО «ОМЗ-ИТ».

Граф $G = (V, E)$ представлен на рис. 3, а. Наименования показателей P анкеты для оценки вершин графа приведены в колонке 1 табл. 1. Шкала значений показателей P приведена в табл. 2. Весовые коэффициенты показателей: $w_j = 0,1$ для всех $j = 1, \dots, k$. Оценки Q вершин графа G относительно показателей P приведены в колонке 2 табл. 1. Предпочтения заказчика Z для просмотра тура приведены в колонке 3 табл. 1.

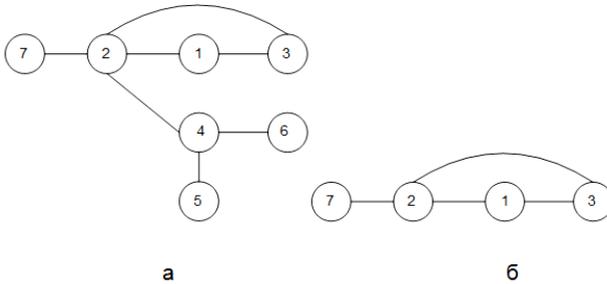


Рис. 3. Графы проекта виртуального тура по предприятию
 а – граф G виртуального тура T ; б – граф G_0 виртуального тура T_0

Таблица 1

Значение показателей P на вершинах графа G

Показатель	Оценки вершин графа							Предпочтения заказчика
	1	2	3	4	5	6	7	
1. Общая информация об организации	3	3	2	1	1	1	4	5
3. Информация о структурных подразделениях общества	1	0	1	3	1	1	0	3
4. Правила внутреннего распорядка	0	0	0	3	3	3	0	3
5. Информация о руководстве	0	0	0	3	1	1	0	3
6. Информация о безопасных условиях труда (инструкции)	0	0	0	3	3	3	0	3
7. Информация о технике и технических изделиях	0	0	0	3	3	3	0	3
8. План эвакуации	0	3	0	0	0	0	0	3
9. Наличие интерактивных эффектов на панораме	3	3	3	3	3	3	3	3

Шкала значений показателей P

Значение	Наименование
0	Отсутствие информации
1	Небольшое наполнение информацией
2	Наполнение информацией ниже среднего
3	Среднее наполнение информацией
4	Наполнение информацией выше среднего
5	Большое наполнение информацией

Для данного примера значения весов вершин равны: $d_1 = 1,51$; $d_2 = 0,76$; $d_3 = 1,89$; $d_4 = 1,51$; $d_5 = 2,07$; $d_6 = 2,07$; $d_7 = 0,26$.

Постановка задачи

Пусть задан виртуальный тур $T = \langle G, P, W, Q, Z \rangle$.

Выделим вес d_i для каждой вершины графа.

Для формулировки задачи построения нового тура зададим число h – количество вершин в графе G_0 . Полагаем, что $h < n$.

Необходимо найти новый тур $T_0 = \langle G_0, P, W, Q, Z \rangle$, в котором граф G_0 содержит h наиболее предпочтительных вершин графа G , а ребра графа G_0 строятся по правилам ниже приведенного алгоритма.

Назовем эту задачу задачей построения виртуального тура, с критерием улучшения – подмножество $V_0 \subset V$ такое, $|V_0| = h$.

Алгоритм построения виртуального тура

Алгоритм построения улучшенного виртуального тура T_0 выглядит следующим образом.

Шаг 1. Вершины упорядочиваются по возрастанию веса d_j .

Шаг 2. В V_0 заносятся первые h вершин из V .

Шаг 3. Строится связный граф G_0 по правилам: если ребро $(i, j) \in E$ и $i, j \in V_0$, то оно входит в E_0 ; иначе ищем путь в графе G из i в j , в котором только i и j принадлежат V_0 (такой путь существует, так как граф G связный) и строим ребро (i, j) в графе G_0 ; если таких путей несколько, то рассматриваем только один (строим только одно ребро $(i, j) \in E_0$).

Результат работы алгоритма для проекта виртуального тура с параметрами, рассмотренными выше, и $h = 4$ приведен на рис. 3, б. Карта связей проекта виртуального тура по подразделениям предприятия с топологией ссылок согласно алгоритму в программе KolorPanotourPro приведена на рис. 4.

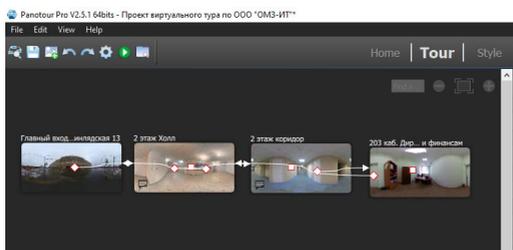


Рис. 4. Карта связей проекта виртуального тура в программе KolorPanotourPro

Выводы

Были рассмотрены теоретические и технологические аспекты создания виртуальных туров и проблемы навигации по ним. Предложен процесс построения виртуального тура по предприятию, устраняющий проблемы навигации с учетом значений показателей анкет. Представлен алгоритм генерации маршрута по предприятию по загруженным файлам шаблонов отдельных 3D-панорам.

В дальнейшем планируется использовать данный алгоритм построения виртуального тура при разработке интерактивного прототипа web-приложения с использованием 3D-панорам и видео 360 для виртуального тура по подразделениям предприятия в рамках вводного инструктажа.

Библиографический список

1. 3D-тур по университету СПбГУТ. URL:<https://www.sut.ru/univer/3d-tur>(дата обращения: 19.03.2021).
2. Виртуальный тур по Оксфордскому университету. URL: <http://www.chem.ox.ac.uk/oxfordtour/default.asp> (дата обращения: 19.03.2021).
3. Проект виртуального тура по ГУАП. URL: <http://guapvrtour.ru> (дата обращения: 19.03.2021).
4. RICON THETA: офиц. сайт. URL: <https://theta360.com/ru/about/theta/s.html> (дата обращения: 19.03.2021).
5. GIMP: офиц. сайт. URL: <https://www.gimp.org> (дата обращения: 19.03.2021).
6. Руководство пользователя видеоредактора MOVAVI 360. URL: https://img.movavi.com/movavi.com.12/affiliate/docs/360VE_RU.pdf (дата обращения: 19.03.2021).
7. Сайт центра поддержки GoPro/Справочные статьи/Учебники Panotour. URL: <https://community.gopro.com/t5/en/Panotour-Tutorials/ta-p/473490#> (дата обращения: 19.03.2021).
8. Рашевский Н. М., Дворянкин А. М. Разработка процесса построения и модели виртуального тура с динамическими переходами // Изв. ВолгГТУ. Серия «Актуальные проблемы управления, вычислительной техники и информатики в технических системах». Вып. 20: межвуз. сб. науч. ст. Волгоград, 2014. № 6 (133). С. 61–68.

УДК 004.92

Л. В. Лемешевская

студент кафедры вычислительных систем и сетей

Н. Н. Решетникова – кандидат технических наук, доцент – научный руководитель

ТРЕХМЕРНОЕ МОДЕЛИРОВАНИЕ КАК ОДИН ИЗ КОМПОНЕНТОВ ПОГРУЖЕНИЯ В ТЕМАТИКУ КВЕСТА

Ежедневно человек сталкивается с множеством графических образов, создаваемых посредством компьютера. На протяжении долгого времени основными устройствами для вывода трехмерных изображений в дополненной реальности выступали компьютер и различные пользовательские консоли. Сейчас же в качестве устройства все чаще используется смартфон, что обусловлено наличием хорошего разрешения экрана, оснащенностью устройства всевозможными датчиками (например, определения пространства) и высокой производительностью, позволяющей воспроизводить большинство трехмерного контента.

Дополненная реальность (англ. augmented reality) – технология, помогающая существенно расширить область данных воспринимаемых человеком. Черты, присущие ей:

- распознаваемость (возможность идентификации присутствия этой формы реальности);
- сочетаемость (взаимодействие и синтез с объективной реальностью);
- искусственность (наличие факта существования обуславливается компьютерными технологиями).

Если рассматривать актуальность использования дополненной реальности, то можно выделить несколько значимых пунктов:

- интерактивность (возможность взаимодействия);
- доступность в реальном времени (быстрая обработка данных и отображение);
- реалистичность (возможность создания объектов, максимально схожих с объективной реальностью);
- инновационность (внедрение и применение в различных сферах жизни впервые);
- «вау»-эффект (неординарность в представлении любой информации).

Общепринято выделяют пять основных перцептивных ресурсов человека – слух, обоняние, вкус, зрение и осязание. Рассмотрим влияние приложений с дополненной реальностью в контексте использования зрения.

На данный момент не существует полных и точных исследований в данной области. Можно проанализировать отзывы пользователей приложения с дополненной реальностью «Доброград» [1], которое было представлено на ПМЭФ в 2016 г. Систематизировав информацию, пользователи ощутили следующие основные эффекты:

- высокий эмоциональный фон (опыт и знания, полученные в процессе использования приложения, были подкреплены положительными эмоциями благодаря легкости использования и новизне);
- высокий уровень восприятия (легкость в запоминании тех или иных технических характеристик проекта, возможность их описания после использования приложения);
- интерактивность (наличие диалога пользователя с приложением, возможность изменять угол обзора, масштаб объектов и др.);
- вовлеченность (интерес к объектам приложения, быстрая концентрация внимания).

Всю приведенную информацию можно использовать для повышения уровня погружения в приложение. Погружение (иммерсивность) – свойство моделируемой среды, вовлекающее человека через его сенсорные возможности в поведение среды и обеспечивающее его непрерывным потоком стимулов [2].

Рассмотрим конкретный пример – создание тематического приложения-квеста с дополненной реальностью. Пользователю предлагается выполнять задания на экране смартфона, чтобы получить доступ к местоположению и виду новых меток. Основной темой для сценария приложения выступает космос, а метками – изображения, выполненные в бренд-стиле ГУАП [3]. В качестве среды разработки выбран игровой движок Unity с использованием плагина Vuforia [4].

На рис. 1 можно увидеть, как, в соответствии со сценарием, отбираются подходящие изображения, которые загружаются на сайт Vuforia для будущей работы.

<input type="checkbox"/>	Target Name	Type	Rating	Status
<input type="checkbox"/>	pic20	Single Image	★★★★★	Active
<input type="checkbox"/>	pic15	Single Image	★★★★★	Active
<input type="checkbox"/>	pic14	Single Image	★★★★★	Active
<input type="checkbox"/>	pic13	Single Image	★★★★★	Active
<input type="checkbox"/>	pic12	Single Image	★★★★★	Active

Рис. 1. Загруженные метки для будущего приложения

Для повышения иммерсивности двухмерные маркеры – изображения из объективной реальности будут дублироваться трехмерными анимированными моделями, но уже в дополненной реальности.

Для создания трехмерных моделей существует множество 3D-редакторов, был выбран Blender [5]. На рис. 2 показана одна из стадий создания 3D-модели ракеты по референсу. Окончательный вариант ее моделирования представлен на рис. 3.

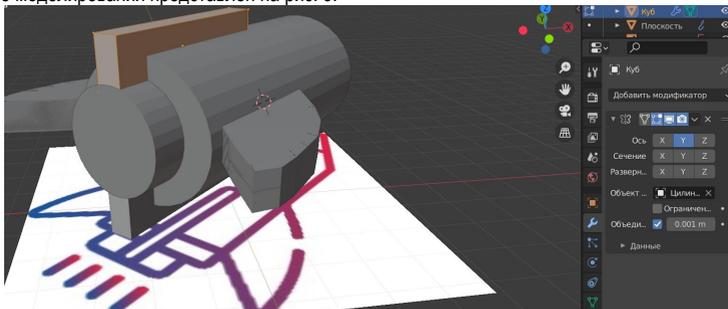


Рис. 2. Процесс 3D-моделирования ракеты по референсу

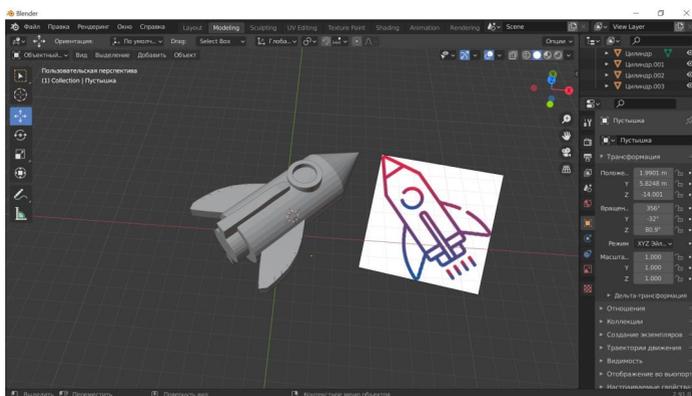


Рис. 3. Конечный вариант 3D-модели ракеты в сравнении с исходным изображением маркера

Далее необходимо задать анимации и текстуры для 3D-моделей. Это можно делать как в редакторе Blender, так и в игровом движке Unity. Потом в Unity происходит создание сцен, исходя из сценария, и интеграция моделей в приложение. На рис. 4 продемонстрирован тестовый вид продукта, который получает пользователь на экране своего смартфона.

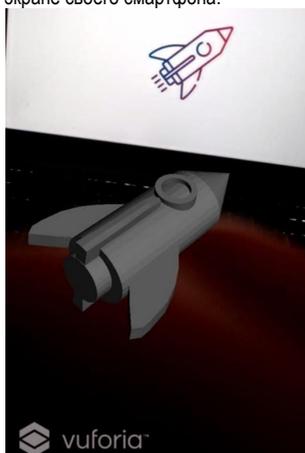


Рис. 4. Пробное отображение взаимодействия маркер – приложение дополненной реальности

Основное преимущество рассмотренного подхода – взаимосвязь компонентов из объективной реальности (меток-изображений) с дополненной реальностью (3D-модели на экране смартфона), что приводит к повышению уровня погружения как в тематику квеста, так и в приложение в целом. Справедливо будет отметить, что технология дополненной реальности одна из самых эффективных технологий смешанной реальности за счет максимальной связи с окружающим миром.

Библиографический список

1. EligoVision – интерактивные технологии. URL: <https://eligovision.ru/ru/project/105/> (дата обращения: 13.05.2021).
2. Никитин А. В., Решетникова Н. Н., Ситников И. А. Цифровые реальности: основные понятия и применения: учеб. пособие. СПб.: ГУАП, 2020. 109 с.
3. Фирменный стиль – ГУАП. URL: <https://new.guar.ru/brand> (дата обращения: 13.05.2021).
4. Unity Documentation. URL: <https://docs.unity3d.com/Manual/index.html> (дата обращения: 13.05.2021).
5. Blender: офиц. сайт. URL: <https://docs.blender.org/manual/en/latest/editors/index.html> (дата обращения: 13.05.2021).

УДК 004.946

Т. Р. Мустафин

магистрант кафедры вычислительных систем и сетей

А. В. Никитин – кандидат технических наук, доцент – научный руководитель

ОБСЛУЖИВАНИЕ АВТОМОБИЛЕЙ НА ОСНОВЕ ЖЕСТОВОГО ИНТЕРФЕЙСА ШЛЕМА ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

Обслуживание автомобилей в реальном мире – достаточно длительный, сложный и трудоемкий процесс, состоящий из множества действий. Возможно ли симулировать его в мир виртуальной реальности с управлением с помощью жестов рук?

Актуальность темы обусловлена разработкой приложения для системы виртуальной реальности Oculus Quest 2 [1], так как система VR активно развивается и уже во многом превосходит конкурентов, таких как HTC Vive Cosmos [2].

Для реализации прототипа было принято решение использовать модель автомобиля Renault Duster и упрощенную модель двигателя, состоящую из поддона картера, коленчатого вала, маховика, четырех поршней, блока цилиндров, головки блока цилиндров, газораспределительных валов и их шестеренок, крышки головки блока цилиндров.

Oculus Quest 2 – это полностью автономные очки виртуальной реальности с шестью степенями свободы. Снабжается памятью в размере 64 или 256 Гбайт. По сравнению с предыдущей версией, он получил меньшую массу, улучшенные системы отслеживания, обновленные контроллеры и новый дизайн. Хотя ему и не нужен компьютер для работы, подключить шлем можно с помощью кабеля Oculus Link или беспроводным способом посредством встроенной технологии Air Link. Гарнитура имеет четыре камеры, с помощью которых ориентируется в пространстве, отслеживает положение контроллеров, а также жесты рук, которые поддерживает «из коробки». Перейдем к созданию сцены.

Создадим новый проект Unity [3] и загрузим библиотеку Oculus Integration [4] из Asset Store (рис. 1).

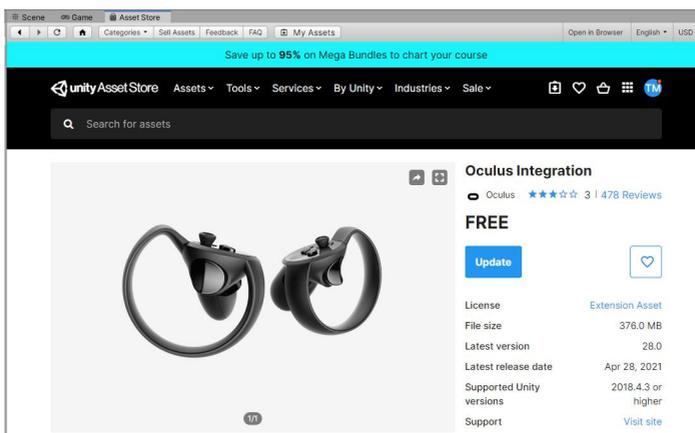


Рис. 1. Библиотека Oculus Integration

После установки библиотеки перейдем к построению сцены. Для начала добавим Plane. Он будет служить основанием для всего. Для того чтобы ограничить игровое пространство и не упасть за карту, добавим дополнительные Box Collider по краям основания (рис. 2).

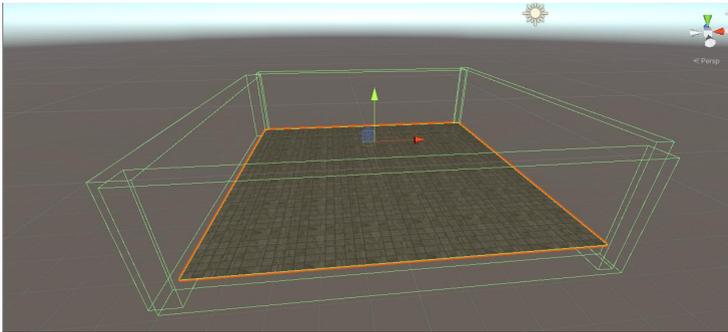


Рис. 2. Plane и Box Collider по краям

Заранее в редакторе Blender была подготовлена модель автомобиля. Внесем ее в сцену. Чтобы автомобиль не проваливался сквозь текстуры, добавим ему Mesh Collider. Включим недостающие детали: капот и дверь водителя и также наложим на них также mesh collider (рис. 3).

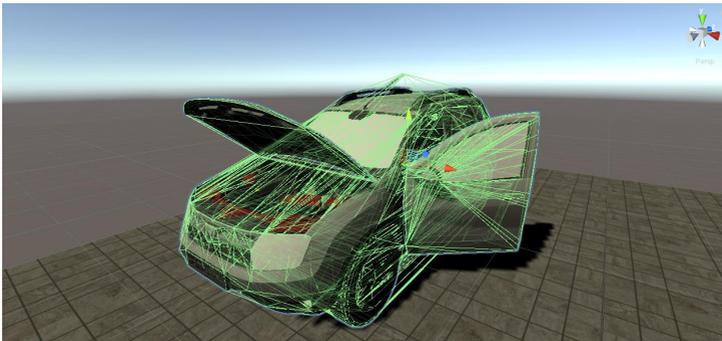


Рис. 3. Установка на сцену модели автомобиля

Сделаем на сцене стол, разместим на нем детали двигателя и добавим им компонент mesh collider (рис. 4).

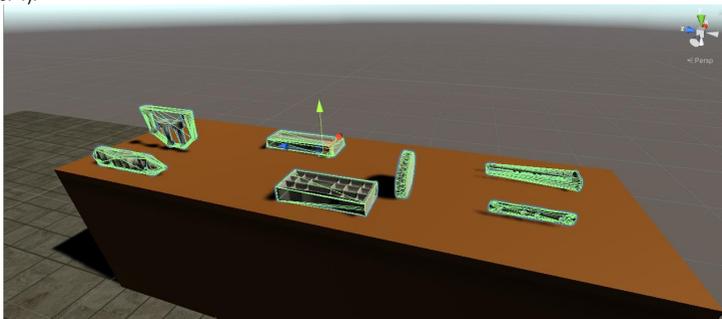


Рис. 4. Стол

Теперь, когда необходимые объекты на сцене, перейдем к добавлению контроллера персонажа Oculus [5]. Выложим в иерархию сцены префаб OVRPlayerController. В нем необходимо настроить OVRCameraRig. В пункты RightHandAnchor и LeftHandAnchor добавим префабы OVRHandPrefab. Он содержит все необходимые настройки и скрипты. Необходимо только для правой руки в разделах OVR Hand, Skeleton и Mesh установить значение Hand Right. В настройках OVRCameraRig в разделе Tracking необходимо заменить Tracking Origin Type на Floor Level (рис. 5).



Рис. 5. OVR Player Controller

Чтобы можно было взаимодействовать с деталями двигателя, им нужно добавить следующие компоненты: Rigid Body, OVR Grabbable. Чтобы крепить детали автоматически, добавим на них скрипт, который будет это делать (рис. 6).

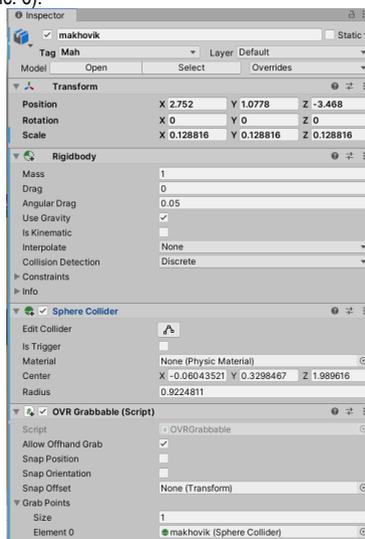


Рис. 6. Компоненты маховика

Чтобы руки могли взаимодействовать с предметами, добавим для каждой компонент Rigid Body, Sphere Collider и скрипт, который будет обрабатывать жест щипка для ухвата деталей:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using OculusSampleFramework;

public class HandGrabbingBehaviour : OVRGrabber
{
    private OVRHand m_hand;
    private float pinchThreshold = 0.7f;
    protected override void Start()
    {
        base.Start();
        m_hand = GetComponent<OVRHand>();
    }

    public override void Update()
    {
        base.Update();
        CheckIndexPinch();
    }

    void CheckIndexPinch()
    {
        float pinchStrength = m_hand.GetFingerPinchStrength(OVRHand.HandFinger.Index);

        if(!m_grabbedObj && pinchStrength > pinchThreshold && m_grabCandidates.Count > 0)
        {
            GrabBegin();
        }
        else if(m_grabbedObj && !(pinchStrength > pinchThreshold))
        {
            GrabEnd();
        }
    }
}
```

Для компонента Rigid Body нужно установить галочки в пунктах Use Gravity и Is Kinematic. У Sphere Collider надо поставить галку Is Trigger. В настройках нашего скрипта нужно задать галку Parent Held Object и префаб руки перенести в Grip Transform (рис. 7).

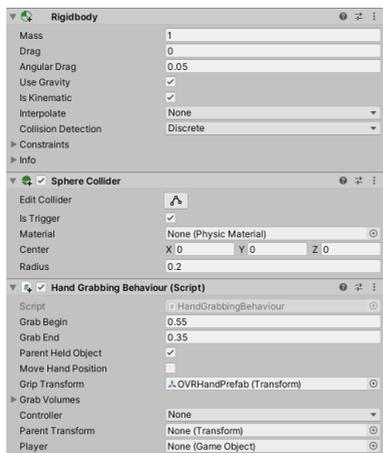


Рис. 7. Добавленные компоненты OVRHandPrefab

Проверим работоспособность (рис. 8–9).

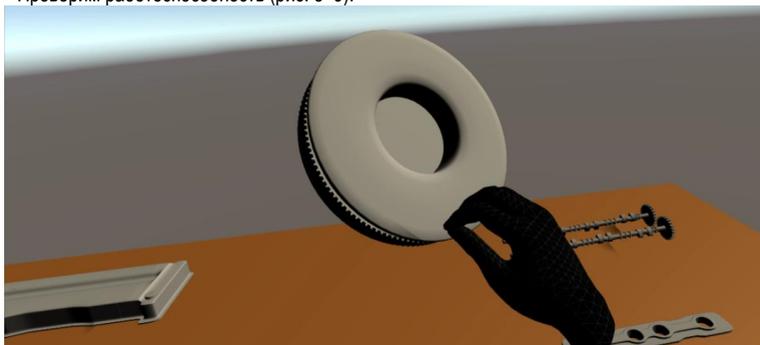


Рис. 8. Взятие детали жестом



Рис. 9. Прикрепление детали

Процесс обслуживания можно симулировать в виртуальном мире, но для этого нужно много времени. Система отслеживания жестов рук достаточно удобная, но все-таки полностью заменить контроллеры пока не может, при этом имеет достаточный потенциал для развития, такой как улучшение распознавания на сложных фонах при слабом освещении, а также увеличение количества распознаваемых жестов, что повышает функционал данной системы.

Библиографический список

1. Oculus Quest 2: сайт. URL: https://www.oculus.com/quest-2/?locale=ru_RU (дата обращения: 15.03.2021).
2. HTC Vive Cosmos: сайт. URL: <https://www.vive.com/ru/product/vive-cosmos/features/> (дата обращения: 15.03.2021).
3. Unity: сайт. URL: <https://unity.com/ru> (дата обращения 20.03.2021).
4. Oculus Integration: сайт. URL: <https://assetstore.unity.com/packages/tools/integration/oculus-integration-82022> (дата обращения: 21.03.2021).
5. Hand Tracking in Unity: сайт. URL: <https://developer.oculus.com/documentation/unity/unity-handtracking/> (дата обращения: 03.04.2021).

УДК 004.058:004.93

М. Э. Нагорных, А. Н. Быков

магистранты кафедры вычислительных систем и сетей

Н. В. Кучин – кандидат технических наук, доцент – научный руководитель

ОСОБЕННОСТИ И АНАЛИЗ ГРАФОВЫХ БАЗ ДАННЫХ

Введение

В последнее время появляются различные подходы к формированию баз данных и их структур. Усложняются требования к базам данных и их возможностям. Разработчики все больше отдают предпочтение нереляционным базам данных.

В некоторых проектах используется не столь известные графовые базы данных. Они активно развиваются и набирают большую популярность. Именно поэтому цель статьи – анализ графовых баз данных и рассмотрение их особенностей.

Графовые базы данных

Графовые базы данных – это базы данных, использующие в процессе работы различные графовые системы. Граф – абстрактный математический объект, представляющий собой множество вершин (точек) и набор ребер (линий), т. е. соединений между парами вершин. Узлы соответствуют объектам базы данных, а ребра – связям между этими объектами [1]. Схематичное изображение графовой структуры представлено на рис. 1.

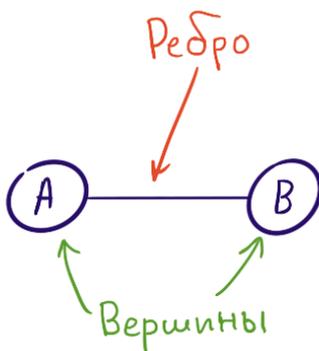


Рис. 1. Схема графовой структуры

Модель хранения информации в виде графов, графов со свойствами в узлах и гиперграфов сложилась в 1990–2000 гг. Первая графовая СУБД Neo4j создана в 2007 г. В наше время существуют десятки других графовых СУБД. Графовые базы данных применяются в областях, где информация о взаимосвязанности данных более важна, чем сами данные. Например, в логистике, сфере торговли, социальных сетях и географических информационных системах.

В проектах, где применяются, графовые базы данных, информация и отношения между данными обычно находятся на одном уровне. Использование графов в качестве инструмента моделирования дает несколько преимуществ для этого типа данных [2].

Особенности графовых баз данных

Каждая графовая система управления базами данных имеет ряд особенностей, однако есть общие черты, повышающие продуктивность их использования.

- Графовые базы данных позволяют более естественно моделировать данные. Структуры графов видны пользователю и позволяют естественным образом обрабатывать данные приложений, например гипертекстовые или географические данные. Графы имеют преимущество: могут хранить всю информацию об объекте в одном узле и отображать связанную информацию в виде ребер, связанных с ним.

- Запросы могут ссылаться непосредственно на структуру графа. С ними связаны такие операции, как поиск кратчайших путей, нахождение определенных подграфов и т. д.
- Графы и операции с ними позволяют пользователям выражать запрос на высоком уровне абстракции.
- Для выражения значимых запросов необязательно полное знание структуры.
- Удобный просмотр структуры взаимосвязей.

Обзор графовых систем управления базами данных

В настоящее время существует множество графовых СУБД. Рассмотрим некоторые.

ArangoDB – многоцелевая открытая СУБД, позволяющая хранить документы, графы и различные данные в виде ключ-значения. Эта СУБД использует язык запросов AQL, который по структуре близок к привычному SQL, благодаря расширению работа может вестись на языке JavaScript [3].

Средства хранения данных отвечают требованиям атомарности, согласованности, изолированности и надежности, т. е. ACID-свойствам. Имеется поддержка транзакций и масштабируемости базы данных, возможна как вертикальная, так и горизонтальная масштабируемость. Работа с СУБД осуществляется как через консольный клиент ArangoSH, так и через веб-интерфейс.

HyperGraphDB – встраиваемая система с открытым исходным кодом. Обладает очень легким и понятным интерфейсом, имеет возможность удобного подключения дополнительных модулей. Система разрабатывалась для использования в проектах с искусственным интеллектом и различными веб-технологиями, также может применяться для любых других задач, для которых требуется объектно-ориентированная база данных [4].

HyperGraphDB расширяет понятие ребер в графовой структуре. В ней ребро не просто может иметь более двух вершин, а также дает возможность ребру указывать на другое ребро. Кроме того, у каждого ребра и вершины есть возможность хранить произвольные сведения в качестве дополнительной информации [6].

Neo4j – графовая СУБД с открытым исходным кодом. На данный момент самая популярная. Работа с Neo4j ведется на языке Cypher, но запросы с помощью дополнительных модулей можно осуществлять и другими способами, например через Java API или с помощью языка Gremlin, созданного при разработке приложения TinkerPop [5].

Использование той или иной СУБД обуславливается целью проекта.

Преимущества графовых баз данных

Графовые базы данных обладают рядом преимуществ, главные из них производительность, гибкость и оперативность [6].

Прирост производительности особенно заметен при работе со взаимосвязанными данными, по сравнению с реляционными базами данных и NOSQL-хранилищами. Сложные запросы обычно выполняются быстрее в графовых базах данных, чем в реляционных. Последние требуют сложных соединений в таблицах данных для выполнения. В графовой базе данных запросы локализуются в определенной части графа. В результате время их выполнения зависит от размера части графа, которую требуется обойти для удовлетворения запроса, а не от общего размера графа.

Под гибкостью понимается возможность легкого изменения структуры базы данных согласно изменяющимся требованиям области применения. В реляционных базах данных модификация структуры сложнее либо невозможна, так как общая картина взаимосвязей предполагается заранее. Присущая графам возможность расширения означает, что можно добавлять новые виды взаимосвязей, узлы, метки и подграфы в существующую структуру, не нарушив при этом существующих запросов и функционала приложения. Это положительно влияет на производительность разработки и снижает риски для

проекта. Благодаря гибкости графовой модели не требуется предварительно моделировать задачу в мельчайших подробностях, что очень удобно из-за быстро меняющихся требований. Способность графов к расширению также позволяет уменьшить количество миграций, что снижает нагрузку при обслуживании данных и уменьшает риск их потери.

Говоря об оперативности, следует помнить, что модель данных не должна отставать от прочих составных частей приложения и использовать технологии, соответствующие современным итерационным методам развертывания программного обеспечения. Современные графовые базы данных оснащены всем необходимым для разработки и системного обслуживания. В частности, встроенная графовая модель данных, лишенная схем, в сочетании со встроенным программным интерфейсом (API) и языком запросов позволяет эффективно вести разработку приложений.

В то же время благодаря отсутствию схемы графовые базы данных не предполагают наличия ориентированных на схемы механизмов контроля данных, которые широко применяются в реляционном мире. В графовых базах данных они заменены гораздо более удобными и действенными видами контроля. Контроль выполняется в программной форме с помощью тестов для моделей данных и запросов, а также с помощью определения бизнес правил, основанных на графе.

Заключение

Область применения графовых баз данных активно расширяется, например они вносят большой вклад в развитие искусственного интеллекта. Более того, сегодня активно идет доработка RDF – основного стандарта, согласно которому работают графовые базы данных. При этом ряд проектов демонстрирует поддержку OData для создания стандартных веб-запросов через HTTP, а также языка SPARQL, обладающего обширными возможностями для работы с различными видами запросов и данных. За счет развития архитектуры производительность графовых баз данных растет, и все идет к тому, что она скоро окажется выше реляционной даже при небольшом количестве связей. Вероятно, в скором времени графовые базы данных станут незаменимы для разработчиков.

Библиографический список

1. Graph Database. URL: <https://www.g2.com/categories/graph-databases> (дата обращения: 26.04.2021).
2. Graph Databases. What's the Big Deal? URL: <https://towardsdatascience.com/graph-databases-whats-the-big-deal-ec310b1bc0ed> (дата обращения: 31.04.2021).
3. ArangoDB Documentation. URL: <https://www.arangodb.com/documentation/> (дата обращения: 25.04.2021).
4. HyperGraphDB Documentation. URL: <http://www.hypergraphdb.org/> (дата обращения: 05.04.2021).
5. Neo4j Documentation. URL: <https://neo4j.com/docs/> (дата обращения: 03.04.2021).
6. Guest View: Relational vs. graph databases: Which to use and when? URL: <https://sdtimes.com/databases/guest-view-relational-vs-graph-databases-use/> (дата обращения: 01.04.2021).

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ЗАЩИТА ИНФОРМАЦИИ

Безопасность информационных систем

УДК 612.087.1:57.087.1

А. И. Альмухамедов

студент кафедры безопасности информационных систем

В. С. Коломойцев – кандидат технических наук, доцент – научный руководитель

ПЕРСПЕКТИВНЫЕ МЕТОДЫ БИОМЕТРИЧЕСКОЙ ИДЕНТИФИКАЦИИ ЛИЧНОСТИ

Биометрическая идентификация (БИ) [1] – процесс сравнения предоставленного идентификатора, являющегося биометрическим параметром, со всеми записями из базы зарегистрированных пользователей. Сегодня в мире имеется ряд широко используемых методов БИ [2]. При этом разрабатываются и исследуются перспективные методы БИ. Целью данного исследования будет выявление перспективного метода БИ, не уступающего уже распространенным.

Для оценки и сравнения биометрических систем, использующих разные методы биометрической идентификации, используют следующие параметры [3]:

- вероятность ложного допуска (ВЛД) – доля транзакций верификаций самозванца, которые будут ошибочно приняты;
- вероятность ложного недопуска (ВЛНД) – доля транзакций верификации подлинного лица, которые будут ошибочно отвергнуты;
- чувствительность к влиянию внешних факторов;
- устойчивость к подделке;
- возможность бесконтактной идентификации;
- скорость идентификации;
- стоимость.

В данный момент выделяют четыре самых распространенных метода БИ [2]:

- по отпечатку пальца;
- по трехмерному изображению лица;
- по радужной оболочке глаза;
- по сетчатке глаза.

Около половины всех систем идентификации используют идентификацию по отпечатку пальца. Данные методы имеют совершенно разные параметры, что влияет на область их применения.

Выделяются перспективные методы БИ, использующие следующие биометрические параметры:

- запах тела;
- тон сердца;
- дезоксирибонуклеиновая кислота (ДНК);
- эмоциональное состояние и мимика;
- рисунок вен ладони.

Однако большинство из этих методов нуждается в исследованиях и дальнейших разработках [4]. Ключевой недостаток – достаточно низкая точность, не способная конкурировать с распространенными методами биометрической идентификации, в случае с ДНК-идентификацией – крайне низкая скорость. Однако идентификация по рисунку вен ладони лишена данных недостатков.

В основе метода БИ по рисунку вен ладони (РВЛ) лежит свойство крови поглощать излучение в ближнем ИК-диапазоне. Данный эффект может зарегистрировать любая современная камера. Устройство сканера (рис. 1) предполагает наличие ИК-подсветки, камеры и ИК-фильтра. При этом на получа-

емых монохромных изображениях можно различить рисунок вен. На рис. 2 изображена левая рука человека, снятая на сканер, сконструированный для дальнейших исследований.



Рис. 1. Схематичное устройство сканера вен ладони

Рисунок вен обладает хорошей биометрической характеристикой, так как он не изменяется с течением времени, уникален у каждого человека, на него не влияют внешние дефекты кожи [5].



Рис. 2. Снимок левой руки, сделанный с помощью сканера вен ладони

Таким образом, проведем сравнение распространенных методов БИ с выбранным перспективным методом БИ. Свойства методов БИ [6], сравниваются по основным параметрам БИ (табл. 1).

Как можно увидеть, метод идентификации по РВЛ по параметрам устойчивости к подделке и стоимости имеет аналогичные показатели, что и идентификация по ружке и сетчатке. При этом также обладает высокой скоростью и возможностью бесконтактной идентификации.

Таблица 1

Сравнительный анализ параметров для различных методов БИ

Метод БИ	Чувствительность к влиянию внешних факторов	Устойчивость к подделке	Возможность бесконтактной идентификации
По отпечатку пальца	Высокая	Низкая	Безуспешна
По трехмерному изображению лица	Низкая	Средняя	На среднем расстоянии
По радужной оболочке глаза	Средняя	Высокая	На большом расстоянии
По сетчатке	Высокая	Высокая	Безуспешна
По рисунку вен ладони	Средняя	Высокая	На небольшом расстоянии

Однако стоит помнить, что идентификация по отпечатку пальца и трехмерному изображению лица – наиболее распространенные на сегодня методы.

Переходя к показателям ВЛД и ВЛНД, отображенными в табл. 2, можно также заметить, что метод идентификации по РВЛ обладает показателями, приближенными к методам по сетчатке и по радужке, в то время как методы идентификации по отпечатку пальца и трёхмерному изображению лица показывают худший результат.

Таблица 2

Параметры ВЛД и ВЛНД для различных методов БИ

Метод БИ	ВЛД, %	ВЛНД, %
По отпечатку пальца	0,001	0,6
По трехмерному изображению лица	0,0005	0,1
По радужной оболочке глаза	0,00001	0,016
По сетчатке	0,0001	0,4
По рисунку вен ладони	0,0008	0,01

Наиболее важными для пользователя являются параметры скорости работы и стоимости сканера. Они рассмотрены в табл. 3.

Таблица 3

Скорость идентификации и стоимость для различных методов БИ

Метод БИ	Скорость идентификации	Стоимость
По отпечатку пальца	Высокая	Низкая
По трехмерному изображению лица	Средняя	Средняя
По радужной оболочке глаза	Высокая	Высокая
По сетчатке	Низкая	Высокая
По рисунку вен ладони	Высокая	Высокая

Таким образом, можно заметить, что распространенные методы идентификации подразделяются на группы, ключевым параметром в которых является стоимость (рис. 3).

Методы идентификации по отпечатку и трехмерному изображению лица (первая группа) обладают высокими показателями ВЛД и ВЛНД, имеют низкую устойчивость к подделке и достаточно низкую стоимость.

Во второй группе низкие параметры ВЛД и ВЛНД, высокую устойчивость к подделке компенсирует достаточно высокая стоимость, в десятки раз превышающая стоимость систем, основанных на первой группе методов.

На данный момент имеющиеся системы можно отнести ко второй группе как достаточно дорогостоящие, но обладающие высокой точностью и низкими показателями ВЛД и ВЛНД. Однако имеется возможность путем изменения конструкционных особенностей добиться результатов, которые позволили бы отнести данный метод к первой группе.

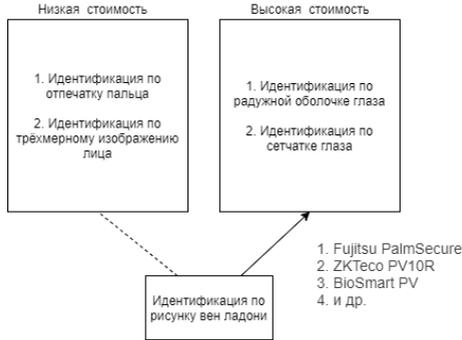


Рис. 3. Распределение методов БИ по стоимости сканеров

Системы идентификации по РВЛ не уступают системам, использующим распространенные методы БИ. Целью дальнейшего исследования будет разработка сканера РВЛ, сохраняющего прежние характеристики при понижении стоимости разработки и повышении показателей ВЛД и ВЛНД до уровня методов из группы с низкой стоимостью.

Библиографический список

1. Руководство по биометрии / Р. М. Болл, Дж. Х. Коннел, Ш. Панканти [и др.]. М.: Техносфера, 2007. 21 с.
2. J'son & Partners. Исследование российского рынка биометрических технологий, 2018–2022 гг. URL: <https://json.tv> (дата обращения: 14.04.2021).
3. ГОСТ Р ИСО/МЭК 19795-1–2007. Автоматическая идентификация. Идентификация биометрическая. Эксплуатационные испытания и протоколы испытаний в биометрии. М.: Стандартинформ, 2019. 6 с.
4. Перспективные методы биометрической аутентификации и идентификации / В. М. Колешко, Е. А. Воробей, П. М. Азизов [и др.]. Минск: БНТУ, 2009. 107 с.
5. Tome P., Marcel S. Palm Vein Database and Experimental Framework for Reproducible Research // 2015 Intern. Conf. of the Biometrics Special Interest Group (BIOSIG). 2015. С. 1–7.
6. Моржаков В. А. Современные биометрические методы идентификации // Безопасность. Достоверность. Информация. 2009. № 2. С. 44–48.

УДК 004.056.5

П. С. Бухвалов

студент кафедры безопасности информационных систем

А. В. Воронов – кандидат технических наук, доцент – научный руководитель

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТИ ИСПОЛЬЗОВАНИЯ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ПРИ ПРОВЕДЕНИИ СПЕЦИАЛЬНЫХ ИССЛЕДОВАНИЙ ТЕХНИЧЕСКИХ СРЕДСТВ

Введение

Глобальная информатизация в современном мире привела к массовому распространению технических средств, позволяющих осуществлять обработку, хранение и передачу больших объемов информации. При этом в процессе работы с информацией необходимо обеспечивать ее конфиденциальность, целостность и доступность. К техническим средствам, на которых осуществляется обработка конфиденциальной информации предъявляется ряд требований, которые позволяют существенно снизить риски реализации угроз со стороны злоумышленников.

Поскольку нарушение конфиденциальности обрабатываемой информации с высокой вероятностью может происходить за счет побочных электромагнитных излучений технического средства, то для выявления и устранения данных технических каналов утечки информации были разработаны комплексы мероприятий, получившие название «Специальные исследования».

Специальные исследования технических средств

Специальные исследования технических средств – исследования, проводимые в целях выявления технических каналов утечки защищаемой информации, возникающих в результате побочных электромагнитных излучений технических средств обработки информации. Основной задачей при проведении специальных исследований является комплексная проверка различных технических средств на соответствие их требованиям по защите информации от утечек по техническим каналам [1].

В рамках проведения специальных исследований выявление технических каналов утечки осуществляется для двух типов технических средств и систем: основных технических средств и систем (ОТСС) и вспомогательных технических средств и систем (ВТСС).

В процессе функционирования любого технического средства, выполняющего обработку информации, по его элементам неизбежно протекают электрические токи различных частот. Это становится причиной возникновения магнитных и электрических полей, называемых побочными электромагнитными излучениями и наводками (ПЭМИН) [2].

Канал утечки информации через ПЭМИН – один из самых актуальных технических каналов утечки, так как при обработке информации с помощью ОТСС неизбежно возникают ПЭМИН, несущие информативный сигнал [3].

В соответствии с методикой проведения специальных исследований технических средств по измерению их собственного электромагнитного излучения проводятся следующие операции [4, 5]:

1) на расстоянии одного метра от исследуемого устройства устанавливаются поочередно антенны для приема электрической и магнитной составляющих поля, излучаемого анализируемым устройством;

2) сигнал с выхода антенны подается на вход приемно-регистрирующего измерительного устройства, с помощью которого по результатам измерений по определенной методике производится расчет опасных зон.

На рис. 1 представлена типовая схема измерения ПЭМИН.

Измерение излучения осуществляется в двух режимах работы технического устройства: *обычном* и *тестовом*, когда излучение исследуемого технического средства является максимально возможным [3].

В результате измерения формируется таблица значений (табл. 1).

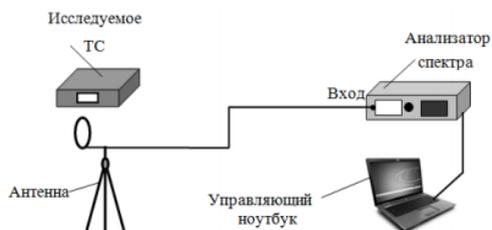


Рис. 1. Схема измерения ПЭМИН

Таблица 1

Результаты измерений ПЭМИН

Частота, Гц	Интенсивность излучения в нормальном режиме, дБ/м ²	Интенсивность излучения в тестовом режиме, дБ/м ²
2,364	39,514 6	44,434 5
2,373	39,950 3	42,531 9
2,382	32,319 4	33,266 1
2,391	50,198 8	51,482 2
2,400	62,619 7	61,075 4
2,409	61,678 5	58,901 8
2,418	51,657 2	41,131 8
2,427	39,935 3	42,930 3

Всего в данной таблице может содержаться примерно до 20 000 числовых значений. Замеры осуществляются в частотном диапазоне от 0 до 6 000 Гц.

Далее из всего множества значений напряженности осуществляется выбор наиболее значимых с точки зрения величины излучения. Предполагается, что именно участки с наибольшей интенсивностью излучения представляют угрозу, так как именно они являются информативными и могут быть перехвачены злоумышленником. Для каждого такого значения осуществляется вычисление значения зоны R2.

Зоной R2 обозначается пространство вокруг технического средства обработки информации, в пределах которого отношение «опасный сигнал / помеха» для составляющих напряженности электромагнитного поля превышает допустимое нормированное значение. Результатом анализа становится выбор максимального значения зоны R2 из всего множества рассчитанных значений, которое в дальнейшем используется как основной критерий в выборе средства защиты.

При определенных условиях сторонние излучения могут в значительной степени повлиять на результаты измерений. Тогда на приборах можно зафиксировать интенсивность излучения опасного информативного сигнала значительно выше, чем она есть на самом деле. Из-за этого будет неверно рассчитано значение зоны R2, что приведет к выбору средства защиты с неверными характеристиками.

Поэтому главной задачей в процессе проведения исследований на объекте информатизации является исключение из рассмотрения сторонних излучений, устранение их влияния и выделение только практически полезного информативного сигнала. Перспективным способом улучшения процедуры специальных исследований выступает использование методов искусственного интеллекта в виде алгоритмов машинного обучения.

В рамках работы бакалавра была поставлена цель исследовать возможность использования методов машинного обучения для решения задачи анализа данных, полученных в результате специальных исследований.

Алгоритмы машинного обучения

Под машинным обучением понимается класс методов, использующих принципы искусственного интеллекта. Особенность данных принципов состоит не в прямом решении задачи, а в обучении в процессе анализа результатов решений для множества сходных задач с уже определенными ответами. За счет этого появляется возможность решать нечетко поставленные задачи в условиях неполноты данных или их искажения.

Основной характеристикой систем, разрабатываемых с помощью методов машинного обучения, становится их способность к обучению. В зависимости от метода обучения данные алгоритмы относятся к одному из определенных классов моделей машинного обучения. Наиболее популярными и активно развивающимися являются классы, использующие следующие два подхода: обучение с учителем и обучение без учителя. Схема основных классов машинного обучения приведена на рис. 2.

Обучение с использованием учителя применяется для решения задач регрессии и классификации [6, 7] (рис. 3).

Обучение без учителя используется в задачах кластеризации или уменьшения размерности данных [6, 7] (рис. 4).

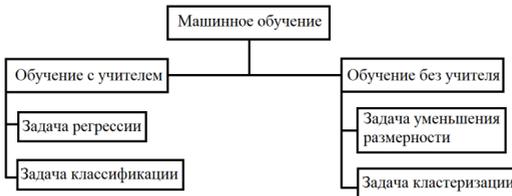


Рис. 2. Основные классы моделей машинного обучения



Рис. 3. Алгоритмы обучения с учителем



Рис. 4. Алгоритмы обучения без учителя

В процессе работы было установлено, что данные, полученные в ходе специальных исследований, необходимо проанализировать и определить, относятся ли они к классу нормальных, или же в них присутствуют посторонние шумы, и данные являются аномальными. Таким образом, алгоритмы, решающие задачи регрессии и уменьшения размерности, концептуально не могут быть использованы для решения заявленной задачи, а значит, данные алгоритмы далее можно не рассматривать.

В классе обучения с учителем задача отнесения объекта к какому-либо классу есть задача классификации, а алгоритмы, решающие ее, называются классификаторами. В классе обучения без учителя сходная задача называется задачей кластеризации и решается с помощью алгоритмов, названных кластеризаторами. Под *классификацией* понимают процесс присваивания заранее определенного класса рассматриваемому объекту по набору его признаков. В процессе *кластеризации* формирование классов происходит в процессе решения задачи. Исходное множество объектов разделяется на классы по сходству признаков.

Задачу поиска аномалий можно представить в виде задачи определения класса принадлежности объекта с одним классом нормальных объектов. При этом определен один класс – класс нормальных объектов. Любые объекты, не отнесенные к нему, считаются аномальными.

Таким образом, для решения поставленной задачи можно использовать как алгоритмы классификации, так и алгоритмы кластеризации. Значит, следует рассмотреть наиболее популярные и универсальные алгоритмы кластеризации и классификации. Для этого были выбраны классификаторы на основе метода опорных векторов и метода *k*-ближайших соседей, а также кластеризаторы на основе эллипсоидальной аппроксимации данных и изолирующего леса.

Алгоритмы классификации

Одни из наиболее популярных алгоритмов классификации – метод *k*-ближайших соседей и метод опорных векторов.

Метод *k*-ближайших соседей – один из самых простых алгоритмов классификации. В процессе обучения алгоритм запоминает все рассматриваемые векторы признаков, а также метки классов принадлежности для данных векторов и сохраняет их [8].

Для классификации тестовых объектов первоначально осуществляется вычисление расстояния до каждого из объектов из обучающей выборки с помощью соответствующей функции расстояния. Классическим вариантом функции расстояния является евклидово расстояние. После вычисления расстояния из всех объектов обучающей выборки выбираются *k* таких, что расстояние от каждого из них до тестового объекта будет минимальным. Тестовому объекту присваивается тот класс, который присвоен наибольшему числу объектов из выбранных *k* штук. На рис. 5, а проиллюстрирован принцип работы метода *k*-ближайших соседей.

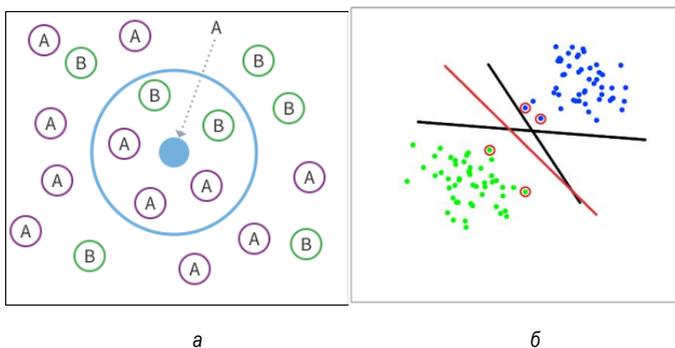


Рис. 5. Примеры процедур классификации: а – метод опорных векторов; б – метод *k*-ближайших соседей

Метод нагляден и прост в понимании, а также достаточно просто реализуется, однако при этом показывает хорошие результаты. Его главный недостаток – высокая вычислительная трудоемкость. При этом увеличение сложности происходит квадратично с ростом числа обучающих примеров [9].

В методе опорных векторов осуществляется перевод исходных векторов в пространство более высокой размерности и построении оптимальной разделяющей гиперплоскости в многомерном пространстве. Идею данного метода удобно проиллюстрировать на примере, представленном на рис. 5, 6.

Исходное множество объектов, которые уже были классифицированы в процессе обучения, представляются в виде множества точек на плоскости, сгруппированных в классы. Далее все новые объекты, которые необходимо классифицировать, также представляются в виде точек и автоматически классифицируются следующим образом: точки, находящиеся выше прямой, попадают в один класс, ниже прямой – в другой.

Если признаков у объектов больше, чем два, то двумерное пространство превращается в пространство большей размерности. Тогда вместо прямой рассматриваются гиперплоскости – пространства, размерность которых на единицу меньше, чем размерность исходного пространства.

Для более точной классификации выбирается гиперплоскость, удаленная от каждого имеющегося класса на максимально возможное расстояние. Она называется оптимальной разделяющей гиперплоскостью. Векторы, лежащие ближе всех к такой гиперплоскости, называются опорными.

В случае если есть возможность линейно разделить исходные обучающие данные, то дополнительно осуществляется построение двух разделяющих гиперплоскостей, которые параллельны оптимальной разделяющей гиперплоскости.

На рис. 6 представлен пример решения задачи классификации объектов методом опорных векторов. Приведены два варианта классификации: неоптимальный (слева) и оптимальный (справа):

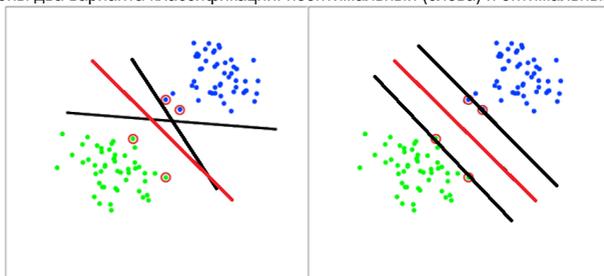


Рис. 6. Пример неоптимальной (слева) и оптимальной (справа) классификации методом опорных векторов

Несмотря на простоту идеи, данный метод получил большое распространение и применяется во множестве областей.

Алгоритмы кластеризации

Метод изолирующего леса использует более простую структуру принятия решений, так называемые деревья решений. Дерево решений состоит из двух ключевых элементов: решающее правило – условие или проверка в узле дерева и лист дерева – один из конечных узлов дерева, содержащий результат проверки.

Само дерево решений можно описать как метод представления решающих правил в некой иерархической структуре в виде дерева, состоящего из узлов и листьев. В каждом узле осуществляется проверка соответствия рассматриваемого объекта решающему правилу по какому-либо из признаков, определенных на основе обучающей выборки. Признаки выбираются случайным образом. В результате проверки множество объектов, попавших в узел, разбивается на два подмножества, в одно из которых попадают объекты, удовлетворяющие правилу, а в другое – не удовлетворяющие. Данная процедура рекурсивно повторяется, пока не будет достигнуто условие остановки алгоритма.

Для того чтобы рассматриваемый объект оказался в листе дерева он должен удовлетворять всем решающим правилам, лежащим на пути к этому листу. Поскольку путь в дереве к каждому листу единственный, то и каждый объект может попасть только в один лист, что обеспечивает единственность решения. При этом аномальным объектам свойственно оказываться в листьях с небольшой глубиной.

Сущность метода изолирующего леса заключается в построении на этапе обучения множества решающих деревьев. Предпочтительно использование их большого количества, так как за счет этого результат получается более точным.

После осуществления процедуры построения деревьев на этапе обучения метод получает возможность прогнозировать, в каком именно листе может оказаться объект. Принцип выявления аномалий строится на том, что неестественные объекты, как правило, очень редки, а потому достаточно быстро определяются. То есть длина пути от корня дерева до листа для аномальных объектов очень мала. На рис. 7 приведена общая структура изолирующего леса. За счет древовидной структуры данный метод имеет меньшую сложность, по сравнению с другими методами, и при этом не требует значительных затрат памяти.

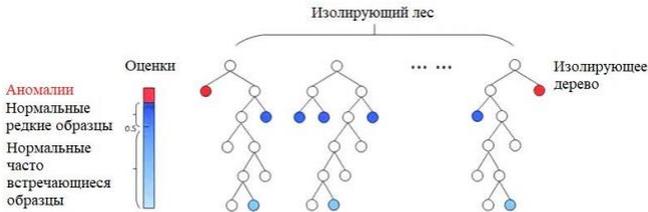


Рис. 7. Схематичное представление изолирующего леса

Метод эллипсоидальной аппроксимации также один из наиболее популярных способов обнаружения аномалий. В нем заложено предположение о том, что рассматриваемые данные распределены по некоторым известным законам. Задача выявления аномалий заключается в определении конкретного закона распределения и выделения объектов, этому распределению не удовлетворяющих. В методе эллипсоидальной аппроксимации данных исходное множество точек моделируется как внутренность эллипсоида, а аномальные объекты определяются за счет вычисления расстояния Махаланобиса [13]. Схематичный принцип работы метода представлен на рис. 8.

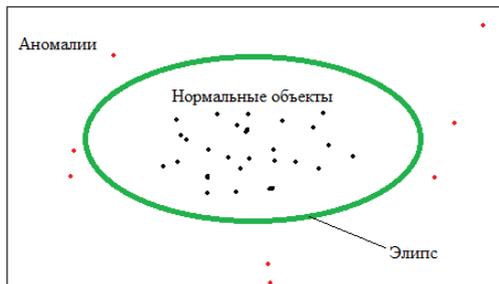


Рис. 8. Метод эллипсоидальной аппроксимации данных

Расстояние Махаланобиса – это мера расстояния между векторами случайных величин, с помощью которой можно определять сходство известной и неизвестной выборки. Принадлежность объекта тому или иному классу определяется по минимуму расстояния Махаланобиса, что эквивалентно ме-

тоду максимального правдоподобия [13]. Метод лучше всего работает на данных, распределенных по нормальному закону.

Заключение

Таким образом, была поставлена задача исследовать возможность использования алгоритмов, использующих методы машинного обучения, для выявления аномальных и неестественных значений измерений, получаемых в ходе проведения специальных исследований технических средств. Был проведен анализ структуры машинного обучения, по результатам которого определено, что для решения поставленной задачи следует рассматривать алгоритмы, решающие задачу классификации в случае алгоритмов обучения с учителем и задачу кластеризации в случае алгоритмов обучения без учителя. Алгоритмы, решающие задачи регрессии и уменьшения размерности, были исключены из рассмотрения, так как в данном случае использоваться не могут.

Среди алгоритмов классификации были выбраны два наиболее распространенных: алгоритм, использующий методы опорных векторов, и алгоритм *k*-ближайших соседей. Среди алгоритмов кластеризации также были выбраны наиболее популярные: изолирующего леса и эллипсоидальной аппроксимации данных.

Для дальнейшего исследования необходимо осуществить экспериментальную оценку эффективности выполнения поставленной задачи каждым из выбранных алгоритмов машинного обучения, а также провести их сравнительный анализ.

Библиографический список

1. ГОСТ Р 50922-2006. Библиографическая запись. Библиографическое описание: защита информации. Основные термины и определения / Федер. агентство по техн. регулированию и метрологии. М.: Изд-во стандартов, 2006.
2. Антипов Д. А. Анализ утечки информации на основе побочных электромагнитных излучений // Докл. ТУСУРа. 2018. Т. 21, № 2. С. 27–32.
3. Артамошин С. А. Защита информации от утечки по каналу побочных электромагнитных излучений // Правовая информатика. 2016. № 3. С. 4–11.
4. Защита информации от утечки по техническим каналам. Ч. 1. Организация работ по технической защите информации. Технические каналы утечки информации. 2020. URL: <https://trinosoft.com/index.php?page=/is/tzitektv1/> (дата обращения: 10.04.2021).
5. Васильев Р. А., Ротков Л. Ю. Обнаружение электромагнитных излучений и наводок с помощью программно-аппаратного комплекса «Легенда»: учеб.-метод. пособие. Нижний Новгород: Нижегород. гос. ун-т, 2018. 45 с.
6. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. М.: ДМК Пресс, 2015. 400 с.
7. Миронов А. М. Машинное обучение. Ч. 1: учеб. пособие. М.: Моск. гос. ун-т им. М. В. Ломоносова: МАКС Пресс, 2018. 83 с.
8. Классификатор kNN. 2012. URL: <https://habr.com/ru/post/149693/> (дата обращения: 09.04.2021).
9. Классификация данных методом опорных векторов. 2010. URL: <https://habr.com/ru/post/105220/> (дата обращения: 10.04.2021).
10. 9 подходов для выявления аномалий. 2019. URL: <https://habr.com/ru/post/477450/> (дата обращения: 09.04.2021).
11. Поиск аномалий (Anomalydetection). 2017. URL: <https://dyakonov.org/2017/04/19/%D0%BF%D0%BE%D0%B8%D1%81%D0%BA%D0%B0%D0%BD%D0%BE%D0%BC%D0%B0%D0%BB%D0%B8%D0%B9-anomaly-detection/> (дата обращения: 09.04.2021).
12. Деревья решений: общие принципы. 2019. URL: <https://loginom.ru/blog/decision-tree-p1> (дата обращения: 10.04.2021).
13. Machine learning for anomaly detection: Elliptic Envelope. 2020. URL: <https://towardsdatascience.com/machine-learning-for-anomaly-detection-elliptic-envelope-2c90528df0a6> (дата обращения: 09.04.2021).

УДК 621.391

А. М. Вересова

магистрант кафедры информационной безопасности

А. А. Овчинников – кандидат технических наук, заведующий кафедрой – научный руководитель

**О СТАТИСТИЧЕСКИХ СВОЙСТВАХ МОДЕЛЕЙ ГИЛБЕРТА И ГИЛБЕРТА – ЭЛЛИОТА
ДЛЯ КАНАЛОВ С ПАМЯТЬЮ**

Введение

Информация постоянно передается по различным каналам: проводным и беспроводным. При этом она может быть искажена вследствие возникновения помех в канале. Как правило, в теории кодирования рассматриваются независимые ошибки, но в реальности они часто группируются [1; 2]. Для борьбы с ошибками может использоваться интерливинг, или перемежение, вследствие которого сгруппированный шум рассеивается, позволяя использовать алгоритмы, построенные для исправления независимых ошибок. Однако известно, что учет памяти канала повышает возможную скорость надежной передачи [3; 4]. Задача построения кодов и алгоритмов для таких каналов связана с их статистическими свойствами. В данной статье рассматриваются статистические свойства наиболее распространенных моделей каналов с памятью.

Модели каналов связи

Для описания среды передачи, в которой возникают помехи, используются математические вероятностные модели – модели каналов связи. Один из простых примеров такой модели – двоичный симметричный канал (ДСК) [5]. Ошибки в ДСК происходят с одинаковой вероятностью p в каждом бите передаваемой последовательности и представляют собой инвертирование значения бита.

Однако, когда канал обладает памятью, вероятность ошибки в текущий момент зависит от предыдущего состояния (или последовательности состояний) канала. Для описания таких каналов используется аппарат марковских цепей, а соответствующие модели называются моделями каналов с конечным числом состояний (ККЧС) [6].

Рассмотрим ККЧС с двумя состояниями: хорошим (good, G) и плохим (bad, B). Когда канал находится в хорошем состоянии, происходит мало ошибок, когда в плохом – много. Таким образом, каждое состояние может быть описано при помощи ДСК с такими вероятностями ошибок, что $0 \leq \eta_G \ll \eta_B \leq 1$, где η_G и η_B – вероятности ошибки в хорошем и плохом состоянии соответственно. Для полного описания модели необходимо также задать вероятности перехода из хорошего состояния в плохое и обратно, которые и определяют наличие памяти в канале. Описанная модель называется каналом Гилберта – Эллиота [7] и показана на рис. 1.

Наиболее наглядной разновидностью модели Гилберта – Эллиота является марковская модель, в которой $\eta_G = 0$, а $\eta_B = 1$. В такой модели состояние канала однозначно определяет наличие или отсутствие ошибки в соответствующий момент времени. Также часто используется модель Гилберта, где $\eta_G = 0$, а $\eta_B = 0,5$ [8].

Зная переходные вероятности модели Гилберта – Эллиота, можно вычислить безусловную вероятность нахождения в одном из состояний следующим образом:

$$P_G = \frac{P_{BG}}{P_{GB} + P_{BG}}, \quad P_B = \frac{P_{GB}}{P_{GB} + P_{BG}}.$$

Кроме того, может быть вычислена безусловная вероятность ошибки в канале:

$$\eta = \eta_G P_G + \eta_B P_B.$$

Для описанных моделей вероятность того, что время t нахождения в канале будет равно некоторому значению x , равна

$$\Pr(t = x) = P_{BB}^{x-1} P_{BG}$$

Таким образом, время пребывания в некотором состоянии является случайной величиной, распределенной по геометрическому закону с математическим ожиданием $1/P_{GB}$ для состояния В и $1/P_{BG}$ для состояния G.

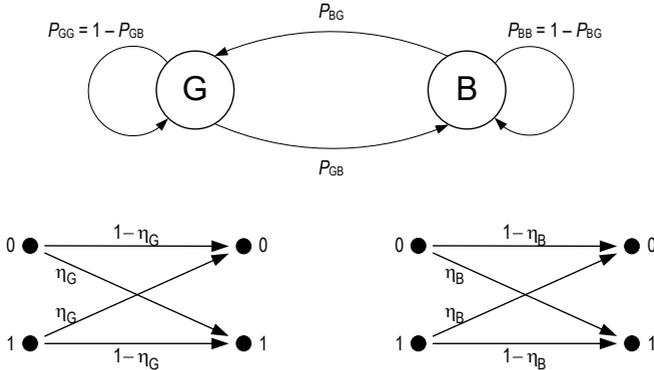


Рис. 1. Модель Гилберта – Эллиота

Набор переходных вероятностей, описывающий модель Гилберта – Эллиота, может быть заменен другим набором из двух параметров, связывающих переходные вероятности между собой [9]. Первый параметр характеризует отношение «хорошего к плохому»:

$$\rho = \frac{P_{BG}}{P_{GB}}$$

его значение находится в интервале $[0, \infty)$. При $\rho = 0$ канал фактически всегда находится в плохом состоянии, а при $\rho = \infty$ – в хорошем.

Второй параметр характеризует память в канале:

$$\mu = 1 - P_{BG} - P_{GB}$$

Он находится в интервале $[-1, +1]$, причем $\mu \geq \max\{-\rho, -\rho^{-1}\}$. При $\mu = 0$ канал не обладает памятью. При положительном значении μ канал стремится к сохранению состояния и полностью определяется начальным состоянием при $\mu = 1$. Когда $\mu < 0$, канал стремится к изменению состояния, и при $\mu = -1$ меняет состояние на каждом шаге.

Переходные вероятности из параметров μ и ρ могут быть найдены с использованием выражений

$$P_{GB} = \frac{1-\mu}{\rho+1}, \quad P_{BG} = \rho P_{GB}$$

Когда канал достаточно долго находится в плохом состоянии, образуется пакет ошибок – комбинация, в которой первый и последний ошибочный символ отстоят друг от друга не более чем на некоторое количество символов, называемое длиной пакета. Параметры пакета ошибок зависят от вероятности ошибки в плохом состоянии, и при низком значении η_B даже при длительном нахождении канала в плохом состоянии может произойти малое количество ошибок. Однако чаще всего рассматри-

ваются случаи, когда η_B сравнимо с 0,5, и время нахождения в состоянии может рассматриваться как длина пакета ошибок.

Статистические свойства моделей

В теории кодирования для борьбы с ошибками в каналах с памятью, помимо интерливинга, существует направление, связанное с разработкой кодов, исправляющих пакеты ошибок [10–12]. При этом кодовые конструкции могут характеризоваться:

- возможностью исправления однократных или многократных пакетов;
- длиной исправляемых пакетов ошибок;
- возможностью исправления одиночных ошибок, возникающих в хорошем состоянии;
- исправляемой долей ошибок в пакете, зависящей от η_B , и другими параметрами.

Для исследования статистических свойств моделей каналов с точки зрения перечисленных характеристик построим модели Гилберта и Гилберта – Эллиота.

В табл. 1 представлены результаты исследования модели Гилберта с различными переходными вероятностями при передаче $n = 1\,000$ символов. Левая часть таблицы содержит заданные параметры канала, а правая – результаты моделирования, где t – время нахождения в плохом состоянии, b – длина пакета ошибок, wt – вес Хемминга пакета, т. е. количество ошибок внутри него. Можно заметить, что средняя длина пакета примерно равна среднему времени нахождения в плохом состоянии, как и было сказано, а вес пакета вдвое меньше его длины, как и должно быть при $\eta_B = 0,5$. Параметры ρ и μ говорят, что при $P_{GB} = P_{BG} = 0,01$ канал стремится к сохранению плохого состояния; действительно, результаты показывают, что с вероятностью, близкой к единице, при передаче возникнет более двух пакетов ошибок. При уменьшении вероятности P_{GB} до 0,001 канал становится лучше: примерно треть передач происходит без ошибок. При дальнейшем уменьшении P_{GB} до 0,0001 в канале становится возможным эффективное применение интерливинга: 90 % передач происходит без помех, что позволяет распределить сгруппированные ошибки так, чтобы доля ошибочных бит в каждой передаче была достаточно малой для исправления классическими алгоритмами.

Таблица 1

Свойства канала Гилберта

Параметры канала				Вероятность возникновения x пакетов				Средние значения		
P_{GB}	P_{BG}	ρ	μ	$x = 0$	$x = 1$	$x = 2$	$x > 2$	t	b	wt
0,01	0,01	1	0,98	0,00004	0,00175	0,01817	0,98004	100	99	51
	0,03	3	0,96	0,00002	0,00059	0,00314	0,99625	33	32	17
	0,1	10	0,89	0,00008	0,00043	0,00252	0,99697	10	9	6
0,001	0,01	10	0,989	0,33645	0,40500	0,19652	0,06203	94	93	48
	0,03	30	0,969	0,35617	0,37971	0,18877	0,07535	33	32	17
	0,1	100	0,899	0,36484	0,37133	0,18587	0,07796	10	9	6
0,0001	0,01	100	0,9899	0,89723	0,09804	0,00461	0,00012	91	90	46
	0,03	300	0,9699	0,90109	0,09418	0,00459	0,00014	32	31	17
	0,1	1000	0,8999	0,90290	0,09217	0,00484	0,00009	10	9	5

На рис. 2 показано распределение длин пакетов ошибок при $P_{GB} = 0,01$, т. е. математическое ожидание данной величины 100. Значения распределены по геометрическому закону, и с наибольшей вероятностью возникают пакеты малой длины.

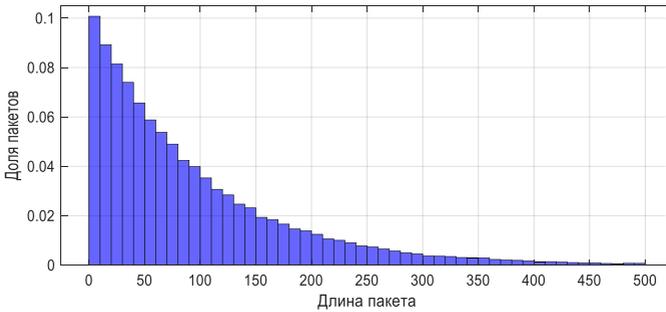


Рис. 2. Распределение длин пакетов ошибок при $P_{GB} = 0,01$

Теперь рассмотрим модель Гилберта – Эллиота. Здесь большой интерес представляет влияние вероятностей η_B и η_G на свойства канала, поэтому можно зафиксировать значения $P_{GB} = 0,001$ и $P_{BG} = 0,01$. Вероятности возникновения пакетов ошибок при этом останутся такими же, как в табл. 1. Результаты оценки свойств канала Гилберта – Эллиота представлены в табл. 2; величина e соответствует количеству ошибок, возникших в хорошем состоянии за время передачи.

Таблица 2

Свойства канала Гилберта – Эллиота

Параметры канала				Средние значения			
P_{GB}	P_{BG}	η_G	η_B	t	b	wt	e
0,001	0,01	0,01	0,01	95	46	2	6
			0,1	94	86	10	6
			0,3	94	92	29	6
			0,5	94	93	48	6
			0,9	94	94	85	6
			1	95	95	95	6

По представленным в табл. 2 результатам видно, что при малых значениях η_B длина пакета заметно меньше, чем время нахождения в плохом состоянии. Фактически пакет длиной 46 и весом 2 представляет собой две независимые ошибки, произошедшие на расстоянии 46. Для исправления ошибок в таком канале может не потребоваться специальных алгоритмов или кодовых конструкций.

Заключение

В работе представлены результаты анализа моделей Гилберта и Гилберта – Эллиота для каналов с памятью. Показано влияние параметров моделей на их статистические свойства, имеющие значение при построении кодовых конструкций для подобных каналов: длину и количество пакетов

ошибок, их вес, а также различие между временем нахождения в плохом состоянии и длиной пакета ошибок.

Библиографический список

1. MacWilliams F. J., Sloane N. J. A. The Theory of Error-Correcting Codes. Vol. 16. North-Holland publ. comp., 1983. 782 p.
2. Morelos-Zaragoza R. H. The art of error correcting coding. 2nd ed. Chichester; Hoboken, NJ: John Wiley, 2006. 263 p.
3. Shannon C. E. A Mathematical Theory of Communication // Bell System Techn. J., 1948. Vol. 27, № 3, P. 379–423.
4. Wolfowitz J. Memory increases capacity // Information and Control. 1967. Vol. 11, № 4, P. 423–428.
5. Кудряшов Б. Д. Теория информации: учеб. для вузов. СПб.: Питер, 2009. 320 с.
6. Gallager R. G. Information theory and reliable communication. N.-Y.: John Wiley & Sons, Inc., 1968. 608 p.
7. Elliott E. O. Estimates of error rates for codes on burst-noise channels // The Bell System Techn. J. 1963. Vol. 42, № 5, P. 1977–1997.
8. Gilbert E. N. Capacity of a burst-noise channel // The Bell System Techn. J. 1960. Vol. 39, № 5, P. 1253–1265.
9. Mushkin M., Bar-David I. Capacity and coding for the gilbert-elliott channels // IEEE Transactions on Information Theory. 1989. Vol. 35, № 6. P. 1277–1290.
10. Zhang W., Wolf J. K. A class of binary burst error-correcting quasi-cyclic codes // IEEE Transactions on Information Theory. 1988. Vol. 34, № 3. P. 463–479.
11. Eckford A. W., Kschischang F. R., Pasupathy S. Designing Very Good Low-Density Parity-Check Codes for the Gilbert-Elliott Channel // Proceedings of Canadian Workshop on Information Theory. Waterloo, Canada, 2003. P. 40–44.
12. Krouk E., Ovchinnikov A. 2-Stripes Block-Circulant LDPC Codes for Single Bursts Correction // Intelligent Interactive Multimedia Systems and Services 2016: Smart Innovation, Systems and Technologies. Cham: Springer Intern. Publ., 2016. P. 11–23.

УДК 004.056.2

Д. А. Зыков

студент кафедры безопасности информационных систем

А. В. Афанасьева – старший преподаватель – научный руководитель

МОДЕЛИРОВАНИЕ СИСТЕМ ХРАНЕНИЯ ДАННЫХ ДЛЯ ОЦЕНКИ ВЛИЯНИЯ ИЗБЫТОЧНОСТИ НА ОТКАЗООУСТОЙЧИВОСТЬ

В связи с развитием информационных технологий увеличивается объем данных в цифровом виде. Для организации их хранения используются системы хранения данных (СХД) – комплексное программно-аппаратное решение, позволяющее хранить большие объемы данных и предоставлять к ним доступ с использованием различных сетевых протоколов.

Многие аналитики отмечают рост спроса на услуги по хранению данных, это закономерная тенденция развития рынка информационных услуг. Основными драйверами роста рынка СХД считаются финансовый и телекоммуникационный сектор, а также государственные органы власти. С ростом объемов данных изменяются требования к параметрам систем их хранения, таким как производительность и энергоэффективность.

Современные центры обработки данных содержат крупномасштабные системы хранения данных, используемые для служб облачных вычислений и аналитики больших данных. Исследования показали, что сбои, временные или постоянные, – обычным явлением в таких центрах. Чтобы защитить хранилище данных от сбоев, применяют кодирование со стиранием с целью добавить избыточность в хранилище данных, чтобы любые недоступные или потерянные данные можно было восстановить из других доступных избыточных данных. Кодирование со стиранием обеспечивает эффективный для хранения способ создания избыточности в хранилище данных и, очевидно, требует меньшей избыточности по объему используемого физического пространства, чем простое зеркалирование данных. С другой стороны, кодирование со стиранием имеет недостаток, связанный с высокими штрафами за восстановление, поскольку восстановление любых потерянных данных с кодированием стирания инициирует передачу гораздо большего количества доступных данных, чем фактический объем потерянных данных. Объем трафика восстановления может достигать сотен терабайт в день в производственных центрах обработки данных и перегружать ресурсы полосы пропускания для приоритетных приложений [1].

Для расчета надежности системы хранения данных, построенной на основе кодирования со стираниями, были использованы теоретическая модель и имитационное моделирование.

При использовании кодов, исправляющих ошибки, данные остаются доступными до тех пор, пока не будет потеряно больше дисков, чем имеется дисков четности. Если отказывает большее число дисков, восстановить данные будет невозможно [2].

Введены следующие обозначения: n – общее количество дисков (информационные диски, диски четности); k – минимальное количество дисков, необходимое для восстановления информации; R – время восстановления диска в днях: сколько времени требуется для замены диска после его сбоя; A – ежегодная интенсивность отказов одного диска; F – интенсивность отказов диска за R дней; P – вероятность отказа диска хотя бы раз за R дней; D – доступность данных в течение R дней.

В течение одного года вероятность отказа диска равномерно распределяется по всем R -дневным периодам в году. Для расчета F применена следующая формула:

$$F = A \cdot \frac{R}{365}.$$

Вероятность отказа для диска в течение R дней вычисляется из распределения Пуассона:

$$P = 1 - e^{-F}.$$

Используя вероятность отказа одного диска, найдена вероятность отказа ровно r из n дисков, где $r = n - k$:

$$C_n^r P^r (1-P)^{n-r} .$$

Потеря данных происходит при выходе из строя более r дисков. Для того чтобы получить вероятность потери данных за R дней, просуммируем приведенную формулу от r до n :

$$\sum_{i=r}^n C_n^i P^i (1-P)^{n-i} .$$

При помощи полученной формулы можно выразить доступность данных в R -дневный период:

$$D = 1 - \sum_{i=r}^n C_n^i P^i (1-P)^{n-i} .$$

Данные считаются доступными в течение года, если они доступны во всех R -дневных периодах, следовательно, доступность вычисляется как произведение вероятностей:

$$D^{365/R} .$$

Избыточность системы хранения по используемому физическому пространству можно оценить при помощи следующего соотношения:

$$\frac{k}{n} \cdot 100\% .$$

Для подтверждения теоретических расчетов была построена имитационная модель, позволяющая оценить доступность системы хранения данных за заданный временной период. Упрощенный псевдокод имитационного моделирования приведен на рис. 1. Усредненные результаты имитационного моделирования совпадают с теоретическими расчетами.

```

1:  function SIMULATE
2:       $t_{unavailable} = 0$ 
3:      while true do
4:          if  $t > T$ 
5:              return  $\frac{t_{unavailable}}{t}$ 
6:           $f \leftarrow$  Poisson distribution
7:          set  $t_{maintenance} = T_f$  for  $f$  random disks
8:          if  $f >$  fault tolerance
9:               $t_{unavailable} = t_{unavailable} + t_{recovery}$ 
10:              $t = t + t_{recovery}$ 
11:             for each disk in failed disks
12:
13:                 if  $t_{maintenance} = 0$ 
14:                     set disk state to "Normal"
15:                  $t = t + 1$ 
16:             end
17:         end

```

Рис. 1. Псевдокод имитационного моделирования

Для наиболее популярных схем кодирования со стираниями (обозначаемых как $k+r$, где k – информационные блоки, r – блоки избыточности) были рассчитаны показатели доступности и избы-

точности. В качестве значения годовой интенсивности отказов для дисков, используемых в системе хранения, было использовано значение 0.00244, являющееся средней годовой интенсивностью отказов для твердотельных накопителей [3].

Полученные результаты доступности и избыточности наиболее популярных схем систем хранения данных, использующих кодирование со стираниями, приведены на рис. 2. Доступность представлена в левых столбцах в виде «количества девяток» (доступность 0.999 обозначается как «3 девятки»), а избыточность в правых столбцах в процентном соотношении.

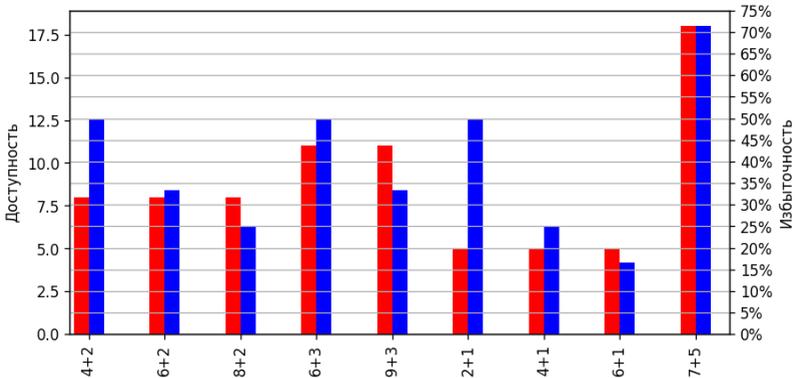


Рис. 2. Доступность и избыточность для различных схем СХД

Из приведенных результатов можно сделать вывод что наиболее эффективны схемы 8 + 2, 9 + 3 и 6 + 1, позволяющие максимизировать доступность при наименьшей избыточности.

В дальнейшем планируется изучить влияние скорости и надежности сети на доступность системы хранения данных, а также рассмотреть схемы, использующие репликацию и сжатие данных.

Библиографический список

1. Venkatesan V., Iliadis I. A General Reliability Model for Data Storage Systems // Ninth Intern. Conf. on Quantitative Evaluation of Systems. London, UK, 2012. P. 209–219.
2. Erasure coding for distributed storage: an overview / S. Balaji, K. Muralee, K. Nikhil [et al.] // Science China Information Sciences. 61. 10.
3. A Study of SSD Reliability in Large Scale Enterprise Storage Deployments / S. Maneas, K. Mahdavian, T. Emami, B. Schroeder // 18th USENIX Conf. on File and Storage Technologies (FAST 20).

УДК 004.056.5

В. С. Коржук

студент кафедры безопасности информационных систем

А. В. Воронов – кандидат технических наук, доцент – научный руководитель

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ МОДЕЛЕЙ НЕЙРОННЫХ СЕТЕЙ ДЛЯ СПЕЦИАЛЬНЫХ ПРОВЕРОК ТЕХНИЧЕСКИХ СРЕДСТВ

Введение

В соответствии с Федеральным законом от 27.07.2006 г. № 149-ФЗ (ред. от 30.12.2020 г.) «Об информации, информационных технологиях и о защите информации» обладатель информации, или оператор информационной системы, в целях обеспечения защиты информации обязан оградить ее от неправомерного доступа, уничтожения, модифицирования, блокирования, копирования, а также от иных неправомерных действий в отношении такой информации. Данные требования можно выполнить, в том числе устранив каналы утечки информации или снизив уровень опасности, исходящей от них.

В современном мире практически во всех организациях очень высок уровень информатизации, а значит, имеется очень большое количество как естественных технических каналов утечки информации (ТКУИ), так и возможностей реализации новых ТКУИ злоумышленниками.

Один из основных способов обнаружения и устранения ТКУИ в средствах обработки информации – специальная проверка (спецпроверка, СП), которая проводится путем вскрытия технического средства и дальнейшего его исследования при помощи специального контрольно-измерительного оборудования.

Специальная проверка – проверка объекта информатизации (технического средства) в целях выявления и изъятия возможно внедренных закладных устройств.

Закладное устройство – средство скрытого получения информации, скрытно внедряемое в места возможного доступа к информации, в том числе в ограждение, оборудование, предметы интерьера, транспортные средства, а также в технические средства и системы обработки информации.

На одном из этапов проверки проводится рентгенография составных частей технического устройства (таких как платы, блоки питания, жесткие диски, кабели питания, аккумуляторы и т. д.), последующие дешифрование и анализ полученного снимка с целью обнаружения закладных устройств внутри элементов исследуемого технического средства.

На рис. 1, 2 приведены снимки двух плат питания. Одна плата является эталонной, вторая – модифицированной, что должно быть установлено на этапе дешифрования рентгеновского снимка. На рис. 2 области, в которых имеются модификации, обведены красным.

В случае необходимости исследования большого количества снимков или невнимательности эксперта модификации вполне могут быть им не замечены. Эта ошибка способна привести к тому, что построенная система защиты информации окажется ненадежной, так как будет иметься ТКУИ. А значит, нарушитель будет обладать возможностями несанкционированного доступа к защищаемой информации, что в свою очередь ведет не только к серьезным экономическим и репутационным потерям, но и к нарушению норм, установленных законодательством Российской Федерации.

Однако реализация вспомогательного инструмента в виде программного модуля позволит экспертам повысить качество исследований большого числа однотипных снимков, ускорит и упростит рабочий процесс и тем самым повысит производительность труда.

Основываясь на опыте крупных поисковых систем, например, Google или Яндекс, можно с уверенностью сказать, что с задачей анализа изображения должна хорошо справляться аналитическая система на базе нейронной сети (нейросети), так как в поисковых системах давно и с большим успехом нейросети используются для анализа и классификации изображений, в том числе фотографий.

Несмотря на то что в области анализа изображений, нейронные сети давно и успешно используются, при СП на этапе рентгеновского контроля данные технологии на текущий момент не нашли широкого применения, и открытая информация о подобных решениях отсутствует. Предпочтение отдается человеческому труду. По всей видимости, это вызвано узкой спецификацией алгоритмов, применяемых при решении данной задачи.

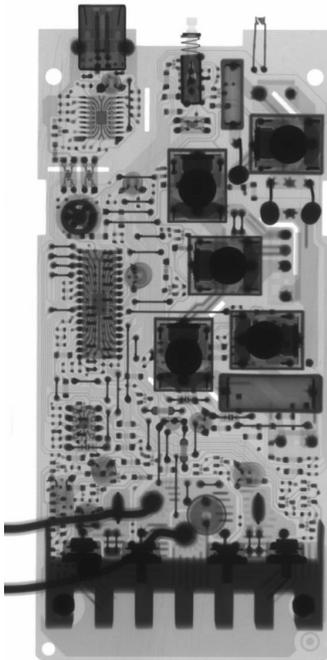


Рис. 1. Эталонная плата питания

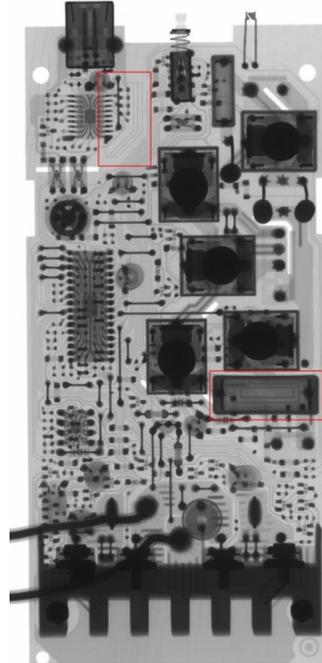


Рис. 2. Модифицированная плата питания

В связи с этим в рамках работы была поставлена задача исследовать возможности применения нейронных сетей в ходе проведения специальных проверок технических средств, определить наиболее подходящую нейросеть и реализовать ее в виде программного модуля. В данной статье будут описаны теоретическая часть проведенных исследований и их результаты.

Нейронные сети

Нейронные сети, или же искусственные нейронные сети (ИНС), названы так из-за схожести алгоритмов их работы с биологическими нейронными сетями. Изобразим схему алгоритма работы ИНС, выполняющей распознавание изображения (рис. 3).

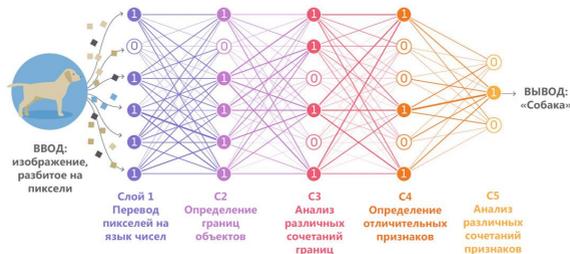


Рис. 3. Схема работы ИНС

Искусственные нейроны обозначим в виде кружков, а их входы и выходы – в виде связей со следующими нейронами. Число внутри нейрона обозначает его ответ: 0 – отрицательный, 1 – положительный [2].

Алгоритм работы ИНС

Для понимания работы нейросети взглянем на ее упрощенную схему (см. рис. 3). За каждой из связей закреплено некое число w_i , являющееся весом данной связи и характеризующее ее. Эти числа могут быть как положительными, так отрицательными, от знака зависит специфика влияния данной связи на принятие решения. Связи с положительным весом называются возбуждающими, с отрицательным – тормозящими (по названию можно понять манеру влияния на результат) [2]. Схему с весовыми коэффициентами можно видеть на рис.4.

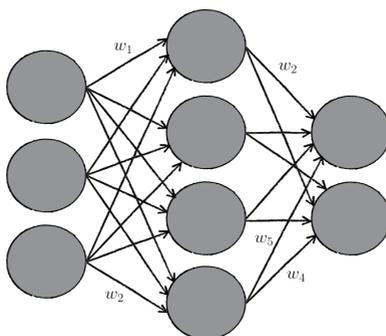


Рис. 4. Схематичное изображение нейронной сети с весом некоторых связей

В процессе прохождения сигнала по связи его значение умножается на вес связи, после чего происходит анализ сигнала внутри искусственных нейронов, принятие решения, изменение сигнала и его отправка дальше. Схема искусственного нейрона изображена на рис.5.

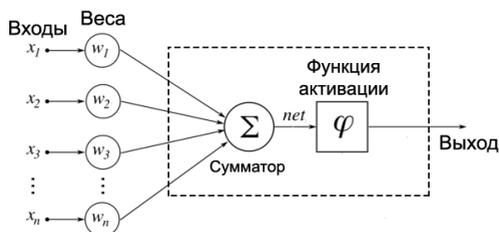


Рис. 5. Модель устройства искусственного нейрона

У каждого нейрона, в том числе у искусственного, должны быть входы, через которые он принимает сигнал. Поступившие на входы сигналы умножаются на свои веса. Сигнал первого входа x_1 умножается на соответствующий этому входу вес w_1 , сигнал второго входа x_2 умножается на вес w_2 , и так далее до последнего сигнала x_n , который умножается на вес w_n . После этого все произведения передаются в сумматор, где входные сигналы, умноженные на соответствующие веса, суммируются:

$$x_1W_1 + x_2W_2 + \dots + x_nW_n = \sum_{i=1}^n x_iW_i .$$

Роль сумматора очевидна: он агрегирует все входные сигналы в одно число – взвешенную сумму, характеризующую поступивший на нейрон сигнал в целом и отражающую степень общего возбуждения нейрона [2].

Просто так подавать взвешенную сумму на выход достаточно бессмысленно. Нейрон должен как-то обработать ее и сформировать адекватный выходной сигнал. Для этого используют функцию активации. Для разных типов искусственных нейронов используют самые разные функции активации. В общем случае их обозначают символом $\varphi(net)$ [2].

Самый универсальный вид функции активации – сигмоидальная функция. Существует целое семейство сигмоидальных функций, некоторые из которых применяют в качестве функции активации в искусственных нейронах. Но чаще всего используется логистическая функция (рис. 6) [2].

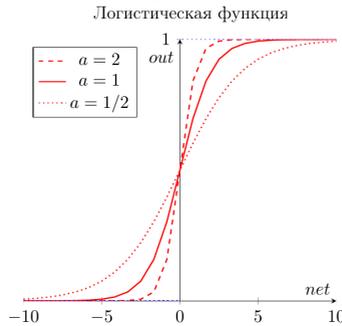


Рис. 6. График функции

Логистическая функция математически записывается следующим образом:

$$out(net) = \frac{1}{1 + \exp(-a \cdot net)},$$

где out – выход нейрона, net – взвешенная сумма, a – коэффициент, отвечающий за крутизну функции и уровень резкости изменения сигнала.

Логистическая функция обладает важными свойствами:

- является «сжимающей» функцией, т. е. вне зависимости от аргумента (взвешенной суммы) выходной сигнал всегда будет в пределах от 0 до 1;
- более гибкая, чем некоторые другие функции, – ее результатом может быть не только 0 и 1, но и любое число между ними, что придаст большую гибкость ответам нейрона;
- во всех точках она имеет производную, и эта производная может быть выражена через эту же функцию, что очень важно для возможности обучения нейросети.

Именно из-за этих свойств логистическая функция чаще всего используется в качестве функции активации в искусственных нейронах [2]. По этим же причинам ее лучше использовать в процессе построения собственной нейросети.

Разновидности нейронных сетей

Можно выделить три основных класса задач, для решения которых используют нейросети:

- распознавание образов (например, распознавание лиц, контуров объектов);

- предсказание следующего шага (например, для повышения эффективности и качества торговли на фондовых рынках);
- классификация входной информации по параметрам (например, для автоматизации принятия решения об одобрении кредита на основе входного набора разнообразных параметров) [3].

Для эффективного решения задач из того или иного класса нейросети должны обладать набором характеристик, которых можно добиться изменением внутренней архитектуры сети. По этой причине уже сейчас существует большое разнообразие ИНС как общего назначения, так и узко специализированных.

Произведем классификацию нейросетей по основным характеристикам их архитектуры в соответствии со схемой на рис. 7



Рис. 7. Схема классификации нейронных сетей

По количеству слоев, различают однослойные и многослойные нейронные сети. В однослойных сигналы со входного слоя направляются на выходной, который преобразует сигнал и сразу же выдает ответ, так как является конечным. Так как первый слой только принимает и распределяет сигналы, нужные вычисления происходят уже во втором слое. Сети такого типа легко реализовать, но они не приспособлены к решению сложных задач [3].

В многослойных нейронных сетях, помимо выходного и входного слоев, имеется несколько «скрытых» промежуточных слоев, число которых зависит от степени сложности нейронной сети. Она в большей степени напоминает структуру биологической нейросети. Сети с данной структурой обладают большими возможностями, так как в процессе обработки данных каждый промежуточный слой – это промежуточный этап, на котором осуществляются обработка и распределение информации, что приводит к более качественной обработке входных данных и позволяет анализировать информацию с учетом большого количества факторов, способных повлиять на результат [3].

Помимо количества слоев, нейронные сети можно классифицировать по способу распределения информации между нейронами, например прямого распространения и рекуррентные. В структуре сети с прямым распространением сигнала он перемещается строго по направлению от входного слоя к выходному. Движение сигнала в обратном направлении не осуществляется и в принципе невозможно. Разработанные на данный момент сети этого плана распространены широко и способны успешно решать задачи распознавания образов, прогнозирования и кластеризации [3].

В рекуррентных нейронных сетях (с обратными связями) сигнал движется и в прямом, и в обратном направлении (рис. 8). В итоге результат выхода способен возвращаться на вход. Выход нейрона определяется весовыми характеристиками и входными сигналами, плюс дополняется предыдущими выходами, вернувшись на вход. Этим нейросетям присуща функция кратковременной памяти, на основании чего сигналы восстанавливаются и дополняются во время их обработки, что позволяет успешно использовать сети данного типа при дополнении информации [3].

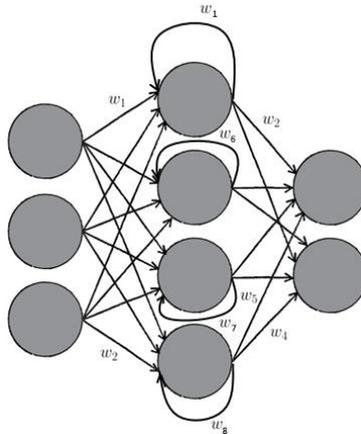


Рис. 8. Схема рекуррентной нейронной сети

Сверточные нейронные сети относятся к многослойным нейронным сетям прямого распространения, однако их архитектура сильно отличается от остальных сетей такого типа и специализирована на распознавание образов (рис. 9). Ей удастся гораздо точнее определять объекты на изображениях. При этом сверточные сети устойчивы к небольшим смещениям, изменениям масштаба и поворотам объектов на входных изображениях. Сети данного типа строятся путем чередования слоев, состоящих из блоков определенного типа. Благодаря этому их можно собирать как конструктор, добавляя и чередуя слои и получая все более мощные и эффективные архитектуры. Основными блоками сверточных нейронных сетей являются сверточные слои, слои подвыборки (пулинга), слои активации и стандартные однонаправленные слои. Сверточные слои производят фильтрацию участка изображения путем применения к нему операции математической свертки, выделяя на нем некие признаки (например, вертикальные, горизонтальные линии) соответствующие ядру свертки (фильтру). Свертка – это математический способ комбинирования двух сигналов f и g для формирования третьего, посредством взаимно-корреляционной функции $f(x)$ и $g(x)$.

Слои подвыборки меняют масштаб анализируемого изображения, ускоряя работу и позволяя следующим слоям свертки выделять более общие и масштабные элементы изображения (овалы, многогранники и т. д.). Чередование слоев позволяет выделить итоговый набор определенных признаков изображения. Анализируя эти признаки, скрытые слои сети проверяют наличие таковых в объекте, который нужно было найти на изображении, и выдают ответ [4].

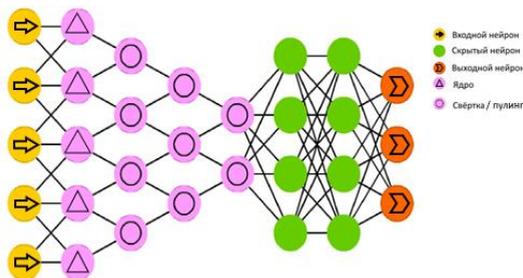


Рис. 9. Схема сверточной функции

Кроме перечисленных типов сетей, есть множество других, которые более специфичны в строении либо же являются модификациями уже перечисленных.

Обучение нейронных сетей

Обучение нейронной сети – процесс, при котором ее параметры настраиваются посредством моделирования среды, в которую эта сеть встроена. Иллюстрацию процесса обучения, можно видеть на рис. 10.

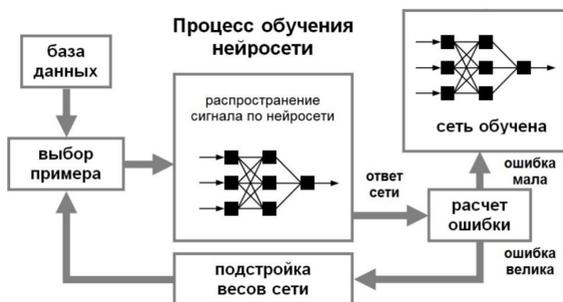


Рис. 10. Иллюстрация обучения ИНС

Различают методы обучения сетей с учителем, без него и с его частичным привлечением.

Процесс обучения с учителем представляет собой предъявление сети выборки обучающих примеров, которые состоят из входных данных, и соответствующих им верных решений. Каждый образец подается на входы сети, затем проходит обработку внутри структуры НС, вычисляется выходной сигнал сети, который сравнивается с требуемым выходом. После этого по определенному правилу вычисляется ошибка, и происходит изменение весовых коэффициентов связей внутри сети в зависимости от выбранного алгоритма. Данные операции производятся до тех пор, пока ошибка по всему обучающему массиву данных не достигнет приемлемо низкого уровня [5].

Обучение с учителем применяется для решения задач классификации, но только в том случае, когда имеется внушительный набор достоверных данных для обучения алгоритма [5].

При обучении без учителя обучающее множество состоит лишь из входных векторов. Обучающий алгоритм подстраивает веса сети так, чтобы предъявление достаточно похожих входных данных давало одинаковые выходные. Процесс обучения выделяет статистические свойства обучающего множества и группирует исходные векторы данных в классы. Предъявление на вход вектора из данного класса даст определенный выходной вектор, но до обучения невозможно предсказать, какой выход будет производиться в ответ на данный класс входных векторов. Поэтому необходимо заранее определить связь между входными данными и выводимыми, которая будет установлена сетью, в чем заключается основная сложность данного метода [5]. Обучение без учителя применяется в том случае, если размеченные данные часто ненадежны (плохо размечено соответствие входных и выходных данных), их слишком мало или слишком дорого получить. В таких случаях, предоставляя модели свободу действий для поиска зависимостей, можно получить хорошие результаты [5].

Обучение с частичным привлечением учителя (semi-supervised learning) выглядит так. Обучающий датасет содержит как размеченные, так и неразмеченные данные. Он особенно полезен в случаях, когда трудно извлечь из данных важные признаки или разметить все объекты. По сути, обучение с частичным привлечением учителя схоже с методом без учителя. Однако наличие небольшого количества размеченных обучающих данных позволяет сети намного проще находить общие черты среди неразмеченных, разделять их на классы и повысить точность результатов. Этот метод распространен для анализа медицинских изображений, таких как сканы компьютерной томографии или МРТ. Опытный

рентгенолог может разметить небольшое подмножество сканов, на которых выявлены опухоли и заболевания. Но вручную разметить все сканы – слишком трудоемкая и дорогостоящая задача [5].

Вне зависимости от метода обучение, по сути, является оптимизационным процессом, а именно поиском минимума функции ошибки путем корректирования весовых коэффициентов. И чем точнее будет определен данный минимум, тем ближе нейросеть будет к тому ответу, которого от нее ожидает человек.

Один из самых популярных алгоритмов поиска минимума функции для обучения нейронных сетей – метод обратного распространения ошибки (или метод градиентного спуска). Он основан на вычислении градиента для функции зависимости ошибки от весов нейросети и постепенном «спуске» к ее минимуму. Градиент – это вектор, указывающий направление наибольшего возрастания некоторой величины (функции) и равный по модулю скорости роста данной величины в этом направлении. Процесс происходит постепенно с последнего слоя до первого, поскольку мы можем посчитать ошибку (а значит, и градиент) только для нейронов выходного слоя, а для промежуточных этого сделать нельзя. В итоге мы вынуждены передавать полученные значения ошибки с поздних слоев на более ранние и, имея эти данные, производить перерасчет уже для них.

Таким образом, нейросети – это гибкий инструмент способный решать разнообразные задачи, однако качество и возможность их решения напрямую зависят от правильности решений, принятых на этапе проектирования сети.

Далее будет произведен аргументированный выбор модели нейронной сети, а также метод и алгоритм ее обучения, наиболее подходящий для использования в специальных проверках технических средств.

Выбор архитектуры нейросети

В данной работе перед нейронной сетью стоит задача сравнения рентгенологических снимков элементов проверяемых технических средств с эталонными и принятия решения об их соответствии, а значит, об отсутствии в проверяемом элементе посторонних элементов или конструктивных изменений.

Это все относится к классу задач по распознаванию образов. Значит, необходимо выбирать типы нейросетей, которые наиболее эффективны в этой области. К ним относятся многослойные сети прямого распространения, а также сверточные ИНС.

Многослойные сети прямого распространения намного проще в реализации и обучении по сравнению со сверточными. Но так как рентгеновские снимки часто имеют очень высокое разрешение, многослойные сети прямого распространения подходят плохо. Ведь в случае полновязности подобной сети (каждый нейрон последующего уровня получает на вход данные от каждого нейрона предыдущего), если требуется работать с цветными изображениями размером 64×64 , то каждый нейрон первого слоя сети должен будет обчислять $64 \cdot 64 \cdot 3 = 12\,288$ входных значений. А если учесть, что наша сеть должна будет распознавать изображения $3\,000 \times 2\,500$ (именно такое разрешение у снимка, изображенного на рис. 1), то входных параметров будет уже 7,5 млн. Помимо входного слоя, есть и другие, где нередко число нейронов больше, чем на входном слое, из-за чего, 7,5 млн превращаются в 55 млн и больше. Для расчета такого количества параметров потребуются большие вычислительные мощности компьютеров, либо время, затрачиваемое на обучение таких сетей, будет невообразимо большим [6].

Сверточные нейронные сети благодаря своей архитектуре позволяют избежать сильного роста количества параметров. Помимо того, что значительно уменьшается и упрощается внутренняя структура сети, это также влияет на временную эффективность, которая у СНС значительно выше, чем у полновязных многослойных сетей прямого распространения.

Обратимся к результатам испытаний, проведенных идейным прародителем сверточных нейронных сетей Яном Лекуном в группе с другими учеными в 1989 г. [7]. Результаты говорят о том, что СНС лучше приспособлены к нахождению на изображении отдельных его элементов (углы, прямые, кривые и т. д.). Также при использовании ядер свертки небольших размеров вместо соединения соседних нейронов, как в обычных сетях, они способны формировать более сложные признаки изображения, на основе более простых (например, многогранника на основе углов). Все это позволяет более точно распознавать изображения с большим количеством мелких деталей, что крайне необходимо для решения задач, стоящих перед нашей нейронной сетью.

Выбор метода и алгоритма обучения нейросети

Помимо архитектуры, также необходимо выбрать метод и алгоритм обучения нейросети. Обучение без учителя, которое используется при отсутствии хорошо размеченной обучающей базы, применять в нашем случае нет смысла. Выборку и разметку обучающих данных мы будем проводить самостоятельно и при этом можем точно отделить снимки, соответствующие эталонным от несоответствующих. Однако и метод обучения с учителем в нашем случае подходит мало. Хотя мы и можем иметь хорошо размеченные обучающие данные, но их количество, необходимое для качественной работы данного метода, собрать будет крайне тяжело. Следовательно, стоит прибегнуть к более редкому методу с частичным привлечением учителя. Он часто применяется в похожей на нашу ситуации ограниченности размеченного датасета и для нейросетей, имеющих похожие задачи – дешифрование рентгенологических снимков с целью обнаружения посторонних элементов в исследуемом объекте.

Также стоит обратить внимание непосредственно на алгоритм обучения. Метод градиентного спуска подходит, однако в нем, чтобы сделать всего одно изменение параметров (весов) сети, необходимо подать на вход сети абсолютно весь набор обучающих данных, вычислить ошибку для каждого объекта и рассчитать необходимую коррекцию коэффициентов сети (но не делать эту коррекцию), и уже после подачи всех данных рассчитать сумму в корректировке каждого коэффициента сети (сумма градиентов) и произвести коррекцию коэффициентов «на один шаг».

Описанный алгоритм будет работать крайне медленно, поэтому на практике часто производят корректировку коэффициентов сети после каждого элемента обучения, при этом значение градиента аппроксимируется. Такой метод называют стохастическим градиентным спуском или оперативным градиентным спуском [8]. По сути, реальный градиент (вычисленный из полного набора данных) заменяется его стохастической аппроксимацией (оценкой, вычисленной из случайно выбранного подмножества данных).

Стохастическая аппроксимация – рекуррентный метод построения близкой по значению к оцениваемому параметру последовательности оценок решений уравнений регрессии и экстремумов функций регрессии в задачах непараметрического оценивания. Задачи непараметрического оценивания – задачи оценивания непосредственно не наблюдаемых параметров сигналов или объектов наблюдения на основе наблюдаемых данных при том, что класс вероятностных распределений не может быть описан с помощью конечного числа параметров.

В ходе реализации нашей нейросети будем использовать конкретно алгоритм Adam. Это модификация стохастического алгоритма градиентного спуска с отдельной для каждого параметра скоростью обучения. Так увеличивается скорость обучения для параметров с редкими данными и уменьшается скорость обучения для параметров с более частыми. Подобные ситуации возникают при обработке естественных языков и распознавании образов. Такой вариант в принципе соответствует задаче, которую необходимо будет решать нашей нейросети, и повышение скорости обучения будет крайне полезно.

Корректировка выбранной архитектуры нейросети

Мы выбрали не совсем стандартный метод обучения нейросети, поэтому необходимо скорректировать ее архитектуру. Дело в том, что обучение с частичным привлечением учителя заключается в использовании генеративно-сопоставительной сети (Generative adversarial network, GAN). Представьте соревнование двух нейронных сетей, где каждая пытается перехитрить другую. При этом одна из сетей, генератор, пытается создать новые объекты данных, которые имитируют обучающую выборку. Другая сеть, дискриминатор, оценивает, являются эти сгенерированные данные реальными или поддельными. Сети взаимодействуют и циклично совершенствуются, поскольку дискриминатор старается лучше отделять подделки от оригиналов, а генератор – создавать убедительные подделки [9].

Так как нам необходимо работать с изображениями, дискриминатором будет сверточная нейронная сеть, в качестве генератора будет выступать развертывающая, которая по своей работе обратна сверточной и представляет собой ее перевернутый аналог. Благодаря этому она способна генерировать необходимые изображения. Пример схемы GAN можно видеть на рис. 11.

По результатам теоретического исследования архитектур нейронных сетей, можно сделать вывод, что для использования в специальных проверках технических средств, лучше использовать гене-

ративно-адаптивную нейросеть, в основе которой сверточная и развертывающая нейросети. Их обучение следует производить по методу частичного привлечения учителя при помощи любого из алгоритмов стохастического градиентного спуска. Однако алгоритм Adam позволит сократить время обучения.

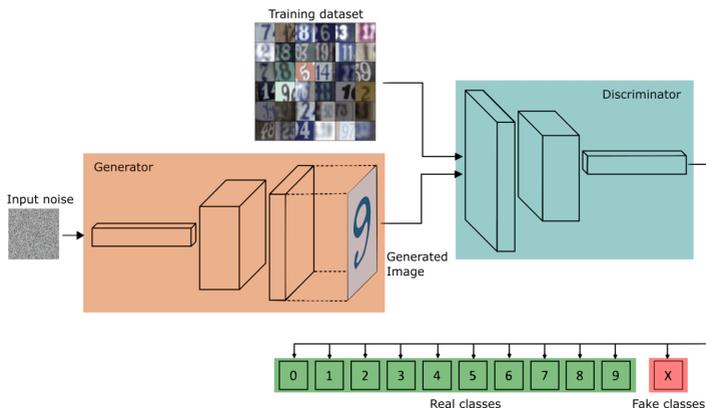


Рис. 11. Пример схемы генеративно-адаптивной нейросети

Помимо теоретических исследований, в дальнейшем будут проведены реализации и практические исследования выбранных типов нейросетей, в ходе которых проверена правильность сделанного выбора и того, насколько нейронная сеть с выбранным алгоритмом обучения и архитектурой подходит для применения в специальных проверках технических средств.

Библиографический список

1. Специальная проверка техники (СП). URL: <https://www.apsecurity.ru/services/special-inspections.html> (дата обращения: 18.03.2021).
2. Радько П. Глава 3. Основы ИНС. URL: <https://neuralnet.info/chapter/основы-инс/#Выводы> (дата обращения: 18.03.2021).
3. Павленко А. Типы нейронных сетей. Принцип их работы и сфера применения. URL: <https://otus.ru/nest/post/1263/> (дата обращения: 22.03.2021).
4. Gradient-based learning applied to document recognition / Y. LeCun, L. Bottou, Y. Bengio, P. Huffer // Proceedings of the IEEE. 1998. Nov. P. 2278–2324.
5. Обучение нейронной сети. URL: <https://neuronus.com/theory/nn/238-obucheniya-nejronnoi-seti.html> (дата обращения: 22.03.2021).
6. Что такое нейронные сети и как они работают. Классификация искусственных нейросетей. URL: <https://cryptowiki.ru/news/chto-takoe-nejronnye-seti-i-kak-oni-robotaut-klassifikaciia-iskusstvennyh-neirosetei.html> (дата обращения: 25.03.2021).
7. Backpropagation applied to handwritten zip code recognition / Y. LeCun, B. Boser, J. S. Denker [et al.] // Neural Computation/ 1989. Vol. 1, Iss. 4. P. 1541–551.
8. Калинин Д. Все о градиентном спуске или как учится нейронная сеть. URL: <https://smartiqa.ru/blog/neural-network-gradient-descent> (дата обращения: 19.04.2021).
9. Беликова К. Обучение нейросети с учителем, без учителя, с подкреплением. URL: <https://neurohive.io/ru/osnovy-data-science/obuchenie-s-uchitelem-bez-uchitelja-s-podkrepleniem/> (дата обращения: 21.04.2021).

УДК 004

В. О. Лебедев

студент кафедры безопасности информационных систем

А. М. Тюрликов – доктор технических наук, профессор – научный руководитель

РАЗРАБОТКА ПРИЛОЖЕНИЯ ПОД ОПЕРАЦИОННУЮ СИСТЕМУ ANDROID ДЛЯ ИЗМЕРЕНИЯ СКОРОСТИ РЕАКЦИИ ЧЕЛОВЕКА

Время реакции (ВР) – основной поведенческий параметр в экспериментальной психофизиологии. Оценивает время от стимула до реакции (обычно момента начала). Делится на три фазы:

- время прохождения нервных импульсов от рецептора до коры головного мозга;
- время, необходимое для обработки нервных импульсов и получения ответной реакции в центральной нервной системе;
- время ответа организма.

Время реакции зависит от вида сигнала-раздражителя, интенсивности, настроенности на принятие сигнала, возраста и сложности реакции индивида.

Скорость реакции имеет важное значение и серьезно влияет на жизнь человека. Хорошая скорость позволяет людям быть гибкими и эффективными при ответе на различные стимулы и ситуации: в разговоре, за рулем, при занятиях спортом и т. д. Адекватная скорость реакции дает преимущества, однако при этом нужно корректно обрабатывать информацию: если во время интервью человеку задают вопрос, ожидается, что он ответит быстро и правильно. Когда человек сталкивается с непредвиденными ситуациями, управляя автомобилем или занимаясь в спортзале, также недостаточно действовать быстро или хорошо, нужно делать это одновременно и эффективно.

Скорость реакции зависит от нескольких факторов.

- Восприятие: возможность уверенно видеть, слышать или чувствовать стимул имеет важнейшее значение для быстрой реакции. В легкой атлетике, когда судья дает стартовый сигнал, звук достигает ушей спортсменов (они воспринимают стимул).

- Обработка информации: для своевременной реакции необходимо сконцентрироваться и хорошо понять информацию. В приведенном примере спортсмены, услышав стартовый сигнал, различают его в общем звуковом потоке и понимают, что можно стартовать (обрабатывают стимул).

- Отклик: развитая моторика необходима, чтобы своевременно реагировать на стимул. Когда спортсмены получили и успешно обработали сигнал, они начинают забег (реагируют на стимул).

Если какой-либо из процессов нарушен, это негативно повлияет на скорость реакции.

Один из примеров проблем со скоростью реакции человека – болезнь Паркинсона (БП). Это заболевание нервной системы, в результате которого поражаются нейронные клетки головного мозга. Она имеет прогрессирующий характер и с медицинской точки зрения проявляется различными двигательными нарушениями (слабая двигательная активность, ограниченная подвижность мышц в состоянии покоя, тремор конечностей) [1]. Современная медицина не может вылечить БП, и даже не имеет стандартных методов – терапия подбирается индивидуально в соответствии с жалобами пациента.

В качестве перспективного путей решения проблемы рассматривается лекарственная терапия, направленная на приостановление и уменьшение нейродегенеративных процессов в nigrostriарных нейронах. В настоящее время разрабатывается ряд новых противопаркинсонических средств, ориентированных не столько на симптоматический эффект, сколько на патогенетические основы болезни [2].

Развитие терапии болезни Паркинсона связано с рациональной комбинацией симптоматических средств с препаратами, воздействующими на различные звенья патогенеза. Неотъемлемая часть этого развития – совершенствование методов диагностики и анализа симптомов БП. Одним из таких способов может стать мобильное приложение в смартфоне. Благодаря имеющимся в мобильном устройстве датчикам, можно считать и анализировать достаточное количество информации об активности и моторике человека. Компания Apple выпустила ResearchKit – открытую программную платформу для медицинских исследований, которая помогает врачам и ученым собирать данные от участников исследований через приложения под управлением операционной системы iOS на iPhone [3].

Как отмечает Journal of Parkinson's Disease, мониторинг симптомов имеет большой потенциал в прецизионной медицине и может способствовать обучению и помощи в жизни пациента с заболеванием Паркинсона. Для развития этой области требуется дополнительная работа, необходимо уделять особое внимание времени отслеживания симптомов проявления заболевания [4].

Приложение Parkinson mPower, разработанное компанией Sage Bionetworks и Университетом Рочестера, помогает людям с болезнью Паркинсона отслеживать свои симптомы, выполняя различные действия с iPhone. В их числе игра на запоминание, касание пальцем, речь и ходьба. Результаты опросов пациента и сведения с датчиков его телефона объединяются с данными множества других пациентов, помогая проводить исследование болезни Паркинсона. Используя гироскоп и другие функции iPhone, медицинские работники могут оценивать ловкость, равновесие, походку и память участников, что расширяет их знания о болезни, включая факторы, облегчающие и ухудшающие состояние больных, такие как сон, физическая активность и настроение.

Для устройств под управлением операционной системы Android на данный момент практически не существует подобных приложений. С целью решения этой задачи такое мобильное приложение было создано. Цель работы – реализация приложения для обеспечения возможности сбора и анализа данных о моторике людей с болезнью Паркинсона. Приложение является альтернативой настоящему физическому устройству, поскольку это дешевле, чем делать реальное устройство.

Рассмотрим один из стандартных сценариев использования мобильного устройства, а именно использование мобильного приложения. На рис. 1 схематично описан процесс взаимодействия человека с мобильным устройством при работе в мобильном приложении.

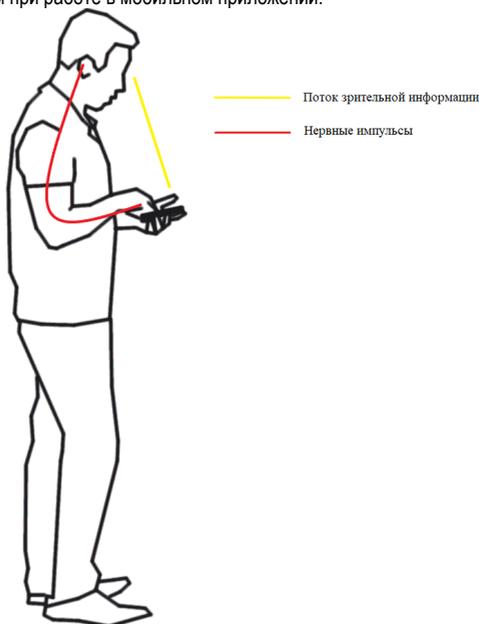


Рис. 1. Схема процесса взаимодействия человека со смартфоном

Во время работы приложение ожидает от пользователя определенного набора действий, таких как: нажатие пальцем на экран, удерживание пальца на экране, смахивание и т. д. Все эти действия требуют определенной точности, например набор сообщения требует точной последовательности символов путем нажатия на иконку соответствующего символа.

Рассмотрим процесс подробнее.

1. На экране устройства имеется страница для выполнения определенного набора действий пользователя.
2. Зрительная информация поступает в мозг человека.
3. Человек анализирует полученную информацию и принимает решение об ответном действии.
4. Генерируется некий «алгоритм» для руки человека для выполнения данного действия.
5. Алгоритм передается на руку в виде нервных импульсов.
6. Совершается заданное действие.

При возникновении проблем на одном из этапов возникает неправильное действие, фиксируется ошибка [5].

В приложении для анализа моторики человека для устройств под управлением операционной системы Android (рис. 2) смоделировано движение шара по наклонной плоскости и перемещение его по экрану на определенное число позиций за единицу времени путем вычисления ускорения шара по осям X и Y при отклонении смартфона от горизонтального положения.

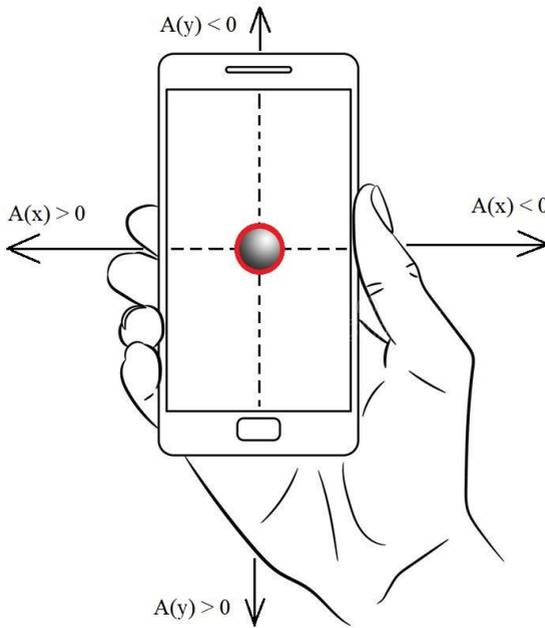


Рис. 2. Схема игрового поля приложения

Алгоритм движения был основан на формулах равноускоренного движения.

$$X_0 = 0, V_0 = 0,$$

$$X_i = X_{i-1} + V_{i-1}\Delta t_{i-1} + \frac{a_{i-1}\Delta t_{i-1}^2}{2}, V_i = V_{i-1} + a_{i-1}\Delta t_{i-1},$$

где X_j – координата шара; V_j – скорость шара; a_{j-1} – ускорение шара в момент времени i ; Δt_j – время между измерениями ускорения.

В результате работы приложения были получены три вида графиков, описывающих движения человека в процессе измерения.

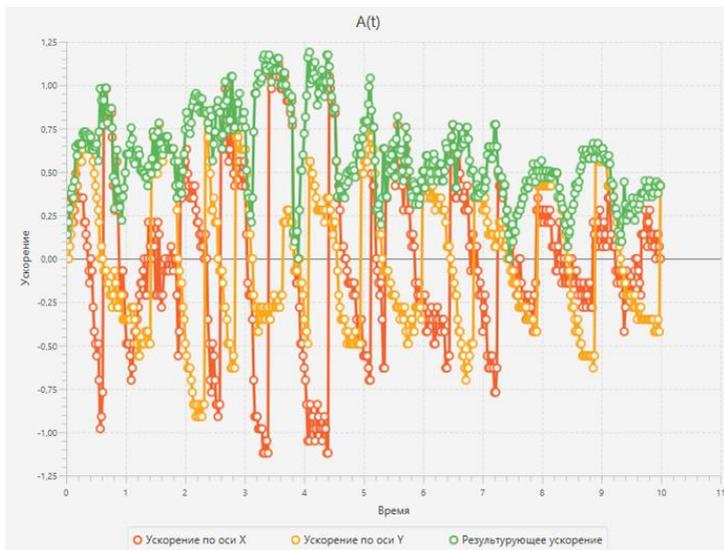


Рис. 3. График зависимости ускорения шара от времени

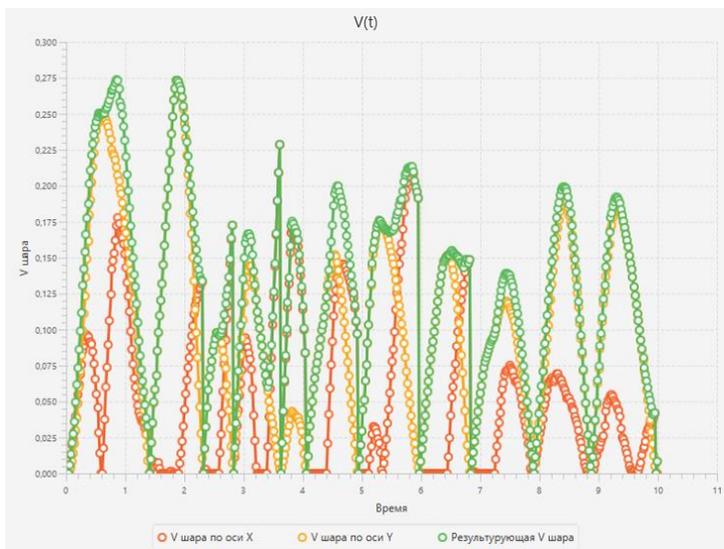


Рис. 4. График зависимости скорости шара от времени

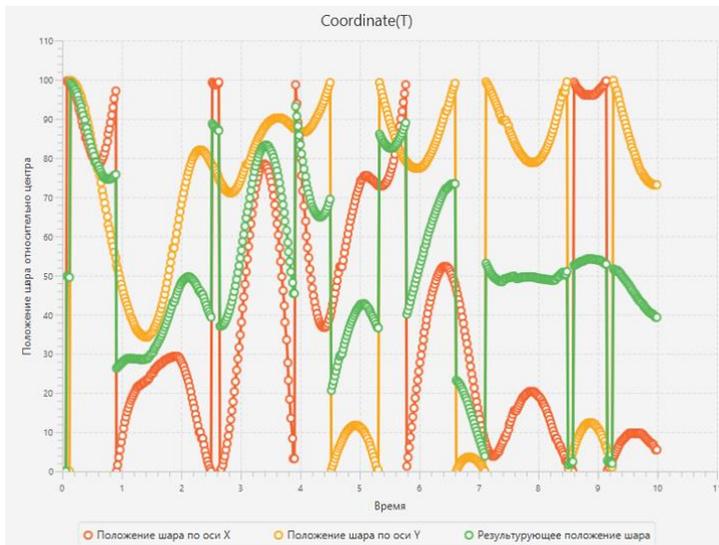


Рис. 5. График зависимости положения шара на игровом поле относительно центра от времени

Таким образом, было реализовано приложение для сбора и анализа информации о моторике человека, а также построены три вида графиков для дальнейшего анализа, на каждом из которых значения изображены относительно двух осей X и Y, а также представлено результирующее значение по двум осям.

По графику зависимости ускорения шара от времени (см. рис. 3) и графику зависимости скорости шара от времени (см. рис. 4) можно судить о плавности движений человека. По графику зависимости положения шара на игровом поле относительно центра шара от времени можно сделать вывод об успешности проводимого измерения.

Цель дальнейшей работы – создание более удобного интерфейса, позволяющего использовать приложение в практической деятельности.

Библиографический список

1. Левин О. С., Федорова Н. В. Болезнь Паркинсона. М.: МЕДпрессинформ, 2012. 315 с.
2. Нодель М. Р. Болезнь Паркинсона. Современные возможности терапии: метод. рек. / подгот. под ред. Н. Н. Яхно. М., 2008. 40 с.
3. Research Kit & Care Kit. URL: <https://www.apple.com/ru/researchkit/> (дата обращения: 12.02.2021).
4. Press Precision Medicine in Parkinson's Disease – Exploring Patient-Initiated Self-Tracking // J. of Parkinson's Disease. 2018. № 8. Р. 441–446.
5. Борисовская А. В., Тюрликов А. М. Систематизированный обзор методов оценки двигательной активности человека с использованием сенсоров мобильных устройств // Обработка, передача и защита информации в компьютерных системах: I Всерос. науч. конф., СПб., 14–22 апр. 2020 г. СПб., 2020. С. 227–233.

УДК 004.728.3.057.4

В. С. Марковский, А. А. Фоминых

магистранты кафедры безопасности информационных систем

А. М. Тюрликов – доктор технических наук, профессор – научный руководитель**ОРГАНИЗАЦИЯ A2G СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ СОТОВОЙ СВЯЗИ**

Работа выполнена при финансовой поддержке Министерства науки и высшего образования Российской Федерации, соглашение № FSRF-2020-0004, «Научные основы построения архитектур и систем связи бортовых информационно-вычислительных комплексов нового поколения для авиационных, космических систем и беспилотных транспортных средств».

В связи с популяризацией смартфонов и ростом потребности к доступу в Интернет сформировался глобальный спрос на повсеместную беспроводную связь. В то время как мобильная широкополосная связь все более распространена на суше, в полете из-за низкой скорости передачи и производительности для большого числа пассажиров на борту самолета она остается ограниченной и ее качество часто воспринимается крайне низким.

В работе [1] описываются требования к системе «воздух – земля» (air-to-ground, A2G). Один из способов достижения требуемых характеристик – использование принципов сотовой связи для установления прямого соединения между самолетом и наземными базовыми станциями. Использование сетей пятого поколения (5G, New Ratio, NR) позволяет формировать луч, достигая соединения с высокой пропускной способностью, высокой скорости передачи данных и надежной радиолинией.

Широкополосная (broadband) A2G связь имеет довольно простую концепцию [2]: базовые станции на земле соединены с лучеобразующей антенной, расположенной на нижней части фюзеляжа самолета, обеспечивающую высокую скорость передачи данных, пока самолет летит на высоте до 12 км. Затем сигнал распространяется в самолете через точку доступа Wi-Fi.

В ряде работ [2–4] предлагается построить систему на основе технологии сотовой связи, однако не приведены алгоритмы, с помощью которых были достигнуты результаты, исходный код закрыт. Другие недостатки – скорость работы и стоимость программ для моделирования. Во-первых, программы учитывают различные тонкие параметры и условия, что, с одной стороны, обеспечивает более точную оценку работы системы, с другой – замедляет скорость. Во-вторых, финансирование разработки программного обеспечения впоследствии должно окупаться, вследствие чего программы для моделирования оказываются платными.



Рис. 1. Схема A2G системы

В данной работе предложен алгоритм моделирования A2G системы, необходимый для исследования особенностей систем A2G, определения и реализации параметров для улучшения системы, и оценка системы с помощью кумулятивной функции распределения (cumulative distribution function, CDF) в зависимости от скорости передачи данных и соотношения «сигнал – шум» (signal-to-noise ratio, SINR). Результаты моделирования сравниваются с результатами существующих работ, посвященных исследованию использования сетей 5G в связи «воздух – земля».

Будем считать, что самолет движется по заранее установленному маршруту со скоростью, меньшей скорости звука, и представляет собой абонента в сотовой сети, на борту самолета находится только одно абонентское устройство. Наземные базовые станции играют роль сотовых вышек с антеннами, наклоненными к небу. Структура такой системы приведена на рис. 1.

Определим параметры, задающие систему (табл. 1).

Таблица 1

Значения параметров

Параметр	Значение
Высота полета	12 км
Расстояние между базовыми станциями, ISD	80, 160 км
Спектр (полоса) частот и несущая частота f	30 МГц, 700 МГц
Мощность передачи наземной станции w и самолета w_a	0,2 В, 10 В
Коэффициенты усиления приемной и передающей антенн, диаграмма направленности каждого антенного элемента соответствует стандарту 3GPP TR 38.901	-17 дБ, -34 дБ

Рассмотрим алгоритм для построения кумулятивной функции распределения для A2G системы.

1. Вычисляются размеры поля и число самолетов. Самолет определяется как точка со случайными координатами в гексагональной сетке. Высота соты вычисляется как

$$h = \frac{ISD}{2},$$

где ISD (inter-site distances) – расстояние между базовыми станциями. Сторона соты (правильного шестиугольника) вычисляется по следующей формуле:

$$a = \frac{2h}{\sqrt{3}}.$$

2. Вычисляется мощность шума: $P_s = w$.

$$P_N(\text{дБм}) = -174 + 10\log_{10}(P_s \cdot 10^6),$$

$$P_N(\text{Вт}) = \frac{10^{P_N(\text{дБм})/10}}{1000}.$$

3. Расстояние до ближайшей базовой станции вычисляется как гипотенуза прямоугольного треугольника, в котором катетами являются высота полета и евклидово расстояние (перпендикуляр) от базовой станции (рис. 2).

$$d_{ij} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2}.$$

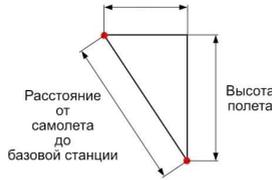


Рис. 2. Определение расстояния от самолета до базовой станции

4. Для вычисления затухания предлагаются различные варианты [9, 10]. В рамках текущей работы затухание (path loss, PL) вычисляется по формуле согласно стандарту [8]:

$$PL = 20\log_{10}(d) + 20\log_{10}(f) + 32.44 + \mathcal{N}(0, \sigma) - G_{Tx} - G_{Rx},$$

где d – расстояние между антеннами; f – несущая частота; G_{Tx} – коэффициент усиления передающей антенны [5]; G_{Rx} – коэффициент усиления приемной антенны [5]; $\mathcal{N}(0, \sigma)$ – гауссовская случайная величина с нулевым средним и дисперсией ($\sigma = 9.29 - 1.58 \log_{10}(f/1\ 000)$) [4]).

В случае $RU > 0$ затухание вычисляется для каждого самолета.

5. Вычисляется мощность принятого сигнала: $P_R = \frac{P_S}{PL}$.

6. Вычисляется SINR: $SINR = \frac{P_R}{P_N(\text{Вт})}$.

7. Вычисляется скорость передачи данных по формуле Шеннона: $C = B \log\left(1 + \frac{P_S}{P_N}\right)$.

Полученное значение скорости сохраняется в массиве.

8. Пункты 1–6 повторяются N раз, где N – количество самолетов, которое, согласно исследованию о плотности самолетов в воздушном пространстве, задано в работе [3] как

$$N = \frac{60 \cdot S}{18000},$$

где S – площадь поля, км². По полученным данным строятся эмпирические интегральные функции распределения для SINR и пропускной способности.

Рассмотрим результаты моделирования для низкочастотного (low bandwidth) спектра частот.

Мощность передачи наземных станций и самолета составляет 80 и 0,2 Вт соответственно. Система использует полосу 30 МГц и несущую частоту 700 МГц. Самолеты летят на высоте 12 км и базовые станции располагаются на расстоянии 80 км друг от друга (рис. 3).

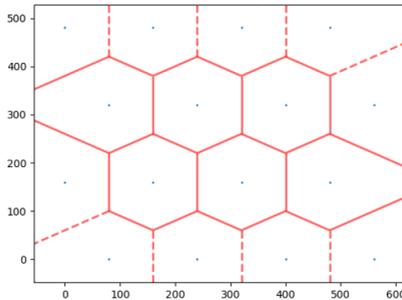


Рис. 3. Расположение базовых станций

Уровень использования ресурсов (resource utilization, RU) может указывать на уровень помех в сети: чем он выше, тем больше помехи внутри канала. Также сравним результаты моделирования с графиками из работы [4], для этого были подобраны параметры, указанные в табл. 2.

Таблица 2

Значения параметров моделирования (RU = 0 %)

ISD	$G_{Tx} + G_{Rx}$ (RU = 0 %)	$G_{Tx} + G_{Rx}$ (RU = 100 %)
80	-33	+15
160	-17	+7

На рис. 4, 5 приведены графики зависимости кумулятивной функции плотности (cumulative density function, CDF) от отношения уровня сигнала к уровню шума (signal interference to noise ratio, SINR) при отсутствии мешающих абонентов для нисходящего канала. Уровень использования ресурсов равен 0 % и при условии, что все абоненты мешающие, 100 % для ISD = 80 и ISD = 160 соответственно. Для проверки адекватности модели сравнение осуществляется с кривыми из статьи [4]. Значение SINR становится хуже по мере увеличения уровня использования ресурсов.

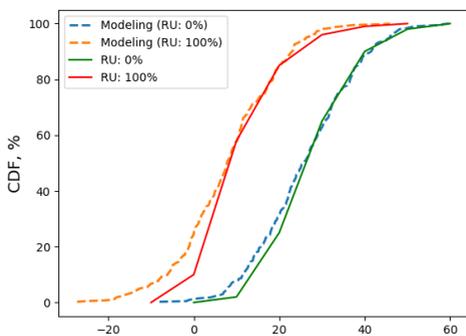


Рис. 4. SINR, 80 км

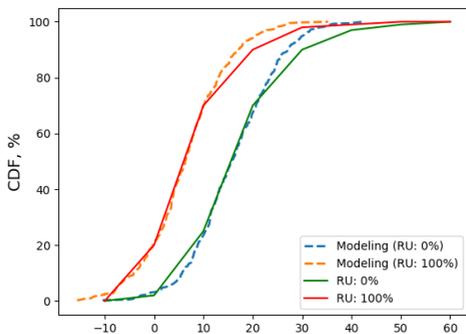


Рис. 5. SINR, 160 км

На рис. 6 приведен график зависимости кумулятивной функции плотности от пропускной способности. По результатам видно, что значение пропускной способности становится хуже по мере увеличения уровня использования ресурсов.

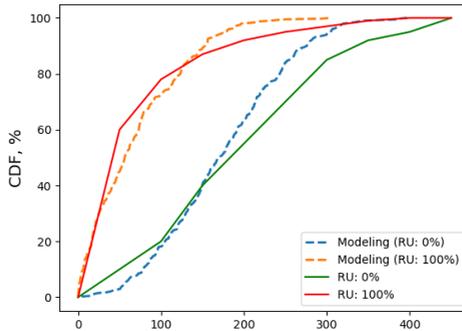


Рис. 6. Пропуская способность, 80 км

Предложены алгоритм моделирования A2G системы, необходимый для исследования особенностей систем «воздух – земля», определения и реализации параметров для улучшения, и оценка системы с помощью кумулятивной функции распределения в зависимости от скорости передачи данных и соотношения «сигнал – шум». Результаты показали, что полученная модель позволяет получать результаты совпадающие с результатами работы [4], при условии соответствующего выбора параметров.

Библиографический список

1. Broadband Direct-Air-to-Ground Communications // ECC Report 2014. May 30.
2. Liu L. Performance evaluation of direct air-to-ground communication using new radio (5G). 2017.
3. Dinc E. Total Cost of Ownership Optimization for Direct Air-to-Ground System Design. 2020.
4. Lin X. et al. Sky High 5G: New Radio for Air-to-Ground Communications. 2020.
5. Broadband Direct-Air-to-Ground Communications System employing beamforming antennas // Recommendation ETSI-TR 101 599, v1.1.3. 2020. Sept.
6. Martiradonna S. et al. Understanding the 5G-air-simulator: A tutorial on design criteria, technical components, and reference use cases // Computer Networks. 2020. T. 177. P. 107314.
7. Garro E. et al. 5G mixed mode: NR multicast-broadcast services // IEEE Transactions on broadcasting. 2020. T. 66, №. 2. С. 390–403.
8. 3GPP TS 38.306 – NR; User Equipment (UE) radio access capabilities.
9. Rodriguez I. et al. Propagation Path Loss Models for 5G Urban Micro-and Macro-Cellular Scenarios.
10. Prasad A. et al. Challenges for enabling virtual reality broadcast using 5G small cell network // 2018 IEEE Wireless Communications and Networking Conf. Workshops (WCNCW). IEEE, 2018. P. 220–225.

УДК 004.93.14

М. А. Мельников

студент кафедры безопасности информационных систем

А. В. Афанасьева – старший преподаватель – научный руководитель

АНАЛИЗ ВЛИЯНИЯ ФАКТОРОВ ОКРУЖЕНИЯ НА БИОМЕТРИЧЕСКУЮ АУТЕНТИФИКАЦИЮ ПО ЛИЦУ

Введение

В настоящее время благодаря значительному развитию алгоритмов и увеличению доступности необходимого оборудования все большую популярность набирают системы аутентификации, основанные на анализе биометрических характеристик пользователя. Неоспоримое преимущество таких систем в том, что биометрические характеристики невозможно намеренно передать, потерять или украсть. Данные свойства делают их практически идеальными для использования в качестве аутентификатора, поскольку надежность системы аутентификации напрямую зависит от возможности попадания аутентификатора в руки злоумышленника.

Среди всех форм биометрических систем технология распознавания лиц одна из главных и стоит в одном ряду с распознаванием по радужной оболочке и сканированию отпечатков пальцев. Согласно исследованиям аналитиков компании J'son&Partners Consulting [1], 50% долю на российском рынке биометрии составляют именно технологии идентификации и аутентификации по лицу. Данная технология подразделяется на двумерное (2D) и трехмерное (3D) распознавание.

Использование биометрических характеристик в качестве аутентификатора не лишено недостатков. Главная проблема заключается в том, что предъявляемые пользователем биометрические данные никогда не будут с абсолютной точностью совпадать с эталоном из базы данных. Это связано с тем, что в процессе извлечения свойств из биометрического образца они подвергаются искажению и наложению шума. Более того, искажающий эффект способны внести факторы окружающей среды во время сканирования биометрических данных пользователя, однако исследованиям в данном направлении уделяется довольно мало внимания.

Невозможность точного совпадения биометрического аутентификатора с эталоном может привести к ошибкам при распознавании. Так как результатом аутентификации является бинарное решение («да» или «нет»), то любая биометрическая система распознавания может характеризоваться двумя параметрами: FAR (false accept rate) – вероятность ложного допуска и FRR (false reject rate) – вероятность ложного отказа (ошибка второго рода). В случае системы контроля доступа FAR проявляется в предоставлении доступа человеку, для которого доступ не предусмотрен, в то время как FRR будет означать, что система ошибочно отказала в доступе авторизованной персоне.

Постановка задачи

Биометрические системы идентификации и аутентификации – неотъемлемая часть многих сфер нашей жизни, вопросы их надежности и работоспособности напрямую определяют безопасность конфиденциальных данных. Кроме качества работы алгоритмов и технических устройств, на точность биометрической аутентификации могут оказывать влияние неподконтрольные человеку факторы окружающей среды. На текущее время эта тема практически неисследована.

Цель данной работы – анализ влияния факторов окружения на точность методов биометрической аутентификации по лицу. В качестве исследуемого метода аутентификации будет использоваться двумерное распознавание по лицу как наиболее распространенное и доступное решение в области биометрии. В ходе работы будут оценены изменения параметров FAR и FRR биометрической системы аутентификации при воздействии различных факторов окружения, выраженных в искажении обрабатываемых двумерных изображений.

Модель внешних факторов

Важнейшими этапами процесса аутентификации по лицу служат обнаружение лица на изображении и вычисление уникальных лицевых характеристик. Методы их реализации во многом опираются на анализ яркостного контраста пикселей, составляющих изображение лица. Вследствие этого в данной работе будут рассматриваться такие факторы окружающей среды, воздействие которых может сильно повлиять на интенсивность пикселей лицевого изображения, а именно:

- изменение (увеличение и уменьшение) освещенности;
- снижающие видимость погодные условия (дождь, снег);
- потеря фокуса видеокамеры на лице;
- расстояние между сканирующим устройством и лицом.

Изменение освещенности во время фото- и видеосъемки можно промоделировать простым изменением яркости изображения: уменьшение яркости для моделирования недостаточной освещенности, и увеличение – для избыточной. В ходе данной работы уровень освещенности будет оцениваться как средняя интенсивность пикселей, составляющих лицевое изображение.

Моделирование изменения освещенности будет выполняться со значениями средней интенсивности, указанными в табл. 1, 2.

Таблица 1

Значения интенсивности при увеличении освещенности

№	Средняя интенсивность
1	100
2	115
3	130
4	145
5	165

Таблица 2

Значения интенсивности при уменьшении освещенности

№	Средняя интенсивность
1	100
2	90
3	75
4	65

Моделирование таких внешних факторов, как снижающие видимость погодные условия и потеря фокуса, может быть выполнено путем применения популярных при решении задач аугментации обучающей выборки инструментов – гауссовского шума и гауссовского фильтра (гауссовского размытия) [2]. В нашем случае гауссовский шум будет использоваться для моделирования погодных условий, а гауссовский фильтр – для моделирования потери фокуса. Для дальнейшего исследования были выбраны степени искажения, представленные в табл. 3, 4.

Таблица 3

Значения параметров гауссовского шума

№	Уровень шума, %
1	–
2	5
3	10
4	15
5	20
6	25

Таблица 4

Значения параметров гауссовского фильтра

№	Радиус размытия, px
1	–
2	1
3	2
4	3
5	4p
6	5

Для того чтобы промоделировать увеличение расстояния между сканирующим устройством и лицом человека, достаточно уменьшить масштаб изображения лица. В ходе дальнейшей работы масштаб изображений будет уменьшаться в процентном соотношении от исходного размера согласно значениям, приведенным в табл. 5.

Таблица 5

Значения масштаба изображения

№	Масштаб, %
1	100
2	85
3	70
4	55
5	40

Экспериментальный стенд

Программный модуль `authentication_system.py`, моделирующий биометрическую систему аутентификации (рис. 1), был реализован на основе нейросетевого модуля распознавания лиц `Face-Recognition-OpenCV` от `pyimagesearch`, разработанного на языке Python [3]. Он предусматривает работу с библиотеками алгоритмов компьютерного зрения и обработки изображений `dlib` и `OpenCV` и состоит из двух основных компонентов:

- `encode_faces.py` – подпрограмма, формирующая базу данных системы распознавания;
- `recognize_faces.py` – подпрограмма, выполняющая весь цикл распознавания лиц на изображении.

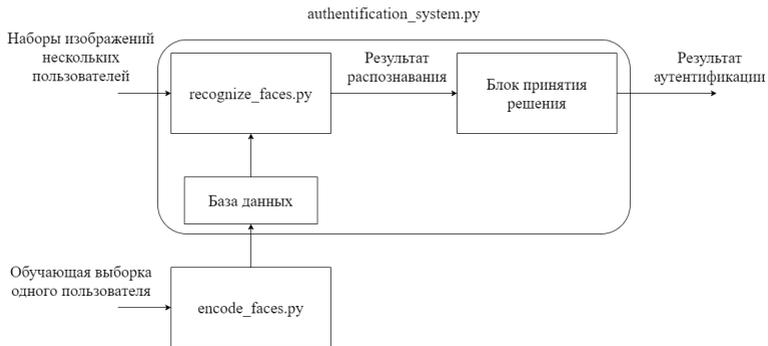


Рис. 1. Модель системы биометрической аутентификации

В качестве исходных данных для эксперимента был сформирован набор двумерных изображений, принадлежащих восьми разным людям, выполняющим роль пользователей биометрической системы. Набор изображений по каждому отдельному пользователю подразделяется на обучающую выборку, состоящую из восьми разных изображений лица пользователя в положении анфас, и тестовую выборку, состоящую из 22 фотографий пользователя, сделанных в различных условиях. Для каждого изображения из всех тестовых выборок было создано по 22 копии, которые в дальнейшем подверглись искажениям, соответствующим модели внешних факторов. Редактирование изображений выполнялось с помощью встроенных инструментов графического редактора `Adobe Photoshop`.

Моделирование процесса аутентификации выполнялось следующим образом.

- Обучающая выборка одного пользователя (легального) загружается в базу данных с помощью подмодуля `encode_faces.py`.
- Искаженная определенным образом совокупность тестовых выборок всех пользователей передается в систему аутентификации через подмодуль `authentication_system.py`.

Данная последовательность действий выполнялась при эталонных данных от каждого из восьми пользователей системы для каждого типа и степени искажений тестовых выборок. На каждую степень искажения пришлось 1 456 попыток аутентификации. Вывод о результате аутентификации принимался следующим образом.

- Корректный допуск – если анализируемое изображение принадлежит легальному пользователю и результат аутентификации «да».
- Корректный отказ – если анализируемое изображение не принадлежит легальному пользователю и результат аутентификации «нет».

- Ложный допуск (FAR) – если анализируемое изображение не принадлежит легальному пользователю и результат аутентификации «допуск».
- Ложный отказ (FRR) – если анализируемое изображение принадлежит легальному пользователю и результат аутентификации «отказ».

На основании результатов всех попыток аутентификации вычислялись значения параметров FAR и FRR биометрической системы при каждом типе и степени искажения данных по следующим формулам:

$$FAR = \frac{\sum_{i=1}^U n_i^{(FAR)}}{N}; \quad (4.1)$$

$$FRR = \frac{\sum_{i=1}^U n_i^{(FRR)}}{N}, \quad (4.2)$$

где U – количество пользователей, $n_i^{(FAR)}$ и $n_i^{(FRR)}$ – число ошибок FAR и FRR при попытках аутентификации пользователя i , N – общее число попыток аутентификации при конкретном типе и степени искажения.

Полученные результаты

Результаты моделирования представлены в табл. 6–10.

Таблица 6

Параметры системы при увеличении освещенности

Средняя интенсивность	FAR	FRR
100	0,014 42	0
115	0,013 73	0
130	0,013 04	0
145	0,012 36	0,000 68
165	0,016 48	0,000 68

Таблица 7

Параметры системы при уменьшении освещенности

Средняя интенсивность	FAR	FRR
100	0,014 42	0
90	0,013 04	0,001 36
75	0,015 10	0,000 68
65	0,017 17	0,000 68

Таблица 8

Параметры системы при снижающих видимость погодных условиях

Уровень шума, %	FAR	FRR
–	0,014 42	0
5	0,017 17	0,000 68
10	0,035 71	0,002 74

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ЗАЩИТА ИНФОРМАЦИИ

15	0,041 89	0,007 55
20	0,057 00	0,028 15
25	0,056 31	0,059 75

Таблица 9

Параметры системы при потере фокуса

Радиус размытия, px	FAR	FRR
–	0,014 42	0
1	0,010 30	0,001 38
2	0,006 86	0,000 68
3	0,009 61	0,003 43
4	0,014 42	0,006 86
5	0,019 23	0,015 10

Таблица 10

Параметры системы при увеличении расстояния между сканером и лицом

Масштаб изображения, %	FAR	FRR
100	0,014 42	0
85	0,012 36	0
70	0,016 48	0,000 68
55	0,013 04	0
40	0,010 98	0
25	0,008 92	0,000 68

Библиографический список

1. J'son & Partners Consulting Отчет об исследовании Российского рынка биометрических технологий, 2018–2022 гг. URL: https://json.tv/en/ict_telecom_analytics_view/research-of-the-russian-market-of-biometric-technologies-2018-2022 (дата обращения: 13.03.2021).

2. Гайер А.В. Аугментация обучающей выборки «на лету» для обучения нейронных сетей // Тр. ИСА РАН. Спецвыпуск. 2018. URL: <http://www.isa.ru/proceedings/images/documents/2018-S/150-157.pdf> (дата обращения: 15.05.2021).

3. Pyimageresearch: сайт. URL: <https://www.pyimageresearch.com> (дата обращения: 08.02.2021).

УДК 004.777

В. С. Налегач

студент кафедры информационных систем и защиты информации

В. Г. Ерышов – кандидат технических наук – научный руководитель

АНАЛИЗ ЭФФЕКТИВНОСТИ СИСТЕМЫ ПРЕДОТВРАЩЕНИЯ УТЕЧЕК ДАННЫХ В КОРПОРАТИВНОЙ СЕТИ НА ОСНОВЕ DLP-СИСТЕМЫ INFOWATCH TRAFFIC MONITOR

Необходимость наличия защиты от внутренних угроз была очевидна на всех этапах развития средств информационной безопасности. Однако изначально внешние угрозы считались более опасными. В последнее время стали обращать больше внимания на внутренние угрозы, из-за этого популярность DLP-систем возросла. Потребность в их использовании стала упоминаться в стандартах и нормативных документах (например, раздел «12.5.4 Утечка информации» в стандарте ГОСТ ISO/IEC 17799–2005). При этом специализированные технические средства для защиты от внутренних угроз стали массово выпускаться после 2000 г.

Утечки информации – добыча недобросовестным образом данных, не предназначенных для публичного доступа и представляющих ценность для государства в целом либо отдельных его субъектов, органов и структур или юридических и физических лиц (для последних особенно актуальна на сегодня утечка персональных данных). В настоящее время самые большие объемы информации хранятся юридическими и физическими лицами на компьютерах и иных электронных носителях. Поэтому вполне логично, что ее утечка наиболее часто происходит именно с ПК. Для получения информации используются специальные программы-вирусы. Получив информацию о логинах и паролях, а также данные банковских карточек и счетов, злоумышленники без особого труда похищают денежные средства, фирменные секреты и идеи компаний, становятся обладателями коммерческих тайн, а также сведений, которые в дальнейшем можно использовать для дискредитации и шантажа [1].

Часто предотвратить утечку информации можно, используя лишь технические средства, такие как DLP-системы, исключающие утечки данных через компьютерную сеть. Но, как показывает практика, многие руководители предприятий проявляют свой интерес к защите информации лишь после того, как произошла утечка данных.

Утечки информации могут быть преднамеренными и непреднамеренными. Преднамеренным является выход конфиденциальных сведений за пределы предприятия, когда лицо, получившее доступ к этим сведениям, передает их третьей стороне, при этом отдавая себе отчет о возможных негативных последствиях как для предприятия, на котором оно трудится, так и для отдельных должностных лиц этого предприятия или организаций, с ним каким-либо образом связанных. Более того, как правило, такое лицо заблаговременно предупреждается руководством о недопустимости подобных действий и ответственности, которую оно несет, тем не менее совершает их. Рост подобных случаев объясняется появлением новых технических средств передачи данных, в том числе в цифровом виде. Из года в год увеличивается количество сервисов, размещенных на онлайн-платформах, а также иных хранилищ данных и каналов их передачи. Средой распространения сведений все чаще становится Интернет. Источники преднамеренной передачи данных следующие: украденные носители информации, передача данных от сотрудников компании, доступ к сведениям извне.

Достаточно распространена утечка в результате преднамеренного похищения электронных устройств (ноутбука, планшета, смартфона). Также похитители могут заинтересовать съемные носители информации: флэш-карты, жесткие компьютерные диски.

Эксклюзивная информация от сотрудников уходит на сторону двумя путями. Первый – информация, которой поделился работник с третьими лицами, была получена им легально: в силу своих служебных обязанностей он имел к ней доступ. Второй – сотрудник, причастный к утечкам данных, получил их неправомысленным образом и сумел передать за пределы предприятия. Но все чаще специалисты в сфере информационной безопасности отмечают, что для съема информации применяются вредоносные программы. Утечка данных из-за использования вредоносного ПО все более распространена. Информацию путем копирования похищают в корыстных целях. Используя вредоносное ПО, хакеры могут даже корректировать данные и удалять их. Дистанционно устанавливая вредоносное ПО, зло-

умышленники могут легко управлять ПК, получив доступ ко всей хранящейся на нем информации, включая пароли к банковским карточкам и счетам. Разумеется, коммерческий сектор намного чаще становится жертвой атак, нежели государственный.

Непреднамеренные утечки информации, как правило, происходят вследствие утери носителя данных (планшета, смартфона, ноутбука, флэш-карты и т. д.). При этом наиболее часто теряются самые небольшие по размерам – маленькие флэш- и SD-карты. Также выходу сведений вонне способствуют неумышленные действия работников предприятий: они ошибочно размещают конфиденциальные данные в Интернете или, например, записывают пароль на листке бумаги, и к нему легко могут получить доступ лица, для которых подобные сведения не предназначены. Человеческий фактор играет не последнюю роль в утечках конфиденциальной информации.

Для защиты данных от посторонних лиц используются DLP-системы, предотвращающие утечку из информационной системы за ее пределы. В данной работе речь пойдет о DLP-системе, продукте компании InfoWatch – InfoWatch Traffic Monitor. Она предотвращает утечки конфиденциальной информации на основе полноценного контентного анализа информационных потоков [2].

InfoWatch Traffic Monitor надежно работает под большими нагрузками на сотнях тысяч рабочих мест не только в режиме мониторинга, но и блокировки. Чтобы минимизировать проблему ложных срабатываний, традиционную для всех DLP, InfoWatch Traffic Monitor сделал ставку на развитие технологий анализа контента. Система «ловит» сложные текстовые и графические объекты даже в случае, если нарушитель сумел значительно видоизменить их и ухитрился замаскировать свои действия. Благодаря многомерному анализу содержания InfoWatch Traffic Monitor понимает, о какой информации идет речь. Это облегчает сотруднику службы информационной безопасности работу с ложноположительными срабатываниями. Работает по всем видам каналов утечки информации, контролирует как стандартные каналы, так и уникальные. Среди них: корпоративная и веб-почта, мессенджеры, облачные хранилища, сетевые папки, FTP, терминальные соединения, локальные и сетевые принтеры, съемные носители [3].

Функции InfoWatch Traffic Monitor следующие [4]. Перехват SMTP, IMAP и POP3-трафика. Возможен перехват трафика (или копии трафика), передаваемого через почтовый relay-сервер; перехват копии трафика, проходящего через управляемый коммутатор. Перехват HTTP- и HTTPS-трафика. Возможен перехват трафика, передаваемого через прокси-сервер, поддерживающий ICAP-протокол; перехват копии трафика, проходящего через управляемый коммутатор. Перехват копии ICQ-трафика (протокол OSCAR), проходящего через управляемый коммутатор. При подключении ICQ через HTTP. Система перехватывает ICQ-трафик аналогично HTTP-трафику. Проверка файлов, находящихся в корпоративной сети (открытых сетевых папок, локальных дисков рабочих станций и файлового хранилища SharePoint 2007/2010/2013), с помощью подсистемы Crawler. Анализ Skype-трафика, теневого копий файлов и заданий на печать, фотографий, передачи трафика по протоколам HTTP, HTTPS и FTP, загрузки данных в облачные хранилища по протоколу HTTPS, приема и передачи электронных писем по протоколам SMTP, POP3, IMAP, Outlook, контроль обмена данными через Jabber (протокол XMPP), WhatsApp (протокол WhatsApp), Telegram (мобильная версия, протоколы Telegram и MProto), Viber (протокол Viber). Также поддерживается перехват голосового трафика в Skype. Перехват перечисленных данных осуществляется системой InfoWatch Device Monitor. Анализ содержимого перехваченного трафика с целью выявления нарушений корпоративной политики безопасности. Фильтрация перехваченного трафика путем выдачи разрешения/запрещения на доставку определенных данных.

Решаемые задачи: предотвращение утечек данных в момент их отправки без приостановки бизнес-процессов, сокращение финансовых и репутационных рисков, связанных с предотвращением утечек и борьбой с их последствиями.

Для развертывания макета были использованы следующие элементы (рис. 1).

DC – контроллер домена с именем master-demo-dc.demo.lab, для реализации используется служба-диспетчер dns в составе os windows server 2012 R2, на нем настроены службы каталогов под названием Active Directory, где создано подразделение с пользователями и администраторами, настроен адрес на интерфейс eth0 – 192.168.111.2 [5].

Почтовый сервер mailserver.demo.lab с операционной системой ubuntu-18.04.3-live-server-amd64, туда установлен iRedMail 1.2.1, настроены конфигурационные файлы агентов передачи и полу-

чения почты – postfix dovecot, настроена LDAP-синхронизация с Active Directory, налажен доступ к веб-интерфейсу roundcube, настроен интерфейс eth0 – 192.168.111.20.

Сервер IWTM версии 6, установленный с помощью образа диска IWTM 6.10 с ос linux red hat enterprise server, на нем настроены интерфейсы eth 0 -192.168.111.5 и eth1 – 192.168.111.111, где были настроены конфигурационные файлы postfix и tm для связи с почтовым сервером mailserver.demo.lab, настроена LDAP-синхронизация с Active Directory [6].

Виртуальный коммутатор с операционной системой 4MLinux версии 30 реализован на основе openwrt для возможности настройки IWTM в режиме работы span-копия, настроен интерфейс eth0 – 192.168.111.222.

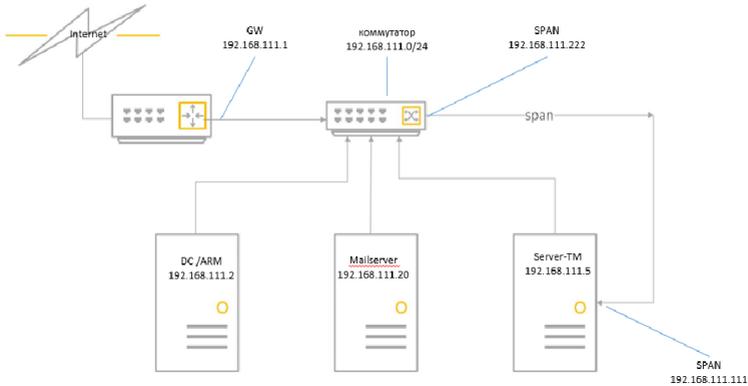


Рис. 1. L2 схема макета

На рис. 2 представлен результат работы системы DLP-системы IWTM 6, после того как на почтовом сервере был сгенерирован трафик почтовых сообщений, содержащих нарушения, на которые были настроены политики Traffic Monitor. Этот результат показывает, что DLP-система контролирует SMTP-трафик [7, 8].

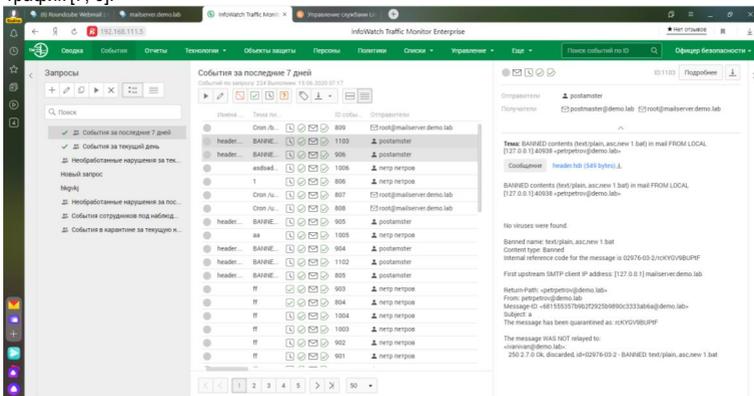


Рис. 2. Результат работы системы DLP-системы

Исходя из результатов работы DLP системы, можно сделать вывод о том, что данное средство предотвращения утечек является эффективным инструментом предотвращения угроз и анализа трафика.

Библиографический список

1. Утечка персональных данных: последствия. Защита персональных данных с помощью DLP-системы. URL: <https://searchinform.ru/> (дата обращения 24.02.2021).
2. Руководство администратора InfoWatch Traffic Monitor 6.9. URL: <https://docplayer.ru/29408422-Infowatch-traffic-monitor-6-1-rukovodstvo-administratora.html> (дата обращения 24.02.2021).
3. Руководство по установке InfoWatch Traffic Monitor 6.9. URL: http://testks.ru/man/iw/IWTM_6_10_InstallGuide_RU.pdf (дата обращения 24.02.2021).
4. DLP-система InfoWatch Traffic Monitor. URL: <https://www.infowatch.ru/> (дата обращения 24.02.2021).
5. Документация iRedMail. URL: <https://docs.iredmail.org/index.html> (дата обращения 24.02.2021).
6. Минимальный почтовый сервер на основе Postfix и Dovecot. Часть 1: Dovecot. URL: <https://habr.com/ru/post/258279/> (дата обращения 24.02.2021).
7. Руководство по iptables. URL: <https://www.opennet.ru/docs/RUS/iptables/> (дата обращения 24.02.2021).
8. Современные технологии обнаружения утечек. URL: <https://www.anti-malware.ru/node/8578#part6> (дата обращения 24.02.2021).

УДК 004.056.5

П. П. Недошивин

студент кафедры безопасности информационных систем

А. А. Овчинников – кандидат технических наук, доцент – научный руководитель**СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ ПОСТКВАНТОВЫХ КРИПТОСИСТЕМ**

В последние годы становится все больше исследований в области разработки квантового компьютера, способного выполнять алгоритмы решения различных математических задач. Еще в конце прошлого века были описаны способы быстрого выполнения разложения большого простого числа на множители (алгоритм Шора), вычисления дискретного логарифма. Это ставит под угрозу множество систем по всему миру, использующих известные криптографические схемы с открытым ключом. В криптографии появилась новая отрасль: постквантовая криптография, задача которой состоит в создании криптосистем и систем электронно-цифровой подписи (ЭЦП), способных выдержать атаки с помощью квантовых вычислительных систем.

В 2017 г. организация NIST, занимающаяся в том числе стандартизацией в области криптографии, объявила конкурс на принятие стандарта по постквантовым криптосистемам и электронно-цифровым подписям. К декабрю 2020 г. уже прошли три раунда отбора, в которых было отсеяно множество предложений. Задачи, на которых предполагалось основывать новые алгоритмы, разнообразны: задачи в криптографических решетках, суперсингулярные эллиптические кривые, кодовые задачи. До финала дошли только три системы на решетках и одна кодовая.

В отличие от многих теоретико-числовых задач, задача декодирования алгебраического кода оказалась более стойкой к квантовым атакам. Используя определенные коды (например, коды Гоппы) и параметры, можно получить криптосистему постквантового периода (время атак на квантовых вычислителях превышает полиномиальное время). За основу взята система МакЭлиса, разработанная еще в 1978 г. Предложенный вариант назван Classic McEliece. Основные параметры алгоритма задают размерность используемого линейного кода: n – длина кода, t – корректирующая способность (число исправляемых декодером ошибок). К третьему раунду число конфигураций увеличилось до 5: минимальная длина кода 3 488, максимальная – 8 192, корректирующая способность – соответственно 64 и 128 [1].

Также представлены три криптосистемы, основанные на NP-полных задачах в криптографических решетках. Один из вариантов – NTRU уже достаточно известен, был разработан еще в 1996 г. Данная криптосистема основана на задаче поиска кратчайшего вектора решетки (SVP). Предлагаются три варианта системы версии hps (версия, предложенная Хоффстайном, Пайфером и Сильверманом), которые различаются основными числами n и q . Они должны быть взаимно простыми, при этом n – простое число. Эти числа напрямую влияют на размеры закрытого и открытого ключей. Варианты: $n = 509, 677, 821$; $q = 2\,048, 2\,048$ и $4\,096$ соответственно.

Еще одна система, основанная на решетках, Kyber использует задачу обучения с ошибками в решетках модулей (MLWE problem: learning-with-errors in module lattices) [2]. Основные параметры: n , k и q , где n определяет полином кольца многочленов $\mathbb{Z}[X]/(X^n + 1)$ и всегда равен 256, q определяет еще одно кольцо $\mathbb{Z}_q[X]/(X^n + 1)$ и всегда равно 3 329, k определяет длину массивов байтов и может быть равен 512, 768 и 1 024.

Последняя оставшаяся на третьем раунде конкурса система – SABER. Ее безопасность основывается на задаче модульного обучения с округлением (Mod-LWR problem), что представляет собой некоторое сокращение задачи обучения с ошибками (LWE), предложенной в системе Kyber. Варианты системы называются LightSaber, Saber и FireSaber и отличаются уровнями безопасности. Параметры криптосистемы Saber следующие: n является степенью кольца полиномов и равно 256, l – ранг модуля, который определяет размерность решетки $l \cdot n$, и равно 2, 3 или 4 в зависимости от требуемого уровня безопасности. Модули, используемые в схеме, являются степенями 2: q , r и T ; q и r неизменны, а оптимальное значение T возрастает с увеличением l , так как необходимо сохранить корректность выполнения преобразований в алгоритме [3, 4].

В табл. 1 представлены основные параметры постквантовых криптосистем. В основном это результаты по вариантам систем с сопоставимыми размерами открытого и закрытого ключей. Вариант McEliece с длиной кода 3 488 является минимальным по длине кода из предложенных в постквантовый стандарт. Системы, основанные на решетках, брались с максимальными предложенными параметрами. Рассматриваются три операции: генерация ключей, шифрование/инкапсуляция, расшифрование/декапсуляция. Результаты приводятся в количестве циклов ЦП. Столбец Intel Haswell предоставлен авторами криптосистем (Intel Xeon E3-1275 v3 с тактовой частотой 4,5 ГГц), а столбец Intel Kaby Lake (Intel Core i5-7200U с тактовой частотой 3 ГГц) – автором данной статьи. Размеры ключей и шифротекста представлены в байтах [4].

Таблица 1

Сравнение времени выполнения постквантовых криптосистем

Система	Размер открытого ключа	Размер закрытого ключа	Размер шифротекста	Операция	Intel Haswell	Intel Kaby Lake
mceliece348864	261 120	6 492	128	keypair	58034411	87051616
				enc	44350	66525
				dec	134745	202117
Kyber1024	1 568	3 168	1 568	gen	307148	460722
				enc	346648	519972
				dec	396584	594876
ntruhs4096821	1 230	1 592	1 230	gen	22511180	33766770
				enc	1566922	2350383
				dec	4237744	6356616
FireSaber	1 312	3 040	1 472	Keygen	126220	189330
				Encapsulation	153832	230748
				Decapsulation	155700	233550

Как видно из таблицы, наиболее оптимальна по затраченному времени система FireSaber. Инкапсуляция и декапсуляция незначительно превышают шифрование и расшифрование системы МакЭлиса, но генерация ключевой пары занимает гораздо меньше циклов центрального процессора, а значит, и времени. FireSaber – самый безопасный вариант постквантовой криптосистемы Saber, однако его параметры можно уменьшить, незначительно меняя сложность решения NP-полной задачи.

В результате сравнения постквантовых криптосистем с использованием как предоставленных авторами в документации результатов выполнения, так и выполнения реализации на обычном процессоре персонального компьютера, можно прийти к выводу, что по памяти кодовая система МакЭлиса однозначно проигрывает своим конкурентам из-за высокого размера открытого ключа. Она является лучшей по времени шифрования и расшифрования, но довольно сильно проигрывает по времени генерации пары ключей. Лидером по времени генерации является выступает Saber, при этом не сильно проигрывая по времени выполнения других операций.

Библиографический список

1. Albrecht M. R., Bernstein D. J. et. al. Classic McEliece: conservative code-based cryptography. 2020.
2. Avanzi R., Bos J. et. al. CRYSTALS-KYBER: Algorithm Specifications And Supporting Documentation. 2020.
3. Chen C., Danba O. et. al. NTRU: Algorithm Specifications and Supporting Documentation. 2020.
4. Basso A., Mera J. M. B. et. al. SABER: Mod-LWR based KEM (Round 3 Submission). 2020.

УДК 004.056.53

Е. В. Перфильев

студент кафедры безопасности информационных систем

В. Г. Ерышов – кандидат технических наук, доцент – научный руководитель

КЛАССИФИКАЦИЯ УГРОЗ БЕЗОПАСНОСТИ ИНФОРМАЦИИ ПО ТЕХНИЧЕСКИМ КАНАЛАМ УТЕЧКИ ИНФОРМАЦИИ

Объектами угроз безопасности информации (БИ) в организациях могут выступать сведения о структуре, составе, состоянии и деятельности объекта информатизации (ОИ), персонала, информационных, материальных и финансовых ценностей и активов. Угрозы БИ проявляются в нарушении целостности, конфиденциальности и доступности информации, составляющей сведения конфиденциального характера. Источниками угроз БИ могут выступать конкуренты, преступники, коррупционеры, которые преследуют следующие основные цели: ознакомление с охраняемыми сведениями, кража, модификация, уничтожение для нанесения прямого материального ущерба организации. Неправомерное овладение конфиденциальной информацией (КИ) возможно за счет ее разглашения источниками сведений, за счет утечки информации через технические средства обработки КИ и за счет несанкционированного доступа к КИ.

Источники КИ – персонал, документы, публикации, технические носители информации, технические средства обработки информации, информационные системы, средства обеспечения производственной и трудовой деятельности, а также продукция и отходы производства.

Одним из способов получения злоумышленником КИ является установка на ОИ специальных технических средств (СТС) негласного добывания КИ (радиомикрофонов, телефонных радиоретрансляторов и других подслушивающих устройств), в том числе на подвижных ОИ, таких как автомобили должностных лиц, аппаратные связи и управления.

Основными элементами описания угроз утечки информации по техническим каналам утечки информации (ТКУИ) выступают источник угрозы (технические средства разведки), среда (путь) распространения информативного сигнала и носитель защищаемой информации (источник сигнала) (рис. 1).

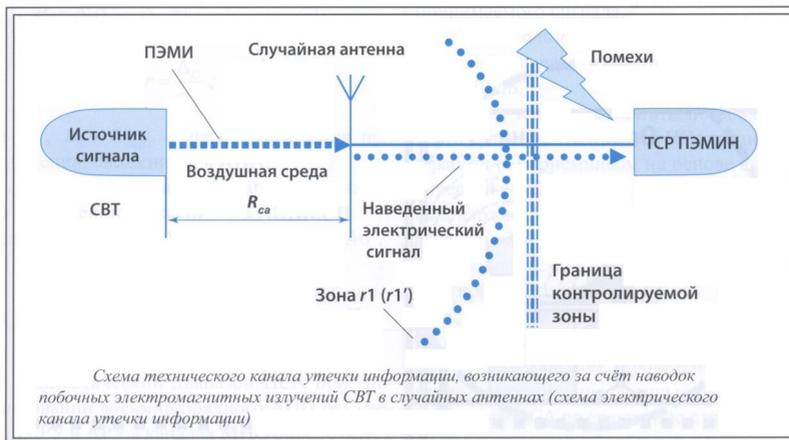


Рис. 1. Схема технического канала утечки информации

Источниками угроз утечки информации по техническим каналам являются физические лица, не имеющие доступа к АС, а также зарубежные спецслужбы или организации (в том числе конкуриру-

щие или террористические), криминальные группировки, осуществляющие перехват (съем) информации с использованием технических средств ее регистрации, приема или фотографирования.

Среда распространения информативного сигнала – это физическая среда, по которой информативный сигнал может распространяться и приниматься (регистрироваться) приемником. Она может быть как однородной (например, только воздушной), так и неоднородной за счет перехода сигнала из одной среды в другую (например, в результате акустозлектрических или виброакустических преобразований).

Носителем конфиденциальной информации (КИ) выступают пользователь АС, осуществляющий голосовой ввод КИ в АС, акустическая система АС, воспроизводящая КИ, а также технические средства АС и ВТСС, создающие физические поля, в которых информация находит отражение в виде символов, образов, сигналов, технических решений и процессов, количественных характеристик физических величин.

При обработке КИ в АС за счет реализации технических каналов утечки информации возможно возникновение следующих УБКИ:

угроз утечки акустической (речевой) информации;

угроз утечки видовой информации;

угроз утечки информации по каналам побочных электромагнитных излучений и наводок (ПЭМИН).

1. Угрозы утечки акустической (речевой) информации

Возникновение угроз утечки акустической (речевой) информации, содержащейся непосредственно в произносимой речи пользователя АС, при обработке КИ в АС обусловлено наличием функций голосового ввода КИ в АС или функций воспроизведения КИ акустическими средствами АС [1].

Перехват акустической (речевой) информации в данных случаях возможен с использованием аппаратуры, регистрирующей акустические (в воздухе) и виброакустические (в упругих средах) волны, а также электромагнитные (в том числе оптические) излучения и электрические сигналы, модулированные информативным акустическим сигналом, возникающие за счет преобразований в технических средствах обработки КИ, ВТСС и строительных конструкциях и инженерно-технических коммуникациях под воздействием акустических волн.

Кроме этого, перехват акустической (речевой) информации возможен с использованием специальных электронных устройств съема речевой информации, внедренных в технические средства обработки КИ, ВТСС и помещения или подключенных к каналам связи.

Угрозы безопасности КИ, связанные с перехватом акустической информации с использованием специальных электронных устройств съема речевой информации («закладочных устройств»), определяются в соответствии с нормативными документами Федеральной службы безопасности Российской Федерации в установленном ею порядке.

Перехват акустической (речевой) информации может вестись:

– стационарной аппаратурой, размещаемой в близлежащих строениях (зданиях) с неконтролируемым пребыванием посторонних лиц;

– портативной возимой аппаратурой, размещаемой в транспортных средствах, осуществляющих движение вблизи служебных помещений или при их парковке рядом с этими помещениями;

– портативной носимой аппаратурой – физическими лицами при их неконтролируемом пребывании в служебных помещениях или в непосредственной близости от них;

– автономной автоматической аппаратурой, скрытно устанавливаемой физическими лицами непосредственно в служебных помещениях или в непосредственной близости от них.

2. Угрозы утечки видовой информации

Угрозы утечки видовой информации реализуются за счет просмотра КИ с помощью оптических (оптикоэлектронных) средств с экранов дисплеев и других средств отображения средств вычислительной техники, информационно-вычислительных комплексов, технических средств обработки графической, видео- и буквенно-цифровой информации, входящих в состав АС [2]. Кроме этого, просмотр (регистрация) КИ возможен с использованием специальных электронных устройств съема, внедренных в служебных

помещениях или скрытно используемых физическими лицами при посещении ими служебных помещений.

Угрозы безопасности КИ, связанные с их перехватом при использовании специальных электронных устройств съема видовой информации (видеозакладок), определяются в соответствии с нормативными документами Федеральной службы безопасности Российской Федерации в установленном ею порядке.

Необходимым условием осуществления просмотра (регистрации) КИ является наличие прямой видимости между средством наблюдения и носителем КИ.

Перехват КИ может вестись:

- стационарной аппаратурой, размещаемой в близлежащих строениях (зданиях) с неконтролируемым пребыванием посторонних лиц;

- портативной возимой аппаратурой, размещаемой в транспортных средствах, осуществляющих движение вблизи служебных помещений или при их парковке рядом с этими помещениями;

- портативной носимой аппаратурой – физическими лицами при их неконтролируемом пребывании в служебных помещениях или в непосредственной близости от них.

Перехват (просмотр) КИ может осуществляться посторонними лицами путем их непосредственного наблюдения в служебных помещениях либо с расстояния прямой видимости из-за пределов АС с использованием оптических (оптикоэлектронных) средств.

3. Угрозы утечки информации по каналам побочных электромагнитных излучений и наводок

Возникновение угрозы КИ по каналам ПЭМИН возможно за счет перехвата техническими средствами побочных (не связанных с прямым функциональным значением элементов АС) информативных электромагнитных полей и электрических сигналов, возникающих при обработке ПД техническими средствами АС [3].

Генерация информации, содержащей КИ и циркулирующей в технических средствах АС в виде электрических информативных сигналов, обработка и передача указанных сигналов в электрических цепях технических средств АС сопровождаются побочными электромагнитными излучениями, которые могут распространяться за пределы служебных помещений в зависимости от мощности излучений и размеров АС.

Регистрация ПЭМИН осуществляется с целью перехвата информации, циркулирующей в технических средствах, обрабатывающих КИ (в средствах вычислительной техники, информационно-вычислительных комплексах и сетях, средствах и системах передачи, приема и обработки КИ, в том числе в средствах и системах звукозаписи, звукоусиления, звуковоспроизведения, переговорных и телевизионных устройствах, средствах изготовления, тиражирования документов и других технических средствах обработки речевой, графической, видео- и буквенно-цифровой информации).

Для регистрации ПЭМИН используется аппаратура в составе радиоприемных устройств и оконечных устройств восстановления информации. Кроме этого, перехват ПЭМИН возможен с использованием электронных устройств перехвата информации, подключенных к каналам связи или техническим средствам обработки КИ.

Регистрация ПЭМИН может вестись с использованием аппаратуры следующих видов:

- стационарной аппаратуры, размещаемой в близлежащих строениях (зданиях) с неконтролируемым пребыванием посторонних лиц;

- портативной возимой аппаратуры, размещаемой в транспортных средствах, осуществляющих движение вблизи служебных помещений или при их парковке рядом с этими помещениями;

- портативной носимой аппаратуры – физическими лицами в непосредственной близости от АС;

- автономной автоматической аппаратуры, скрытно устанавливаемой физическими лицами в непосредственной близости от АС.

Каналы утечки информации, обусловленные наводками, образуются за счет соединительных линий технических средств АС и ВТСС и посторонних проводников (в том числе цепей электропитания и заземления) [4]. Наводки электромагнитных излучений технических средств АС возникают при излуче-

нии элементами технических средств АС информативных сигналов при наличии емкостной, индуктивной или гальванической связей соединительных линий технических средств АС, линий ВТСС и посторонних проводников. В результате на случайных антеннах (цепях ВТСС или посторонних проводниках) наводятся информативный сигнал.

Прохождение информативных сигналов в цепи электропитания возможно при наличии емкостной, индуктивной или гальванической связи источника информативных сигналов в составе технических средств АС и цепей питания.

Прохождение информативных сигналов в цепи заземления обусловлено наличием емкостной, индуктивной или гальванической связи источника информативных сигналов в составе аппаратуры ТСПИ и цепей заземления.

Для съема информации с проводных линий могут использоваться:

– средства съема сигналов, содержащих защищаемую информацию, с цепей технических средств АС и ВТСС, линий связи и передачи данных, выходящих за пределы служебных помещений (эквиваленты сети, токовые трансформаторы, пробники);

– средства съема наведенных информативных сигналов с цепей электропитания;

– средства съема наведенных информативных сигналов с шин заземления;

– средства съема наведенных информативных сигналов с проводящих инженерных коммуникаций.

Для волоконно-оптической системы передачи данных угрозой утечки информации выступает утечка оптического излучения, содержащего защищаемую информацию, с боковой поверхности оптического волокна.

Появление новых каналов связи – сотовой связи, пейджинговых сообщений, спутниковых и беспроводных сетей передачи данных – привело к развитию специализированных систем и средств контроля и перехвата информации, ориентированных на используемые в них информационные технологии, в том числе средств:

– перехвата пейджинговых сообщений и сотовой связи;

– перехвата информации в каналах передачи данных вычислительных сетей.

Выводы

В статье рассмотрена классификация угроз безопасности информации по техническим каналам утечки информации, учет которых позволит в дальнейшем построить модель угроз БИ, а также адекватную угрозам БИ систему технической защиты информации.

Библиографический список

1. ГОСТ Р 51275–2006. Защита информации. Объект информатизации. Факторы, воздействующие на информацию. Понятие объекта информатизации. URL: <https://docs.cntd.ru/document/1200057516> (дата обращения: 16.03.2021).

2. ГОСТ Р 50922–2006. Защита информации. Основные термины и определения. URL: <https://docs.cntd.ru/document/1200058320> (дата обращения: 16.03.2021).

3. Хорев А. А. Защита информации от утечки по техническим каналам. Часть 1. Технические каналы утечки информации: учеб. пособие. М.: Гостехкомиссия России, 1998. 320 с.

4. Ярочкин В. И. Технические каналы утечки информации. М.: ИППКР, 1994. 102 с.

УДК 004.056.53

Е. В. Перфильев

студент кафедры безопасности информационных систем

В. Г. Ерышов – кандидат технических наук, доцент – научный руководитель

КЛАССИФИКАЦИЯ ИСТОЧНИКОВ УГРОЗ БЕЗОПАСНОСТИ ИНФОРМАЦИИ

Известно, что одним из важнейших направлений защиты информации в автоматизированных системах (АС) является защита информации (ЗИ) от несанкционированного доступа (НСД) к ней. Данное направление предусматривает как организационные, так и программно-технические методы защиты, реализуемые с помощью специального программного обеспечения – систем ЗИ от НСД в АС, а также наличие обученных и квалифицированных специалистов по ЗИ и подразделений по ЗИ в организациях.

Для оценки эффективности системы ЗИ от НСД в АС актуальна задача оценки и классификации источников угроз безопасности информации (БИ) с целью выявления их характеристик и дальнейшего учета при построении адекватной системы ЗИ ОИ организации.

Источниками угроз НСД в АС могут быть (рис. 1):

- нарушитель;
- носитель вредоносной программы;
- аппаратная закладка.

Угрозы безопасности конфиденциальной информации (КИ), связанные с внедрением аппаратных закладок, определяются в соответствии с нормативным документами Федеральной службы безопасности Российской Федерации в установленном ею порядке.

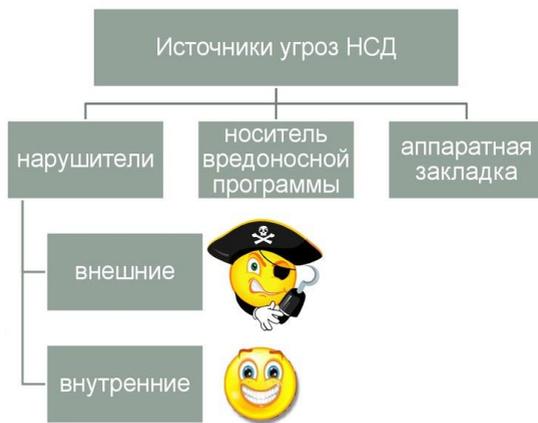


Рис. 1. Источники угроз НСД в АС

По наличию права постоянного или разового доступа в контролируемую зону (КЗ) АС нарушители подразделяются на два типа (рис. 2) [1]:

- нарушители, не имеющие доступа к АС, реализующие угрозы из внешних сетей связи общего пользования и (или) сетей международного информационного обмена, – внешние нарушители;
- нарушители, имеющие доступ к АС, включая пользователей АС, реализующие угрозы непосредственно в АС, – внутренние нарушители.

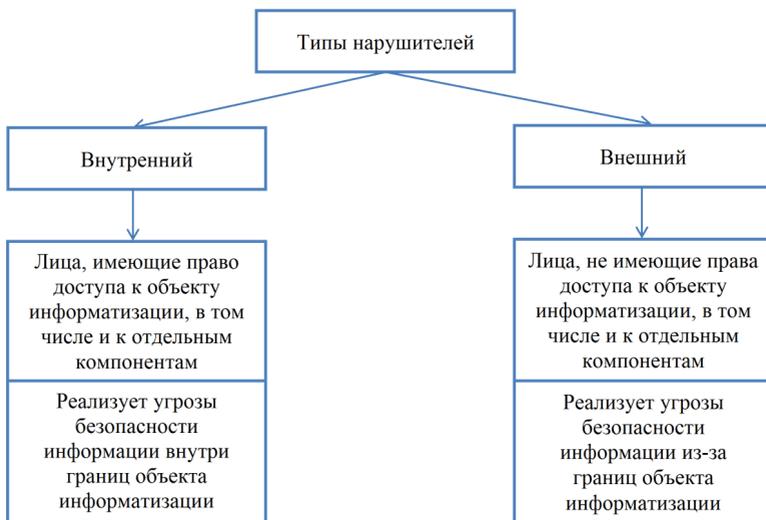


Рис. 2. Типы нарушителей ИБ

Внешними нарушителями могут быть: разведывательные службы государств; криминальные структуры; конкуренты (конкурирующие организации); недобросовестные партнеры; внешние субъекты (физические лица).

Внешний нарушитель имеет следующие возможности:

- осуществлять несанкционированный доступ к каналам связи, выходящим за пределы служебных помещений;
- осуществлять несанкционированный доступ через автоматизированные рабочие места, подключенные к сетям связи общего пользования и (или) сетям международного информационного обмена;
- осуществлять несанкционированный доступ к информации с использованием специальных программных воздействий посредством программных вирусов, вредоносных программ, алгоритмических или программных закладок;
- осуществлять несанкционированный доступ через элементы информационной инфраструктуры АС, которые в процессе своего жизненного цикла (модернизации, сопровождения, ремонта, утилизации) оказываются за пределами контролируемой зоны;
- осуществлять несанкционированный доступ через информационные системы взаимодействующих ведомств, организаций и учреждений при их подключении к АС.

Возможности внутреннего нарушителя существенным образом зависят от действующих в пределах контролируемой зоны режимных и организационно-технических мер защиты, в том числе по допуску физических лиц к ИИ и контролю порядка проведения работ.

Внутренние потенциальные нарушители подразделяются на восемь категорий в зависимости от способа доступа и полномочий доступа к ИИ.

К первой категории относятся лица, имеющие санкционированный доступ к АС, но не имеющие доступа к ИИ. К этому типу нарушителей относятся должностные лица, обеспечивающие нормальное функционирование АС. Лицо этой категории может:

- иметь доступ к фрагментам информации, содержащей ИИ и распространяющейся по внутренним каналам связи АС;
- располагать фрагментами информации о топологии АС (коммуникационной части подсети) и об используемых коммуникационных протоколах и их сервисах;

- располагать именами и вести выявление паролей зарегистрированных пользователей;
- изменять конфигурацию технических средств АС, вносить в нее программно-аппаратные закладки и обеспечивать съем информации, используя непосредственное подключение к техническим средствам АС.

Ко второй категории относятся зарегистрированные пользователи АС, осуществляющие ограниченный доступ к ресурсам АС с рабочего места. Лицо этой категории:

- обладает всеми возможностями лиц первой категории;
- знает, по меньшей мере, одно легальное имя доступа;
- обладает всеми необходимыми атрибутами (например, паролем), обеспечивающими доступ к некоторому подмножеству КИ;
- располагает конфиденциальными данными, к которым имеет доступ.

Доступ, аутентификация и права указанного лица по доступу к некоторому подмножеству КИ должны регламентироваться соответствующими правилами разграничения доступа.

К третьей категории относятся зарегистрированные пользователи АС, осуществляющие удаленный доступ к КИ по локальным и (или) распределенным информационным системам. Лицо этой категории:

- обладает всеми возможностями лиц первой и второй категорий;
- располагает информацией о топологии АС на базе локальной и (или) распределенной информационной системы, через которую осуществляется доступ, и о составе технических средств АС;
- имеет возможность прямого (физического) доступа к фрагментам технических средств АС.

К четвертой категории относятся зарегистрированные пользователи АС с полномочиями администратора безопасности сегмента (фрагмента) АС. Лицо этой категории:

- обладает всеми возможностями лиц предыдущих категорий;
- обладает полной информацией о системном и прикладном программном обеспечении, используемом в сегменте (фрагменте) АС;
- обладает полной информацией о технических средствах и конфигурации сегмента (фрагмента) АС;
- имеет доступ к средствам защиты информации и протоколирования, а также к отдельным элементам, используемым в сегменте (фрагменте) АС;
- имеет доступ ко всем техническим средствам сегмента (фрагмента) АС;
- обладает правами конфигурирования и административной настройки некоторого подмножества технических средств сегмента (фрагмента) АС.

К пятой категории относятся зарегистрированные пользователи с полномочиями системного администратора АС. Лицо этой категории:

- обладает всеми возможностями лиц предыдущих категорий;
- обладает полной информацией о системном и прикладном программном обеспечении АС;
- обладает полной информацией о технических средствах и конфигурации АС;
- имеет доступ ко всем техническим средствам обработки информации и данным АС;
- обладает правами конфигурирования и административной настройки технических средств АС.

Системный администратор выполняет конфигурирование и управление программным обеспечением (ПО) и оборудованием, включая оборудование, отвечающее за безопасность защищаемого объекта: средства криптографической защиты информации, мониторинга, регистрации, архивации, защиты от НСД [2].

К шестой категории относятся зарегистрированные пользователи с полномочиями администратора безопасности АС. Лицо этой категории [3]:

- обладает всеми возможностями лиц предыдущих категорий;
- обладает полной информацией об АС;
- имеет доступ к средствам защиты информации и протоколирования и к части ключевых элементов АС;
- не имеет прав доступа к конфигурированию технических средств сети за исключением контрольных (инспекционных).

Администратор безопасности отвечает за соблюдение правил разграничения доступа, за генерацию ключевых элементов, смену паролей. Администратор безопасности осуществляет аудит тех же средств защиты объекта, что и системный администратор.

К седьмой категории относятся программисты-разработчики (поставщики) прикладного программного обеспечения и лица, обеспечивающие его сопровождение на защищаемом объекте. Лицо этой категории [3]:

- обладает информацией об алгоритмах и программах обработки информации на АС;
- обладает возможностями внесения ошибок, недеklarированных возможностей, программных закладок, вредоносных программ в программное обеспечение АС на стадии ее разработки, внедрения и сопровождения;
- может располагать любыми фрагментами информации о топологии АС и технических средствах обработки и защиты КИ, обрабатываемых в АС.

К восьмой категории относятся разработчики и лица, обеспечивающие поставку, сопровождение и ремонт технических средств на АС. Лицо этой категории:

- обладает возможностями внесения закладок в технические средства АС на стадии их разработки, внедрения и сопровождения;
- может располагать любыми фрагментами информации о топологии АС и технических средствах обработки и защиты информации в АС.

Указанные категории нарушителей должны учитываться при оценке возможностей реализации УБКИ.

Носителем вредоносной программы может быть аппаратный элемент компьютера или программный контейнер. Если вредоносная программа не ассоциируется с какой-либо прикладной программой, то в качестве ее носителя рассматриваются:

- отчуждаемый носитель, т. е. дискета, оптический диск (CD-R, CD-RW), флэш-память, отчуждаемый винчестер и т. п.;
- встроенные носители информации (винчестеры, микросхемы оперативной памяти, процессор, микросхемы системной платы, микросхемы устройств, встраиваемых в системный блок, – видеоадаптера, сетевой платы, звуковой платы, модема, устройств ввода/вывода магнитных жестких и оптических дисков, блока питания и т. п., микросхемы прямого доступа к памяти, шин передачи данных, портов ввода/вывода);
- микросхемы внешних устройств (монитора, клавиатуры, принтера, модема, сканера и т. .).

Если вредоносная программа ассоциируется с какой-либо прикладной программой, с файлами, имеющими определенные расширения или иные атрибуты, с сообщениями, передаваемыми по сети, то ее носителями являются:

- передаваемых по компьютерной сети сообщений;
- файлы (текстовые, графические, исполняемые и т. д.).

Вывод

В статье рассмотрена классификация источников угроз БИ, учет которых позволит в дальнейшем построить модель угроз БИ, а также адекватную рассмотренным источникам угроз БИ систему защиты информации объектов информатизации организации.

Библиографический список

1. ГОСТ Р 51275–2006. Защита информации. Объект информатизации. Факторы, воздействующие на информацию. Понятие объекта информатизации. URL: docs.cntd.ru/document/1200057516 (дата обращения: 16.03.2021).
2. Хорев А. А. Защита информации от утечки по техническим каналам. Часть 1. Технические каналы утечки информации: учеб. пособие. М.: Гостехкомиссия России, 1998. 320 с.
3. ГОСТ Р 50922–2006. Защита информации. Основные термины и определения. URL: https://docs.cntd.ru/document/1200058320 (дата обращения: 16.03.2021).

УДК 004.855.5

М. С. Савинов

студент кафедры безопасности информационных систем

А. И. Веселов – кандидат технических наук, доцент – научный руководитель

УМЕНЬШЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ПОИСКА ОБЪЕКТОВ НА ИЗОБРАЖЕНИЯХ

Сегодня мощным инструментом для решения научных и прикладных задач стали нейронные сети. Некоторые алгоритмы, основанные на них, оказываются лучшим способом решения тех или иных задач, в частности детектирования, распознавания и классификации объектов на изображениях. Благодаря нейронным сетям существенного успеха удалось достичь в области компьютерного зрения.

Развертывание нейронных сетей на устройствах с ограниченными ресурсами, например на мобильных телефонах, сложно не только из-за их большой вычислительной сложности, но и из-за высоких требований к памяти. Современные мобильные устройства обладают мощностью и памятью, достаточной для запуска только небольших нейронных сетей, но для более сложных моделей ресурсов не хватает. Для преодоления этой проблемы были разработаны методы сжатия и ускорения моделей.

Цель данной работы – реализация методов сжатия и упрощения сверточных нейронных сетей для детектирования людей на изображениях. Нейронные сети тестируются на Jetson Nano [1]. Jetson Nano – это микрокомпьютер от компании Nvidia, в который встроена видеокарта. Имеет 128 ядер CUDA и не является высокопроизводительным микрокомпьютером. Для сравнения, видеокарта TITAN X [2] имеет 3 584 ядра. Но низкая стоимость Jetson Nano и возможность запускать нейронные сети открывает новые возможности для бюджетных проектов, в которых необходимо использовать нейронные сети.

Чтобы определить человека на изображении, можно воспользоваться двумя типами нейронных сетей. Первый детектирует или сегментирует человека, это нейросети: Mask-RCNN [3], Pose2Seq [4], YOLACT++ [5]. Второй тип определяет скелет человека, примером служит TRT Pose [6].

Для перечисленных нейронных сетей были посчитаны вычислительная сложность и скорость работы. Для этого с помощью видеокарты TITAN X было измерено количество кадров размера 512×512 в секунду (frame per second, FPS). Для измерения использовался программный таймер. Также была измерена вычислительная сложность нейронной сети (Floating-point Operations Per Second, FLOPSs). Для расчета FLOPs использовалась библиотека ptflops [7]. Результаты представлены в табл. 1.

Таблица 1

Результаты измерений

Название	FPS	GFLOPs
Mask-RCNN	10	447
Pose2Seq	20	150
YOLACT++	33	100
TRT Pose	211	28

Наиболее быстрая и наименее сложная нейронная сеть из рассмотренных – TRT Pose. Далее в работе была использована именно она. Обычный запуск TRT Pose на Jetson Nano дает 5 кадров в секунду. Для ускорения работы сети применяется утилита от Nvidia – TensorRT [8], которая позволяет адаптировать нейронную сеть под определенную видеокарту. С утилитой была получена скорость 16 кадров в секунду, что недостаточно для использования этой сети в режиме реального времени.

К способам уменьшения вычислительной сложности нейронных сетей относятся: дистилляция, квантование, обрезание сети, использование специальных сверточных блоков и уменьшение размера сети [9]. На рис. 1 представлена структура сети TRT Pose. Видно, что самая тяжелая часть этой нейронной сети – блок Upsample (рис. 2), который имеет сложность 9 GFLOPs, а общая сложность сети равна 28 GFLOPs.

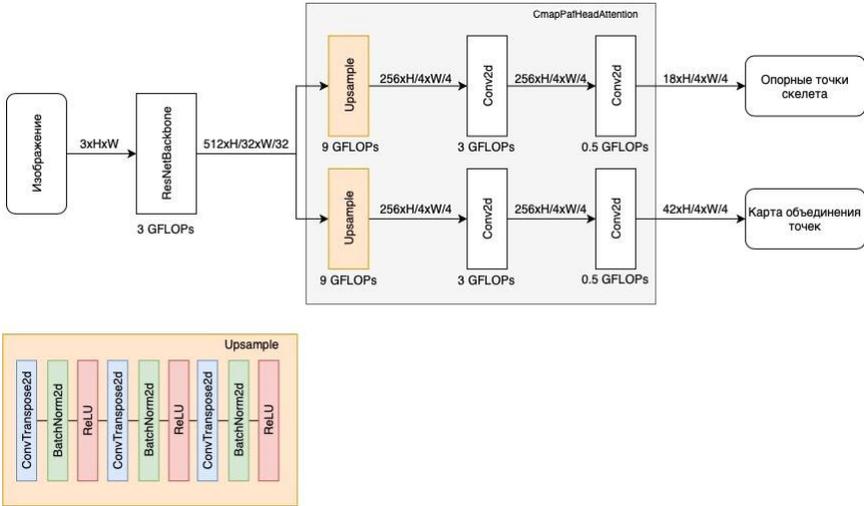


Рис. 1. Структура сети TRT Pose

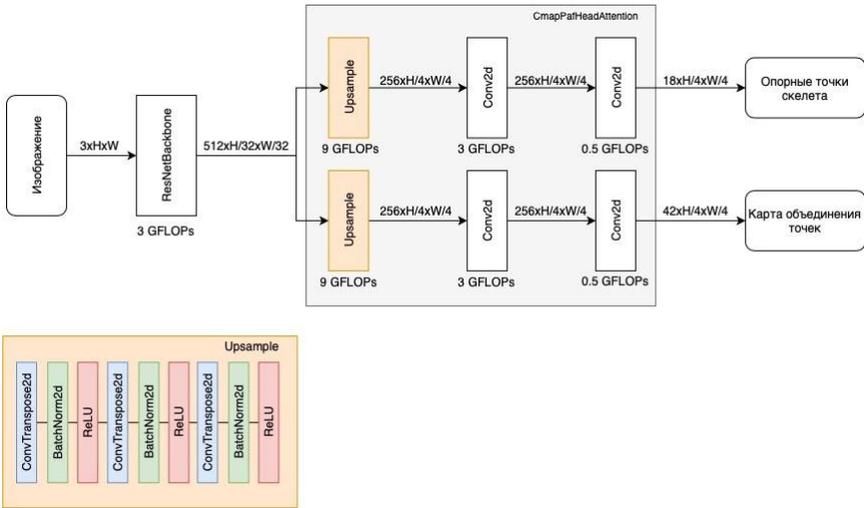


Рис. 2. Структура блока upsample

С целью увеличения скорости сети были использованы сепарабельные свертки (рис. 3), расширенные свертки (рис. 4) и уменьшение числа каналов.

Также было протестировано уменьшение числа каналов. Для этого глубина каждого сверточного слоя в CmapPatHeadAttention (см. рис. 1) была уменьшена в два раза.

В результате использования методов увеличения скорости были получены результаты, представленные в табл. 2.

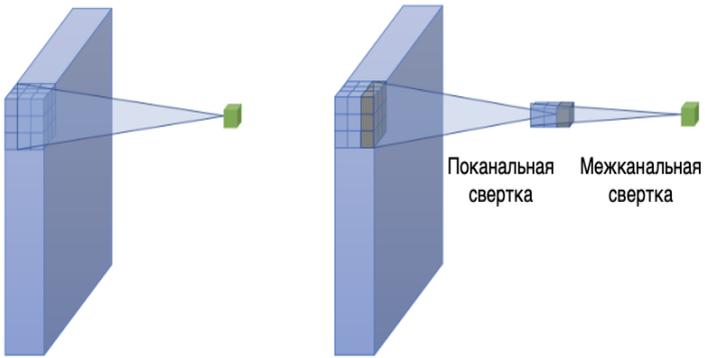


Рис. 3. Стандартная свертка (слева) и сепарабельная свертка (справа)

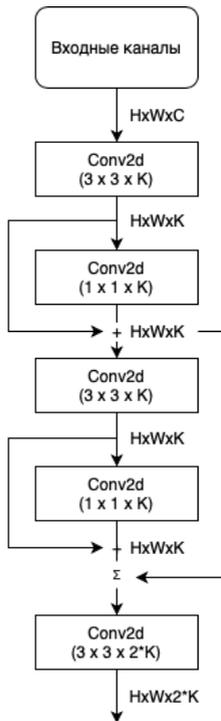


Рис. 4. Расширенные свертки

Результаты

Примененный подход	FPS	GFLOPs	Точность
Оригинальная сеть	16	28	0,9
Сепарабельные свертки	15	12	0,57
Расшитые свертки	14	26	0,94
Уменьшение числа каналов	36	10	0,45
Уменьшение числа каналов + расшитые свертки	38	9	0,84

При уменьшении GFLOPs на Jetson Nano не всегда увеличивается скорость работы. Например, при использовании сепарабельных свертков уменьшилось количество GFLOPs, но при этом также снизилась скорость работы. Это связано с тем, что TensorRT не может ускорить работу сепарабельных свертков на Jetson Nano из-за маленькой вычислительной мощности видеокарты [10].

В данной работе были запущены некоторые модели нейронных сетей для детектирования людей. Была произведена адаптация нейронной сети TRT Pose, позволяющая поднять кадровую скорость с 16 до 38. Выявлены особенности оптимизации нейронных сетей под Jetson Nano: отсутствует выигрыш от сепарабельных свертков.

Цель дальнейшей работы – тестирование других методов уменьшения сложности нейронной сети (прунинг, дистилляция), а также использование альтернативных нейронных сетей для адаптации. Также планируется проектирование компактной нейронной сети.

Библиографический список

1. JETSON NANO. URL: <https://www.nvidia.com/ru-ru/autonomous-machines/embedded-systems/jetson-nano/product-development/> (дата обращения: 17.05.2021).
2. NVIDIA TITAN X. URL: <https://www.nvidia.com/en-us/geforce/products/10series/titan-x-pascal/> (дата обращения: 17.05.2021).
3. Mask R-CNN / K. He, G. Gkioxari, P. Dollár, R. Girshick // Computer Vision and Pattern Recognition. 2017. arXiv:1703.06870.
4. Pose2Seg: Detection Free Human Instance Segmentation / S.-H. Zhang, R. Li, X. Dong [et al.] // Computer Vision and Pattern Recognition. 2018. CVPR 2019. arXiv:1803.10683.
5. YOLACT++: Better Real-time Instance Segmentation / D. Bolya, C. Zhou, F. Xiao, Y. J. Lee. // Computer Vision and Pattern Recognition. 2019. arXiv:1912.06218.
6. NVIDIA-AI-IOT/trt_pose. URL: https://github.com/NVIDIA-AI-IOT/trt_pose (дата обращения: 17.05.2021).
7. ptflops. URL: <https://github.com/sovrasov/flops-counter.pytorch> (дата обращения: 17.05.2021).
8. NVIDIA TensorRT. URL: <https://developer.nvidia.com/tensorrt> (дата обращения: 17.05.2021).
9. He Y., Qian J., Wang J. Depth-wise Decomposition for Accelerating Separable Convolutions in Efficient Convolutional Neural Networks // CVPR 2019 workshop, Efficient Deep Learning for Computer Vision. 2019. arXiv:1910.09455.
10. Design of a Scalable and Fast YOLO for Edge-Computing Devices / B.-G. Han, J.-G. Lee, K.-T. Lim, D.-H. Choi // MDPI. Sensors 2020, 20(23), 6779.

УДК 004.855.5

М. Д. Савинова

студент кафедры безопасности информационных систем

А. И. Веселов – кандидат технических наук, доцент – научный руководитель

РАСПРЕДЕЛЕННОЕ КОДИРОВАНИЕ ИЗОБРАЖЕНИЙ С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ

С распространением Интернета и социальных сетей стало доступно и популярно снимать и делиться фото. Чем выше разрешение изображения, тем больше требуется памяти для его хранения и времени для передачи в оригинальном виде. Многие из пользовательских устройств (смартфоны, планшеты, ноутбуки) работают от аккумулятора. Актуальна разработка таких алгоритмов, которые использовали бы меньше вычислительных ресурсов устройства пользователя.

Распределенное кодирование – перспективный подход к сжатию изображений, в рамках которого теоретически возможно существенное уменьшение вычислительной сложности на стороне передатчика за счет усложнения приемника. Оно применяется в том случае, когда существует некоторая дополнительная информация на декодере, которая там заранее хранится или появляется в процессе декодирования. При кодировании видео или стереоизображений эта дополнительная информация берется из уже восстановленных кадров.

Распространение и развитие распределенного кодирования (Distributed Source Coding) началось с результата Слепиана и Вульфа [1], которые доказали, что для пары зависимых источников X и Y можно кодировать выход источника X , используя информацию об Y только на декодере (рис. 1), при этом не теряя скорости передачи по сравнению со схемами, где информация об Y доступна как на кодере, так и на декодере [2]. Позже этот результат был применен Вайнером и Зивом [3] на случай кодирования с потерями информации.

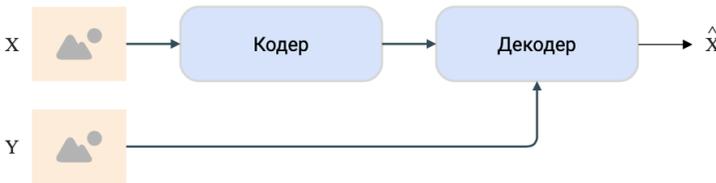


Рис. 1. Схема распределенного кодирования

Хотя теория распределенного кодирования восходит к 1970-м годам, только спустя 30 лет была представлена первая практическая реализация. Сейчас большинство схем распределенного кодирования видеоданных использует классические методы оценки движения на стороне декодера. В течение последнего десятилетия во многих областях обработки изображений существенного прогресса удалось достигнуть за счет применения нейронных сетей. При сжатии видеоинформации они позволяют не искать вручную признаки, используемые для кодирования.

Цель данной работы заключается в исследовании алгоритма распределенного кодирования изображений с использованием нейронных сетей. В качестве базовой работы для исследования была взята статья [4], в которой впервые было предложено использовать нейронные сети для рассматриваемой задачи.

Предлагаемая авторами [4] схема сжатия представлена на рис. 2 и включает две сверточные нейронные сети (СНС1, СНС2), блок квантования (Q), арифметический кодер (АК) и декодер (АД), а также блок генерации дополнительной информации (ДИ).

Передатчик имеет доступ к входному изображению X , а приемник – к изображению Y . Предполагается, что картинка X похожа на Y . СНС1, включающая СНС-кодер и СНС-декодер, предназначена для сжатия изображений и основана на модели [5]. Кодер принимает входное изображение X размера $H \times W \times 3$ и выдает тензор размера $H/8 \times W/8 \times C$, где H , W – высота и ширина исходного изображения в 234

пикселях, 3 – число каналов изображения до кодирования (RGB), C – число каналов после кодирования. После этого тензор квантуется и кодируется арифметическим кодером. Степень сжатия можно управлять с помощью параметра C .

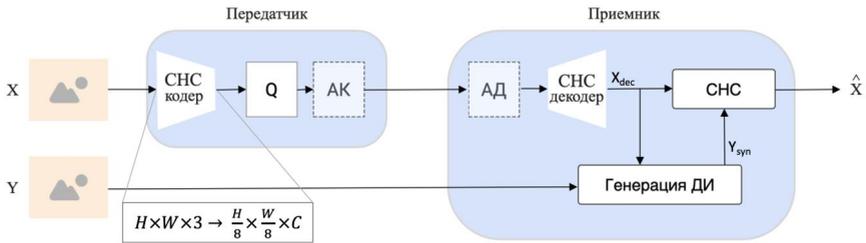


Рис. 2. Исследуемая схема распределенного кодирования с помощью нейронных сетей [4]

На приемной стороне после арифметического декодирования тензор поступает на декодер СНС1. Блок генерации ДИ принимает декодированное изображение X_{dec} вместе с изображением Y и использует их для создания дополнительной информации – Y_{syn} (рис. 3). Декодированное изображение X_{dec} и синтезированное изображение Y_{syn} используются для получения выходного изображения \hat{X} с помощью СНС2, основанной на [6].

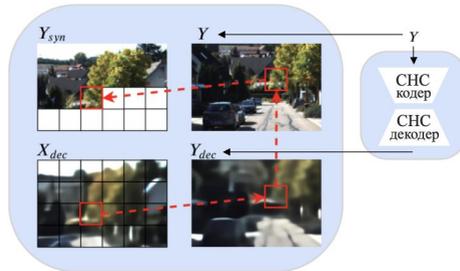


Рис. 3. Генерация дополнительной информации [4]

В рамках данной работы не были реализованы блоки, относящиеся к арифметическому кодированию. Для анализа эффективности сжатия использовались следующие величины.

• Величина BPP (bits per pixel, число бит на пиксель) – степень сжатия. При отсутствии арифметического кодера можно оценить степень сжатия с помощью следующей формулы:

$$BPP = \frac{H \cdot N_{compressed}}{b \cdot N_{original}},$$

$$H = -\sum_i f_i \log f_i,$$

где H – оценка энтропии при независимом кодировании, $N_{compressed}$ – число символов сжатой последовательности, b – число бит на символ оригинального изображения [7], $N_{original}$ – число символов в оригинальном изображении, f_i – частота появления i -го символа после сжатия.

• Величина PSNR (пиковое отношение сигнала к шуму), которая рассчитывается по формуле

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{L^2}{\text{MSE}} \right),$$

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (p_i - \hat{p}_i)^2,$$

где L – максимально возможное значение пикселя (для 8-битных изображений – 255), N – общее число пикселей, p_i – значение i -го пикселя оригинального изображения, \hat{p}_i – значение i -го пикселя после восстановления. Чем больше величина PSNR, тем лучше восстановление.

Нейронная сеть была написана на языке Python с помощью библиотеки PyTorch и обучена в течение 200 эпох при разных параметрах C . Вся сеть, включающая две подсети, обучалась совместно. Во время обучения использовались 400 пар изображений из стерео датасета KITTI [8]. Результаты оценки степени сжатия (BPP) и восстановления (PSNR) при разных значениях C представлены в табл. 1.

Таблица 1

Результаты

C	BPP	PSNR
16	0,1401	28,08
32	0,397	28,69
64	0,507 7	29,54
128	0,638 7	29,59

На рис. 4 представлены кривые скорости-искажения, построенные для кодирования с помощью описанной архитектуры и для кодирования с помощью стандарта сжатия jpeg, которое было реализовано с использованием библиотеки turbojpeg.

В результате анализа кривых можно сделать вывод, что реализованная система пока проигрывает стандарту сжатия jpeg. Проигрыш может быть обусловлен тем, что при оценке степени сжатия исследуемым способом не учитывается нестационарность обрабатываемых данных. Jpeg же учитывает эту нестационарность.

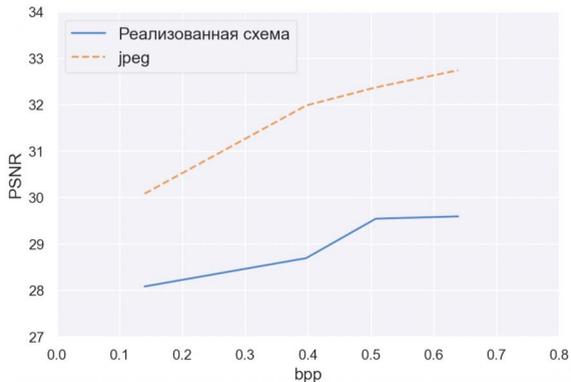


Рис. 4. Кривые скорости-искажения

Целью дальнейшей работы является реализация блоков, отвечающих за арифметическое кодирование, а также подбор параметров обучения с целью улучшения восстановления изображения. В качестве еще одного направления рассматривается изменение блока генерации дополнительной информации с целью увеличения скорости его работы.

Библиографический список

1. Slepian D., Wolf J. K. Noiseless coding of correlated information sources // IEEE Transactions on Information Theory. 1973. Vol. 19(4). P. 471–480.
2. Cover T. M. A proof of the data compression theorem of Slepian and Wolf for ergodic sources // IEEE Transactions on Information Theory. 1975. Vol. 21(2). P. 226–228.
3. Wyner A. D., Ziv J. The rate-distortion function for source coding with side information at the decoder // IEEE Transactions on Information Theory. 1976. Vol. 22(1). P. 1–10.
4. Ayzik S., Avidan S. Deep Image Compression using Decoder Side Information. 2020. arXiv: 2001.04753.
5. Conditional probability models for deep image compression / F. Mentzer, E. Agustsson, M. Tschanen [et al.] // Computer Vision and Pattern Recognition. 2018. P. 4394–4402.
6. Chen Q., Xu J., Koltun V. Fast image processing with fully-convolutional net-works // International Conference on Computer Vision. 2017. P. 2516–2525.
7. Stereo Evaluation 2015. The KITTI Vision Benchmark Suite. URL: http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo (дата обращения: 24.04.2021).
8. PyTurboJPEG 1.5.0. A Python wrapper of libjpeg-turbo for decoding and encoding JPEG image. URL: <https://pypi.org/project/PyTurboJPEG/> (дата обращения: 02.05.2021).

УДК 004.414.38

Б. С. Шром

студент кафедры безопасности информационных систем

А. А. Овчинников – кандидат технических наук, доцент – научный руководитель

АРХИТЕКТУРА БЕЗОПАСНОСТИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ОПЕРАЦИОННОЙ СИСТЕМЫ ANDROID

Введение

В современном мире почти у каждого человека есть смартфон. Он хранит много информации, которой сам человек не захочет делиться с посторонними, что может стать мотивацией для злоумышленников к неправомерному получению доступа к данной информации. Для защиты конфиденциальности информации пользователей смартфонов разрабатываются приложения, осуществляющие шифрование. При разработке таких приложений возникает необходимость учесть возможные уязвимости, которые могут привести к нарушению конфиденциальности защищаемой информации. При этом перечень актуальных уязвимостей для конкретного приложения может быть уникальным.

Для выявления актуальных уязвимостей приложения перед началом его разработки создается архитектура его безопасности, в которой рассматривается среда исполнения приложения и определяется структура потоков данных, после чего среда исполнения и структура потоков данных сопоставляются друг с другом и на основе этого описываются модель нарушителя и модель угроз.

Принцип работы защищаемого приложения

Защищаемое приложение служит для шифрования текста с применением симметричного шифра. Симметричный шифр – такой шифр, в котором преобразование открытого текста в шифртекст и обратное преобразование из шифртекста в открытый текст происходит с применением одного и того же секретного ключа [1]. В нашем приложении ключ генерируется средствами самого приложения.

Данное приложение можно применять для ведения конфиденциальной переписки между сторонами А и Б. Для этого одной из сторон (в данном примере это сторона А) необходимо сгенерировать секретный ключ и передать его стороне Б при помощи защищенного канала связи, после чего с помощью полученного секретного ключа зашифровать открытое сообщение. В результате шифрования будет получен шифртекст, который можно передавать стороне Б с использованием любого доступного канала связи (в том числе открытого). Сторона Б, получив шифртекст, дешифрует его с применением секретного ключа, полученного ранее и получает изначальное открытое сообщение. Принцип работы приложения схематично изображен на рис. 1.

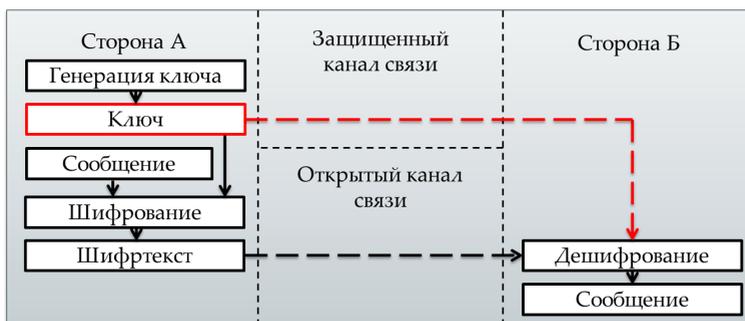


Рис. 1. Принцип работы приложения

Среда исполнения приложения

Будущее приложение будет разработано для операционной системы Android. Оно будет запускаться внутри данной операционной системы, которая в свою очередь будет запущена на электронном устройстве (например, смартфоне). Обеспечить безопасность приложения можно лишь при обеспечении безопасности среды исполнения данного приложения: операционной системы и устройства, на которое установлены эта операционная система и само приложение [2].

Структура потоков данных, обрабатываемых в приложении

Все данные, с которыми работает приложение можно разделить на две группы: «защищаемые данные» и «незащищаемые данные». В рассмотренном алгоритме использования приложения фигурируют следующие данные: ключ, сообщение и шифртекст (рис. 2).

Ключ используется как для шифрования, так и для дешифрования. В случае если злоумышленник получает доступ к ключу, он может провести дешифрование, перехватив посылаемый по открытому каналу шифртекст, и получить доступ к открытому сообщению. Для сохранения конфиденциальности передаваемой информации необходимо позаботиться сохранностью ключа, т. е. ключ относится к «защищаемым данным».

Передаваемое сообщение следует защищать исходя из самой задачи приложения, т. е. данный вид информации также относится к «защищаемым данным».

Шифртекст защищен криптографическим алгоритмом и доступ злоумышленника к нему не обеспечит доступ к сообщению. Шифртекст относится к «незащищаемым данным».

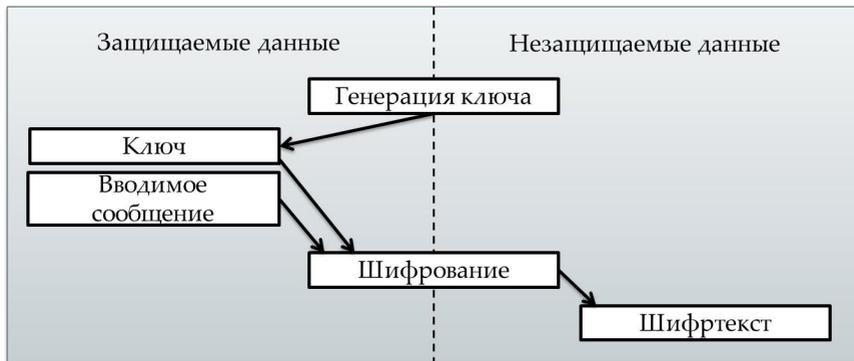


Рис. 2. Обрабатываемые приложением данные

Таким образом, для обеспечения конфиденциальности передаваемого сообщения необходимо защитить лишь само сообщение и секретный ключ, применяющийся для шифрования и дешифрования.

Обмен данными с точки зрения операционной системы Android выглядит следующим образом [3] (рис. 3). Приложение запускается в изолированной программной среде, доступа к которой иные установленные на устройстве приложения не имеют. В данной среде генерируются ключ и шифртекст, в нее вводится открытое сообщение. Для передачи ключа стороне Б ключ экспортируется во внешнее хранилище. Так как ключ относится к защищаемым данным, при разработке приложения необходимо уделить особое внимание его безопасному экспорту. Шифртекст после своей генерации копируется в буфер обмена для дальнейшей передачи стороне Б любым доступным способом. Шифртекст покидает изолированную программную среду, но относится к «незащищаемым данным», т. е. заботиться о его безопасности не нужно.

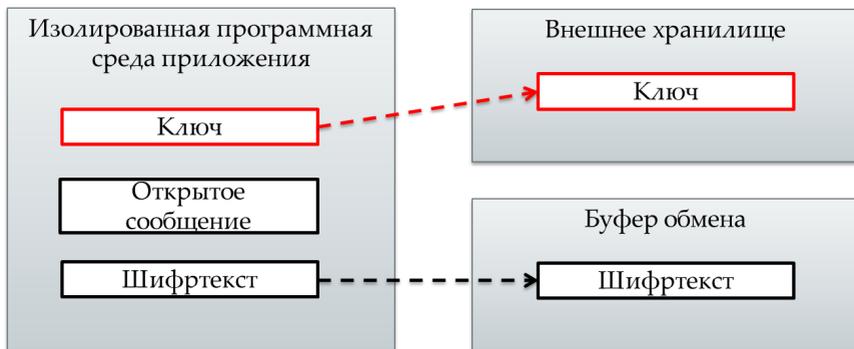


Рис. 3. Обмен данными с точки зрения операционной системы

Создание модели нарушителя и модели угроз

Для более полного понимания реализации возможных угроз необходимо сформировать модель нарушителя. Введем два вида нарушителей: внутреннего и внешнего. Под внутренним будем понимать лицо, получившее неправомерный доступ к устройству, операционной системе или приложению. Под внешним – лицо, не являющееся легальным пользователем и не имеющее прямого доступа к устройству или операционной системе устройства пользователя.

После рассмотрения возможных источников угроз, защищаемых данных и среды исполнения приложения возможно создание модели актуальных угроз, защиту от которых необходимо предусмотреть (если это возможно) в ходе последующей разработки приложения (табл. 1).

Таблица 1

Модель актуальных угроз для рассматриваемого приложения

Угроза	Источник угрозы	Способ реализации угрозы	Активы, попадающие под угрозу	Последствия реализации угрозы
Несанкционированный экспорт ключа	Внутренний нарушитель	Доступ к устройству; доступ к операционной системе	Ключ, сообщения	Вся последующая переписка будет доступна злоумышленнику до смены ключа
Утечка сообщения	Внутренний и внешний нарушители	Использование сторонних клавиатур	Сообщение	Злоумышленнику доступно лишь данное сообщение
Удаление приложения	Внутренний нарушитель	Доступ к устройству; доступ к операционной системе	Приложение, ключ, сообщения	Теряется возможность вести переписку
Кража устройства	Внутренний нарушитель	Доступ к устройству	Устройство, операционная система приложения, ключ, сообщения	Теряется возможность вести переписку
Несанкционированный доступ к операционной системе	Внешний нарушитель	Вредоносное ПО на устройстве с root правами	Операционная система приложения, ключ, сообщения	Злоумышленник получает полный контроль над операционной системой

Библиографический список:

1. Криптографическая защита информации / А. В. Яковлев, А. А. Безбогов, В. В. Родин, В. Н. Шамкин. Тамбов: ТГТУ, 2006. 140 с.
2. Уязвимости и угрозы мобильных приложений. 2019. URL: <https://www.ptsecurity.com/ru-ru/research/analytics/mobile-application-security-threats-and-vulnerabilities-2019/> (дата обращения: 21.05.2021).
3. Android Security Features. URL: <https://source.android.com/security/features> (дата обращения: 18.05.2021).

УДК 004.7

Д. А. Васильев

студент кафедры инфокоммуникационных систем

А. М. Тюрликов – доктор технических наук, профессор – научный руководитель**ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ УДАЛЕННОГО БЕСПРОВОДНОГО
ДОСТУПА ПОЛЬЗОВАТЕЛЯ К РАСПРЕДЕЛЕННЫМ ВЫЧИСЛИТЕЛЬНЫМ РЕСУРСАМ**

Сейчас активно развивается концепция интернета вещей (Internet of things – IoT), основанная на передаче данных между различными физическими объектами. Данные объекты могут представлять собой различные вещи, начиная от лампочек и заканчивая автомобилями. Очень скоро IoT настолько плотно войдет в нашу жизнь, что станет неотъемлемой ее частью. Таким образом, количество устройств с возможностью передачи данных будет только увеличиваться. Значит, и нагрузка на мобильные сети передачи информации будет возрастать. Для того чтобы частично снизить нагрузку на сети, в их состав, начиная с пятого поколения были добавлены МЕС-сервера (Multi-Access Edge Computing). Задачей данных серверов, стала обработка части задач «на месте», т. е. они вычисляют задачи без передачи данных на облачные серверы. Для данного типа серверов возникает задача, как грамотно распределить предоставляемые ими вычислительные мощности, в случае когда устройства могут перемещаться между различными областями сети, а следовательно, оказываться в зоне работы разных серверов. Поэтому необходимо понять, как максимально эффективно организовать нагрузку на серверы, когда пользователю лучше перенести свою задачу на другой сервер, а куда оставить на текущем. Модель, представленная в данной статье, требует меньшего количества параметров для того, чтобы получить аналитические данные о работе сети. В другой работе [1] представлена более общая модель системы. Также отличительной особенностью представленной модели, от моделей, описанных в работах [2–8], является меньшее количество параметров, необходимых для описания их работы. Данная особенность упрощает процесс создания программы для имитационного моделирования, а также упрощает процесс сбора необходимых данных с реальной сети.

Система допущений для модели

Представленная в данной статье модель базируется на следующих допущениях.

Допущение 0. В системе имеется n областей одинакового размера. Пользователи могут появляться в системе в любой области и перемещаться из одной в другую.

Допущение 1. Процесс появления пользователя в i -й области описывается пуассоновским потоком с интенсивностью λ_i . Объем задачи, который должен выполнить пользователь, распределен по экспоненциальному закону с параметром 1. Вычислительная мощность в j -й области равняется μ_j , это означает, что при условии если пользователь не перемещается в другие области, а также является единственным в локации, то время выполнения задачи будет распределено по экспоненциальному закону с параметром μ_j . Если в области имеются другие пользователи, то время выполнения увеличивается (см. допущение 4). Если пользователь перемещается в другую область, то время выполнения задачи также увеличивается (см. допущение 3).

Допущение 2. Если пользователь находится в области i , то может произойти одно из следующих событий. Выполнение задачи пользователя завершается, и пользователь покидает систему или перемещается в некоторую область j . Какое событие произойдет, определяется так: для всех $j \neq i$ формируются случайные величины, распределенные по экспоненциальному закону с параметром q_{ij} . На данном этапе мы считаем, что q всегда равно 1, поскольку предполагаем, что пользователь равноудален от других локаций. Таких величин у нас формируется $n - 1$, и k меньшей из них прибавляется

время выполнения работы, если оно меньше, чем время перехода в другую область, то пользователь покидает систему по окончании выполнения своей работы.

Допущение 3. Если пользователь переходит из любой области в любую, то на перенос задачи с одних мощностей на другие, затрачивается время, распределенное по экспоненциальному закону с параметром d . В этот момент решение задачи не производится.

Допущение 4. Если пользователь, у которого работа выполнялась в области k , в соответствии с допущением 2, перешел из области i в область j , то он в случае если задача обрабатывается на серверах той же области, где расположен пользователь, с вероятностью a_{ijk} решит перенести свою работу из области k в область j , а с вероятностью $1 - a_{ijk}$ оставит свою работу в области k . В случае если области расположения пользователя и обработки задачи не совпадают, тогда пользователь решит не переносить свою задачу на другие серверы.

Список параметров

Для получения аналитических данных при помощи предложенной модели необходимо задать ей следующие параметры:

n – количество областей, оснащенных вышкой сотовой связи и сервером для проведения необходимых вычислений;

λ_i – интенсивность появления пользователей в i -й локации;

a_{ijk} – вероятность того, что пользователь станет переносить свою задачу на серверы другой области;

d – коэффициент, при помощи которого рассчитывается время, необходимое для переноса задачи между различными серверами;

μ_j – коэффициент, который показывает, как быстро может производиться работа в локации i .

Для всех описанных величин параметры i, j принимают значения $i, j \in [1, n]$.

Результаты экспериментов работы системы, в случае когда пользователи не перемещаются между локациями

Так как для описанной модели в общем случае отсутствуют аналитические данные, то для проверки работы моделирующей программы была взята известная теоретическая формула [9]:

$$T(x) = \frac{x}{1 - \rho}, \quad (1)$$

где x равно 1, ρ – значение входной интенсивности.

Таким образом, мы задаем системе ряд параметров, при помощи которых приводим нашу систему к виду системы пакетной обработки, для которой уже есть известные формулы, которые позволяют оценить выходные параметры системы (рис. 1). Поэтому использование формулы (1) позволяет получить теоретические значения в случае, когда задача пользователя обрабатывается только на одном сервере и интенсивность входного потока меньше 1. Параметр вероятности перемещения задачи $a_{ijk} = 0$. Мы получаем график изменения среднего значения задержки при увеличении входной интенсивности (рис. 2). Для оценки мы рассматриваем график только до значения входной интенсивности не больше 1, так как формула корректна только для случаев, когда значение входной интенсивности меньше 1. Оценив теоретический график, а также график значения, для которого были получены путем моделирования, можно сделать вывод, что в данном случае система работает верно.

Так же мы сравниваем значения, рассчитанные по формуле (1), со значениями, полученными при помощи моделирования в случае, когда задача пользователя обрабатывается только на одном сервере, а также имеет фиксированный размер равный 1 (рис. 3). Полученные графики отображены на рис. 4. Также, оценив представленные на рис. 4 графики, можно сделать вывод, что в данном случае система работает верно, потому что значения теоретического и практического графиков совпадают.

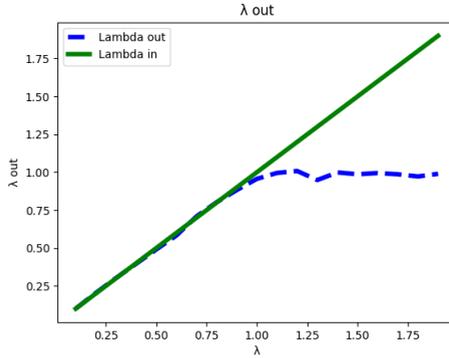


Рис. 1. Изменение выходной интенсивности системы при изменении входной интенсивности для случая, когда пользователи не переходят между локациями

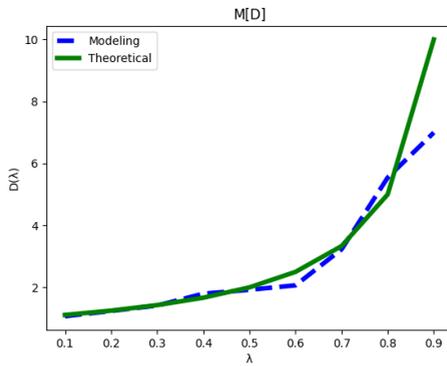


Рис. 2. Изменение среднего значения задержки при увеличении входной интенсивности для случая, когда пользователи не переходят между локациями

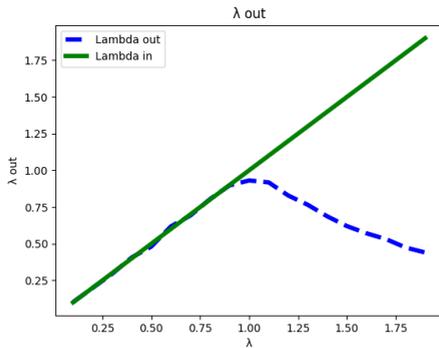


Рис. 3. Изменение выходной интенсивности системы при изменении входной интенсивности для случая, когда пользователи не переходят между локациями и размер их задачи имеет фиксированный размер, равный 1

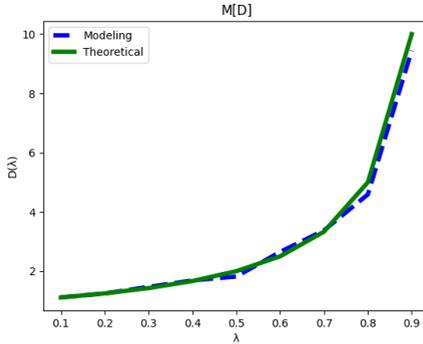


Рис. 4. Изменение среднего значения задержки при увеличении входной интенсивности для случая, когда пользователи не переходят между локациями и размер их задачи имеет фиксированный размер, равный 1

Мы провели два теста системы с заданными условиями. При их помощи мы проверили правильность работы системы в случаях, когда известны теоретические данные, и обнаружили, что время обслуживания заявок в системе не зависит от закона обслуживания.

Результаты экспериментов работы системы в случае с перемещением пользователей

На рис. 5–12 можно оценить, как изменяются выходные данные системы при условии, когда пользователи перемещаются между различными областями и объем их задачи равен случайной величине, распределенной по экспоненциальному закону с параметром 1. На данных рисунках значение параметра a_{ijk} изменяется.

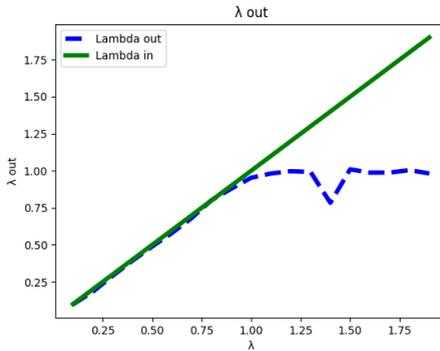


Рис. 5. Изменение выходной интенсивности системы при изменении входной интенсивности для случая, когда пользователи всегда переносят свои работы на серверы новой локации

На рис. 6, 8, 10 можно заметить, что при низкой интенсивности входного потока величина среднего значения задержки выше, чем при части следующих значений. Скорее всего, такое поведение значения задержки можно объяснить тем, что при низких значениях входной интенсивности и достаточно большом значении параметра a_{ijk} в системе формируется малое количество пользователей, которые достаточно часто решают перенести свою задачу на другой сервер, вследствие чего их задача остается в системе дольше на период сервисного времени, в течение которого задача не обслуживает-

ся сервером, а только переносится с одних серверов на другие. Данное сервисное время рассчитывается как случайная величина, распределенная по экспоненциальному закону с параметром d .

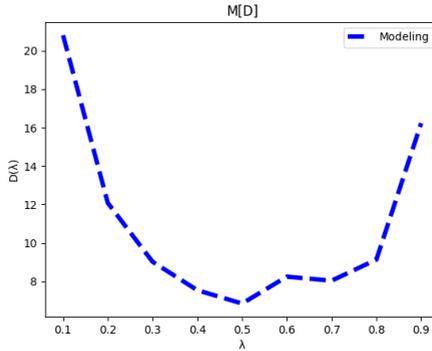


Рис. 6. Изменение среднего значения задержки при увеличении входной интенсивности для случая, когда пользователи всегда переносят свои работы на серверы новой локации

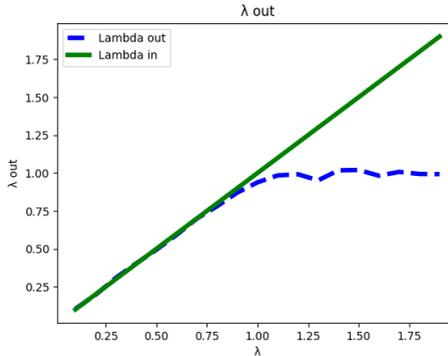


Рис. 7. Изменение выходной интенсивности системы при изменении входной интенсивности для случая, когда пользователи решают перенести свои работы на серверы новой локации с вероятностью 0,8

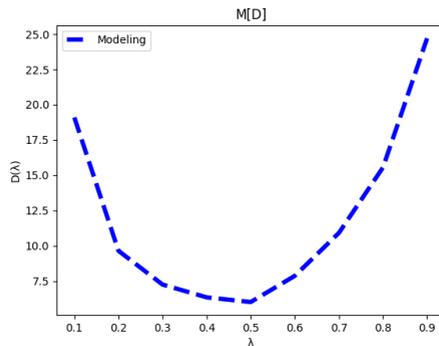


Рис. 8. Изменение среднего значения задержки при увеличении входной интенсивности для случая, когда пользователи решают перенести свои работы на серверы новой локации с вероятностью 0,8

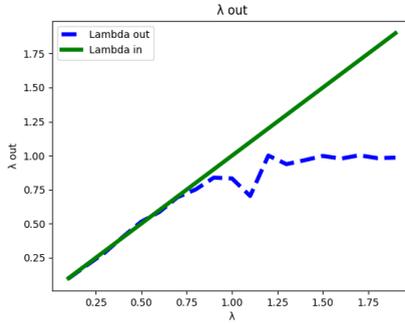


Рис. 9. Изменение выходной интенсивности системы при изменении входной интенсивности для случая, когда пользователи решают перенести свои работы на серверы новой локации с вероятностью 0,5

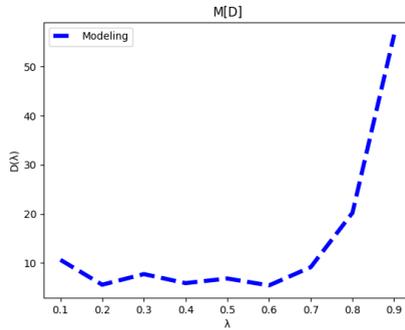


Рис. 10. Изменение среднего значения задержки при увеличении входной интенсивности для случая, когда пользователи решают перенести свои работы на серверы новой локации с вероятностью 0,5

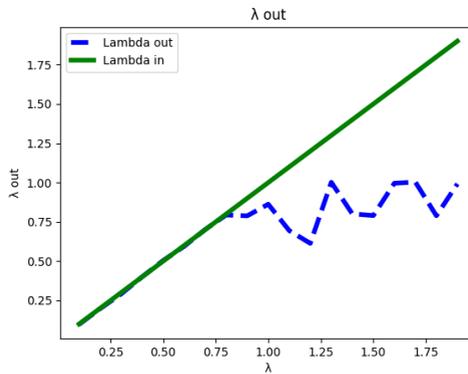


Рис. 11. Изменение выходной интенсивности системы при изменении входной интенсивности для случая, когда пользователи решают перенести свои работы на серверы новой локации с вероятностью 0,2

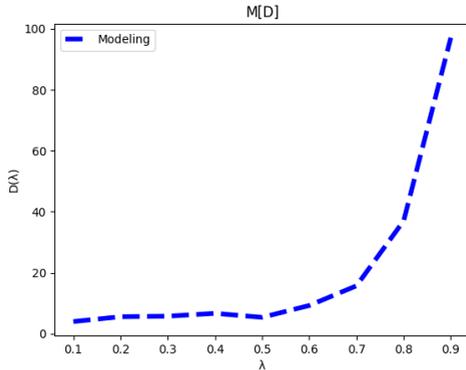


Рис. 12. Изменение среднего значения задержки при увеличении входной интенсивности для случая, когда пользователи решают перенести свои работы на серверы новой локации с вероятностью 0,2

Заключение

В работе представлена модель удаленного беспроводного доступа пользователя к распределенным вычислительным ресурсам. Данная модель в частном случае, когда пользователи не перемещаются между локациями и при заданных параметрах работы, демонстрирует выходные данные, соответствующие системе с пакетной обработкой. Также при рассмотрении варианта, когда пользователи перемещаются между областями, был обнаружен эффект, когда при более низких значениях интенсивности и большой вероятности переноса задачи пользователем задержка выше, чем в случаях, когда входная интенсивность выше. Данный эффект не характерен для систем типа пакетной обработки. Дальнейшим направлением нашей работы будет добавление в систему ряда параметров, которые бы позволили приблизить данную модель к работе реальной сети.

Библиографический список

1. Модель использования вычислительных ресурсов при перемещении пользователя в сетях 5G / Д. А. Васильев, А. М. Тюрликов, П. Ван Де Вен, В. В. Шнеер. URL: <https://www.elibrary.ru/item.asp?id=45480255> (дата обращения: 15.05.2021).
2. Ouyang T., Zhou Z., Chen X. Follow Me at the Edge: Mobility-Aware Dynamic Service Placement for Mobile Edge Computing // IEEE J. on Selected Areas in Communications. 2018. № 36. P. 2333–2345.
3. Mach P., Becvar Z. Mobile Edge Computing: A Survey on Architecture and Computation Offloading // IEEE Communications Surveys & Tutorials. 2017. № 19. P. 1628–1656.
4. Borst S. C., Proutiere A., Hegde N. Capacity of wireless data networks with intra-and inter-cell mobility // INFOCOM. 2006.
5. Networks and devices for the 5G / B. Bangerter, S. Talwar, R. Arefi, K. Stewart. // IEEE Communications Magazine. 2014. № 52. P. 90–96.
6. Ko S.-W., Han K., Huang K. Wireless Networks for Mobile Edge Computing: Spatial Modeling and Latency Analysis // IEEE Transactions on Wireless Communications. 2018. № 17. P. 5225–5240.
7. Ksentini A., Taleb T., Chen M. A Markov Decision Process-based Service Migration Procedure for Follow Me Cloud // 2014 IEEE Intern. Conf. on Communications (ICC). 2014. P. 1350–1354.
8. Mobility-Induced Service Migration in Mobile Micro-Clouds / S. Wang, R. Uргаonkar, T. He [et al.] // 2014 IEEE Military Communications Conf. 2014. P. 835–840.
9. Клейнрок Л. Вычислительные системы с очередями. М.: Мир, 1979. 198 с.

УДК 654.021.8

А. Ю. Глушенкова

студент кафедры инфокоммуникационных систем

А. М. Тюрликов – доктор технических наук, профессор – научный руководитель**ИССЛЕДОВАНИЕ АЛГОРИТМОВ МНОЖЕСТВЕННОГО ДОСТУПА
С ИСПОЛЬЗОВАНИЕМ ОРТОГОНАЛЬНЫХ ПРЕАМБУЛ В СИСТЕМАХ ИНТЕРНЕТА ВЕЩЕЙ**

На данный момент ведутся исследования различных алгоритмов множественного доступа с использованием ортогональных преамбул (TDMA). В работах [1, 2] приводятся различные реализации такого типа алгоритмов. В данной статье будет рассмотрен алгоритм множественного доступа с использованием ортогональных преамбул и временным разделением.

Опишем модель исследуемого алгоритма случайного множественного доступа с использованием заданного числа ортогональных преамбул. Преамбулой называется сигнал, позволяющий базовой станции определить абонента. Ортогональные преамбулы – это преамбулы, которые возможно различить при наложении сигналов. Главный принцип TDMA – разделение канала на временные промежутки между абонентами.

Для описания системы введем ряд допущений, при которых будут рассматриваться работа алгоритма и рассчитываться характеристики системы:

Допущение 1. Задано число абонентов K . Все абоненты имеют одинаковую длину сообщения. Интервал времени, в пределах которого происходит выбор преамбул и передача сообщений, называется фреймом. Каждый фрейм разбит на окна, длительность которых соответствует одной единице времени. Абоненты точно знают моменты разделения окон и могут начинать передачу только в начале окна.

Допущение 2. В окне возможны три события:

– «Конфликт» – событие, когда в одном окне одновременно передают два или более абонента. Предполагается, что при наложении сигналов сообщения полностью искажаются и не могут быть правильно декодированы базовой станцией;

– «Успех» – событие, когда в одном окне передает только один абонент. В таком случае базовая станция успешно декодирует сообщение от единственного передающего абонента;

– «Пусто» – событие, когда в одном окне ни один из абонентов не передает.

Допущение 3. В данной модели параметры модуляции, с помощью которой происходит передача, не влияют на систему.

Допущение 4. В канале отсутствуют шумы, т. е. передача данных по прямому и обратному каналу происходит безошибочно, таким образом, в конце окна абоненты наблюдают выход канала и достоверно определяют, какое событие произошло в окне.

Допущение 5. Количество появляющихся у абонентов сообщений в окне распределяется по пуассоновскому закону с параметром λ/D , где λ – интенсивность входного потока, D – длина сообщения в битах.

Допущение 6. Для алгоритма задано число ортогональных преамбул L . Базовая станция достоверно определяет номера преамбул, которые передавались вне зависимости от события, которое произошло в окне. Каждый абонент с готовым для передачи сообщением выбирает преамбулу только во время отведенного для выбора преамбул интервала, который равен L окон.

Можно сказать, что модель системы довольно простая.

Учитывая допущения, опишем работу алгоритма с L ортогональными преамбулами и длиной сообщения, равной D бит [3].

1. Абонент при появлении сообщения дожидается интервала для выбора преамбулы.
2. Во время интервала для выбора преамбул, который длится L окон, абонент выбирает одну из преамбул. Во время этого интервала у абонентов могут появляться сообщения.
3. Если все абоненты выбрали себе преамбулы, то базовая станция выделяет $K \cdot D$ окон, в течение которых абоненты будут передавать, где K – число разных выбранных преамбул.

4. По окончании интервала для выбора преамбул начинается интервал для передачи сообщений, в конце каждого D окна абоненты будут передавать сообщения в порядке возрастания их преамбул. Во время интервала передачи у абонентов могут появляться сообщения. Если абоненты выбрали одинаковую преамбулу, то происходит конфликт при передаче, и абоненты будут заново выбирать преамбулы во время следующего интервала для выбора преамбул.

5. По окончании интервала для передачи начинается интервал для выбора преамбул. Если абонентов с сообщениями в начале этого интервала нет, то в течение L окон сообщения могут только появляться. После окончания интервала для выбора преамбул, в котором никто не выбирал преамбулу, заново начинается этот интервал.

Рассмотрим работу алгоритма на примере. Пусть в системе 5 абонентов, доступно 5 преамбул, длина сообщения 2 бита [4].

Алгоритм начинает работу с интервала выбора преамбул (P), но поскольку на начало работы алгоритма абоненты (A) не имеют сообщений, то в течение 5 окон они генерируют сообщения. Пусть у $A1, A2, A4$ появились сообщения. Начинается интервал выбора преамбул.

Рассмотрим событие «Успех» (рис. 1). Пусть $A4$ выбирает преамбулу $P1, A1 - P3, A2 - P5$. Тогда интервал для передачи будет длиться $3D$ окон, т. е. 6 окон.

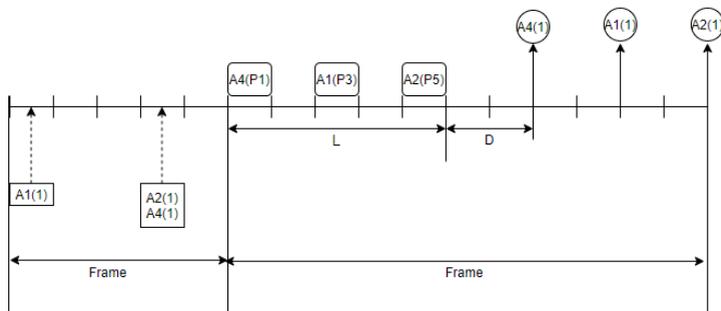


Рис. 1. Пример работы алгоритма (событие «Успех»)

Рассмотрим событие «Конфликт» (рис. 2). Пусть $A1$ и $A4$ выбрали $P1$ и $A2$ выбрал $P3$. Тогда $A1$ и $A4$ при передаче сообщений попадут в конфликт и не передадут свои сообщения, они будут передавать сообщения заново. $A2$ успешно передаст сообщение.

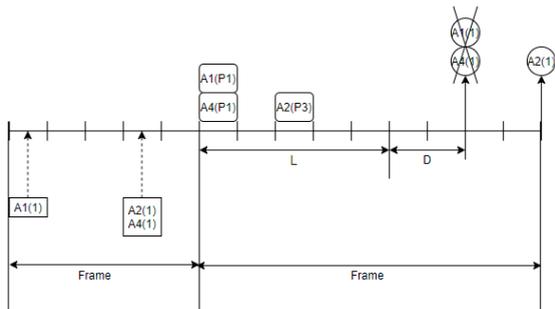


Рис. 2. Пример работы алгоритма (событие «Конфликт»)

Введем понятие интенсивности. Интенсивность – это число бит, которое поступает от всех абонентов в единицу времени, при этом все абоненты имеют сообщения [5].

Проведем анализ критической интенсивности системы. Критическая интенсивность – это интенсивность, при которой система работает стабильно, т. е. входная интенсивность равняется выходной, иными словами, все поступившие сообщения выходят из системы.

Введем следующее обозначение:

$$X_i = \begin{cases} 1, & i\text{-ю преамбулу выбрал один абонент} \\ 0, & \text{иначе} \end{cases}$$

Найдем вероятность, что $X_i = 1$, т. е. вероятность того, что каждый абонент выберет уникальную преамбулу:

$$\Pr\{X_i = 1\} = K \frac{1}{L} \left(1 - \frac{1}{L}\right)^{K-1}$$

Определим математическое ожидание числа успехов за интервал передачи:

$$M\left[\sum_{i=1}^L X_i\right] = LK \frac{1}{L} \left(1 - \frac{1}{L}\right)^{K-1} = K \left(1 - \frac{1}{L}\right)^{K-1}$$

Найдем вероятность, что $X_i \neq 1$, т. е. вероятность того, что i -ю преамбулу выберет больше чем один абонент:

$$\Pr\{X_i \neq 1\} = 1 - K \frac{1}{L} \left(1 - \frac{1}{L}\right)^{K-1} - \left(1 - \frac{1}{L}\right)^K$$

Определим математическое ожидание числа конфликтов за интервал передачи:

$$M\left[\sum_{i=1}^L X_i\right] = L \left(1 - K \frac{1}{L} \left(1 - \frac{1}{L}\right)^{K-1} - \left(1 - \frac{1}{L}\right)^K\right)$$

Рассчитаем критическую интенсивность. Для этого необходимо число успешно вышедших битов поделить на длину интервала, который состоит из интервала для выбора преамбул и интервала передачи:

$$\lambda_{кр} = \frac{DK \left(1 - \frac{1}{L}\right)^{M-1}}{\left(K \left(1 - \frac{1}{L}\right)^{K-1} + L \left(1 - K \frac{1}{L} \left(1 - \frac{1}{L}\right)^{K-1} - \left(1 - \frac{1}{L}\right)^K \right) \right) D + L}$$

После расчета значений можно сравнить результаты с моделированием (рис. 3–5).

Отметим, что критическая интенсивность выше, если число преамбул превышает число абонентов, поскольку в системе происходит меньше конфликтов [6].

Рассчитать задержку теоретически достаточно трудно, поэтому оценим ее с помощью имитационного моделирования. Для поиска минимальной задержки при заданном числе абонентов, входной интенсивности и длине сообщения, используем следующий подход.

1. Найдем значения критической интенсивности при заданном числе абонентов $K = 10$, входной интенсивности $\lambda = 0,8$ и длине сообщения $D = 5$, изменяя число преамбул L . На рис. 6 изображен график, построенный по этим значениям.

2. Зафиксируем входную интенсивность, которая должна быть меньше самого максимального значения. Допустим, входная интенсивность равняется 0,5. По графику видно, что значения 0,5 дости-

гается при двух значениях преамбул – 13 и 30. Также по графику можно определить число преамбул, при котором будет максимальная входная интенсивность.

3. Найдем значения задержки для $K = 10$, $\lambda = 0,5$, $D = 5$ и $L \in [13;30]$. График на рис. 7 соответствует этим значениям. По нему можно определить число преамбул, при котором будет минимальная задержка для заданных параметрах.

Таким образом, с помощью моделирования получена оценка минимальной задержки системы для конкретных параметров [7].

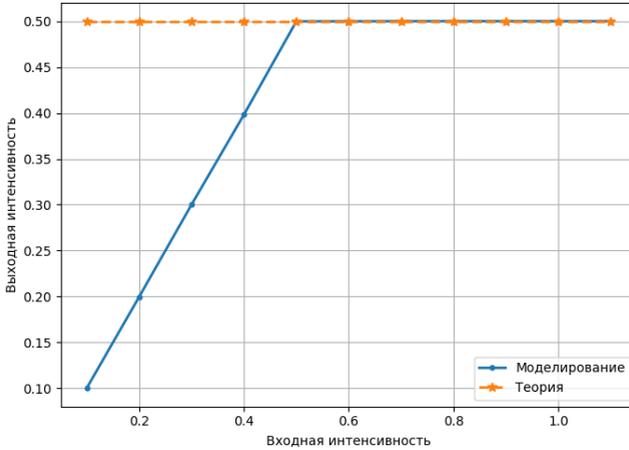


Рис. 3. Пример работы системы для параметров $K = 1$, $L = 1$, $D = 1$

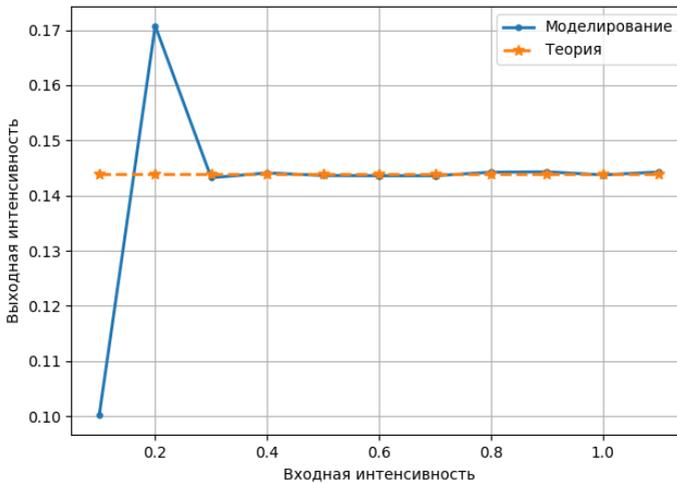


Рис. 4. Пример работы системы для параметров $K = 20$, $L = 10$, $D = 1$

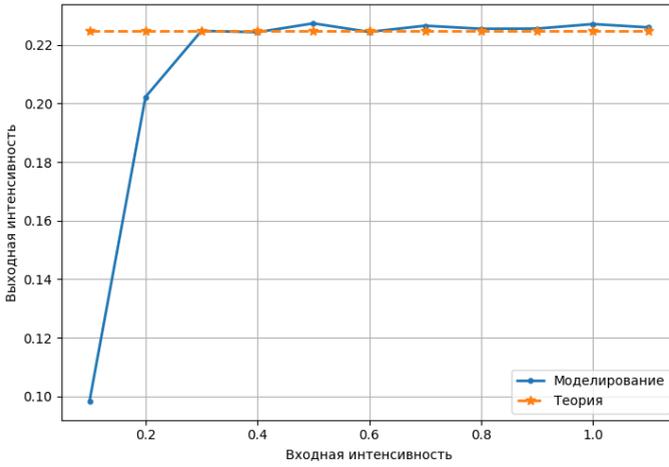


Рис. 5. Пример работы системы для параметров $K = 10, L = 20, D = 1$

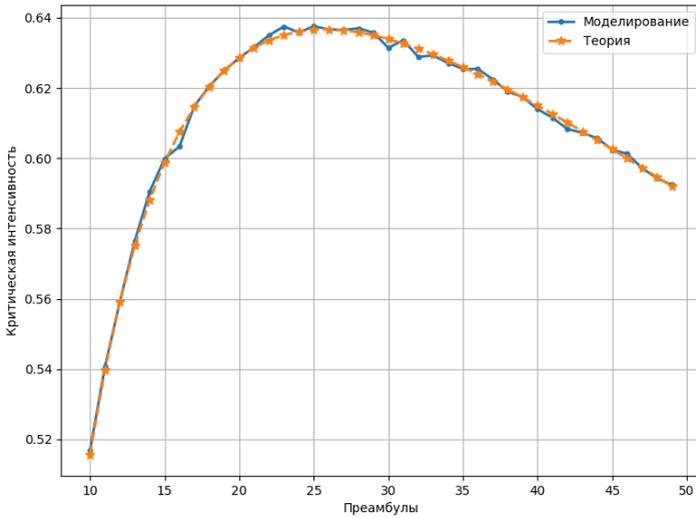


Рис. 6. График критической интенсивности для $K = 10, D = 10$ и $L \in [10;50]$

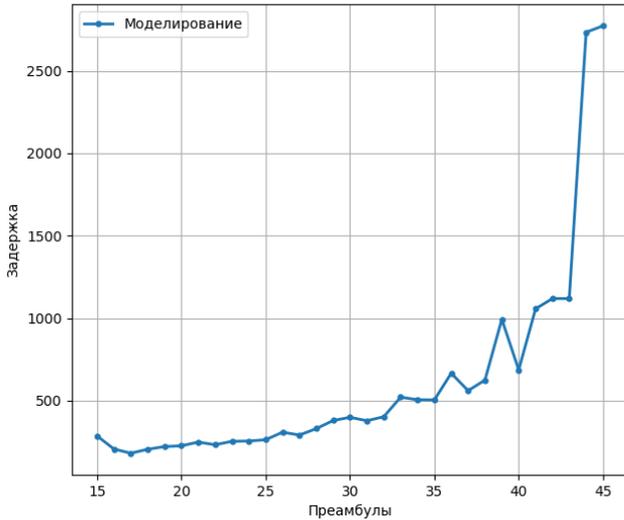


Рис. 7. График задержки для $K = 10$, $\lambda = 0.5$, $D = 10$ и $L \in [15; 45]$

Рассмотрен алгоритм множественного доступа с временным разделением в сочетании с ортогональными преамбулами. В статье приведена простая модель без шумов. Для системы рассчитаны критическая интенсивность, число преамбул для максимального значения выходной интенсивности и оценена минимальная задержка для системы с заданными параметрами. При дальнейшем исследовании в систему будет добавлен шум, который повлияет на ее характеристики.

Библиографический список

1. Бурков А. А., Тюрликов А. М. Основы построения инфокоммуникационных систем и сетей: лаборат. практикум. СПб.: ГУАП 2018. 50 с.
2. Performance analysis and location optimization for massive MIMO systems with circularly distributed antennas / A. Yang, Y. Jing, C. Xing [et al.] // IEEE Trans. Wireless Commun. 2015. Vol. 14, № 10. P. 5659–5671.
3. Falin G. I. Performance Evaluation for a Class of Algorithms of Random Multiple Access to a Radio-Communication Channel // ProblemyPeredachiInformatsii. 1982. № 18. P. 85–90.
4. Synchronization of TDMA bursts with short preamble for satellite receiver / B. Purushothaman, A. C. Veeresh, S. V. S. Prasad [et al.] // 2016 IEEE Int. Conf. on Advanced Networks and Telecommun. Systems.
5. Preamble design and detection for 5G enabled satellite random access / L. Zhen, T. Sun, G. Lu [et al.] // IEEE Access. 2020. Vol. 8. P. 49873–49884.
6. Multiple preambles for high success rate of grant-free random access with massive MIMO / H. Jiang, D. M. Qu, J. Ding, T. Jiang // IEEE Trans. Wireless Commun. 2019. July. P. 1–12.
7. Choi J. On improving throughput of multichannel ALOHA using preamble-based exploration // J. of Communications and Networks. 2020. Vol. 22, Iss. 5. P. 380–389.

УДК 004.728.3.057.4

А. Р. Осиков

студент кафедры инфокоммуникационных систем

Н. В. Степанов – ассистент – научный руководитель

ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ДЛЯ ИССЛЕДОВАНИЯ СПОСОБОВ ПЕРЕДАЧИ В НЕЛИЦЕНЗИРОВАННЫХ ДИАПАЗОНАХ ЧАСТОТ

Введение

На сегодняшний день все активнее внедряются сети беспроводной передачи информации. Они используются для различных целей, таких как любительские, коммерческие, военные. Для контроля за разграничением частот с разной спецификой применения следит Министерство связи и массовой коммуникации Российской Федерации. В его состав входит государственная комиссия по радиочастотам (ГКРЧ). Цели данного подразделения:

- совершенствование нормативной правовой базы в области регулирования использования радиочастотного спектра;
- сближение национального распределения радиочастот с международным, обеспечение электромагнитной совместимости радиоэлектронных средств различного назначения;
- повышение эффективности использования радиочастотного спектра, включая вопросы его конверсии.

В нашей работе главным вопросом применения радиочастотного спектра являются способы передачи сигналов в нелицензированных диапазонах частот. Для этого рассматриваются основные проблемы, связанные с формированием сигналов, излучением в канал, фиксацией на приемной стороне и обработка полученных результатов. Для простоты использования и во избежание перекрытий радиочастотных каналов на начальном этапе создания программно-аппаратного комплекса сигналы передаются по коаксиальному кабелю от передатчика к приемнику. В дальнейшем планируется учитывать ограничения ГКРЧ при использовании беспроводного канала в учебном процессе.

Главная задача – научиться пользоваться оборудованием генерации и фиксации сигналов. Второстепенная – обеспечение возможности формирования сигналов в цифровом виде. Поэтому концепция схемы построения системы строится на моделировании исходных сигналов с помощью персонального компьютера (ПК). Такой способ формирования сигналов прост для внедрения в учебный процесс и не требует большого финансирования, так как следующей задачей изучения процессов передачи сигналов является множественный доступ, где большое число физических источников тяжело синхронизировать.

1. Модель системы

Схематическая модель системы представлена на рис. 1. Она состоит из персонального компьютера (ПК), векторного генератора сигналов (ГС) agilent E4438C, SDR, анализатора спектра agilent 4405B и осциллографа agilent 54853A DSO [1].

В первую очередь на ПК в программе matlab задается исходный сигнал и подаются остальные входные данные. Далее формируется массив данных в формате int16, соответствующий дискретным отсчетам заданного сигнала, при этом отсчеты реальной и мнимой части сигнала чередуются. Структура массива показана на рис. 2.

Передача данных с ПК на генератор происходит следующим образом: сигнал, который формируется в компьютере при помощи программы matlab, отправляется через кабель, которым соединены генератор и ПК.

Данные, требуемые для того, чтобы генератор сигналов корректно принял, обработал и начал излучать сигнал, следующие: несущая частота, на которой будет работать генератор, массив данных в формате int16, а также частота дискретизации [2].

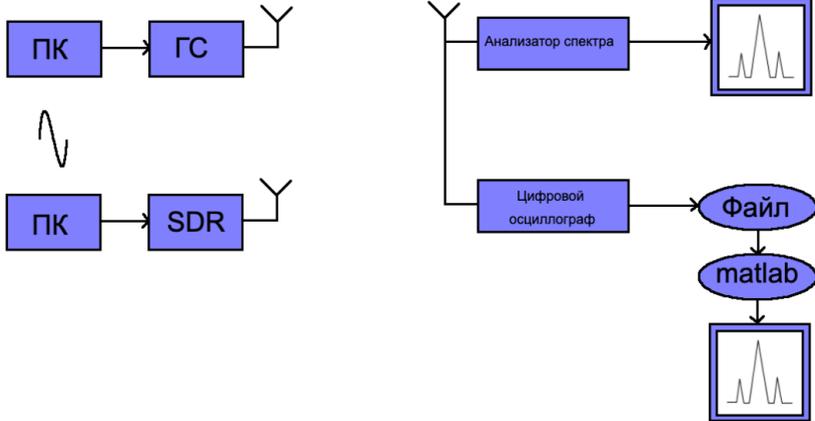


Рис. 1. Схематическая модель системы

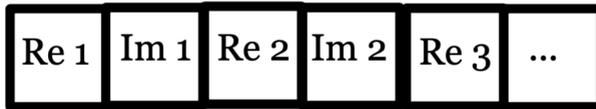


Рис. 2. Структура массива данных

После успешного приема данных, генератор сигналов начинает постоянно повторять заданный сигнал. Далее, при помощи кабеля или по радиоканалу можно передать сигнал на анализатор спектра и осциллограф и таким образом отследить корректность работы модели [3].

- Векторный генератор сигналов agilent E4438C – может генерировать сигнал на частотах от 250 kHz до 6 GHz [1].
- Анализатор спектра agilent 4405B – может работать со спектром сигнала на частотах от 9 kHz до 13.2 GHz. [2]
- Осциллограф agilent 54853A DSO – максимальная частота распознавания сигнала, равная 2,5 GHz. Частота сэмпирования 20 GSa/s [3].

2. Результаты

Во всех тестах была выставлена несущая частота равная 3 МГц.

На рис. 3–5 результат первого теста, в котором в ГС подается, синусоида с частотой 200 кГц, амплитуда которой меняется от 0 до 1 и только в реальной части, а в мнимой нули.

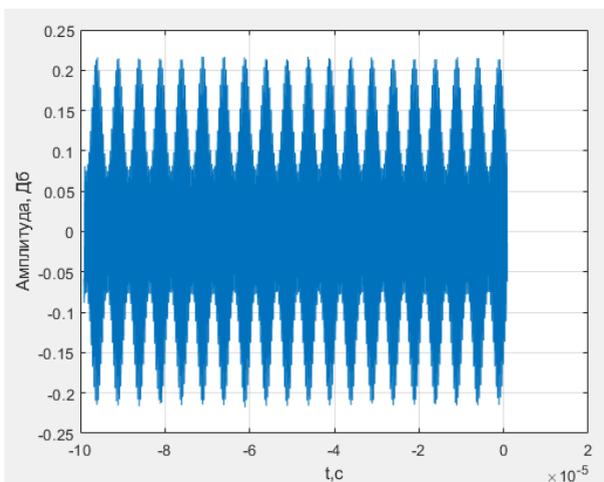


Рис. 3. Файл с осциллографа при первом тесте

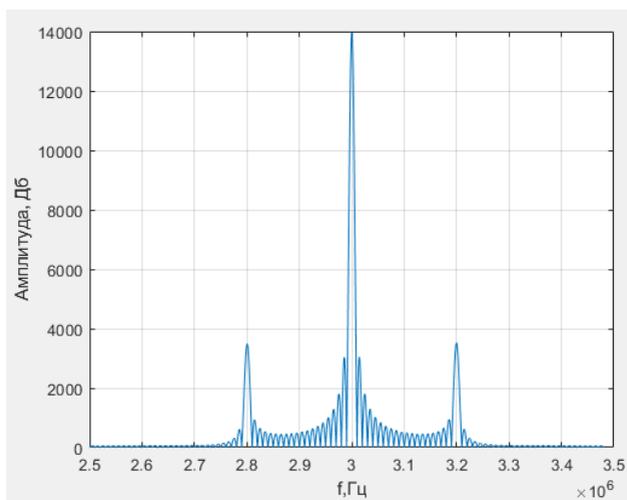


Рис. 4. Спектр сигнала, посчитанный программой matlab

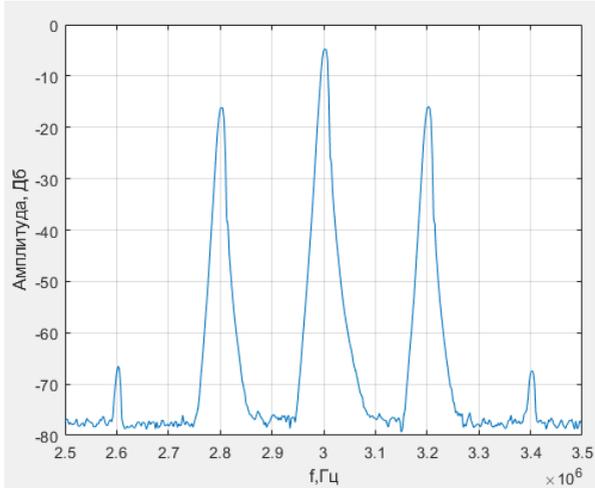


Рис. 5. Спектр сигнала, взятый с анализатора спектра

Во втором тесте аналогичный первому сигнал был подан в SDR. На рис. 6–9 можно увидеть разницу между качеством сигнала по сравнению с ГС.

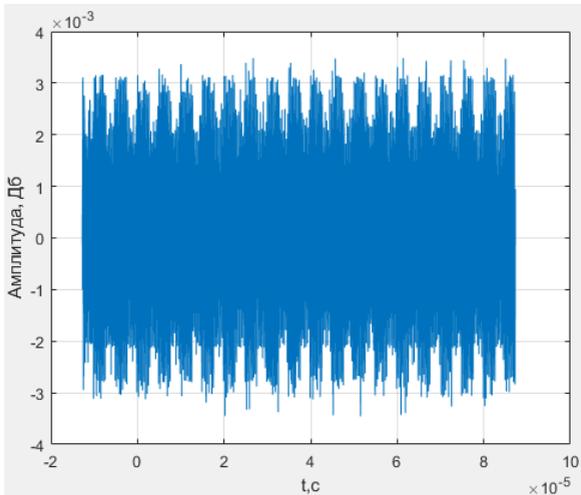


Рис. 6. Файл с осциллографа при втором тесте

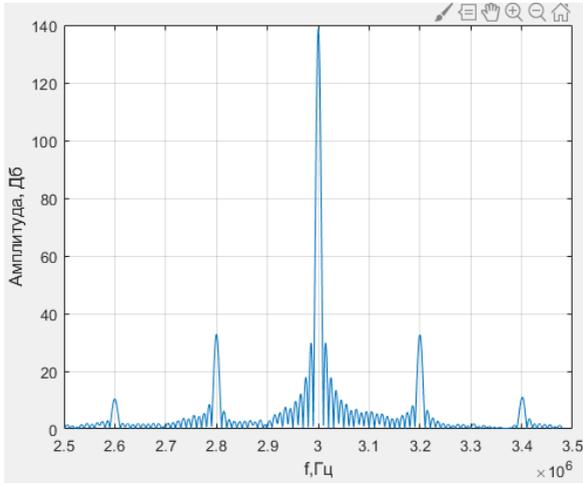


Рис. 7. Спектр сигнала, посчитанный программой matlab

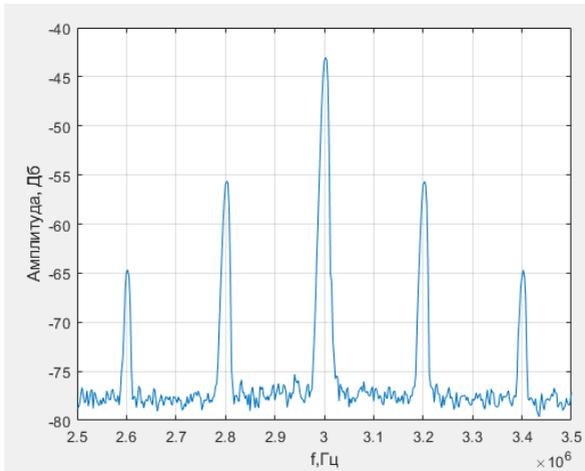


Рис. 8. Спектр сигнала, взятый с анализатора спектра

Также на ГС была подана сумма двух синусоид в реально и мнимой части с частотой 200 и 300 кГц. При таких входных данных ГС использует однополосную модуляцию, что хорошо видно на рис. 9–11.

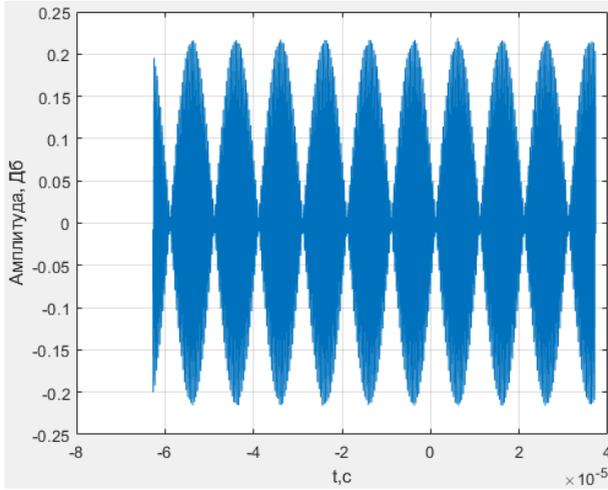


Рис. 9. Файл с осциллографа при третьем тесте

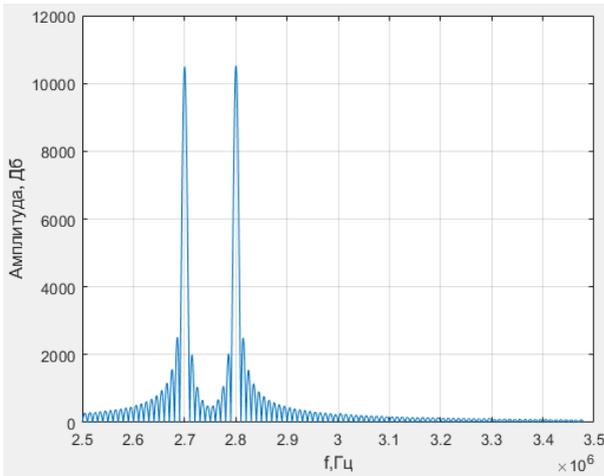


Рис. 10. Спектр сигнала, посчитанный программой matlab

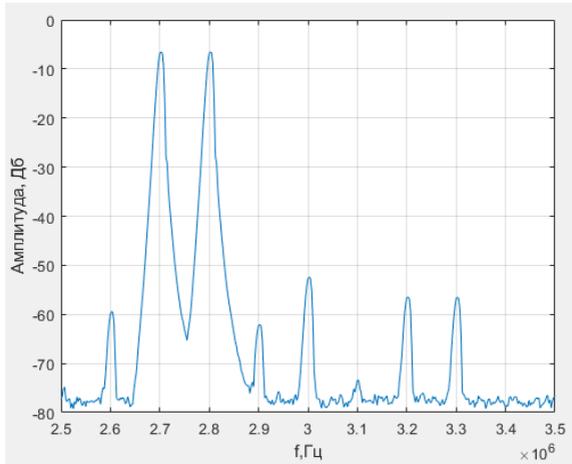


Рис. 11. Спектр сигнала, взятый с анализатора спектра

Заключение

Были рассмотрены простые сигналы, изучено использование оборудования, которое позволяет проверять, насколько правильно они сгенерированы с помощью программы. По результатам, полученным при тестировании, можно сделать вывод, что на выходе из ГС получается более качественный сигнал, чем на более дешевом и компактном устройстве – SDR. Данный программно-аппаратный комплекс позволит изучить известные открытые технологии IoT, где важным вопросом является масштабируемость сети.

Библиографический список

1. Agilent Technologies. Agilent E4438C ESG Vector Signal Generator. URL: <https://www.keysight.com/ru/ru/assets/7018-01039/data-sheets/5988-4039.pdf> (дата обращения: 14.05.2021).
2. Keysight Technologies. ESA-E Series Spectrum Analyzer. URL: <https://www.keysight.com/ru/ru/assets/7018-01953/data-sheets/5989-9815.pdf> (дата обращения: 14.05.2021).
3. Agilent Technologies. Agilent Model 54853A, 54854A, and 54855A Infiniium Oscilloscopes. URL: <https://literature.cdn.keysight.com/litweb/pdf/54855-97008.pdf> (дата обращения: 14.05.2021).

УДК 004.7

Р. О. Рачугин

студент кафедры инфокоммуникационных систем

А. А. Бурков – ассистент – научный руководитель**ИССЛЕДОВАНИЕ МНОГОКАНАЛЬНОГО АЛГОРИТМА АЛОХА С ФАЗОЙ ИССЛЕДОВАНИЯ**

В настоящее время увеличивается количество устройств и датчиков Интернета вещей. Для поддержки этих устройств и датчиков используется многоканальный алгоритм АЛОХА, который имеет ограниченную пропускную способность. Для ее увеличения предлагаются различные методы. Например, метод последовательного погашения интерференции или метод, основанный на принципе неортогонального множественного доступа. Предложенные методы требуют больших вычислительных ресурсов на базовой станции. В данной работе рассматривается другой метод, основанный на использовании ортогональных преамбул. Он может быть использован, когда базовая станция не обладает большими вычислительными ресурсами. При этом она должна быть способна определить количество переданных преамбул в каждом канале.

Описание моделей, используемых в работе**Модель системы 1**

Модель базируется на следующих допущениях.

Допущение 1: все сообщения имеют одинаковую длину, одинаковую скорость передачи, время передачи сообщения принимается за единицу времени. Все время передачи по каналу разбито на окна, длительность одного окна принята за единицу времени. Абоненты знают границы разделения окон и могут начинать передачу только в начале следующего окна.

Допущение 2: в каждом окне может произойти одно из следующих событий:

- событие «Успех». В окне передает один абонент. В этом случае считается, что абонент успешно передает сообщение;
- событие «Пусто». Ни один из абонентов не передает сообщение в текущем окне;
- событие «Конфликт». В окне одновременно передают два и более абонентов. Сообщения абонентов накладываются друг на друга и не могут быть обработаны на приемной стороне.

В конце каждого окна все абоненты достоверно узнают о том, какое событие произошло в канале.

Допущение 3: система имеет некоторый набор K независимых каналов. Известно количество абонентов в окне. Абоненты случайным образом выбирают один из каналов.

Допущение 4: система имеет M абонентов. На вход системы поступает пуассоновский входной поток интенсивности λ . Интенсивность у всех абонентов в системе одинаковая и равна λ/M . У каждого абонента есть неограниченный буфер, в который добавляются новые заявки абонента.

Описание алгоритма

Пользователь с готовым сообщением случайным образом решает, будет ли он передавать в этом окне или нет. Вероятность передачи $p = K / M$. Далее, если абонент решил передавать, он выбирает один из K каналов, после чего идет передача сообщений через выбранный канал. В конце окна все абоненты достоверно узнают какие события произошли в каждом канале.

Пример работы системы приведен на рис. 1.

В системе имеется $M = 3$ активных абонентов с несколькими готовыми сообщениями в буфере и $K = 3$ каналов передачи данных. В первом окне абонент U_1 выбирает третий канал, абонент U_2 – первый канал, абонент U_3 – второй, в результате во всех каналах произойдет событие «Успех». Во втором окне абонент U_1 выбирает третий канал, абоненты U_2 и U_3 – первый. В результате в первом канале произойдет событие «Конфликт», во втором – «Пусто», в третьем – событие «Успех».

Теоретический расчет пропускной способности многоканального алгоритма АЛОХА

Среднее количество пакетов без коллизий находится по следующей формуле:

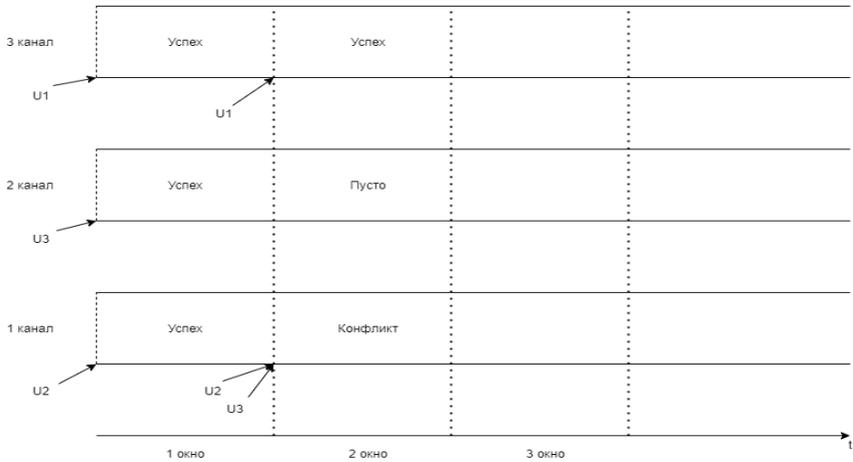


Рис. 1. Пример работы многоканального алгоритма АЛОХА

$$\bar{N}(M, K) = M \left(1 - \frac{1}{K} \right)^{M-1}, \quad (1)$$

где M – количество абонентов, K – количество каналов.

При больших M уравнение (1) будет иметь следующий вид:

$$\bar{N}(M, K) \approx M e^{-\frac{M}{K}}. \quad (2)$$

Максимальная пропускная способность η может быть достигнута при $K = M$:

$$\bar{N}_{ma}(K) = K e^{-1}. \quad (3)$$

Модель системы 2

Модель базируется на следующих допущениях.

Допущение 1: все сообщения имеют одинаковую длину, одинаковую скорость передачи, время передачи сообщения принимается за единицу времени. Все время канала разбито на фреймы. Фрейм отличается от окна тем, что имеет дополнительную фазу исследования перед отправкой сообщения. Время одного фрейма принято за единицу времени. Абоненты знают границы фрейма и могут передавать сообщения в начале нового фрейма.

Допущения 2–4 совпадают с аналогичными допущениями модели 1.

Допущение 5: все абоненты имеют уникальные преамбулы. Преамбула – это последовательность бит некоторой длины, которая много меньше длины сообщения. Скорость передачи преамбул одинаковая. Базовая станция достоверно может определить количество преамбул в каждом канале.

Описание работы алгоритма (рис. 2)

Передача разбивается на две фазы: фазу исследования и фазу передачи данных. Длительность фазы исследования меньше длительности фазы передачи данных и предполагается, что она равна 0 единиц времени. Длительность фазы передачи данных равна длительности передачи данных модели 1.

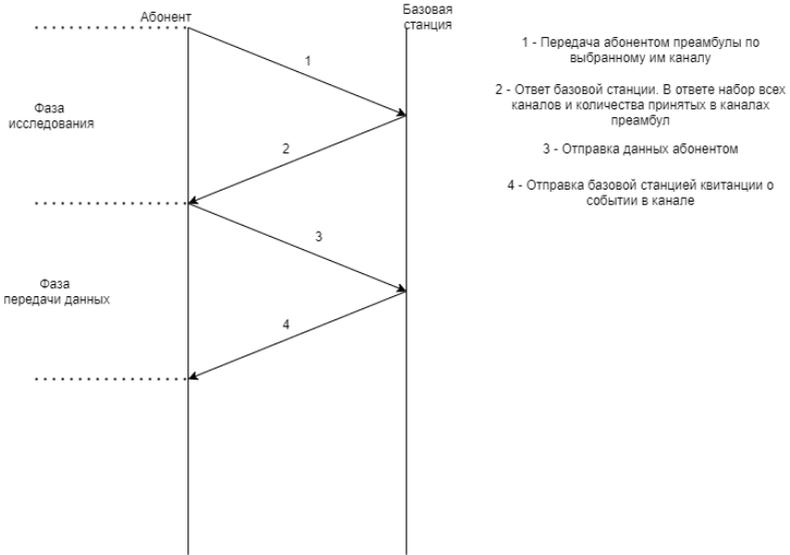


Рис. 2. Описание структуры фрейма

В фазе передачи данных абоненты случайным образом решают, будут ли они передавать в текущем фрейме или нет. Вероятность передачи $p = K / M$. Далее, если абонент решил передавать, он выбирает один из K каналов, после чего идет передача преамбулы через выбранный канал (предполагается, что у каждого абонента уникальная преамбула). Базовая станция принимает преамбулы и оценивает их количество в каждом канале. По каналу обратной связи базовая станция передает эту информацию пользователям, после чего каждый пользователь смотрит, к какой группе принадлежит канал, через который он передавал свою преамбулу. Имеются две группы. Первая включает абонентов без конфликта, обозначим ее S (некоторый набор каналов из множества K). Вторая группа включает абонентов в конфликте обозначим ее S^c (дополнение к множеству S) [1, 2].

Затем следует фаза передачи данных. Абоненты из первой группы передают данные с вероятностью, равной единице. Абоненты из второй группы случайным образом решают, передавать им сообщение или нет. Вероятность передачи $p_{dtp} = \min\left\{1, \frac{L}{W}\right\}$, где L – количество каналов, которые доступны для передачи для группы 2. W – число активных пользователей в группе 2 (S^c). Далее, если абонент решил передавать, он выбирает один из L каналов, после чего идет передача сообщений через выбранный канал. В конце фазы передачи данных все абоненты достоверно узнают, какие события произошли в каждом канале.

Пример работы алгоритма

В системе имеется $M = 5$ активных абонентов и $K = 3$ каналов передачи данных (рис. 3). В первом фрейме пользователи $U1$ и $U2$ выбрали третий канал, пользователь $U5$ выбрал второй канал, пользователи $U3$ и $U4$ – первый. После фазы исследования пользователь $U5$ принадлежит группе 1, а пользователи $U1, U2, U3$ и $U4$ – группе 2. Каналы один и три принадлежат S^c , следовательно, пользователи $U1, U2, U3$ и $U4$ перебирают каналы. Допустим, что пользователи $U1, U2$ решили передавать и выбрали третий канал, пользователь $U3$ решил передавать и выбрал первый канал, а пользователь $U4$ решил не передавать. В результате в каналах один и два произойдет событие «Успех», а в третьем – «Конфликт» [3].

Во втором фрейме пользователи U1, U5, U3 выбрали третий канал, а пользователи U4, U2 – канал один. После фазы исследования все пользователи принадлежат группе 2 и все каналы принадлежат множеству S^c . Далее все пользователи решают, будут ли они передавать. Допустим пользователи U1, U5 решили не передавать. Пользователи U3 и U2 решили передавать и перевыбрали второй канал. Пользователь U4 решил передавать и перевыбрал первый канал. В результате в первом канале произошло событие «Успех», во втором – «Конфликт», в третьем – «Пусто» [4].

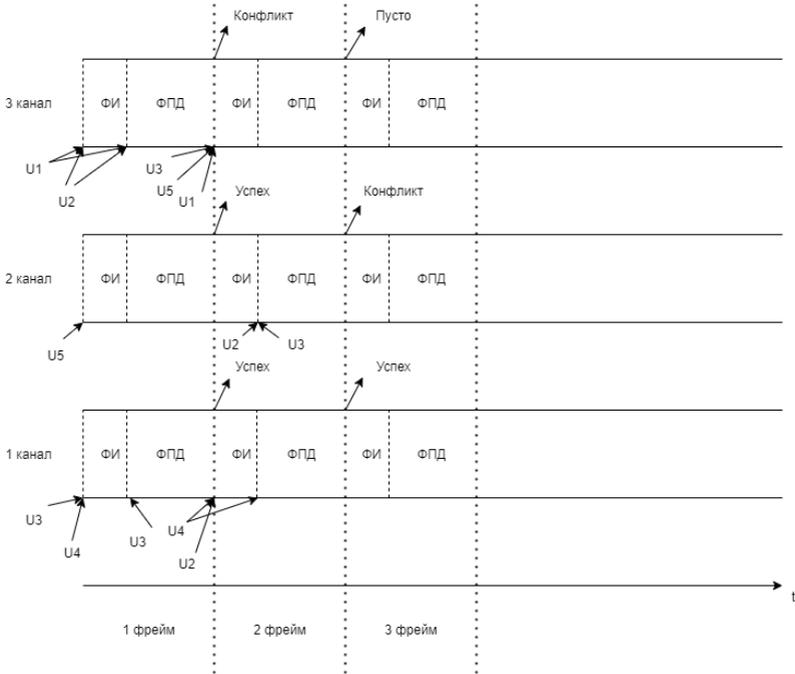


Рис. 3. Пример работы многоканального алгоритма АЛОХА с фазой исследования:
 ФИ – фаза исследования. ФПД – фаза передачи данных.

Теоретический расчет пропускной способности системы многоканальная АЛОХА с фазой исследования

Обозначим S количество активных пользователей группы 1:

$$S \leq \min\{M, K\}, \tag{4}$$

где M – число пользователей в системе, K – количество каналов.

Тогда получим, что W – количество активных пользователей в группе 2:

$$W = M - S. \tag{5}$$

Обозначим L количество каналов, принадлежащих множеству S^c :

$$L = K - S. \tag{6}$$

Обозначим U количество абонентов из группы 2, которые решат передавать.

Среднее количество пакетов без коллизий для группы 2 находится по следующей формуле:

$$\bar{N}_{ep2}(U, L) = U \left(1 - \frac{1}{L}\right)^{U-1}. \quad (7)$$

Среднее количество пакетов для системы многоканальная АЛОХА с фазой исследования можно рассчитать по следующей формуле:

$$\bar{N}_{ep}(U, L, S) = S + U \left(1 - \frac{1}{L}\right)^{U-1}. \quad (8)$$

Значения U , L , S зависят от значений K и M :

$$\bar{N}_{ep}(M, K) = \bar{N}(S | M) + \bar{N}_{ep2}(U, L | M). \quad (9)$$

При больших значениях M и K значения L и W также будут велики. Из выражения (8) можно увидеть, что

$$\bar{N}(S | M) = M \left(1 - \frac{1}{K}\right)^{M-1}. \quad (10)$$

Верхняя оценка \bar{N}_{ep2} будет иметь следующий вид:

$$\bar{N}_{ep2}(L) = L e^{-1}. \quad (11)$$

Подставляя (6) в (11), получим следующее:

$$\bar{N}_{ep2}(L) = (K - \bar{N}(S | M)) e^{-1}. \quad (12)$$

Далее, подставив (10) и (12) в (9), получим максимальную пропускную способность:

$$\bar{N}_{ep}(K) = \bar{N}(S | M) + (K - \bar{N}(S | M)) e^{-1} = K e^{-1} + \bar{N}(S | M) (1 - e^{-1}). \quad (13)$$

При больших значениях M $\bar{N}(S | M) \approx K e^{-1}$. Тогда максимальная пропускная способность η многоканальной АЛОХА с фазой исследования равна:

$$\bar{N}_{ep}(K) = K e^{-1} + K e^{-1} (1 - e^{-1}) = K e^{-1} (2 - e^{-1}). \quad (14)$$

Сравнивая две системы, можно сделать вывод, что многоканальная АЛОХА с фазой исследования дает выигрыш в пропускной способности равный:

$$\varphi = \frac{\bar{N}_{ep}(K)}{\bar{N}_{ma}(K)} = \frac{K e^{-1} (2 - e^{-1})}{K e^{-1}} = 2 - e^{-1}. \quad (15)$$

Заключение

На рис. 4 представлено сравнение выходной интенсивности для описанных двух моделей, на рис. 5 – график задержки на один канал двух моделей, на рис. 6 – график среднего числа заявок для одного канала двух моделей. Данные результаты были получены для моделей с одним каналом. В результате можно сделать вывод, что пропускная способность модели 2 в 1,35 раза больше, чем модели 1. Как показано на рис. 7, с увеличением числа каналов пропускная способность модели 1 не изменяется, а пропускная способность модели 2 стремится к значению 0,6. Следовательно, выигрыш модели 2 стремится к значению 1,6, что было доказано теоретически в формуле (15). Также показатель задержки для модели 2 ниже аналогичного показателя модели 1. Дальнейшим направлением исследований станет попытка увеличения стоимости фазы исследования и ограничения числа преамбул для приближения к реальным системам.

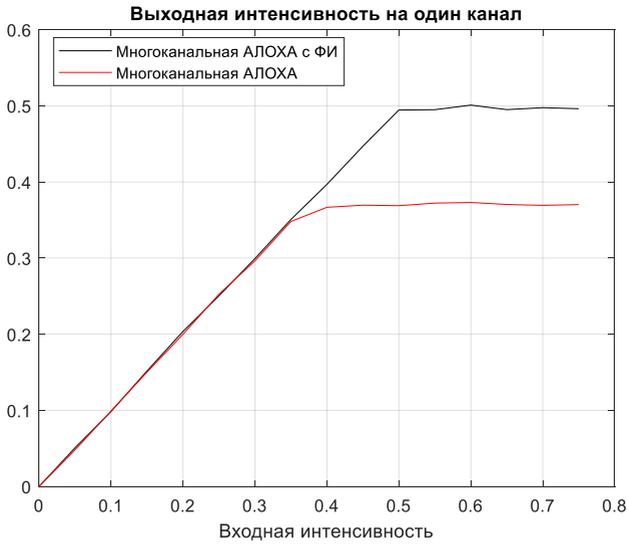


Рис. 4. График выходной интенсивности на один канал двух моделей

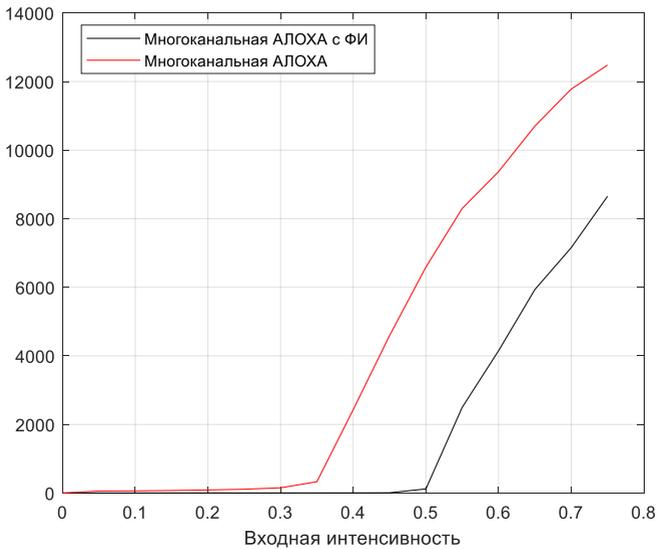


Рис. 5. График задержки на один канал двух моделей

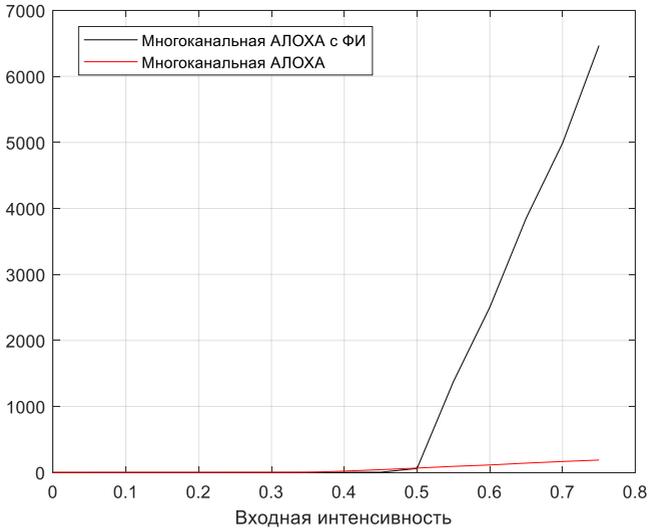


Рис. 6. График среднего числа заявок для одного канала двух моделей

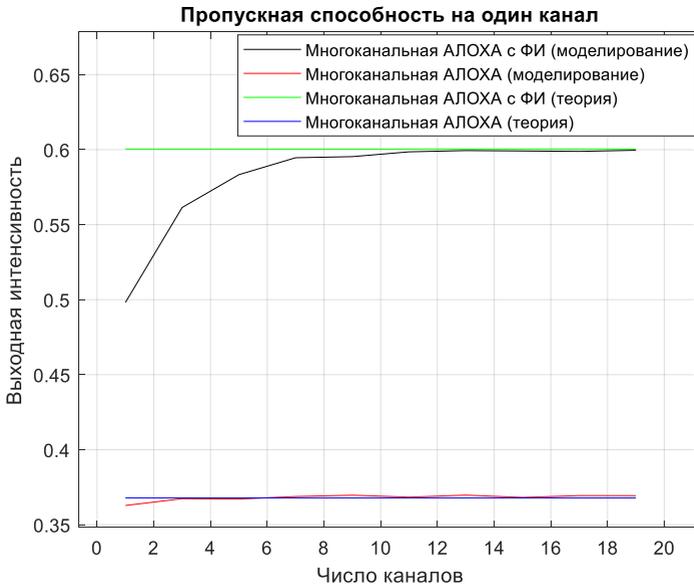


Рис. 7. График зависимости пропускной способности от числа каналов

Библиографический список

1. Тюрликов А. М., Пастушок И. А., Борисовская А. В. Основы построения инфокоммуникационных систем и сетей: учеб. пособие: в 2 ч. Ч. 1. СПб.: ГУАП, 2019. 111 с.
2. Choi J. On improving throughput of multichannel ALOHA using preamble-based exploration // *J. of Communications and Networks*. 2020. Vol. 22, Iss. 5. P. 380–389.
3. Shen D., Li V. O. K. Performance analysis for a stabilized multichannel slotted ALOHA algorithm // *Proc. IEEE PIMRC*. 2003. Vol. 1. P. 249–253.
4. Chang C. H., Chang R. Y. Design and analysis of multichannel slotted ALOHA for machine-to-machine communication // *Proc. IEEE GLOBECOM*. 2015. Dec. P. 1–6.

УДК 681.324

Ю. О. Фиклистова

студент кафедры инфокоммуникационных систем

А. М. Тюрликов – доктор технических наук, профессор – научный руководитель

**ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ КАНАЛА ПЕРЕДАЧИ СЛУЖЕБНОЙ ИНФОРМАЦИИ
В СЕТЯХ ПЯТОГО ПОКОЛЕНИЯ**

Сегодня беспроводные сети получили большое развитие в области передачи данных. Технология беспроводного доступа стала основой любой системы мобильной связи. Технология 5G NR (5th Generation New Radio) – это поколение мобильной связи, действующее на основе стандартов телекоммуникаций, следующих за существующей технологией 4G LTE (4th Generation Long-Term Evolution).

Радиочастотные ресурсы, передающие информацию с уровней выше физического, называются каналами. В 5G определены следующие каналы для восходящей и нисходящей передачи:

- PDSCH (Physical downlink shared channel) – для передачи данных в нисходящем канале;
- PDCCH (Physical downlink control channel) – для передачи служебной информации в нисходящем канале;
- PBCH (Physical broadcast channel) – для широковещательной передачи служебной информации, необходимой абоненту для доступа к сети;
- PUSCH (Physical uplink shared channel) – для передачи данных в восходящем канале;
- PUCCH (Physical uplink control channel) – для служебной информации в восходящем канале;
- PRACH (Physical random access channel) – для запроса абонентом (UE) доступа к сети.

Рассмотрим подробно канал PDCCH. Он несет назначение планирования, предоставления ресурсов, подтверждения получения пакетов. Данные такого типа передаются в блоках DCI (Downlink control information, служебная информация в нисходящем канале). На рис. 1 представлена схема канала передачи служебной информации.



Рис. 1. Схема нисходящего канала передачи служебной информации 5G

Сначала на вход поступают данные DCI, далее к ним добавляется CRC длиной 24 бита. После этого последние 16 бит полученного сообщения маскируются (производится битовая операция XOR) специальным RNTI (Radio Network Temporary Identifier, временный идентификатор радиосети). С помощью этого RNTI абонент определяет, для какого абонента предназначены данные DCI. После маскирования RNTI данные перемежаются, так что биты CRC распределяются между информационными битами. Этот перемежитель поддерживает максимальный размер входного сигнала 164 бита, что означает, что DCI без CRC может иметь максимальную длину 140 бит [1].

Для канального кодирования в нисходящем канале передачи служебной информации используется полярный код со следующими параметрами: $n_{\max} = 9$, $l_{IL} = 1$, $n_{PC} = 0$, $n_{PC}^{wm} = 0$. Назначение этих параметров будет показано далее. На вход кодера поступает последовательность $c_0, c_1, c_2, c_3, \dots, c_{K-1}$, где K – количество бит, которое необходимо закодировать. На выходе кодера получается последовательность $d_0, d_1, d_2, d_3, \dots, d_{N-1}$, где $N = 2^n$ и значение n определяется следующим образом [2].

1. Обозначим как E длину выходной последовательности после выравнивания скорости.
2. Если $E \leq (9/8) \cdot 2^{\lceil \log_2 E \rceil - 1}$ и $K/E < 9/16$, то $n_1 = \lceil \log_2 E \rceil - 1$.
3. Иначе $n_1 = \lceil \log_2 E \rceil$.
4. $R_{\min} = 1/8$.
5. $n_2 = \lceil \log_2 (K/R_{\min}) \rceil$.
6. $n = \max\{\min\{n_1, n_2, n_{\max}\}, n_{\min}\}$, где $n_{\min} = 5$.

Битовая последовательность $c_0, c_1, c_2, c_3, \dots, c_{K-1}$ перемежается в битовую последовательность $\dot{c}_0, \dot{c}_1, \dot{c}_2, \dot{c}_3, \dots, \dot{c}_{K-1}$ следующим образом: $\dot{c}_k = c_{\Pi(k)}$, $k = 0, 1, \dots, K-1$. Номера $\Pi(k)$ выбираются так.

1. Если $l_{IL} = 0$, то $\Pi(k) = k$ для каждого $k = 0, 1, \dots, K-1$.

2. Иначе $k = 0$, $K_{IL}^{\max} = 164$, для каждого $m = 0, 1, \dots, K_{IL}^{\max} - 1$ если $\Pi_{IL}^{\max}(m) \geq K_{IL}^{\max} - K$, то $\Pi(k) = \Pi_{IL}^{\max}(m) - (K_{IL}^{\max} - K)$, $k = k + 1$, где $\Pi_{IL}^{\max}(m)$ задается табл. 5.3.1.1-1 в стандарте 3GPP 38.212.

В табл. 5.3.2.1-1 стандарта 3GPP 38.212 определена последовательность $\mathbf{Q}_0^{N_{\max}-1} = \{Q_0^{N_{\max}}, Q_1^{N_{\max}}, \dots, Q_{N_{\max}-1}^{N_{\max}}\}$, где $0 \leq Q_i^{N_{\max}} \leq N_{\max} - 1$ обозначает индекс бита перед полярным кодированием для $i = 0, 1, \dots, N_{\max} - 1$ и $N_{\max} = 1024$. Последовательность $\mathbf{Q}_0^{N_{\max}-1}$ находится в порядке возрастания надежности $W(Q_0^{N_{\max}}) < W(Q_1^{N_{\max}}) < \dots < W(Q_{N_{\max}-1}^{N_{\max}})$, где $W(Q_i^{N_{\max}})$ обозначает надежность индекса бита $Q_i^{N_{\max}}$. Для блока длины N используется такая же последовательность $\mathbf{Q}_0^{N-1} = \{Q_0^N, Q_1^N, Q_2^N, \dots, Q_{N-1}^N\}$. Последовательность \mathbf{Q}_0^{N-1} – это подмножество полярной последовательности $\mathbf{Q}_0^{N_{\max}-1}$ со всеми ее значениями меньшими, чем N , расположенными в порядке возрастания надежности $W(Q_0^N) < W(Q_1^N) < W(Q_2^N) < \dots < W(Q_{N-1}^N)$. Таким образом, создается последовательность, которая будет использоваться для кодирования всех кодовых блоков длины N .

Обозначим как $\bar{\mathbf{Q}}_I^N$ множество индексов бит в полярной последовательности \mathbf{Q}_0^{N-1} и как $\bar{\mathbf{Q}}_F^N$ множество других индексов бит в полярной последовательности. Вычисление $\bar{\mathbf{Q}}_I^N$ и $\bar{\mathbf{Q}}_F^N$ будет рас-

смотрено далее. $|\bar{\mathbf{Q}}_I^N| = K + n_{PC}$, $|\bar{\mathbf{Q}}_F^N| = N - |\bar{\mathbf{Q}}_I^N|$, где n_{PC} – количество бит контроля четности. Для PDCCH $n_{PC} = 0$.

Обозначим как $\mathbf{G}_N = (\mathbf{G}_2)^{\otimes n}$ произведение Кронкера матрицы \mathbf{G}_2 самой на себя n раз, где $\mathbf{G}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$.

Для индекса бита j , $j = 0, 1, \dots, N-1$, обозначим как \mathbf{g}_j строку с номером j матрицы \mathbf{G}_N и как $w(\mathbf{g}_j)$ вес \mathbf{g}_j . Обозначим как \mathbf{Q}_{PC}^N множество индексов бит для контроля четности, где $|\mathbf{Q}_{PC}^N| = n_{PC}$. Ряд из $(n_{PC} - n_{PC}^{wm})$ бит контроля четности расположен в $(n_{PC} - n_{PC}^{wm})$ наименее надежных индексов бит. Ряд из n_{PC}^{wm} других бит контроля четности расположен в индексах бит строки с минимальным весом в $\tilde{\mathbf{Q}}_I^N$, где $\tilde{\mathbf{Q}}_I^N$ обозначает $(|\bar{\mathbf{Q}}_I^N| - n_{PC})$ наиболее надежных индексов бит $\bar{\mathbf{Q}}_I^N$.

Сгенерируем последовательность $\mathbf{u} = [u_0 \ u_1 \ u_2 \ \dots \ u_{N-1}]$ при $n_{PC} = 0$ следующим образом.

1. $k = 0$.
2. Для каждого $n = 0, 1, \dots, N-1$, если $n \in \bar{\mathbf{Q}}_I^N$ то $u_n = c'_k$, $k = k + 1$.
3. Иначе $u_n = 0$.

Выходная последовательность $\mathbf{d} = [d_0 \ d_1 \ d_2 \ \dots \ d_{N-1}]$ вычисляется как $\mathbf{d} = \mathbf{uG}_N$.

Выравнивание скорости для полярного кода определяется для каждого кодированного блока и состоит из перемежения подблоков, сбора бит и перемежения бит (в PDCCH последнее не используется).

Последовательность бит $d_0, d_1, d_2, d_3, \dots, d_{N-1}$ разбивается на 32 подблока. Выходные биты после перемежения подблоков, обозначенные как $y_0, y_1, y_2, \dots, y_{N-1}$, вычисляются для каждого $n = 0, 1, \dots, N-1$ следующим образом.

1. $i = \lfloor 32n / N \rfloor$.
2. $J(n) = P(i) \times (N / 32) + \text{mod}(n, N / 32)$.
3. $y_n = d_{J(n)}$.

$P(i)$ определяется табл. 5.4.1.1-1 в стандарте 3GPP 38.212.

Множества индексов бит $\bar{\mathbf{Q}}_I^N$ и $\bar{\mathbf{Q}}_F^N$ определяются следующим образом.

1. $\bar{\mathbf{Q}}_{F,tmp}^N = \emptyset$.
2. Если $E < N$ и $K/E \leq 7/16$, то $\bar{\mathbf{Q}}_{F,tmp}^N = \bar{\mathbf{Q}}_{F,tmp}^N \cup \{J(n)\}$ для каждого $n = 0, 1, \dots, N-E-1$.
3. Если $E < N$, $K/E \leq 7/16$ и $E \geq 3N/4$, то $\bar{\mathbf{Q}}_{F,tmp}^N = \bar{\mathbf{Q}}_{F,tmp}^N \cup \{0, 1, \dots, \lfloor 3N/4 - E/2 \rfloor - 1\}$.
4. Если $E < N$, $K/E \leq 7/16$ и $E < 3N/4$, то $\bar{\mathbf{Q}}_{F,tmp}^N = \bar{\mathbf{Q}}_{F,tmp}^N \cup \{0, 1, \dots, \lfloor 9N/16 - E/4 \rfloor - 1\}$.
5. Если $E < N$ и $K/E > 7/16$, то $\bar{\mathbf{Q}}_{F,tmp}^N = \bar{\mathbf{Q}}_{F,tmp}^N \cup \{J(n)\}$ для каждого $n = E, E+1, \dots, N-1$.
6. $\bar{\mathbf{Q}}_{I,tmp}^N = \mathbf{Q}_0^{N-1} \setminus \bar{\mathbf{Q}}_{F,tmp}^N$.
7. $\bar{\mathbf{Q}}_I^N$ состоит из $(K + n_{PC})$ наиболее надежных индексов бит в $\bar{\mathbf{Q}}_{I,tmp}^N$.
8. $\bar{\mathbf{Q}}_F^N = \mathbf{Q}_0^{N-1} \setminus \bar{\mathbf{Q}}_I^N$.

Далее рассмотрим процедуру сбора бит. Последовательность $y_0, y_1, y_2, \dots, y_{N-1}$ записывается в кольцевой буфер длины N . Выходная последовательность $e_k, k = 0, 1, 2, \dots, E-1$, генерируется следующим образом.

1. Если $E \geq N$, то $e_k = y_{\text{mod}(k, N)}$ для каждого $k = 0, 1, 2, \dots, E-1$.
2. Если $E < N$ и $K/E \leq 7/16$, то $e_k = y_{k+N-E}$ для каждого $k = 0, 1, 2, \dots, E-1$.
3. Если $E < N$ и $K/E > 7/16$, то $e_k = y_k$ для каждого $k = 0, 1, 2, \dots, E-1$.

Переमेжение бит не используется в PDCCH. Таким образом, после выравнивания скорости получаем выходную последовательность $f_0, f_1, f_2, \dots, f_{E-1}$, где $f_i = e_i$.

Перед модуляцией блок бит

$$b(0), b(1), \dots, b(M_{bit} - 1),$$

где M_{bit} – количество бит, переданных по физическому каналу, скремблируется следующим образом:

$$\tilde{b}(i) = (b(i) + c(i)) \text{mod} 2.$$

Генерация последовательности $c(i)$ описана в стандарте 3GPP 38.212 – 5.2.1. Генератор для скремблирования последовательности должен инициализироваться как

$$c_{init} = (\eta_{RNTI} + \eta_D) \text{mod} 2^{31},$$

где возможные значения η_{RNTI} и η_D определены в стандарте 3GPP 38.212 – 7.3.2.3.

Далее блок бит $\tilde{b}(0), \dots, \tilde{b}(M_{bit} - 1)$ модулируется с помощью QPSK (Quadrature Phase Shift Keying, квадратурная фазовая модуляция). В результате получается блок символов с комплексными значениями $d(0), \dots, d(M_{ymb} - 1)$.

Для имитационного моделирования нисходящего канала передачи служебной информации был использован MATLAB версии R2019b с инструментом 5G Toolbox, в котором представлены готовые решения для моделирования канала PDCCH в соответствии со стандартами 3GPP [3, 4].

Входные данные для моделирования:

- длина сообщения вместе с CRC $K = 54$ бита;
- длина кодового слова после выравнивания скорости $E = 124$ бита.

Для оценки результатов имитационного моделирования были использованы два критерия: вероятность ошибки на бит (BER, bit error rate) и вероятность ошибки на блок (BLER, block error rate). Значение BER рассчитывается по формуле

$$\text{BER} = \frac{N_{err}^{bit}}{N^{bit}},$$

где N_{err}^{bit} – количество бит с ошибками, N^{bit} – количество всех принятых бит. Значение BLER рассчитывается по формуле

$$\text{BLER} = \frac{N_{err}^{block}}{N^{block}},$$

где N_{err}^{block} – количество блоков с ошибками, N^{block} – количество всех принятых блоков.

В результате моделирования были получены зависимости BLER от SNR и BLE от SNR, представленные на рис. 2 и 3 соответственно.

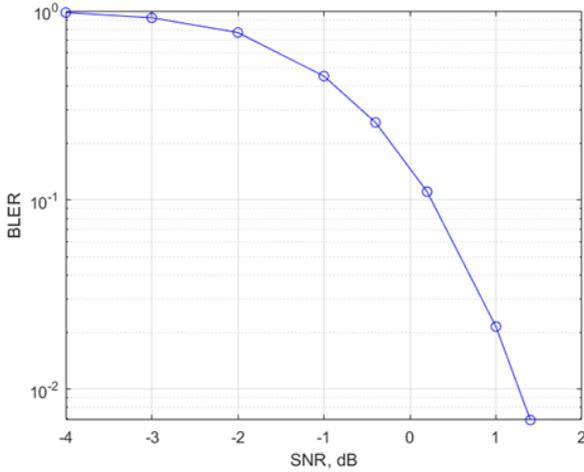


Рис. 2. Зависимость BLER от SNR для канального кодирования с полярным кодом

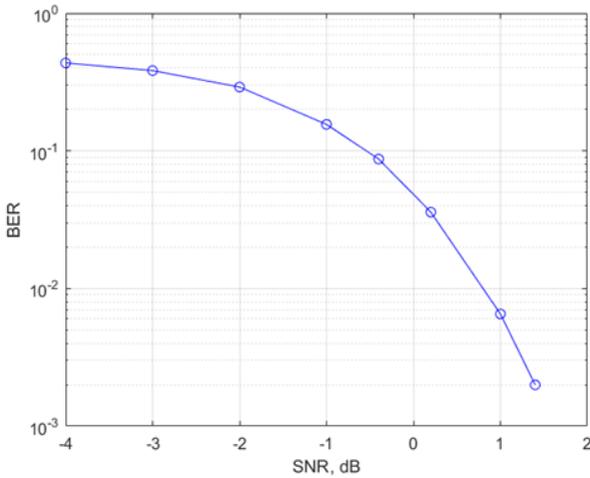


Рис. 3. Зависимость BER от SNR для канального кодирования с полярным кодом

Было подробно рассмотрено устройство нисходящего канала передачи служебной информации в сетях 5G, выделены характерные особенности каналов передачи служебной информации, такие как небольшие объемы данных, использование ограниченного количества схем модуляции и применение полярных кодов для канального кодирования. Также была разработана система имитационного моделирования нисходящего канала передачи служебной информации с учетом его особенностей в 5G.

Библиографический список

1. 3rd Generation Partnership Project (3GPP) "5G; NR; Physical channels and modulation," 3GPP TS 38.211 version 15.8.0 Release 15.
2. 3rd Generation Partnership Project (3GPP) "5G; NR; Multiplexing and channel coding," 3GPP TS 38.212 version 15.3.0 Release 15.
3. PDCCH Transport Process. URL: http://www.sharetechnote.com/html/5G/5G_PDCCH.bak (дата обращения: 21.05.2021).
4. Технологии связи. URL: <https://itechinfo.ru> (дата обращения 21.05.2021).

УДК 621.391.1

А. В. Царевский

студент кафедры инфокоммуникационных систем

И. А. Пастушок – кандидат технических наук – научный руководитель

СИСТЕМА ИДЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЕЙ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА НА ОСНОВЕ БЕСПРОВОДНЫХ ТЕХНОЛОГИЙ МАЛОГО РАДИУСА ДЕЙСТВИЯ

1. Обзор технологии радиочастотной идентификации

1.1. Системы радиочастотной идентификации

RFID (англ. Radio Frequency IDentification, радиочастотная идентификация) – способ автоматической идентификации объектов, в котором посредством радиосигналов считываются или записываются данные, хранящиеся в так называемых транспондерах, или RFID-метках.

Любая RFID-система состоит из считывающего устройства (считыватель, ридер или интерпретатор) и транспондера (он же RFID-метка, иногда также применяется термин RFID-тег) [1]

По дальности считывания RFID-системы можно подразделить на следующие группы:

- ближней идентификации (считывание производится на расстоянии до 20 см);
- идентификации средней дальности (от 20 см до 5 м);
- дальней идентификации (от 5 до 300 м).

Большинство RFID-меток состоит из двух частей.

Первая – интегральная схема (ИС) для хранения и обработки информации, модулирования и демодулирования радиочастотного (RF) сигнала и некоторых других функций.

Вторая – антенна для приема и передачи сигнала.

Классификация RFID меток.

Существует несколько способов систематизации RFID-меток и систем:

- по рабочей частоте;
- по источнику питания;
- по типу памяти;
- по исполнению;
- по источнику питания.

По типу источника питания RFID-метки делятся:

- на пассивные;
- активные;
- полупассивные.

По типу используемой памяти RFID-метки бывают:

- RO (англ. Read Only) – данные записываются только один раз, сразу при изготовлении. Такие метки пригодны только для идентификации. Никакую новую информацию в них записать нельзя, и их практически невозможно подделать;

- WORM (англ. Write Once Read Many) – кроме уникального идентификатора, содержат блок однократно записываемой памяти, которую в дальнейшем можно многократно читать;

- RW (англ. Read and Write) – такие метки содержат идентификатор и блок памяти для чтения/записи информации. Данные в них могут быть перезаписаны многократно.

По рабочей частоте:

- метки диапазона LF (англ. Low frequency, низкочастотный диапазон) (125–134 кГц);
- метки диапазона HF (англ. High Frequency, высокочастотный диапазон) (13,56 МГц) – используется в данной работе;
- метки диапазона UHF (англ. Ultra-High Frequency, сверхвысокочастотный диапазон) (860–960 МГц).
- радиочастотные UHF-метки ближнего поля.

1.2. Стандартизация RFID-технологий

В стандартах RFID можно найти ссылки на все стандарты, включая ГОСТы РФ. Стандарты ISO (Международная организация по стандартизации, ИСО (англ. International Organization for Standardization, ISO) [2] в области RFID относятся к интерфейсам и протоколам взаимодействия. Цель деятельности организации EPC Global – международной некоммерческой организации в области идентификации продукции – создание таких стандартов, которые позволят на практике идентифицировать любой объект в цепи поставок товаров компаний всего мира. Физический уровень обмена данными основан на ISO/IEC 18000-6. Разработано два совместимых поколения стандарта: Gen1 и Gen2. В табл. 1 приведены основные стандарты радиочастотной идентификации.

Таблица 1

Стандарты RFID

Стандарт	Основное содержание
ISO 11784	Радиочастотная идентификация животных. Структура информации
ISO 11785	Радиочастотная идентификация животных. Техническая концепция
ISO/IEC 14443	Карты идентификации. Бесконтактные карты с интегральной схемой. Proximity-карты
ISO/IEC 15693	Карты идентификации. Бесконтактные карты с интегральной схемой. Vicinity-карты
ISO/IEC 18001	Инф. технология. Технология AIDC. RFID для управления объектами. Требования к приложениям
ISO/IEC 18000-1	Интерфейс радиосвязи (часть 1). Общие параметры каналов связи для разрешенных частотных диапазонов
ISO/IEC 18000-2	Интерфейс радиосвязи (часть 2). Параметры интерфейса радиосвязи с частотой до 135 кГц
ISO/IEC 18000-3	Интерфейс радиосвязи (часть 3). Параметры интерфейса радиосвязи на частоте 13,56 МГц
ISO/IEC 18000-4	Интерфейс радиосвязи (часть 4). Параметры для интерфейса радиосвязи на частоте 2,45 ГГц
ISO/IEC 18000-5	Интерфейс радиосвязи (часть 5). Параметры для интерфейса радиосвязи на частоте 5,8 ГГц
ISO/IEC 18000-6	Интерфейс радиосвязи (часть 6). Параметры для интерфейса радиосвязи в диапазоне частот 860–930 МГц
ISO/IEC 18000-6	Интерфейс радиосвязи (часть 6). Параметры для интерфейса радиосвязи на частоте 433,92 МГц
ISO/IEC 15960	Синтаксис данных. Требования к прикладному сообщению
ISO/IEC 15961	RFID для управления объектами. Протокол передачи данных – прикладной интерфейс
ISO/IEC 15962	RFID для управления объектами. Протокол правил кодировки данных и логических функций памяти
ISO/IEC 15963	RFID для управления объектами. Уникальная идентификация радиочастотной метки

Существует пять функциональных групп (классов) меток.

Класс 0. Группа пассивных меток для идентификации объекта (Passive Identity Tag). Эти метки содержат только так называемый «электронный код продукта» (Electronic Product Code, EPC) в неизменяемом виде и использующий проверку при помощи циклического избыточного кода (CRC – Cyclic redundancy check) для обнаружения ошибок.

Класс 1. Группа пассивных меток с функциональными возможностями (Passive Functional Tag). Эта большая группа меток содержит все метки, имеющие какие-либо дополнительные функции, отличающие их от первой группы. Примером могут быть перезаписываемый EPC, шифрование данных и т. п.

Класс 2. Группа «полупассивных» меток (Semi-Passive Tag). К этой группе были отнесены все метки, использующие дополнительно источник питания. При этом основным источником питания должен являться считыватель, а точнее излучаемая им энергия.

Класс 3. Группа активных меток (Active Tag). Эти метки содержат встроенный источник питания, полностью обеспечивающий метку необходимой энергией вне зависимости от считывателя.

Класс 4. Группа активных RFID меток (RFID Tag). Эти метки не только содержат встроенный источник питания, но и набор определенной логики, позволяющей метке обмениваться данными с такой же меткой или обычным считывателем.

1.3. Основы передачи данных в RFID

RFID 14443 использует индуктивную связь [3]. Подобно трансформаторам, ближнее магнитное поле двух витков проводников используется для связи опрашивающего устройства (инициатора) и приемного устройства (адресата (рис. 1)).



Рис. 1. Схема взаимодействия опрашивающего (инициатора) и приемного(адресата) устройств

Рабочая частота составляет 13,56 МГц, а скорость передачи данных равна 106 кбит/с (иногда 212 кбит/с и 848 кбит/с).

Передача данных от приемного устройства

Из-за индуктивной связи витков опрашивающего и приемного устройств пассивное приемное устройство оказывает влияние на активное опрашивающее устройство. Изменения импеданса приемного устройства вызывают амплитудные или фазовые изменения в напряжении антенны опрашивающего устройства. Данный метод модуляции носит название нагрузочной модуляции. Он реализован в режиме приема (как в ISO/IEC 14443) с использованием вспомогательной несущей на частоте 848 кГц, которая модулируется полосовым сигналом и приводит к изменению импеданса приемного устройства. На рис. 2 показан спектр нагрузочной модуляции.

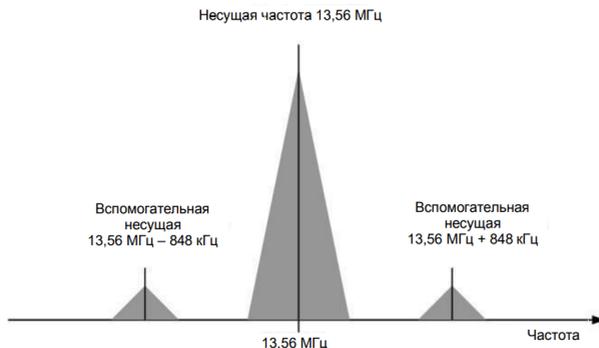


Рис. 2. Спектр нагрузочной модуляции

В качестве схемы модуляции используется амплитудная манипуляция (как в смарт-картах ISO/IEC 14443 A) или двоичная фазовая манипуляция (как в смарт-картах 14443 B). Существует и третий пассивный режим, совместимый с картами FeliCa, в котором метод нагрузочной модуляции реализован без использования вспомогательных несущих непосредственно на несущей частоте 13,56 МГц, подобно амплитудной манипуляции.

2. Описание стенда

В данной практической задаче используется плата для разработки под ESP8266 с поддержкой Bluetooth и Wi Fi, а также RFID/NFC модуль RC522 (рис. 3), предназначенный для считывания данных с меток и записи информации на них. Связка этих устройств позволяет реализовать считыватель NFC меток мобильных устройств, а также информацию с бесконтактных смарт-карт и брелоков.



Рис. 3 Комплект сборки

2.1. Комплект сборки

- МН-ET LIVE ESP32 DevKit-плата.
- RFID/NFC-модуль RC522.
- Набор соединительных проводов female/female.
- RFID-карта.
- RFID-брелок.

3. Сборка оборудования

Для считывания RFID-меток необходимо запрограммировать модуль RC522 под режим работы со стандартом ISO1443. Это делается для того, чтобы можно было считать UID-метки.

На первом этапе работы нужно соединить плату ESP8266 NodeMcu v3 с модулем RC522 с помощью проводов. Подключение осуществляется по схеме, изображенной на рис. 4.

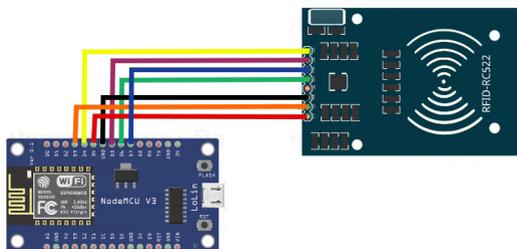


Рис. 4. Схема подключения RFID модуля RC522 к плате ESP8266 NodeMcu v3

После сборки необходимо подать питание на плату посредством подключения ее к компьютеру через USB порт (рис. 5).

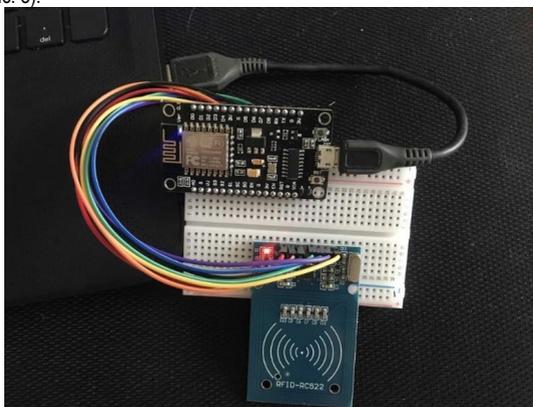


Рис. 5. Первая версия системы контроля доступа

Затем необходимо открыть Arduino IDE и загрузить на плату один из примеров скетчей. Это поможет убедиться в том, что устройство работает исправно.

4. EEPROM память

EEPROM (англ. Electrically Erasable Programmable Read-Only Memory) – электрически стираемое перепрограммируемое ПЗУ (Постоянное запоминающее устройство) [4], один из видов энергонезависимой памяти. Память такого типа может стираться и заполняться данными до миллиона раз.

Скорость работы с EEPROM:

- запись одного байта занимает ~ 3,3 мс;
- чтение одного байта занимает ~ 0,4 мкс.

У платы ESP8266 NodeMcu v3 объем энергонезависимой памяти равняется 4 096 байтам. По современным меркам объем памяти небольшой, но так как вес одной метки, записываемой в память, равняется 4 байтам, то этого количества хватает с запасом.

5. Алгоритм системы контроля доступа и примеры работы макета

На рис. 6 изображен алгоритм работы системы контроля доступа к ПК.

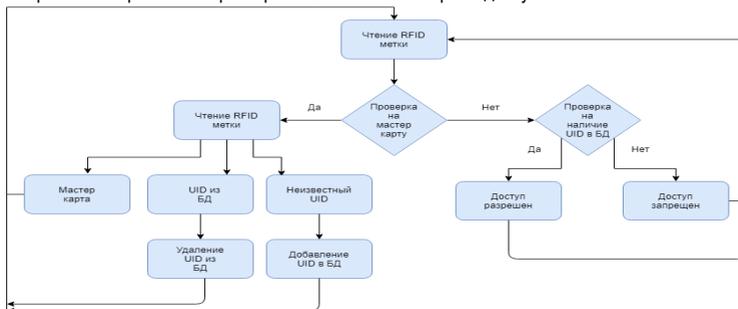


Рис. 6. Алгоритм системы контроля доступа

5.1. Примеры работы макета

На рис. 7 изображен случай, в котором отсканированная RFID-метка не внесена в память устройства, следовательно, на монитор порта выведется «You shall not pass», это говорит о том, что в доступе отказано. Также на монитор порта выводится информация о отсканированной метке, ее UID.

На рис. 8 изображен пример предоставления доступа после сканирования RFID-метки, об этом свидетельствует фраза «Welcome, You shall pass».

```
You shall not pass
Scanned PICC's UID:
83C88E4
```

Рис. 7. Отказ в доступе

```
Welcome, You shall pass
Scanned PICC's UID:
88B12BD
```

Рис. 8. Пример предоставления доступа

При сканировании Мастер Карты осуществляется переход в режим записи/удаления, данный режим работы программы позволяет также просмотреть количество пользователей, которые имеют доступ к данному ПК. На рис. 9 представлен пример добавления новой карты в базу данных системы контроля доступа.

```
Hello Master - Entered Program Mode
I have 3 record(s) on EEPROM
Scan a PICC to ADD or REMOVE to EEPROM
Scan Master Card again to Exit Program Mode
-----
Scanned PICC's UID:
19929E5
I do not know this PICC, adding...
Successfully added ID record to EEPROM
-----
Scanned PICC's UID:
CBD9C215
Master Card Scanned
Exiting Program Mode
-----
Scanned PICC's UID:
19929E5
Welcome, You shall pass
```

Рис. 9. Добавление новой карты в БД

Для того чтобы удалить пользователя из БД системы контроля доступа осуществляется переход в режим записи/удаления, далее сканируется карта, которую нужно удалить из БД, после успешного удаления на экране появится надпись «Successfully removed ID record from EEPROM» (рис. 10).

```
Hello Master - Entered Program Mode
I have 4 record(s) on EEPROM
Scan a PICC to ADD or REMOVE to EEPROM
Scan Master Card again to Exit Program Mode
-----
Scanned PICC's UID:
19929E5
I know this PICC, removing...
Successfully removed ID record from EEPROM
-----
Scan a PICC to ADD or REMOVE to EEPROM
Scanned PICC's UID:
CBD9C215
Master Card Scanned
Exiting Program Mode
-----
Scanned PICC's UID:
19929E5
You shall not pass
```

Рис. 10. Удаление карты из БД

Заключение

В статье продемонстрирована сборка стенда, сформированы учебно-методические указания для изучения технологии RFID и ее применения на практике, разработана схема устройства считывателя пассивных RFID-меток, написана программа для контроля доступа. Устройство было успешно собрано и протестировано на работоспособность.

Библиографический список

1. MIFARE ISO/IEC 14443 PICC Selection Rev. 4.1.2020. March, 20. P. 1–15.
2. Стандарты RFID. URL: <http://uhf-rfid.info/technology/standards> (дата обращения: 18.03.2021).
3. Molnar D., Wagner D. Privacy and Security in Library RFID: Issues, Practice and Architectures // Proceedings of the 11th ACM Conf. on Computer and Communications. Washington, 2004. P. 1–7.
4. Susy D. H. The cutting edge of RFID technology and applications for manufacturing and distribution. P. 122.

Информационно-сетевые технологии

УДК 681.325

М. П. Агеев, С. Ю. Гуков

аспиранты кафедры информационно-сетевых технологий

С. В. Мичурин – доктор технических наук, доцент – научный руководитель

АНАЛИЗ ЭФФЕКТИВНОСТИ НОВЫХ ГРАФИЧЕСКИХ ФОРМАТОВ КОДИРОВАНИЯ

До настоящего времени доминирующими графическими форматами являются JPEG и PNG, использующие алгоритмы сжатия с потерями. На протяжении многих лет они обеспечивают правильный баланс между размером файла, качеством изображения и совместимостью. JPEG был создан около 30 лет назад, примерно в то же время, когда появилась самая первая веб-страница. Однако, в то время как IT-технологии стремительно меняются, развиваются и выходят на новые рынки – смартфоны, умные часы, фитнес-браслеты и т. п. – графика в этих задачах все еще используется в формате JPEG. Очевидно, что IT-сфера нуждается в альтернативе JPEG, причем в такой, которая будет:

- так же широко поддерживаться,
- иметь лучшую эффективность сжатия,
- иметь более широкий набор функций и настроек.

В статье рассматриваются и сравниваются несколько новых форматов изображений, которые вполне могут претендовать на то, чтобы стать следующим JPEG: HEIC (HEIF), WebP и AVIF. В табл. 1 представлены общие характеристики форматов.

Таблица 1

Общие характеристики форматов

Характеристика	JPEG	HEIC (HEIF)	WebP	AVIF
Прозрачность	–	+	+	+
Сжатие с потерями	+	+	+	+
Сжатие без потерь	–	+	+	+
Анимация	–	+	+	+
Поддержка ОС	все настольные ОС, все мобильные ОС	ios, mac os, android 9+, windows 10 2018 update+	windows 7+, android 7+	windows10+
Поддержка браузеров	все	safari	chrome, opera	chrome
Процент сжатия по сравнению с JPEG	–	выше на 50–60 %	выше на 25–34 %	выше до 50 %
Возможность восстановить оригинал	–	+	–	+
Поддержка битности	8-бит	16-бит	8-бит	16-бит

Формат HEIC (HEIF)

Формат HEIC (High-Efficiency Image Container) был разработан в соответствии со стандартом ISO / IEC 23008-12 в 2015 г. группой экспертов по Moving Picture Experts Group (MPEG), которая известна благодаря разработке видео- и аудиоформатов таких как, например, MP3 или MP4. Здесь используются стандарты сжатия видео HEVC (High-Efficiency Video Container) или H.265 для кодирования изображений. Формат также известен под аббревиатурой HEIF (High-Efficiency Image Format) – «формат изображений высокой эффективности». С тех пор он был принят рядом производителей, включая Microsoft, Canon и Apple, которая в 2017 г. для устройств с iOS 11 сделала его форматом по умолчанию как замену JPEG, так как новый формат меньше по размеру, но сопоставим с ним по качеству.

Файл HEIC может содержать либо одно изображение, либо последовательность изображений, сохраненных в формате HEIF с метаданными, описывающими каждое изображение. Тем самым он может использоваться для хранения последовательностей изображений, таких как GIF-анимация или HDR-последовательность.

Еще одна особенность HEIF в том, что он поддерживает прозрачность, как это делает PNG.

У этого формата есть еще одно преимущество, когда дело доходит до редактирования: возможность отменить любые изменения даже после сохранения, так как в таких файлах также хранится информация о редактировании изображений, например о повороте и его обрезке.

Важно отметить, что в то время, как JPEG всегда производит 8-битные изображения с потерями (даже при самых высоких параметрах), HEIC может хранить 16-битные, с потерями и без потерь. Здесь имеется в виду, сколько различных цветов может хранить изображение. В то время как человеческий глаз не может различать между триллионами цветов, доступных в 16-битном формате, большее количество цветов создает большую гибкость при редактировании фотографии. 8-битная фотография может быть отредактирована до такой степени, что в цветах появляются полосы, в то время как в 16-битной такого эффекта не возникает. Это делает HEIC похожим по потенциалу на исходный RAW-файл, с которым JPEG даже отдаленно не может конкурировать. И самое главное, он может хранить все эти данные очень эффективно, что приводит к сравнительно небольшим размерам файлов [1].

HEIC превосходит формат JPEG по следующим параметрам:

- улучшенные блики, детали теней и средние тона,
- расширенный динамический диапазон,
- 16-битные цвета (в отличие от 8-битного цветового профиля JPEG),
- информация о глубине,
- прозрачность,
- хранение списка изменений,
- хранение последовательности изображений (анимация, HDR и панорамные последовательности, серийная съемка и т. д.).

Новый формат довольно универсален, так как может объединить возможности нескольких форматов изображений – JPEG, PNG и GIF – в одном файле, сохраняя при этом высокое качество изображения.

Формат WebP

WebP – формат изображений, созданный в 2010 г. и развиваемый компанией Google. Он обеспечивает сжатие изображений как с потерями, так и без потерь. Был создан как альтернатива PNG и JPEG и отличается от них гораздо меньшим размером при том же качестве изображения. Некоторые крупные компании, такие как Google, Facebook и eBay, все чаще советуют использовать данный формат для сжатия изображений.

Согласно данным Google:

- WebP сжимает изображения без потерь на 26 % лучше, чем PNG,
- WebP сжимает изображения с потерями лучше, чем JPEG на 25–34% при одинаковом индексе структурного сходства (SSIM),
- WebP поддерживает прозрачность без потерь (альфа-канал) при увеличении размера всего лишь на 22 %.

WebP использует новый алгоритм сжатия, поэтому искажение (т. е. деформация и ухудшение качества) выглядит иначе относительно других типов файлов. WebP оставляет четкие края фотографии, но при этом ухудшаются детализация и текстура, что неизбежно при сжатии с потерями. В то время как сопоставимый JPEG-файл показывает дрожание на сплошных участках изображения, WEBP может похвастаться гладкими переходами даже на самых низких настройках качества. Недостаток в том, что лица людей могут выглядеть «пластиковыми» при низких настройках качества [2].

Есть еще пара особенностей формата WebP. Настройки сжатия не повторяют в точности таковые в JPEG. Не стоит ожидать, что 50% качество JPEG будет соответствовать 50% качеству WebP. В случае в WebP качество падает довольно стремительно, так что начинать лучше с наибольших значений и постепенно их уменьшать. Однако плюс в пользу WebP – способность добавлять маску альфа-канала, так же как в PNG. В отличие от конкурирующего формата, изображение формата WebP можно сжать до одной десятой размера изображения в формате PNG.

Преимущества формата изображений WebP:

- меньший размер файла,
- улучшенный алгоритм сжатия,
- более качественные переходы цветов,
- маска альфа-канала.

Недостатки формата изображений WebP:

- неполная поддержка браузером,
- «пластиковый» вид искажений (особенно на объектах людей),
- неудобный интерфейс экспорта.

Формат AVIF

Формат изображения AVIF расшифровывается как AV1 Image Format, так как он основан на алгоритме AV1. AV1 – это самый последний видеокодек, разработанный Alliance for Open Media. Формат стал результатом сотрудничества между Google, Xiph и Cisco, которые решили объединить собственные видеокодеки в один открытый, бесплатный видеокодек [3].

AVIF использует тот же контейнер HEIF, что и формат HEIC в камере iOS. AVIF и HEIC предлагают аналогичную производительность сжатия. Разница в том, что HEIC основан на коммерческом, обремененном патентами формате H.265. В странах, разрешающих патенты на программное обеспечение, H.265 запрещено использовать без получения патентных лицензий. Он покрыт тысячами патентов, принадлежащих сотням компаний, которые разделились на две конкурирующие лицензионные организации. В настоящее время, когда видео в основном потребляется через бесплатные браузеры и приложения, старая модель лицензирования стала неустойчивой. Отсюда огромный стимул для таких веб-гигантов, как Google, Netflix и Amazon, чтобы поддержать альтернативу бесплатных лицензий.

AVIF, так же как и базовый формат HEIF, предлагает типичные функции, ожидаемые от формата изображения: поддержка любого кодека изображения, возможность использовать режим с потерями или без потерь для сжатия, поддержка различных субдискретизаций и битовых глубин и т. д. Формат позволяет хранить серию анимированных кадров (предлагая эффективную и долгожданную альтернативу анимированным gif-файлам), указать альфа-канал (который имеет огромное применение в UI), сохранять метаданные, такие как цветовая гамма и информация о высоком динамическом диапазоне (HDR) [4]. Использование алгоритмов AV1 оказывается полезным как относительно времени обработки, так и в решении аппаратных проблем.

Особенности, которые делают AVIF потенциально лучше конкурентов:

- поддержка высокого динамического диапазона (HDR), который обеспечивает лучшее и яркое изображение,
- поддержка сжатия без потерь и сжатия с потерями,
- наличие альфа-канала,
- обеспечение 8, 10 и 12-битной глубины цвета,
- бесплатная лицензия.

Однако при всех преимуществах есть несколько недостатков: скорость кодирования и отсутствие прогрессивного рендеринга.

Декодирование AVIF-изображений для отображения требует относительно большей мощности процессора, чем другие кодеки, но на практике оно остается достаточно быстрым. Даже бюджетные Android-устройства могут воспроизводить видео AV1 в формате FullHD без помощи аппаратного ускорения, а AVIF-изображения – это всего лишь один кадр видео AV1. Однако кодирование изображений AVIF происходит гораздо медленнее. На создание одного образа может уйти до нескольких секунд. AVIF поддерживает тайлинг, который ускоряет кодирование на многоядерных процессорах.

Второй существенный недостаток AVIF: в то время как прогрессивный JPEG может отображать более низкое качество изображения во время загрузки полного изображения, AVIF ничего не показывает на экране до тех пор, пока он не будет полностью загружен.

Имеется также проблема доступности на рынке. AVIF – довольно новый формат изображений и, хотя он получает много похвал во всем мире, по-прежнему не поддерживается большинством браузеров. Также, учитывая, что AVIF сохраняет изображение в формате .avif, он пока плохо поддерживается устройствами и операционными системами. Но благодаря заинтересованности крупных компаний, поддерживающих этот формат, его будущее выглядит довольно радужным и обнадеживающим.

Практическое сравнение форматов

Компания Netflix проводила сравнение форматов JPEG и AVIF [5]. Мы проанализируем форматы JPEG, AVIF, HEIC, WebP. Для этого необходимо произвести сжатие исходного кадра в разных форматах с разным коэффициентом сжатия. Размер исходного изображения составляет 711 Кб (рис. 1).



Рис. 1. Исходное изображение

В табл. 2 приведены размеры после сжатия для каждого из формата с разными коэффициентами сжатия.

Таблица 2

Сравнение размеров файлов после сжатия, Кб

Коэффициент сжатия (%)	HEIC	JPEG	WebP	AVIF
80	238	65	54	32
40	35	27	26	10
10	3	8	11	3

При коэффициенте сжатия 10 % абсолютно все форматы сильно искажают изображение. При коэффициенте сжатия 40 % меньше всего потерь у формата HEIC. На рис. 2 приведено исходное изображение после сжатия.



Рис. 2. Изображение в формате HEIC с коэффициентом сжатия 40 %

При высоком коэффициенте сжатия – 80 % лучшим форматом с точки зрения потерь стал также HEIC, но размер файла уменьшился только в 3 раза. У формата AVIF после сжатия при таком же коэффициенте размер файла уменьшился в 22 раза, при этом заметных потерь также не наблюдается. На рис. 3 представлено исходное изображение в формате AVIF.



Рис. 3. Изображение в формате AVIF с коэффициентом сжатия 80 %

Заключение

Были рассмотрены современные графические форматы HEIC (HEIF), WebP и AVIF, которые имеют наибольшую перспективу в ближайшее время заменить самый популярный в наше время формат JPEG. Произведено их сравнение по общим характеристикам, возможностям и качеству сжатия.

На данный момент активно развиваются три формата – HEIC, WebP и AVIF. Все они имеют схожие возможности и закономерности, современная тенденция заключается в объединении возможностей разных на данный момент форматов в один, например таких функций как альфа-канал, т. е.

прозрачность (как в PNG), анимация (как в GIF) и сжатие без потерь. Два из трех новых форматов позволяют сохранять прогресс редактирования изображения и, соответственно, в любой момент восстановить оригинал.

Самым перспективным форматом из рассмотренных считаем AVIF. При высоком коэффициенте сжатия он позволяет максимально эффективно сжать изображение без потери качества. AVIF является бесплатным в распространении, что может способствовать его скорейшей популяризации.

Библиографический список

1. McKeown S., Russell G. Forensic Considerations for the High Efficiency Image File Format (HEIF) // 2020 Intern. Conf. on Cyber Security and Protection of Digital Services (Cyber Security).
2. Si Z., Shen K. Research on the WebP Image Format // Advanced Graphic Communications, Packaging Technology and Materials. P. 271–277.
3. Barman N., Martin M. G. An Evaluation of the Next-Generation Image Coding Standard // 2020 Twelfth Intern. Conf. on Quality of Multimedia Experience (QoMEX).
4. Testolina M., Upenik E., Ebrahimi T. Comprehensive assessment of image compression algorithms // SPIE Optical Engineering + Applications. 2020.
5. AVIF for Next-Generation Image Coding / A. Mavlankar, J. De Cock, C. Concolato [et al.]. URL: <https://netflixtechblog.com/avif-for-next-generation-image-coding-b1d75675fe4> (дата обращения: 15.12.2020).

УДК 681.325

С. Ю. Гуков

аспирант кафедры информационно-сетевых технологий

А. М. Тюрликов – заведующий кафедрой, доктор технических наук, доцент – научный руководитель**ПРОБЛЕМА ФОРМИРОВАНИЯ КАРТЫ ГЛУБИНЫ С ИСПОЛЬЗОВАНИЕМ ЗАДЕРЖКИ СВЕТОВОГО СИГНАЛА В УСЛОВИЯХ ВЗАИМНЫХ ПОМЕХ ПРИ МУЛЬТИКАМЕРНОЙ СЪЕМКЕ**

Рассматриваются методы построения карт глубины с использованием временной задержки инфракрасного света технологии «Time of Flight» в 3D-сенсорах. Предлагается вариант решения проблемы, возникающей при мультикамерной съемке.

Трехмерная графика все более востребована в различных сферах науки и творчества. Но помимо создания виртуальных 3D-объектов и сцен, необходимо иметь объемное представление реально существующих предметов, реальных окружающих сцен.

Один из основных принципов всех современных 3D-технологий – метод пространственного разделения – разнесение изображений отдельно для каждого глаза. Для создания иллюзии трехмерности каждую сцену необходимо снимать одновременно двумя камерами с разных ракурсов, что не всегда удобно и возможно. Главное, что требуется для создания 3D-изображения, информация о глубине сцены. Для ее хранения обычно создается карта глубины.

Карта глубины (depth map) – это изображение, на котором для каждого пикселя вместо цвета хранится его расстояние до камеры (рис. 1). Идея, лежащая в основе построения карты глубины по стереопаре очень проста. Для каждой точки на одном изображении выполняется поиск парной ей точки на другом изображении. А по паре соответствующих точек можно выполнить триангуляцию и определить координаты их прообраза в трехмерном пространстве. При известных трехмерных координатах прообраза глубина вычисляется как расстояние до плоскости камеры.



Рис. 1. Типичная стереопара и ее карта глубины

Расстояние между глазами человека в среднем составляет 6,5 см. Мы прекрасно определяем объем и глубину, если посмотреть на объект на небольшом расстоянии, до полуметра, так как при попеременном закрывании правого и левого глаза картинка меняется. Но если посмотреть на этот же объект с расстояния в несколько метров, разницы почти не будет, т. е. картинка становится практически плоской. Именно поэтому, например, при съемке фильма «Хоббит» операторы разносили камеры на несколько метров, когда снимали актеров на дистанции до 10 м до объектива, чтобы получить большие углы и объем изображения.

Несколько лет назад крупные компании выпускали 3D-телевизоры и всячески их рекламировали. Однако по ряду факторов особого взрыва популярности так и не произошло, и на данный момент их выпуск остановился.

Разумеется, на волне всеобщего интереса к 3D, эту тему не могли обойти стороной и производители смартфонов. Некоторые из них несколько лет назад попытались использовать эту технологию и

в камерах своих смартфонов, позволяя снимать 3D-контент прямо на мобильный телефон. При этом модели LG Optimus 3D и HTC Evo 3D (рис. 2), как и несколько других, не взыскали популярности.



Рис. 2. Две задние камеры смартфона HTC Evo 3D

Для получения стереоизображений использовалась пара объективов на задней панели с расстоянием между камерами в два раза меньше, чем расстояние между глазами человека. Однако такой подход тоже далек от совершенства. Самый очевидный недостаток – снижение разрешения картинки на экране вдвое в режиме стерео, поскольку дисплей должен демонстрировать два изображения сразу. Второй по значимости – небольшая область комфортного просмотра: отклонение взгляда чуть в сторону приводит к исчезновению эффекта 3D. Третий – стереоэффект доступен только в альбомном режиме работы смартфона. Не стоит забывать и о субъективных недостатках: впечатления от 3D-экрана с параллаксной решеткой сильно зависят от восприятия конкретного человека.

Казалось бы, с тех пор про идею съемки 3D-сцен прямо со смартфона забыли. Однако такое предположение будет неверным. Последние годы все чаще можно наблюдать в смартфонах так называемые 3D-сенсоры или сенсоры глубины.

Если в первых попытках реализации большинство телефонов полагались на стереовидение, которое использовало две камеры для расчета глубины, которая была далеко не точной, так как этот метод не работает при слабом освещении или в темноте, то 3D-сенсоры основаны на другом подходе. Здесь камеры состоят из датчика, который использует крошечный лазер для «стрельбы» инфракрасным светом, производя тем самым 3D-зондирование окружающих объектов. Похожая идея с использованием инфракрасного излучения применяется, например, на фронтальных камерах смартфонов для распознавания лиц, используя структурированное световое освещение, когда точечный узор проецируется на лицо, а датчик измеряет расстояние между точками и смотрит на искажение в узоре, чтобы вычислить глубину (рис. 3) [1].



Рис. 3. Технология распознавания лиц

Большинство таких сенсоров называют ToF-сенсорами (Time of Flight, сенсорами времени полета) аналогично одноименной технологии [2]. Инфракрасный свет, испускаемый камерами, отражается от чего- или кого-либо перед камерой и возвращается обратно в ресивер. Однако, в отличие от технологии, используемой в распознавании лиц, здесь не используется узор из точек. Поскольку метод основан на инфракрасном свете, он хорошо работает в условиях низкой освещенности и в темноте.

Учитывая, что скорость света – величина неизменная, считывая разницу в скорости полета луча до объекта и обратно, можно определить расстояние и составить глубинную 3D-карту местности, улицы или комнаты. Масштаб и точность такой карты зависят от размера датчиков, количества лучей, их яркости и множества других факторов. Идеино это очень похоже на эхо в пещере, оно тоже после отражения от стенок возвращается с запаздыванием.

Хотя идея этой технологии и звучит довольно просто, возникает проблема – слишком маленькие промежутки времени. Есть два основных подхода для решения этой проблемы: dToF (direct ToF) и iToF (indirect ToF) [3, 4].

В сенсорах iToF излучатель отправляет высокочастотный модулированный свет, т. е. этот свет постоянно включается и выключается с частотой десятки миллионов раз в секунду. За счет того, что свету нужно время для полета до объекта и обратно, фаза – состояние где-то между включенностью и выключенностью света, вернувшегося в камеру, – немного отличается от фазы света в момент отправки. На сенсоре исходный и отраженный обратно от объекта сигналы накладываются друг на друга, и за счет этого определяется сдвиг фаз, который и позволяет понять расстояние до каждой точки объекта (рис. 4) [5, 6].

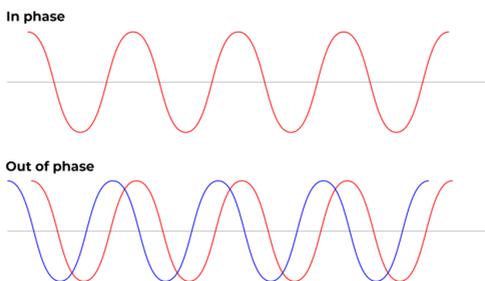


Рис. 4. Сенсор iToF

Сенсоры dToF работают немного иначе [6]. В них напрямую измеряется разница во времени между отправкой света и детектированием его отражения на сенсоре. Для этого используются так называемые SPAD (single photon avalanche diodes, однофотонный лавинный диод). Они могут детектировать крайне маленькие импульсы света, фактически даже ловить единичные фотоны. Такие SPAD расположены в каждом пикселе сенсора. А в качестве излучателя используются как правило так называемые VCSEL (Vertical-Cavity Surface-Emitting Laser, поверхностно-излучающий лазер с вертикальным резонатором) (рис. 5). Это лазерный излучатель, подобный тем, что используются в лазерных мышках. Сенсор LiDAR (Light Detection and Ranging, обнаружение и определение дальности с помощью света) разработан совместно с Sony и является первым массовым коммерческим dToF сенсором.

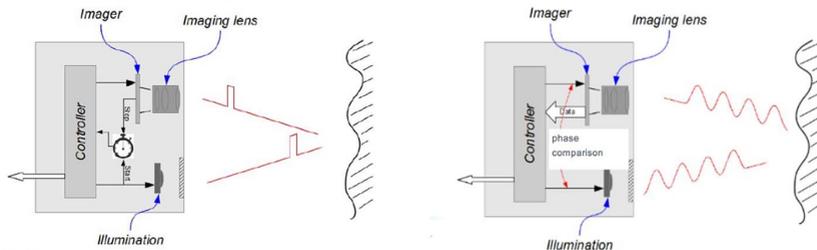


Рис. 5. Сенсор dToF

Android-смартфоны, имеющие такие сенсоры, используют iToF-сенсоры (например, LG G8 ThinQ и Honor View 20). Сенсоры dToF на данный момент применяются только в технике компании Apple – iPad Pro (2020) и iPhone 12 Pro [7].

Скорее всего, Apple оставили свой выбор на данном сенсоре по следующим причинам. Во-первых, в отличие от iToF-сенсора, излучатель в dToF испускает не сплошную стену света, а лишь светит в отдельных направлениях, что позволяет экономить батарейку. Во-вторых, dToF-сенсор меньше подвержен ошибкам в измерении глубины из-за так называемой multipath interference. Это типичная проблема iToF-сенсоров. Она возникает из-за переотражения света между объектами перед попаданием обратно в сенсор и искажает измерения сенсора.

В зависимости от размера и точности сенсора можно получить следующие данные: дальность и высоту, температуру, диффузию и плотность газа, степень облачности, скорость ветра, форму и размер ландшафта, высоту и плотность лесов, шероховатость морской поверхности и т. п. [8].

Таким образом, 3D-сенсоры уже применяются в таких сферах, как безопасность (распознавание лиц и т. п.), виртуальная реальность, качественно улучшение фотографий, образование и искусство и т. д.

Самые заметные улучшения пользователь может заметить в дополненной реальности. Например, приложение «Рулетка» становится более быстрым и точным, когда подключается LiDAR. Теперь его можно использовать как полноценный инструмент измерения, а не просто как пример использования AR.

Доступ к данным сенсоров у некоторых смартфонов открыт, поэтому появляется все больше приложений, предлагающих новые применения.

Согласно данным Apple, сканер LiDAR внутри iPad Pro (2020) может работать на уровне отдельных фотонов света, на расстоянии до пяти метров и на скоростях, которые обозначаются в наносекундах (т. е. кадр может быть захвачен моментально). Однако такие сенсоры пока довольно дорогие, поэтому в ближайшее время не стоит ждать LiDAR на бюджетных смартфонах.

За счет регистрации не только времени, которое требуется лазерному лучу для возврата, но и угла, под которым он отражается, данные LiDAR можно комбинировать с другой информацией для получения очень точных трехмерных карт. Профессиональные системы LiDAR часто объединяются с устройствами GPS. Стоит отметить, что датчики LiDAR стали популярны не прямо сейчас. Особо попу-

ляры они стали с появлением первых беспилотных автомобилей. Например, LiDAR установлен в беспилотных авто «Яндекса», датчик помогает ориентироваться машине в пространстве (рис. 6).

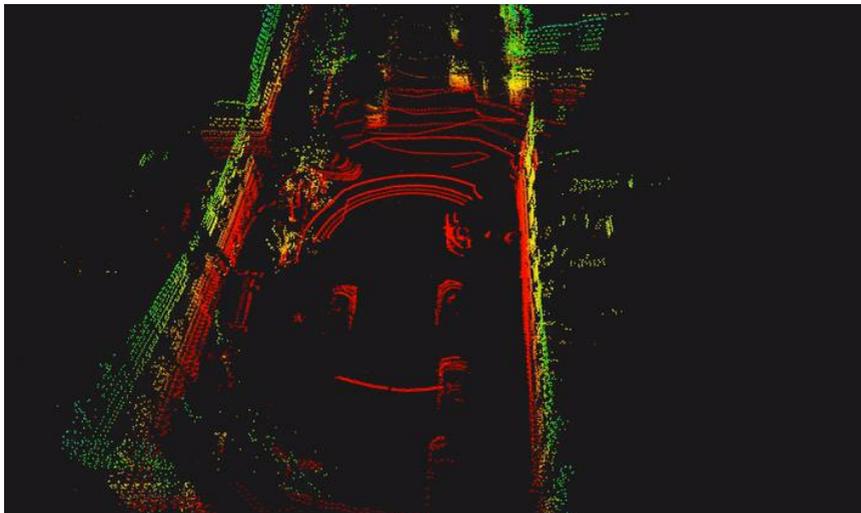


Рис. 6. Работа LiDAR датчика в беспилотных авто «Яндекса»

Учитывая, что 3D-сенсоры могут работать в любых условиях освещения, очень быстро и очень точно, есть все предпосылки к тому, что в следующее десятилетие они взлетят до ключевой технологии.

Проблема состоит в следующем. На данный момент всех устройств с 3D-сенсорами не так много, на рынке представлены единичные смартфоны, на дороге тестируются единичные автопилотируемые автомобили. На сигналы, отправляемые и получаемые излучателем, действует минимальное количество помех. Однако вполне очевидно, что при ситуации, когда по городу в час пик будет ехать огромное количество машин с сенсорами, на концертах огромное количество людей захотят одновременно снимать сцену (рис. 7), используя устройства с 3D-сенсорами, начнут возникать помехи. Посылающихся и отражающихся сигналов будет так много, что они начнут друг другу мешать, и устройства будут строить неверные карты глубин. К сожалению, работ, в которых предлагалось бы решение этой проблемы пока не опубликовано ни в русскоязычном сегменте, ни в иностранном.

Наиболее простым решением этой проблемы мы считаем следующий способ. Необходимо каким-либо способом (например, по радиоканалу) синхронизировать все находящиеся рядом в некотором радиусе устройства и тем самым создать своеобразную очередь, с учетом которой будет происходить последовательная съемка каждого из этих устройств, не мешая в этот момент другим. Учитывая достаточно высокую скорость съемки, измеряемую в миллисекундах и предположив, что в радиусе полутора метров теоретически может находиться не более пары десятков устройств, можно сделать предварительные выводы, что время, затраченное на съемку для каждого устройства, увеличится незначительно и незаметно для пользователя.

Однако для реализации решения необходимо, чтобы все разработчики устройств договорились о возможности синхронизации своих устройств между собой. Скорее всего, для этого потребуются создать единый стандарт, на разработку которого может уйти еще несколько лет, тогда как потребность в решении этой проблемы наблюдается уже сейчас.



Рис. 7. Стандартная ситуация на концертах с огромным количеством смартфонов

Актуально также найти решение описанной проблемы без необходимости разрабатывать единый стандарт для синхронизации находящихся рядом устройств.

Библиографический список

1. MixIn3D: 3D Mixed Reality with ToF-Camera / R. Koch, I. Schiller, B. Bartczak [et al.] // Dynamic 3D Imaging. Dyn3D 2009: Lecture Notes in Computer Science. Vol. 5742. Berlin, Heidelberg.
2. Fuchs S., Hirzinger G. Extrinsic and depth calibration of ToF-cameras // IEEE Conf. on Computer Vision and Pattern Recognition. Anchorage, AK, 2008. P. 1–6.
3. Generation of a high-resolution 3D-printed freeform collimator for VCSEL-based 3D-depth sensing / B. Chen, D. Claus, D. Russ, M. R. Nizami // Optics Letters. 2020. Vol. 45, Iss. 19. P. 5583–5586.
4. Van den Bergh M., Van Gool L. Combining RGB and ToF cameras for real-time 3D hand gesture interaction // 2011 IEEE Workshop on Applications of Computer Vision (WACV). Kona, HI, 2011. P. 66–72.
5. Rapp H. Experimental and Theoretical Investigation of Correlating TOF-Camera Systems // Interdisciplinary Center for Scientific Computing. 2007.
6. Что делают 3D-сенсоры в смартфонах? Разбор. URL: <https://habr.com/ru/company/droider/blog/519740/> (дата обращения: 14.12.2020).
7. iPhone 12 What Apple's 3D-sensing camera means for the future of photography. URL: <https://www.inverse.com/innovation/iphone-12-what-apples-3d-sensing-camera-means-for-the-future-of-photography> (дата обращения: 14.12.2020).
8. The newest phones use a time-of-flight camera, but what is it? We ask an expert. URL: <https://www.digitaltrends.com/mobile/what-is-a-time-of-flight-camera-we-asked-an-expert/> (дата обращения: 14.12.2020).

УДК 537.322.15

А. Д. Ефремов

студент кафедры информационных систем и технологий

Л. А. Осипов – профессор, доктор технических наук – научный руководитель

ОПТИМИЗАЦИЯ ТЕРМОЭЛЕКТРИЧЕСКОЙ СИСТЕМЫ ОХЛАЖДЕНИЯ ДЛЯ ПРИМЕНЕНИЯ В ПРОЦЕССОРНОМ КУЛЕРЕ

Введение

С развитием электронных технологий увеличение теплового потока электронных устройств с меньшими размерами и более высокой инкапсуляцией привело к тому, что основное внимание уделяется управлению температурой. Одновременно традиционные технологии пассивного охлаждения, такие как естественное конвективное охлаждение, принудительное воздушное охлаждение, жидкостное охлаждение, которые широко применяются, достигают пределов холодопроизводительности и эффективности охлаждения для электрических устройств высокой мощности. По сравнению с традиционным охлаждением, термоэлектрический обладает потенциалом для решения проблемы рассеивания тепла благодаря надежной работе, адаптивности и быстрым характеристикам реагирования. Для оптимизации производительности системы термоэлектрического охладителя было исследовано множество термоэлектрических модулей. Часто возникает противоречие между температурой поверхности и входной мощностью, когда речь заходит об оптимальной производительности системы термоэлектрического охладителя. Кроме того, для анализа и оценки термоэлектрической системы охлаждения была применена многоцелевая функция.

В данной работе представлена многоцелевая оптимизация на основе термоэлектрического теплообменного модуля для минимальной температуры поверхности и минимальной выходной мощности. Исследуются оптимальные условия эксплуатации для обеспечения надлежащего баланса между температурой поверхности и входной мощностью с точки зрения рабочего тока и теплового сопротивления горячей стороны. Эта оптимизация также способствует изучению термоэлектрической системы охлаждения в процессорном кулере.

Теоретический анализ

Схема термоэлектрической системы охлаждения состоит из термоэлектрического охладителя (ТЭО), ребристого радиатора, воздуховода, расширительного блока, источника тепла и изоляционной платы. ТЭО состоит из множества пар полупроводниковых параллелепипедов типа P-N, металлических разъемов и двух электрически изолирующих керамических пластин. Холодная сторона термоэлектрического охладителя прикреплена к поверхности источника тепла алюминиевым расширительным блоком, который может предотвратить снижение холодопроизводительности (Q_c) и коэффициента производительности (СР). Теплообменник воздуховода с реберным радиатором используется для отвода тепла от источника на горячей стороне. Кроме того, контакт термоэлектрического двухстороннего и алюминиевого расширительного блока покрыт термопастой для уменьшения теплового сопротивления. Осевой вентилятор, подключенный к началу и концу воздуховода, захватывает окружающий воздух. Поперечный воздух проходит через ребра радиатора для усиления тепловыделения в воздуховоде. Изоляционная плата используется для предотвращения потери тепла вокруг термоэлектрического охладителя, чтобы избежать снижения производительности ТЭО [1, 2].

Эффекты керамического покрытия с медными соединениями и электрических контактных сопротивлений малы, по сравнению с типом P-N, и не учитываются в уравнении теплового баланса. Таким образом, теоретическое уравнение для термоэлектрического модуля может выглядеть следующим образом:

$$Q_x = ST_x I - 1/2 I^2 R - K(T_2 - T_x), \quad (1)$$

$$Q_2 = ST_2 I + 1/2 I^2 R - K(T_2 - T_x), \quad (2)$$

$$P = Q_2 - Q_x = S(T_2 - T_x) + I^2 R. \quad (3)$$

Тогда СР термоэлектрического охладителя определяется как

$$CP = \frac{Q_x}{P}, \quad (4)$$

где I – электрический рабочий ток, T_r и T_x – температуры термоэлектрического охладителя (горячая и холодная стороны). Коэффициент Зеебека S , теплопроводность и общее электрическое сопротивление R являются физическими свойствами термоэлектрического модуля. В данной статье разработан модуль термоэлектрического теплообменника для анализа производительности системы термоэлектрического охладителя на основе одномерной теории теплопередачи на горячей стороне термоэлектрической системы. Как известно, метод эффективного числа единиц теплопередачи хорошо известен при решении задачи теплопередачи для горячей стороны ТЭО [3]. Уравнение теплопередачи с горячей стороны между теплоносителем и окружающей средой определяется как

$$Q_2 = \frac{T_2 - T_c}{\frac{1 - \sigma_2}{\varepsilon_2 C_2}}, \quad (5)$$

где σ_r – тепловые потери для воздуховода, т. е. приблизительно 10 %. ε_r – эффективность теплообменника, определяется как

$$\varepsilon_2 = 1 - \exp\left(-\frac{U_2 A_2}{C_2}\right). \quad (6)$$

Значение $U_2 A_2$ можно записать как

$$U_2 A_2 = \frac{1}{R_{об}}, \quad (7)$$

где тепловое сопротивление $R_{об}$ учитывает все эффекты конвективного теплового сопротивления потока и все проводящие эффекты теплового сопротивления конструкций теплообменника [4]. Производительность теплопередачи горячей стороны может быть представлена значением $U_2 A_2$, которое является фундаментальным параметром для теплообменника. Комбинируя уравнения (5)–(7), теплорассеивающая способность равна:

$$Q_2 = \varphi U_2 A_2 (T_2 - T_{cp}), \quad (8)$$

где φ – эквивалентная тепловая эффективность для теплообменника с горячей стороны, она может быть определена как

$$\varphi = \frac{\varepsilon_2}{\left(\ln \frac{1}{1 - \varepsilon_2}\right)(1 - \sigma_2)}, \quad (9)$$

где электрическая микросхема заменяется источником тепла, что соответствует действительности. Контактное сопротивление границы раздела между источником тепла и холодной стороной ТЭО можно определить как [5]

$$Q_x = \frac{T_{ucm} - T_x}{R_x}. \quad (10)$$

Наконец, всеобъемлющая формула может быть получена с помощью уравнений (1)–(10):

$$T_{ист} = \frac{\{K - SI + \frac{\varphi}{R_{об}} + [(K + SI) \frac{\varphi}{R_{об}} - S^2 I^2] R_x\} Q_x + 1/2 I^2 R (\frac{\varphi}{R_{об}} - SI + 2K) + K \frac{\varphi}{R_{об}} T_{cp}}{(K + SI) \frac{\varphi}{R_{об}} - S^2 I^2}, \quad (11)$$

$$P = \frac{SI(SI - \frac{\varphi}{R_{об}}) T_{ист} + \frac{1}{2} I^2 R [2K - SI + 2 \frac{\varphi}{R_{об}} + R_x (S^2 I^2 + SI \frac{\varphi}{R_{об}} + 2K \frac{\varphi}{R_{об}})] + SI \frac{\varphi}{R_{об}} T_{cp} [1 + SI R_x]}{K - SI + \frac{\varphi}{R_{об}} + [(K + SI) \frac{\varphi}{R_{об}} - S^2 I^2] R_x} \quad (12)$$

Многоцелевая оптимизация была определена как поиск новой переменной при одновременной оптимизации отдельных целей (т. е. минимизации $T_{ист}$ или P) с заданными ограничениями. В настоящей работе для многоцелевой оптимизации одновременно рассматриваются две задачи, минимизирующие $T_{ист}$ и P . Многоцелевая оптимизация представлена следующим образом:

$$J = f \frac{T_{ист}}{T_{ист, мин}} + (1-f) \frac{P}{P_{мин}}, \quad (13)$$

где f – весовой коэффициент для первой целевой функции $T_{ист}$ или P , $T_{ист, мин}$ и $P_{мин}$ – минимальное значение целевой функции $T_{ист}$ и P , когда эти функции рассматриваются независимо при заданных условиях управления соответственно.

Чтобы получить минимальное значение J , крайне необходимо дать ограниченный диапазон независимых переменных, основанный на практическом требовании. Из-за наименьшего тока, не проводящего к термоэлектрическому охлаждающему модулю, Rt не может быть относительно больше, поэтому для практических целей условия ограничения выбираются как $3 \text{ A} \leq I \leq 12 \text{ A}$, $0 \leq R_{об} \leq 0,8 \text{ } \Omega$ [6, 7].

Эксперименты и результаты

Для моделирования был использован термоэлектрический модуль типа TEC-12708, физические свойства термоэлектрического модуля получены практически, а именно: S составляет $0,0517 \text{ В/К}$, $K = 0,772 \text{ Вт/К}$ и $R = 1,411 \text{ Ом}$. В то же время условия для других параметров составляют $Q_x = 65 \text{ Вт}$, $R_x = 0,0347 \text{ Вт/К}$, $T_{cp} = 296 \text{ К}$. Настоящая многоцелевая функция была решена в различных рабочих условиях с известными параметрами.

Перед анализом многоцелевой функции была исследована оптимизация одной цели с учетом рабочего тока и теплового сопротивления горячей стороны, когда речь идет о целевой функции $T_{ист}$ и P . Результаты моделирования были показаны на рис. 1, 2, где указано, что оптимальные параметры могут быть получены на основе условий ограничения.

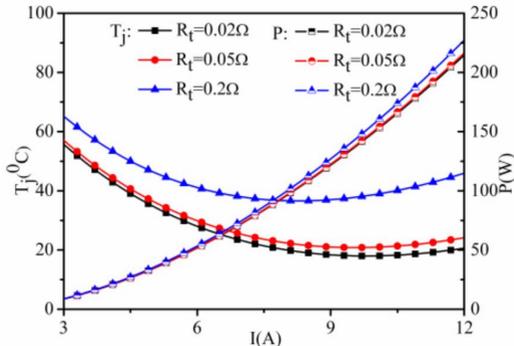


Рис. 1. Изменение тока в зависимости от температуры поверхности источника тепла и входной мощности при различном общем сопротивлении

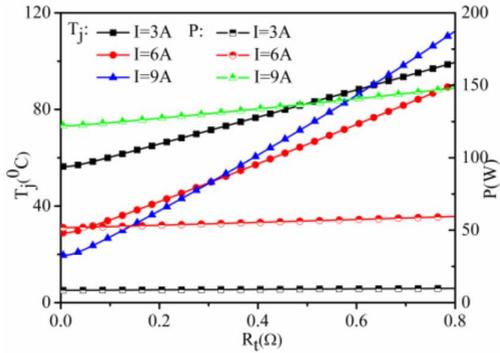


Рис. 2. Изменение теплового сопротивления горячей стороны в зависимости от температуры поверхности источника тепла и входной мощности при различном токе

Как показано на рис. 1, температура поверхности $T_{ист}$ уменьшается при запуске, а затем повышается с увеличением рабочего тока. Разница температур на горячей и холодной стороне термоэлектрического охладителя будет увеличиваться с течением времени, что приведет к увеличению нагрузки на охлаждение. Как следствие, температура поверхности будет снижаться. Температура поверхности также увеличивается при более высоком токе из-за увеличения Джоулевой теплоты по сравнению с теплотой Пельтье. Очевидно, что оптимальный ток для минимального $T_{ист}$ составляет 18 °C, $I = 9,8$ А при $R_{об} = 0,02$ Ом, 20,83 °C, $I = 9,6$ А при $R_{об} = 0,05$ Ом, 36,6 °C, $I = 8,3$ А при $R_{об} = 0,05$ Ом соответственно. Это ясно показывает, что для данного $R_{об}$ существует оптимальное I , приводящее к минимальному $T_{ист}$. Чем больше $R_{об}$, тем больше минимальный $T_{ист}$ и тем ниже соответствующий оптимальный I . Это связано с тем, что более низкий $R_{об}$ усиливает теплопередачу между жидкостью и горячей стороной ТЕС, что приводит к более низкому $T_{ист}$. Для различных $R_{об}$ оптимальный ток для минимизации P составляет при $I = 3$ А, $P = 8,58$ Вт. Это означает, что невозможно одновременно получить оптимальное значение $T_{ист}$ или P при одном и том же токе. Напротив, оптимальный $R_{об}$ для минимизации $T_{ист}$ и P находится при $I = 3$ А, когда $T_{ист}$ и P составляют 56,37 °C и 8,55 Вт при $I = 3$ А, 28,9 °C и 51,8 Вт, 19,38 °C и 139 Вт соответственно. Оптимальный $R_{об}$ может быть одновременно получен для минимизации $T_{ист}$ и P , это является проводящим для практического спроса. Однако с увеличением тока P также увеличивается. Поэтому необходимо контролировать I и $R_{об}$, чтобы получить оптимальные условия эксплуатации для практического приобретения.

По сравнению с одноцелевой оптимизацией, многоцелевая оптимизация реализуется с целью уменьшения $T_{ист}$ и P . На рис. 3 и 4 показано изменение I и $R_{об}$ на многоцелевой функции (J) при различном весовом коэффициенте (f). Следует отметить, что весовой коэффициент ($0 \leq f \leq 1$) представляет относительную важность для единственной целевой функции. На рис. 3 оптимальное значение J равно $1/3$ А, $1,43/3$ А, $1,87/3$ А, $2,2/4$ А, $1,02/9,6$ А. J увеличивается с увеличением I , когда $f > 0,75$. Если $f \rightarrow 1$, то нижний J существует при $I = 9,6$ А. Что касается тока, то минимальное значение J может изменяться в зависимости от f . Однако оптимальный ток будет при $I = 3$ А или $I = 9,6$ А, когда многоцелевая функция J будет ниже в глобальном масштабе. Также можно одновременно уменьшить $T_{ист}$ или P , чтобы сохранить J ниже при определенном f . Важно, что точка безубыточности находится на уровне $2,2/4$ А. Он относительно больше/ниже, когда I больше/меньше 4 А. На рис. 4 представлены результаты оптимизации, аналогичные результатам на рис. 2. Более низкий $R_{об}$ улучшает теплопередачу между жидкостью и горячей стороной или ТЕС, поэтому многоцелевая оптимизация согласуется с одноцелевой оптимизацией для $R_{об}$. На рис. 5 показан трехмерный график многоцелевой функции (J) с изменением I и $R_{об}$ для $f = 0,05$, что указывает на то, что нижняя точка находится при $R_{об} \rightarrow 0$ и $I = 3$ А. На рис. 6 и 7 показаны кривые $T_{ист} / T_{ист\ мин}$ и $P / P_{мин}$ на многоцелевой функции (J) при $f = 0,5$ с различными $R_{об}$. Цель представить кривые $T_{ист} / T_{ист\ мин}$ и $P / P_{мин}$ обеспечивает разумный баланс между оптимизаци-

ей и практическим спросом. Если практический спрос на $T_{ист}$ и P устанавливается в качестве определенного диапазона, то значение J будет определено, чтобы сделать условия эксплуатации более оптимальными.

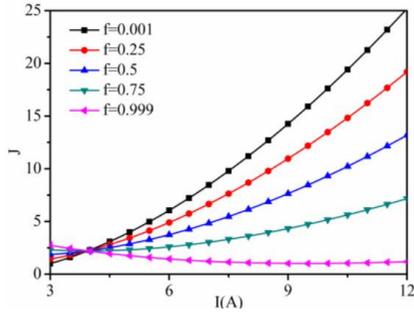


Рис. 3. Изменение тока на многоцелевой функции при различном весовом коэффициенте

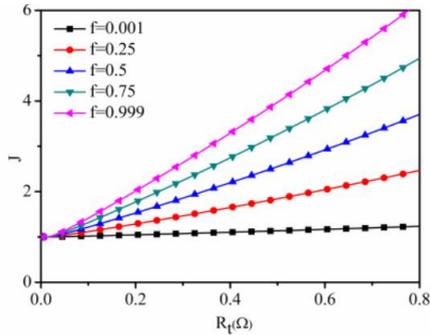


Рис. 4. Изменение теплового сопротивления горячей стороны на многоцелевой функции при различном весовом коэффициенте

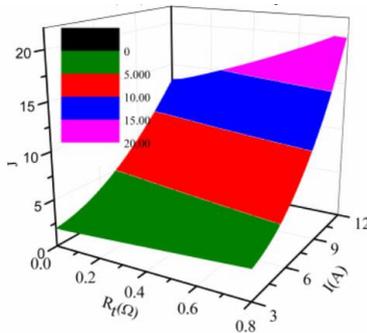


Рис. 5. Трехмерный график многоцелевой функции с изменением I и $R_{об}$ для $f = 0,05$

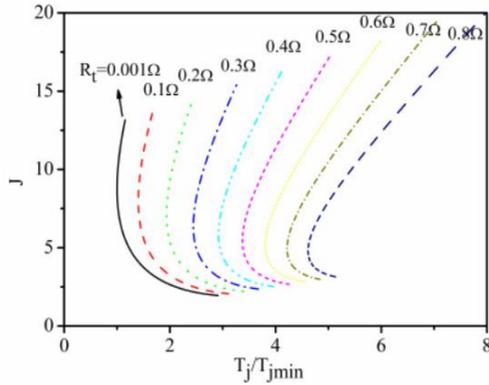


Рис. 6. Изменение $T_{ист} / T_{ист\ мин}$ на многоцелевой функции при $f = 0,5$ с различным общим сопротивлением

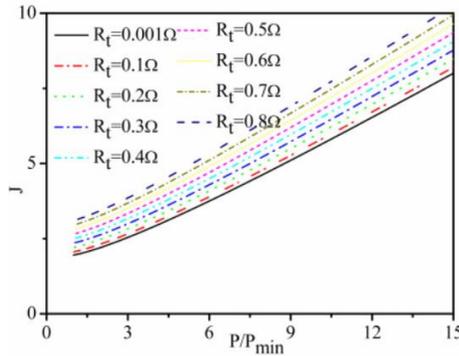


Рис. 7. Изменение $P / P_{мин}$ на многоцелевой функции при $f = 0,5$ с различным общим сопротивлением

Заключение

Представлена многоцелевая оптимизация на основе термоэлектрического теплообменного модуля для применения в процессорном кулере. Модуль термоэлектрического теплообменника, соединяющий несколько параметров, используется в качестве нового метода решения проблемы рассеивания тепла.

Во-первых, изучается одноцелевая оптимизация для обеспечения эталонных характеристик многоцелевой функции. оптимальный ток для минимального $T_{ист}$ составляет $18\text{ }^{\circ}\text{C}$, $I = 9,8\text{ A}$ при $R_{об} = 0,02\text{ }\Omega$, $20,83\text{ }^{\circ}\text{C}$, $I = 9,6\text{ A}$ при $R_{об} = 0,05\text{ }\Omega$, $36,6\text{ }^{\circ}\text{C}$, $I = 8,3\text{ A}$ при $R_{об} = 0,05\text{ }\Omega$ соответственно.

Кроме того, при комбинации температуры поверхности процессора и входной мощности с весовым коэффициентом в качестве многоцелевой функции снова реализуется оптимизация. Оптимальные переменные для одной целевой функции иногда не подходят для нескольких целевых функций. Необходимо учитывать относительную важность объективных ценностей, когда речь заходит о практическом спросе. Также оптимальные условия эксплуатации зависят от интенсивности этих рабочих параметров. Это исследование может иметь важное значение для оптимальной конструкции процессорного кулера. Анализ может быть расширен для теплового сопротивления холодной стороны, расхода воздуха или других параметров, влияющих на производительность ТЭО.

Библиографический список

1. Булат Л. П., Бузин Е. В. Термоэлектрические охлаждающие устройства. СПб.: СПбГУНИПТ, 2001.
2. Коленко Е. А. Термоэлектрические охлаждающие приборы, Л.: Наука. 1967.
3. Прохоров А. М. Физическая энциклопедия: в 5 т. М.: Сов. энцикл., 1988.
4. Drabkin I. Optimization of Thermoelectric Cooling and Generator Batteries. OmniScriptum Publishing KS, 2017
5. Hanreich G., Nicolics J., Musiejovsky L. High resolution thermal simulation of electronic components. Microelectronics Reliability, 2000.
6. Гнусин П. И. Исследование эффективности элемента Пельтье при различных режимах работы. СПб: Нева Технолджи, 2016.
7. Zhao D., Tan G. A review of thermoelectric cooling: materials, modeling and applications. Microelectronics Reliability, 2000.

УДК 004.622

Д. О. Шевяков

студент кафедры информационно-сетевых технологий

А. С. Афанасенко – кандидат технических наук, доцент – научный руководитель

ОБЛАСТЬ ПРИМЕНЕНИЯ И ПРИНЦИП РАБОТЫ МОДЕЛЕЙ WORD2VEC

Word2vec – общее название для совокупности моделей на основе нейронных сетей, которые предназначены для получения векторных представлений слов на естественном языке.

Область применения векторного представления слов – любые информационные системы, основным функционалом которых является работа с неструктурированными текстовыми запросами. Среди множества применений алгоритмов векторных представлений слов можно выделить несколько, где они просто необходимы. Самыми популярными приложениями, применяющими такое представление, выступают чат-боты. Чат-бот – это виртуальный помощник, имитирующий реального человека, который выстраивает диалог с пользователем, исходя из его запросов. Сложно представить, как подобная программа может работать, не используя векторное представление слов на естественном языке. Также одно из популярных направлений, использующих Word2vec, машинный перевод. Хотя машинный перевод был придуман раньше, чем Word2vec, применение моделей, реализующих векторное представление слов, позволило существенно улучшить качество перевода этих алгоритмов. Все современные приложения, которые используют обработку естественного языка, основываются на алгоритмах Word2vec. Они позволяют отобразить семантически сходные слова в близкие друг другу векторы в некоторой модели, в то время как далекие по смыслу слова будут выглядеть по-разному [1].

В Word2vec реализованы два основных алгоритма обучения – «мешок слов» и skip-gram. Порядок слов контекста не оказывает влияния на результат ни в одном из этих алгоритмов.

В модели мешка слов текст представляется в виде мультимножества его слов. Реализовано это так: все слова, участвующие в тексте, заносятся в массив. После этого фраза или слово, которое требуется представить в виде вектора, представляется в виде набора нулей и единиц, где составляющие ее слова записаны, как 1, а слова, не участвующие в ней, как 0. Данный алгоритм предназначен для определения слова из окружающего его контекста. Пример его алгоритма представлен в табл. 1.

Таблица 1

Пример работы алгоритма мешка слов

Слово/фраза	Словарь						
	мне	сегодня	себя	плохо	хорошо	чувствую	я
	Вектор						
Мне сегодня плохо	1	1	0	1	0	0	0
Я чувствую себя хорошо	0	0	1	0	1	1	1
Плохо	0	0	0	1	0	0	0

Skip-gram работает в обратную сторону. Этот алгоритм предназначен для предсказания окружающих слов, исходя из текущего. Принцип работы несколько сложнее, чем мешок слов. skip-gram построена на модели искусственной нейронной сети, которая состоит из трех слоев: входной слой, слой Embedding и выходной слой. Архитектура нейросети skip-gram представлена на рис. 1

Входной слой принимает одно слово в формате бинарного вектора, где единицей отмечена позиция слова в словаре. Может показаться, что для этого могут быть использованы векторы из алгоритма мешка слов, однако это не всегда так, поскольку skip-gram работает только с одним словом.

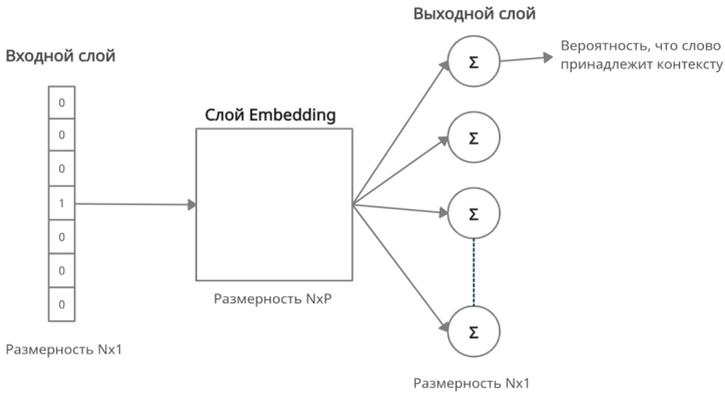


Рис. 1. Архитектура нейросети skip-gram

Слой Embedding представляет собой матрицу размером $N \times P$, где N – размер словаря, составленного из всех слов, участвующих в тексте, P – параметр, который подбирается эмпирически. Выходной слой – набор нейронов размером $N \times 1$ с функцией активации, каждый из которых выдает вероятность того, что входное слово принадлежит определенному контексту, т. е. другим словам [2].

Для извлечения нужной строки из слоя Embedding входной вектор умножается на матрицу (рис. 2). После этого нейроны выходного слоя, содержащие функцию активации, считают вероятности по следующей формуле:

$$softmax(z_i) = \frac{\exp(z_i)}{\sum_{k=0}^N \exp(z_k)}$$

$$[0 \ 0 \ 0 \ 1 \ 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} \rightarrow [10 \ 12 \ 19]$$

Рис. 2. Умножение входного вектора на матрицу Embedding

Благодаря знаменателю значения функции активации всех нейронов дадут в сумме 1 [3].

Библиографический список

1. Thushan Ganegedara. Intuitive Guide to Understanding Word2vec. URL: <https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-understanding-word2vec-e0128a460f0f> (дата обращения: 22.01.2021).
2. Роман Котюбеев. Как работает Word2vec. URL <https://python-school.ru/what-is-word2vec/> (дата обращения: 22.01.2021).
3. Tomas Mikolov. Distributed Representations of Words and Phrases and their Compositionality 2013. 3 с.

ИВАНГОРОДСКИЙ ФИЛИАЛ

Прикладная математика, информатика и информационные таможенные технологии

УДК 004.772

Д. О. Белоусов

студент кафедры прикладной математики, информатики и информационных таможенных технологий
Л. Н. Бариков – кандидат технических наук, доцент – научный руководитель

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРИМЕНЕНИЯ ДОМАШНИХ ОБЛАЧНЫХ ХРАНИЛИЩ В КОРПОРАТИВНЫХ ЦЕЛЯХ

Вопрос организации электронного документооборота на производстве с целью оптимизации рабочих процессов, улучшения контроля качества конечной продукции, наблюдение за технологическим процессом – вопрос, к решению которого в различных организациях подходят по-разному. В данной статье рассматривается решение, нацеленное на использование общедоступных технологий. В данном случае подразумеваются механизмы облачных хранилищ данных.

Применение сетевой инфраструктуры контроллера AD, в том числе распределенной файловой системы, сетевой файловой системы Samba, банального FTP-сервера или аналогов – самый распространенный подход к решению этой задачи. Несмотря на то что он имеет право на существование, применение его в период господства облачных технологий можно считать архаизмом.

Рассмотрим общий набор требований к современному подходу:

- классический механизм синхронизации локальных каталогов и файлов с облачным хранилищем;
- возможность работы с данными с разного класса устройств (персональные компьютеры, планшеты и смартфоны);
- совместное использование данных;
- возможность работы с данными и синхронизации с облачным хранилищем без подключения к ГВС.

Первые три пункта описывают любое бесплатное (или платное) облачное хранилище: Box; Dropbox; Google Drive; iCloud; MediaFire; Mega; OneDrive; pCloud; Облако Mail.ru; Яндекс.Диск и др.

Практически в каждом из приведенных случаев существуют корпоративные решения, которые могли бы решить множество вопросов, связанных с организацией распределенного или облачного хранения файлов. Но четвертый пункт, который в реалиях современной промышленности, когда объекты находятся достаточно удаленно от городской инфраструктуры, когда связь с ГВС – сетью Интернет – нестабильна или отсутствует, уничтожает преимущества этих систем [1, 2]. Поэтому далее будут рассматриваться только те решения, которые позволяют организовать свое «локальное» облачное хранилище.

Некоторые организации для решения подобных задач пользуются не облачными технологиями, а средствами контроля версий – SVN, Git и т. д. Этот подход тоже имеет право на существование и в некоторых случаях весьма практичен. Но возникает вопрос порога вхождения в использование подобных средств для рядового пользователя или организации внутреннего обучения по работе с подобной системой.

Итак, в ходе анализа открытых источников сети Интернет и различных публикаций, в том числе научных, был определен перечень наиболее распространенных «домашних» облачных хранилищ [3, 4]. Одним из основных параметров при выборе решений было наличие бесплатной, неограниченной по числу пользователей или объему дискового пространства версии. Чаще всего такие решения построены на основе открытого исходного кода, а организация серверной платформы и механизмов резервирования – и без того достаточно затратная статья бюджета предприятия. Платные версии, подписки

ИВАНГОРОДСКИЙ ФИЛИАЛ

и т. д. чаще базируются на более стабильных обновлениях платформы, более отзывчивой техподдержке и дополнительной функциональности, наличие которой может никак не пересекаться с требованиями организации.

Перечень облачных хранилищ (детальная информация приведена в табл. 1–3): ownCloud; Nextcloud; Sparkle Share; Seafile; Pydio; ProjectSend.

Таблица 1

Доступность облачных хранилищ

Облачное хранилище	Открытое ПО	Наличие бесплатной версии	Платная версия на 100 пользователей (\$/мес.)
ownCloud	+	+	1 150
Nextcloud	+	+	945
Sparkle Share	+	+	–
Seafile	+	+	400
Pydio	+	+	495
ProjectSend	+	+	–

Таблица 2

Операционные системы для серверных и клиентских решений

Облачное хранилище	ОС для сервера	ОС для клиента	Мобильный клиент
ownCloud	Linux	Mac OS, Windows, Linux	iOS, Android
Nextcloud	Linux	Mac OS, Windows, Linux	iOS, Android
Sparkle Share	–	Mac OS, Windows, Linux	*
Seafile	Linux	Mac OS, Windows, Linux	iOS, Android
Pydio	Mac OS, Windows, Linux	Mac OS, Windows, Linux	iOS, Android
ProjectSend	Mac OS, Windows, Linux	–	–

* – решения от сторонних разработчиков.

Таблица 3

Дополнительные возможности

Облачное хранилище	Работа через браузер	Корпоративные возможности (контакты, календарь)	Расширение функционала
ownCloud	+	+	+
Nextcloud	+	+	+
Sparkle Share	–	–	–
Seafile	+	частично	+
Pydio	+	частично	–
ProjectSend	+	–	–

Все приведенные облачные хранилища бесплатны. Работают в основном под управлением Linux [4, 5]. Снабжены клиентами для распространенных операционных систем, в том числе мобильных. Ко всем можно организовать доступ клиентов как из глобальной, так и из локальной сети, а последнее было одним из основных требований.

В качестве распространенного недостатка можно выделить попытки клиентов облачных хранилищ тесно интегрироваться с ОС и профилем пользователя – попытка интеграции со списком контак-

тов, документами, фотографиями, планировщиками. Этот недостаток требует более тонкой настройки серверной платформы и контроля за действиями пользователей, которые, даже неосознанно, могут превратить корпоративное облачное хранилище в свое персональное облако.

В целом отметим следующие моменты:

- построение корпоративного облачного хранилища без платных услуг от гигантов, таких как Google, Microsoft, Яндекс и т. д., вполне возможно;
- опираясь на приведенные в статье данные, можно выбрать наиболее оптимальное решение для каждого конкретного случая.

В дальнейшем планируется рассмотреть возможность построения своего файлового хранилища облачного типа с учетом возможностей распространенных открытых решений.

Библиографический список

1. Хрусталева В. Д., Гилькова М. С., Кумратова А. М. Преимущества и недостатки облачных хранилищ // Информационное общество: современное состояние и перспективы развития: сб. матер. X Междунар. студ. форума, Краснодар, 25–29 дек. 2017. Краснодар: КГАУ, 2018. С. 297–299.
2. Мурзак И. Облачные технологии для индивидуального и корпоративного пользователя // Техника и технологии, политика и экономика: проблемы и перспективы: сб. матер. III Междунар. науч.-практ. конф., Коломна, 20 апр. 2016. Коломна: КИ (филиал) МАМИ, 2016. С. 327–334.
3. Васяткин М. А., Белоусов К. В. Облачное хранилище данных // Studnet. 2020. Т. 3, № 10. С. 78.
4. Дудин П. О. Облачное хранение данных // Молодые ученые в решении актуальных проблем науки: сб. матер. Всерос. науч.-практ. конф. студентов, аспирантов и молодых ученых, Красноярск, 2020. Красноярск: ФГБОУ ВО СГУНиТ им. акад. М. Ф. Решетнева, 2020. С. 523–525.
5. Мишкин С. А. Облачный сервер для предприятия // Проблемы техники и технологий телекоммуникаций ПТИТТ-2020: сб. матер. XXII Междунар. науч.-техн. конф., Самара, 2020. Самара: ПГУ-ТиИ, 2020. С. 192–193.

УДК 528.514

А. П. Викторова

студент кафедры прикладной математики, информатики и информационных таможенных технологий

А. А. Сорокин – старший преподаватель – научный руководитель

ЗАВИСИМОСТЬ ПОГРЕШНОСТИ ИЗМЕРЕНИЯ РАССТОЯНИЯ ИК-ДАЛЬНОМЕРОМ ОТ СВОЙСТВ ПОВЕРХНОСТИ

Введение

Дальномеры – приборы, которые используются для измерения расстояния в различных областях человеческой деятельности. Они имеют существенные преимущества перед измерительными рулетками и линейками, так как позволяют производить замеры быстро и не сходя с одного места, при этом обеспечивают более высокую точность, чем традиционные методы измерения расстояния. Например, при измерении расстояния между объектами на местности, при использовании рулетки возможна значительная погрешность из-за неровностей на поверхности. Дальномеры измеряют расстояние по абсолютно ровной линии, что повышает точность во много раз.

Однако из-за высокой стоимости подобных точных приборов их практически не используют в быту, где достаточно точности обычной рулетки. В тех областях деятельности, где необходима высокая точность, например в строительстве зданий или когда измерение традиционными методами невозможно по техническим причинам, дальномеры крайне необходимы.

Качество дальномера напрямую зависит от его точности. Однако заявленная точность часто отличается от реальной. Ультразвуковые дальномеры сильно зависят от состояния воздуха, инфракрасные – от сторонних источников инфракрасных волн. В данной статье представлены результаты исследования зависимости погрешности измерения расстояния ИК-дальномером от свойств поверхности [1].

Принцип работы ИК-дальномера

Инфракрасный дальномер имеет простой принцип работы: излучатель испускает инфракрасную волну, которая доходит до непрозрачного тела, отражается от него и попадает в приемник [2]. Проходя через среду, излучение частично рассеивается и ослабевает, поэтому приемник должен быть очень чувствительным (рис. 1). Измеряя время между испусканием волны и попаданием ее в приемник, можно рассчитать расстояние между дальномером и объектом.

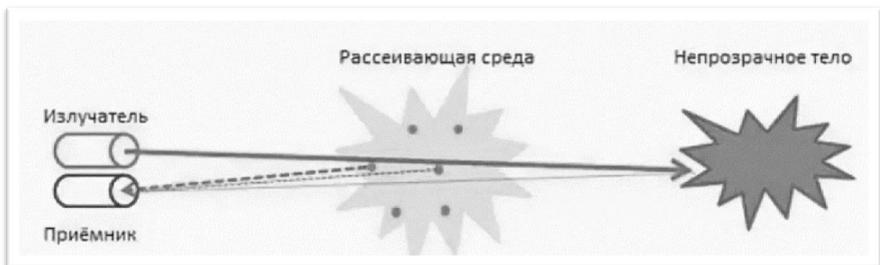


Рис. 1. Принцип работы

Ход эксперимента

Исследование проводилось с ИК-дальномером SHARP GP2Y0A021 (рис. 2), присоединенным к Amperka Strela [3]. Для эксперимента в качестве объекта была взята гладкая бумага четырех цветов: белая, красная, голубая и черная, и бархатная бумага трех цветов: красная, синяя и черная (с большей поглощающей способностью, чем у гладкой). Датчик располагался на расстоянии 30 см от объекта

(расстояние измерено по линейке с ценой деления 0,1 см). На основании проведенных опытов были построены табл. 1 и 2 и рис. 3.



Рис. 2. Внешний вид ИК-дальномера SHARP

Таблица 1

Результаты 100 измерений для разных материалов

Материал препятствия (бумага)	Показатель (см)									
	23	24	25	26	27	28	29	30	31	32
Белая гладкая	–	–	–	11	67	19	3	–	–	–
Красная гладкая	–	–	–	–	–	23	53	19	4	1
Синяя гладкая	–	–	–	1	78	18	1	2	–	–
Черная гладкая	–	–	–	6	93	1	–	–	–	–
Красная бархатная	–	4	2	1	–	43	41	7	2	–
Синяя бархатная	1	10	–	–	–	–	74	15	–	–
Черная бархатная	–	–	–	–	–	–	36	47	13	4

Таблица 2

Показатели измерений для разных материалов

Материал препятствия (бумага)	Среднее значение	Абсолютная погрешность	Относительная погрешность (%)
Белая гладкая	27,14	2,86	9,53
Красная гладкая	29,07	0,93	3,10
Синяя гладкая	27,25	2,75	9,17
Черная гладкая	26,95	3,05	10,17
Красная бархатная	28,37	1,63	4,43
Синяя бархатная	28,59	1,41	4,70
Черная бархатная	29,85	0,15	0,50

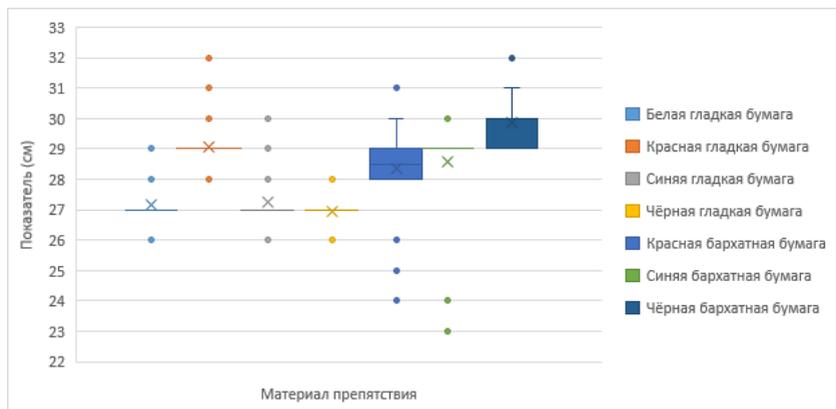


Рис. 3. Результаты измерений

Проанализировав полученные результаты, можно сказать, что бархатная бумага за счет лучшего поглощения инфракрасных волн дает более точные результаты, чем гладкая, а именно черная бархатная бумага (с максимальным коэффициентом поглощения из всех перечисленных материалов). Зависимость от цвета не была выявлена.

Вывод

По данным, полученным в ходе эксперимента, можно увидеть, что ИК-дальномеры не всегда бывают точными. На результат могут влиять солнечные лучи, нагревательные приборы, отражательные свойства поверхности, ошибки экспериментатора и многие другие факторы. Данный дальномер можно использовать там, где не требуется высокая точность, например для роботов-пылесосов.

Библиографический список

1. Дальномер. Виды и работа. Применение и как выбрать. Особенности // Тех.Приборы.Ру: сайт. URL: <https://tehpribory.ru/glavnaia/pribory/dalnomer.html> (дата обращения: 05.04.2021).
2. Какими инструментами пользуются для измерения расстояний? Названия и описания инструментов // Инструмент+: сайт. URL: <https://delta-instrument.ru/strojka/ustrojstvo-lazernogo-dalnomera.html> (дата обращения: 05.04.2021).
3. Инфракрасный дальномер Sharp (10–80 см) // Амперка: сайт. URL: <https://amperka.ru/product/infrared-range-meter-80> (дата обращения: 05.04.2021).

Е. Н. Долгов

студент кафедры прикладной математики, информатики и информационных таможенных технологий

А. А. Сорокин – старший преподаватель – научный руководитель

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ДЛЯ МОДЕЛИРОВАНИЯ

Введение

В процессах проектирования используются интеллектуальные, а не физические навыки, и их было гораздо сложнее автоматизировать до изобретения цифрового компьютера. Поскольку большая часть инженерного проектирования включает математическое моделирование, а компьютер очень эффективен при выполнении математических вычислений, очевидно, что теперь есть возможности для значительного повышения эффективности и автоматизации.

Прогресс в области технологий на протяжении всей истории оказывал влияние на множество отраслей. Эти изменения демонстрируются посредством анализа изменений в индустрии графического дизайна. Ранние программные инструменты были сконцентрированы на определенных аспектах проектирования, и, хотя они приводили к значительному повышению эффективности, это не рассматривалось как «автоматический» дизайн. Однако с появлением популяризированного искусственного интеллекта и благодаря анализу текущих продуктов для автоматизированного проектирования на потребительском рынке в области автоматизации дизайна это стало реальностью.

Процесс автоматического проектирования можно охарактеризовать четырьмя фазами: открытие, оценка, экстраполяция и интерполяция. Текущая фаза разработки программного обеспечения для инженерного проектирования имеет два аспекта, которые изменили это положение. Во-первых, это способность ранее разрозненных пакетов автоматически передавать информацию друг другу. Во-вторых, это развитие и растущее использование 3D-моделей.

Влияние компьютерного проектирования

В настоящее время проектирование и информационные технологии сильно дополняют друг друга. Вместе они обеспечивают мощное сочетание необычного мышления и навыков человека с машинными расчетами. Такой метод ускоренными темпами проникает в деятельность исследовательских и проектных организаций, поднимая проектную работу на принципиально новый уровень, при котором значительно ускоряются скорость и качество проектирования, сложные инженерные задачи решаются с большим обоснованием. Влияние компьютеризации настолько велико, что высокоуровневые навыки инженеров-проектировщиков и дизайнеров теперь в значительной степени излишни, но существует еще большая потребность в глубоком понимании поведения и «ощущении» подходящих решений.

Автоматизация проектирования позволяет быстро обрабатывать предложения, эффективно проектировать. Деятельность по созданию технических средств для автоматизации проектных работ осуществляется в рамках создания систем автоматического проектирования (САПР, англ. CAD, Computer-Aided Design) – программного пакета, предназначенного для создания технических объектов и оформления технологической и конструкторской документации [1].

Компьютеры выполняют вычисления одним нажатием кнопки, и большая часть проектирования вращается вокруг математического моделирования и автоматизированного черчения (САПР), что делает процессы проектирования идеальным кандидатом для автоматизации. Программное обеспечение сейчас настолько продвинуто, что фактически содержит большую часть подробных инженерных знаний (хотя, в сущности, не суждения). Это может привести к ложному предположению, что компьютер обладает инженерными знаниями. Оператору больше не нужно понимать инженерные процессы или вычисления, чтобы получить решение.

Сейчас для создания сложных проектов применяются компьютерные системы технологического моделирования (CAM/CAE-системы), сменившие программы расчета технологических процессов. CAM/CAE-системы представляют собой «программные конструкторы», позволяющие достаточно быст-

ро «собрать» практически любые технологические схемы, рассчитывать многовариантные технологические режимы, материальных и тепловых балансов, а также показателей качества сырья и продукции.

Относительная простота современного проектирования также способствует усложнению либо в ситуациях, когда достаточно более простого проектирования, либо когда вычислительные возможности позволяют проектировать более совершенную систему. В обоих случаях эта сложность может маскировать ошибку или даже способствовать ее возникновению. Молодым инженерам становится все труднее развить интуитивное «чувство» технических проектных решений и поведения, которые разработали их предшественники при выполнении ручных расчетов. Именно этот опыт направляет дизайнеров и предупреждает их о неподходящих решениях [2].

Поэтому строительная отрасль медленно продвигается к эффективному внедрению ИТ из-за ее уникальных особенностей, которые отличают ее от других отраслей. Во-первых, это разовые проекты, фрагментация отрасли, предварительные инвестиции и сопротивление изменениям, как указано в (FENG, 2006). ИТ могут оказать большую помощь во всех аспектах планирования, организации, эксплуатации и контроля проекта. Во-вторых, есть также данные, позволяющие предположить, что в некоторых случаях мощность программного обеспечения побуждает людей с недостаточной подготовкой или знаниями заниматься анализом и проектированием, а в других случаях чрезмерное доверие придается компьютерным результатам. Комбинация этих факторов может повысить вероятность ошибки и затруднить ее обнаружение. Этот риск необходимо осознавать и управлять им.

При этом если преодолеть эти особенности можно получить преимущества при применении на этапе планирования:

1) предсказание графика строительства на этапе проектирования и, следовательно, упрощение процесса проектирования, оптимизация дизайна;

2) полностью использовать данные, имеющиеся на чертежах, для управленческих целей на этапе строительства, таких как планирование и оценка, чтобы уменьшить утомительное манипулирование данными со стороны человека и потенциальный источник многочисленных ошибок.

Автоматизация процесса

Строительное проектирование можно разделить на различные функции, большинство из которых идеально подходит для компьютеризации. Техническое проектирование обычно начинается с концептуальной фазы, на которой анализируются и идеализируются реальные проблемы. Это навык высокого уровня, основанный на опыте, и, хотя некоторые программные инструменты могут помочь, например основанные на искусственном интеллекте, этот этап по мнению некоторых не очень подходит для компьютеризации.

Преимущества

Компьютеризация приносит существенные выгоды в виде экономии квалифицированной рабочей силы, более быстрого проектирования, уменьшения количества ошибок и возможности улучшить дизайн. Однако часто упускаемое преимущество компьютеров, предоставляющих «больше времени для размышлений», наивно, поскольку потребность в эффективности все-таки перевешивает.

Применение технологий моделирования весьма эффективно при поиске решений по реконструкции. Как правило, в ходе эксплуатации время от времени возникает необходимость реконструкции некоторых технологических установок, с предварительной проработкой альтернатив технических решений. С помощью расчета условий эксплуатации эта задача может решаться с минимумом ошибок. Это объясняется тем, что расчетный анализ моделей в итоге позволяет оптимизировать варианты, уточнить концепции, определить все приемлемые решения с минимумом затрат времени и ресурсов.

Для реального применения модели предварительно настраивают с использованием результатов исследований параметров внутреннего состояния и режимные параметры моделируемого объекта так, чтобы в процессе вычислительного эксперимента воспроизвести их с высокой степенью точности. Качество проектирования и расчета объектов зависит во многом от точности имеющейся информации о составе сырья. При этом модели разрабатываемых объектов необходимо настраивать на основании данных об эксплуатации аналогичных действующих объектов.

Вычисления большой сложности, на выполнение которых вручную потребовалось бы долгие часы, теперь могут быть выполнены за минуты. Поскольку программное обеспечение объединяет анализ, проектирование, составление чертежей и планирование, выходные данные, подходящие для производства, могут быть созданы одним оператором из единого набора исходных данных.

Современные средства очень многообразны и позволяют оптимизировать большую часть стадий инженерного труда и минимизировать затраты денежных средств и рабочего времени. При этом получается оптимальное решение поставленной задачи с учетом имеющихся данных и опыта, что приводит к повышению производительности продукта, а также к более экономичному решению.

В отличие от многих методов проектирования, мы обнаруживаем, что этап оценки обычно предшествует этапу экстраполяции. Кроме того, мы определяем общие вычислительные стратегии и проблемы, с которыми сталкиваются практики. Понимание этой информации может помочь дать рекомендации и дать указания дизайнерам с разными уровнями знаний, чтобы лучше интегрировать компьютеры и вычисления в существующую проектную работу, а также направить дальнейшие усилия на исследования автоматизации проектирования.

Проблемы, связанные с автоматизацией

Компьютерное проектирование фундаментально влияет на процесс проектирования и может увеличивать или уменьшать риск ошибки в зависимости от обстоятельств. Относительная простота современного проектирования также способствует усложнению в ситуациях, когда достаточно более простого проектирования [3].

Первая проблема информационных технологий и автоматизированного дизайна в ограничении сложности дизайна реальными техническими возможностями. Проблемы, связанные с компьютеризированным проектированием, возникают из-за возможности возникновения непредвиденных последствий, все более сложного дизайна. Поскольку отдельные пакеты связываются вместе и данные автоматически передаются между ними, возрастает риск дальнейшего распространения и усугубления ошибки, поскольку ее труднее обнаружить. Потенциал повышенного риска проиллюстрирован двумя примерами катастрофических обрушений, которые были приписаны компьютерной ошибке.

1. Гражданский центр в Хартфорде, Коннектикут, США, состоял из крыши площадью 2,2 акра, поддерживаемой только четырьмя колоннами. В 1978 г. он рухнул, когда был загружен снегом. Сложная крыша была спроектирована при значительной компьютерной помощи, что было практически невозможно сделать вручную.

Обрушение было объяснено двумя факторами: предполагалось, что эксцентричный шарнир не имеет эксцентриситета, а стойка считалась закрепленной, хотя на самом деле это не так. Стойка, вызвавшая отказ, фактически имела нагрузочную способность 9 % от расчетной, хотя приложенные нагрузки во время обрушения были в пределах проектных. Фактический рабочий прогиб вдвое превышал расчетный компьютерный анализ.

2. Sleipner – морская нефтяная платформа, построенная в Норвегии. Сообщается, что ошибка в создании сетки конечных элементов дала плохое представление о силах сдвига в критическом стыке, что привело к занижению оценки силы сдвига на 45 %. Соединение также было плохо детализировано, что усугубляло проблему. Сочетание этих двух факторов привело к его отказу и последующей потере всей конструкции.

В случае со стадионом Хартфорд можно утверждать, что доступность компьютера способствовала созданию более сложной конструкции. Построение платформы Слейпнера, вероятно, нельзя было значительно упростить, и это хороший пример преимуществ использования мощности компьютера для понимания более сложных структур посредством комплексного. Проблема в обоих случаях заключалась в том, что не были полностью оценены все последствия сложности дизайна и ограничений программного обеспечения. Обе структуры были представлены моделями, которые не были достаточно репрезентативными для реальной структуры, чтобы дать надежные результаты. Интересно, что, несмотря на размер этих ошибок, в обоих случаях обрушение было связано с комбинацией факторов, а не с одной ошибкой.

Данные свидетельствуют о том, что большинство ошибок, генерируемых компьютером, происходит из-за недостатков в процессе моделирования или непонимания ограничений программного обеспечения, а не из-за фактических вычислений или ошибок в самом программном обеспечении.

Вторая и главная проблема – человеческий фактор, который нельзя исключить полностью. Каждую ошибку в конечном итоге следует приписать людям, а не компьютеру, но сама проблема гораздо сложнее. Анализ известных ошибок показывает, что гораздо чаще они возникают в результате правильного вычисления несоответствующей модели, создается ложная уверенность в результате, так как он исходит от компьютера. Это создает парадоксальную ситуацию, когда относительно легко сгенерировать ответ на сложную проблему с помощью компьютера и сложно сделать это вручную, но сложность анализа или самой проблемы требует большего опыта для распознавания проблем с результатом.

Это устанавливает прямую связь между образованием и опытом оператора компьютера с вероятностью возникновения ошибки. Компьютеризация, которая включает информацию о конструкции в программу, также позволяет управлять программным обеспечением менее опытным сотрудникам. Считается что путем ручных расчетов инженеры могут развить интуитивное чувство правильного решения. В век компьютеров необходимость в ручных вычислениях почти исчезла, что затруднило инженеры развитие этого ощущения.

Фактически проблема не в чувстве решения, а в том, что компьютерный дизайн увеличивает важность понимания внутреннего поведения процессов, из-за их инкапсуляции новички часто концентрируются на результате процесса, отбрасывая их понимание.

Проблема усугубляется, когда компьютеры используются для проектирования сложных структур, которые фактически выходят за рамки возможностей ручных вычислений. Все становится еще сложнее, если делается вывод о том, что набор навыков, необходимый для традиционного ручного проектирования, отличается от набора навыков, необходимого для компьютеризованного проектирования. В традиционном подходе больше были необходимы разные математические навыки, которые дополнялись творческими, сейчас акцент смещается на моделирование и интерпретацию результатов на реальные условия.

Существующие системы

В настоящий момент существует много систем, разделяемых по мощности инструментария, различным возможностям и т. д. Представленные ниже являются достойными конкурентами, но предназначены для различных пользователей.

Система CATIA (Computer Aided Three-dimensional Interactive Application) – одна из наиболее распространенных высокоуровневых систем автоматизированного проектирования. Эта система проектирования (CAD), технологической подготовки производства (CAM) и инженерного анализа (CAE) включает подсистемы программной имитации технологических процессов, средства анализа и единую базу данных графической и текстовой информации.

3D Max – универсальный инструмент САПР для продвинутых и начинающих проектировщиков 2D/3D-моделирования. В программе предусмотрена возможность твердотельного моделирования, например проектирование двигателей, механических деталей, моделирование печатных плат и др.

IronCAD – это профессиональная САПР последнего поколения. Она представляет собой полнофункциональный инструмент разработчиков, в котором используются как классические методы параметрического моделирования, так и инновационный метод прямого редактирования. Несколько избыточна для обычных пользователей.

T-FLEX CAD – САПР-система с геометрической параметризацией, эффективное средство для профессиональной работы конструктора. Включает средства 2D-черчения, 3D-проектирования, модули динамического и конечно-элементного анализа.

TurboCAD – полностью настраивается, совместим с Microsoft Office и содержит встроенный Microsoft VBA. Приложение также содержит Software Development Kit и Visual Basic Macro Recorder.

КОМПАС – разработанная российской компанией «АСКОН» система автоматизированного проектирования с возможностями оформления конструкторской и проектной документации по стандартам ЕСКД и СПДС.

SolidWorks – продукт компании SolidWorks Corporation, система автоматизированного проектирования, инженерного анализа и подготовки производства изделий любой сложности и назначения. Пакет представляет собой инструментальную среду, предназначенную для автоматизации проектирования сложных изделий в машиностроении и других областях промышленности.

NaпoCAD – первая свободно распространяемая отечественная САПР-платформа для различных отраслей. То, что она распространяется свободно, в отличие от представленных выше, делает ее заслуживающей внимания альтернативной САПР.

Кроме систем, позволяющих создавать проекты, можно выделить системы, позволяющие воплотить созданные проекты – программы-слайсеры, например Ultimaker Cura, осуществляющая подготовку 3D-модели к печати. Она осуществляет оптимальное размещение модели на столе, заполнение согласно шаблону, автоматическое создание поддержек для сохранения формы. А плагины позволяют ей, например, расширить возможности заполнения и возможности настройки печати с целью создания наилучшего результата.

Расширение возможностей систем

Представленные системы обладают различными инструментами, но иногда они не могут удовлетворить специфические требования пользователей, например имеют особенности оформления, обладают недостаточной автоматизацией инструментов или не включают необычных для себя возможностей: работа с неподдерживаемыми форматами, другие методы расчета, отсутствие интеграции с непопулярными программами и т. д. Поэтому ко всем ним существуют плагины, не входящие в базовую версию, чтобы не перегружать программу, и загружающиеся при необходимости в нестандартных ситуациях. Если пользователей не устраивают возможности существующих плагинов, они могут создать собственные для специфических задач с помощью специальных инструментов. Например, Auto-Cad позволяет разрабатывать плагины на C#.

С целью предотвращения человеческого фактора примером плагина может послужить интеллектуальная экспертная система – комплекс, который, в отличие от обычных, не только аккумулирует знания специалистов в разных областях, но и в зависимости от действий пользователя выдает ему рекомендации и предупреждения. Такая ЭС должна одновременно интерпретировать и диагностировать данные, прогнозировать действия пользователей и данные с целью динамической поддержки принятия решений. Также она может использоваться для знакомства с САПР начинающих пользователей, главное, чтобы процесс объяснения был понятен пользователю, а база данных достаточно полной.

Разработка такой интеллектуальной ЭС достаточно сложный процесс (табл. 1) и не только потому, что она должна работать не с конкретными проблемами, а с разнонаправленным взаимосвязанным комплексом проблем.

Таблица 1

Этапы разработки экспертных систем

Этап	Функции
Демонстрационный прототип ЭС	Система решает часть задач, демонстрируя жизнеспособность подхода (БЗ содержит несколько десятков правил или понятий)
Исследовательский прототип ЭС	Система решает большинство задач, но неустойчива в работе и не полностью проверена (несколько сотен правил или понятий)
Действующий прототип ЭС	Система надежно решает все задачи на реальных примерах, но при сложной задаче требуется много времени и памяти
Промышленная система	Система обеспечивает высокое качество решений при минимизации требуемого времени и памяти; окончательная версия ЭС разрабатывается с использованием более эффективных средств представления знаний

Выводы

Сегодня перспективой развития САПР-систем является тесная интеграция со смежными программами. Суть этой интеграции заключается в установлении тесной взаимосвязи между расчетными и чертежными программами. Она позволяет за счет автоматизации процессов объединить в едином информационном пространстве все стадии моделирования и проектирования и, следовательно, избежать дорогостоящих и длительных циклов разработки типа «проектирование – изготовление – испытания».

В документе не утверждается, что компьютеризированному проектированию присущи недостатки, но есть различия между компьютеризированным и ручным проектированием, которые необходимо осознавать, понимать и управлять ими.

Как и во всех других разработках, есть преимущества и недостатки. Автоматизация значительно повысила эффективность и устранила многие утомительные аспекты проектирования, но какой ценой? Как правило, дизайнеры сопротивляются идее автоматического проектирования и предпочитают говорить об эффективности процессов и о компьютере, который «обрабатывает числа», оставляя дизайнеру «свободу мысли». Однако, несмотря на нежелание признать этот феномен, пришла автоматизация процесса проектирования. Причина, по которой дизайнерам не нравится понятие автоматического проектирования, заключается в том, что автоматизация подразумевает потерю контроля, и все дизайнеры должны нести (и несут) полную ответственность за все аспекты процесса, каким бы эффективным и автоматизированным он ни становился.

Когда реальная проблема разбита на дискретные элементы, которые можно смоделировать: анализ, проектирование, рисование, детализация и и др., то даже планирование можно компьютеризировать. Как только эти процессы будут переведены в электронную форму, будет достигнута частичная автоматизация. Когда программное обеспечение передает выходные данные от одной фазы к другой, наступает полная автоматизация.

Как прямое следствие этой автоматизации, многие этапы могут быть подготовлены одним оператором, то баланс навыков дизайнеров изменится, с уменьшением акцента на математические навыки и большей потребностью в концептуальных способностях и знании работы, и, что особенно важно, ограничениями пакетов программного обеспечения. Чрезвычайно важно полностью понять последствия этих изменений.

Вскоре роль графического дизайнера, скорее всего, будет вынуждена адаптироваться к меняющимся условиям. Однако, хотя роль профессионала в области графического дизайна может измениться или полностью исчезнуть, потребность в продуманном дизайне, вероятно, сохранится в обозримом будущем.

Библиографический список

1. Информационные технологии в моделировании и проектировании технических объектов // Тамбовский государственный технический университет: сайт. URL: <https://tstu.ru/book/elib3/mm/2017/maistrenko/t4.html> (дата обращения: 05.04.2021).
2. Гарднер П. Д. Влияние информационных технологий на профессию строительного инженера. Civil Comp пресс, 1995.
3. ISE. 2002. Руководство по безопасному использованию компьютеров при проектировании конструкций.

УДК 004+79

А. Р. Иванов

студент кафедры прикладной математики, информатики и информационных таможенных технологий

А. А. Сорокин – старший преподаватель – научный руководитель**МЕХАНИКИ ВЗАИМОДЕЙСТВИЯ ИГРОКОВ В КООПЕРАТИВНЫХ ИГРАХ**

Если попытаться разобрать видеоигры на составляющие, то можно выделить три основных части, из которых они формируются:

- визуал;
- звук;
- геймплей.

Визуалом является вся графика, применяемая в игре. Это то, с чего начинается знакомство игрока с игрой, через что игрок непосредственно взаимодействует на всем ее протяжении. По разным данным, человек получает от 75 до 85 % всей информации через органы зрения. Поэтому визуал – очень важная составляющая любой игры. Эмоции, получаемые от красивой картинке, обычно яркие и разнообразные. Основу нашего восприятия образуют и многие визуальные триггеры. Они являются врожденными и практически одинаково воздействуют на большинство игроков.

Звуки в игре – достаточно многофункциональный инструмент для создания разнообразного эмоционального отклика у игрока. Музыка создает или дополняет атмосферу игры. В боевых моментах играет напряженная, быстрая и ритмичная музыка. Когда же игрок бездействует, звучит легкий эмбиент (фоновая легкая музыка). Звуки подчеркивают любые действия и события, происходящие в игре. Многие разработчики не уделяют должного внимания этой составляющей, что большое упущение. Хоть игрок может и не обращать внимание на звуковой фон и сознание его игнорирует, подсознание все равно превращает это в постоянный поток чувств.

Геймплей – еще одна важная составляющая, которая отвечает за интерактивное взаимодействие игры и игрока посредством доступных игровых механик. Без геймплея нет игры. Механики реализуют в игре способы взаимодействия с игровыми объектами в рамках заранее созданных правил и ограничений, а также на индивидуальном поведении каждого игрока. Такое взаимодействие изменяет состояние игры, что впоследствии изменяет и сами решения игрока.

Игра является искусственной машиной по генерации опыта, опыт в свою очередь выступает конечным звеном в концептуальной модели игр.

Сначала создаются механики, которые опираются на сеттинг (среда, в которой происходит действие) игры. Посредством механик игроки порождают различные последовательности событий. Чем элегантнее разработана механика, тем в больших моментах игры игрок сможет ее использовать, генерируя тем самым больше разнообразных событий [1].

События влияют на игрока и вызывают эмоции. В отличие от фильмов или книг, события в играх не создаются напрямую. Геймдизайнеры создают механики, которые генерируют события. Этот уровень абстракции является базовым и отличает игры от всего другого.

Эмоции складываются в интегрированный опыт, который может надолго остаться с игроком. Наглядно эта модель показана на рис. 1.

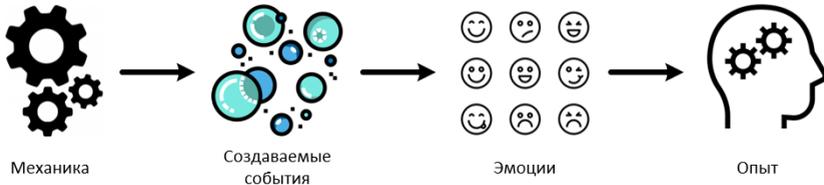


Рис. 1. Модель получения игрового опыта

Существуют два разных жанра – мультиплеер и кооператив. По сути, кооператив – частный случай мультиплеера. Все это многопользовательские игры, где два и более игроков сотрудничают друг с другом для выполнения какой-либо цели или решая задачи в процессе ее достижения, это является их общим сходством [2]. Отличия же проявляются даже в названии.

В мультиплеере (от англ. Multiplayer) присутствует большое количество игроков. Не так давно (28 марта 2020 г.) игра Planetside 2 побила собственный мировой рекорд по одновременному количеству игроков, играющих на одной карте. В этот раз в одной битве на карте «Эсамир» принимало участие 1 283 чел. по сравнению с предыдущей попыткой 2015 г. в 1 158 чел. [3]. Также к играм того же жанра можно отнести: Grand Theft Auto 5 (с ежедневным онлайн в 50 тыс. чел.), PlayerUnknown's BattleGrounds (с пиковым онлайн в 3,2 млн пользователей в 2018 г.), Fortnite Battle Royale (с пиковым онлайн уже 8,3 млн) [4].

В кооперативной игре (от англ. cooperative game) смысл заключается в совместном прохождении сюжетной игры со своими знакомыми или друзьями. Количество человек в кооперативе обычно находится в диапазоне от 2 до 4 и за редким исключением доходит до 8. В игре присутствует главная цель, которую игроки должны достигнуть, проходя игру совместно. Обычно она преподносится в общем сюжетном ключе. На пути всего прохождения встречаются множество задач, которые необходимо решить, чтобы пройти дальше. Все персонажи в таких играх, кроме самой команды, являются NPC (от англ. Non-Player Character). К играм в таком жанре можно отнести: Dead Space 3, Diablo 3, Far Cry 5, Lego: Star Wars, Portal 2.

Как уже было сказано, главное в играх – получение уникального опыта и эмоций, кооперативные игры могут сполна восполнить эту потребность. Вместе интереснее и проще добиться поставленной цели, а игроки могут «показать себя» перед другими. Проходить интересную сюжетную линию в компании друзей намного приятнее, чем исследовать игровые миры в одиночку. Кооператив позволяет генерировать уникальный опыт, потому как нельзя заранее предугадать поведение всей группы, вследствие чего с каждой командой могут происходить совершенно разные истории даже при прохождении одних и тех же миссий.

Геймдизайн кооперативных игр сильно отличается от однопользовательских. Специалистам требуется разработать определенные механики и учитывать психологию поведения групп игроков, которые позволят интересно исследовать и проходить игру.

Тьерри Лорет, дизайнер уровней из студии Ubisoft Quebec, выделяет четыре основных типа кооператива, каждый из которых представляет отдельную механику [5].

«Врата» – довольно классический дизайн, не требующий особых затрат. При этом подходе игрокам не нужно постоянно взаимодействовать между собой, каждый выполняет собственную локальную задачу, приводящую обоих к победе. Взаимодействие требуется лишь тогда, когда «врата» блокируют игроков и не дают пройти уровень дальше. Например, если даже один игрок способен пройти уровень в одиночку, но его прохождение дальше блокирует что-то, это вынуждает напарника прийти к нему на помощь и выполнить какие-либо действия совместно. Если такая механика присутствует в игре, то ее можно назвать «вратами». Такой режим в явном виде используется в игре Tom Clancy's Splinter Cell: Conviction.

«Комфорт» – наиболее популярный кооперативный режим. При нем игроки должны постоянно работать в команде. Успех в прохождении напрямую зависит от слаженности работы всех ее участников из-за особенностей хорошего баланса сложности и геймдизайна, позволяя игрокам сотрудничать, например из-за количества, сложности или типов врагов. В теории один «суперигрок» может выполнить всю работу. Рейды в ММО (массовая многопользовательская онлайн-игра) в большинстве случаев рассчитаны на хорошо подготовленную команду, и прохождение рейда с недобором игроков становится невыполнимым. Left for Dead хорошо демонстрирует описанный режим взаимодействия.

«Класс». В этом режиме каждый игрок имеет строго определенную роль. Без навыков какого-нибудь персонажа может быть трудно или невозможно пройти уровень или миссию. В таком режиме общение является частью игровой механики и обязательно для успешного прохождения, так как игрокам становится просто необходимо координировать свои действия. Например, в игре Operation: Tango. Игра о головоломках, где двум игрокам необходимо взаимодействовать друг с другом на вербальном уровне, чтобы решить загадки и достичь цели. Пока один игрок пытается проникнуть на охраняемый

объект, другой координирует его действия дистанционно. Роли задаются при старте игры и поменять их в процессе уже нельзя. *Keep Talking and Nobody Explodes* – игра с таким же приемом.

«Работа». В режиме рабочего кооператива специализация может быть, но она необязательна. В игре *Portal 2* роботы имеют одинаковые наборы навыков что никак не ограничивает игроков и позволяет самостоятельно решать, кто и что будет делать в зависимости от собственных предпочтений. Чтобы пройти уровень в *Portal 2*, необходимо использовать порталную пушку для прокладки пути до следующей точки и решения головоломки. Но уровень построен таким образом, что сделать это можно только совместно. Сложность подхода состоит в том, что необходимо проектировать уровень таким образом, чтобы интерес от прохождения распределялся равномерно между игроками, а прохождения уровня только кем-то одним было невозможным из-за особенностей игрового или особого дизайна уровней.

Помимо основных механик кооперативных игр, существует и другая проблема. Как заставить игроков объединиться и уйти от ситуаций, когда один игрок выполняет все цели один? Чтобы этого избежать, от слаженных действий каждого должна зависеть возможность всей команды пройти дальше. Например, как в ранее упомянутой *Tom Clancy's Splinter Cell: Conviction*, где были двери («врата»), которые можно было открыть только вместе.

Другой способ побудить игроков работать сообща – ввести систему наказаний. Например, популярная механика большинства кооперативных игр – воскрешение товарища, которого можно вернуть к жизни только определенное количество раз или в ограниченный интервал времени. Если не успеть это сделать, то игрок не сможет проходить игру вместе со всеми, а остальным станет ощутимо сложнее. Система наказаний склоняет к кооперативу двумя способами. Она побуждает следить за товарищами, чтобы вовремя успеть прийти на помощь, а с другой стороны, не дает опытным игрокам проходить задания в одиночку.

Еще одной механикой является наложение разных улучшений на характеристики других игроков. Такие улучшения могут действовать только определенное время, но при этом упростить прохождение остальным игрокам. В *Deer Rock Galactic* персонаж может раскрыть защитный купол, который отпугивает врагов и перезаряжает щиты напарников, а в *Remnant: From the Ashes* в особо сложные моменты игрок может увеличить урон всей группы. Одиночки же будут лишены таких возможностей.

Прием совместных действий вынуждает игроков распределять совместные задачи, чтобы пройти задание быстрее и эффективнее.

Все игроки разные и при разработке игры с совместным прохождением необходимо учитывать особенности каждого. Выделяют несколько психологических типов игроков, потребности которых следует учитывать:

- первопроходец – человек, который всегда стремится вперед и часто мешает другим игрокам;
- странникам нравится исследовать игру, и они часто уходят от основных миссий в поисках чего-то интересного или сбора внутри игровых предметов;
- помощники не любят сражаться и хотят помогать более умелым игрокам и никогда не лезут вперед;
- оставленные, ими становятся игроки, которые отстают от первопроходцев, или те, кто плохо разбирается в игре;
 - новички;
 - шутники получают удовольствие, подшучивая над другими игроками, что сильно мешает им наслаждаться игрой.

Первопроходцев можно остановить, например, выставлением ограничения на расстояние между игроками, что не позволит далеко им разбегаться. Странников нет необходимости как-то ограничивать, наоборот, можно сделать так что бы их действия приносили пользу команде, например поощрять за сбор необязательных предметов. Для помощников можно внедрить сторонние задания, которые не будут требовать особых навыков от игроков, но будут помогать выполнить основную задачу. Для шутников необходимо подобрать способы нейтрального подшучивания над другими игроками. Игроки, например, могут бить друг друга, но это не причинит им вреда.

Для создания хорошей кооперативной игры необходимо учесть и продумать огромное количество факторов. Знания различных существующих механик и психологических паттернов позволяет не только лучше разбираться в геймдизайне других игр, но и создает фундамент для создания собственных интересных приемов. Нет необходимости использовать какую-то определенную схему или придерживаться одного жанра, комбинируя их и дополняя, можно создать нечто новое, что позволит игрокам получить ранее не испытанный и уникальный опыт.

Библиографический список

1. Тайтан С. Геймдизайн. Рецепты успеха лучших компьютерных игр от Super Mario и Doom до Assassin`s Creed и дальше / пер. с англ. М. Панин, А. Попова. СПб.: Питер, 2021. 448с.
2. The difference between multiplayer and cooperative game // LORECENTRAL: сайт. URL: <https://www.lorecentral.org/2017/12/difference-multiplayer-cooperative-game.html> (дата обращения: 22.04.2021).
3. Morics P. PlanetSide 2 Breaks Its Own Guinness World Record With Recent Player Surge // Screen Rant: сайт. URL: <https://screenrant.com/planetside-2-guinness-world-record-player-count-high/> (дата обращения: 22.04.2021).
4. Bening С. Игры с самым большим онлайнom // Cubiq.ru: сайт. URL: <https://cubiq.ru/igry-s-samyim-bolshim-onlajnom/> (дата обращения: 23.04.2021).
5. Thierry L. The four atoms of cooperative video games // Gamasutra – The Art & Business of Making Games: сайт. URL: https://www.gamasutra.com/view/news/328756/The_four_atoms_of_cooperative_video_games.php (дата обращения: 23.04.2021).

УДК 004.4:33

Д. О. Михалев

студент кафедры прикладной математики, информатики и информационных таможенных технологий
Л. Н. Бариков – кандидат технических наук, доцент – научный руководитель

ВЕБ-СЕРВИСЫ ДЛЯ ПОЛУЧЕНИЯ ЕЖЕДНЕВНЫХ ЭКОНОМИЧЕСКИХ ДАННЫХ ПО ЭКОНОМИЧЕСКИМ ПОКАЗАТЕЛЯМ

Современная экономическая сфера и области государственного корпоративного сектора активно используют передачу информационных сообщений в различных форматах представления. Для целей синхронизации и в качестве эталона в экономическом (банковском секторе) принят в употребление базовый стандарт передачи данных основанный на XML.

Аналогично финансовым институтам, XML, формат для передачи данных, используют и бухгалтерские системы, экспортируют в XML ответы для последующей выгрузки в государственные информационные системы [1]. Например, 1С Бухгалтерия выгружает все стандартные виды отчетов, регламентированные законодательством, в формате XML. Пользователю достаточно лишь зайти в личный кабинет информационной системы оператора передачи данных и загрузить сформированный файл. Также в системах для электронного документооборота предусмотрены различные сервисы по автоматической конвертации данных из Excel-файлов в XML-формат. На данный момент отчетность в ПФР, ФНС и ФСС сдается через автоматизированные механизмы в формате XML. Ответы на отчеты и сообщения от принимающей стороны также приходят в формате XML.

Сам XML-формат имеет четкую древовидную структуру, каждый элемент обрамляется «тегами», информационными рамками – отрывающим и закрывающим тегом. Каждый тег в дополнение к своей информационной и логической идентификации может нести дополнительные атрибуты. XML имеет крепкие позиции и по праву является основой многих систем по декларативному описанию информации, которая представляется и обрабатывается в других интерфейсах.

HTML, представляемый браузером виде графического пользовательского интерфейса, основан на парадигме синтаксического описания, имеющей в своей основе XML-подобный формат описания данных.

Второе крупное ответвление в использование XML-образного описания GUI – XAML. Как основа для описания графической компоненты, XAML появился в технологической связке инструментов WPF/Silverlight.

Следующей реинкарнацией технологии WPF и соответственно XAML (XML) стало появление XAML в качестве основы для построения интерфейса в декларативном виде на платформе Windows 10/UWP.

XAML, по своей сути, это HTML для десктопного применения, обладающий полной синтаксической мощностью для кластеризации и использования пользовательской логики в части построения динамического контента.

Обилие готовых контейнерных элементов для различных шаблонов поведения GUI позволяет реализовать современный пользовательский интерфейс и иметь четкое логическое представление архитектуры интерфейсной части для реализации динамических элементов управления. В программной части достаточно знать имя, указанное в качестве атрибута в отрывающем теге. Дальнейшая работа строится на основе добавления или удаления элементов у выбранного узла (из дерева элементов). Операции перебора или движение по виртуальному дереву элементов интерфейса также возможны. В принципе, модель элементов XAML аналогична по функциональности модели DOM для HTML.

XML, как фундаментальный тип описания, по сей день активно используется для выгрузки информации информационными порталами на основе RSS каналов. Публикация происходит на основе базовых элементов. Цель RSS – предоставить структурированные данные по определенному шаблону на сторону, которая уже на своей площадке их обработает и выведет в графический интерфейс пользователя со своим кастомизированным набором стилей.

Выгрузка данных в RSS на основе XML – это путь передачи данных неограниченному кругу лиц и на незнакомые информационные ресурсы. Клиенты обращаются на канал RSS по адресу и соответственно получают ответ от сервера в формате XML.

В случае когда требуется выводить данные на собственном ресурсе, не используя дополнительные средства обработки, может быть применен подход, использующий преобразования XSL [2]. Данный формат позволяет определить схему шаблона и осуществить связывание с данными из дерева XML-документа. Связанный источник данных с XSL-шаблоном трансформируется в HTML, представление для отображения в стандартном браузере.

Центральный банк Российской Федерации, будучи финансовым регулятором и эмитентом национальной валюты на территории РФ, имеет собственные веб-сервисы для публикации необходимых экономических данных в формате SOAP (на основе XML) [1, 3]. Европейский банк аналогично имеет ряд сервисов для передачи широкому кругу интересующихся клиентов статистических и текущих данных в формате XML. Торговые площадки имеют подобные сервисы по формированию сводок, содержащих статистические данные и текущие котировки ценных бумаг, курсы национальных валют, различные индексы.

В текущий момент при рассмотрении тематики XML все чаще возникает вопрос о переходе с классического формата представления на набравший популярность JSON. В поддержку данного представления выступает стек технологий, использующий данную нотацию как основу обмена данными и как конфигурационное хранилище. Ярким примером такой трансформации можно назвать тип проекта ASP.NET Web Forms и его правопреемника ASP.NET Core. В данном сравнении первый имеет конфигурационный файл `web.config` именно в формате, основанном на XML, а в случае ASP.NET Core мы имеем конфигурационные файлы, описанные в JSON.

Инфраструктура JS-фреймворков также сильно опирается на JSON файлы. Помимо веб-составляющей, активно переходящей на использование JSON, настольные инструменты тоже тяготеют к использованию более легковесного собрата по представлению данных.

В разрезе вопросов разработки под UWP активно используется класс `DataContractJsonSerializer` из пространства имен `System.Runtime.Serialization.Json` в связке с методами из `Newtonsoft.Json` для сериализации и десериализации структуры данных, основанных на классических классах. Данная компоновка позволяет готовыми методами работать в памяти программы со списком, элементами которого является определенный класс по шаблону `List<T>`. К коллекции типа `List<T>` применимы все стандартные операции для работы со списками, а при необходимости сохранения данных при завершении работы программы данные сохраняются в `json`-файл.

JSON, как формат представления, имеет схожую структуру хранения данных в виде иерархического дерева и соответствующих узлов. Но имеет ряд отличий, главное из них – отсутствие возможности записи атрибутов в узлы дерева.

JSON и XML объединяет тип представления, данный способ хранения данных иногда называют документно-ориентированным подходом. В таком виде в одном документе хранится как структура данных, так и сами данные.

Использование документно-ориентированного подхода имеет как плюсы, так и минусы реализации. В случае отказа от классических СУБД с их готовыми механизмами по работе с данными требуется использование определенных библиотек или написание собственных классов и методов обработки данных. Возникает вопрос об индексации данных, ключевых атрибутов и операциях, обеспечивающих ограничение целостности информации. Возможны проблемы со связью различных файлов, имеющих отношение по ключевым атрибутам, при попытке устранения дублирования данных и соответственном создании справочных таблиц, записанных в виде `json`-иерархии.

От тенденции перехода к JSON-формату отходят все больше информационных порталов, и на данный момент они имеют выгрузку экономических данных как в XML, так и в JSON. Но корпоративный сектор все же имеет некую степень инерционности и не может в один момент времени перестроиться на новый формат обмена сообщениями в контексте использования всего объема сервисного программного обеспечения, что в итоге сохраняет позиции XML как основы для обмена информацией между интернет-ресурсами.

Поводя итог рассмотрению формата для выгрузки и методов обработки информации, важно проявлять гибкий подход к выбору стека технологий и остановиться на более корректном варианте работы на всех этапах обработки данных (с учетом возможностей технологии, текущей задачи и формата данных).

Библиографический список

1. Трэвис Б. XML и SOAP: программирование для серверов BizTalk. М.: Рус. редакция, 2001. 474 с.
2. Мангано С. XSLT. Сборник рецептов. М.: ДМК Пресс, 2008. 417 с.
3. Хабибуллин И. Самоучитель XML. М.: БХВ-Петербург, 2003. 336 с.

УДК 519.862.6:64

Э. О. Никашкин

студент кафедры прикладной математики, информатики и информационных таможенных технологий

Е. А. Яковлева – кандидат физико-математических наук, доцент – научный руководитель

ЭКОНОМЕТРИЧЕСКИЕ МЕТОДЫ, ИСПОЛЬЗУЕМЫЕ ДЛЯ ОЦЕНКИ ЭФФЕКТИВНОСТИ В СФЕРЕ ЖКХ

Уровень прибыли, которую получают фирмы в сфере ЖКХ, зависит от многих факторов. Наиболее важными выступают уровень качества услуг, своевременности выполнения работ, гибкость и оперативность предоставления услуг, своевременное техническое и профилактические мероприятия, комплексное предоставления услуг и выполнение работ [1–3]. Управление этими факторами повышает конкурентную способность предприятий, работающих в сфере ЖКХ, обеспечивает экономическую устойчивость предприятий [4].

Распределим факторы, влияющие на прибыль, по группам оказываемых услуг: техническое обслуживание, санитарное обслуживание, текущий ремонт. Для оценки влияния введем интегральный фактор, характеризующий качество услуг.

Математический аппарат, с помощью которого изучаются взаимосвязи между факторами, базируется на методах корреляционно-регрессионного анализа. В качестве результативного фактора выбираем эффективность (прибыль предприятия ЖКХ), которая является интегральным комплексным критерием.

Объясняющими факторами в разрабатываемой модели будут уровень качества технического обслуживания x_1 , уровень качества текущего ремонта x_2 , уровень качества санитарного обслуживания x_3 [5]. Отобранные параметры имеют наибольшее влияние на стабильность работы предприятия ЖКХ и на социальные показатели. Для того, чтобы установить степень влияния факторов, необходимо построить корреляционную матрицу между результативным и объясняющими факторами, между объясняющими факторами.

В табл. 1 представлены статистические данные для предприятий ЖКХ, которые обслуживают разные ТСЖ и работают с различными управляющими компаниями.

Таблица 1

Данные для предприятий ЖКХ

Предприятие	Уровень прибыли Y , %	Техническое обслуживание X_1	Текущий ремонт X_2	Санитарное обслуживание X_3
Φ_1	18,70	0,86	1,00	0,83
Φ_2	24,60	0,68	0,92	0,92
Φ_3	17,70	0,62	0,69	0,89
Φ_4	9,30	0,33	0,41	0,76
Φ_5	1,50	0,02	0,06	0,73
Φ_6	6,80	0,43	0,28	0,83
Φ_7	11,00	0,28	0,45	0,72
Φ_8	25,10	1,00	0,82	0,92
Φ_9	12,20	0,48	0,68	0,91
Φ_{10}	9,20	0,43	0,27	1,00
Φ_{11}	1,30	0,01	0,03	0,76

Значение каждого из объясняющих факторов определялось полуэмпирическими методами. Максимальное значение принималось равным единице для каждого фактора, для остальных предприятий значения показателей соответствуют степени предоставления. С помощью встроенного модуля «Анализ данных» Microsoft Excel были получены коэффициенты корреляции для линейной множественной регрессии (табл. 2).

Таблица 2

Коэффициенты корреляции

	у	X ₁	X ₂	X ₃
у	1	–	–	–
X ₁	0,909 030 419	1	–	–
X ₂	0,932 386 059	0,848 205 108	1	–
X ₃	0,557 779 369	0,558 899 984	0,454 576 17	1

Полученные результаты показывают, что уровень прибыли наиболее тесно связан с показателем Текущий ремонт x₂ (коэффициент корреляции 0,93) и показателем Техническое обслуживание x₁ (коэффициент корреляции 0,91), а показатель Санитарное обслуживание x₃ имеет среднюю силу связи и результатов.

Расчеты показывают, что в построенной регрессионной модели имеется коллинеарность объясняющих факторов. Коэффициент межфакторной корреляции между факторами Текущий ремонт и Техническое обслуживание равен 0,848, соответствует сильной степени связи между этим факторами, поэтому надежность результатов и качество модели будут низкими. Коллинеарность факторов подтверждает вычисление определителя корреляционной матрицы, который равен 0,19. Известно, что чем ближе величина определителя к нулю, тем более сильно выражена коллинеарность факторов.

Частные коэффициенты корреляции показывают тесноту связи между результирующим фактором при устранении влияния других факторов [6] (табл. 3).

Расчеты свидетельствуют, что наибольшее сокращение остаточной дисперсии происходит при включении в модель дополнительного фактора.

Таблица 3

Теснота связи

Ryx ₁ *x ₂	0,617 25
Ryx ₂ *x ₁	0,842 71
Rx ₁ x ₂ *y	0,004 23
Ryx ₁ *x ₃	0,867 83
Ryx ₃ *x ₁	0,143 89
Rx ₁ x ₃ *y	0,149 94
Ryx ₂ *x ₃	0,918 24
Ryx ₃ *x ₂	0,416 01
Rx ₂ x ₃ *y	-0,218 29

Из-за коллинеарности факторов модель множественной регрессии не подходит, поэтому стоит построить систему одновременных уравнений:

$$\begin{cases} y = f_1(X_1) \\ y = f_2(X_2) \end{cases}$$

после применения функции Регрессия получаем следующие уравнения:

$$\begin{cases} y = 1,9601 + 23,4968 X_1 \\ y = 0,8476 + 22,8300 X_2 \end{cases}$$

Оба уравнения проходят проверку на значимость по критерию Фишера, расчетные коэффициенты соответственно равны 42,82 и 59,88, при табличном значении коэффициента 5,12 при уровне значимости 0,05. Ошибка аппроксимации оказалась значительной: для первого уравнения 31 %, для второго 18,8 %. Для улучшения качества уравнений и снижения ошибки аппроксимации предлагается:

- 1) использовать нелинейную регрессию для каждого уравнений или для множественной регрессии;
- 2) дополнительно провести анализ данных и устранить выбросы;
- 3) увеличить объем данных.

Библиографический список

1. Астратова Г. В., Астратова С. В., Вихарева Я. Г. ЖКХ как объект государственного управления // Вестн. Евраз. науки. 2018. Т. 10., № 2. 14 с. URL: <https://esj.today/PDF/10ECVN218.pdf> (дата обращения: 19.03.2021).
2. Егоров А. Л. О реформе ЖКХ, ее успехах, неудачах и возможностях достижения конечного результата // Коммунальщик. 2015. № 9. С. 10–18.
3. Яковлева Е. А., Сорокин А. А., Коваленко Р. А. Моделирование потоков оказания услуг предприятиями сферы жилищно-коммунального хозяйства // Актуальные вопросы права, экономики и управления: сб. ст. XXVIII Междунар. науч.-практ. конф. Пенза: МЦНС «Наука и Просвещение», 2021. С. 59–62.
4. Айвазян С. А. Анализ динамики изменения стоимости жилищно-коммунальных услуг // Соврем. проблемы менеджмента и маркетинга рос. предприятий. 2016. № 11. С. 44–46.
5. Никитин С. И. Классические методы оптимизации социально-экономических процессов сферы сервиса: монография. СПб: Изд-во СПбГУСЭ, 2010. 151 с.
6. Эконометрика: учебник / И. И. Елисеева, С. В. Курышева, Т. В. Костеева [и др.]; под. ред. И. И. Елисеевой. 2-е изд. М.: Финансы и статистика, 2008. 576 с.

А. В. Овчаренко

студент кафедры прикладной математики, информатики и информационных технологий

Л. Н. Бариков – кандидат технических наук, доцент – научный руководитель

ДИНАМИЧЕСКОЕ РАНЖИРОВАНИЕ ЗАДАЧ ДЛЯ СКЛАДСКИХ РАБОЧИХ

Динамическое ранжирование – одно из представлений классической задачи многокритериальной оптимизации. Есть ряд факторов (критериев), влияние которых на условный вес элементов оптимизируемого процесса подчиняется какому-то закону. Задача оптимизации сводится к формулировке этого закона. Отличие динамического ранжирования от статического сводится к выводу нескольких закономерностей для разных случаев или же добавлению в формулировку закона дополнительного фактора. Цель данной статьи – выявление подхода к формулированию подобного закона, что позволит равномерно распределять нагрузку среди рабочих, снизит вероятность уклонения рабочими от своих обязанностей и, соответственно, позволит повысить производительность.

Тема данной статьи – динамическое ранжирование задач для складских рабочих. На самом деле тему можно было бы сформулировать и более емко, исключив складских рабочих. Но в данном случае конкретизация влияет на определение набора факторов.

Источником (набором факторов) для определения ранга или веса задачи будет как сама задача, так и характеристики рабочих, которые ее выполняют. Задачи должны выдаваться с каким-то приоритетом, например задача перемещения товаров с целью оптимизации их размещения менее приоритетная, чем загрузка или разгрузка фуры. Значения таких приоритетов вычисляются эмпирическим путем и с высокой вероятностью будут детерминированными. У задач может варьироваться сложность (перенести пару коробок или пару тонн груза). Определение сложности – вопрос отдельного исследования. Для упрощения можно измерять ее в человеко-часах, т. е. к усредненному времени выполнения.

Отдельно стоит выделить такой параметр, как спецсредства, необходимые для выполнения задачи (например, погрузчик), т. е. наличие этих спецсредств у рабочего. Использование подобных параметров делает описание закона интервальным [1]. При программном описании в таком случае имеет смысл сделать мультиочередь и описать каждый подзакон (при их возникновении) отдельно. То есть для каждого рабочего будет просматриваться своя очередь задач вместо выполнения оператора множественного выбора для каждой задачи в единой очереди.

Из параметров рабочих, влияющих на решение задачи динамического ранжирования, стоит учесть условную близость задачи к месту работы. На складах с большими площадями выполнение задачи, находящейся в противоположном от рабочего углу, будет менее выгодно, чем выполнение задачи, расположенной непосредственно там же, где находится он сам.

Следующим немаловажным параметром надо выделить накопительную нагрузку на рабочего. Предположим, что два рабочих находятся в одинаковых условиях, пользуются одинаковыми спецсредствами и получают задачи одинаковой сложности. Первый регулярно получает и выполняет задания, второй практически ничего не делает. В таком случае эффективность работы второго низкая, а первого – высокая, но снижается из-за усталости, а также износа используемых спецсредств. Таким образом, следует накапливать для рабочих сложность выполненных работ. Этот параметр не только должен влиять на ранжирование, но и использоваться для контроля. Если у отдельного рабочего необоснованно не изменяется или изменяется слишком медленно накопленная сложность выполненных работ, то его нужно контролировать.

Зададим следующие обозначения:

- приоритет задачи – ω ;
- сложность задачи – δ ;
- удаленность места выполнения задачи от рабочего – l ;
- необходимые спецсредства – k ;
- функция определения сложности выполненных задач – $s(u)$, в рамках данной статьи ее определение не рассматривается;
- веса факторов – вектор a .

В общем виде закон определения ранга задачи можно представить следующим образом:

$$F(\omega, \delta, l, k, \nu) = \begin{cases} \text{если } k = k_0, f_0(\omega, \delta, l, \nu) \\ \dots \\ \text{если } k = k_n, f_n(\omega, \delta, l, \nu) \end{cases} \quad (1)$$

Для случаев, когда $f(\omega, \delta, l, \nu)$ имеет одинаковые значения a , его можно описать как

$$f(\omega, \delta, l, \nu) = \frac{a_\omega \omega + a_\delta \frac{\delta}{s(\nu)}}{a_l} \quad (2)$$

Для выполнения свободному рабочему i должна назначаться задача с наибольшим рангом:

$$\max(F(\omega_1, \delta_1, l_1, k_1, \nu_1), \dots, F(\omega_n, \delta_n, l_n, k_n, \nu_n)), \quad (3)$$

где n – количество доступных задач.

Однако предложенная модель позволяет возникать ситуациям, когда отдельная задача находится в очереди слишком долго или же не выполняется совсем, так как для всех рабочих ее конечный ранг слишком низок. В этом случае необходимо ввести понятие динамического приоритета, который будет расти в зависимости от времени нахождения задачи в очереди [2, 3]. Не вдаваясь в подробности формирования закона зависимости приоритета от времени ожидания t , его можно выразить как $u(t)$, тогда (1) можно представить как

$$F(\omega, \delta, t, l, k, \nu) = \begin{cases} \text{если } k = k_0, f_0(\omega, \delta, t, l, \nu) \\ \dots \\ \text{если } k = k_n, f_n(\omega, \delta, t, l, \nu) \end{cases} \quad (4)$$

(2) будет выражаться как

$$f(\omega, \delta, t, l, \nu) = \frac{a_\omega (\omega + u(t)) + a_\delta \frac{\delta}{s(\nu)}}{a_l} \quad (5)$$

а определение задачи с максимальным рангом для каждого рабочего (3) дополнится еще одним параметром:

$$\max(F(\omega_1, \delta_1, t_1, l_1, k_1, \nu_1), \dots, F(\omega_n, \delta_n, t_n, l_n, k_n, \nu_n)). \quad (6)$$

При программной реализации с использованием мультиочереди (где номер очереди соответствует значению k) значение n будет обозначать не общее количество задач, а количество задач в данной очереди, поэтому определение ранга задач для отдельного рабочего (6) будет выражаться не через (4), а через (5) следующим образом:

$$\max(f(\omega_1, \delta_1, t_1, l_1, \nu_1), \dots, f(\omega_k, \delta_k, t_k, l_k, \nu_k)). \quad (7)$$

Полученные в итоге модели (6) и (7) описывают обобщенный подход к динамическому ранжированию задач для складских рабочих. Определение значений весовых коэффициентов a в общем виде или для каждой из очередей k – задача эмпирического наблюдения и имитационного моделирования, совмещенных с теорией нечетких множеств или нейронными сетями [4]. Конечной целью оптимизации будет равномерное распределение накопленной сложности выполненных задач u для всех рабочих, максимизация количества выполненных за смену задач и минимизация времени нахождения задачи t в очереди. Определение баланса между этими показателями, вывод законов $s(u)$ и $u(t)$ – цель дальнейшего исследования.

Библиографический список

1. Винокуров А. Ф., Машуров А. Ю., Левочко А. И. Ранжирование и выбор наиболее важных критериев для решения многокритериальной задачи // Молодой ученый. 2019. № 17(255). С. 8–13.
2. Коротченко А. Г., Трошина Е. А. О принципах ранжирования в задачах принятия решений // Оптимизация и моделирование в автоматизированных системах: сб. тр. Междунар. молодеж. науч. шк., Воронеж, 23–24 окт. 2019. Воронеж: ВГТУ, 2019. С. 6–9.
3. Коротченко А. Г., Трошина Е. А. О принципах «тонкого» ранжирования в задачах принятия решений // Информационные системы и технологии – 2019: сб. матер. XXV Междунар. науч.-техн. конф., Н. Новгород, 19 апр. 2019. Н. Новгород: НГТУ, 2019. С. 688–693.
4. Гапонов А. И. Синтез метода иерархий и теории нечетких множеств для решения задач ранжирования // Теория и практика экономики и предпринимательства: матер. XVI Всерос. с междунар. участием науч.-практ. конф., Симферополь–Гурзуф, 18–20 апр. 2019). Симферополь: ИП Зуева Т. В., 2019. С. 184–185.

УДК 004.4'2

А. Ф. Одегов

студент кафедры прикладной математики, информатики и информационных таможенных технологий

А. А. Сорокин – старший преподаватель – научный руководитель

ОСОБЕННОСТИ РАЗРАБОТКИ КРОССПЛАТФОРМЕННЫХ ПРИЛОЖЕНИЙ

Кроссплатформенные приложения позволяют работать в различных операционных системах, не беспокоясь о проблемах с совместимостью. Данный подход является универсальным, так как позволяет экономить ресурсы. Для этого существуют различные кроссплатформенные фреймворки.

Наиболее популярные инструменты для кроссплатформенной разработки

Рассмотрим наиболее популярные инструменты для кроссплатформенной разработки (табл. 1) [1].

Таблица 1

Кроссплатформенные Фреймворки

Фреймворк	Xamarin	Flutter	Qt	React Native
Язык	C#	Dart	C++ QML	JavaScript+React
Платформа	Android,iOS,UWP	Android,iOS ,Desktop	Android,iOS,UWP	Android,iOS,UWP

Xamarin

Платформа для создания приложений от Microsoft, поддерживает разработку для мобильных устройств и под Windows. Позволяет создавать единую логику для приложения с применением C# и .NET.

Xamarin представляет собой несколько субплатформ. Они позволяют приложению направлять запросы к интерфейсу на устройстве [2, 3].

Преимущества:

- достаточно быстро развитие;
- приложение под разные платформы будет выглядеть похожим;
- возможность использовать TestCloud для тестирования;
- возможность повторно использовать код.

Недостатки:

- Android-страницы невозможно расположить как часть уже существующего Activity/Fragment;
- реализованы не все контролы.

Flutter

Главная особенность данного фреймворка – высокая графическая производительность. Позволяет создавать приложения под три платформы:

- Android;
- iOS;
- Desktop (Windows, macOS, Linux, Google Fuchsia).

Преимущества:

- достаточно быстро развитие;
- собственный графический движок.

Недостатки:

- интерфейс создается с помощью кода;
- громоздкий размер установочного пакета из-за виртуальной машины.

Qt

Платформу стоит рассматривать как полноценный набор инструментов на любой случай. Существует возможность разрабатывать не только с помощью C++, но и при помощи QML, который имеет сильное сходство с JavaScript.

Преимущества:

- имеется большое количество хороших инструментов, которые могут пригодиться в разработке;
- существуют библиотеки, содержащие понятные api-интерфейсы.

Недостатки:

- сложен в освоении.

React Native

Инструмент позволяет легко создавать интерфейсы. Среда поставляется с большим набором компонентов, что является плюсом, однако они не всегда могут быть адаптированы под разные платформы. Это потребует дополнительной корректировки в коде.

Преимущества:

- возможность повторного использования кода;
- интегрируемые нативные компоненты.

Недостатки:

- ограниченный API;
- низкая производительность.

Плюсы нативных приложений

Высокая производительность. Так как при разработке использовались технологии, связанные с определенной платформой, код будет иметь доступ ко всем функциям системы, что повышает производительность приложений. Это означает что при создании приложений с высокой нагрузкой использование нативного кода поможет снизить время отклика, вероятность сбоев и т. д.

Использование всех возможностей платформы. Приложения разрабатываются под конкретные задачи для определенных платформ. И это приводит к наиболее хорошему соответствию возможностей приложения аппаратным возможностям. Это нужно, когда приложению необходимо иметь доступ к разным видам данных [4].

Пользовательский интерфейс. Приложения интегрируются напрямую с операционной системой, могут использовать все возможности «железа», пользователи без проблем могут пользоваться привычным интерфейсом, что обеспечивает положительный опыт использования. Например, в данный момент при огромной вариативности разрешений экранов очень важно оптимизировать приложения под определенный экран, чтобы пользователям было комфортно при использовании приложения.

Лучшее позиционирование в магазинах приложений. Пользовательский опыт – один из важнейших показателей в рейтинге приложения. Если данная оценка высока, то оно будет высоко оценено магазином и чаще попадать в рекомендации для различной аудитории, что соответственно увеличит доход.

Минусы нативных приложений

Создание приложений под несколько платформ сразу может значительно замедлить процесс разработки. Одинаковый код не получится развернуть на разных платформах, и разработчикам потребуется дополнительное время для написания кода под другие платформы, что естественно увеличит время и затраты.

Если компании необходимо создавать для каждой платформы отдельные приложения, ей, скорее всего, потребуется нанять дополнительных сотрудников, которые будут сосредоточены на разработке под конкретную платформу.

При разработке приложений под конкретную операционную систему придется смириться с тем, что оно будет несовместимо с остальными платформами, так как при разработке под конкретную систему используются языки, подходящие только для этой системы.

Разработка для определенной операционной системы приводит к упущению возможностей. Это заведомо сокращает целевой рынок что приведет к потере прибыли.

Плюсы кроссплатформенных приложений

Один код доступен для повторного использования на других платформах

Главное преимущество в том, что одинаковый код может быть использован на разных мобильных платформах. Этим и отличается кроссплатформенная разработка от нативной: она не требует использования другого технического стека для разных ОС. Использование такого кода позволяет с легкостью развертывать приложения на других системах, использовать возможности, реализованные для одной платформы, на других [5].

Кроссплатформенная разработка менее ресурсозатратна. Нет необходимости нанимать дополнительных сотрудников для разработки под отдельные платформы, так как возможно использовать один и тот же стек.

Более простой и быстрый способ развертки. Нет необходимости в изучения нескольких технологических стеков для различных платформ перед разработкой приложений, хватит хорошего знания одного стека разработки и его применения.

Так как нет необходимости создавать разные основы приложения, стартовый этап развертывания на разных ОС происходит намного быстрее. Кроме этого, все изменения в будущем могут быть выполнены одновременно без внесения изменений отдельно для каждой платформы.

Кроссплатформенность как способ охвата более широкой аудитории. При такой разработке у приложения есть возможность для более широкого охвата аудитории, так как такое приложение сможет достигнуть всех платформ независимо от используемой ими ОС. Такой способ более выгоден для бизнеса, чем разработка для конкретной платформы.

Возможность не изменять интерфейс и UX. Внешний вид приложения и опыт использования пользовательского интерфейса не менее важны, чем его производительность. Работа одной команды и одного кода позволяет компаниям не изменять внешний вид разрабатываемого продукта для каждой платформы. Это означает, что оно будет выглядеть одинаково для всех платформ.

Минусы кроссплатформенных приложений

Нативные приложения более гибкие по сравнению с кроссплатформенными. При данном подходе будут реализовываться задачи для всех платформ, но, вероятнее всего, не получится адаптировать приложения для использования всех возможностей для всех платформ. Так как работа с одним стеком не обеспечивает той же гибкости и оптимизации, как нативная разработка отдельно для каждой ОС.

Пониженная производительность по сравнению с нативными приложениями. При работе с одним и тем же стеком будет страдать гибкость приложения. Также это означает потерю возможности улучшения производительности.

Расхождения в UI в зависимости от платформы. Настройка пользовательского интерфейса для различных ОС может доставить проблемы. Так как у разных платформ существуют требования к дизайну, в некоторых случаях они могут оказаться взаимоисключающими.

Сложности в загрузке кроссплатформенных приложений в соответствующие магазины. При добавлении кроссплатформенного программного продукта в магазин могут возникнуть проблемы, так как требования к программному продукту будут различаться, и это вызовет трудности в прохождении и выполнении всех правил магазина.

Библиографический список

1. Майер Р. Android 4. Программирование приложений для планшетных компьютеров и смартфонов / пер. с англ. ООО «Аудиономикс». М.: Эксмо, 2013. 816 с.

2. Подробно о Xamarin. URL: <http://habrahabr.ru/post/188130/> (дата обращения: 22.04.2021).
3. Джозеф Х. Unity в действии. Мультиплатформенная разработка на C#. М.: Питер, 2016. 336 с.
4. Джесс Р., Рональд К. PhoneGap. М., 2013. 104 с.
5. Захаров В., Мостяев А. Особенности переноса приложений на мобильные платформы // Программные системы и инструменты / под ред. А. Н. Терехина. Т. 15. М.: Издат. отд. ф-та ВМК МГУ, 2014. С. 16–24.

УДК 004.92

М. С. Смирнов

студент кафедры прикладной математики, информатики и информационных таможенных технологий
Р. А. Коваленко – старший преподаватель – научный руководитель

ПРОЦЕСС ПОКАДРОВОГО РЕНДЕРИНГА С ПРИМЕНЕНИЕМ ГРАФИЧЕСКИХ АРІ

Введение

В первую очередь API (англ. Application Programming Interface) – это описание способов, которыми одна компьютерная программа может взаимодействовать с другой. Проще говоря, это набор функций для взаимодействия как с программами, так и, как в нашем случае, с драйвером графического устройства компьютера и не только. За все время развития IT-сферы появилось огромное количество графических API разного рода и задач. Но чтобы показать взаимодействие рендеринга кадров и API, будет достаточно взять один из наиболее популярных. Например, Vulkan API.

Процесс покадрового рендеринга с применением Vulkan API

Vulkan – графический API для создания 2D- или 3D-графики, разработанный создателями OpenGL – Khronos Group. Этот API дает возможность обрабатывать приложения с 3D-графикой в реальном времени и с хорошей производительностью, как и OpenGL, а также, по сравнению с его конкурентами DirectX 12 или Metal, показывает лучшую производительность при меньшей нагрузке на процессор. Был разработан на основе технологий AMD – Mantle.

Vulkan работает при помощи двух понятий – хост и устройство (host и device). Хост отправляет команды на устройство, а оно их выполняет. По факту, приложение, которое использует Vulkan и является хостом. Из-за того что Vulkan разработали создатели OpenGL, название функций (синтаксис) схожи с ним. Все команды, связанные с Vulkan, имеют вначале приписку vk (prefix). Функции выглядят примерно так: vkCreateSomething(...), структуры и хендлы тоже используют подобный префикс, например VkSomething [1].

Рендер кадра

Чистый Vulkan не имеет строгих проверок отправляемых данных на проблемность. Он выполняет то, что ему сказали, даже если это может привести к ошибке. Это сделано в угоду производительности. В любом случае дается возможность подключить дополнительные проверочные слои и расширения.

Все команды, отправленные хостом устройству, помещаются в командный буфер. Проще говоря, в Vulkan нет функции, которая могла бы заставить выполнить что-то сразу. Имеются только функции, предназначенные для заполнения буфера. Только после заполнения буфера на хосте, становится возможной отправка этих данных в очередь, которая находится на устройстве. Существуют всего две разновидности командных буферов:

- 1) первичный сразу отправляется в очередь;
- 2) вторичный запускается в первичном, поэтому не может быть отправлен в очередь.

Порядок записи команды и их последующее поступление в очередь происходят в таком же порядке, в каком вызывались функции для командного буфера. Однако их выполнение будет происходить в почти случайном порядке. Чтобы не было хаотичности, программисты из Khronos Group придумали средства для синхронизации.

Хосту сразу отдается управление после отправления команд и не ожидается завершения выполнения команд из буфера, если Вы не укажите это явным способом. Также выделение памяти для выполнения задач всецело ложится на хост.

Vulkan имеет два конвейера: графический и вычислительный. Часто из вычислительного блока данных отправляются в графический (рис. 1). Изменить конвейер невозможно. Исключение – программируемые стадии рендеринга, иначе говоря, шейдеры [2].

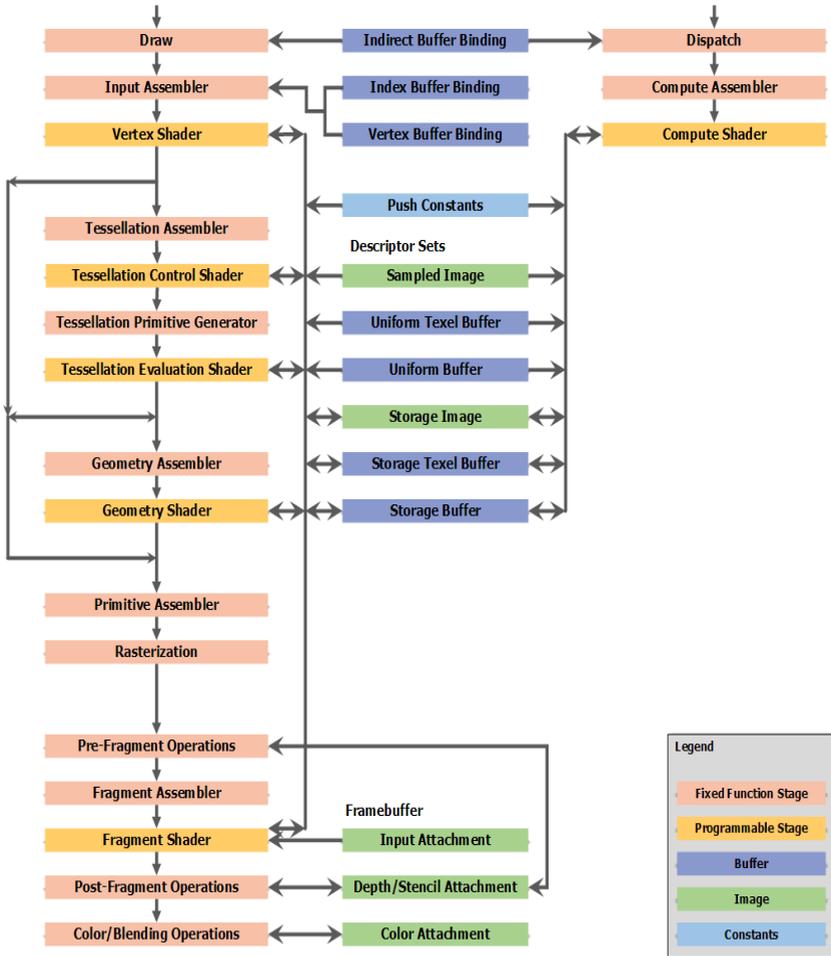


Рис. 1. Конвейеры Vulkan API

Это самое интересное, ведь пайплайн является отображением почти всего процесса рендеринга кадра. Проще говоря, весь процесс обработки кадра при помощи графического API можно изобразить в виде конвейера. При этом сам пайплайн для каждого API уникален, поэтому для каждого API будет свой конвейер. На рис. 2 можно наблюдать такой для Vulkan API [3, 4].

Графический пайплайн показывает, в какой последовательности начинается обработка вершин, шейдеров и т. д.

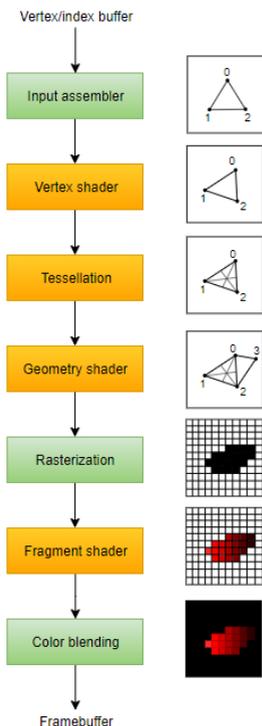


Рис. 2. Графический пайплайн

Заключение

Графический API всецело влияет на процесс обработки кадров. На выделение памяти, графический пайплайн и даже нагруженность процессора при обработке графики влияет способ реализации той или иной функции в API.

Библиографический список

1. Что такое API // Хабр: сайт. URL: <https://habr.com/ru/post/464261/> (дата обращения: 03.04.2021).
2. Vulkan API (glNext) от Khronos Group // Хабр: сайт. URL: <https://habr.com/ru/post/283490/> (дата обращения: 03.04.2021).
3. Introduction – Vulkan Tutorial // Vulkan Tutorial: сайт. URL: https://vulkan-tutorial.com/Drawing_a_triangle/Graphics_pipeline_basics/Introduction (дата обращения: 03.04.2021).
4. Vulkan API // Khronos Group: сайт. URL: <https://www.khronos.org/vulkan/> (дата обращения: 03.04.2021).

УДК 004.921

А.С. Столярчук

студент кафедры прикладной математики, информатики и информационных таможенных технологий

А. А. Сорокин – старший преподаватель – научный руководитель**СРАВНЕНИЕ МЕТОДОВ СОЗДАНИЯ ПРОВОДОВ В UNITY****Введение**

Unity – среда разработки компьютерных игр, обладающая наличием визуальной среды разработки, межплатформенной поддержки и модульной системы компонентов.

Создание провода с использованием Hinge Joint

Hinge Joint – это группа из двух Rigidbodies, заставляющая их перемещаться совместно, словно они соединены петлей [1]. Это идеально подходит для дверей, но также может быть использовано для моделирования цепей, маятников и пр.

Для создания провода будут использоваться цилиндры и сферы, соединенные друг за другом. Сфера будет участвовать в качестве сустава между цилиндрами. Красный провод мультиметра создадим из Hinge Joint. Провод будет идти от «кусачек» и до соответствующего входа мультиметра, изображенного на рис. 1.

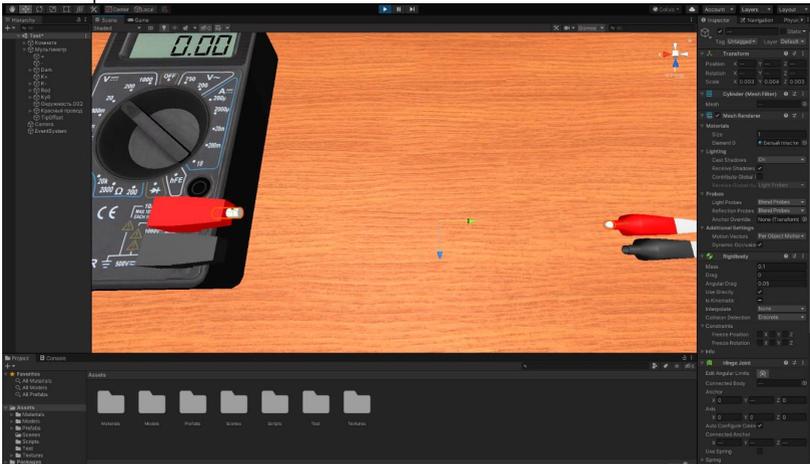


Рис. 1. Мультиметр

Создание провода идет от сферы, расположенной впритык ко входу «кусачек». На сферу необходимо добавить коллайдер, Rigidbody и Hinge Joint, у которого в параметрах Connected Body указаны «кусачки». Затем к сфере добавляется цилиндр с такими же параметрами, за исключением настроек последнего. В Connected Body у цилиндра необходимо указать сферу, к которой он будет прикреплен. И в такой последовательности добавляя сферу/цилиндр, удлиняем провод, указывая в Connected Body каждого объекта тот объект, который следует перед ним. В конце провода оставим цилиндр в включенном Is Kinematic в параметрах Rigidbody. Конец провода будет расположен в красном контакте мультиметра, где расположен белый цилиндр на рис. 1. В итоге получаем провод, который обладает коллайдером и Rigidbody (физикой) (рис. 2). Минус такого провода в том, что при удлинении он разваливается, т. е. если «кусачки» потянуть дальше от мультиметра, то все сферы и цилиндры, из которых состоит провод, начнут отдаляться друг от друга, а потом попросту разлетятся (рис. 3).

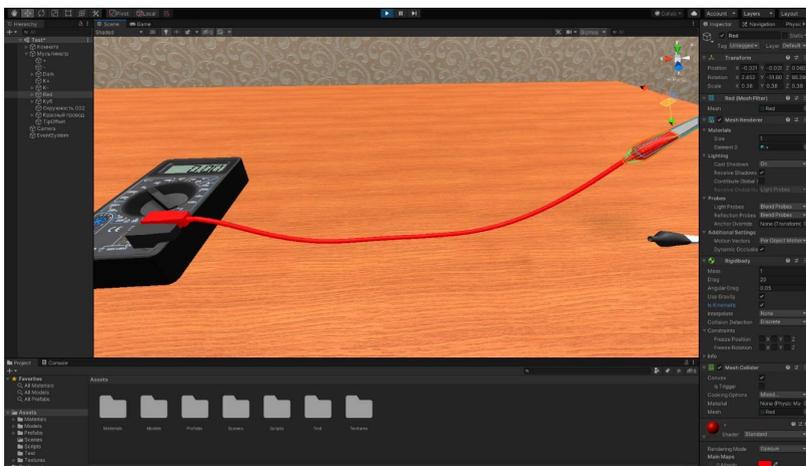


Рис. 2. Провод из Hinge Joint

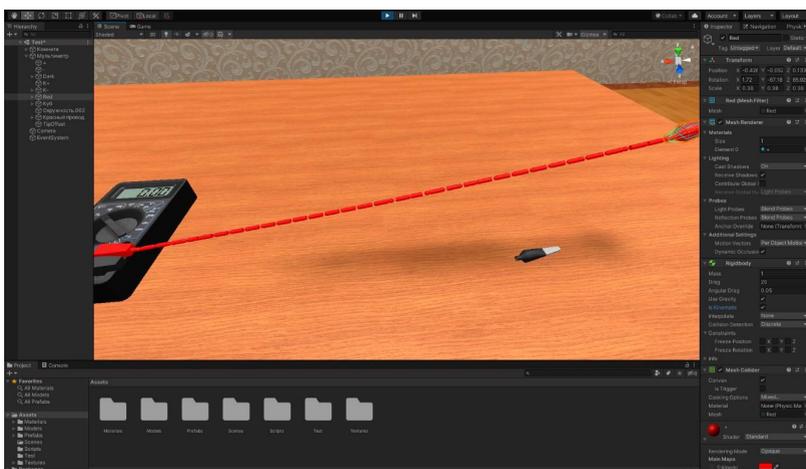


Рис. 3. Провод из Hinge Joint

Создание провода с использованием скрипта

Скрипт скачан с сайта <https://github.com/NoxWings/Cable-Component> [2]. По словам автора, он представляет собой простую и оптимизированную реализацию кабельной физики в Unity3D, в нем используется интеграция Верле для моделирования физики, как и в подходе Unreal Engine 4. Также автор приложил демонстрацию своей работы (рис. 4–5).

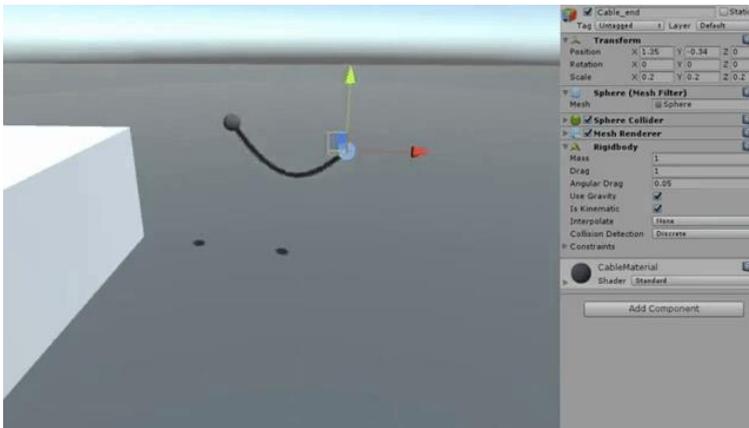


Рис. 4. Демонстрации скриптового провода

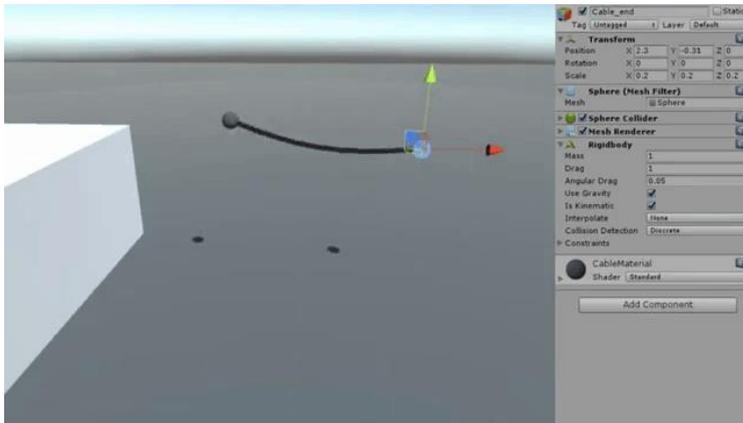


Рис. 5. Демонстрации скриптового провода

В целом выглядит неплохо. Попробуем использовать это для создания второго провода.

Для создания провода из скрипта требуются два объекта: первый – начало провода, второй – соответственно конец. В качестве объектов будут использоваться две сферы. Как и в создании предыдущего провода, первая сфера располагается во входе «кусачек», а вторая – так же у соответствующего входа мультиметра. Обе сферы имеют коллайдер и Rigidbody с включенным Is Kinematic. Все, что осталось сделать, это добавить на одну из сфер сам скрипт, указав в нем вторую сферу как конец провода, и добавить Spring Joint, также указав в нем вторую сферу. Дополнительно можно окрасить сферы и материал в черный цвет, установить длину и толщину провода, и можно приступить к тестированию.

Такой провод выглядит менее реалистично, чем первый, а также не обладает коллайдерами, что позволяет ему проходить сквозь объекты (рис. 6). Его преимущество, в отличие от первого, в том, что его можно растягивать, он не рассыпается, и его создание требует намного меньше времени (рис. 7). Влияние на производительность у данного провода меньше, чем у предыдущего, из-за отсутствия коллайдеров и дополнительных объектов.

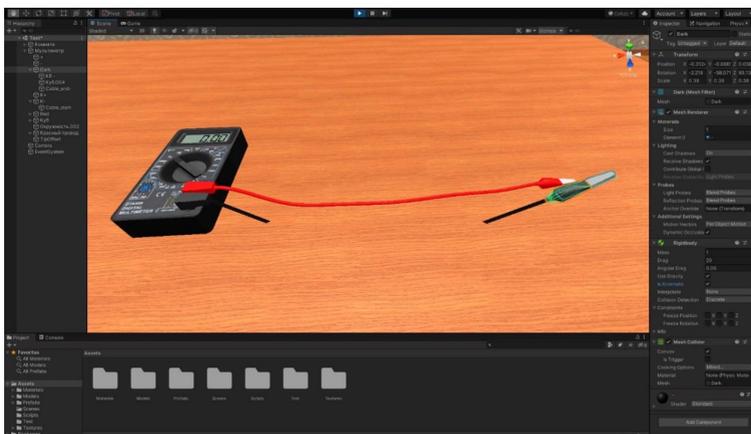


Рис. 6. Заскриптованный провод

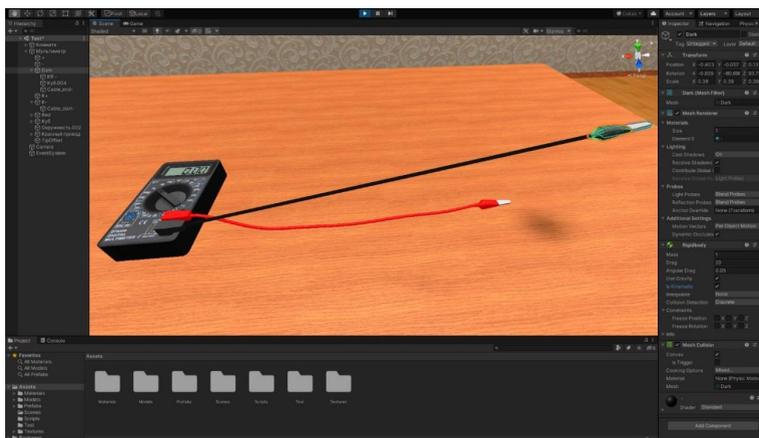


Рис. 7. Заскриптованный провод

Вывод

Оба провода неидеальны и обладают как плюсами, так и минусами. Если необходимо создание простого провода с минимальными затратами времени и ресурсов, то провод из скрипта подойдет лучше всего. Если нужен реалистичный провод, то придется использовать Hinge Joint.

Библиографический список

1. Unity User Manual // Unity Documentation: сайт. URL: <https://docs.unity3d.com/Manual/index.html> (дата обращения: 28.02.2021).
2. NoxWings/Cable-Component // GitHub: сайт. URL: <https://github.com/NoxWings/Cable-Component> (дата обращения: 29.02.2021).

УДК 004.383.3, 004.42:534.83

Р. Р. Шайхутдинов

студент кафедры прикладной математики, информатики и информационных таможенных технологий

А. В. Дагаев – кандидат технических наук – научный руководитель

РАЗРАБОТКА АУДИОУСТРОЙСТВА ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ НА ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ МИКРОКОНТРОЛЛЕРАХ КОМПАНИИ STMICROELECTRONICS

Сегодня технологии ЦОС внедрились во многие сферы деятельности человека, используют собственные, глубоко проработанные направления со своими алгоритмами, математическим аппаратом и специализированными методами [1]. Одно из направлений ЦОС – обработка аудиосигнала в реальном времени, которое нашло применение в концертно-сценическом и студийном оборудовании, для обработки голоса и инструментов. К ним относятся устройства динамической обработки или амплитудного преобразования (компрессор, овердрайв и т. д.), частотного преобразования (фильтры и эквалайзеры) и временного преобразования (эхо/дилэй, реверберация, хорус и т. д.); в их основе лежат фундаментальные концепции ЦОС, такие как аналого-цифровое (далее АЦП) и цифро-аналоговое (далее ЦАП) преобразование, фильтры с конечной и бесконечной импульсной характеристикой, преобразование Фурье и т. д. Существуют приборы разных видов и классов, соответственно, они имеют широкий ценовой диапазон, причиной чему служат основные факторы – качество/уникальность аудиозффекта, отказоустойчивость, функциональность и выходные характеристики (класс) прибора, которые в свою очередь напрямую зависят от используемых электронных компонентов, проработанной схемотехники и качества выполняющего программного кода.

В основе устройств обработки звука (рис. 1) лежит цифровой сигнальный процессор, именно он и выполняет основные задачи по обработке звукового сигнала. Помимо ЦСП, в звуковом тракте важную роль играют устройства для преобразования аналогово звука в цифровой и обратно – ЦАП и АЦП; если они выполнены на одной микросхеме, то устройство называется Codec. Для контроля (кнопки, энкодеры) и вывода информации на дисплей в паре с ЦСП используют дополнительный микроконтроллер, дабы не нагружать процессор обработкой не связанных со звуком команд. Часто в конструкцию приборов включают аналоговые фильтры как в начале, так и в конце пути прохождения звука для его предварительной и постобработки. Рассмотрим отдельно каждый из этих пунктов.

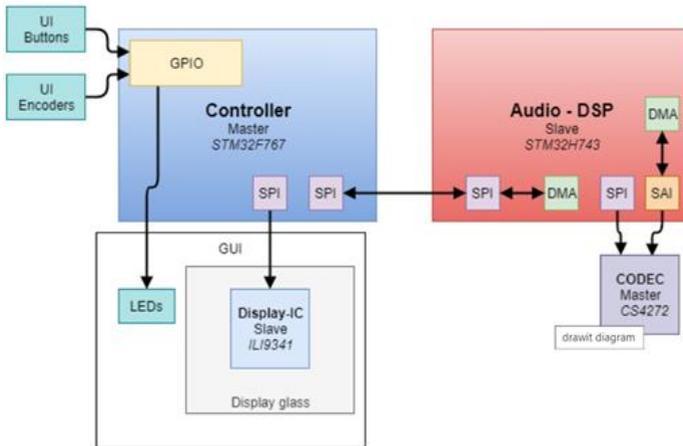


Рис. 1. Аппаратная схема процессора эфффектов

В сравнении с микропроцессорами общего применения, архитектура DSP имеет ряд особенностей, призванных увеличить скорость выполнения типовых задач ЦОС, что является главным условием, потому как в устройстве, работающем в реальном времени, задержка сигнала должна быть минимальной. Многие компании имеют линейки интегральных микросхем, решающих задачи аудиоЦОС, такие как Analog Devices или Texas Instruments, но из-за высоких цен на их оценочные и отладочные платы приходится рассматривать альтернативные варианты. Наиболее актуальна сегодня микропроцессорная архитектура ARM, доминирующая в производстве 32-битных и 64-битных ядер и используемая многими производителями микропроцессоров и микроконтроллеров. На данный момент используется несколько значимых семейств процессоров и самым популярным сейчас стало новое семейство Cortex-M. Особо интересна платформа для MCU Cortex-M7, производителем она нацелена на использование во встраиваемых системах достаточно широкого диапазона применения, от автомобилей до промышленного оборудования. Процессор обладает достаточно высокой производительностью, на борту имеет блок, производящий операции с плавающей запятой с поддержкой чисел одинарной и двойной точности, полный набор DSP-инструкций, аппаратные умножение и деление. Ряд производителей микроэлектроники, например STMicroelectronics, будучи лицензиатом готовой топологии ядер ARM, разрабатывают микроконтроллеры, внося свои изменения, комбинируя ядра, дополняя их периферийными интерфейсами и памятью. Один из новых в линейке высокопроизводительных MCU – STM32H743VI, он содержит большой объем RAM и FLASH памяти, а также все необходимые блоки периферии (рис. 2).



Рис. 2. Рост производительности MCU STM32

В сравнении с предшественником STM32F7, он имеет новую, усовершенствованную шинную архитектуру, объединившую процессорное ядро и блоки памяти, измененную систему тактирования и питания, что и стало причиной роста производительности. Chin Beckmann, являющаяся соучредителем и генеральным директором компании DSP Concepts, провела сравнительный тест бенчмарка на производительность DSP-процессоров от Analog Devices и MCU STM32, в котором STM32H7 показал очень хорошие результаты [2].

Одним из изменений в периферийных устройствах стало появление 16-битного АЦП, но его работу заменит аппаратный аудиокодек (codec). Codec – это микросхема, которая объединяет в едином корпусе модули АЦП и ЦАП, в ней и происходит кодирование/декодирование сигнала (рис. 3). Сам codec «общается» с микроконтроллером по одному из интерфейсов SPI/I2S или SAI, последний наиболее актуален, так как специально разработан для обмена данными с внешними звуковыми устройствами.

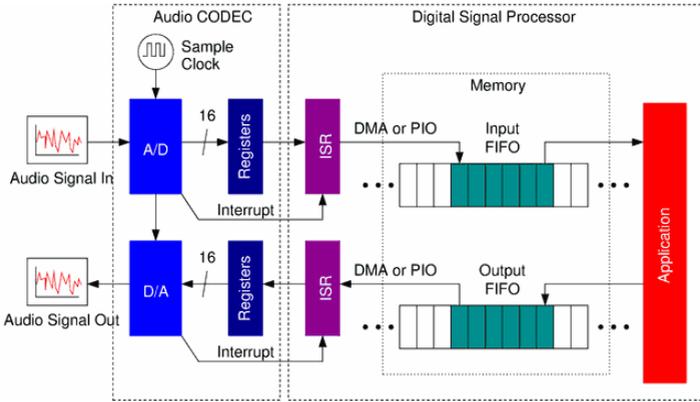
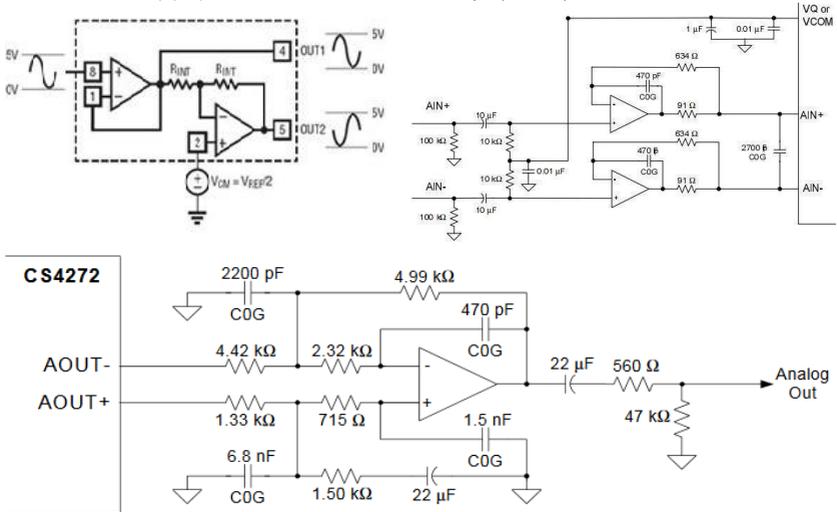


Рис. 3. Общая схема взаимодействия кодека и MCU

Наиболее важными характеристиками, которыми должен обладать содек, являются:

- поддержка качества обработки сигнала 24-бит/48 КГц;
- высокий динамический диапазон АЦП/ЦАП;
- уровень гармонических искажений;
- низкий коэффициент нелинейных искажений + Шум (THD+N).



слева – для несимметричного/инструментального входа
 справа – для дифференциального входа
 снизу – выходной фильтр кодека

Рис. 4. Схема входных буферов кодека

Подходящими параметрами обладает codec от Cirrus Logic CS4272, имеющий дифференциальную аналоговую архитектуру с высокой производительностью, регулировки выходной громкости с плавным нарастанием и т. д. Стоит обратить внимание на техническую документацию от производителя, в даташите [3] есть рекомендации по проектированию «обвязки» кодека на печатной плате.

В конструкциях большинства устройств аудиообработки используются два типа входа: небалансный/инструментальный и дифференциальный. Кодек имеет дифференциальный вход, поэтому необходимо предусмотреть буфер для преобразования несимметричного сигнала 0–5 В в дифференциальный сигнал ± 5 В (рис. 4, слева). Для дифференциально входа устройства можно использовать схему из даташита (рис. 4, справа). Обе схемы на входе создают высокоимпедансную нагрузку, что позволит подключать микрофон и инструменты со звукоинимателями, также они достаточно эффективно подавляют синфазный шум, вызванный высокой длиной кабеля. Для аналогового выхода кодека, также будет применена схема из даташита (рис. 4, снизу).

Для регулировки параметров звукового эффекта в устройстве применяются инкрементальные энкодеры (рис. 5), работа которых заключается в подсчете импульсов, соответствующих количеству поворотов вала на определенный угол.

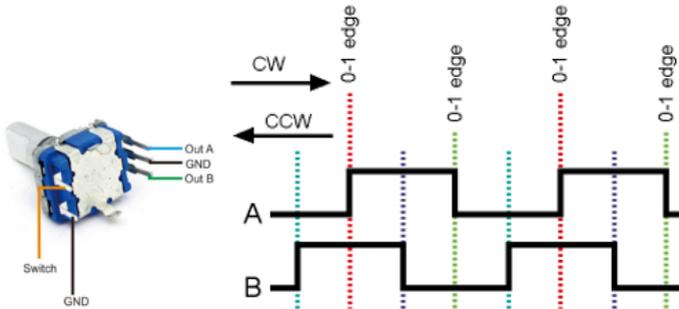


Рис.5. Энкодер и его работа

Визуальный контроль управления параметрами эффектов осуществляется с помощью TFT LCD или OLED-дисплеев (рис. 6).



слева - TFT LCD дисплей, справа – OLED дисплей

Рис. 6. Дисплеи для отображения информации

При подключении дисплея и энкодеров следует использовать дополнительный микроконтроллер, чтобы дополнительно не нагружать основной процессор. Помимо ввода-вывода информации, с ним возможна реализация записи и хранения удачно отрегулированных параметров эффекта, реализовав меню с сохранением пресетов. С этой задачей прекрасно справятся микроконтроллеры из младших линеек, STM32F7 или STM32F4. Для подключения дисплея и сопряжения микроконтроллеров используется SPI интерфейс, так как он обладает высокой скоростью передачи данных, требует всего четыре контакта для подключения и является наиболее простым в использовании.

Библиографический список

1. Смит С. Цифровая обработка сигналов. Практическое руководство для инженеров и научных работников / пер. с англ. А. Ю. Линович, С. В. Витязев, И. С. Гусинский. М.: Додэка-XXI, 2012. 720 с.
2. Beckmann C. How to choose the best processor for your audio DSP application? URL: <https://community.arm.com/developer/ip-products/system/b/embedded-blog/posts/how-to-choose-the-best-processor-for-your-audio-dsp-application> (дата обращения: 05.04.15).
3. CS4272 24-Bit, 192 kHz Stereo Audio CODEC [Datasheet] / Cirrus Logic, Inc. URL: https://statics.cirrus.com/pubs/proDatasheet/CS4272_F1.pdf (дата обращения: 05.04.15).

УДК 51-77:378

Д. П. Шаповалова

студент кафедры прикладной математики, информатики и информационных технологий

Е. А. Яковлева – кандидат физико-математических наук, доцент – научный руководитель**РАЗРАБОТКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ ВЗАИМОДЕЙСТВИЯ ФИЛИАЛА И ГОЛОВНОГО ВУЗА**

Российские вузы имеют филиалы в разных городах России. Рассмотрим положительные стороны, которые дает обучение в филиале [1,2]:

- в дипломе указывается только название головного вуза и не отмечено, что выпускник обучался в филиале,
- образовательные стандарты в головном вузе и в филиале одинаковые,
- нередко проходные баллы по ЕГЭ ниже, чем в головном вузе,
- стоимость обучения на контрактной основе ниже,
- обычно все факультеты располнены компактно или в одном здании, что сокращает время переезда из корпуса в корпус,
- все студенты имеют доступ в одну и ту же образовательную среду,
- в филиале число обучающихся меньше, поэтому преподаватели могут уделять им больше внимания,
- многие преподаватели работают в головном вузе и в филиале, вузы имеют программу мобильности, по которой студенты филиала могут обучаться несколько семестров в головном вузе,
- может появиться возможность перевести в головной вуз.

С другой стороны, есть и некоторые минусы при обучении в филиале [1, 2]:

- может реализовываться меньшее количество образовательных программ,
- штат профессорско-преподавательского состава в филиале меньше,
- студентам, проходящим обучение в филиале, не всегда доступна инфраструктура крупного вуза,
- уровень преподавательского состава может оказаться ниже, чем в головном вузе, чем выше уровень головного вуза, тем выше квалификация преподавателей филиала,
- филиал находится не в крупном городе, поэтому отсутствуют возможности, которые предоставляет крупный город в плане отдыха и работы.

Каждый вуз сталкивается с проблемой финансирования, поэтому ему необходимо усилить поток финансовых средств, основным источником которых выступает оказание образовательных услуг на коммерческой основе.

При этом возникает ряд взаимосвязанных задач:

- оказание коммерческих образовательных услуг как можно большему количеству возможных заказчиков;
- предоставление образовательных услуг различных формах, включая дистанционное обучение;
- обеспечение максимального эффекта от реализации образовательных услуг для вуза.

Один из путей решения этих задач – оптимальное распределение бюджетных мест между головным вузом и филиалом и оптимальным набором обучающихся на коммерческой основе. Обычно филиал определяет потребности в специалистах, при этом он руководствуется региональными потребностями. Головной вуз устанавливает контрольные цифры приема, окончательный план вырабатывается и согласовывается на уровне управления филиалом. Возникает необходимость построения совместной стратегии головного вуза и филиалов, которая учитывает интересы и возможности обеих сторон.

Для решения таких задач можно использовать методы линейного и нелинейного программирования, сетевые методы, возможности теории игр и эконометрического анализа.

Одна из моделей – математическая модель директивного управления распределением образовательных услуг в системе «головной вуз – филиал» [3]. В научных работах были рассмотрены подоб-

ные задачи [1–3]. В частности, была построена математическая модель, которая приводилась к закрытой транспортной задаче, которая решалась не на минимум, а на максимум:

$$Z = \sum_{i=1}^m \sum_{j=1}^n C_{ij} \cdot x_{ij} \Rightarrow \max,$$

$$\sum_{i=1}^m x_{ij} = M_j, j=1, 2, \dots, n,$$

$$\sum_{j=1}^n x_{ij} = N_i, i=1, 2, \dots, m.$$

Введены следующие обозначения: $F_i, i = 1 \dots n$ – количество филиалов, N_1, N_2, \dots, N_m – группы, которые набираются в филиале по разным специальностям и формам обучения, C_{ij} – эффективность обучения одной группы студентов вида i ($i = 1, 2, \dots, m$) в филиале F_j ($j = 1, 2, \dots, n$). В качестве неизвестных в математической модели являются переменные x_{ij} студентов i -й специальности или формы обучения, определенные для набора в j -м филиале. Согласно модели, определяется оптимальный план распределения суммарного набора студентов по филиалам, при котором экономическая эффективность окажется наибольшей [3].

Построенная математическая модель является моделью директивного управления и имеет следующие недостатки: запрещает изменение структуры плана набора, сохраняя общий план набора студентов. В связи с этим требуется построить более гибкую модель набора, которая будет ближе к реальным условиям и позволит, с одной стороны, вносить изменения в структуру набора, ориентироваться и на максимальную экономическую эффективность, и на требования головного вуза на структуру плана набора.

С решением такой задачи может справиться теория игр, для этого следует определить стратегии, которых будут придерживаться головной вуз и филиалы [4, 5]. Филиал может изменять в установленных пределах количество студентов, определенные ему на основании «директивной» модели управления. Головной вуз, допуская возможность изменения структуры оптимального плана набора, должен располагать несколькими стратегиями для своих филиалов. Каждая стратегия представляет собой оптимальный план распределения набора, полученный по модели директивного управления. В табл. 1 представлены стратегии набора A_i ($i = 1, 2, \dots, k$) головного вуза по отношению филиалам.

Стратегии B_j – это стратегии филиалов при формировании набора студентов. Эти стратегии различаются степенью самостоятельности ограниченного планом головного вуза с оптимальным планом распределения студентов по направлениям и формам обучения, а, с другой стороны, филиал может предложить свой вариант плана набора в зависимости от его возможностей и спроса на образовательные услуги в регионе. Филиалу необходима наибольшая экономическая эффективность, которая может не достигаться при квоте, которая ему предоставляется.

Таблица 1

Матрица стратегий поведения института и филиалов

A_i	B_1	B_2	B_l
A_1	b_{11}	b_{12}	b_{1l}
A_2	b_{21}	b_{22}	B_{2l}
.....
A_k	b_{k1}	b_{k2}	b_{kl}

Математическая модель такой задачи является моделью антагонистической игры. Величина b_{ij} – это проигрыш филиала при реализации стратегии A_i головным вузом, при использовании филиалами (филиалом) стратегии B_j . Фактически b_{ij} является разностью между эффективностью, полученной

ИВАНГОРОДСКИЙ ФИЛИАЛ

в результате реализации стратегии головного вуза, и эффективностью, полученной при реализации стратегии, предлагаемой филиалом. Табл. 1 представляет собой платежную матрицу антагонистической игры.

Рассмотрим пример взаимодействия на примере, когда головной вуз имеет стратегии A_1, A_2, A_3, A_4 , а действия филиалов – стратегии B_1, B_2, B_3, B_4 , или когда у вуза один филиал, который предлагает четыре стратегии поведения. В результате получаем антагонистическую игру (табл. 2).

Таблица 2

A_i	B_1	B_2	B_3	B_4
A_1	0	5	1	-3
A_2	-3	4	2	5
A_3	7	8	-1	2
A_4	6	6	0	3

Удалим заведомо проигрышные стратегии. В столбце, соответствующем стратегии B_2 , у филиала наибольшие проигрыши, поэтому ее исключаем. Изменяем платежную матрицу и определяем нижнюю и верхнюю цену игры (табл. 3).

Таблица 3

Модифицированный результат реализации различных стратегий института и филиалов

A_i	B_1	B_3	B_4	$\min_j b_{ij}$
A_1	0	1	-3	-3
A_2	-3	2	5	-3
A_3	7	-1	2	-1
A_4	6	0	3	0
$\max_i b_{ij}$	7	2	5	

Нижняя цена игры равна $\alpha = \max_i \min_j b_{ij} = 0$, а верхняя $\beta = \min_j \max_i b_{ij} = 2$. Верхняя и нижняя цены игры не совпадают, поэтому задача не решается в чистых стратегиях. Далее построим матрицу риска R , элементы которой вычисляются следующим образом: $r_{ij} = \max_j b_{ij} - b_{ij}$:

$$R = \begin{pmatrix} 7 & 1 & 8 \\ 10 & 0 & 0 \\ 0 & 3 & 3 \\ 3 & 2 & 2 \end{pmatrix}.$$

Сравним матрицу рисков с табл. 2: стратегии института A_2 и A_3 в сочетании с состоянием среды филиалов B_3 и B_4 дают ему одинаковый прирост экономической эффективности: $b_{23} = b_{34} = 2$, но $r_{23} = 0$, $r_{34} = 3$. Это означает, что наименьший риск реализуется в случае парной стратегии A_2B_3 .

Рассмотрим критерии определения оптимальной стратегии института при условии случайной реализации состояний действия филиалов (табл. 4).

Критерии определения оптимальной стратегии

Название критерия	Вероятностный критерий	Максиминный критерий Вальда	Критерий минимального риска Сэвиджа	Критерий Гурвица
Содержание критерия	максимум математического ожидания	выбирается стратегия A_i , которая соответствует низшей цене игры,	величина риска r_{ij} , наименьшая при самых неблагоприятных B_j	Учитывается пессимистическое и оптимистическое решение. Уровень оптимизма 0,5
стратегия	A_4	A_3, A_4	A_3, A_4	A_3, A_4

По результатам расчетов, оптимальной признается стратегия A_4 . Корпоративная модель позволяет учесть и самостоятельность филиалов и определить головному вузу оптимальную стратегию при выстраивании отношений с филиалами.

Библиографический список

1. Яковлева Е. А., Сорокин А. А., Коваленко Р. А. Математическая модель директивного управления распределением образовательных услуг в системе «Головной ВУЗ – Филиал» // European Scientific Conference: сб. ст. XXIV Междунар. науч.-практ. конф. Пенза: МЦНС «Наука и Просвещение», 2021. С. 11–14.
2. Формирование нелинейной системы высшего образования в макрорегионе: монография / Г. Е. Зборовский, П. А. Амбарова, В. С. Каташинских [и др.]; под ред. Г. Е. Зборовского. Екатеринбург: Гуманитар. ун-т, 2018. 251 с.
3. Никитин С. И. Экономико-математические метод моделирования процессов сферы сервиса: учеб. пособие / под ред. д. э. н., проф. И. Н. Гаврильчака. СПб.: Изд-во СПбГАСЭ, 2004. 139 с.
4. Вентцель Е. С. Исследование операций: задачи, принципы, методология: учеб. пособие. 5-е изд., стер. М.: КНОРУС, 2013. 192 с.
5. Исследование операций в экономике: учеб. для акад. бакалавриата / Н. Ш. Кремер, Б. А. Путко, И. М. Тришин, М. Н. Фридман; под ред. Н. Ш. Кремера. 3-е изд., перераб. и доп. М.: Юрайт, 2017. 438 с.

УДК 519.83+347.453

К. В. Шубин

студент кафедры прикладной математики, информатики и информационных таможенных технологий

Л. Н. Бариков – кандидат технических наук, доцент – научный руководитель

МЕТОДЫ ТЕОРИИ ИГР ДЛЯ ОЦЕНКИ ОПТИМАЛЬНОГО РАЗМЕЩЕНИЯ ТОРГОВОЙ ТОЧКИ

Автоматизированные складские помещения

Наверное, каждый видел в Интернете видеоролики, посвященный таким системам, где группа роботов-транспортников лавирует между стеллажами, достает или убирает товары. Такие системы все больше вытесняют человеческий труд с крупных складов. Системы эти типовые, но, если склад не строился под нее, поставщик организует адаптацию с учетом архитектурных и логистических особенностей помещений.

Существует несколько основных методов оценки эффективности размещения торговых точек [1].

Метод финансового анализа состоит в оценке точки безубыточности предполагаемого объекта на базе текущих арендных ставок и экспертной оценки возможности достижения данного показателя. В той или иной форме используется при принятии решения для каждой локации.

Метод контрольного чек-листа используется для предварительного отбора потенциальных площадок под открытие торгового объекта. Представляет собой набор процедур по оценке местоположения локации относительно заданных параметров, таких как: площадь объекта, удобство подъездных путей, близость конкурентов, наличие витрин и места под вывеску и т. д. По сути, это обязательные требования к локации.

Аналоговый подход заключается в том, чтобы рассчитать возможный объем продаж нового объекта путем сравнения (аналогии) с другим объектом той же торговой сети, имеющим схожие характеристики.

Метод с использованием гравитационной модели Хаффа основан на гипотезе, что покупатель выбирает магазин для совершения покупки по двум параметрам: площадь магазина (чем она больше, тем лучше) и расстояние до торговой точки (чем меньше, тем лучше). При этом зависимость от данных параметров повторяет закон гравитации Ньютона. Метод востребован при открытии гипермаркетов и торговых центров, но не подходит для малых объектов.

Математические модели и регрессионный анализ строятся на определении зависимой переменной, такой как товарооборот, и ее соотношении с набором данных, описывающих расположение других точек. Этот метод статистически более точен, чем методы сравнения «вручную» по контрольным чек-листам и аналогам.

От теории к практике

В теории считается, что применение сразу нескольких методов повышает эффективность прогноза, когда полученные значения сравниваются между собой. На практике действительно используются несколько подходов в рамках одного проекта. Например, посещаемость считается по гравитационной модели, а средний чек по методу аналогов.

Для минимизации рисков при расположении торговой точки необходимо использовать неустойчивость состояний рыночной среды. Такое состояние связано с рисками – экономическими, транспортными, финансовыми. Указанные риски влияют на конечный результат любого экономического процесса, на функционирование любого предприятия. Учет влияния рисков меняет результат, который может быть получен в результате решения оптимизационной задачи без учета неопределенности ситуации. С другой стороны, возможно выделить одну или более стратегий, которые дадут результат, близкий к оптимальному, полученному без учета влияния среды.

Процесс выбора расположения торговой точки состоит из двух этапов [2]:

– оценка макрорасположения (краткое сведение о состоянии экономики и демографической ситуации в регионе);

– оценка микрорасположения (географическое положение участка, его физические характеристики, инфраструктура района, важным показателем уровня развития инфраструктуры района является доступность объекта для потенциальных покупателей).

При оценке микрорасположения важны следующие факторы [1, 2]:

- объем и структура пассажиропотоков и транспортных средств (x_1);
- анализ транспортной доступности участка (x_2);
- анализ временной доступности участка (x_3);
- улично-дорожная сеть – оценивается, насколько удобны подъезды к торговому центру (x_4);
- возможность размещения парковки (x_5);
- визуальная доступность участка (x_6);
- этажность, а также плотность окружающей застройки (x_7).

Решение разобьем на два этапа. Сначала построим классическую задачу математического программирования (линейного программирования), а потом получим решение в условиях неопределенности [3].

Используем метод экспертных оценок, оценим вклад каждого фактора в общую привлекательность выбранной локации, для этого введем коэффициенты: c_{ij} – коэффициенты платежной матрицы, характеризующие экономическую эффективность при состоянии среды Π_i при стратегии поведения Y_j . Введем состояния среды Π_i и стратегии поведения X_i (например, их будет 3) (табл. 1).

Таблица 1

Метод экспертных оценок

Стратегии поведения	Π_1	Π_2	Π_3	Π_4	Π_5	Π_6	Π_7
Y_1	C_{11}	C_{21}	C_{31}	C_{41}	C_{51}	C_{61}	C_{71}
Y_2	C_{12}	C_{22}	C_{32}	C_{42}	C_{52}	C_{62}	C_{72}
Y_3	C_{13}	C_{23}	C_{33}	C_{43}	C_{53}	C_{63}	C_{73}

Коэффициенты платежной матрицы фактически являются реализациями состояния городской среды, которая варьируется в определенных границах. Для построенной модели антагонистической игры построим матрицу риска [4, 5]

$$R = \begin{pmatrix} r_{11}r_{12} \dots r_{1n} \\ r_{21}r_{22} \dots r_{2n} \\ \dots \dots \dots \\ r_{m1}r_{m2} \dots r_{mn} \end{pmatrix},$$

элементы которой определяются по формуле

$$r_{ij} = \max_i b_{ij} - b_{ij}.$$

Матрица рисков позволяет количественно оценить эффективность различных стратегий Y_j в зависимости от состояния потребительской среды Π_j .

Используем теоретико-вероятностный подход. Введем в рассмотрение риски [4, 5]:

- R_1 – риск конкуренции;
- R_2 – риск колебания объема покупателей;
- R_3 – риски транспортной доступности;
- R_4 – экономические (инфляция и другие) риски;
- R_5 – законодательно-юридические риски.

Методом экспертных оценок оценим вероятности реализации этих событий.

Предположим, что у проекта возник только один из рисков. Это соответствует событиям: R_1, R_2, R_3, R_4, R_5 ; появление двух рисков влечет события $R_i * R_j$ при $i \neq j$; появление трех рисков $R_i * R_j * R_k$ при

$i \neq j \neq k$; появление четырех рисков $R_i * R_j * R_k * R_m$ при $i \neq j \neq k \neq m$; появление всех пяти видов риска $R_1 * R_2 * R_3 * R_4 * R_5$; ситуация, когда риски не возникают, \bar{R} .

Вычислим вероятности этих событий:

$$P(R_j * R_i) = p_i * p_j,$$

$$P(R_j * R_i * R_k) = p_i * p_j * p_k,$$

$$P(R_j * R_i * R_k * R_m) = p_i * p_j * p_k * p_m,$$

$$P(R_1 * R_2 * R_3 * R_4 * R_5) = p_1 * p_2 * p_3 * p_4 * p_5.$$

Вероятность отсутствия риска: $P(\bar{R}) = \bar{p}$.

Исходные события должны составлять полную группу событий, поэтому вероятности должны быть перенормированы:

$$\begin{aligned} \bar{p} + \sum_{i=1}^5 p_i + \sum_{i \neq j}^5 p_i * p_j + \sum_{i \neq j, k \neq j}^5 p_i * p_j * p_k + \\ + \sum_{i \neq j, k \neq j, m \neq 1}^5 p_i * p_j * p_k * p_m + p_1 * p_2 * p_3 * p_4 * p_5 = 1. \end{aligned}$$

Набор вероятностей определяется методом экспертных оценок при переходе от одной стратегии к другой.

Далее определим финансовые потери S_1, S_2, S_3, S_4, S_5 при возникновении каждого из рисков R_1, R_2, R_3, R_4, R_5 , которые также могут быть определены методом экспертных оценок.

Вычислим суммарные усредненные потери:

$$\begin{aligned} \bar{S}(X) = \sum_{i=1}^5 p_i * S_i + \sum_{i \neq j}^5 (S_i + S_j) p_i * p_j + \\ + \sum_{i \neq j, k \neq j}^5 (S_i + S_j + S_k) p_i * p_j * p_k + \\ + \sum_{i \neq j, k \neq j, m \neq 1}^5 (S_i + S_j + S_k + S_m) p_i * p_j * p_k * p_m + \\ + (S_1, S_2, S_3, S_4, S_5) p_1 * p_2 * p_3 * p_4 * p_5. \end{aligned}$$

Эти потери должны быть соотнесены с прибылью от реализации проекта, полученной при решении задачи линейного программирования, не учитывая риски. Тогда окончательный результат, определяющий прибыль задается целевой функцией:

$$Z(X) = F(X) - \bar{S}(X) \rightarrow \max.$$

Библиографический список

1. Угаров А. С. Методы выбора местоположения торговой точки // Маркетинг в России и за рубежом. 2005. №6. URL: <https://www.ipponu.ru/print/004142/> (дата обращения: 07.04.2021).

2. Определение оптимального места расположения торгового центра // Национальная единая риэлторская сеть: сайт. 2009. URL: <https://news.ners.ru/opredelenie-optimalnogo-mesta-raspolozheniya-torgovogo-tsentra.html> (дата обращения: 07.04.2021).

3. Методы прикладной математики в прогнозировании и управлении сложными процессами / под ред. С. И. Никитина. СПб.: Изд-во СПбГЭУ, 2014. 107 с.

4. Никитин С. И. Классические методы оптимизации социально-экономических процессов сферы сервиса: монография. СПб: Изд-во СПбГУСЭ, 2010. 151 с.

5. Вентцель Е. С. Исследование операций: задачи, принципы, методология: учеб. пособие. 5-е изд., стер. М.: КНОРУС, 2013. 192 с.

УДК 519.17, 656.05

А. Д. Шуйский

студент кафедры прикладной математики, информатики и информационных таможенных технологий

Л. Н. Бариков – кандидат технических наук, доцент – научный руководитель**ОПИСАНИЕ МАРШРУТОВ НА СКЛАДЕ С ПОМОЩЬЮ ГРАФОВОЙ МОДЕЛИ**

Наверное, каждый видел в Интернете видеоролики, посвященные автоматизированным складским помещениям, где группа роботов-транспортников лавирует между стеллажами, достает или убирает товары. Такие системы все больше вытесняют человеческий труд с крупных складов. Системы эти типовые, но, если склад не строился под такую систему, поставщик организует адаптацию с учетом архитектурных и логистических особенностей помещений.

На самом деле склады, где работают грузчики с гидравлическими тележками и различными погрузчиками, никуда не делись и сегодня их подавляющее большинство. Логистика на таких складах часто оставляет желать лучшего, соответственно, возникает угроза безопасности рабочих и целостности товаров в результате пересечений маршрутов. Конечно, не стоит исключать и человеческий фактор, но оптимизировать основные аспекты вполне возможно.

Рассмотрим, например, комплекс складских помещений на большой территории, куда подъезжает грузовой транспорт, где работают как погрузчики, так и гидравлические тележки, где рабочие могут перемещаться по территории в том числе без спецсредств. Как в таком случае можно было бы описать сеть потенциальных маршрутов? С помощью графов. В задачах сетевой маршрутизации, на обход сети улиц, о коммивояжере всегда используется одна и та же модель – представление конечных и промежуточных точек в виде вершин графа, а маршрутов между ними – в виде ребер. Подобные алгоритмы и методы уже существуют, в том числе в решениях отечественных производителей. Предлагаемый подход – альтернатива существующему.

На первом этапе определяются места размещения вершин графа. При этом стоит учесть не только внутреннюю логистику отдельного помещения, но и способы перемещения между помещениями. На втором этапе определяются виды маршрутов (исходя из используемых спецсредств и способов перемещения рабочих). Здесь, скорее всего, возникнет ситуация, когда между несколькими вершинами происходят перемещения разных видов. Например, рабочий на погрузчике и рабочий-пешеход. Это значит, что между парой вершин у нас будет несколько ребер, т. е. граф будет мультиграфом. Математические модели и алгоритмы работы с такими графами детально показаны в [1].

С точки зрения оптимизации графовой модели сама структура графа не играет особой роли. В данном случае речь пойдет о многокритериальной задаче, где один из критериев – вид маршрута, т. е. ребра. Алгоритмически необходимо будет выбрать тип и проверять только те ребра, которые этому типу соответствуют [2]. К тому же речь в первую очередь идет не о динамической задаче, т. е. решать мы ее планируем только один раз для каждого частного случая.

На третьем этапе строится граф, где вершины – точки на схеме склада, а каждое ребро соответствует какому-либо способу транспортировки груза и содержит, например, информацию о длине маршрута. Теперь можно приступить к решению задачи. В первую очередь необходимо определить маршруты для самого опасного способа транспортировки. Здесь подразумевается опасность для рабочих, пользующихся другими техническими средствами или способами передвижения, в случае столкновения. Затем определяются маршруты для самого опасного способа из оставшихся и т. д. На каждой очередной итерации определения маршрутов уже занятые получают наименьший приоритет, что в идеальном случае позволяет максимально разделить виды транспортировки грузов с маршрутами. Алгоритм решения такой задачи не может быть универсальным и в каждом конкретном случае должен адаптироваться.

Например, если на складе предусмотрена условная центральная магистраль с четким разделением по способам транспортировки или перемещения (в качестве примера можно привести привычное всем разделение на пешеходную зону, велодорожку и автомобильную трассу), то логичным будет, если маршруты погрузчика и гидравлической тележки совпадут, а безопасность маршрута будет базироваться на уже существующем разделении.

Последний этап базируется на определении маршрутов, связывающих комплекс складских помещений воедино. Общий порядок действий будет базироваться на уже описанном ранее, но в качестве вершин графа будут использоваться точки входа в помещение. Здесь нужно будет в первую очередь учесть грузовой транспорт, маршрут которого вряд ли можно изменять. Аналогичные ситуации при решении других задач приводятся в [3].

Грузовой транспорт, а точнее периоды разгрузки и погрузки, вносит изменения в полученную схему маршрутов. В таких случаях возможно существование нескольких схем или, например, применение технологического окна для некоторых видов транспортировки [4]. Этот подход требует дополнительного вмешательства: кроме проектирования маршрутов, необходимо разработать систему оповещения работников о таком окне или временном изменении логистики.

Приведенный подход нельзя назвать полноценным алгоритмом, скорее, его фрагментами или концептом. При решении задачи по организации маршрутов в каждом частном случае возникает множество вспомогательных факторов, которые могут кардинально повлиять на конечный результат. К тому же задача оптимизации, которая была рассмотрена, не является единственной. Можно рассмотреть вопрос о размещении дополнительных вершин на графе, т. е. перепланировку склада. В дальнейшем, несмотря на утверждение о статичности метода и его неуниверсальности, планируется разработка унифицированного динамического подхода, который, при организации соответствующей инфраструктуры сможет упростить работу практически любого склада и снизить риски как для работников, так и для хранимых товаров.

Библиографический список

1. Фомичев В. М. Свойства путей в графах и мультиграфах // Приклад. дискрет. математика. 2010. № 1(7). С. 118–124.
2. Sorokin A., Kovalenko R., Yakovleva E. Data structures for storing multigraph // Recent Scientific Investigation: XVII Intern. Multidisciplinary Conf. Proceedings of the Conf. Shawnee, USA: Primedia E-launch LLC, 2021. P. 18–22.
3. Казаков А. Л., Петров М. Б., Маслов А. М. Многокритериальная оптимизация транспортной системы региона на основе ее гиперграфа // Экономика региона. 2014. № 4(40). С. 199–208.
4. О задаче назначения «технологического окна» на участках железнодорожной сети / Д. Н. Гайнанов, А. Н. Игнатов, А. В. Наумов, В. А. Рассказова // Автоматика и телемеханика. 2020. № 6. С. 3–16.

УДК 004

Е. Ю. Язецкий

студент кафедры прикладной математики, информатики и информационных таможенных технологий

А. А. Сорокин – старший преподаватель – научный руководитель

УСТРОЙСТВО ЧАТ-БОТОВ

Введение

Чат-боты сегодня стремительно развиваются. Они помогают организациям в различных отраслях консультировать пользователей на различные темы с помощью заранее заданного сценария. Так как большую популярность чат-боты приобрели в мессенджерах, далее будет идти речь именно об их устройстве в мессенджерах.

Чат-боты могут быть созданы на любом языке программирования, поддерживающем функционал обмена данными по необходимому для API-протоколу или с использованием специальных библиотек, предоставляемых разработчиками API, под которое разрабатывается бот. Все известные мессенджеры всегда предоставляют подробные руководства, как связать своего чат-бота с их платформой [1].

Виды ботов

В основном существует два вида ботов: первые работают по заданному сценарию, это значит, что они ограничены «свободой» действия и при любой непонятной (неправильной) команде, пришедшей от пользователя, попросят его ввести команду заново. Ботов, работающих по сценарию, проще создавать, но по указанной причине они смогут решить не все поставленные задачи. Таких ботов часто используют там, где процесс поддается алгоритмизации, например заказ такси, покупка билетов, поиск книги, просмотр расписания и т. д. Вторые не привязаны к конкретному сценарию и самообучаются при помощи алгоритмов машинного обучения. Данная категория ботов позволяет решать больше различных задач, но их создание гораздо сложнее, и не стоит забывать, что со временем знания, полученные от пользователей накапливаются, и чат-бот становится более эффективным.

Принцип работы

Цикл работы чат-бота по сценарию, и в принципе любого бота, описывается следующими этапами: получение запроса от пользователя; определение точки сценария; генерирование ответа для пользователя; сохранение новой точки сценария; отправка ответа пользователю. Алгоритмически данный процесс можно описать способом, представленным на рис. 1.

Получение запроса от пользователя

В данном этапе от сервера мессенджера приходят данные с запросом или при запросе у мессенджера новых данных. В данных чаще все приходят текстовые сообщения от пользователя, но также могут прийти аудиосообщения или файлы различных форматов.

Определение точки сценария

На данном этапе определяется точка сценария, в которой находится сейчас пользователь. Это необходимо для того, чтобы присланный запрос от пользователя рассматривался в контексте, к которому в процессе общения пользователь пришел. Например, пользователь заказывает билеты на поезд, чат-бот его спрашивает о точке отправления, точке прибытия, времени отправления, данных от пользователя и т. д., в таком случае бот после каждого ответа на поставленный вопрос переводит пользователя на другую точку в сценарии, например после вопроса «Откуда направляемся?» перейдет на точку «Ожидание точки отправления», после того, как придет ответ, бот задаст новый вопрос «Куда сегодня поедем?», и опять же перевод пользователя на другую точку в сценарии «Ожидание точки назначения» и т. д.



Рис. 1. Алгоритм работы бота

Генерация ответа

На основе точки в сценарии, полученной из предыдущего этапа, и присланного сообщения от пользователя, генерируется ответ пользователю. На примере того же сценария с покупкой билета на поезд, рассмотрим варианты, которые могут быть сгенерированы в качестве ответа. Пользователь после ответа на вопрос «Откуда направляемся», присылает город, в таком случае генерируется ответ «С какого вокзала отправляемся?», так как в сообщении находился только город и название вокзала не было указано.

Сохранение новой точки сценария

После того как сгенерировался ответ для пользователя, необходимо сохранить новую точку сценария, пример для данного этапа был описан в определении точки сценария. Также на данном этапе сохраняется дополнительный контекст для пользователя для дальнейшей работы бота. Например, сохраняются точка отправления пользователя, точка прибытия, время отправления, время прибытия, место и т. д.

Отправка ответа пользователю

На последнем этапе происходит отправка ранее сгенерированного ответа пользователю. Для выполнения этого действия бот отправляет запрос на сервер мессенджера. В запросе как раз находится ответ, предназначенный для пользователя.

Боты с простым сценарием

Помимо ранее описанного цикла работы бота, существует множество ботов, которые устроены еще проще. Они состоят только из трех этапов: получение запроса от пользователя; генерация ответа; отправка ответа пользователю. Пример алгоритма, по которому работают данные боты, продемонстрирован на рис. 2. В нем можно увидеть, как на основе заранее заданного сообщения (команды), генерируется ответ для пользователя.

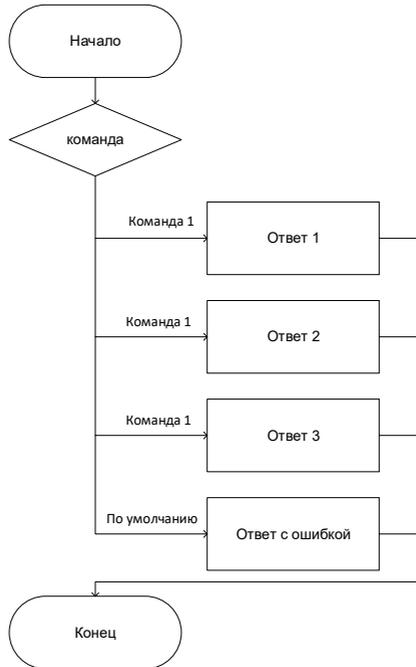


Рис. 2. Алгоритм генерации ответа

Выводы

Чат-боты работающие по сценарию, устроены не так сложно, чтобы внедрить их в большинство отраслей. При этом они помогут пользователям в решении повседневных задач – от получения расписания для занятий до управления различными системами из любой точки мира.

Библиографический список

1. Цыганова Е. В. Чат-боты // Студ. вестн. 2020. № 31-2(129). С. 34–36.

СОДЕРЖАНИЕ

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ	3
Проблемно-ориентированные вычислительные комплексы	3
<i>Асоскова В. В., Крот А. И.</i> АНАЛИЗ АЛГОРИТМОВ ОБРАБОТКИ ИЗОБРАЖЕНИЙ РАЗЛИЧНОЙ ФИЗИЧЕСКОЙ ПРИРОДЫ.....	3
<i>Афанасьев В. А., Пенкин Д. В.</i> ПРОЕКТИРОВАНИЕ БЕСПИЛОТНОЙ МАШИНЫ ПАТРУЛИРОВАНИЯ ЛЕСНЫХ НАСАЖДЕНИЙ.....	8
<i>Веселов А. Р., Золотарев С. В.</i> ВЫБОР ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ИНВЕНТАРИЗАЦИИ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА ПРЕДПРИЯТИЯ.....	10
<i>Вихров В. В.</i> АВТОМАТИЗАЦИЯ СБОРА ИНФОРМАЦИИ О СТОИМОСТИ НЕДВИЖИМОСТИ ИЗ ОТКРЫТЫХ ИСТОЧНИКОВ ДАННЫХ.....	12
<i>Волкова А. С., Раскопина А. С.</i> СРАВНЕНИЕ АЛГОРИТМОВ ПО СКОРОСТИ ФОРМИРОВАНИЯ ВЫБОРКИ СЛУЧАЙНЫХ ЧИСЕЛ	14
<i>Горбунова А. М.</i> НЕЗАВИСИМАЯ МЕХАНИЧЕСКАЯ СИСТЕМА БЕЗОПАСНОСТИ АЭС.....	18
<i>Горбунова А. М.</i> ОПТИМИЗАЦИЯ ЭНЕРГОПОТРЕБЛЕНИЯ ЛИФТОВ И ЭСКАЛАТОРОВ. СИСТЕМА БЕЗОПАСНОСТИ	22
<i>Гордеев И. Р.</i> ИССЛЕДОВАНИЕ И АНАЛИЗ КОРРЕЛЯЦИОННЫХ ХАРАКТЕРИСТИК КОДОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ МАКСИМАЛЬНОЙ ДЛИНЫ.....	26
<i>Душеба А. В.</i> ПОСТРОЕНИЕ ТРАЕКТОРИИ ДВИЖЕНИЯ НЕБЕСНЫХ ТЕЛ	30
<i>Душеба А. В.</i> ПРОЕКТИРОВАНИЕ МОДЕЛЕЙ ТРЕХМЕРНЫХ ОБЪЕКТОВ С ПОМОЩЬЮ АЛГОРИТМА RAY MARCHING.....	32
<i>Класс И. Ю.</i> РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ МОНИТОРИНГА ТЕХНИЧЕСКОГО СОСТОЯНИЯ АВТОТРАНСПОРТНОГО СРЕДСТВА	35
<i>Матренинский С. А.</i> ЭЛЕКТРОННЫЕ УСТРОЙСТВА КАК ЧАСТЬ ЭКИПИРОВКИ ДЛЯ ВОЕННО-ТАКТИЧЕСКОЙ ИГРЫ.....	37
<i>Мирошниченко Н. И.</i> МЕТОДЫ МОДЕРНИЗАЦИИ ЭЛЕКТРИЧЕСКИХ СЕТЕЙ. ВНЕДРЕНИЕ SMART GRID	41
<i>Михайлов В. К.</i> АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ СРЕДСТВ ВИЗУАЛИЗАЦИИ ТРЕХМЕРНОЙ ГРАФИКИ В WEB.....	43
<i>Раскопина А. С.</i> СИСТЕМА МОНИТОРИНГА ВЫСОКОВОЛЬТНЫХ СЕТЕЙ ДЛЯ ПОВЫШЕНИЯ ИХ ПРОПУСКНОЙ СПОСОБНОСТИ	50
<i>Рогозин П. А.</i> АНАЛИЗ ПОДХОДОВ К РАЗРАБОТКЕ УСТРОЙСТВ ДЛЯ ОЦЕНИВАНИЯ ФУНКЦИОНАЛЬНОГО СОСТОЯНИЯ ЧЕЛОВЕКА-ОПЕРАТОРА	52
<i>Русанов М. В.</i> РЕАЛИЗАЦИЯ ГРАФИЧЕСКОГО МОДУЛЯ TURTLE С ПОМОЩЬЮ OPENGL	56
<i>Сергеева Е. И.</i> ВИРТУАЛЬНЫЙ МУЗЕЙ КАК ИНФОРМАЦИОННАЯ СИСТЕМА.....	60
<i>Синкин М. Д.</i> ДИАГНОСТИКА НЕИСПРАВНОСТЕЙ СЕТЕЙ ЭЛЕКТРОСНАБЖЕНИЯ.....	64
<i>Судьин М. И., Кадьивкин В. А.</i> ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ПЛИС – QUARTUS.....	66
<i>Фаталиева Е. О.</i> УСТРОЙСТВО ДЛЯ ОЦЕНКИ ОТНОСИТЕЛЬНОЙ ВЛАЖНОСТИ КОЖНОГО ПОКРОВА	68
<i>Цыганова М. А.</i> СОВРЕМЕННЫЕ СИСТЕМЫ РАСПОЗНАВАНИЯ РЕЧИ.....	71
<i>Чекменева А. А.</i> УСТРОЙСТВО ДЛЯ ОЦЕНКИ КИСЛОРОДА И УГЛЕКИСЛОГО ГАЗА В ЗАМКНУТОМ ОБЪЕМЕ С ЭКСПЕРИМЕНТАЛЬНЫМ ЖИВОТНЫМ.....	73
<i>Чекменева А. А.</i> УСТРОЙСТВО ДЛЯ ОЦЕНКИ ВОДОРОДА В ВЫДЫХАЕМОМ ВОЗДУХЕ.....	77

Компьютерные технологии и программная инженерия	81
<i>Блудов В. А.</i> ПОСТРОЕНИЕ ПУТИ РОБОТА В СЛОЖНОЙ СРЕДЕ НА ОСНОВЕ ГИБРИДНОГО МУРАВЬИНОГО АЛГОРИТМА	81
<i>Гришин С. А.</i> СОЗДАНИЕ ОТКАЗОУСТОЙЧИВОЙ ИНФРАСТРУКТУРЫ НА ОСНОВЕ KUBERNETES ДЛЯ СБОРА ИНФОРМАЦИИ С СЕТЕВЫХ УСТРОЙСТВ	86
<i>Килин Д. В.</i> СИСТЕМА ПОМОЩИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ АНАЛИЗЕ ДВИЖЕНИЙ ФОНДОВОГО РЫНКА	90
<i>Лёвин Г. С.</i> МОДЕЛИРОВАНИЕ НЕЛИНЕЙНОГО ВЕКТОРНОГО УПРАВЛЕНИЯ ПРОДОЛЬНОМ ДВИЖЕНИЕМ САМОЛЕТОМ-АМФИБИЕЙ	95
<i>Погодина И. А.</i> РЕШЕНИЕ ЗАДАЧИ CVRP С ПЛАВАЮЩИМИ ВРЕМЕННЫМИ ОКНАМИ	101
<i>Самусенко Д. В.</i> ОПТИМИЗАЦИЯ ДИСТРИБУТИВНОЙ СЕТИ ПОСТАВОК С ОДНИМ ДИСТРИБЬЮТЕРОМ И ДВУМЯ ПОСТАВЩИКАМИ	105
<i>Скрыпник А. Н.</i> РЕШЕНИЕ ЗАДАЧИ CVRP С ДВУМЯ ДЕПО И N КЛИЕНТАМИ	110
<i>Ткаченко А. С.</i> ИССЛЕДОВАНИЕ МОДЕЛИ «ХИЩНИК – ЖЕРТВА С ПИТАНИЕМ И УПРАВЛЕНИЕМ БЕЗ РАСШИРЕНИЯ ФАЗОВОГО ПРОСТРАНСТВА»	115
<i>Хмельва Ю. А.</i> ПОДХОДЫ К РАЗРАБОТКЕ СИСТЕМЫ ВЫДЕЛЕНИЯ, КЛАССИФИКАЦИИ И РАСПОЗНАВАНИЯ ОБЪЕКТОВ ВИДЕОРЯДА	120
<i>Щеголева А. А.</i> АПРОБАЦИЯ МОДЕЛИ «ХИЩНИК – ЖЕРТВА С ПИТАНИЕМ» НА ДАННЫХ ГИДРОБИОЛОГИЧЕСКОГО МОНИТОРИНГА БАЛТИЙСКОГО МОРЯ	125
Вычислительные системы и сети	130
<i>Александров И. А.</i> РАЗРАБОТКА КООПЕРАТИВНОЙ ИГРЫ В СТИЛЕ ШУТЕР НА ИГРОВОМ ДВИЖКЕ UNREAL ENGINE 4	130
<i>Антюпова А. А., Салова Н. В.</i> ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ ДЛЯ ОБУЧЕНИЯ ГИБКИМ НАВЫКАМ	135
<i>Антонов А. А., Дук Г. В.</i> ПРИМЕНЕНИЕ МУЛЬТИАГЕНТНОГО ПОДХОДА ПРИ РАЗРАБОТКЕ ИМИТАЦИОННЫХ МОДЕЛЕЙ	139
<i>Афанасьева В. И.</i> КЛАССИФИКАЦИЯ НАБЛЮДАЕМЫХ ЗОН НА СФОРМИРОВАННЫХ СНИМКАХ ПОТОКА ВИДЕОКАДРОВ В БОРТОВЫХ СИСТЕМАХ МОНИТОРИНГА ЗЕМНОЙ ПОВЕРХНОСТИ	142
<i>Гайков Б. С.</i> МОДЕЛИРОВАНИЕ ТРЕХМЕРНЫХ ОБЪЕКТОВ ДЛЯ ИНТЕРАКТИВНЫХ ПРИЛОЖЕНИЙ. ТЕХНОЛОГИЯ МОДЕЛИРОВАНИЯ ТВЕРДЫХ ПОВЕРХНОСТЕЙ	147
<i>Карамышева В. А.</i> ПОСТРОЕНИЕ ВИРТУАЛЬНОГО 3D-ТУРА ПО УДАЛЕННЫМ ПОДРАЗДЕЛЕНИЯМ ПРЕДПРИЯТИЯ	152
<i>Лемешевская Л. В.</i> ТРЕХМЕРНОЕ МОДЕЛИРОВАНИЕ КАК ОДИН ИЗ КОМПОНЕНТОВ ПОГРУЖЕНИЯ В ТЕМАТИКУ КВЕСТА	159
<i>Мустафин Т. Р.</i> ОБСЛУЖИВАНИЕ АВТОМОБИЛЕЙ НА ОСНОВЕ ЖЕСТОВОГО ИНТЕРФЕЙСА ШЛЕМА ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ	162
<i>Нагорных М. Э., Быков А. Н.</i> ОСОБЕННОСТИ И АНАЛИЗ ГРАФОВЫХ БАЗ ДАННЫХ	168
ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ЗАЩИТА ИНФОРМАЦИИ	171
Безопасность информационных систем	171
<i>Альмухамедов А. И.</i> ПЕРСПЕКТИВНЫЕ МЕТОДЫ БИОМЕТРИЧЕСКОЙ ИДЕНТИФИКАЦИИ ЛИЧНОСТИ	171
<i>Бухвалов П. С.</i> ИССЛЕДОВАНИЕ ВОЗМОЖНОСТИ ИСПОЛЬЗОВАНИЯ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ПРИ ПРОВЕДЕНИИ СПЕЦИАЛЬНЫХ ИССЛЕДОВАНИЙ ТЕХНИЧЕСКИХ СРЕДСТВ	175

<i>Вересова А. М.</i> О СТАТИСТИЧЕСКИХ СВОЙСТВАХ МОДЕЛЕЙ ГИЛБЕРТА И ГИЛБЕРТА – ЭЛЛИОТА ДЛЯ КАНАЛОВ С ПАМЯТЬЮ.....	182
<i>Зыков Д. А.</i> МОДЕЛИРОВАНИЕ СИСТЕМ ХРАНЕНИЯ ДАННЫХ ДЛЯ ОЦЕНКИ ВЛИЯНИЯ ИЗБЫТОЧНОСТИ НА ОТКАЗОУСТОЙЧИВОСТЬ.....	187
<i>Коржук В. С.</i> ИССЛЕДОВАНИЕ ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ МОДЕЛЕЙ НЕЙРОННЫХ СЕТЕЙ ДЛЯ СПЕЦИАЛЬНЫХ ПРОВЕРЕК ТЕХНИЧЕСКИХ СРЕДСТВ.....	190
<i>Лебедев В. О.</i> РАЗРАБОТКА ПРИЛОЖЕНИЯ ПОД ОПЕРАЦИОННУЮ СИСТЕМУ ANDROID ДЛЯ ИЗМЕРЕНИЯ СКОРОСТИ РЕАКЦИИ ЧЕЛОВЕКА	200
<i>Марковский В. С., Фоминых А. А.</i> ОРГАНИЗАЦИЯ 4G СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ СОТОВОЙ СВЯЗИ.....	205
<i>Мельников М. А.</i> АНАЛИЗ ВЛИЯНИЯ ФАКТОРОВ ОКРУЖЕНИЯ НА БИОМЕТРИЧЕСКУЮ АУТЕНТИФИКАЦИЮ ПО ЛИЦУ.....	210
<i>Налегач В. С.</i> АНАЛИЗ ЭФФЕКТИВНОСТИ СИСТЕМЫ ПРЕДОТВРАЩЕНИЯ УТЕЧЕК ДАННЫХ В КОРПОРАТИВНОЙ СЕТИ НА ОСНОВЕ DLP-СИСТЕМЫ INFOWATCH TRAFFIC MONITOR	216
<i>Недошивин П. П.</i> СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ ПОСТКВАНТОВЫХ КРИПТОСИСТЕМ	220
<i>Перфильев Е. В.</i> КЛАССИФИКАЦИЯ УГРОЗ БЕЗОПАСНОСТИ ИНФОРМАЦИИ ПО ТЕХНИЧЕСКИМ КАНАЛАМ УТЕЧКИ ИНФОРМАЦИИ	222
<i>Перфильев Е. В.</i> КЛАССИФИКАЦИЯ ИСТОЧНИКОВ УГРОЗ БЕЗОПАСНОСТИ ИНФОРМАЦИИ	226
<i>Савинов М. С.</i> УМЕНЬШЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ПОИСКА ОБЪЕКТОВ НА ИЗОБРАЖЕНИЯХ	230
<i>Савинова М. Д.</i> РАСПРЕДЕЛЕННОЕ КОДИРОВАНИЕ ИЗОБРАЖЕНИЙ С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ	234
<i>Шром Б. С.</i> АРХИТЕКТУРА БЕЗОПАСНОСТИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ОПЕРАЦИОННОЙ СИСТЕМЫ ANDROID	238
Инфокоммуникационные системы	242
<i>Васильев Д. А.</i> ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ УДАЛЕННОГО БЕСПРОВОДНОГО ДОСТУПА ПОЛЬЗОВАТЕЛЯ К РАСПРЕДЕЛЕННЫМ ВЫЧИСЛИТЕЛЬНЫМ РЕСУРСАМ.....	242
<i>Глушеникова А. Ю.</i> ИССЛЕДОВАНИЕ АЛГОРИТМОВ МНОЖЕСТВЕННОГО ДОСТУПА С ИСПОЛЬЗОВАНИЕМ ОРТОГОНАЛЬНЫХ ПРЕАМБУЛ В СИСТЕМАХ ИНТЕРНЕТА ВЕЩЕЙ.....	249
<i>Осиков А. Р.</i> ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ДЛЯ ИССЛЕДОВАНИЯ СПОСОБОВ ПЕРЕДАЧИ В НЕЛИЦЕНЗИРОВАННЫХ ДИАПАЗОНАХ ЧАСТОТ.....	255
<i>Рачугин Р. О.</i> ИССЛЕДОВАНИЕ МНОГОКАНАЛЬНОГО АЛГОРИТМА АЛОХА С ФАЗОЙ ИССЛЕДОВАНИЯ	262
<i>Фиклистова Ю. О.</i> ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ КАНАЛА ПЕРЕДАЧИ СЛУЖЕБНОЙ ИНФОРМАЦИИ В СЕТЯХ ПЯТОГО ПОКОЛЕНИЯ.....	270
<i>Царевский А. В.</i> СИСТЕМА ИДЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЕЙ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА НА ОСНОВЕ БЕСПРОВОДНЫХ ТЕХНОЛОГИЙ МАЛОГО РАДИУСА ДЕЙСТВИЯ.....	276
Информационно-сетевые технологии	283
<i>Агеев М. П., Гуков С. Ю.</i> АНАЛИЗ ЭФФЕКТИВНОСТИ НОВЫХ ГРАФИЧЕСКИХ ФОРМАТОВ КОДИРОВАНИЯ	283

<i>Гуков С. Ю.</i> ПРОБЛЕМА ФОРМИРОВАНИЯ КАРТЫ ГЛУБИНЫ С ИСПОЛЬЗОВАНИЕМ ЗАДЕРЖКИ СВЕТОВОГО СИГНАЛА В УСЛОВИЯХ ВЗАИМНЫХ ПОМЕХ ПРИ МУЛЬТИКАМЕРНОЙ СЪЕМКЕ	289
<i>Ефремов А. Д.</i> ОПТИМИЗАЦИЯ ТЕРМОЭЛЕКТРИЧЕСКОЙ СИСТЕМЫ ОХЛАЖДЕНИЯ ДЛЯ ПРИМЕНЕНИЯ В ПРОЦЕССОРНОМ КУЛЕРЕ	295
<i>Шевяков Д. О.</i> ОБЛАСТЬ ПРИМЕНЕНИЯ И ПРИНЦИП РАБОТЫ МОДЕЛЕЙ WORD2VEC	302
ИВАНГОРОДСКИЙ ФИЛИАЛ	304
Прикладная математика, информатика и информационные таможенные технологии	304
<i>Белоусов Д. О.</i> СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРИМЕНЕНИЯ ДОМАШНИХ ОБЛАЧНЫХ ХРАНИЛИЩ В КОРПОРАТИВНЫХ ЦЕЛЯХ	304
<i>Викторова А. П.</i> ЗАВИСИМОСТЬ ПОГРЕШНОСТИ ИЗМЕРЕНИЯ РАССТОЯНИЯ ИК-ДАЛЬНОМЕРОМ ОТ СВОЙСТВ ПОВЕРХНОСТИ	307
<i>Долгов Е. Н.</i> ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ДЛЯ МОДЕЛИРОВАНИЯ	310
<i>Иванов А. Р.</i> МЕХАНИКИ ВЗАИМОДЕЙСТВИЯ ИГРОКОВ В КООПЕРАТИВНЫХ ИГРАХ	316
<i>Михалев Д. О.</i> ВЕБ-СЕРВИСЫ ДЛЯ ПОЛУЧЕНИЯ ЕЖЕДНЕВНЫХ ЭКОНОМИЧЕСКИХ ДАННЫХ ПО ЭКОНОМИЧЕСКИМ ПОКАЗАТЕЛЯМ	320
<i>Никашкин Э. О.</i> ЭКОНОМЕТРИЧЕСКИЕ МЕТОДЫ, ИСПОЛЬЗУЕМЫЕ ДЛЯ ОЦЕНКИ ЭФФЕКТИВНОСТИ В СФЕРЕ ЖКХ	323
<i>Овчаренко А. В.</i> ДИНАМИЧЕСКОЕ РАНЖИРОВАНИЕ ЗАДАЧ ДЛЯ СКЛАДСКИХ РАБОЧИХ	326
<i>Одегов А. Ф.</i> ОСОБЕННОСТИ РАЗРАБОТКИ КРОССПЛАТФОРМЕННЫХ ПРИЛОЖЕНИЙ	329
<i>Смирнов М. С.</i> ПРОЦЕСС ПОКАДРОВОГО РЕНДЕРИНГА С ПРИМЕНЕНИЕМ ГРАФИЧЕСКИХ API	333
<i>Столярчук А. С.</i> СРАВНЕНИЕ МЕТОДОВ СОЗДАНИЯ ПРОВОДОВ В UNITY	336
<i>Шайхутдинов Р. Р.</i> РАЗРАБОТКА АУДИОУСТРОЙСТВА ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ НА ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ МИКРОКОНТРОЛЛЕРАХ КОМПАНИИ STMICROELECTRONICS	340
<i>Шаповалова Д. П.</i> РАЗРАБОТКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ ВЗАИМОДЕЙСТВИЯ ФИЛИАЛА И ГОЛОВНОГО ВУЗА	345
<i>Шубин К. В.</i> МЕТОДЫ ТЕОРИИ ИГР ДЛЯ ОЦЕНКИ ОПТИМАЛЬНОГО РАЗМЕЩЕНИЯ ТОРГОВОЙ ТОЧКИ	349
<i>Шуйский А. Д.</i> ОПИСАНИЕ МАРШРУТОВ НА СКЛАДЕ С ПОМОЩЬЮ ГРАФОВОЙ МОДЕЛИ	353
<i>Язецкий Е. Ю.</i> УСТРОЙСТВО ЧАТ-БОТОВ	355

Научное издание

СЕМЬДЕСЯТ ЧЕТВЕРТАЯ
МЕЖДУНАРОДНАЯ СТУДЕНЧЕСКАЯ
НАУЧНАЯ КОНФЕРЕНЦИЯ ГУАП
(19–23 апреля 2021 г.)

Часть 2

ТЕХНИЧЕСКИЕ НАУКИ

Сборник докладов

ISBN: 978-5-8088-1611-4



9 785808 816114

Ответственный за выпуск:
А. В. Бобович

Редакторская подготовка *С. В. Денисовой*
Компьютерная верстка *А. Н. Колешко*

Подписано к печати 01.09.20. Формат 60x84 1/16.
Усл. печ. л. 21,0. Уч.-изд. л. 48,2. Тираж 150 экз. Заказ № 337.

Редакционно-издательский центр ГУАП
190000, Санкт-Петербург, ул. Б. Морская, д. 67