

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

СЕМЬДЕСЯТ ТРЕТЬЯ
МЕЖДУНАРОДНАЯ СТУДЕНЧЕСКАЯ
НАУЧНАЯ КОНФЕРЕНЦИЯ ГУАП
(13–17 апреля 2020 г.)

Часть 3

ТЕХНИЧЕСКИЕ НАУКИ

Сборник докладов



УДК 001
ББК 72
С30

С30 Семьдесят третья Международная студенческая научная конференция ГУАП (13–17 апреля 2020 г.): сб. докл.: в 5 ч. Ч. 3: Технические науки. – СПб.: ГУАП, 2020. – 326 с.
ISBN 978-5-8088-1507-0 (Ч. 3)
ISBN 978-5-8088-1504-9

Представлены доклады Семьдесят третьей Международной студенческой научной конференции ГУАП, прошедшей в апреле 2020 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
К. В. Лосев, доктор экономических наук, доцент
Н. Н. Майоров, кандидат технических наук, доцент
М. О. Самоловов, директор РИЦ
Е. Г. Семенова, доктор технических наук, профессор
М. Б. Сергеев, доктор технических наук, профессор
А. М. Тюрликов, доктор технических наук, профессор
Н. А. Чернова, кандидат экономических наук

Ответственный секретарь редакционной коллегии *Л. И. Корнеева*

ISBN 978-5-8088-1507-0 (Ч. 3)
ISBN 978-5-8088-1504-9

© Санкт-Петербургский государственный
университет аэрокосмического
приборостроения, 2020

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ

Проблемно-ориентированные вычислительные комплексы

УДК 004.056.55

Р. З. Алирзаев

магистрант кафедры проблемно-ориентированных вычислительных комплексов

А. А. Шевяков

магистрант кафедры проблемно-ориентированных вычислительных комплексов

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

АЛГОРИТМЫ ШИФРОВАНИЯ ХРАНИМЫХ ДАННЫХ В MYSQL

По данным рейтинга компании DB-engines (<https://db-engines.com>), наиболее востребованной среди систем управления баз данных (СУБД) на данный момент является MySQL, которая занимает 39 % мирового рынка программных продуктов баз данных и 59 % среди СУБД в России [1, 2]. Такие высокие результаты стали возможным благодаря тому, что эта СУБД наиболее часто используется при реализации веб-приложений. В настоящее время грамотные интернет пользователи могут самостоятельно реализовать веб-приложение с использованием фреймворков, CMS или облачных сервисов, в которых для хранения данных автоматически выбирается база данных (БД) MySQL. В БД может храниться персонализированные данные пользователя или другая информация, доступ к которой может иметь ограниченный круг людей. Для решения проблемы защищенности данных от кибератак и других злоумышленных действий в СУБД MySQL предусмотрены следующие механизмы шифрования данных [3].

Функции криптографии: AES_ENCRYPT(str, key) и AES_DECRYPT(crypt_str, key); DES_ENCRYPT(str, key) и DES_DECRYPT(crypt_str, key); ENCODE(str, pass_str) и DECODE(crypt_str, pass_str).

Хэш функции: MD5(string), SHA2(string), SHA1(string).

Для шифрования данных в БД чаще используют обратимые алгоритмы шифрования, например DES (Data Encryption Standard) или AES (Advanced Encryption Standard). Как видно из вышеприведенного описания в шифровании принимают участие две функции. Первая зашифровывает данные с использованием специального ключа, вторая расшифровывает полученный набор данных ключом, переданным для дешифрации. Для различных языков существуют готовые наборы функций шифрования. Для шифрования в среде MySQL можно использовать язык программирования PHP PDO. Рассмотрим примеры такой программной реализации алгоритмов AES и DES [4].

1. Алгоритм DES.

1.1. Шифрация строки данных с использование метода Triple-DES:

```
$dbh = new PDO('mysql:dbname=db_name;host=localhost', 'логин', 'пароль');
```

```
$sth = $dbh->prepare("
```

```
    INSERT INTO
```

```
        'table'
```

```
    SET
```

```
        'id' = NULL,
```

```
        'text' = DES_ENCRYPT(:text, :key)
```

```
");
```

```
$sth->execute(array('text' => Данные для шифрования', 'key' => 'пароль'));
```

1.2. Дешифрация строки:

```
$dbh = new PDO('mysql:dbname=db_name;host=localhost', 'логин', 'пароль');
```

```

$sth = $dbh->prepare("
    SELECT
        *
        ,
        DES_DECRYPT(`text`, :key) AS `text`
    FROM
        `table`
    WHERE
        `id` = 1
");
$sth->execute(array('key' => 'пароль'));
$array = $sth->fetch(PDO::FETCH_ASSOC);
print_r($array['text']);

```

При ошибки дешифрации возвращает значение NULL.

2. Алгоритм AES

2.1. Шифрация строки данных

```

$dbh = new PDO('mysql:dbname=db_name;host=localhost', 'логин', 'пароль');
$sth = $dbh->prepare("
    INSERT INTO
        'table'
    SET
        'id' = NULL,
        'text' = AES_ENCRYPT(:text, :key)
");

```

```

$sth->execute(array('text' => Данные для шифрования ', 'key' => 'пароль'));

```

2.2. Дешифрация строки:

```

$dbh = new PDO('mysql:dbname=db_name;host=localhost', 'логин', 'пароль');
$sth = $dbh->prepare("
    SELECT
        *
        ,
        AES_DECRYPT('text', :key) AS `text`
    FROM
        'table'
    WHERE
        'id' = 1
");

```

```

$sth->execute(array('key' => 'пароль'));
$array = $sth->fetch(PDO::FETCH_ASSOC);
print_r($array['text']);

```

Алгоритм MD5 (Message-Digest algorithm) относят к алгоритмам необратимого шифрования, т.е. однозначное декодирование зашифрованного сообщения отсутствует [5]. Однако правильнее MD5 называть не алгоритмом, а хеш-функцией. Она предназначена для хранения паролей и для проверки правильности передачи данных для копирования. Рассмотрим подробнее эти процессы.

Защита пароля. Пользователь отправляет первый раз пароль при регистрации, например на веб-приложении. От пароля вычисляется хеш-функция MD5. И результат сохраняется в одной из таблиц БД. При всех последующих вводах пароля пользователя будет происходить сравнение вновь вычисляемого значения с результатом первичной обработки, хранящимся в БД. При их совпадении пользователь сможет авторизоваться.

Проверка подлинности данных при копировании. Перед отправкой у файла с данными вычисляют контрольную сумму с помощью функции MD5. Полученный результат передают вместе с файлом. На стороне получателя от принятого файла также вычисляют контрольную сумму. Если результаты вычислений совпадают, следовательно, файл доставлен без потерь.

Пример шифрования пароля алгоритмом MD5:

Пароль	ГУАП
Результат хеширования	d4604bc4e106dfc298f0e0a0a9ef6caf

Алгоритмы `DES_ENCRYPT(str, key)` и `DES_DECRYPT(encrypt_str, key)` используют в том случае, если конфигурация MySQL поддерживает криптографический протокол SSL (Secure Sockets Layer).

Более совершенными алгоритмами шифрования являются алгоритмы SHA2 (Secure Hash Algorithm Version 2) и SHA1 (Secure Hash Algorithm Version 1), которые также используют для аутентификации пользователя [6]. В этом случае на стороне пользователя должен храниться открытый ключ от сервера БД. На самом сервере хранится пароль пользователя. При запросе клиента на доступ к данным сервер посылает клиенту строку случайных 20 символов (*scramble*). Клиент открывает RSA-ключ (Rivest, Shamir и Adleman) сервера из заранее полученного файла. Затем с помощью логической операции XOR соединяет пароль с присланной строкой *scramble*, шифрует открытым ключом сервера и отправляет серверу. Сервер расшифровывает присланные данные от пользователя своим секретным ключом, повторяет логическую операцию XOR в обратную сторону, получает пароль в исходном открытом виде, считывает от него SHA256 и сравнивает с тем паролем, что хранится в базе данных.

На основании вышеизложенного видно, что криптографическая защита базы данных MySQL постоянно совершенствуется. Новые алгоритмы шифрования строятся на основе методов криптографии с открытым ключом. Основным недостатком таких алгоритмов является их медленное выполнение в аппаратной и программной реализациях, поэтому чаще всего их применяют не для передачи данных, а для шифрования паролей пользователей.

Библиографический список

1. Рейтинг баз данных компании DB-engines. URL: <https://db-engines.com> (дата обращения: 24.05.2020).
2. Доля рынка программных стеков. URL: <https://dzone.com/articles/software-stacks-market-share-second-quarter-of-201> (дата обращения: 24.05.2020)
3. Шифрование полей в MySQL. URL: <https://snipp.ru/mysql/mysql-encrypt> (дата обращения: 24.05.2020).
4. Шифрование и дешифрование внутри MySQL. URL: <https://www.delphiplus.org/mysql-optimizatsiya-proizvoditelnosti/shifrovanie-i-deshifirovanie-vnutri-mysql.html> (дата обращения: 24.05.2020)
5. Алгоритм шифрования MD5. URL: <http://www.nestor.minsk.by/> (дата обращения: 24.05.2020).
6. Эволюция протоколов аутентификации MySQL и MariaDB в лицах. URL: <https://habr.com/ru/post/323670/> (дата обращения: 24.05.2020).

УДК 004.7, 004.382.7

Л. Н. Балезин

студент кафедры проблемно-ориентированных вычислительных комплексов

В. В. Берёза

студент кафедры проблемно-ориентированных вычислительных комплексов

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

УЧЕБНЫЙ КОМПЛЕКС НА ОСНОВЕ СЕТИ МИКРОКОМПЬЮТЕРОВ RASPBERRY PI

Современные предметы, в которых изучаются локальные и глобальные вычислительные сети, требуют сетевое оснащение для формирования практических навыков и необходим компьютерный класс. В качестве бюджетной, компактной и маломощной альтернативы можно использовать микрокомпьютеры

Цель состоит в разработке учебного универсального комплекса сетевого взаимодействия на основе массовых микрокомпьютеров.

Для достижения поставленной цели необходимо выбрать рабочую станцию, сетевое устройство и периферию.

Выбор микрокомпьютера

В качестве базы для учебного комплекса были выбраны одноплатные компьютеры. Их список представлен в табл. 1, вместе с изученными характеристиками микрокомпьютеров.

Таблица 1

Сравнение микрокомпьютеров

Хар-ка	Наименование				
	ConnectCore 6UL SBC Pro	UP Board	Raspberry pi 3 model B	Orange Pi Lite 2	BeagleBone Black Rev C / Embest
Процес-сор	NXP i.MX6UL-2 (Cortex-A7) – 528 МГц	Intel® Atom™ x5-z8350 (2M Cache, 1.44, 1.92 ГГц) CPU с 64-бит, 4 ядра	4 × ARM Cortex-A53, 1,2 ГГц	4 × ARM Cortex-A53, 1,2 ГГц	ARM Cor-tex-A8, 1 ГГц
Flash	256 МБ NAND SLC, 4 ГБ eMMC	eMMC 16GB / 32GB/ 64GB			4 ГБ eMMC
RAM	256 МБ (DDR3)	1GB/ 2GB/ 4GB (DDR3)	1 ГБ LPDDR2 (900 МГц)	(SDRAM): 1Гб DDR3 (совместно с GPU)	512 МБ DDR3L
Интер-фейсы	Ethernet ,PCI Ex-press Mini, Con-sole, 3 x UART, I2C, SPI, GPIO, Dual CAN, Head-phone Jack, USB 2.0, Bluetooth 4.2, Wi-Fi 802.11a	Gb Ethernet, HDMI, I2S, MIPI-CSI, USB 2.0 USB3.0, DSI/ eDP	HDMI, 3,5 мм, 4 × USB 2.0, Ethernet, Camera Serial Inter-face (CSI), Display Serial Interface (DSI), Ethernet на 10/100 Мбит Wi-Fi 802.11 b g n и Bluetooth 4.1	Wi-Fi + BT AP6255, YUV422 CMOS , HDMI, USB 3.0, USB 2.0, GPIO, UART	Ethernet, USB, OTG, JTAG, HDMI, ADC, I2C, SPI, PWM , LCD.
Память внеш-няя	microSD	-	microSD	TF card / MMC card slot	microSD

В качестве рабочей станции был выбран Raspberry pi 3 model B. Одним из важнейших критериев при выборе данного микрокомпьютера была поддержка Ethernet на 10/100 Мбит, встроенные Wi-Fi и Bluetooth модуля, специализированная операционная система. Его возможностей хватает для минимально комфортной работы.

Периферия и коммутация

Взаимодействие с установкой происходит при помощи сенсорного дисплея 10.1inch HDMI LCD для Raspberry pi с интерфейсом подключения GPIO-HDMI. Беспроводное сетевое взаимодействие осуществляется посредством роутера D-LINK DIR-825

Основными параметрами для выбора роутера были: наличие двух диапазонов частот, два порта, высокая скорость и простота настройки.

Краткие технические характеристики D-LINK DIR-825:

- стандарт Wi-Fi: 802.11a/b/g/n/ac
- макс. скорость: 1167 Мбит/с
- поддержка опционального 4G-модема
- коммутатор 4xLAN
- поддержка VPN
- скорость портов 1000 Мбит/сек
- Поддержка двух диапазонов частот 2.4/5 Гц

Внешний вид составных частей установки представлен на рис. 1.

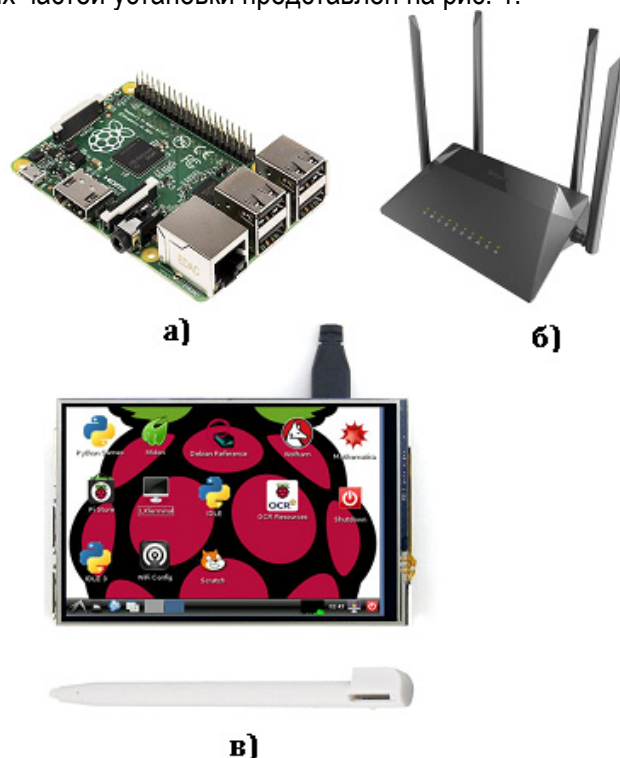


Рис. 1. Основные компоненты установки:

а) Raspberry pi 3 model B , б) D-LINK DIR-825 , в) 10.1inch HDMI LCD для Raspberry pi

Программное обеспечение

В качестве операционной системы была выбрана Raspbian. Это дистрибутив Linux на основе Debian для нашей аппаратной платформы.

Raspbian – это модификация дистрибутива GNU/Linux Debian, который распространяется бесплатно. Он имеет множество отличительных особенностей. Перечислим основные из них:

- возможность работы на ARM-процессорах;
- наличие в комплекте всех необходимых драйверов;

- хорошая оптимизация под сравнительно небольшой объем ОЗУ;
- поддержка GPIO «из коробки».

При выборе ОС для установки основным аргументом было наличие в комплекте всех необходимых драйверов для работы с одноплатным компьютером и периферией. Благодаря этому отпадает необходимость в дополнительных настройках.

Разработанная компактная лаборатория, может использоваться для некоторых дисциплин, которые ведутся на базе кафедры № 41. Для ряда предметов может понадобиться периферийное оборудование. Для большинства дисциплин подойдет предустановленное на устройство программное обеспечение; часть предметов может нуждаться в дополнительном программном обеспечении. Перечень дисциплин представлен в табл. 2.

Таблица 2

Перечень дисциплин, в рамках которых может быть использована разработанная компактная лаборатория

Дисциплина	Темы лабораторных занятий [6–11], которые могут выполняться на основе компактной лаборатории	Аппаратное обеспечение	Программное обеспечение
Компьютерные системы и сети	Создание одноранговой локальной сети посредством Ethernet Подключение типа звезда с использованием Ethernet кабеля Подключение типа точка-точка с использованием Wi-Fi Подключение типа точка-точка с использованием Bluetooth	Одноплатный компьютер Raspberry pi 3 model b; Сенсорный дисплей 10.1inch HDMI LCD для Raspberry pi с интерфейсом подключения GPIO-HDMI; Роутер D-LINK DIR-825 (далее «полный комплект лаборатории»)	ОС Raspbian (предустановлено)
Схемотехника цифровых и импульсных устройств	Синтез комбинационных схем Синтез регистров Синтез пересчетной схемы	Одноплатный компьютер Raspberry pi 3 model b; Сенсорный дисплей 10.1inch HDMI LCD для Raspberry pi с интерфейсом подключения GPIO-HDMI;	smartSim (предустановлено)
Математические методы моделирования информационных процессов и систем	Моделирование гауссовского белого шума Моделирование гауссовского окрашенного шума Моделирование Рэлеевского процесса Моделирование фильтра низких частот Моделирование RC цепочек Моделирование системы с заданной передаточной функцией Моделирование датчиков равномерно распределенных чисел Моделирование датчиков нормально распределенных чисел	Одноплатный компьютер Raspberry pi 3 model b; Сенсорный дисплей 10.1inch HDMI LCD для Raspberry pi с интерфейсом подключения GPIO-HDMI;	Mathematica (предустановлено), Python 2, Python 3 (предустановлено)

Дисциплина	Темы лабораторных занятий [6–11], которые могут выполняться на основе компактной лаборатории	Аппаратное обеспечение	Программное обеспечение
	Оценка точности моделирования		
Статистическая теория информационно-измерительных систем	Построение вариационных рядов. Расчет числовых характеристик. Построение модели линейной корреляции по несгруппированным данным Построение нелинейного уравнения модельной регрессии.	Одноплатный компьютер Raspberry pi 3 model b; Сенсорный дисплей 10.1inch HDMI LCD для Raspberry pi с интерфейсом подключения GPIO-HDMI;	Mathematica (предусмотрено), Python 2, Python 3 (предусмотрено)
Аппаратные интерфейсы	Аппаратные интерфейсы в ПК Интерфейсы компьютерных сетей Интерфейсы программных средств	Полный комплект лаборатории, предназначенный для дисциплины «Компьютерные системы и сети», может потребоваться дополнительное периферийное оборудование	OS Raspbian (предусмотрено)
Датчики и преобразователи информационно-измерительных систем	Доплеровский датчик присутствия и движения объектов Ультразвуковой датчик Инфракрасный датчик Лазерный гироскоп Изучение принципов работы температурного датчика DS1621	Одноплатный компьютер Raspberry pi 3 model b; Сенсорный дисплей 10.1inch HDMI LCD для Raspberry pi с интерфейсом подключения GPIO-HDMI; Комплект датчиков	--

Заключение

Разработанная компактная лаборатория, может использоваться для таких дисциплин: компьютерные системы и сети, схемотехника цифровых и импульсных устройств, математические методы моделирования информационных процессов и систем, статистическая теория информационно-измерительных систем, аппаратные интерфейсы, датчики и преобразователи информационно-измерительных систем. Для ряда предметов может потребоваться периферийное оборудование. Для большинства дисциплин подойдет предустановленное на устройство программное обеспечение; часть предметов может нуждаться в дополнительном программном обеспечении.

Благодаря возможности подключать периферийное оборудование, возможно расширить список дисциплин, использующих компактную лабораторию. Кроме того, её можно использовать в качестве мобильного компьютерного класса, что позволит снять нагрузку со специализированных лабораторий.

Библиографический список

1. Приказ Минобрнауки России от 12 марта 2015 года № 218 «Об утверждении федерального государственного образовательного стандарта высшего образования по направлению подготовки 11.03.04 – «Электроника и наноэлектроника (уровень бакалавриата)». URL: <http://fgosvo.ru/uploadfiles/fgosvob/110304.pdf>. (дата обращения: 12.02.2020).

2. Локальная сеть: основы функционирования компьютерных сетей. URL: <https://2hpc.ru/локальная-сети/> (дата обращения: 11.02.2020).
3. Настольная книга администратора Debian. URL: <https://debian-handbook.info/browse/ru-RU/stable/sect.network-config.html> (дата обращения: 20.10.2019).
4. Новиков Ю. В., Кондратенко С. В. Основы локальных сетей. М.: Интернет – Ун-т информ. технологий, 2005. 360 с.
5. Компьютерные сети: учеб. пособие по администрированию локальных и объединенных сетей / А. В. Велихов, К. С. Строчников, Б. Л. Кошкин. М.: Буккпресс, 2006. 304 с.
6. Рабочая программа дисциплины «Компьютерные системы и сети». СПб.: ГУАП, 2018.
7. Рабочая программа дисциплины «Схемотехника цифровых и импульсных устройств». СПб.: ГУАП, 2018.
8. Рабочая программа дисциплины «Математические методы моделирования информационных процессов и систем». СПб.: ГУАП, 2018.
9. Рабочая программа дисциплины «Статистическая теория информационно-измерительных систем». СПб.: ГУАП, 2018.
10. Рабочая программа дисциплины «Аппаратные интерфейсы». СПб.: ГУАП, 2018.
11. Рабочая программа дисциплины «Датчики и преобразователи информационно-измерительных систем». СПб.: ГУАП, 2018.

УДК 657:004(075.8)

А. Г. Бальков

магистрант кафедры вычислительных систем и программирования

В. И. Исаков – кандидат технических наук, доцент – научный руководитель

ОСНОВНЫЕ МЕХАНИЗМЫ ИНТЕГРАЦИИ 1С С ДРУГИМИ СИСТЕМАМИ

В современном мире существует множество систем, которые позволяют автоматизировать бизнес. Каждое предприятие старается объединить все свои системы в единую структуру. Для этого используются различные механизмы интеграций между системами.

1С – самый распространённый программный продукт для автоматизации бизнеса в РФ, а также по совместительству язык программирования. В ходе работы каждого предприятия возникает потребность интеграции базы данных 1С с системами на 1С или других платформах, поэтому целью данной работы является анализ основных механизмов интеграции 1С с другими системами.

Интеграция — это процесс установки связей между информационными системами предприятий и организаций для получения единого информационного пространства и организации поддержки сквозных бизнес-процессов предприятий и организаций [1].

Задача интеграции информационных и учетных систем состоит из двух взаимосвязанных частей: интеграция приложений и интеграция данных. Без интеграции данных невозможно провести интеграцию приложений.

Интеграция данных – это процесс компоновки информации из различных информационных систем (ИС) предприятий и организаций, установки ее однозначного соответствия в разных системах (мэппинг таблиц, мэппинг полей, мэппинг записей), синхронизация одинаковых информационных объектов в различных ИС.

Решая задачу интеграции данных, компания должна провести унификацию и стандартизацию нормативно-справочной информации (НСИ). Нормативно-справочная информация – условно-постоянная составляющая общей корпоративной информации. Эта информация используется при регламентации деятельности компании, она обеспечивает «сшивку» данных, сопровождающих бизнес-процессы компании. Другими словами, НСИ – это набор справочников, словарей, классификаторов, стандартов, регламентов, используемых в деятельности компании. НСИ является ядром информационного пространства компании. Наличие однозначной, структурированной, стандартизированной НСИ, управление которой ведется в соответствии с продуманными правилами и алгоритмами, – базис, обязательное условие создания эффективных интеграционных решений.

К основным механизмам интеграции относятся:

– **СОМ соединение** (*Component Object Model*) – это технологический стандарт от компании *Microsoft*, предназначенный для создания программного обеспечения на основе взаимодействующих компонентов объекта, каждый из которых может использоваться во многих программах одновременно.

– **API** (*Application Programming Interface*) – интерфейс программирования приложений, программный интерфейс приложений, реализуется с использованием *Web/HTTP* сервисов.

– **Выгрузка/Загрузка из Excel, XML** – *Excel* программа предназначена для обработки электронных таблиц (ЭТ). *XML* структурированный формат представления данных

Анализ механизмов

При выборе механизма интеграции обращают внимание на такие основные критерии как: скорость, корректность, универсальность, кроссплатформенность.

СОМ соединение

Плюсы:

1. Для интеграции с системами на платформе 1С не требуется дополнительного ПО.
2. Удобство и простота в разработке.
3. Не требуется доработка сторонней системы.

4. Корректность переносимых данных.

Минусы:

1. Для интеграции с другими системами требуется дополнительное ПО от поставщика.

2. Скорость обмена низкая.

3. Отсутствие универсальности, разработанная интеграция для одной системы не подойдет для другой.

4. Отсутствует кроссплатформенность.

Механизм API

Плюсы:

1. Универсальность.

2. Скорость обмена измеряется в миллисекундах.

3. Корректность переносимых данных.

4. Документация по эксплуатации.

5. Кроссплатформенность.

Минусы:

1. Требуется дополнительно ПО (*apache, IIS*).

2. Сложность в разработке.

3. Сложность в отладке кода.

4. Требуется постоянный доступ в интернет.

Заметим, что у каждого механизма есть свои плюсы и минусы. Поэтому для разработки используется каждый из них.

Практика применения механизмов интеграции

Интеграция с Telegram

Задача: увеличить скорость согласования заявок на расходование денежных средств (ДС), согласование заказов поставщику/клиента.

Текущие проблемы: руководитель подразделения зачастую не всегда находится у рабочего места, так как большое количество времени проводится на встречах, вдали от компьютера с 1С. А согласовать заявки на расходование ДС, заказы поставщику/клиента можно только в программе 1С. При этом без согласования ответственным лицом рабочий процесс останавливается, что влечет за собой потерю прибыли и клиентов.

Варианты решения проблемы: согласование заказов через почту, телефон, мессенджеры.

Решение:

Главным критерием задачи является оперативность и мобильность согласования. Вне зависимости от места положения ответственного лица, требуется иметь возможность оперативно согласовать заявку для дальнейшей работы сотрудников. В качестве среды для согласования выбран мессенджер *telegram*.

После анализа механизмов интеграции был сделан выбор в пользу *API*, так как этот механизм уже реализован *telegram* и является самым быстрым из всех рассмотренных.

Реализация интеграции 1С и *telegram* выполнялась с использованием *HTTP* запросов. В фоновом режиме 1С обращается к серверам *telegram* для получения новых событий. При создании новой заявки на согласование 1С автоматически отправляет ответственному лицу сообщение о согласовании. Предварительно пользователю необходимо авторизоваться в *telegram*.

Интеграция 1С – 1С

Задача: настроить нетиповой обмен данными между разными конфигурациями 1С.

Текущие проблемы: дублирование ввода информации в системы, ручной ввод, человеческий фактор.

Решение:

Для реализации данной задачи был выбран механизм *COM* соединения, который по умолчанию идёт в платформе 1С, то есть никакого дополнительного ПО не требуется. Кроме того, все конфигурации 1С поддерживают данный механизм. Также скорость разработки с использованием этого

механизма заметно выше, чем с *API*. По сравнению с выгрузкой/загрузкой из *EXCEL/XML* данный способ позволяет доработать только одну систему для реализации задачи. Минусом *COM* соединения является скорость, но для данной задачи скоростью можно пренебречь, так как обмен требуется не чаще $\frac{1}{2}$ раз в день. При этом важна надежность переносимых данных, что и обеспечивает *COM* соединение.

Загрузка данных из *EXCEL/XML*

Задача: загружать данные из онлайн магазина *OZON*.

Текущие проблемы: дублирование ввода информации в системы, ручной ввод, человеческий фактор.

Решение:

При подборе механизма интеграции для этой задачи приоритетным является использование *API*, но со стороны сервиса *OZON* не реализован требуемый функционал *API*, из-за чего невозможно использовать данный механизм. Кроме того, *COM* соединения с *OZON* не разработано. Таким образом, для этой задачи был выбран *EXCEL/XML* – универсальный механизм, позволяющий интегрироваться с 99 % систем.

Пользователь выгружает необходимую информацию из личного кабинета *OZON* и загружает её в 1С. Данная интеграция заметно уменьшила ручной труд, выполняемый сотрудниками, а также уменьшило количество ошибок при вводе информации.

В данной работе были рассмотрены три основных механизма интеграции 1С с сторонними системами. Наиболее предпочтительным механизмом интеграции является *API* в силу своей универсальности и скорости обмена, однако на практике не всегда возможно использование данного механизма.

Библиографический список

1. Морозова О. А. Интеграция корпоративных информационных систем.
2. Радченко М. Г., Хрусталева Е. Ю. 1С:Предприятие 8.3: практ. пособие разработчика: примеры и типовые приемы.
3. Документация API telegram. URL: <https://core.telegram.org> (дата обращения: 15.03.2020).
4. Методическая поддержка разработчиков 1С, Передача объектов 1С:Предприятия через COM. URL: <https://its.1c.ru/db/metod8dev/content/2262/hdoc> (дата обращения: 17.03.2020).

УДК 004.07

Д. А. Борисов

студент кафедры проблемно-ориентированных вычислительных комплексов

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

ОСОБЕННОСТИ И ХАРАКТЕРИСТИКИ ТВЕРДОТЕЛЬНЫХ УСТРОЙСТВ ФОРМАТА M.2

При покупке компьютерной техники важны технические и функциональные параметры оборудования. Покупателю требуется найти оптимальное решение соответствия производительности вычислительного средства и его цены. Для этого необходимо рассмотреть различные варианты конфигурации персонального компьютера, одним из компонентов которой является накопитель постоянной памяти. Основными характеристиками накопителя являются скорость записи и считывания данных, емкость и надежности накопителя данных. В настоящее время на рынке вычислительной технике представлены накопители двух видов: HDD (Hard Disk Drive) и SSD (Solid State Drive). Перспективным технологическим решением является SSD-диск, который представляет собой RAID-контроллер (микропроцессор, «распределитель») с подключенными к нему микросхемами флеш-памяти. Архитектура современного компьютера требует соединения материнской платы и винчестера. При этом существуют различные технологические стандарты их подключения – форм-факторы. Соблюдение стандартов гарантирует, что при установке диска, материнской платы или блока питания в корпус все крепежные отверстия, расположение интерфейсных разъемов будет единым для всех устройств вне зависимости от производителя, модели, функциональных возможностей. В данном обзоре рассмотрены особенности и характеристики SSD-накопителей формата M.2.

1. РАЗВИТИЕ ИНТЕРФЕЙСА SATA

В 2002 году разработан интерфейс SATA (Serial Advanced Technology Attachment), придя на смену PATA (Parallel ATA). Сейчас он используется для подключения абсолютного большинства жестких дисков и SSD-накопителей. В 2009 году появился вариант mSATA (Mini-SATA) для использования в мобильных устройствах [1].

На смену mSATA пришел интерфейс M.2. Обратим внимание, что в аббревиатуре нет букв «SATA». Поддержку разъёма M.2 на материнских платах начали делать с 2013 года. По своему назначению он очень схож с mSATA, однако позволяет обойти ограничение по пропускной способности SATA интерфейса.

Размеры M.2 SSD описываются одним числом, например, 2242. Первые 2 цифры – ширина, вторые 2 цифры – длина. Длина может варьироваться, и составлять 42, 60 или 80 мм. На материнской плате или корпусе ноутбука размещены фиксирующие накопитель винты, которые рассчитаны на разную длину устанавливаемых модулей.

И mSATA, и M.2 позволяют обходиться без шлейфов, кабелей питания, что повышает удобство, позволяет сделать компьютер компактнее. M.2 еще меньше, чем mSATA.

В жестких дисках (HDD (hard disk drive)) контроллер связывается с операционной системой посредством протокола AHCI (Advanced Host Controller Interface). Но, этот протокол не в состоянии использовать все возможности современных твердотельных накопителей. Это подтолкнуло к появлению нового протокола, который называется NVMe (Non-Volatile Memory Host Controller Interface Specification). Он характеризуется малыми задержками и позволяет выполнить больше операций в секунду, при этом минимально нагружая процессор.

1 M-KEY и B-KEY

Современные жесткие диски, и SSD в том числе, традиционно подключают к шине SATA7, цифра 7 означает 7-контактный разъём. Интерфейс SATA III имеет максимальную пропускную способность в 6 Гбит/с. Для обычных винчестеров такие скорости недостижимы, а вот для SSD-дисков развить гораздо большую скорость труда не составляет. Только смысла в этом нет, если интерфейс все равно не сможет «прокачать» поток данных с скоростью большей, на которую сам способен.

Поэтому появилась возможность задействовать шину PCI-Express (Peripheral Component Interconnect Express), имеющую большую пропускную способность.

То, какой интерфейс используется для подключения устройств, и определяет положение ключа (перемычки) у M.2 плашки. Также есть M+B ключ, который чаще всего подключается по SATA [2]. В таблице 1 проведено сравнение B и M ключа.

Таблица 1

Сравнение B key и M key

Тип ключа	B key (M.2 Socket2)	M key (M.2 Socket3)
Количество контактов	6	5
Расположение ключа	Контакты 12-19	Контакты 59-66
Поддерживаемые интерфейсы	PCIe x2, SATA, USB 3.0 (Universal Serial Bus), Audio, PCM (Pulse-code modulation)	PCIe x4 и SATA

Преимущество интерфейса PCI заключается в возможности передачи данных при использовании до четырех линий одновременно. В сочетании с логическим интерфейсом NVMe скорость чтения/записи у SSD-накопителя получается выше, чем при использовании интерфейса SATA и логического протокола AHCI. Сравним их поподробнее.

2. ХАРАКТЕРИСТИКИ

В табл.2 представлены сравнительные характеристики выше рассмотренных интерфейсов.

Таблица 2

Сравнение характеристик интерфейсов

Спецификация	SATA 3.0	PCI Express 2.0			PCI Express 3.0		
		x1	x2	x4	x1	x2	x4
Число линий	-	x1	x2	x4	x1	x2	x4
Полоса пропускания, Гбит/с	6	5	10	20	8	16	32
Схема кодирования	8b/10b	8b/10b	8b/10b	8b/10b	128b/130b	128b/130b	128b/130b
Эффективная пропускная способность, Гбайт/с	0,6	0,5	1	2	0,99	1,97	3,94

Для работы интерфейса NVMe на низком уровне выбрана стандартная последовательная шина PCI Express, которая не только легко масштабируется путём увеличения числа используемых линий, но и даже в варианте с одной линией (версии 3.0) как видно из таблицы 2, может обеспечить пиковую пропускную способность порядка 985 Мбайт/с, то есть, как минимум в полтора раза выше, чем привычный SATA-интерфейс. Также большинство современных массовых NVMe SSD используют две или четыре линии PCI Express. Таким образом, диапазон пропускной способности будет от 1,97 до 3,94 Гбайт/с. Основанные на флеш-памяти накопители благодаря своей многоканальной архитектуре могут задействовать такую быструю шину, поэтому скорость NVMe SSD при последовательных операциях оказывается в разы выше, чем у SATA SSD.

Переход SSD с SATA-интерфейса на NVMe предполагался не столько с целью увеличения пропускной способности, сколько для снижения латентностей, то есть задержек, и добавления в работу накопителей многопоточности. Открывшиеся возможности масштабирования скоростей последовательного доступа стали лишь полезным дополнением. Главное же в NVMe то, что данный интерфейс конструктивно направлен на работу с накопителями, которые имеют многоканальную параллельную архитектуру с быстрым случайным доступом.

В SATA протокол AHCI строится вокруг обслуживания одной общей очереди команд и явным образом предполагает, что поступающие команды обрабатываются устройством с высоким временем доступа – накопителем на жёстких магнитных дисках. По этой причине в AHCI не заложено никакого

параллелизма, ведь магнитные носители информации читают и записывают данные исключительно последовательно. Кроме того, HDD не могут предложить задержки меньше нескольких миллисекунд, а в AHCI нет никаких оптимизаций для снижения латентностей. На фоне высокого времени доступа физического носителя накладные расходы протокола попросту не имеют никакого значения. Когда SATA-интерфейс пришёл в SSD, протокол AHCI сразу же стал неудобным и даже неуместным [3]. В табл.3 приведены сравнительные характеристики рассмотренных протоколов.

Таблица 3

Сравнение AHCI и NVMe протоколов

	AHCI	NVMe
Максимальная глубина очереди	1 очередь	65536 очередей
	32 команды на очередь	65536 команд на очередь
Некэшируемые доступы к регистрам (2000 циклов каждый)	6 на команды вне очереди	2 на команду
	9 на команды очереди	
MSI-X и управление прерываниями	Одно прерывание	2048 MSI-X прерываний
	Управление отсутствует	
Параллелизм и многопоточность	Требуется фиксация синхронизации для выдачи команды	Фиксация не требуется
Эффективность для 4K-команд	Требуется два серийных запроса DRAM	Требуется один 64-байтовый запрос DRAM

Как видно из таблицы 3 NVMe выигрывает по пропускной способности и специально разработан для архитектуры SSD, что позволяет минимизировать задержки при работе с данными.

Он обладает двумя преимуществами. Во-первых, поддерживает многопоточность, позволяя формировать множественные очереди команд, которые могут обрабатываться многоканальным контроллером SSD в параллельном режиме. Во-вторых, соединяя процессор и носитель информации напрямую, без каких-либо посредников, NVMe-интерфейс приближает данные к точке их использования и убирает два ставших ненужными промежуточных звена: SATA-контроллер и транслятор SATA-команд, который преобразует принятые в протоколе AHCI обращения к секторам и дорожкам в запросы данных из страниц флеш-памяти.

Всё это уменьшает задержки к доступу информации. Как показывает практика, предлагаемый NVMe-интерфейсом способ доступа к данным, который естественен для твердотельного накопителя, позволяет уменьшить накладные расходы чуть ли не вдвое [2]. Это открывает путь для создания накопителей с латентностью на уровне десятка микросекунд. Такие NVMe-накопители уже существуют: например, Intel Optane SSD или Samsung Z-SSD [3]. Подобное качество обслуживания на базе SATA-интерфейса реализовать попросту невозможно.

Таблица 4

Сравнение SSD с интерфейсом SATA и SSD с интерфейсом PCIe и протоколом NVMe

	Типичный SATA SSD	Типичный NVMe SSD
Скорость последовательных операций	До 550 Мбайт/с чтение	До 3500 Мбайт/с чтение
	До 500 Мбайт/с запись	До 2500 Мбайт/с запись
Скорость случайных операций	До 100K IOPS (input/output operations per second) чтение	До 400K IOPS чтение
	До 90K IOPS запись	До 400K IOPS запись

Для примера посмотрим характеристики среднего представителя категории SATA и NVMe SSD, которые были представлены на портале 3DNEWS в результате тестирования множества SSD. Как можно видеть по данным таблицы 4, скорости на NVMe в разы выше в связи с описанными ранее особенностями NVMe SSD.

Сравнение жёсткого диска, SSD с SATA интерфейсом и NVME SSD с помощью ПО CrystalDiskMark

	SATA HDD		SATA SSD		NVME SSD	
	Чтение (Мб/с)	Запись (Мб/с)	Чтение (Мб/с)	Запись (Мб/с)	Чтение (Мб/с)	Запись (Мб/с)
Seq Q32T1	175.2	174.4	553.8	532.7	3539.5	1519.4
4KiB Q8T8	1.618	1.385	399.5	359.8	994.7	1508.5
4KiB Q32T1	1.621	1.289	334.0	312.8	735.2	589.3
4KiB Q1T1	0.646	1.290	36.90	129.2	66.46	215.7
	Seagate 3TB BarraCuda SATA		Samsung SSD 850 EVO SATA		Samsung SSD 850 EVO SATA	

В таблице 5 представлено сравнение жёсткого диска, SSD на SATA и SSD на PCI с NVME, собранное из отзывов пользователей сети Ситилинк, которые проводили тестирование накопителей в CrystalDiskMark [5].

1-й тест, Seq Q32T1 – Последовательная запись/чтение с глубиной очереди запросов 32 (Q), в 1 (T) поток. В этом тесте скорость обычно получается самой высокой, поскольку файл записывается в последовательные секторы диска, расположенные линейно. Этот результат не в полной мере отражает реальную скорость работы SSD при использовании в реальных условиях, но обычно сравнивают именно его.

4KiB Q8T8 – Произвольная запись/чтение в случайные секторы по 4 Килобайта, 8 – очередь запросов, 8 потоков.

3-й и 4-й тест аналогичны предыдущему, но с иным числом потоков и глубиной очереди запросов.

Из результатов тестирования можно сделать вывод, что любой из SSD гораздо быстрее жёсткого диска. NVME SSD выигрывает у SATA в разы по последовательной записи, но самое важное в 2 или больше раза в тестах на произвольную запись.

2 ЗАКЛЮЧЕНИЕ

При покупке вычислительного средства необходимо обратить внимание на тип интерфейса материнской платы и выбрать тот, который поддерживает форм-фактор M.2 SSD накопителей. Одновременно с этим следует предусмотреть возможность ее подключения через интерфейс NVMe. Это связано с тем, что ассортимент M.2 NVMe SSD по сравнению с NVMe-накопителями в других форм-факторах гораздо шире, и подобрать подходящий по сочетанию характеристик вариант куда проще. Они компактнее остальных вариантов: не требуют никакого кабельного подключения и не изменяют габаритов сборки после своей установки. Практически отсутствуют проблемы совместимости со стационарной или мобильной системами. Также для M.2 NVMe SSD разработаны специализированные платы-переходники, которые позволяют устанавливать их не только в специализированные разъемы, но и в стандартные слоты PCI Express на материнской плате.

Библиографический список

1. Новостной портал о новинках в компьютерном железе. 3D News – Daily Digital Digest. URL: <https://3dnews.ru/> (дата обращения: 24.05.2020).
2. Техноблог от Andriney. URL: <https://andiriney.ru/> (дата обращения: 24.04.2020).
3. Техноблог о ПК и всех сопутствующими вещами. URL: <http://pc-information-guide.ru/> (дата обращения: 24.05.2020).
4. Портал Remontka.pro о ремонте и лечении компьютеров. URL: <https://remontka.pro/> (дата обращения: 24.05.2020).

УДК 519.252

В. А. Велюга

магистр кафедры проблемно-ориентированных вычислительных комплексов

А. В. Яковлев – кандидат технических наук, доцент – научный руководитель

ИСПОЛЬЗОВАНИЕ ИНСТРУМЕНТОВ DEEPSPECTRUM И AUDEEP ДЛЯ АНАЛИЗА РЕЧИ

Введение

Сегодня становится все больше областей человеческой деятельности, где велика цена человеческой ошибки и нет возможности оценить состояние человека дистанционно без датчиков и специального оборудования. К таким областям относятся, здравоохранение (дистанционный мониторинг здоровья), транспортная безопасность (дистанционный контроль состояния водителя транспортного средства) и прочие. Неудовлетворительное состояние здоровья специалиста, его нахождение в состоянии утомления или даже стресса может привести к ошибкам или даже катастрофам. Для пресечения такого исхода событий разрабатываются и совершенствуются системы и алгоритмы мониторинга состояния человека в процессе исполнения им рабочих обязанностей. Одним из перспективных направлений исследований является использование речи для оценки состояния человека.

Целью проводимого исследования является изучение возможностей современных программных и алгоритмических средств для оценки уровня стресса человека по его речи. В данной работе исследован программный инструментарий, на который можно ориентироваться для достижения цели.

1. Речевой сигнал и методы его анализа

Речевой сигнал — это акустический сигнал, который представляет собой процесс распространения энергии акустических колебаний в упругой среде. С точки зрения физиологии — это продукт высшей нервной деятельности человека. Речевой сигнал легко регистрируется, для его регистрации не требуется специальное оборудование и не происходит какого-либо стеснения деятельности человека.

Графически речь можно представить в виде волноформы (рис. 1) и спектрограммы (рис. 2).

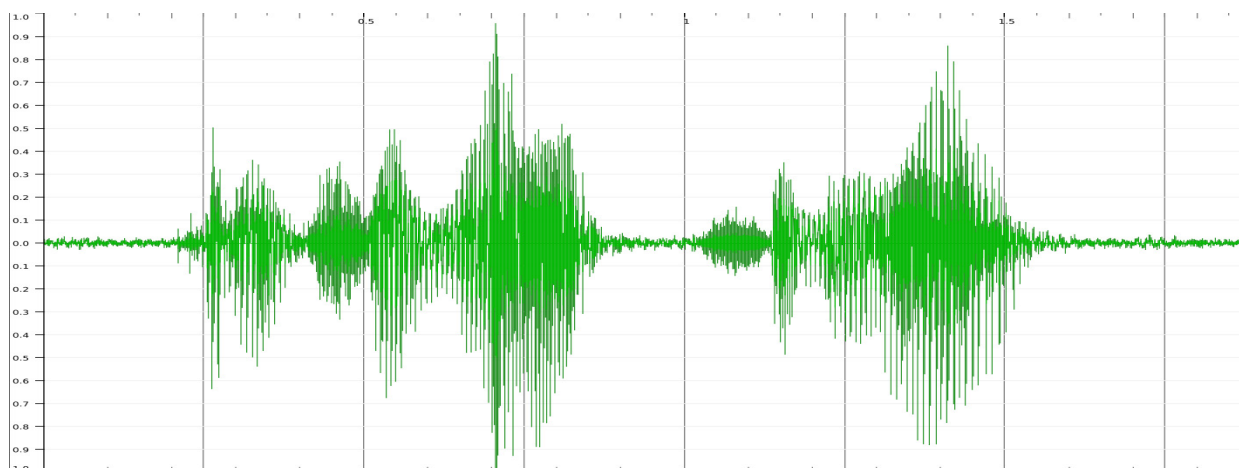


Рис. 1. Речевой сигнал фразы «речевой сигнал»

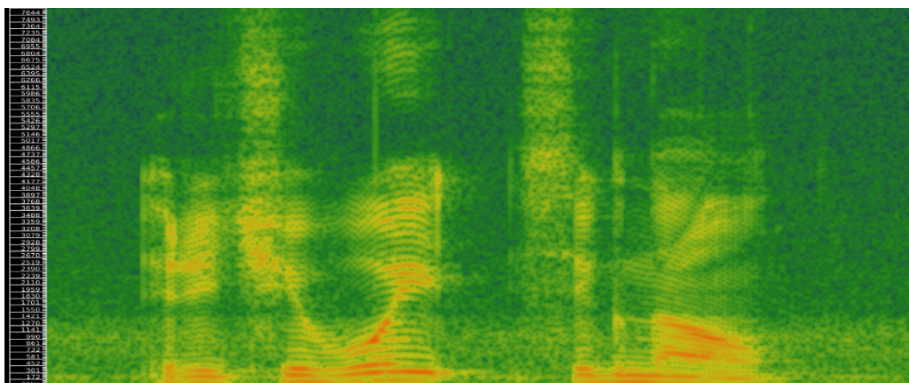


Рис. 2. Спектрограмма речевого сигнала фразы «речевой сигнал»

Наиболее распространенными путями анализа речевого сигнала для построения систем оценки состояния человека являются:

- Преобразование речевого сигнала в массив цифр, путем расчета паралингвистических характеристик речи (около 6 000 характеристик каждые 25 мс). Наиболее известный инструмент с таким подходом – openSMILE [1]. Такой подход проблематичен, поскольку он ресурсоёмок.
- Обработка речевого сигнала как целостного образа. В этом варианте наиболее часто используются нейронные сети.

Проведенный нами анализ показал, что перспективным для достижения цели исследования могут быть библиотеки DeepSpectrum и auDeep.

2. Библиотека DeepSpectrum

DeepSpectrum — это открытая библиотека, написанная на Python, для извлечения признаков из аудиоданных с помощью предварительно обученных свёрточных нейронных сетей (CNN). Эта библиотека, с помощью конвейера извлечения, создает визуальные представления аудиоданных, такие как спектрограммы или хромограммы, которые затем передаются в предварительно обученную свёрточную нейронную сеть. Активации определенного слоя формируют конечные векторы признаков. Этапы использования библиотеки DeepSpectrum приведены на рис. 3.

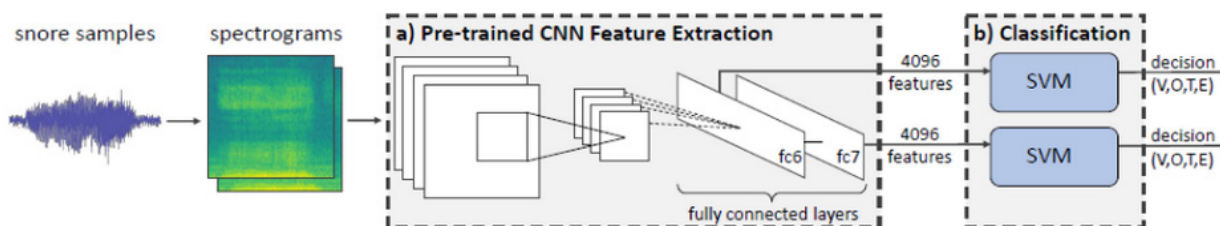


Рис. 3. Технология работы библиотеки DeepSpectrum [2]

Код DeepSpectrum оказался достаточно сложным для понимания ввиду отсутствия литературы и почти полного отсутствия документации. При детальном изучении DeepSpectrum, внутри была обнаружена библиотека auDeep, которая в большей степени удовлетворяет задачам нашего исследования и имеет более качественную документацию. По этой причине дальнейшее исследование проводилось именно с использованием auDeep.

3. Библиотека auDeep

Библиотека auDeep [3] — это набор инструментов, написанных на Python, для обучения неконтролируемых аудио признаков с помощью глубоких нейронных сетей (DNN, от англ. Deep Neural Network) [3].

Основным отличием этой библиотеки является извлечение признаков из аудиоданных с помощью глубоко повторяющихся автоэнкодеров. Однако алгоритмы обучения основных признаков не ограничиваются аудиоданными. Использование auDeer происходит в пять этапов (рис. 4):

1. Извлечение спектрограммы (англ. preprocess): извлечение спектрограмм и метаданных из необработанных аудиофайлов.
2. Обучение автоэнкодера (англ. train): обучение DNN на извлечённых спектрограммах.
3. Генерация признаков (англ. generate): генерация признаков с использованием обученного DNN.
4. Оценка признаков (англ. evaluate): оценка сгенерированных признаков.
5. Экспорт (англ. export): экспорт сгенерированных признаков в CSV/ARFF.



Рис. 4. Этапы использования auDeer [3]

В качестве тестовых данных для работы библиотеки auDeer нами был использован открытый датасет ESC-10 [4]. Этот датасет содержит классы различных звуков (звуки природы, животных и пр.). Это открытый высококачественный датасет, в котором присутствуют также и человеческие звуки (плач ребенка). Отметим, что формирование специфического датасета (в нашем случае — семплов речи людей в разных состояниях) — это отдельная исследовательская задача.

Результаты оценки качества обучения библиотекой auDeer на датасете ESC-10 в виде матрицы метрики ошибки (англ. confusion matrix) представлены на рис. 5.

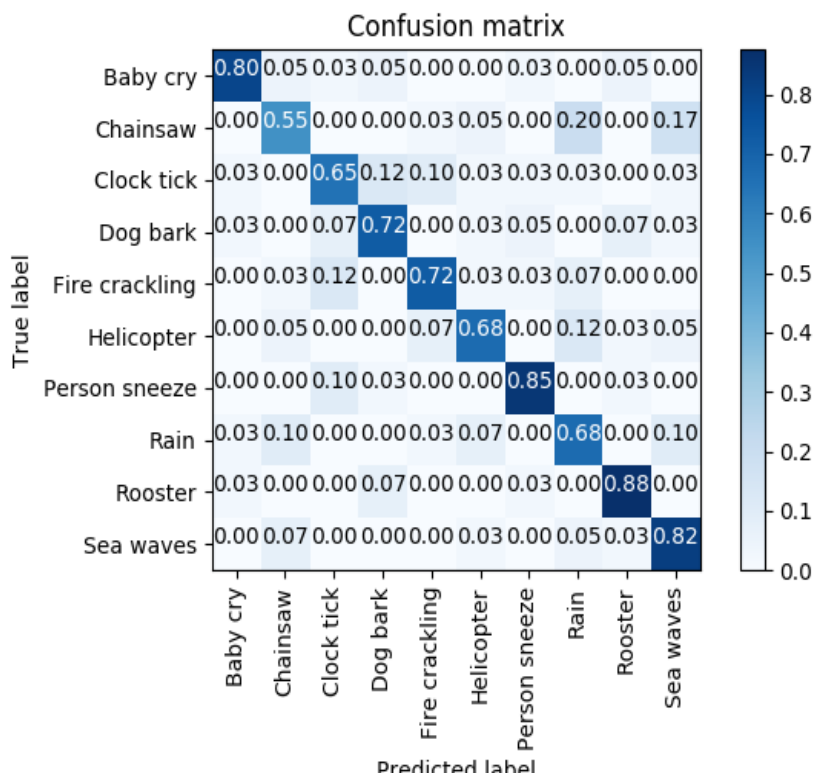


Рис. 5. Confusion matrix

Матрица метрики ошибки используется для определения эффективности модели классификации, то есть, насколько качественно и однозначно обученная модель может определить верный класс [5]. Как можно видеть в нашем примере на рис. 5, средняя вероятность верной классификации звука составляет ~75 %. Например, звуки моря определялись верно в 82 % случаев, но со звуками бензопилы наша модель ошибалась в разы чаще и классифицировала верно только в 55 % случаев.

Поскольку это не человеческая речь, на данный момент невозможно сказать о том, какого качества будет модель классификации состояния стресса человека по его речи, но средний показатель в 75 % по датасету ESC-10 был бы желателен.

Заключение

Проведенные расчеты показали, что библиотека `auDeep` обеспечивает приемлемое качество классификации акустических данных. Вместе с тем, анализ публикаций показывает, что качество классификации в нашей предметной области в значительной степени зависит от уровня подготовки датасета, поступающего на вход алгоритма. Поэтому актуальной задачей является подбор датасетов из области оценки состояния человека. В частности состояния стресса. К сожалению, пока таких датасетов найдено не было. Поэтому ближайшими задачами, помимо работы с рассмотренными в докладе библиотеками, будет создание собственных датасетов.

Библиографический список

1. OpenSMILE // `audEERING`. URL: <https://www.audeering.com/opensmile/> (дата обращения: 30.04.2020).
2. *Amiriparian S. et al.* Snore Sound Classification Using Image-Based Deep Spectrum Features // Proc. Interspeech 2017. Stockholm, Sweden, 2017. Vol. 17. P. 2017–2434.
3. *Freitag M. et al.* `auDeep`: Unsupervised Learning of Representations from Audio with Deep Recurrent Neural Networks // J. Mach. Learn. Res. 2017. Vol. 18.
4. ESC-10 // GitHub. URL: <https://github.com/karolpiczak/ESC-10> (дата обращения: 13.05.2020).
5. Confusion Matrix // Medium. 2019. URL: <https://medium.com/@r.arunadevi96/confusion-matrix-22502fea1cbd> (дата обращения: 15.05.2020).

УДК 004.652

А. А. Ветошеев

магистрант кафедры аэрокосмических измерительно-вычислительных комплексов

Н. С. Кукушкина

магистрант кафедры аэрокосмических измерительно-вычислительных комплексов

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

ОСОБЕННОСТИ РЕАЛИЗАЦИИ ETL-ПРОЦЕССА ПРИ КОНСОЛИДАЦИИ ДАННЫХ В СИСТЕМЕ ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ

Анализировать можно как качественные, так и некачественные данные. Результат, полученный в ходе интеллектуального анализа данных, напрямую зависит от того, насколько правильно подготовлены исходные данные.

На практике данные, необходимые для бизнес-анализа, могут быть взяты из различных источников и представлены в различных типах и форматах. Они могут храниться в отдельных файлах, в учетных системах или в базах данных. Кроме того, данные часто бывают избыточными или, напротив, недостаточными, а также могут содержать отклонения и «выбросы», дубликаты и пропуски.

Для обеспечения полноценного и максимально качественного анализа необходимо проведение предварительной подготовки данных, которая является важным этапом процесса интеллектуального анализа данных. Целью этого этапа является доведение данных до необходимого уровня информативности и правильная организация их хранения.

Консолидация – комплекс методов и процедур, направленных на извлечение данных из различных источников, обеспечение необходимого уровня их информативности и качества, преобразование в единый формат, в котором они могут быть загружены в хранилище данных или аналитическую систему [1].

Консолидация данных – начальный этап реализации любого аналитического проекта или задачи. К основным критериям, позволяющим назвать оптимальным полученный в ходе консолидации набор данных, можно отнести:

- высокую скорость доступа к данным;
- компактность их хранения;
- целостность структуры данных;
- непротиворечивость данных.

В процессе консолидации данных решаются следующие задачи:

- выбор источников данных;
- разработка стратегии консолидации;
- оценка качества данных, их обогащение и очистка;
- перенос в хранилище данных.

Наличие на предприятии хранилища данных делает анализ намного более простым и эффективным. Использование хранилищ обходится дешевле, чем использование отдельных файлов и баз данных. Однако далеко не организации имеют такие хранилища. Тогда в качестве источника данных используются данные из применяющихся на предприятии информационных систем.

Процесс консолидации данных можно представить в виде структурной схемы, показанной на рис. 1.

Основным процессом, лежащим в основе консолидации данных, является ETL-процесс: extraction (извлечение), transformation (преобразование), loading (загрузка). Под ETL понимается составной процесс переноса данных из одного или нескольких источников в целевую систему хранения. Процесс ETL решает следующие задачи:

- загрузка данных из различных источников;
- поиск, очистка и исправление ошибок в данных;
- приведение к единым метрикам и размерностям;
- агрегация до необходимой детализации;
- выгрузка данных в систему или хранилище.

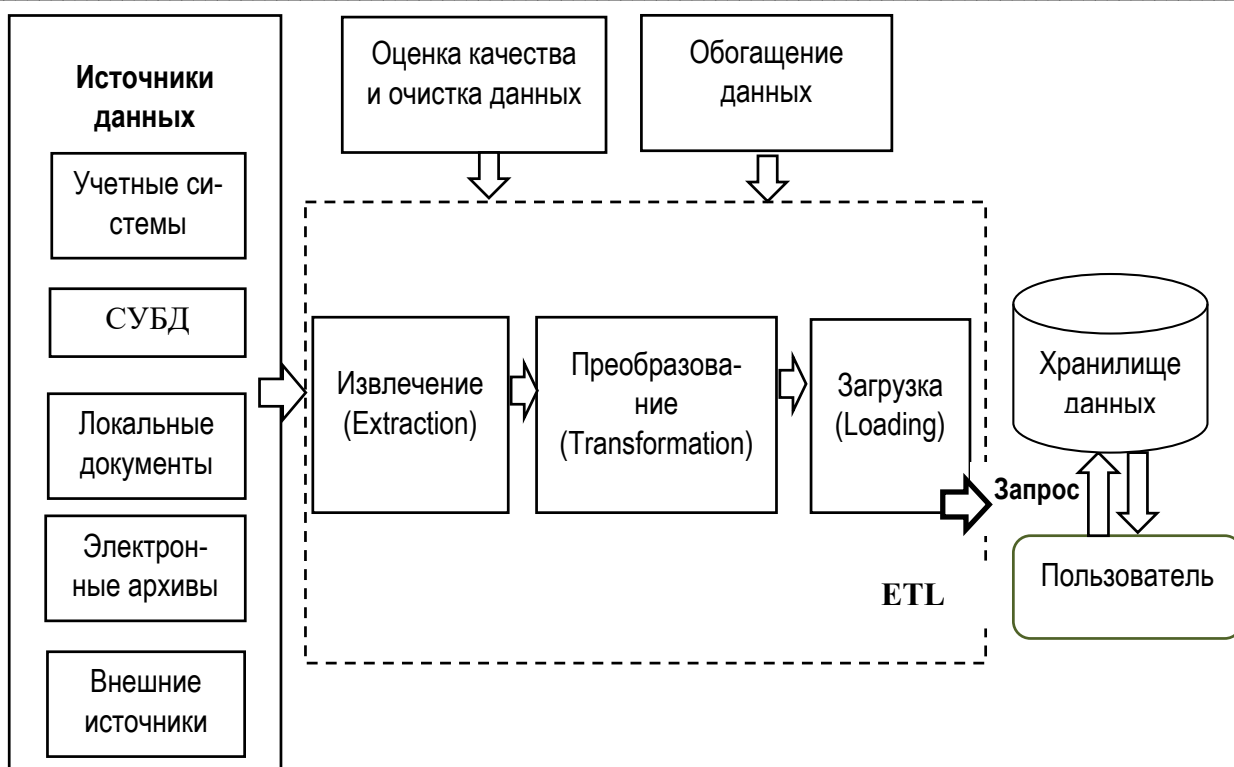


Рис. 1. Процесс консолидации данных

Существуют различные пути реализации процесса ETL. Из наиболее традиционных можно выделить разработку приложения ETL, создание комплекса встроенных программных процедур, использование специального инструментария ETL.

ETL-приложения извлекают данные из исходных баз данных, преобразуют их в формат, который поддерживается базой данных целевой системы, а после загружают в эту базу подготовленные данные.

Целью любого ETL-приложения является своевременная доставка информации из внешних источников в систему, с которой работают пользователи [2]. Приложения ETL используются при миграции данных внешних систем в хранилище данных систем бизнес-аналитики. Таким образом, организация ETL-процесса является составной частью при разработке или эксплуатации любого хранилища данных.

Проектирование и разработка ETL-процесса – одна из самых важных задач проектировщика хранилища данных [3]. В этом случае процесс ETL имеет следующие свойства:

- объем данных, извлекаемый из систем источников и интегрируемый в хранилище, может быть достаточно большим и занимать десятки терабайт;
- периодичность этого процесса не всегда определяется потребностью конечного пользователя в оперативных данных, но часто и размером загружаемых данных, так как ETL-процесс может занимать большое количество времени;
- на разных этапах процесса ETL формируются метаданные хранилища данных и обеспечивается качество данных;
- во время миграции данных может произойти потеря некоторой информации, поэтому необходимо обеспечить контроль за поступлением данных в хранилище;
- ETL-процесс может восстанавливаться после сбоев без потери данных.

При разработке ETL-сценария для хранилища данных часто имеет место следующая последовательность действий [4]:

- Планирование ETL-процесса, включающее в себя создание диаграммы потоков данных от исходных источников, определение требуемых преобразований, методов генерации ключей и алгоритма операций для каждой целевой таблицы.

– Заполнение таблиц измерений, включающее разработку и верификацию процесса загрузки данных в статические таблицы измерений и механизмов изменения данных для каждой таблицы измерений.

– Заполнение таблиц фактов, состоящее из разработки процессов первоначальной загрузки данных и периодического заполнения таблиц фактов, построения агрегатов и разработки процедур автоматизации процесса ETL.

Для реализации ETL-сценария существует множество инструментов интеграции данных: платные и бесплатные «open-source» системы с открытым исходным кодом. Некоторые из бесплатных и открытых инструментов ETL принадлежат поставщикам, которые стремятся продать корпоративный продукт, другие обслуживаются и управляются сообществом разработчиков.

Рассмотрим бесплатные ETL-инструменты для интеграции данных.

Apache Airflow (<https://airflow.apache.org/>) – платформа с удобным веб-интерфейсом, позволяющая создавать, планировать и отслеживать рабочие процессы. Дает возможность пользователям объединять задачи, которые нужно выполнить в строго определенной последовательности по заданному расписанию. Пользовательский интерфейс поддерживает визуализацию рабочих процессов, что помогает отслеживать прогресс и видеть возникающие проблемы.

Apache Kafka (<https://kafka.apache.org/>) – распределенная потоковая платформа, позволяющая пользователям публиковать и подписываться на потоки записей, хранить потоки записей и обрабатывать их по мере появления. Kafka используют для создания конвейеров данных в реальном времени. Он работает как кластер на одном или нескольких серверах, отказоустойчив и масштабируем.

Talend Open Studio (<https://www.talend.com/products/talend-open-studio/>) – платформа интеграции программного обеспечения, которая предоставляет решения для интеграции данных, управления данными, качества данных и их подготовки. Основным средством настройки процесса преобразования данных в Talend является специальный визуальный редактор, позволяющий добавлять и настраивать отдельные узлы преобразования данных и связи между ними.

Таким образом, в данной работе был рассмотрен ETL-процесс, описаны его основные задачи и пути реализации. Проведен обзор свойств ETL-процесса и последовательности действий при его конструировании. Рассмотрены наиболее популярные open-source решения реализации ETL-процесса и интеграции данных.

Библиографический список

1. Паклин Н. Б., Орешков В. И. Бизнес-аналитика от данных к знаниям. СПб.: Питер, 2012. 706 с.
2. Проектирование и разработка процесса ETL. URL: <https://www.intuit.ru/studies/courses/599/455/lecture/10175> (дата обращения: 15.05.2020).
3. Что такое ETL: как справиться с анализом big data. URL: <https://mcs.mail.ru/blog/что-такое-etl-ili-kak-spravitsya-s-analizom-big-data> (дата обращения: 15.05.2020).
4. Консолидация данных – ключевые понятия. URL: <http://asu-analitika.ru/konsolidacija-dannyh-kljuchevye-ponjatija> (дата обращения: 15.05.2020).

УДК 303.721

В. В. Вихров

студент кафедры проблемно-ориентированных вычислительных комплексов

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

ДЕСКРИПТИВНЫЙ АНАЛИЗ ФАКТОРОВ, ВЛИЯЮЩИХ НА СТОИМОСТЬ НЕДВИЖИМОСТИ

Дескриптивная теория (описательная статистика) – один из разделов статистической науки, в рамках которого изучаются методы описания и представления основных свойств данных. Позволяет обобщать первичные результаты, полученные при наблюдении или в эксперименте [1].

Изменение рыночной стоимости любой жилой недвижимости зависит от целого ряда факторов. Эти факторы могут быть отнесены к трем различным иерархическим уровням.

Первый уровень (региональный) – уровень влияния факторов, носящий общий характер, не связанных с конкретным объектом недвижимости и не зависящих непосредственно от него, но косвенно влияющих на процессы, происходящие с недвижимостью на рынке.

Второй уровень (местный) – уровень влияния локальных факторов в масштабе города, района, поселения.

Третий уровень (непосредственного окружения) – уровень влияния факторов, связанных с объектом недвижимости и во многом обусловленных его характеристиками [2].

В исследовании будет рассматриваться недвижимость по третьему уровню, так как, в основном, люди, ищущие недвижимость, рассматривают ее на этом уровне.

Постановка задачи

В результате исследования были решены следующие задачи:

- поиск данных по городской недвижимости в России, на основе которых будет проводиться анализ факторов, влияющих на стоимость;
- выбор метода агрегирования данных;
- проведение дескриптивного анализа данных выбранным методом;

Описание исследования

Объектом исследования данной работы является массив данных по городской недвижимости, на основе которого будет проведено исследование влияния характеристик недвижимости на ценообразование дескриптивным методом.

Всё исследование массива данных было проведено на языке python с использованием библиотек seaborn.

После получения результата работы скриптов была проведена оценка факторов, влияющих на стоимость недвижимости.

Результаты исследования

Для поиска датасета по недвижимости был выбран ресурс [kaggle.com/](https://www.kaggle.com/) – система организации конкурсов по исследованию данных, а также социальная сеть специалистов по обработке данных и машинному обучению[3]. В этой системе сосредоточено огромное количество различных датасетов, поэтому найти в ней данные для исследования не представляет сложности. Рассмотрим пару массивов данных и сравним их:

1. «UK Housing Prices Paid» – содержит в себе все продажи в Англии за 1995 год[4].
2. «Flats for rent in Budapest» – в нем сосредоточены цены на аренду в Будапеште[5].
3. «Price of flats in Moscow» – датасет со стоимостью квартир в Москве за 2018 год[6].

В данной работе был рассмотрен третий вариант, так как в отличие от первого и второго вариантов в нем представлены более свежие и актуальные с точки зрения географии данные по стоимости покупки жилья.

На сегодняшний день для дескриптивного анализа в первую очередь характерно графическое представление данных, так как стандартные методы представления типа таблиц быстро теряют свою наглядность с ростом объема информации.

Рассмотрим данные факторным методом – то есть будем сравнивать цену с каждым из показателей и оценивать зависимость. В выбранном наборе данных присутствуют следующие показатели:

- жилая площадь;
- вид дома (кирпичный, монолит и так далее);
- этаж (в данном массиве данных или первый и последний этажи или остальные);
- общая площадь;
- расстояние от центра;
- время на преодоление расстояния от метро в минутах на машине;
- время на преодоление расстояние от метро в минутах пешком;
- район квартиры;
- площадь кухни;
- расстояние от центра Москвы;

Будет использоваться такие библиотеки в python как seaborn, pandas, matplotlib, функционал которых представлен в табл. 1.

Таблица 1

Используемые библиотеки и их функционал

Библиотека	Функционал
seaborn	высокоуровневое API на базе библиотеки matplotlib. В библиотеке есть достаточно сложные типы визуализации, которые в matplotlib потребовали бы большого количество кода.
pandas	высокоуровневая Python библиотека для анализа данных. В экосистеме Python, pandas является наиболее продвинутой и быстроразвивающейся библиотекой для обработки и анализа данных.
matplotlib	библиотека на языке программирования Python для визуализации данных двумерной графикой. Получаемые изображения могут быть использованы в качестве иллюстраций в публикациях.

В частности в seaborn будут использоваться два основных метода построения графиков – это boxplot и jointplot. Остановимся на них подробнее.

Jointplot – это гибрид гистограммы и графика двух переменных, он наглядно показывает зависимости двух переменных.

Boxplot – состоит из коробки, усиков и точек. Коробка показывает интерквартильный размах распределения, то есть соответственно 25 % (Q1) и 75 % (Q3) перцентили. Черта внутри коробки обозначает медиану распределения [7].

Усы отображают весь разброс точек кроме выбросов, то есть минимальные и максимальные значения, которые попадают в промежуток (Q1 – 1.5IQR, Q3 + 1.5IQR), где IQR = Q3 – Q1 — межквартильный размах. Точками на графике обозначаются выбросы (outliers) — те значения, которые не

вписываются в промежуток значений, заданный усами графика. Для лучшего понимания обратим внимание на рис. 1 [8].

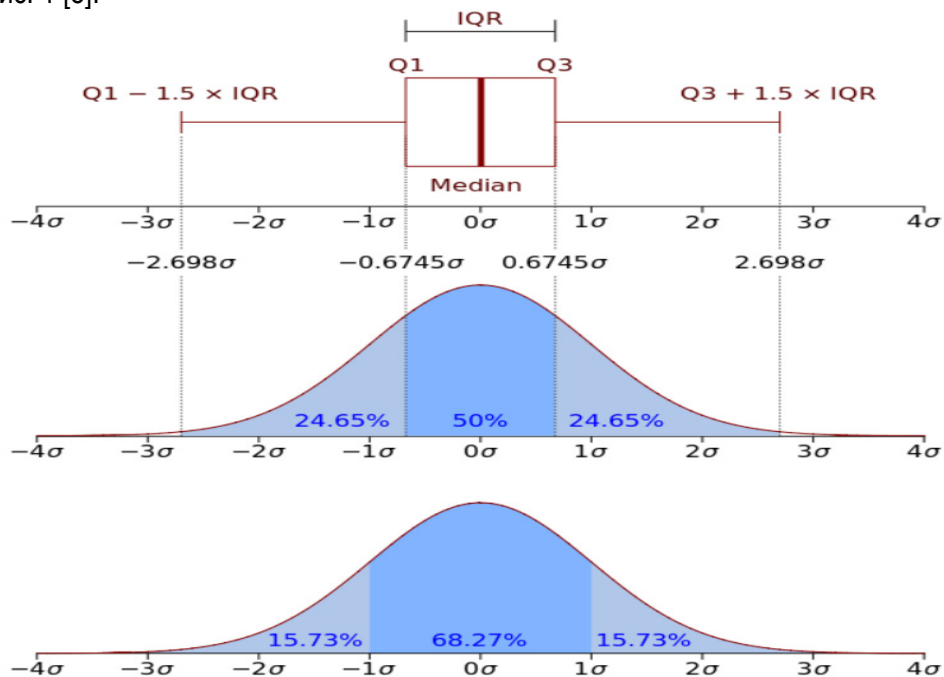


Рис. 1. Наглядное объяснение boxplot

Рассмотрим некоторые параметры с помощью методов jointplot и boxplot. На рис. 2 можно увидеть зависимость цены от жилой площади и от общей площади квартиры.

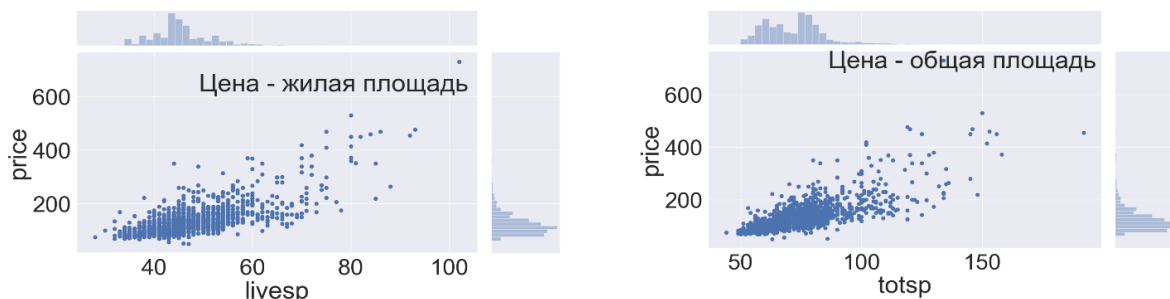


Рис. 2. Графики зависимости цены от общей площади, жилой площади

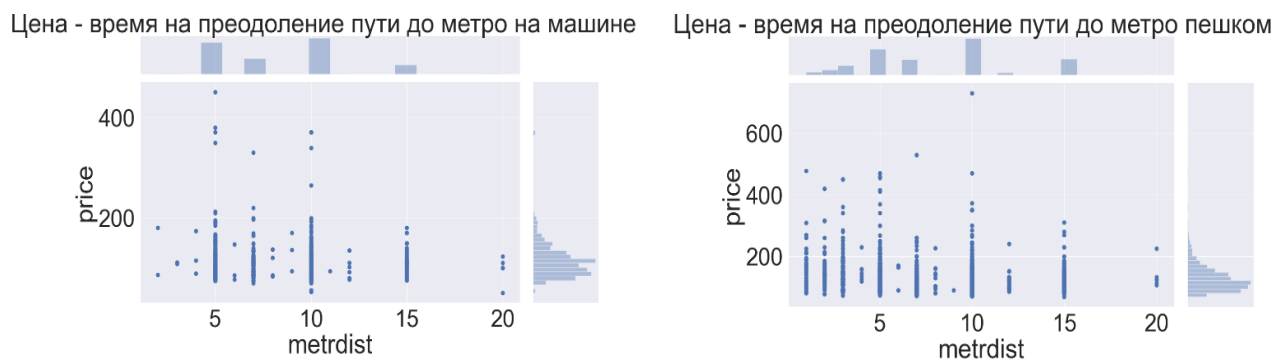


Рис. 3. Графики зависимости цены от времени на преодоление расстояния от метро на машине и пешком

На них отлично прослеживается зависимость цены от жилой и общей площадей – скопление точек (отдельных квартир) уходит вверх, то есть чем больше площади, тем больше цена.

На рис. 3, где приведен график зависимости цены от времени на преодоление расстояния от метро на машине и пешком соответственно. На обоих графиках наблюдается более выделенная зависимость, так как столбцы точек более резко уменьшаются по оси цены (price).

На рис. 4 приведен график зависимости цены от площади кухни. На нем существует некоторый спуск 5 точек после 22.5 метров, им можно пренебречь, так как общее количество точек составляет более 2000, прослеживается общая тенденция – облако точек уходит вверх. Это значит, что зависимость прямая – чем больше площадь кухни тем больше цена.



Рис. 4. График зависимости цены от площади кухни

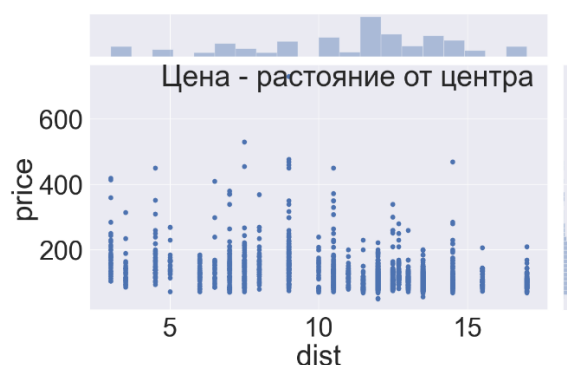


Рис. 5. График зависимости цены от расстояния от центра

На рис. 5 не так наглядно, но прослеживается обратная зависимость цены на квартиру от расстояния до центра (столбцы точек постепенно становятся ниже).

В связи с тем, что на графике зависимости цены от расстояния до центра (рис. 5) плохо видна зависимость, датасет был поделен по возрастанию общей площади квартиры на 4 разных и сделаны соответствующие графики, их можно увидеть на рис. 6.

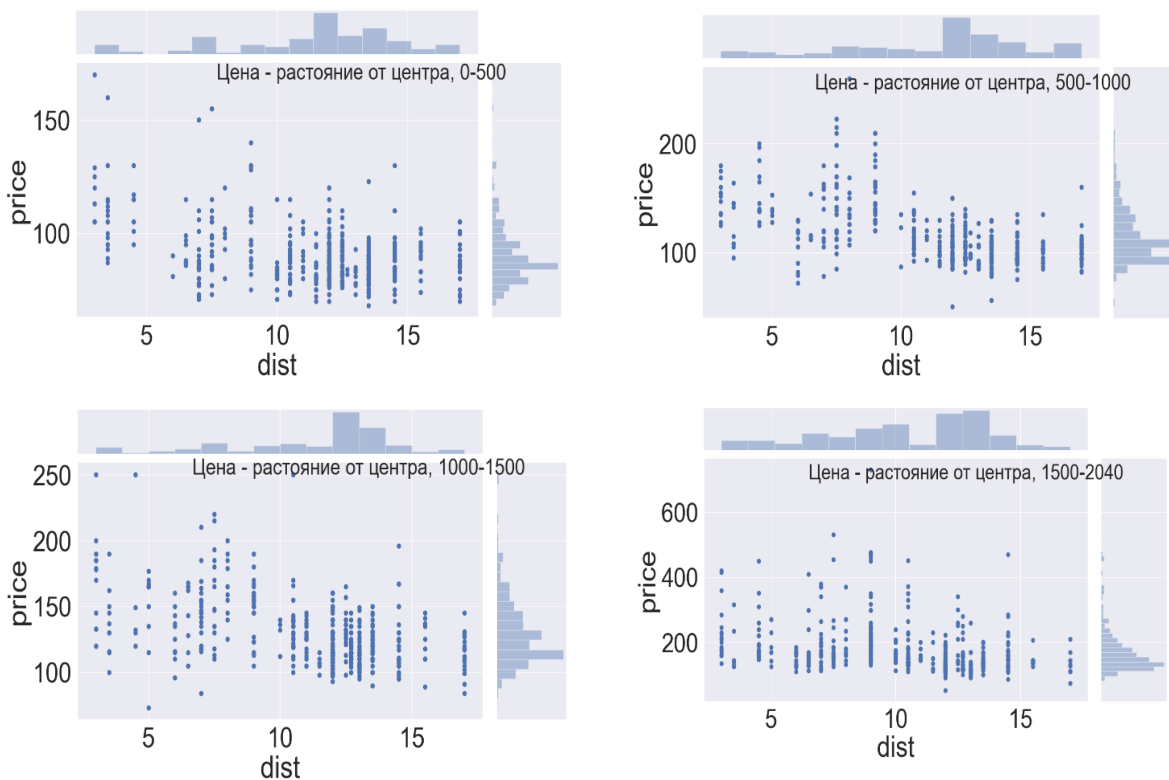


Рис. 6. Графики зависимости цены от расстояния от центра по возрастанию общей площади квартиры

После деления массива данных на 4 разных массива по возрастанию площади лучше видна зависимость цены от расстояния до центра. На всех графиках видны выбросы точек, возможно, это связано с завышенной стоимостью жилья в элитных районах- Москвы на расстоянии 6-10 километров от центра Москвы. Общая тенденция прослеживается – чем больше расстояние от центра, тем меньше цена.

Теперь проведем анализ с использованием boxplot. На рис. 7 представлен график зависимости цены от вида дома (0 – кирпич и монолит, 1 – другие) по графику видно, что кирпич и монолит в общем случае для Москвы дешевле, чем остальные виды материалов для постройки дома, так как коробка и усики расположены ниже по оси цены как раз у кирпичных и монолитных домов.

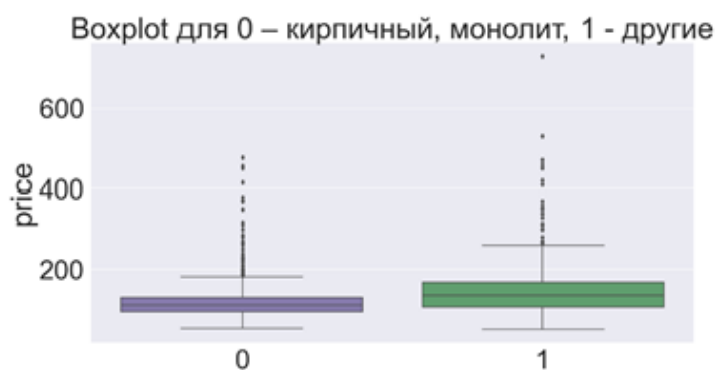


Рис. 7. Boxplot для вида дома

Так же воспользуемся методом boxplot для рассмотрения зависимости цены от этажа квартиры, на рис. 8 абсолютно так же как и на предыдущем графике видно, что коробка и усики для первого и последнего этажей ниже по оси цены, это значит, что первый и последние этажи дешевле, чем остальные.

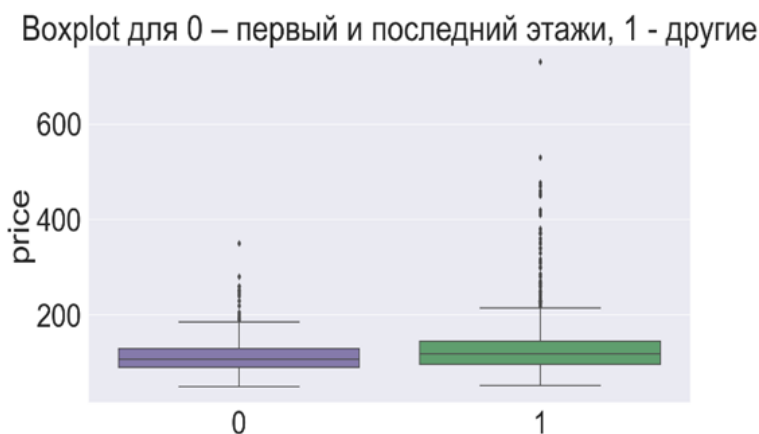


Рис. 8. Boxplot для этажей

Наконец, на рис. 9 рассматриваются разные районы Москвы (1. Наблюдения сгруппированы на севере, вокруг Калужско-Рижской линии метрополитена 2. Север, вокруг Серпуховско-Тимирязевской линии метрополитена 3. Северо-запад, вокруг Замоскворецкой линии метрополитена 4. Северо-запад, вокруг Таганско-Краснопресненской линии метрополитена 5. Юго-восток, вокруг Люблинской линии метрополитена 6. Юго-восток, вокруг Таганско-Краснопресненской линии метрополитена 7. Восток, вокруг Калининской линии метрополитена 8. Восток, вокруг Арбатско-Покровской линии метрополитена).

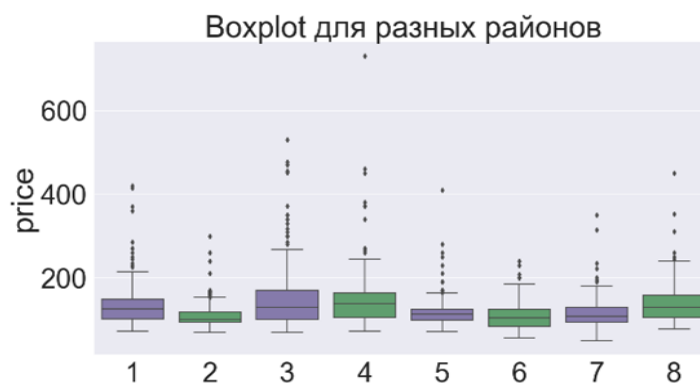


Рис. 9. Boxplot для районов

По графику видно как распределена средняя цена на квартиры в различных районах (коробки и усики различных районов лежат по разному на оси цены), это значит что в общем случае дешевле всего квартиры вокруг Серпуховско-Тимирязевской линии метрополитена Москвы, а дороже на северо-западе.

Выводы

В ходе исследования были получены следующие результаты.

1. Был найден качественный массив данных, содержащих большое количество факторов, влияющих на стоимость недвижимости.
2. Рассмотрены методы обработки данных и выбраны наиболее подходящие в контексте решаемой задачи.
3. Проведен дескриптивный анализ массива данных, из которого можно понять, во-первых, что все рассмотренные факторы напрямую влияют на стоимость квартиры, а во-вторых, с помощью приведенного анализа можно вывести примерную область цен на желаемую квартиру. В дальнейшем представляет интерес применение методов машинного обучения для построения предсказателя на основе выделенных факторов.

Библиографический список

1. Профессиональный информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных. URL: <http://www.machinelearning.ru/> (дата обращения: 13.04.2020).
2. Асаул А. Н., Князь И. П., Коротаяева Ю. В. Теория и практика принятия решений по выходу организаций из кризиса / АНО «ИПЭВ». 2007. 224 с.
3. Система организации конкурсов по исследованию данных. URL: <https://www.kaggle.com/> (дата обращения: 01.04.2020).
4. UK Housing prices paid. URL: <https://www.kaggle.com/hm-land-registry/uk-housing-prices-paid> (дата обращения: 05.04.2020).
5. Flats to rent at Budapest. URL: <https://www.kaggle.com/gaborvecsei/flats-to-rent-at-budapest> (дата обращения: 05.04.2020).
6. Price of flats in Moscow. URL: <https://www.kaggle.com/hugoncosta/price-of-flats-in-moscow> (дата обращения: 05.04.2020).
7. Открытый курс машинного обучения. URL: <https://habr.com/ru/company/ods/blog/323210/> (дата обращения: 05.04.2020).
8. BOX PLOT. URL: https://en.wikipedia.org/wiki/Box_plot (дата обращения: 13.04.2020).

УДК 004.942

С. М. Герасимов

студент кафедры проблемно-ориентированных вычислительных комплексов

Е. П. Виноградова – старший преподаватель – научный руководитель

ОПРЕДЕЛЕНИЕ ДЕФЕКТА АКУСТИЧЕСКИМ МЕТОДОМ

Наиболее частыми авариями, происходящих на различных трубопроводах, являются образования течей. При этом может происходить как утечка газа, так и жидкости. Существуют различные методы для обнаружения течей. Применение течеискателей акустического типа является довольно распространенной практикой. Такие устройства позволяют довольно точно определить место возникновения протечки и при этом являются относительно недорогими.

Корреляционный течеискатель

Одним из типов акустических течеискателей является корреляционный. Работа данного устройства основывается на использовании двух датчиков вибрации (рис. 1) [1]. Помимо датчиков устройство также содержит блок автоматической обработки информации, в котором выполняется расчет расстояния от датчиков до места утечки.

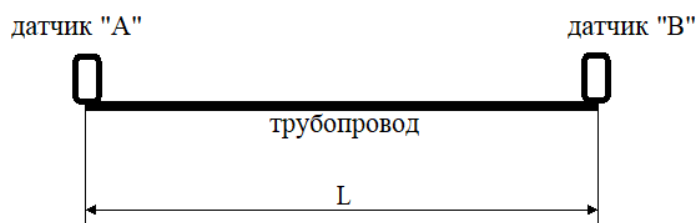


Рис. 1. Установка датчиков на участке трубопровода размером L для работы корреляционного течеискателя

Принцип работы устройства представлен на рис. 2. Воздействие на трубу приводит к возникновению вибрации, представляющей собой импульс, который регистрируют датчики «А» и «В».

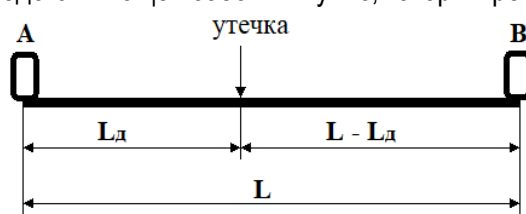


Рис. 2. Принцип работы корреляционного течеискателя

Регистрация импульсов происходит через определенные промежутки времени для каждого из датчиков. Это изображено на рис. 3.

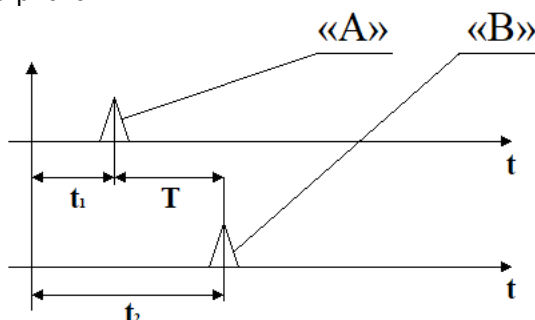


Рис. 3. Регистрация импульсов для датчиков «А» и «В»

Поскольку момент начала импульса неизвестен, необходимо узнать разность прихода волн. Для того, чтобы автоматизировать расчет данного показателя применяют процедуру взаимной корреляции.

Поскольку момент начала импульса неизвестен, необходимо узнать разность прихода волн. Для того, чтобы автоматизировать расчет данного показателя применяют взаимную корреляцию. Результат данной процедуры показан на рис. 4.

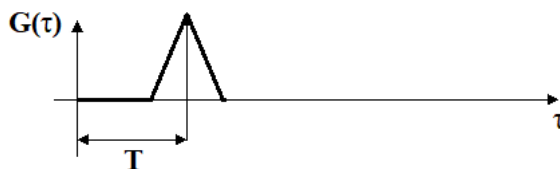


Рис. 4. График взаимной корреляции сигналов с датчиков «А» и «В»

Время прихода импульса на датчик «А» определяется по формуле (1)

$$t_1 = \frac{L_D}{V}, \quad (1)$$

где L_D – расстояние от датчика до места утечки; V – скорость распространения звука в исследуемой среде (жидкость/газ);

Время прихода импульса на датчик «В» определяется по формуле (2)

$$t_2 = \frac{L - L_D}{V}, \quad (2)$$

где L – длина исследуемого участка трубопровода.

Данные величины напрямую определить нельзя, для этого и требуется расчет разности прихода волн T (3):

$$T = t_2 - t_1, \quad (3)$$

что можно записать в виде (4):

$$T = \frac{L - 2L_D}{V}. \quad (4)$$

Исходя из этого, получаем итоговую формулу для расчета расстояния до места утечки (5):

$$L_D = \frac{L - T \cdot V}{2}. \quad (5)$$

Модель сигнала устройства в MathCAD

В рамках исследования корреляционных течеискателей, были проведены испытания на учебном стенде, позволяющем проверить работоспособность рассматриваемого метода. В качестве выходных данных выступают отсчеты взаимнокорреляционной функции сигналов с датчиков.

Данные записываются в текстовый файл и для дальнейшей работы необходимо произвести импорт в MathCAD [2]. Процедура представлена на рис. 5.

График взаимнокорреляционного сигнала в различных масштабах представлен на рис. 6.

В реальных условиях форма импульса не является идеально треугольной. К тому же, могут возникать эффекты реверберации, как это показано на рис. 6. В связи с этим возникает вопрос о

правильном выборе максимума, который характеризует место утечки (рис. 7). Данный вопрос содержит в себе множество аспектов и его решение будет осуществляться в дальнейших работах по исследуемой теме.

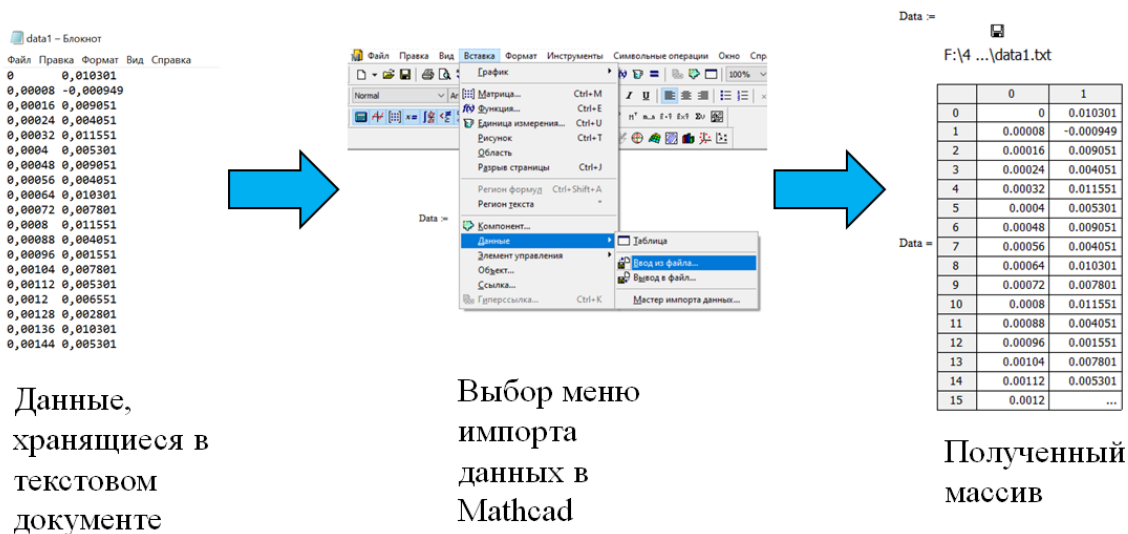


Рис. 5. Импорт данных в MathCAD

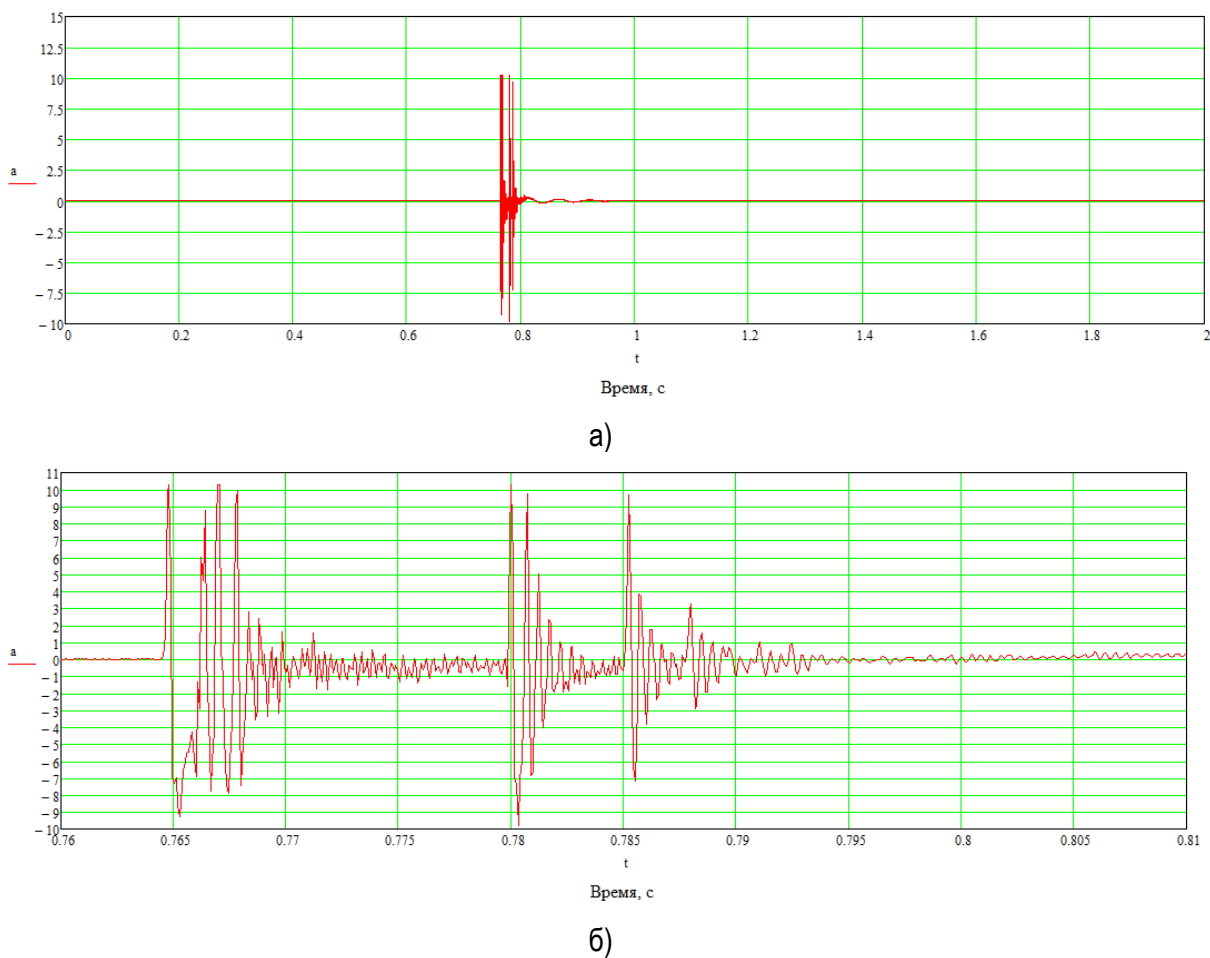


Рис. 6. График взаимнокорреляционного сигнала:
а) общий вид; б) увеличенный масштаб в области вибрации

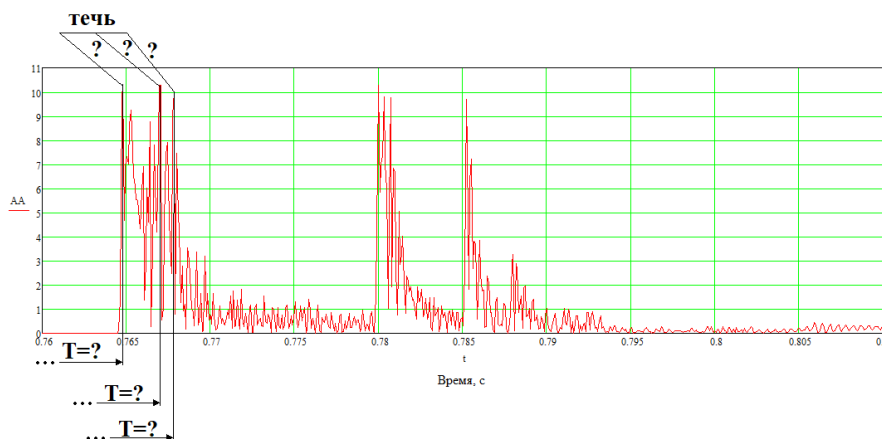


Рис. 7. График иллюстрирующий проблему выбора максимума, характеризующего место утечки

На текущем этапе в качестве возможной точки локализации утечки, определяющей разность прихода волн T , выбирается самый первый максимум.

Расчет расстояний в MathCAD на основе полученных данных, представлен на рис. 8.

$$hi := \text{match}(\max(a), a) = \begin{pmatrix} 9560 \\ 9587 \\ 9588 \\ 9750 \end{pmatrix} \quad \text{Нахождение пикового значения, соответствующего локализации течи}$$

$$high := hi_0$$

$$high = 9560$$

$$T := t_{(high)} = 0.007648 \quad \text{Время задержки звука относительно момента удара}$$

Из расчета, что при 20 °С., скорость звука в пресной воде равна 1484м/с

$$V := 1484$$

$$Ld := T \cdot V = 11.349632 \quad \text{Полученное расстояние между местом удара и местом крепления микрофона}$$

Рис. 8. Расчет расстояния между местом удара и местом крепления микрофона, полученный в MathCAD

Также, помимо выбора пиковых значений, необходимо устанавливать порог, который характеризует наличие утечки, как таковой, во избежание ложных результатов, что представлено на рис. 9.

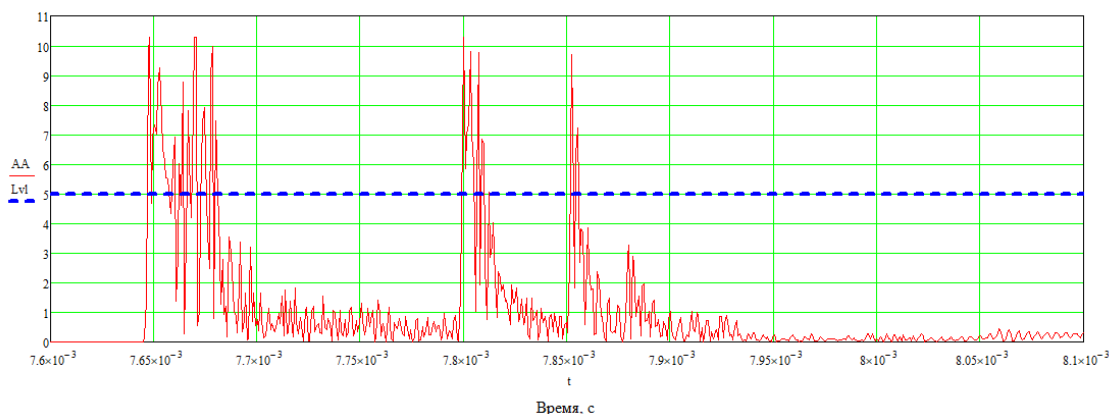


Рис. 9. Добавление порогового значения

Процедура принятия решений при разных значениях порога представлена на рис. 10.

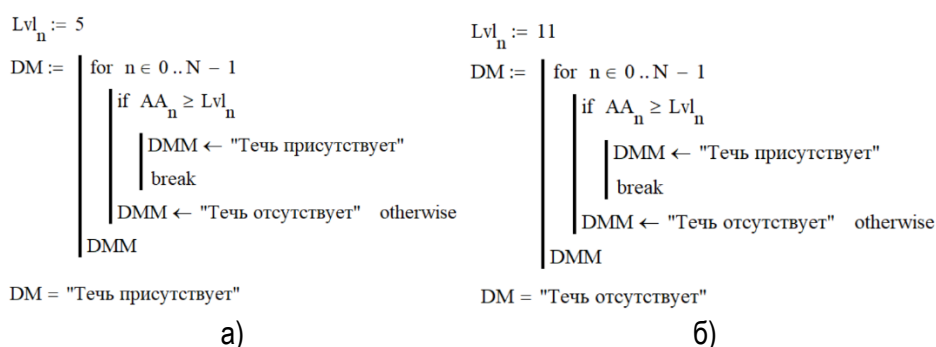


Рис. 10. Процедура принятия решений при пороговых значениях: а) 5; б) 11

Заключение

В ходе проведенных исследований были решены следующие задачи:

- исследованы модели приборов для акустического метода контроля и поиска течей в трубопроводах;
- разобран принцип работы корреляционных течеискателей;
- проведена работа со стендом, имитирующим трубопровод с наличием дефекта, к которому подключен датчик звука;
- данные, полученные при работе со стендом, были обработаны в среде MathCAD для расчета расстояния между местом дефекта и датчиком.

Библиографический список

1. Корреляционный течеискатель Т10: рук. по эксплуатации. URL: <http://www.svpribor.ru/docs/5cad6fdb.pdf> (дата обращения: 19.03.20).
2. Кирьянов Д. В. Самоучитель Mathcad 13. СПб.: БХВ, 2006. 528 с.

УДК 004.522

С. М. Герасимов

студент кафедры проблемно-ориентированных вычислительных комплексов

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

ИССЛЕДОВАНИЕ МЕТОДОВ АНАЛИЗА РЕЧЕВЫХ СИГНАЛОВ

Существует различные методы анализа речевых сигналов как во временной, так и в частотной областях. Когда необходимо обрабатывать последовательности слов, предложения, то обычно используются комплексные методы, сочетающие в себе скрытые марковские модели либо нейросети для прогнозирования наиболее вероятных звуков в последовательностях [1]. При обработке одного слова можно использовать более простые методы, основанные на спектральном анализе или сопоставлении временных рядов.

Целью данной работы является исследование различных методов анализа речевых сигналов для разработки алгоритма классификации голосовых команд, состоящих из одного слова. Будут рассмотрены следующие методы:

- построение и анализ спектрограмм;
- применение мел-частотных кепстральных коэффициентов;
- алгоритм динамической трансформации временной шкалы.

В качестве объекта исследования рассматривается набор из 4 предварительно записанных команд:

- две команды «включить» с одинаковым произношением;
- команда «выключить»;
- команда «закрыть».

Запись команд производится одним диктором.

Для моделирования различных методов анализа речевых сигналов было принято решение использовать компьютерный пакет Mathcad. Данная среда хорошо позволяет визуализировать результаты математического моделирования, позволяет работать с форматом WAV аудио-данных, содержит встроенные функции для обработки сигналов и позволяет быстро моделировать различные типы графиков.

Для того, чтобы работать с отсчётами звукового сигнала, необходимо сначала записать его и конвертировать его в удобную для обработки форму. Файл в формате WAV обычно содержит в себе несжатые отчёты звука в импульсно-кодовой модуляции.

На рис. 1 представлен график речевого сигнала слова «включить» в среде MathCAD.

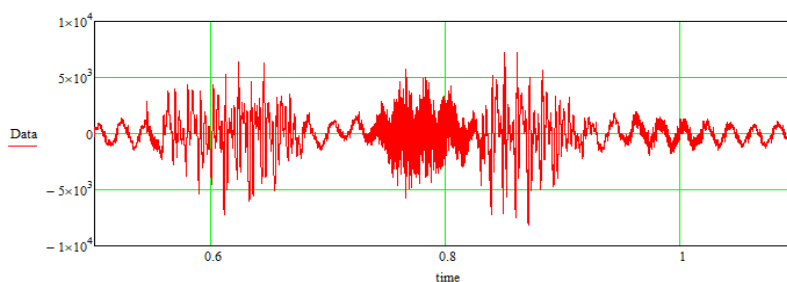


Рис. 1. График сигнала команды «включить» в среде MathCAD

Применение спектрограмм

Первым исследуемым методом является получение спектрограммы, которая представляет собой график зависимости спектральной плотности мощности сигнала от времени. Для построения такой зависимости используется оконное преобразование Фурье. В результате чего получают промежутки с определёнными диапазонами частоты и времени характеризующиеся амплитудой сигнала на

этом участке. На графике амплитуда визуализируется при помощи яркости либо цвета (обычно чем ярче область, тем больше амплитуда) [2].

Все последовательности разбиваются на одинаковые по частотному диапазону отрезки. Полученные отрезки также разделяются на временные отрезки, которые различаются, поскольку длительность сигналов может быть разной. К разделённым фрагментам данных применяется быстрое преобразование Фурье (БПФ).

Итоговые результаты моделирования исследуемого набора команд представлены в виде ряда спектрограмм, изображенных на рис. 2.

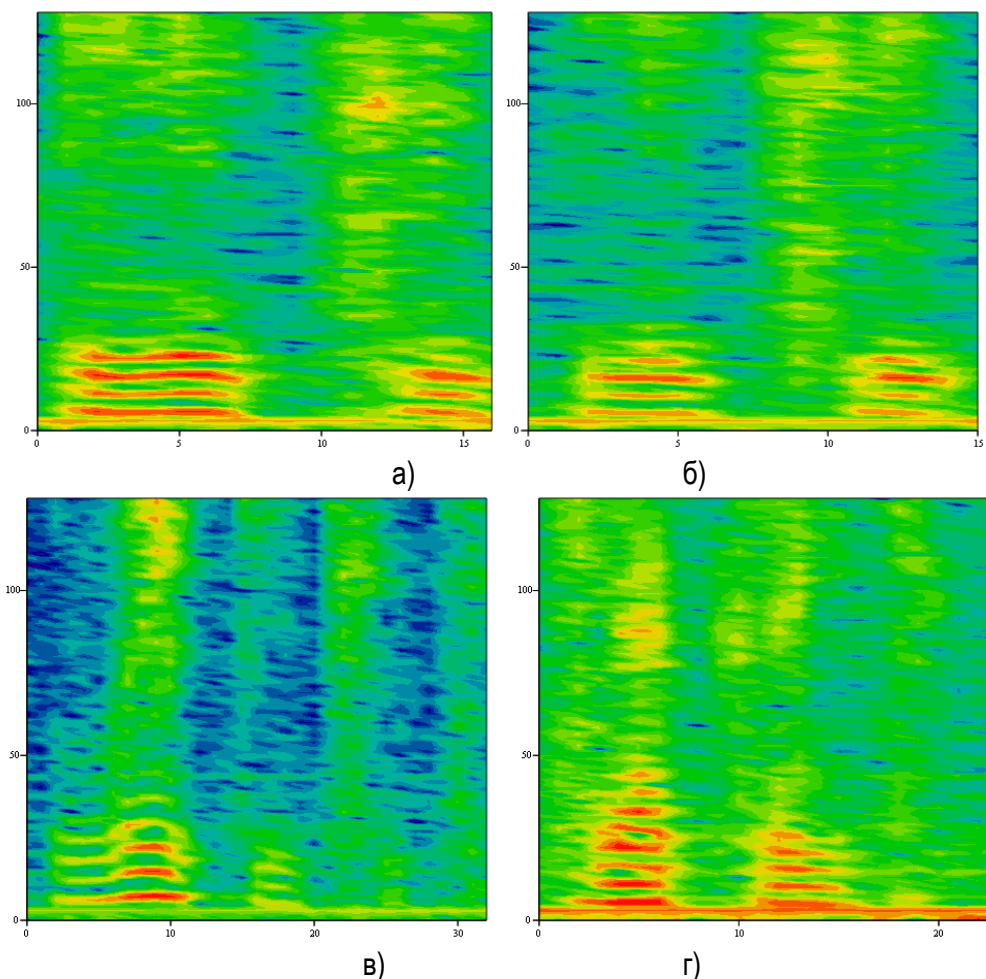


Рис. 2. Спектрограммы различных голосовых команд:
 а – б) команда «включить» произнесённая с разной интонацией; в) команда «выключить»;
 г) команда «закрыть»

Как видно из рисунка 2, спектрограммы (а – б) практически полностью совпадают по своему частотному составу, хотя имеют разные временные промежутки. Такое сравнение возможно благодаря тому, что частотный диапазон поделён на одинаковые отрезки для всех последовательностей. Несмотря на то, что слово «выключить» состоит почти из тех же звуков, её спектрограмма значительно отличается от спектрограмм команд «включить». Команда «Закрыть» обладает высокочастотной составляющей, которая не свойственна всем остальным представленным аудио-выборкам. Это можно объяснить наличием звонкой согласной в составе слова.

Вопреки эффективности визуальных результатов данного метода, его реализация в виде алгоритма распознавания голосовых команд будет затруднительна ввиду того, что отсчеты спектрограмм представляют собой двумерные массивы, размерность которых определяется количеством выборки исходных сигналов. Для того, чтобы сравнивать данные массивы между собой необходимо будет записывать команды с одинаковым промежутком времени и частотой дискретизации, что может негативно сказаться при обработке в режиме реального времени. Также необходимо сопоставлять

начальные отсчеты речевого сигнала с другой сравниваемой последовательностью. Это значительно усложняет реализацию алгоритма и делает его неэффективным.

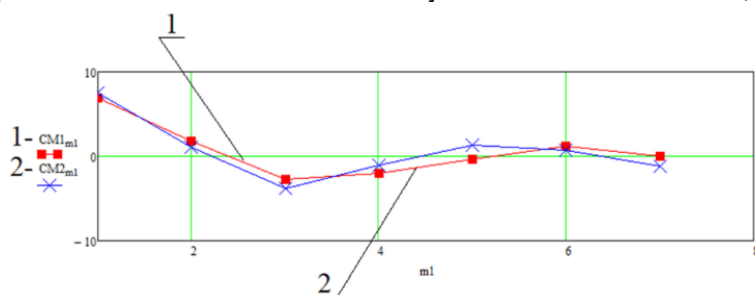
Мел-кепстральные коэффициенты

Мел-кепстральные коэффициенты (MFCC) представляют собой относительно небольшой набор значений, которые характеризуют частотные свойства сигнала.

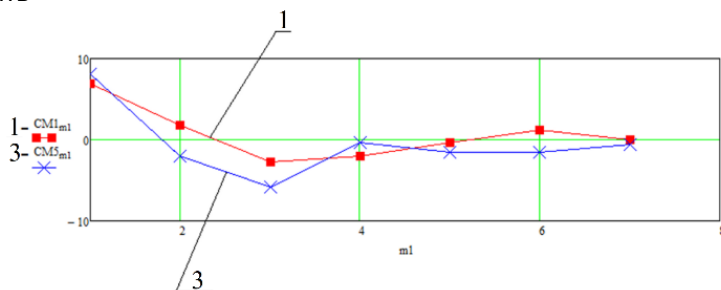
Данный метод основывается на том, что частотные полосы распределяются равномерно по логарифмической шкале, которая характерна для звуковосприятия человека [3]. Получаемую при преобразовании величину принято называть мел.

Как и при построении спектрограммы в данном методе тоже используется оконное преобразование Фурье, но для голосовой команды, представленной одним словом, преобразование можно брать по всему объему выборки. Полученные спектральные значения проходят через треугольные оконные фильтры, а затем к полученным значениям в каждом окне применяется дискретное косинусное преобразование. Амплитуды результирующего спектра представляют собой итоговый результат – мел-частотные кепстральные коэффициенты.

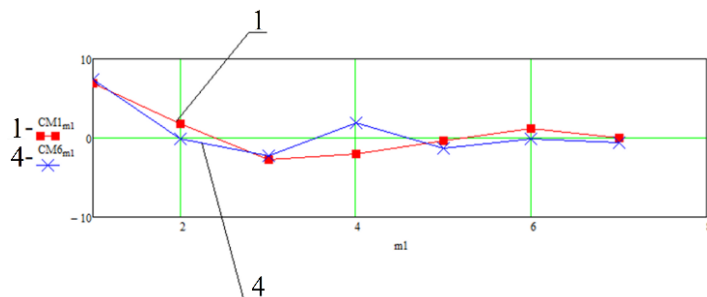
Итоговые результаты моделирования исследуемого набора команд представлены в виде последовательностей MFC коэффициентов, изображенных рисунке 3. Одна из последовательностей, полученная при обработке слова «включить» используется в качестве эталона для сравнения.



1 – команда «включить»,
2 – команда «включить»



1 – команда «включить»,
3 – команда «выключить»



1 – команда «включить»,
4 – команда «заккрыть»

Рис. 3. Сравнение MFCC последовательностей голосовых команд

Как видно из рисунка 4, MFCC голосовых команд «включить» (1 и 2) очень близки по значениям. Остальные голосовые команды сильно различаются между собой по коэффициентам.

Алгоритм динамической трансформации временной шкалы

Одной из простых мер для расчета расстояний между последовательностями является евклидова метрика. Расстояние в таком пространстве определяется по формуле (1):

$$d(p, q) = \sqrt{\sum_{k=1}^n (p_k - q_k)^2}, \quad (1)$$

где p и q – последовательности в евклидовом пространстве.

Данная мера не может быть использована при обработке речевых сигналов. Одно и то же слово нельзя произнести одинаково – всегда будут происходить какие-то смещения по временной и частотной шкале, даже без учета помех. Dynamic Time Warping (DTW) алгоритм разрабатывался с целью установления соответствия последовательностей по временной шкале [4]. Сопоставление временных рядов евклидовой метрикой и DTW алгоритмом представлено на рис. 4.



Рис. 4. Сопоставление временных рядов евклидовой метрикой и DTW алгоритмом

На первом этапе построения алгоритма рассчитывается матрица расстояний между точками обеих последовательностей. Далее начинается главный этап, который состоит в нахождении минимального пути деформации. Процедура обработки записывает индексы матрицы, которые будут соответствовать минимальным деформациям на пути от начального значения до конечного. При этом шаг процедуры всегда должен идти «вверх» и/или «вправо» по значениям массива. Подставляя полученные номера отсчетов вместо исходных устанавливается соответствие по времени. Если обрабатываемые слова были одинаковыми, то выборки сигналов будут сходиться.

Полученные результаты моделирования DTW алгоритма представлены на рисунках 5-7. Одна из последовательностей, полученная при обработке слова «включить» используется в качестве эталона для сравнения.

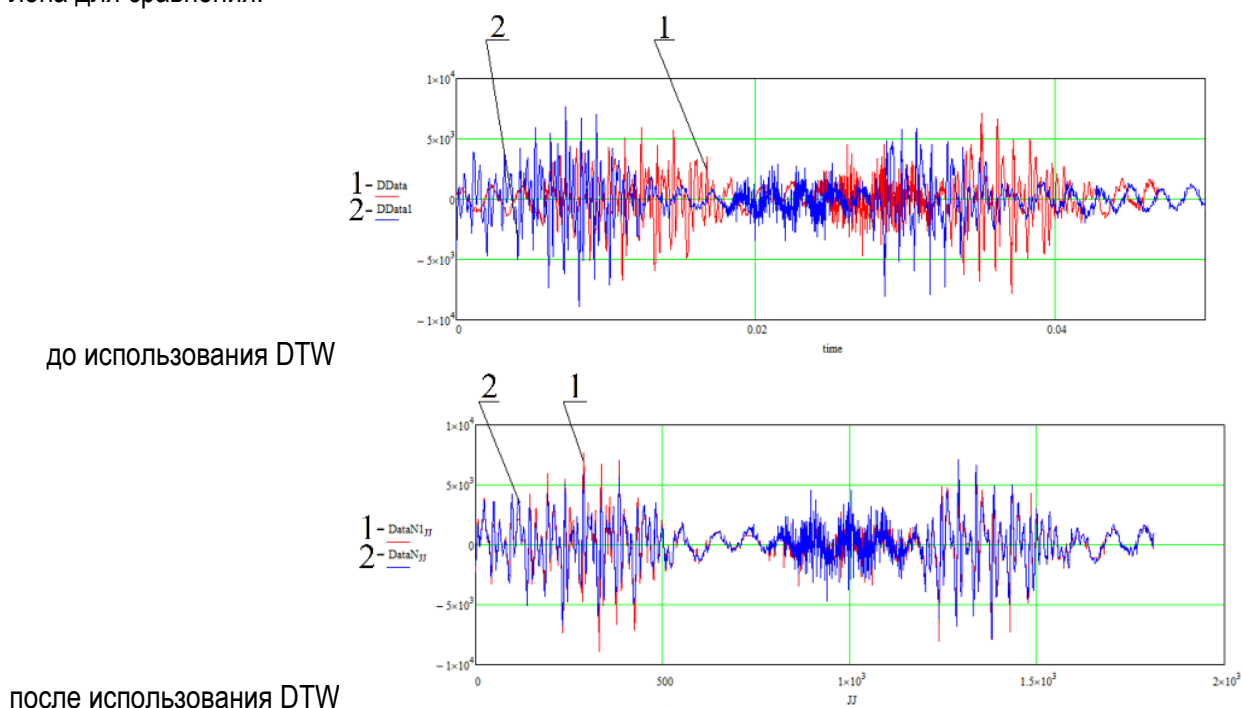
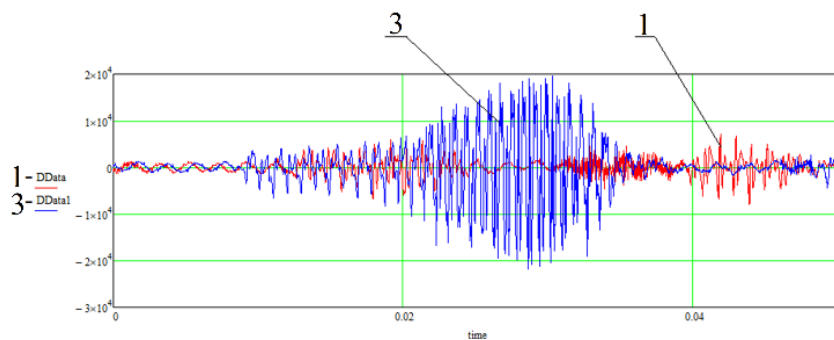
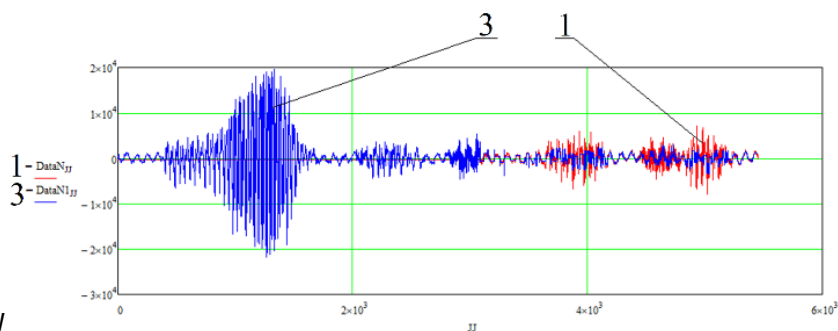


Рис. 5. Сопоставление речевых сигналов до и после использования DTW алгоритма команд:

1 – «включить»; 2 – «включить»

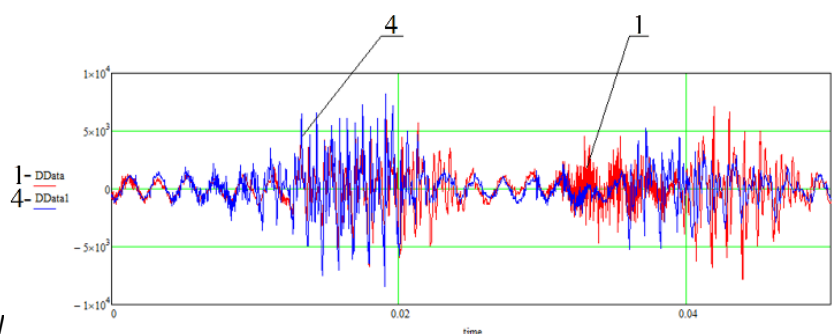


до использования DTW

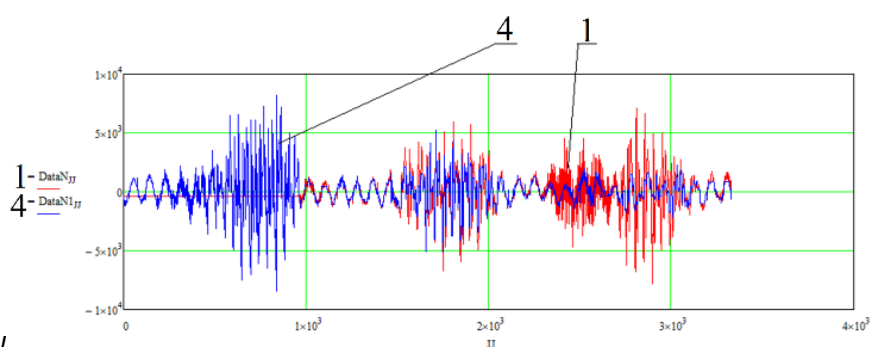


после использования DTW

Рис. 6. Сопоставление речевых сигналов до и после использования DTW алгоритма команд: 1 – «включить»; 3 – «выключить»



до использования DTW



после использования DTW

Рис. 7. Сопоставление речевых сигналов до и после использования DTW алгоритма команд: 1 – «включить»; 4 – «закрывать»

Если речевые сигналы соответствуют одному и тому же слову, то после применения алгоритма динамической трансформации временной шкалы данные последовательности практически совпадают, имея некоторые отличия по уровню сигнала и времени (рис. 6). При сопоставлении разных слов алгоритм может найти похожие фрагменты, но в целом последовательности не совпадают по уровню сигнала и обладают вырождающимися в прямую частями (рис. 7, 8).

Заключение

Все представленные в данной статье методы в той или иной степени справились с задачей сопоставления различных речевых сигналов.

Метод построения спектрограмм является наиболее быстрым в плане выполнения вычислительных операций и при правильной настройке временных и частотных границ хорошо визуализирует полученные результаты. Но при рассмотренной реализации данного метода невозможно численное сравнение.

Метод расчета MFCC требует чуть больше времени на обработку, в сравнении со спектрограммой, поскольку в нем реализуется быстрое преобразование Фурье, а также дискретное косинусное преобразование. Данный метод позволяет выделять основные частотные свойства слова для сличения при небольшом наборе коэффициентов, что даёт возможность эффективно сравнивать голосовые команды, представленные речевыми сигналами.

DTW алгоритм позволяет осуществлять обработку сигнала во временной, а не частотной шкале. Существенным недостатком является то, что из представленных методов данный обрабатывается дольше всех. При таком способе сравнения последовательностей хорошо заметны различия сигналов, так как при их несоответствии они не совпадают по уровню.

Наиболее быстрым и результативным методом для решения задачи распознавания является MFCC. Применение таких коэффициентов позволяет упростить дальнейший процесс обработки, за счет того, что эталонные записи команд хранятся в виде небольшого набора значений и занимают меньше памяти в устройстве, чем у других рассмотренных подходов. При этом сам процесс распознавания сводится к сравнению значений коэффициентов наборов и дальнейшему решению на основе заданного порогового значения.

Библиографический список

1. *Lawrence R. Rabiner*. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition // Proceedings of the IEEE. 1989. Vol. 77, № 2.
2. *Трахтман А. М., Трахтман В. А.* Основы теории дискретных сигналов на конечных интервалах. М.: Сов. радио, 1975. 208 с.
3. *Tyagi V., Wellekens C.* On desensitizing the Mel-cepstrum to spurious spectral components for robust speech recognition // Proceedings (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. P. 1–21.
4. *Lawrence R. Rabiner, Aaron G. Rosenberg, Stephen E. Levinson* Considerations in Dynamic Time Warping for Discrete Word Recognition/ *Lawrence R. Rabiner* // IEEE Transactions on Acoustics Speech and Signal Processing. 1989. Jan. P. 575–582.

УДК 004.056.53

Б. В. Давидович

студент кафедры проблемно-ориентированных вычислительных комплексов

Б. Р. Хасанов

студент кафедры проблемно-ориентированных вычислительных комплексов

О. А. Кононов – кандидат технических наук, доцент – научный руководитель

ПОВЫШЕНИЕ УРОВНЯ ЗАЩИТЫ БЕСПРОВОДНОЙ ПЕРЕДАЧИ ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ БИОМЕТРИЧЕСКИХ ДАННЫХ

Введение

В настоящее время развитие инфокоммуникационных технологий происходит настолько активно, что делать прогнозы о том, какими будут технологические решения далее, очень сложно. Одним из факторов, давшим начало технологической гонке, стало появление Интернета. С целью упрощения доступа к интернету была разработана технология беспроводной локальной сети на основе стандартов *IEEE 802.11*, имеющая логотип «*Wi-fi*».

Существующие способы защиты *Wi-fi* сетей

Основные известные или используемые способы защиты следующие.

WEP-шифрование (*Wired Equivalent Privacy*)

В данном типе шифрования используется симметричный потоковый шифр *RC4*. Сегодня эти протоколы уже не являются криптостойкими. У *WEP* есть два основных протокола – 40- и 104-битный. Главным недостатком такого типа шифрования является необходимость введения пользователем пароля, длина которого невелика, недостаточна для надежности и легко взламывается подбором [1].

TKIP-шифрование (*Temporal Key Integrity Protocol*)

Протокол *TKIP* представляет собой усовершенствованный *WEP*-протокол. В данный протокол внесены изменения, которые позволяют успешно противодействовать атакам типа *replay* (повторное использование ключей) и *forgery* (изменение содержимого передаваемых пакетов). Добавлен протокол *Message Integrity Check* для проверки целостности сообщений, который блокирует станцию на 60 секунд, если в течение 60 секунд были посланы два сообщения, не прошедшие проверку целостности [1].

SKIP-шифрование (*Cisco Key Integrity Protocol*)

SKIP протокол по своей сути похож на *TKIP*-протокол. Он также использует протокол проверки целостности сообщений, разница лишь в компании-разработчике. У *SKIP* протокола это *Cisco* [1].

WPA -шифрование

WPA – это следующая веха в развитии шифрования, вместо *RC4* используется алгоритм шифрования *AES*, также возможно использование протокола аутентификации *EAP*. Существует два режима работы: с использованием статического ключа *PSK*, при котором каждый узел вводит пароль для доступа *Personal*, и *Enterprise*, когда проверка осуществляется серверами *RADIUS* [1].

WPA2-шифрование (*IEEE 802.11i*)

В *WPA2* используются протоколы *CCMP RSN AES*. Для совместимости со старым оборудованием имеется поддержка *TKIP* и *EAP*. Как и в *WPA*, в *WPA2* есть два режима работы: *Pre-Shared Key* и *Enterprise* [1].

Экспериментальная проверка надежности средств защиты

Следует отметить, что все перечисленные выше протоколы безопасности не являются до конца надежными. Нами был проведен эксперимент по проверке безопасности *Wi-Fi* сети. Была создана тестовая сеть с наиболее совершенным на данный момент протоколом *WPA2* (рис. 1).

Затем путем несложных манипуляций в программе *AirSlax Pro 6.02* произвели сканирование сетей (рис. 2) [2].

После этого выбирали нашу сеть, чтобы продолжить проверку безопасности. Произвели отключение одного из абонентов, чтобы перехватить пароль при следующем подключении, а затем выбрали способ подбора пароля на *CPU*.

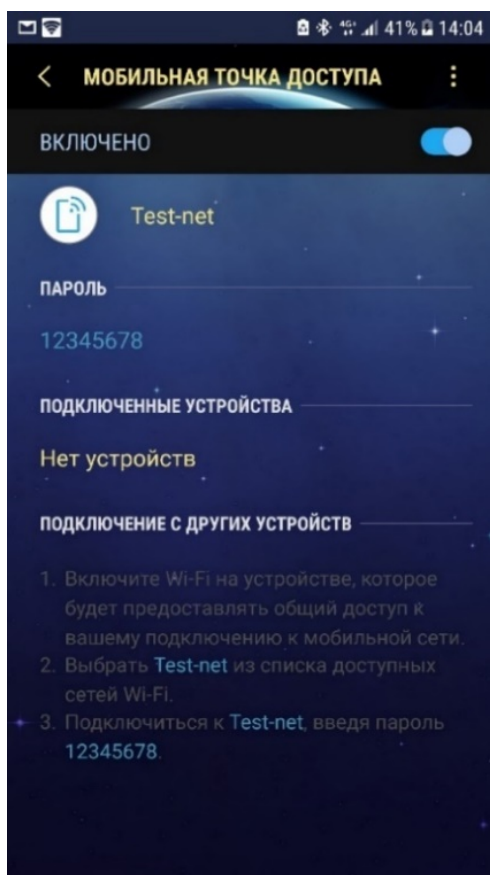


Рис. 1. Созданная сеть

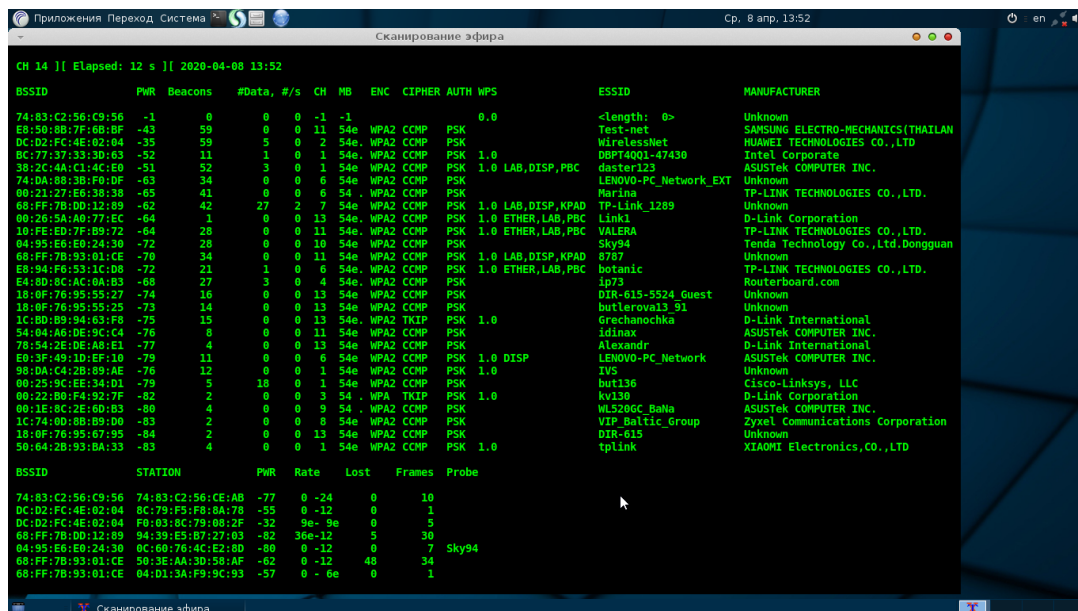


Рис. 2. Сканирование Wi-Fi сетей

```

Подбор пароля для: Testnet-E8508B7F6BBF
Aircrack-ng 1.2 rc4 r2915

[00:00:36] 11680/53104 keys tested (321.00 k/s)

Time left: 2 minutes, 9 seconds                21.99%

KEY FOUND! [ 12345678 ]

Master Key   : 1E 54 B2 33 51 D6 0E EE 07 65 DF D3 49 AA D7 EA
              D4 E1 7D A6 51 79 FD 39 47 84 7E C5 35 9B 58 9A

Transient Key : 34 E3 E8 C9 3B BE 91 92 A0 07 1E 6C 8A 75 26 4C
              4D 97 1F FE 78 9C 0E 61 D9 8E 9C F9 CD 30 5E 8C
              88 84 D5 25 1B 05 F2 40 55 89 DF 24 46 94 78 6C
              29 BA A9 A4 D1 13 75 27 09 A9 1B 09 7D 4A 10 2C

EAPOL HMAC   : 9E F5 42 82 D5 57 5E 78 D4 44 AC 43 E5 F2 53 14
    
```

Рис. 3. Результат работы программы

Результат работы программы оказался таковым: на подбор пароля на CPU затрачено 2 минуты 9 секунд. Подбор мог бы произойти быстрее в случае использования более производительного процессора.

Исходя из описанного опыта, можно сделать вывод, что сети Wi-Fi не являются достаточно надежными. Мы поставили перед собой задачу разработки более надежного метода защиты Wi-Fi сетей [2].

Предлагаемое решение

При помощи AVR-IoT WG development board, построенной на базе микроконтроллера ATmega4808 MCU, и биометрического датчика (оптический датчик отпечатков пальца) осуществляется дополнительная защита сети Wi-Fi. Делается это следующим образом.

При подключении к сети устройство запрашивает у DHCP-сервера IP-адрес, под которым его будет понимать микросхема аутентификации. На подключаемом устройстве при этом происходит переход на страницу авторизации, на которой пользователь должен предоставить биометрические данные.

Маршрутизатор сети просматривает свою базу в поисках предъявленного ему mac-адреса и биометрических данных. Далее возможны два сценария:

- 1) пользователь уже авторизован в сети, т.е. mac-адрес устройства уже есть в базе сети, и биометрические данные пользователя совпали с сохраненными в базе;
- 2) пользователь не авторизован в сети, т.е. вне зависимости от того есть его mac-адрес в базе или нет, и его биометрические данные не определились базой.

Если пользователь уже авторизован в сети, то ему присваивается ip-адрес в сети с доступом в интернет, а если нет, то доступ в сеть ему блокируется и запрос перенаправляется на страницу авторизации.

На данный момент времени нами осуществлено подключение биометрического датчика к микроконтроллеру Arduino UNO. Был выбран именно этот контроллер, так как он построен на базе микроконтроллера Atmega 328P, схожего по архитектуре с микроконтроллером, на базе которого произведена AVR-IoT WG development board [3].

Изображение макета устройства и результаты работы программы для вывода изображения отпечатка представлены на рис. 4 и 5.

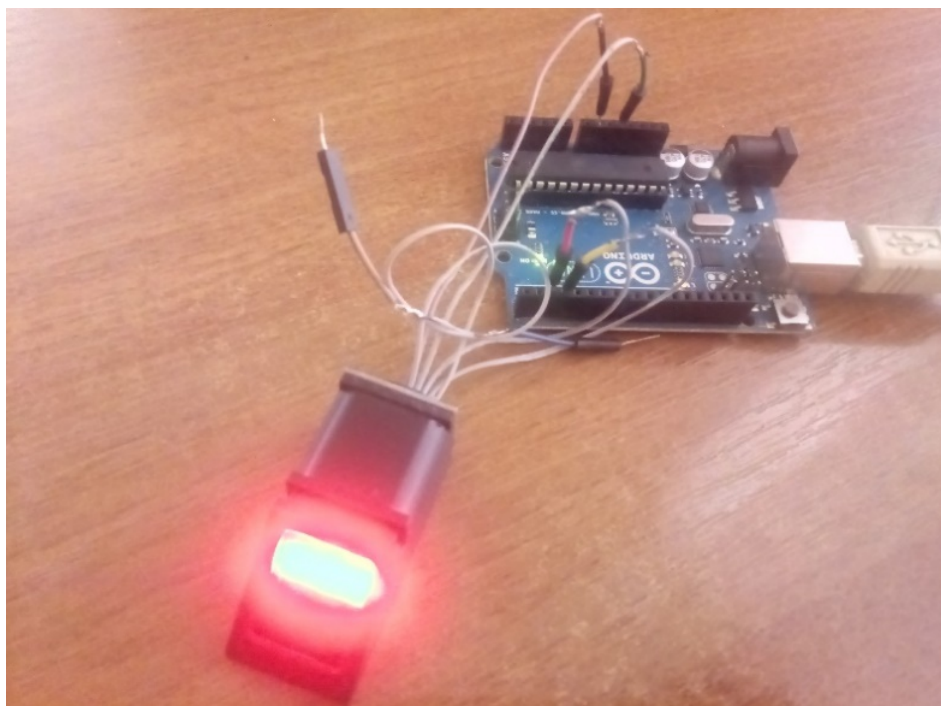


Рис. 4. Макет устройства

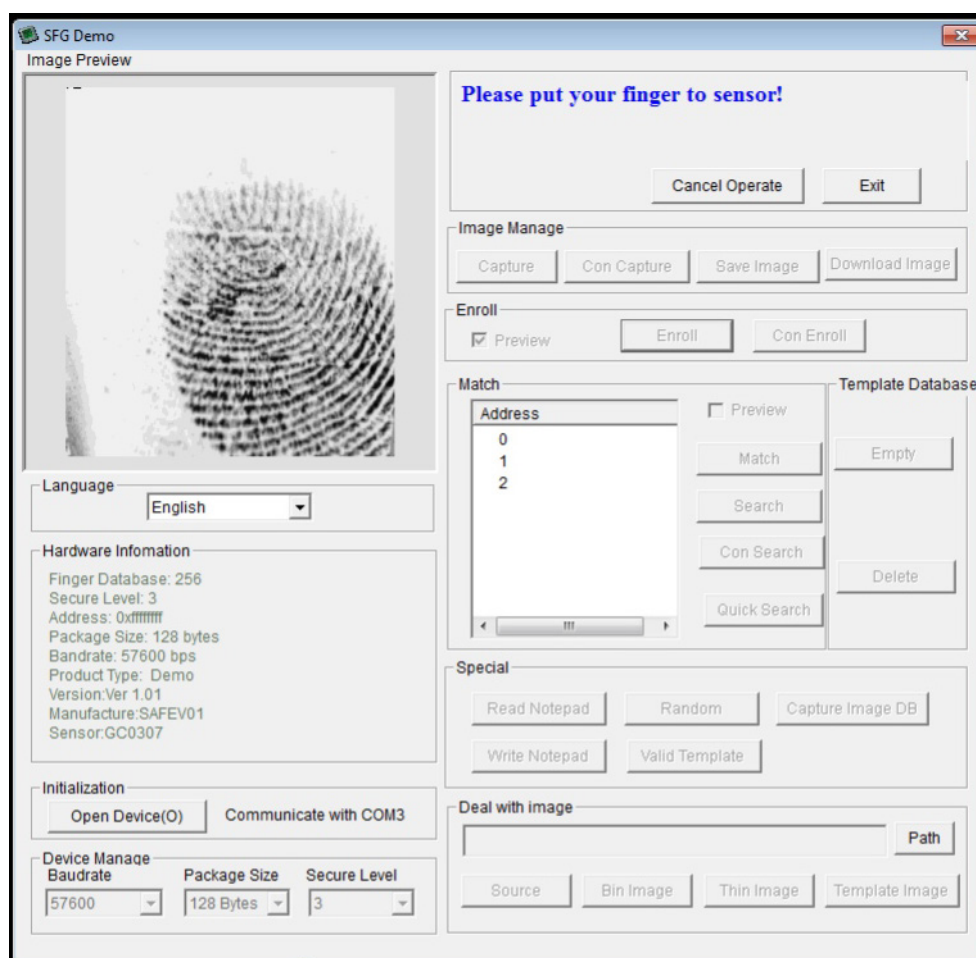


Рис. 5. Снятый отпечаток

На данном этапе нашей работы реализовано аппаратное и программное подключение датчика к микроконтроллеру, написана программа для поиска отпечатка в базе.

Заключение

Мы предполагаем, что благодаря разрабатываемому нами методу защиты сети *Wi-Fi*, данные пользователей будут защищены более надежно, чем сейчас. Следующим этапом нашей работы будет подключение биометрического датчика к *AVR-IoT WG development board*.

Библиографический список

1. Гордейчик С. В., Дубровин В. В. Безопасность беспроводных сетей. М.: Горячая линия – Телеком, 2008. 287 с.
2. Описание AirSlax Pro 6.02. URL: <https://manshet.org/1854-airslax-pro-602-repack-base-modules.html> (дата обращения: 16.03.2020).
3. Fingerprint sensor-scanner with Arduino. URL: <https://www.theorycircuit.com/fingerprint-sensor-scanner-arduino/> (дата обращения: 20.03.2020).

УДК 681.5.013

А. А. Заболотская

студент кафедры электропривода, автоматики и управления в технических системах

Е. М. Васильев – кандидат технических наук, доцент – научный руководитель

Воронежский государственный технический университет, г. Воронеж, Россия

СИНТЕЗ УПРУГОЙ ЭЛЕКТРОМЕХАНИЧЕСКОЙ СИСТЕМЫ С НЕПОЛНЫМ МОДАЛЬНЫМ УПРАВЛЕНИЕМ

1. Введение

Современным разделом теории автоматического регулирования является модальное управление. Суть модального управления состоит в том, что с помощью регулятора формируется желаемое положение корней характеристического полинома замкнутой системы [1-4].

В рамках методов модального управления можно выделить метод, называемый «неполное модальное управление», которому незаслуженно уделяется мало внимания.

Практика показывает, что реальные системы имеют систему корней, в которой доминирующими является только их небольшая часть. В связи с этим, можно предпринять попытку управления только доминирующими корнями, которые выбираются по заданным показателям качества системы. При этом предполагается, что недоминирующие корни будут располагаться на комплексной плоскости значительно левее, нежели управляемые нами корни. В этом случае, они не будут оказывать влияние на систему.

Отсюда вытекает возможность неполного управления системой корней объекта; часть корней остается свободной. Соответственно, размерность регулятора будет уменьшена. Кроме того, уменьшение числа используемых координат состояния позволит избавиться от части датчиков состояния системы.

В данной статье будет проведен теоретический анализ и представлен практический пример синтеза упругой электромеханической системы с неполным модальным управлением.

2. Синтез регулятора

Рассмотрим для определенности систему пятого порядка. Для полного модального управления необходимо на вход регулятора подавать информацию о пяти координатах состояния

$$x = [x_1 \ x_2 \ x_3 \ x_4 \ x_5]^T.$$

Коэффициенты полного регулятора $R = [r_1 \ r_2 \ r_3 \ r_4 \ r_5]$ определяются в результате приравнивания коэффициентов полиномов p_i , a_i ($i=1..5$) левой и правой частей уравнения:

$$|sE - B + NDRK| = a_5s^5 + a_4s^4 + a_3s^3 + a_2s^2 + a_1s + a_0, \quad (1)$$

в котором: B – характеристическая матрица; N – матрица управления; K – матрица датчиков; D – дополнительный коэффициент передачи; E – единичная матрица.

Поставим задачу неполного модального управления. Расположим желаемым образом три корня характеристического полинома $\lambda_1, \lambda_2, \lambda_3$, то есть часть из всех пяти корней. Тогда матрица регулятора и уравнение (1) примут вид:

$$R = [r_1 \ r_2 \ r_3 \ 0 \ 0],$$

$$|sE - B + ND \cdot [r_1 \ r_2 \ r_3 \ 0 \ 0]K| = a_5s^5 + a_4s^4 + a_3s^3 + a_2s^2 + a_1s + a_0. \quad (2)$$

Значения корней $\lambda_1, \lambda_2, \lambda_3$ не в состоянии определить данный характеристический полином, то есть коэффициенты a_0, a_1, a_2, a_3, a_4 являются неизвестными. В результате общее количество неизвестных равно восьми: $a_0, a_1, a_2, a_3, a_4, r_1, r_2, r_3$. Для определения неизвестных составим систему уравнений:

$$\begin{cases} p_i = a_i, i = 1 \dots 5; \\ a_5 \lambda_1^5 + a_4 \lambda_1^4 + a_3 \lambda_1^3 + a_2 \lambda_1^2 + a_1 \lambda_1 + a_0 = 0; \\ a_5 \lambda_2^5 + a_4 \lambda_2^4 + a_3 \lambda_2^3 + a_2 \lambda_2^2 + a_1 \lambda_2 + a_0 = 0; \\ a_5 \lambda_3^5 + a_4 \lambda_3^4 + a_3 \lambda_3^3 + a_2 \lambda_3^2 + a_1 \lambda_3 + a_0 = 0. \end{cases} \quad (3)$$

Таким образом, было получено восемь уравнений с восемью неизвестными. Решение этих уравнений позволяет определить все коэффициенты регулятора $R = [r_1 \ r_2 \ r_3 \ 0 \ 0]$. При этом необходимо проверять расположение неуправляемых корней.

Продемонстрируем реализацию неполного модального управления на конкретном примере.

3. Практическая проверка результатов синтеза

Рассмотрим упругую электромеханическую систему, структура которой показана на рис.1.

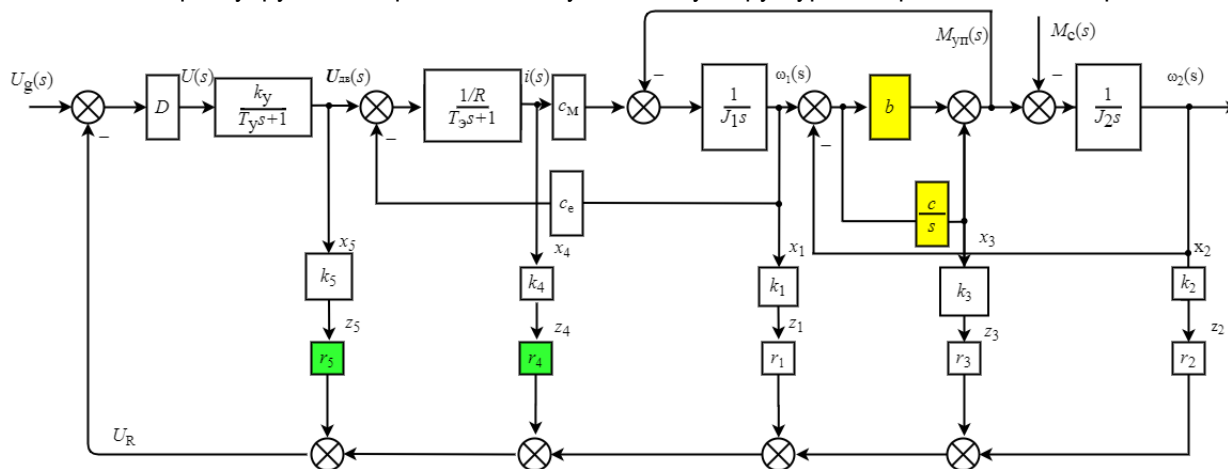


Рис. 1. Структура упругой электромеханической системы

На рис. 1 обозначены: $\omega_1(t)$, $\omega_2(t)$, J_1 , J_2 – частоты вращения и моменты инерции двигателя и рабочего механизма соответственно; $U(t)$, $U_{дв}(t)$ – напряжения, приложенные ко входу усилителя мощности и к цепи якоря двигателя постоянного тока соответственно; $M_{уп}(t)$, $M_c(t)$ – момент сил упругости и момент сопротивления рабочемеханизма; $i(t)$ – ток якоря; R , $L_я$ – активное сопротивление и индуктивность цепи якоря; c_e , c_M – постоянные двигателя для противоЭДС и для вращающего момента; c , b – коэффициенты жесткости внутреннего вязкого трения упругой связи; k_y , T_y – коэффициент передачи и постоянная времени усилителя мощности.

Описание объекта в пространстве координат x_1, x_2, x_3, x_4, x_5 :

$$B = \begin{bmatrix} \frac{b}{J_1} & \frac{b}{J_1} & -\frac{1}{J_1} & \frac{c_M}{J_1} & 0 \\ \frac{b}{J_2} & -\frac{b}{J_2} & \frac{b}{J_1} & 0 & 0 \\ c & -c & 0 & 0 & 0 \\ -\frac{c_e}{L_я} & 0 & 0 & -\frac{R}{L_я} & \frac{1}{L_я} \\ 0 & 0 & 0 & 0 & -\frac{1}{T_y} \end{bmatrix}; \quad K = \begin{bmatrix} k_1 & 0 & 0 & 0 & 0 \\ 0 & k_2 & 0 & 0 & 0 \\ 0 & 0 & k_3 & 0 & 0 \\ 0 & 0 & 0 & k_4 & 0 \\ 0 & 0 & 0 & 0 & k_5 \end{bmatrix};$$

$$N = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{k_y}{T_y} \end{bmatrix}^T \quad A = [0 \quad k_2 \quad 0 \quad 0 \quad 0].$$

Решение системы (3) представлено в таблице.

Результаты синтеза регуляторов

Объект или система	Корни характеристического полинома				
	$\lambda_1.$	$\lambda_2.$	$\lambda_3.$	$\lambda_4.$	$\lambda_5.$
Объект	-3333	-714	-0.31	-0.005+0.03j	-0.005-0.03j
Система с полным модальным управлением	-3333	-714	-0.31	-0.31	-0.31
Система с неполным модальным управлением	-3333	-714	-0.34	-0.31	-0.28

Коэффициенты полного регулятора $R = [r_1 \ r_2 \ r_3 \ r_4 \ r_5]$: $r_1=0.0012$; $r_2= 0.0512$; $r_3=171.4$; $r_4=0.00005$; $r_5=0.00015$.

Коэффициенты неполного регулятора $R = [r_1 \ r_2 \ r_3 \ 0 \ 0]$: $r_1=0.0012$; $r_2=0.0506$; $r_3=171$.

На рис. 2 показаны переходные процессы на выходе системы без регулятора и с модальными регуляторами.

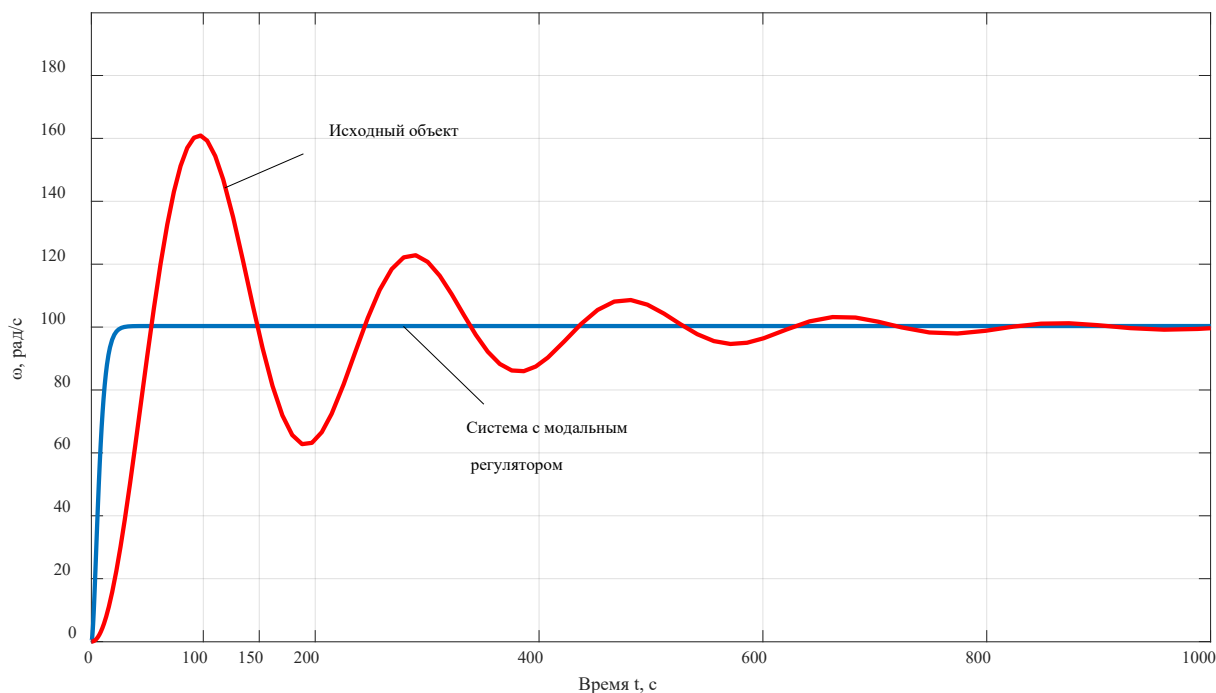


Рис. 2. Переходные процессы на выходе объекта и замкнутой системы

Переходные процессы с неполным и полным модальным управлением совпали.

Рис. 2 подтверждает подавление колебаний в упругой системе с помощью неполного модального управления. При этом в системе с неполным регулятором датчики k_4, k_5 исключены, (на рис.2 они выделены цветом).

Представленные графики переходного процесса не учитывают того обстоятельства, что ток двигателя ограничен. Проверим работу неполного модального регулятора при ограничении тока значением $i_{max}=200$ А.

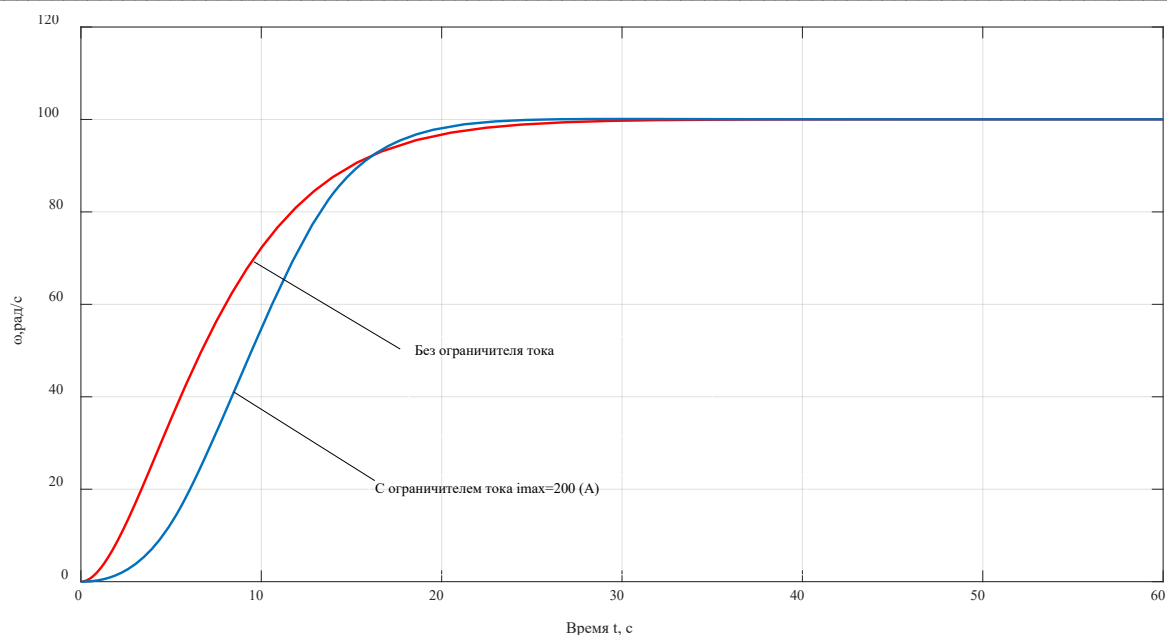


Рис. 3. Переходные процессы в системе с учетом ограничения тока

Работоспособность регулятора сохранилась в условиях ограниченного тока.

Таким образом, проверка результатов синтеза подтвердила эффективность неполного модального управления в системах высокого порядка.

4. Заключение

Изложенная в работе методика синтеза регулятора систем с неполным модальным управлением позволяет проектировать указанные системы с показателями качества, не уступающими системам с полным модальным управлением. При этом достигается существенное сокращение числа датчиков в системе и снижение размерности регуляторов.

Библиографический список

1. Борковская И. М., Пыжкова О. Н. Построение регуляторов по типу обратной связи для решения задач стабилизации и управляемости динамических систем // Труды Белорус. технолог. ун-та. Сер. 3: Физико-математические науки и информатика. 2019. № 2. С. 5–12.
2. Belyaev A. K., Polyanskiy V. A., Smirnova N. A., Fedotov A. V. Identification procedure in the modal control of a distributed elastic system // St. Petersburg Polytechnical University Journal: Physics and Mathematics. 2017. Vol. 3, № 2. P. 135–143.
3. Gabr W. I., Dorrah H. T., Elsayed M. S. A new symbolic-based continuous (infinite) modal approach for systems control and operation using computational mathematics // Ain Shams Engineering Journal. 2019. Vol. 10, № 4. URL: <https://doi.org/10.1016/j.asej.2019.11.001> (дата обращения: 15.03.2020).
4. Yang C., Li J. Z. L., Agrawal S. K. Design and implementation of a novel modal space active force control concept for spatial multi-DOF parallel robotic manipulators actuated by electrical actuators // ISA Transactions. 2018. Vol. 72. P. 273–286.

УДК 004.051

Е. Д. Зорина

студент кафедры проблемно-ориентированных вычислительных комплексов

В. С. Павлов – доктор технических наук, профессор – научный руководитель**АНАЛИЗ ПРОГРАММНЫХ ШАБЛОНОВ РЕАЛИЗАЦИИ ИНТЕРАКТИВНЫХ ВЕБ-ПРИЛОЖЕНИЙ**

На сегодняшний день в сети Интернет имеется большое количество ресурсов с постоянно обновляющимся контентом, что требует от разработчиков быстро и качественно создавать новые проекты. Ускорение процесса разработки веб-приложений без ущерба для их надежности и безопасности представляет собой насущную и своевременную задачу веб-программирования. Использование программного шаблона (фреймворка), выступающего в качестве каркаса для создания приложений, обеспечивает централизацию управления, реализацию единого стандарта разработки, упрощение развертывания, что в значительной степени сокращает временные затраты на создание и поддержку веб-приложения. Поэтому исследование программных шаблонов для создания веб-приложений является актуальным вопросом сегодняшнего дня.

Множественность имеющихся программных платформ, а также типов интерактивных веб-приложений, вносит неопределенность при решении конкретных практических задач веб-разработки. При этом возникает задача поиска наиболее эффективного технического решения, решение которой рассматривается в настоящей статье.

Цель работы — обоснование выбора программных шаблонов для наиболее распространенных типов интерактивных веб-приложений.

Стремительное развитие JavaScript привело к большому развитию frontend-разработки (frontend — уровень представления клиент-серверного приложения). За несколько лет появились несомненные лидеры в frontend-разработке: React, Angular и Vue.js. Далее будут сравниваться эти фреймворки [1].

Фреймворки Javascript в первую очередь интересно сравнить по показателям рендеринга (отображения конечного результата) страницы.

Основой фреймворков Javascript является объектная модель html-документа DOM (Document Object Model), позволяющая считывать и менять содержимое, оформление и структуру html-документов. Каждый из фреймворков подходит по-своему к обработке DOM что и оказывает влияние на рендеринг конечной страницы, отображаемой на экране пользователя.

Фреймворки Vue.js и React создают копию DOM, обрабатывают ее, а затем результат сравнивается с исходной версией документа. На экране пользователя заменяются только те части страницы, которые отличаются от результатов обработки. Это значительно ускоряет загрузку и рендеринг страницы. Соответственно сокращается объем трафика, что особенно важно для пользователей мобильных устройств.

В корне отличается подход к обработке DOM фреймворком Angular. Здесь происходит разделение на два потока, причем рендеринг DOM осуществляется браузером (клиентской частью), а создание директив, загрузка кода и сервисов – общим потоком (серверной частью). Но это совсем не значит, что рендеринг происходит на стороне клиента – визуализация по-прежнему производится серверами. Следовательно, SEO-оптимизация [2] не вызовет затруднений, а поисковым роботам будет предоставлена корректная страница при индексации.

В версии 9 Angular представлен новый движок рендеринга Angular Ivy, кардинально отличающийся от всех аналогичных технологий популярных фреймворков тем, что он использует Incremental DOM.

Incremental DOM можно трактовать как библиотеку для выражения и применения обновлений деревьев DOM [3].

Наряду с этим, Virtual DOM — это концепция программирования, в которой идеальное или виртуальное представление пользовательского интерфейса хранится в памяти и синхронизируется с клиентским DOM [4].

Рассмотрим важные различия Incremental DOM от Virtual DOM [5].

1. Принцип работы. Базовая концепция Virtual DOM используется для фреймворка React. Основная идея состоит в том, что каждый компонент создаёт новое VDOM-дерево (представление HTML-документа в виде дерева тегов) всякий раз, когда он компилируется. Фреймворк React сравнивает новое дерево с предыдущим (рис. 1), после чего вносит набор изменений в DOM браузера (блоки под номерами 4 и 6 на рис. 1), чтобы привести его в соответствие с новым VDOM-деревом.

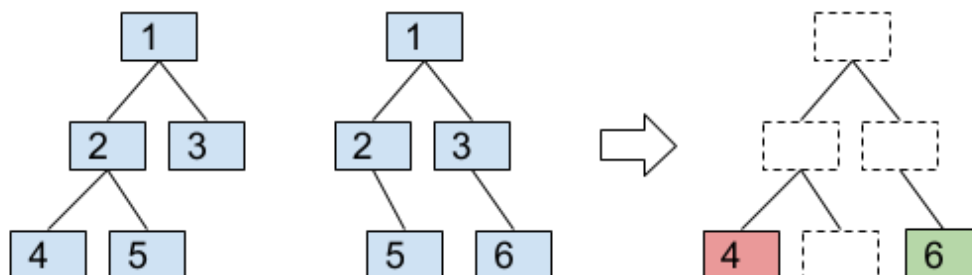


Рис. 1. Принцип работы Virtual DOM

Virtual DOM имеет два основных преимущества:

- Возможность использования любого языка программирования для реализации функции рендеринга компонента и отсутствие необходимости в компиляции.
- В результате рендеринга компонента мы получаем значение, что может пригодиться при тестировании, отладке и т.п.

Incremental DOM используется компанией Google для внутренних нужд. Его основная идея заключается в том, что каждый компонент компилируется в набор инструкций, которые создают DOM-деревья и непосредственно обновляют их при изменении данных. В Google решили предпочесть Incremental DOM (взамен Virtual DOM) для достижения приложениями мобильных устройств хорошей производительности. А значит, была необходима оптимизация размера bundle (совокупность программных данных/файлов, объединенных по какому-либо признаку) и объёма потребляемой памяти.

2. Работа с избыточным программным кодом. Известен метод оптимизации программных библиотек «Встряхивание дерева» (Tree shaking), в котором осуществляется удаление любого кода из окончательного файла, если этот код не используется.

При использовании Incremental DOM фреймворк не интерпретирует компонент; вместо этого компонент ссылается на инструкции. Если какая-либо инструкция осталась нетронутой, то она не будет использоваться в будущем (рис. 2). Поскольку данная информация известна во время компиляции, можно исключить неиспользуемые инструкции из bundle.

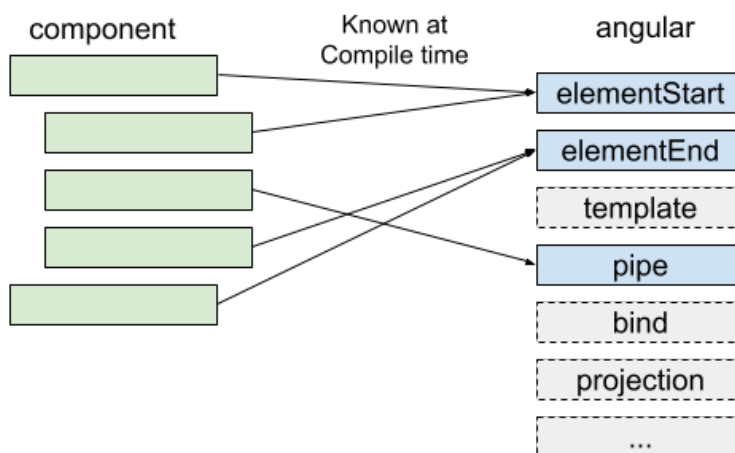


Рис. 2. Рендеринг веб-приложения с использованием Incremental DOM

Для работы Virtual DOM необходим интерпретатор. В момент компиляции неизвестно, какая именно его часть понадобится, а какая — нет, поэтому необходимо поместить его в браузер целиком (рис. 3).

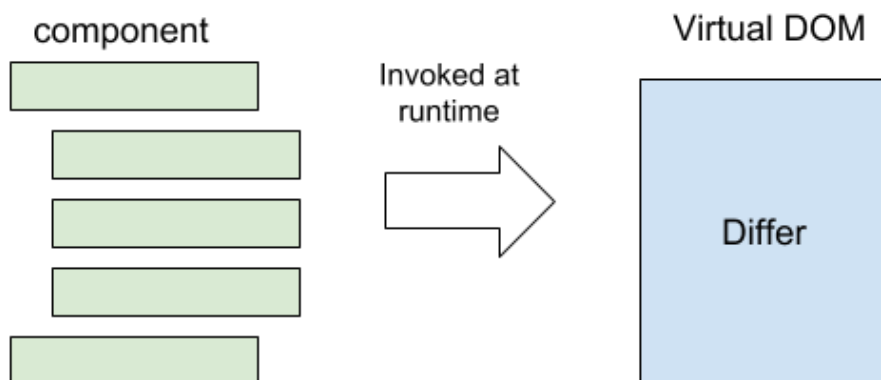


Рис. 3. Рендеринг веб-приложения с использованием Virtual DOM

4. Потребление памяти. Фреймворк Virtual DOM создаёт всё дерево с нуля при каждом повторном рендеринге (рис. 4).

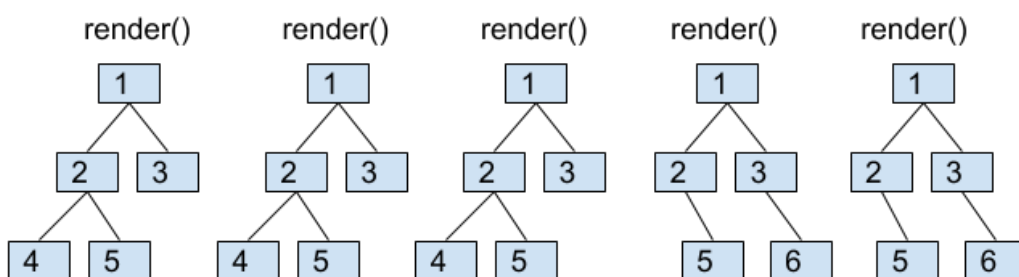


Рис. 4. Дерево памяти при использовании Virtual DOM

Incremental DOM не требует памяти для повторного рендеринга представления, если оно не вносит изменения в DOM. Память необходимо будет выделить только в том случае, если будут добавлены или удалены DOM-узлы, а объём выделяемой памяти будет пропорционален производимым изменениям в DOM (рис. 5).

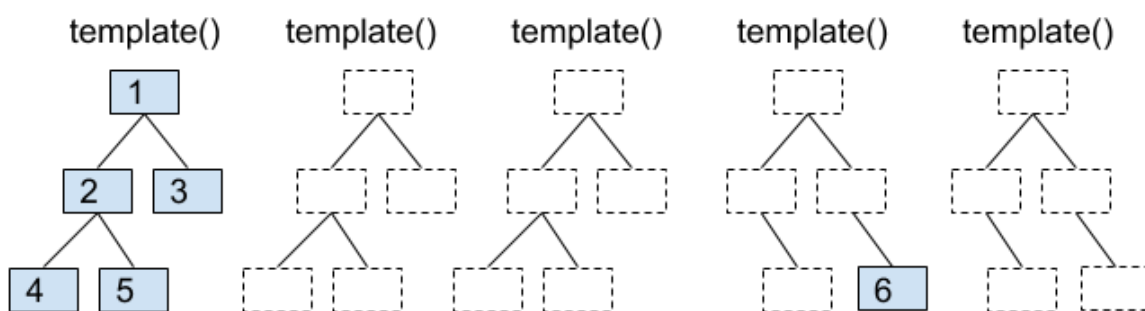


Рис. 5. Дерево памяти при использовании Incremental DOM

Исходя из описанных выше критерий составим таблицу сравнения (табл. 1).

Количественные оценки платформ продемонстрированы в табл. 2 – 4. Для проведения тестов использовался ноутбук MacBook Pro со следующим аппаратно-программным обеспечением: Intel Core i5 (2,3 ГГц/8 ГБ), macOS 10.15.4, Safari 13.1.

Таблица 1

Таблица сравнения фреймворков


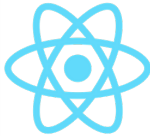

	 Angular	 React	 Vue
Первый выпуск	2010 (2016)	2013	2014
Время на изучение	для его освоения необходимо изучить Typescript	React предоставляет краткое руководство, благодаря которому настроить данный продукт можно за час. Однако есть необходимость использования сторонних библиотек.	Vue интуитивно понятен, поэтому освоение будет достаточно быстрым. Однако требует постоянной отладки и дополнительного тестирования.
Рендеринг	На стороне клиента и сервера	На стороне сервера	На стороне сервера
Концепция программирования	Incremental DOM	Virtual DOM	Virtual DOM
Модель	MVVM	MVC	MVVM
Документация и сообщество программистов	Подробная документация и отличное комьюнити	Часто обновляется/добавляется документация, но большое комьюнити	Подробная документация, но комьюнити небольшое
Рейтинг на GitHub	59531	146224	160793
Кол-во копий репозитория на GitHub (forks)	16291	28223	24298
Кол-во вопросов/улучшений на GitHub (Issue)	3362 открытых / 17663 закрытых	537 открытых / 8492 закрытых	474 открытых / 8606 закрытых
Размер bundle	304,7 KB	6,3 KB	63,5 KB
Google Trend	В среднем 62	В среднем 87	В среднем 27
Популярность на 2020 год	Популярный	Наиболее популярный	Менее популярный
В каких компаниях используется	Google, Forbes, Wix, weather.com	Facebook, Uber, Netflix	Alibaba, GitLab

Таблица 2

Метрики запуска

Название фреймворка	Vue	React	Angular
Ситуация, когда процессор и сеть простаивают, мс	2313,1	2529,8	2879,7
Общее время, необходимое для разбора/компиляции/оценки всех скриптов страницы, мс	62,4	99,1	194,4
Общий размер передачи по сети (пост-сжатие) всех ресурсов, загруженных на страницу, КБ	210,9	260,6	302,1

Таблица 3

Длительность операций

Название фреймворка	Vue	React	Angular
Операции			
Создание 1000 строк, мс	167,0	170,0	167,3
Замена 1000 строк (5 прогонов), мс	139,1	141,0	144,1
Обновление каждой десятой строки для 1000 строк (3 прогона), мс	241,9	179,1	151,5
Выделение выбранной строки (5 прогонов), мс	103,2	32,4	32,7
Изменение местами 2 строк для таблицы с 1000 строками (5 прогонов), мс	68,5	461,2	464,7
Удаление строк, мс	51,1	44,7	45,6
Создание 10000 строк, мс	1388,0	1670,8	1428,9
Добавление 1000 к таблице из 10000 строк, мс	326,4	335,5	313,0
Очистка таблицы с 1000 строк, мс	166,1	150,7	248,8

В табл. 3 сравнивается длительность разных операций для каждого фреймворка. В среднем фреймворк Vue.js оказался самым быстрым по сравнению с другими платформами, хотя при обновлении каждой десятой строки для 1000 строк, и выделении выбранной строки он замедляется.

Таблица 4

Распределение памяти

Название фреймворка	Vue	React	Angular
Использование памяти после загрузки страницы, Мбайт	1,2	1,3	2,7
Использование памяти после добавления 1000 строк, Мбайт	4,0	3,9	5,1
Использование памяти после нажатия кнопки «Обновить каждую 10-ю строку 5 раз», Мбайт	4,4	4,7	5,5
Использование памяти после нажатия кнопки «Создать 1000 строк 5 раз», Мбайт	4,6	5,2	5,9
Использование памяти после создания и очистки 1000 строк 5 раз, Мбайт	2,6	3,2	4,3

При исследовании разных показателей быстродействия в среднем фреймворк Vue.js показал наилучшие результаты по сравнению с другими платформами.

Сравнить фреймворки по многочисленным критериям качества программного обеспечения, например [7] и [8], достаточно трудно. Но проведенный анализ позволил понять главное – для каких целей и для какого уровня профессиональных навыков разработчика стоит выбирать ту или иную программную платформу.

- При единичных разработках выбор программного шаблона интерактивного веб-приложения ограничивается фреймворками Vue.js и React, издержки на освоение и развертывание которых минимальны. Также стоит выбирать эти фреймворки, если для проекта предусматривается многоступенчатое обновление и расширение функциональности, так как они имеют обратную совместимость.
- Для разработки крупномасштабного проекта в качестве программного шаблона лучше рассматривать Angular. Этот фреймворк обеспечивает максимальную гибкость и скорость рендеринга. Опыт других разработчиков позволит решить вопросы, которые могут возникнуть при работе над веб-приложением.

Библиографический список

1. Описание программных шаблонов. URL: <https://medium.com/@TechMagic/reactjs-vs-angular5-vs-vue-js-what-to-choose-in-2018-b91e028fa91d> (дата обращения: 12.03.2020).
2. SEO-оптимизация. URL: <https://seo.ru/chto-takoe-seo> (дата обращения: 12.03.2020).
3. Incremental DOM. URL: <http://google.github.io/incremental-dom> (дата обращения: 17.03.2020).
4. Virtual DOM. URL: <https://ru.reactjs.org/docs/faq-internals.html> (дата обращения: 17.03.2020).
5. Статья Incremental DOM и Virtual DOM. URL: <https://blog.nrwl.io/understanding-angular-ivy-incremental-dom-and-virtual-dom-243be844bf36> (дата обращения: 17.03.2020).
6. Анализ быстродействия. URL: <https://krausest.github.io/js-framework-benchmark/current.html> (дата обращения: 29.03.2020).
7. Стандарт ISO/IEC 25000:2014. URL: <http://www.gostinfo.ru/catalog/Details/?id=5348411> (дата обращения: 29.03.2020).
8. Стандарт ISO/IEC 9126. URL: <http://docs.cntd.ru/document/gost-r-iso-mek-9126-93> (дата обращения: 29.03.2020).

УДК 004

Е. А. Клычкова

студент кафедры проблемно-ориентированных вычислительных комплексов

Н. Н. Григорьева – старший преподаватель – научный руководитель

АНАЛИЗ МНОГОКОМПОНЕНТНОЙ ТРАНСПОРТНОЙ ЗАДАЧИ С ОГРАНИЧЕНИЯМИ НА ПРОПУСКНУЮ СПОСОБНОСТЬ

Транспортная задача (ТЗ) представляет собой частный случай общей задачи линейного программирования, специфическая структура которой может быть использована для описания разнообразных проблем, не связанных с организацией перевозок. Транспортные модели достаточно разнообразны, их анализ и создание алгоритмов их решения могут служить основой для разработки эффективных вычислительных методов решения разнородных практических задач. [1]

В работе [2] была проанализирована модель многокомпонентной ТЗ, предполагающая выбор пользователем услуг одного из нескольких перевозчиков, и рассматривались различные способы её решения. Была предложена комплексная оценка перевозки, учитывающая как собственно стоимость перевозки, так и субъективную оценку пользователем качества работы перевозчика; разработан алгоритм для решения такой задачи. Достоинством полученного алгоритма является то, что он имеет однозначные критерии поиска решения, учитывает как стоимостные, так и качественные критерии, позволяет использовать стандартные алгоритмы построения опорного плана и его дальнейшей оптимизации. Впоследствии на основе данной работы было создано программное средство для решения описанной многокомпонентной транспортной задачи [3].

В процессе дальнейшей работы над моделью в алгоритм формирования комплексной оценки перевозки была внесена модификация с целью сделать эту оценку более гибкой и индивидуализированной.

В качестве показателей качества перевозки можно принять для примера следующие факторы:

V1 – надежность соблюдения графика доставки;

V2 – способность перевозить разнообразные грузы;

V3 – безопасность и/или сохранность груза;

V4 – возможная частота отправки груза;

V5 – оперативность доставки груза.

Процесс формирования субъективной составляющей оценки показан на примере в табл. 1.

Таблица 1

Пример формирования оценки качества транспортных услуг

Перевозчик	Показатели качества услуг					Результирующая оценка качества услуг перевозчика T_k
	V1	V2	V3	V4	V5	
P1	9	7	7	5	6	107
P2	8	6	6	4	5	92
P3	7	7	5	8	7	96
P4	5	4	5	7	9	81
k_m (значимость показателя V_{km} для потребителя)	3	4	5	1	2	

Для определения оценки качества услуг перевозчика P потребитель сначала оценивает показатели качества разных перевозчиков, к примеру, по десятибалльной шкале. Кроме этого, может быть введен весовой коэффициент k_m , показывающий значимость для конкретного потребителя транспорт-

ных услуг того или иного показателя V_m (кому-то важна скорость, кому-то безопасность). В простейшем случае коэффициент k_m может быть оценкой от 1 до 5 (5 – самый важный показатель). Сама же результирующая оценка качества услуг T_k рассчитывается по формуле: $T_k = \sum_{m=1}^5 k_m V_{km}$. Такой поряд

док формирования оценки качества транспортных услуг является более адаптивным к пожеланиям потребителя, чем тот, что предложен в [2].

Далее, в соответствии с алгоритмом, разработанным в [2], полученная оценка качества используется для пересчёта стоимостной матрицы:

- производится масштабирование оценки качества услуг перевозчика $T'_k = \frac{T_k}{T_{\max}} \cdot C_{\max}$.

- определяется новая комплексная оценка перевозки $C'_{ijk} = \sqrt{C_{ijk}^2 + (T'_{\max} - T'_k)^2}$.

Далее мы переходим к рассмотрению усложнённого варианта многокомпонентной транспортной задачи, в котором дополнительно могут вводиться ограничения на пропускную способность d_{ij} для отдельных перевозок из пункта отправления в пункт назначения (от поставщика i к потребителю j).

Табличное представление многокомпонентной модели ТЗ с ограничениями на пропускную способность приведено в табл. 2. Математическое описание – основные ограничения и целевые функции, остаются теми же, что и в базовом варианте ТЗ с выбором перевозчика [2].

Добавляются ограничения, связанные с ограничениями на значения пропускной способности d_{ij} :

- ограничения на пропускную способность для поставщиков и перевозчиков $\sum_{j=1}^n d_{ij} \geq a_i$ и

$$\sum_{i=1}^m d_{ij} \geq b_j;$$

- ограничения по пропускной способности для перевозок: $0 \leq X_{ijk} \leq d_{ij}$.

Таблица 2

Модель многокомпонентной транспортной задачи с правильным балансом и ограничениями на пропускную способность

Пункты отправления и объёмы груза	Пункты назначения и их спроса												
	В1:b1				В2:b2				...	Вn:bn			
	Перевозчики												
	P1	P2	...	Ph	P1	P2	...	Ph	...	P1	P2	...	Ph
A1:a1	C111	C112	...	C11h	C121	C122	...	C12h	...	C1n1	C1n2	...	C1nh
	X111	X112	...	X11h	X121	X122	...	X12h	...	X1n1	X1n2	...	X1nh
	d11				d12				...	d1n			
A1:a2	C211	C212	...	C21h	C221	C222	...	C22h	...	C2n1	C2n2	...	C2nh
	X211	X212	...	X21h	X221	X222	...	X22h	...	X2n1	X2n2	...	X2nh
	d21				d22				...	d2n			

Am:am	Cm11	Cm12	...	Cm1h	Cm21	Cm22	...	Cm2h	...	Cmn1	Cmn2	...	Cmnh
	Xm11	Xm12	...	Xm1h	Xm21	Xm22	...	Xm2h	...	Xmn1	Xmn2	...	Xmnh
	dm1				dm2				...	dmn			

Как показано в [4], в соответствии с теоремой о существовании потоков, чтобы задача с ограничениями на пропускную способность имела решение, для любых совокупностей пунктов назначения и пунктов отправления должно выполняться условие:

$$\sum_{s=1}^p a_{i_s} - \sum_{r=1}^q b_{j_r} \leq \sum_{s=1}^p \sum_{j=1, j \neq j_1, \dots, j_q}^m d_{i_s j},$$

В качестве примера будем рассматривать задачу, представленную в табл. 3, где буквой М обозначим неограниченную пропускную способность.

Для задачи из таблицы 3 взяты значения оценок качества для первого перевозчика равно $T_1 = 200$, для второго - $T_2 = 120$, а значения мощностей перевозчиков соответственно $P_1 = 300$ и $P_2 = 800$

В таблице 4 приведены пересчитанные оценки комплексные оценки транспортной таблицы – модифицированная стоимостная матрица. По данной таблице и будет идти поиск решения.

Алгоритм составления опорного плана при наличии ограничений приведен, в частности, в учебном пособии [5].

Таблица 3

Пример транспортной задачи (исходная транспортная таблица)

	B ₁		B ₂		B ₃		B ₄		
	P ₁	P ₂	P ₁	P ₂	P ₁	P ₂	P ₁	P ₂	
A ₁	1	3	3	1	6	5	8	3	200
	150		300		50		M		
A ₂	2	1	1	2	3	3	6	1	350
	300		140		200		160		
A ₃	3	1	2	4	7	7	8	4	350
	M		200		30		130		
	195		305		195		205		900

Таблица 4

Транспортная таблица с модифицированными оценками

	B ₁		B ₂		B ₃		B ₄		
	P ₁	P ₂	P ₁	P ₂	P ₁	P ₂	P ₁	P ₂	
A ₁	1,887	5,128	3,4	4,2785	6,209	6,59	8,158	5,128	200
	150		300		50		M		
A ₂	2,561	4,279	1,887	4,616	3,4	5,128	6,209	4,2785	350
	300		140		200		160		
A ₃	3,4	4,279	2,561	5,771	7,18	8,14	8,158	5,771	350
	M		200		30		130		
	195		305		195		205		900

Согласно приведенной методике, при наличии ограничений вместо потребителя с запросами, большими конечной пропускной способности, вводятся два новых потребителя, для первого из которых устанавливается запрос, равный пропускной способности, а для второго – запрос, соответствующий разнице между первоначальной заявкой и ограничением. При этом стоимости перевозок для второго нового перевозчика остаются прежними, только для клетки с ограничением устанавливается бесконечно большая стоимость. Таким образом, получается эквивалентная таблица, к которой могут быть применены методы построения опорного плана, учитывающие стоимости перевозок, такие как метод минимальной стоимости и метод двойного предпочтения.

Но при попытке применить описанный алгоритм к транспортной таблице с большим количеством ограничений (как в табл. 4) выяснилось, что не всегда с его помощью удастся сразу составить полностью сбалансированный опорный план. В этом случае приходилось рассматривать несбалансированный план в качестве квазиплана и, используя искусственные дополнительные клетки, за несколько итераций доводить его до сбалансированного состояния [4]. Данная ситуация существенно усложняет порядок действий и увеличивает время решения задачи.

Указанным способом задача была решена, полученный опорный план перевозок приведен в табл. 5. Общая целевая функция задачи, пересчитанная на реальные стоимости перевозок, составила $Z(X)=2155$ единиц.

Таблица 5

План перевозок, полученный методом минимальной стоимости

	B ₁		B ₂		B ₃		B ₄		
	P ₁	P ₂	P ₁	P ₂	P ₁	P ₂	P ₁	P ₂	
A ₁	1 150	3	3	1 50	6	5	8	3	200
	150		300		50		M		
A ₂	2	1	1	2 110	3	3 165	6	1 75	350
	300		140		200		160		
A ₃	3	1 45	2	4 105	7	7 30	8	4 130	350
	M		200		30		130		
	195		305		195		205		900

Выполненный анализ многокомпонентной ТЗ с ограничениями на пропускную способность позволяет построить алгоритм решения такой задачи и определяет основные направления дальнейшего анализа, в частности – необходимость выявления условий, при которых метод введения дополнительных столбцов позволяет сразу получить сбалансированный опорный план, и разработки алгоритма оптимизации полученного опорного плана.

Библиографический список

1. Таха Х. А. Введение в исследование операций. М.: Вильямс, 2005. 912 с.
2. Клычкова Е. А. Анализ многокомпонентной транспортной задачи // Семьдесят вторая международная студенческая научная конференция ГУАП: сб. докл.: в 4 ч. Ч. 2: Техн. науки. СПб.: ГУАП, 2019. С. 125–128.
3. Клычкова Е. А. Разработка программного средства для решения многокритериальной транспортной задачи: [рукопись] : дипломный проект / Санкт-Петербург. гос. ун-т аэрокосм. приборостроения. СПб., 2019. 65 л.
4. Габасов Р., Кириллова Ф. М. Методы линейного программирования. Ч. 2: Транспортные задачи. М.: Книжный дом «ЛИБРОКОМ», 2018. 240 с.
5. Общий курс высшей математики для экономистов: учебник / под общ. ред. В. И. Ермакова. М.: ИНФРА-М, 2007. 659 с.

УДК 004.932

И. Р. Кондратьев

студент кафедры проблемно-ориентированных вычислительных комплексов

Е. П. Виноградова – старший преподаватель – научный руководитель

ВОПРОСЫ ОБРАБОТКИ ИЗОБРАЖЕНИЙ В MATLAB

Обработка изображений — любая форма обработки информации, для которой входные данные представлены изображением, например, фотографиями, видеокадрами или иными видами изображений. Обработка может осуществляться как для получения изображения на выходе (например, подготовка к полиграфическому тиражированию, к телетрансляции и т. д.), так и для получения другой информации (например, распознавание объектов, географических координат). Хорошим примером использования обработки изображений являются датчики корреляционно-экстремальных Системы автоматической навигации используют обработку изображений для получения координат местоположения. Для этого корреляционно-экстремальные системы навигации сопоставляют текущие изображения с эталонами и получают с помощью алгоритмов кросс корреляции положение экстремум-максимум, соответствующий контрольным точкам маршрута движения.

Для простой обработки изображений был выбран инструментальный язык программирования Matlab. Потому что это язык программирования высокого уровня, оснащенный банком специальных библиотек с функциями. Эти функции позволяют из набора стандартных команд, создать необходимые программы. Документация – подробная и хорошая, комьюнити (сообщество)- большое и общительное, единственный минус, это все на английском языке. МАБЛАВ сегодня используется многими университетами мира и промышленными организациями мира.

Краткая теория. Свертка является главным инструментом обработки данных в линейных компонентах усиления электронных систем. Такую систему считают линейной, если кривизна функции усиления участка компенсируется следующим участком с обратной кривизной, поэтому весь тракт усиления эквивалентен действию линейного усилителя.

Соотношение 'вход-выход' в классе цифровых линейных апериодических инвариантных относительно сдвига систем определено уравнением свертки.

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{k=\infty} x[k]h(n-k),$$

где $y[n]$ – выходной сигнал, $x[n]$ - входной сигнал, $h[x]$ – импульсная характеристика системы. Символ '*' означает одномерную свертку, n и k - целые числа; суммирование представлено интервалом от $-\infty$ до $+\infty$.

MATLAB предлагает использовать двумерную функцию свертки в виде

$$C = \text{conv2}(A, B, \text{shape}).$$

Здесь свертываются две матрицы A и B. Если одна из матриц представляет собой двумерную конечную импульсную характеристику (FIR фильтр), то другая фильтруется по двум направлениям координат.

В алгоритмах линейных преобразований используют несколько типов стандартных двумерных фильтров $h(m,n)$, которые используют совместно с функцией Matlab `imfilter`. Эти фильтры имеют такие имена: `average`, `disk`, `gaussian`, `laplasian`, `log`, `motion`, `previtt`, `sobel`.

Sobel фильтр или оператор Собеля — дискретный дифференциальный оператор, вычисляющий приближенное значение градиента яркости изображения. Результатом применения оператора Собеля в каждой точке изображения является либо вектор градиента яркости в этой точке, либо его норма. Используется в алгоритмах выделения границ. Оператор собеля, можно увидеть на рис. 1.

Sx по горизонтали, Sy по вертикали.

$$S_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}; \quad S_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

Рис. 1. Операторы Собеля

Пример sobel фильтра можно увидеть на рис. 2.

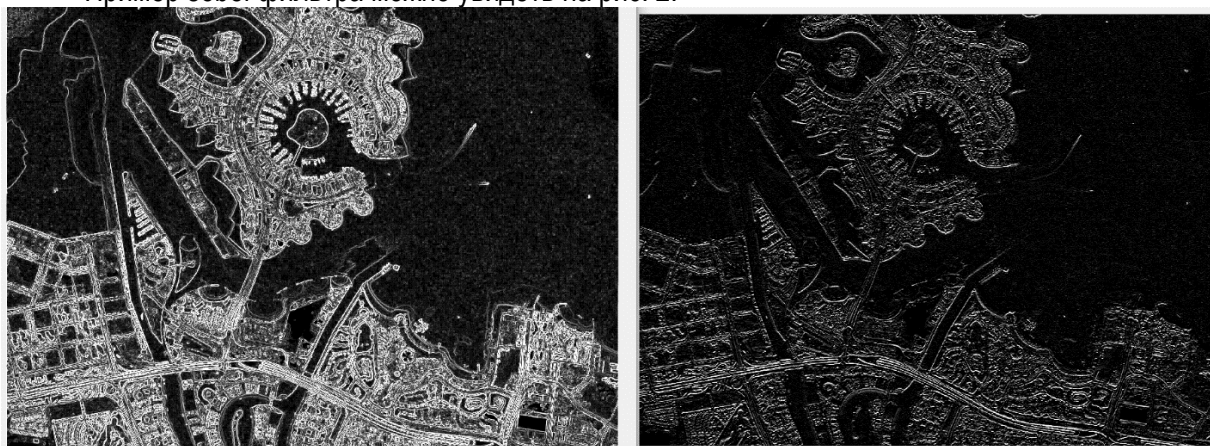


Рис. 2. Собел фильтр справа – sobel встроенный Matlab, слева – sobel через оператора собела

Как устанавливает теорема Фурье о сдвиге, оказывающая влияние на поведение алгоритма кросс-корреляции, он пригоден для определения только плоских взаимных сдвигов. Поэтому до начала действия кросскорреляционного алгоритма обычно запускается в действие технология подгонки текущего изображения к эталону. Это процесс образует этап его предобработки. Стандартно он включает в себя фильтрацию шумов, разрушающих изображение, определение вида и их уровня геометрических искажений с последующей их коррекцией.

Корреляция или корреляционная зависимость — статистическая взаимосвязь двух или более случайных величин. При этом изменения значений одной или нескольких из этих величин сопутствуют систематическому изменению значений другой или других величин, или уменьшаются в одну сторону одновременно.

Для численной оценки коррелированности двух переменных используют коэффициенты корреляции. Часто применяют коэффициент r линейной корреляции Пирсона. С его помощью измеряют силу линейного взаимоотношения двух переменных x, y длины n . Величина коэффициента корреляции r ограничена следующими пределами изменений: $-1 < r < +1$. $R=1$ Корреляция положительная – увеличивается один параметр, увеличивается другой, $r=0$ параметры не зависят, $r=-1$ – один паретр увеличивается, другой уменьшается.

Меры оценки теории подобия объектов

Для количественной оценки близости сопоставляемых объектов предлагается использовать следующие группы мер оценки подобия: аддитивные, фильтрующие корреляционные и корреляционные. Аддитивная группа включает пять основных мер:

1. SAD – сумма абсолютных разностей двух двумерных массивов данных.
2. ZSAD – центрированная относительно своих средних значений сумма абсолютных разностей.
3. SSD – сумма квадратов разностей элементов массивов.
4. RMS – сумма среднего квадрата разностей.
5. ZSSD – центрированная относительно средних значений массива A и массива B сумма квадратов разностей.

Группа корреляционных и корреляционно-фильтрующих представлена в таком составе: NCC – нормированная фильтрующая кросс-корреляция; ZNCC – нормированная и центрированная относительно своих средних значений кросс-корреляция.

Двумерная корреляция (кросс-корреляция) в обработке изображений играет важную роль. Особенно в связи с использованием быстрых преобразований Фурье. Формула двумерной корреляции:

$$f(x,y) * h(x,y) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n)h(x+m y+n),$$

Знак '+' в уравнении корреляции исключает зеркальный поворот $h(x,y)$ и первые отсчеты корреляции располагаются правее нуля.

Знак '-' в уравнении свертки вынуждает функцию $h(x,y)$ к зеркальному повороту относительно вертикальной оси. Первые отсчеты свертки располагаются левее нуля.

Стандартный алгоритм двумерной кросс-корреляции очень прост и понятен. Для начала используется «текущее» изображение $I_1(x,y)$ т.е. изображение которое получает алгоритм, и $I_2(x,y)$ – эталон, изображение которое хранит у себя алгоритм. Эталон сканирует окно, содержащее изображение $I_1(x,y)$ так, что можно записать $I_2(x+i, y+j)$, где i, j – текущие координаты. Это также проиллюстрировано на рис. 3.

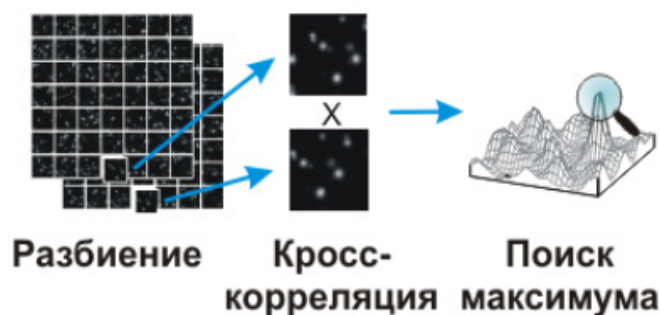


Рис. 3. Алгоритм кросс-корреляции

Идея этого алгоритма в том, чтобы в полученном изображении найти фрагмент, максимально похожий на эталон. Для понимания, насколько эталон и фрагмент изображения соответствуют друг другу нужна мера оценки соответствия. Мера может быть разной. Например: мера $SSD(i, j)$ – сумма квадратов разностей.

Процедура определения и компенсации плоского сдвига в задаче компенсации взаимного сдвига изображений является центральной. Известно, что фазовая кросс-корреляция является одним из лучших средств ее практического осуществления, позволяя быстрее других определить взаимный сдвиг x_0, y_0 изображений как отклонение пика корреляции от центра их совмещения.

Результаты работы кросс корреляции на рисунках ниже. На рис. 4 можно увидеть оригинал изображения, на рис. 5 два изображения имеющие сдвиг относительно друг друга на 256 пикселей по x и по y .

Если найти максимум кросс-корреляционной функции, то мы получим координаты максимума. С помощью этих координат можно получить сдвиг, для этого длину сканирующего окна делят пополам, и вычитают максимум, либо по x либо по y .

С помощью Matlab можно добавить шума в изображение, как на рис. 8, если попытаться также найти сдвиг и посмотреть на график кросс-корреляционной функции на рис. 9, то можно заметить, что пик находится ниже, а холм рядом больше и шире, что приводит алгоритм по поиску сдвига к ошибке. Чтобы избавиться от этой ошибки необходимо применять фильтры, к примеру, разобранный выше: sobel фильтр. На рис. 10 можно увидеть оригинальное изображение с примененным фильтром, без фильтра эти же изображения можно увидеть на рис. 5. На рис. 11 можно увидеть одинокий пик, который является искомым, если повторить процедуру поиска сдвига, она даст такие же результаты, что и идеальный вариант на рис. 7.



Рис. 4. Оригинал изображения



Рис. 5. Два изображения со сдвигом

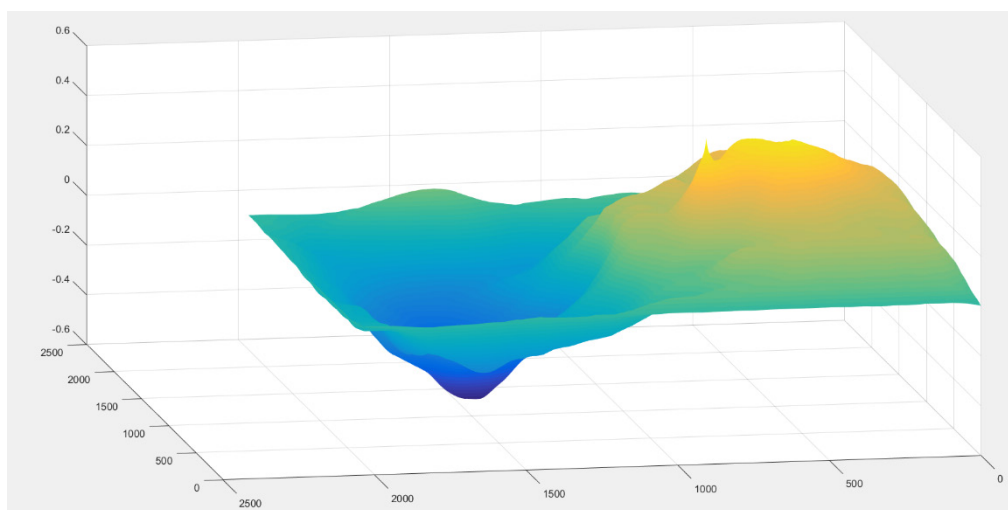


Рис. 6. График кросс-корреляционной функции

В результате получилось, что едва заметный пик на рис. 6 оказался выше, чем холм рядом. Сдвиг полученный благодаря координатам этого пика оказался правильным: 256 по x и по y. Если совместить два изображения на рис. 5, то получится результат на рис. 7



Рис. 7. Результат объединения изображений



Рис. 8. Зашумленное изображение

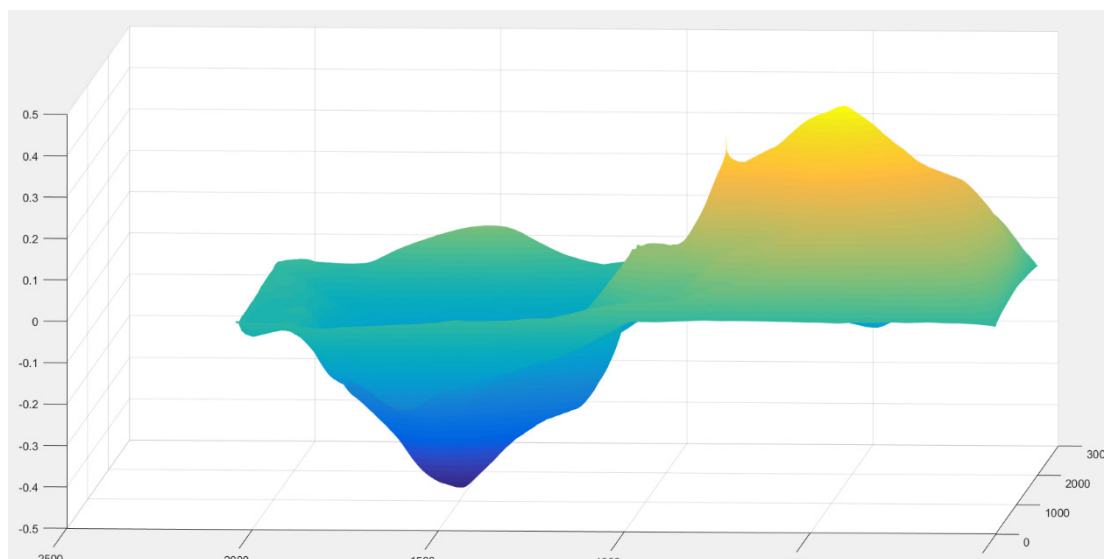


Рис. 9. График кросс корреляционной функции



Рис. 10. Изображение с фильтром Собел

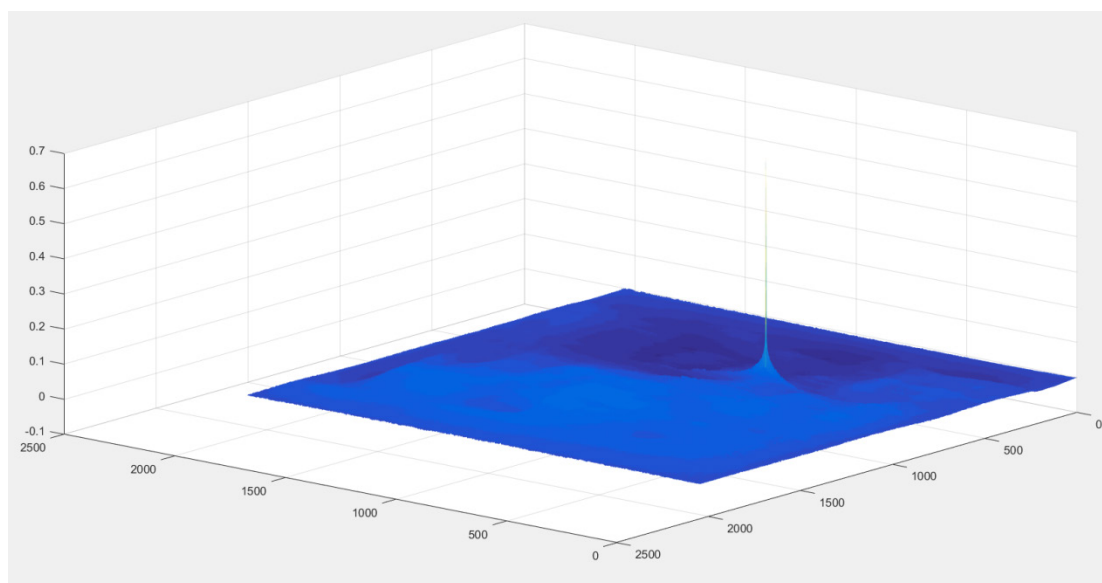


Рис. 11. График кросс-корреляционной функции с фильтром

В статье разобраны только минимальные и начальные способы обработки изображений, есть огромное количество фильтров, которые могут до неузнаваемости изменить изображение, для человеческого субъективного восприятия лучше рис. 5, а для машины 10.

Стандартная кросс корреляция является не лучшим выбором для использования в корреляционно-экстремальных системах навигации, т.к. её основной недостаток — это скорость, лучше с этим справится фазовая корреляция. Плюс ко всему, как можно было увидеть на рис. 6, холм рядом может в некоторых случаях выше пика, даже на чистых, не искаженных изображениях. Координаты пика и являются необходимыми для определения сдвига изображений, а это в свою очередь позволяет определить координаты положения самолета.

Библиографический список

1. *Анисимов Б. В., Курганов В. Д., Злобин В. К.* Распознавание и цифровая обработка изображений. М.: Высшая школа, 1983. 295 с.

УДК 621.382

П. А. Корепин

студент кафедры проблемно-ориентированных вычислительных комплексов
О. А. Кононов – кандидат технических наук, доцент – научный руководитель

УСТРОЙСТВО ОПРЕДЕЛЕНИЯ ПАДЕНИЯ ЧЕЛОВЕКА

Падение для человека может быть очень опасно, в особенности, если он находится в группе риска. Важно своевременно оказать первую помощь упавшему.

Согласно статистике всемирной организации здравоохранения, падения являются второй по значимости причиной смерти в результате несчастных случаев и непреднамеренных травм в мире. Наибольшее количество смертельных падений происходит среди людей старше 65 лет. Ежегодно происходит 37,3 миллиона падений с достаточно серьезными последствиями, при которых требуется медицинская помощь [1].

Поэтому задача обнаружения падения и оповещения о нем, актуальна. Данная задача может быть решена с помощью акселерометра, объединенного в систему с микроконтроллером, который обрабатывает информацию, полученную от акселерометра.

В современном мире развиваются системы медицинской сигнализации или Системы Экстренного Вызова Помощи (*PERS – Personal Emergency Response System*). Эти системы построены по принципу минимизации времени оказания медицинской помощи пострадавшему человеку, что существенно повышает шансы на выживание и выздоровление. Основным производителем приборов с функцией ручного вызова экстренных служб и функцией автоматического определения падения человека являются производители из Китая и канадский производитель *Lairac*. *PERS* системы в России появились относительно недавно, например проект «Кнопка жизни».

Есть два основных показателя, которые характеризуют оценку тяжести повреждения человека: *ASI* (*Acceleration Severity Index* – индекс степени опасности ускорения) и *HIC* (*Head Injury Criterion* – критерий травмы головы). Оба критерия изначально использовались для оценки показателей тяжести травм во время автомобильных аварий, но в дальнейшем стали использоваться для определения факта падения человека. Для этой цели больше подходит критерий *ASI*, так как вычисление *HIC* сравнительно дорогое и требует большой вычислительной мощности (6) [2].

Критерий *ASI* вычисляется следующим образом:

$$ASI(t) = \sqrt{\left(\frac{\bar{a}_x(t)}{a_x}\right)^2 + \left(\frac{\bar{a}_y(t)}{a_y}\right)^2 + \left(\frac{\bar{a}_z(t)}{a_z}\right)^2}, \quad (1)$$

$$\bar{a}_x(t) = \frac{1}{\delta} \int_t^{t+\delta} a_x(t) dt, \quad (2)$$

$$\bar{a}_y(t) = \frac{1}{\delta} \int_t^{t+\delta} a_y(t) dt, \quad (3)$$

$$\bar{a}_z(t) = \frac{1}{\delta} \int_t^{t+\delta} a_z(t) dt, \quad (4)$$

где $\bar{a}_x = \bar{a}_y = \bar{a}_z = 19.6 \frac{m}{c^2}$ пределы измерения акселерометра.

$$HIC = \max_{t_1, t_2} \left[(t_2 - t_1) \left(\frac{1}{(t_2 - t_1)} \int_{t_1}^{t_2} a(t) dt \right)^{2.5} \right], \quad (5)$$

$$a(t) = \sqrt{a_x^2(t) + a_y^2(t) + a_z^2(t)}, \quad (6)$$

Критерий *HIC* вычисляется следующим образом:

Следует учитывать, что изначально данный критерий был разработан для случаев автомобильных аварий, в этом случае значения пределов акселерометра должны быть следующие:

$\hat{a}_x = 12g, \hat{a}_y = 9g, \hat{a}_z = 10g$, где $g=9.8 \text{ m/c}^2$. Но так как ускорения автомобиля и падающего человека отличаются на порядок, значения пределов акселерометра тоже следует уменьшить [3].

Для обычного состояния показания акселерометра показаны на рис. 1.

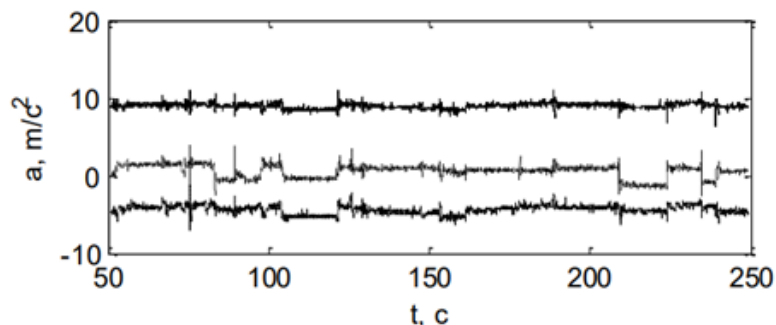


Рис. 1. Показания акселерометра для обычного состояния

Вычисления критерия *ASI* для обычного состояния показаны на рис. 2.

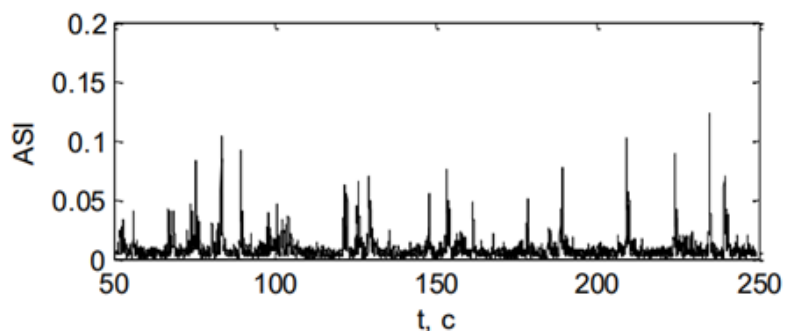


Рис. 2. Вычисления критерия *ASI* для обычного состояния

Видно, что показания всех трех осей акселерометра находятся на постоянном уровне, а *ASI* не превышает уровень 0,15.

На рис. 3 хорошо видны четыре фазы характеристики падения.

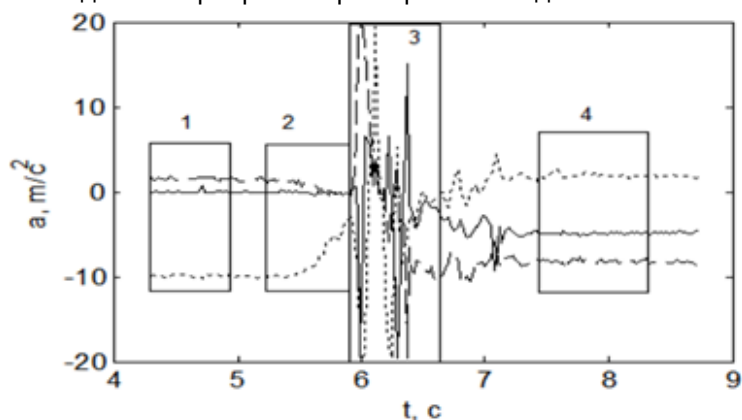


Рис. 3. Показания акселерометра при падении

Первая – это первоначальное состояние до падения. Вторая – это начало падения. Здесь видно, как показания акселерометра по всем осям стремятся к нулю. Третья фаза – столкновение

человека с поверхностью, это видно по большому скачку. Четвертая фаза – последствие. Обычно человек после падения и удара о поверхность некоторое время остается в лежачем состоянии.

На рис. 4 по уровню критерия ASI , рассчитанного во время падения человека, можно заметить скачок до 0,65. Для детектирования падения предлагается использовать уровень $ASI=0,4$. В таком случае факт падения человека будет обнаружен.

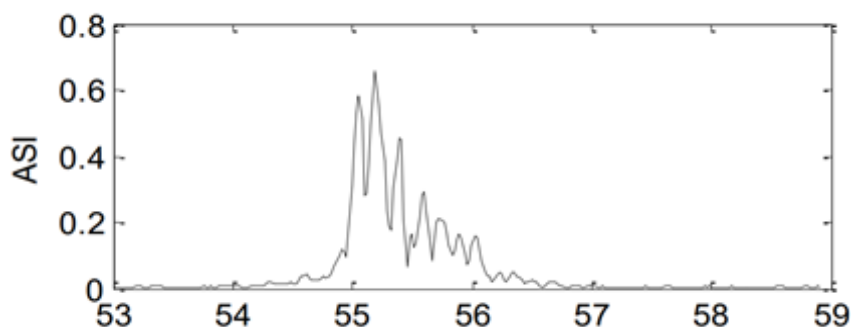


Рис. 4. Вычисления критерия ASI при падении

Функционал устройства следующий: работа акселерометра делится на две фазы. В ходе первой фазы акселерометр работает отдельно от основной вычислительной системы, тем самым экономя ресурс батареи. Результатом этой фазы является событие, по которому необходимо переводить основную систему из спящего состояния в рабочее. Во второй фазе акселерометр передает свои показания вычислительной системе, которая реализует проверку факта падения. Затем, после определение факта падения, микроконтроллер отправляет информацию на цифровой индикатор (*LCD* дисплей), а также подаётся звуковой сигнал. По истечению некоторого времени, информация о факте падения передается в соответствующие органы помощи. Информация, которая будет передана: местоположение, время происшествия. Следовательно, необходимо также обеспечить определение местоположения с помощью *GPS* датчика. У пользователя будет возможность отменить вызов, в случае ложного срабатывания, путем нажатия кнопки отмены, на это ему дается необходимое количество времени, все это сопровождается индикацией обратного отсчёта на экране, а также звуковым эффектом.

Библиографический список

1. Проблема падений среди пожилых людей: что можно считать основными факторами риска и наиболее эффективными профилактическими мерами? URL: http://www.euro.who.int/data/assets/pdf_file/0019/74701/E82552R.pdf (дата обращения: 26.05.2020).
2. Rolf Eppinger, Emily Sun, Faris Bandak et al. Development of Improved Injury Criteria for the Assessment of Advanced Automotive Restraint Systems – II – 1999 – National Highway Traffic Safety Administration National Transportation Biomechanics Research Center (NTBRC)/ URL: https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/rev_criteria_0.pdf (дата обращения^ 26.05.2020).
3. Воронов А. С., Калигин Н. Н. Интеллектуальный алгоритм определения падения человека. М.: ОАО «МКБ Компас». URL: http://elib.altstu.ru/journals/Files/pa2015_01/pdf/099voronov.pdf (дата обращения: 26.05.2020).

УДК 004.65

В. Е. Крылова

студент кафедры высшей математики и механики

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

ПРИМЕНЕНИЕ АЛГОРИТМОВ НЕЧЕТКОГО ПОИСКА ДЛЯ УСТРАНЕНИЯ ИНФОРМАЦИОННОЙ НЕОПРЕДЕЛЕННОСТИ

Постановка проблемы

На сегодняшний день Интернет и другого вида информационные системы используются человеком для удовлетворения своих информационных потребностей, а значит перед специалистами данной области встает главная задача: решение проблем поиска информации. Современный информационный поиск столкнулся с большим количеством самых различных проблем, но самыми глобальными и актуальными являются проблемы извлечения нужной информации и рождающаяся в ходе этого процесса явление информационной неопределенности. Дело в том, что смысловая обработка (соотнесение содержания сообщения с реальной, т. е. информационной потребностью) происходит в сознании человека, а отбор документов, формально соответствующих потребности, — в машинной среде с жесткой двоичной логикой. Это приводит к дефициту информации, её недостоверности, противоречивости, искажению, невозможности четкой интерпретации полученных данных.

Поэтому необходимо использовать новые инструменты поиска информации, позволяющие решить эти проблемы.

Алгоритмы нечеткого поиска

Одним из таких инструментов является нечеткий поиск (поиск по сходству, приближенное сопоставление строк) и его алгоритмы. Алгоритмы нечеткого поиска строк в массивах данных являются основой для построения современных систем проверки орфографии, которые используются в текстовых редакторах, системах оптического распознавания символов и поисковых системах, вроде Google или Yandex. Например, такие алгоритмы используются для функций, которые выдают пользователю сообщение «возможно вы имели в виду...» в поисковых системах, то есть поиск должен учитывать возможные ошибки и опечатки пользователей при вводе запросов, а также приводятся альтернативные варианты написания и слова, которые звучат одинаково, но пишутся по-разному.

Нечеткий поиск предоставляет достаточное количество средств для оценки сходства двух строк (сигнатурные алгоритмы, алгоритмы n-граммной индексации, строковые trie-деревья и прочие), но среди всех хотелось бы выделить *метрику Левенштейна (расстояние Левенштейна)*, позволяющей оценить степень сходства между двумя строками, и являющейся основой алгоритмов нечеткого поиска.

В чем суть расстояния Левенштейна? Расстояние Левенштейна равно минимальному числу элементарных операций редактирования, необходимых для преобразования одной строки в другую, в том числе операции замены, вставки и удаления одного символа.

Пусть S_1 и S_2 – две строки (длиной n и m соответственно) над некоторым алфавитом, тогда редакционное расстояние $D(S_1, S_2)$ можно подсчитать по следующей рекуррентной формуле:

$$D(n, m) = \begin{cases} 0; i = 0; j = 0 \\ i; j = 0; i > 0 \\ j; i = 0; j > 0 \\ \min \begin{pmatrix} D(i, j - 1) + 1 \\ D(i - 1, j) + 1; j > 0, i > 0 \\ D(i - 1, j - 1) + m(S_1[i], S_2[j]) \end{pmatrix} \end{cases}$$

где $m(a, b)$ равна нулю, если $a = b$ и единице в противном случае; $\min(a, b, c)$ возвращает наименьший из аргументов [1].

Программная реализация

Рассмотрим применение расстояния Левенштейна на практике. В данной работе он применяется для нечёткого поиска в базе данных с применением СУБД MySQL, так как она не предлагает встроенных решений для организации такого вида поиска. База данных является частью информационной системы, задача которой поиск сведений о клиенте по его *фамилии*, а также борьба с информационной неопределённостью путем нахождения пользовательских ошибок в запросах пользователей. Вспомогательным элементом при создании данной информационной системы выступает функционал языка PHP.

1. Вычисление расстояния Левенштейна будет вычисляться не на прямую от пользовательского запроса, а от его метафона – фонетического алгоритма для индексирования слов по их звучанию с учётом основных правил английского произношения. Для этого русское слово предварительно транслитерируется на английский язык. Полная функция расчёта метафона слова (рис.1):

```
function mtphn($str) {
    $rus = array('а', 'б', 'в', 'г', 'д', 'е', 'ё', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц',
    'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'ю', 'я');
    $lat = array('a', 'b', 'v', 'g', 'd', 'e', 'yo', 'zh', 'z', 'i', 'y', 'k', 'l', 'm', 'n', 'o', 'p', 'r', 's', 't', 'u', 'f', 'h', 'ts', 'ch',
    'sh', 'shch', ' ', 'y', ' ', 'e', 'yu', 'ya');
    return metaphone(str_replace($rus, $lat, mb_strtolower($str)));
}
```

id_user	surname	patronymic	name	phone	email	url	status	metaphone
1	Иванов	Валерьевич	Александр	58-98-78	ivanov@email.ru	NULL	active	IFNF
2	Лосев	Иванович	Сергей	9057777777	losev@email.ru	NULL	passive	LSF
3	Симдянов	Вячеславович	Игорь	9056666100	simdyanov@softtime.ru	http://www.softtime.ru/	active	SMTYNF
4	Кузнецов	Валерьевич	Максим	NULL	kuznetsov@softtime.ru	http://www.softtime.ru	active	KSNTSF
5	Нехорошев	Юрьевич	Анатолий	NULL	NULL	NULL	lock	NHRXF
6	Корнеев	Александрович	Александр	89-78-36	korneev@domen.ru	NULL	gold	KRNF
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рис. 1. Исходные данные с вычисленными метафонами

На следующем этапе необходимо реализовать функцию вычисляющее расстояние Левенштейна в среде MySQL [2]:

```
CREATE FUNCTION levenshtein( s1 VARCHAR(255), s2 VARCHAR(255) )
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE s1_len, s2_len, i, j, c, c_temp, cost INT;
    DECLARE s1_char CHAR;
    -- max strlen=255
    DECLARE cv0, cv1 VARBINARY(256);
    SET s1_len = CHAR_LENGTH(s1), s2_len = CHAR_LENGTH(s2), cv1 = 0x00, j = 1, i = 1, c = 0;
    IF s1 = s2 THEN
        RETURN 0;
    ELSEIF s1_len = 0 THEN
        RETURN s2_len;
    ELSEIF s2_len = 0 THEN
        RETURN s1_len;
    ELSE
        WHILE j <= s2_len DO
            SET cv1 = CONCAT(cv1, UNHEX(HEX(j))), j = j + 1;
        END WHILE;
        WHILE i <= s1_len DO
            SET s1_char = SUBSTRING(s1, i, 1), c = i, cv0 = UNHEX(HEX(i)), j = 1;
            WHILE j <= s2_len DO
                SET c = c + 1;
```



```

IF s1_char = SUBSTRING(s2, j, 1) THEN
  SET cost = 0; ELSE SET cost = 1;
END IF;
SET c_temp = CONV(HEX(SUBSTRING(cv1, j, 1)), 16, 10) + cost;
IF c > c_temp THEN SET c = c_temp; END IF;
SET c_temp = CONV(HEX(SUBSTRING(cv1, j+1, 1)), 16, 10) + 1;
IF c > c_temp THEN
  SET c = c_temp;
END IF;
SET cv0 = CONCAT(cv0, UNHEX(HEX(c))), j = j + 1;
END WHILE;
SET cv1 = cv0, i = i + 1;
END WHILE;
END IF;
RETURN c;
END;

```

Функция будет возвращать расстояние между метафоном слова из запроса и поля базы данных, по которому происходит поиск. Например, для некоторых полей столбца, по которому производится запрос, имеем следующий результат (рис.2):

	id	lastname	metaphone1	lnwm	matapnone2	rast
▶	2	Корнеев	KRNF	Канееф	KNF	1
	3	Лосев	LSF	Лфсев	LFSF	1
	4	Симдянов	SMTYNF	Семденав	SMTNF	1

Рис. 2. Результат вычислений в табличной форме

2. Оставшаяся программная реализация сводится к логике выборки данных в ответ на запрос:

–Находим все слова в столбце запроса по метафону с расстоянием Левенштейна меньше двух символов.

–Если ничего не найдено, то можно говорить о том, что пользователь сделал слишком много ошибок в запросе, на этом поиск заканчивается без возврата результата.

–Если найден один результат, то возвращаем его

–Если найдено более одного, то вычисляем максимальный процент схожести с поисковым запросом для каждого слова. Возвращаем все слова с этим процентом.

Результат работы разработанного программного средства представлен на рис.3:

Ищем название: Лфсев

Возможно, Вы ищите: [Лосев](#)

Рис. 3. Сообщение для пользователя после выполнения программы

Таким образом, на основании алгоритма нечеткого поиска для устранения информационной неопределённости создана программная реализация, вычисляющая расстояние Левенштейна в среде MySQL.

Библиографический список

1. *Карахтанов Д. С.* Программная реализация алгоритма Левенштейна для устранения опечаток в записях баз данных // Молодой ученый. 2010. № 8. С. 158–162.
2. Levenshtein distance. URL: <http://www.artfulsoftware.com/infotree/qrytip.php?id=552> (дата обращения: 12.03.2020).

УДК 621.396.96

Е. О. Лукашова

магистрант кафедры проблемно-ориентированных вычислительных комплексов

С. А. Ненашев

студент кафедры вычислительных систем и сетей

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

ИССЛЕДОВАНИЕ РЕЖИМОВ ОПТИКО-ЭЛЕКТРОННОГО МОНИТОРИНГА, РЕАЛИЗУЕМЫХ НА БАЗЕ МАЛЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ

Введение

Анализ тенденций в мировом развитии МЛА показывает, что наиболее перспективным направлением в настоящее время является разработка и производство комплексов воздушного мониторинга поверхности различного класса. Широкое развитие и распространение получают комплексы с МЛА и его составными частями, предназначенными для решения задач наблюдения и освоения территорий.

Оптико-электронная система (ОЭС) на МЛА предназначена для:

- отслеживания грузов, транспортных средств;
- наблюдения, поиска и целеуказания объектов;
- транспортировки, логистики, доставки;
- работы в опасных зонах, получения информации для служб спасения;
- поиск людей в горах, в дорожно-транспортных происшествиях, в чрезвычайных происшествиях и чрезвычайных ситуациях.

Выдача результата обнаружения осуществляется в виде угловых координат (азимут и угол места) на индикатор оператора и в бортовую цифровую вычислительную машину МЛА. Оборудование МЛА для наблюдения состоит из:

- навигационного модуля для связи с системами GPS/ГЛОНАСС;
- модулей наблюдения и ведения объектов;
- подвесов для установки дополнительных камер и оптической техники;
- кронштейна для монтажа тепловизоров и другой специальной техники.

Методы обнаружения объекта

Целью обнаружения объекта является разделение интересующей области видеоизображения и отслеживание его перемещения и расположения. Обнаружение и классификация объекта является предшествующими этапами для отслеживания объекта в последовательности потока кадров. Обнаружение объекта выполняется для проверки наличия объектов на видео и точного определения местоположения этого объекта.[1]

В сложных условиях, улучшить качество получаемой информации и результата наблюдения можно с помощью алгоритмов оценивания параметров геометрических преобразований изображений и методов пространственно-временной фильтрации. На рис. 1 представлена структура систем обнаружения объектов.

Структура системы обеспечения оптического контроля для определения реперных знаков и других фигур представлена на рис. 2. Структура состоит из устройства ввода-вывода отображения информации, восприятия, предварительной обработки и анализирующего устройства, содержащее блок памяти. На представленной структурной схеме устройство анализирует, производит сравнение эталонного изображения с поступающим изображением проверяемого объекта, и на основе анализа выдает вердикт о наличии объекта. После обработки эталонного изображения система создает двумерный массив данных, содержащий данные о цвете. Большое количество методов и алгоритмов сравнения изображений, использующих различные подходы и математические модели.



Рис. 1. Структура программно-алгоритмического обеспечения ОЭС обнаружения объектов

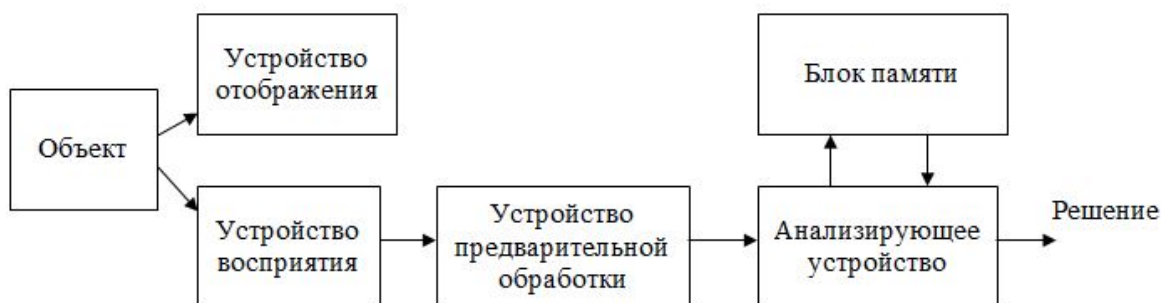


Рис. 2. Структурная схема распознавания объектов

Первым шагом в процессе отслеживания объектов является идентификация объектов в видеопоследовательности, и кластеризация пикселей этих объектов. Поскольку движущиеся объекты обычно являются основным источником информации, большинство методов основываются на обнаружении движущихся объектов. Основные методы, для 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 является фоновое моделирование, которое должно быть достаточно чувствительным, чтобы распознать движущиеся объекты. Фоновое моделирование — это создание эталонной модели. Эталонная модель используется для вычитания фона, в котором каждая видеопоследовательность сравнивается с эталонной моделью, чтобы определить возможное изменение. Различия между текущими видеокадрами и не подвижными 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

Выбранная подвижная область может быть с различными объектами, такими как люди, транспортные средства, птицы, двигающиеся облака, и другие движущиеся объекты. Для этого используются особенные формы областей движения. Подходы к классификации объектов следующие:

- классификация на основе формы;
- цветовая классификация;
- классификация на основе текстуры.

а) Классификация на основе формы

Для классификации подвижных объектов доступны различные описания информации о форме областей движения, такие как представления точек, прямоугольников и больших двоичных объектов. Входные данные представляют собой смесь параметров объекта на основе изображения и сцены, таких как область изображения, видимое соотношение сторон ограничивающего прямоугольника и масштабирование камеры.

Классификация выполняется на каждом большом двоичном объекте в каждом кадре, и результаты сохраняются в гистограмме.

б) Цветовая классификация

В отличие от многих других характеристик изображения (например, формы), цвет относительно постоянен при изменении точек, и его легко получить. Хотя цвет не всегда подходит в качестве единственного средства обнаружения и отслеживания объектов, но низкая вычислительная стоимость алгоритмов делает цвет желательным признаком для использования в случае необходимости. Для обнаружения и отслеживания транспортных средств или пешеходов в режиме реального времени используется метод на основе цветных гистограмм. Модель гауссовой смеси создается для описания распределения цвета в последовательность изображений и сегментирование изображения на фон и объекты. Окклюзия объекта обрабатывается с использованием буфера окклюзии.

в) Классификация на основе текстуры

Метод на основе текстуры, представлен на рис. 3, анализирует появление градиента в локализованных частях изображения, вычисляется на плотной сетке равномерно расположенных ячеек и использует перекрывающуюся локальную нормализацию контраста для повышения точности.

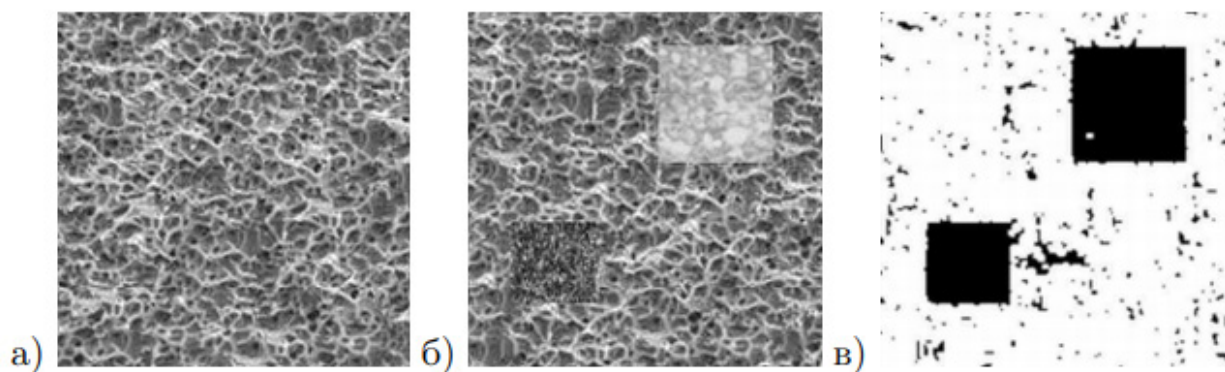


Рис. 3. Сегментация по методу обнаружения текстурных неоднородностей:
а) обучающий снимок; б) исследуемый снимок; в) результат работы метода

Виды систем оптико-электронного мониторинга для МЛА

Системы мониторинга решают задачи приема и обработки сигналов, одними из которых являются задачи обнаружения объектов и фильтрации полученных сигналов на фоне помех. Полезные сигналы соответствуют обнаруженным объектам в зоне мониторинга или на поверхности Земли, обычно это объекты искусственного происхождения (самолеты, корабли, автомобили, здания, сооружения, дороги, мосты). Так как обнаруженные объекты физически различны, то описание их математических моделей не всегда точно передает все физические особенности. Некоторые объекты как

естественного, так и искусственного происхождения, являются помехами (рельеф местности, растительность на земной поверхности, излучающие радиотехнические системы и пр.). Кроме них, могут действовать и активные помехи как естественные, так и искусственные.

Если задача неопределенна, обнаружение и различение объектов становятся частями более сложной проблемы распознавания.

Современные системы МПА обычно формируют двумерные сигналы, т.е. изображения. К ним относятся радары высокого разрешения, лазерным локаторам и к инфракрасным системам наблюдения. Изображение может быть телевизионным (представлено на рис. 4(а)) как в радарях, так и специальным (радиолокационным, представлено на рис. 4(б)), например, в координатах дальность — скорость.

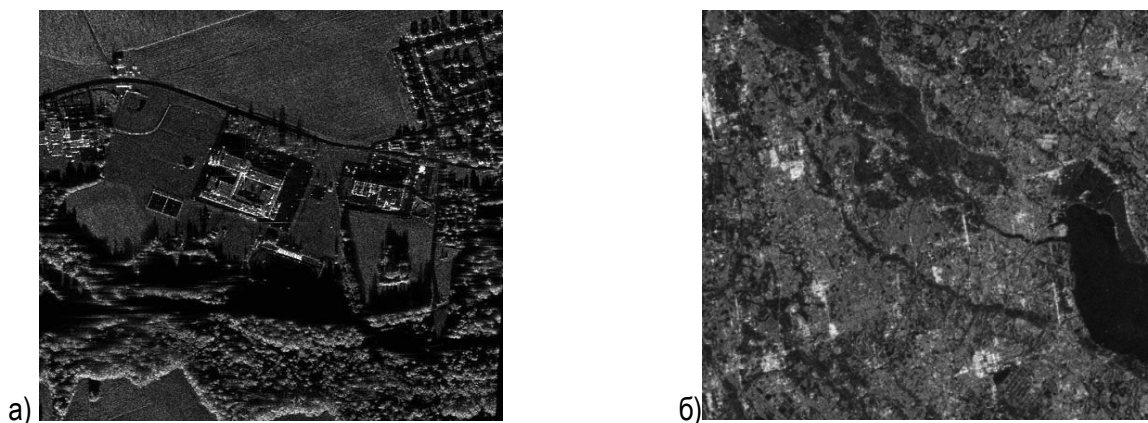


Рис. 4. Телевизионное изображение высокого разрешения (а), радиолокационное изображение высокого разрешения (б)

Из-за сложности моделей, описывающих объекты (корабли, автомобили, строения, мосты и др.), обнаружение объектов осуществляется для фрагментов таких изображений. Объекты могут иметь интегральное представление (в виде пятен) или дифференциальное (в виде контурных линий). Для получения таких представлений исходное изображение подвергается предварительной линейной обработке, включающей сглаживание и дифференцирование.

Выделение простейших геометрических примитивов является довольно простой задачей: прямолинейные отрезки, пересечения линий и углы на контурах. Ключевые точки выделяют центры соответствующих областей, которые соответствуют искомым объектам на анализируемом изображении. Вышеописанные точки применяются далее для сравнения изображений, полученных с разных источников. Такое сравнение необходимо, например, для решения навигационных задач и для мониторинга за движущимися объектами.

Обработка изображения

Целью обработки изображения является решение задачи приема: получение, извлечение или восстановление информации. Задача приема состоит из: событий, сигналов, наблюдений и решений, уточнение которых образует частные задачи. Задачи обнаружения и различения сигналов являются наиболее простыми. Конечный результат, решения задачи обнаружения и различения является принятие решения d_i относительно события e_i , которое связано с определенным сигналом s_i из заданного набора M сигналов, $i = 0, 1, 2, \dots, M-1$. Нулевое событие e_0 , показывает, что сигнал отсутствует. Такая формулировка задачи соответствует многоальтернативному обнаружению.

Решение принимается в результате обработки наблюдения $y(t)$, которое представляет собой процесс (напряжение или электрический ток) в некой точке модуля приемника (на выходе чувствительного элемента или детектора). Выбор точки наблюдения зависит от того, какая часть приемного модуля исследуется. Всегда предполагается некоторая предварительная обработка, связанная с видом системы. С учетом дискретизации по времени и многоканальной обработки исходное наблюдение можно представить массивом данных или выборкой $y = (y_1, y_2, \dots, y_n)$. Множество возможных значений (реализаций) выборки образует множество наблюдений Y . [2]

Моделирование объектов на фоне шумов и алгоритмов с постоянным порогом

Программа моделирования наблюдений и алгоритмов обнаружения с постоянным порогом формирует тестовые изображения для исследования характеристик случайных полей, а также для анализа характеристик алгоритмов обработки, в частности, для обнаружения и распознавания объектов на изображениях.

Алгоритм обнаружения с постоянным порогом является простейшим алгоритмом, который преобразует поступающее на вход яркостное изображение (массив наблюдения) в бинарное изображение. Решение о наличии полезного сигнала принимается в каждой точке (пикселе) исходного изображения. [3]

В начале моделирования формируется бинарное изображение (представлено на рис. 5) тестового объекта с параметрами:

- размер окна 256*256 пикселей;
- размер объекта 30*45 пикселей;
- смещение объекта относительно центра окна по оси x: 40 пикселей;
- смещение объекта относительно центра окна по оси y: 60 пикселей.

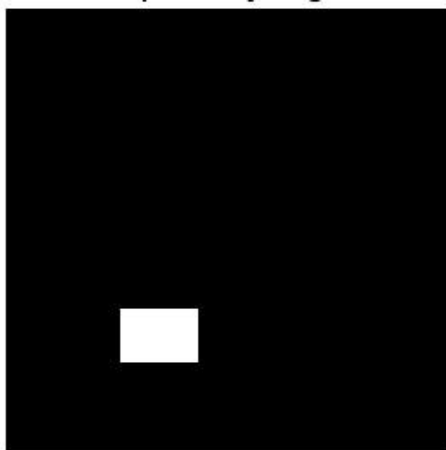


Рис. 5. Формирование бинарного изображения объекта

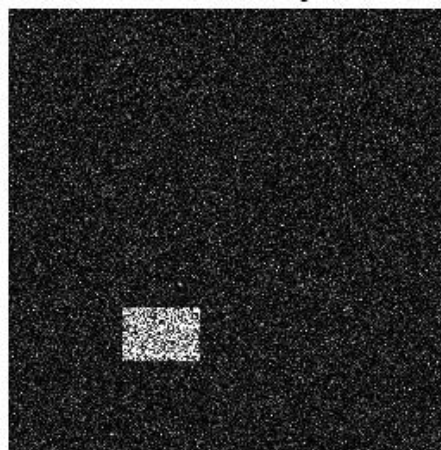


Рис. 6. Результат моделирования объекта на фоне шума

Далее производится вычисление теоретического значения порога обнаружения $thr_0=117,2607$ и порогового значения дефлексии решающей статистики $d=6,6582$ по заданным вероятностям ложной тревоги ($F_0=0,02$) и правильного обнаружения ($D_0=0,6$) производится с использованием критерия Неймана-Пирсона.

Следующим шагом формируются два массива случайных чисел: z_0 и z_d из которых создается массив наблюдения y :

- фоновая область массива наблюдений соответствует массиву z_0 ;
- сигнальная область массива наблюдений соответствует массиву z_d .

Массив наблюдения представлен на рис. 6. Гистограммы, соответствующие экспоненциальному распределению шумов представлены на рис. 7.

Сигнальная область представляет собой квадрат 256*256, параметр шума обеспечивает заданное значение отношения сигнал/шум в каждом пикселе.

Далее при помощи функции для фоновой области массива z_0 вычисляется:

- количество превышений порога $A=1238$;
- размер изображения $N=65536$;
- относительное число превышений порога $F=A/N=0,0189$.

Для обработки фоновой области массива z_0 при использовании критерия Неймана-Пирсона значение порога thr берется равным теоретическому значению порога моделирования thr_0 .

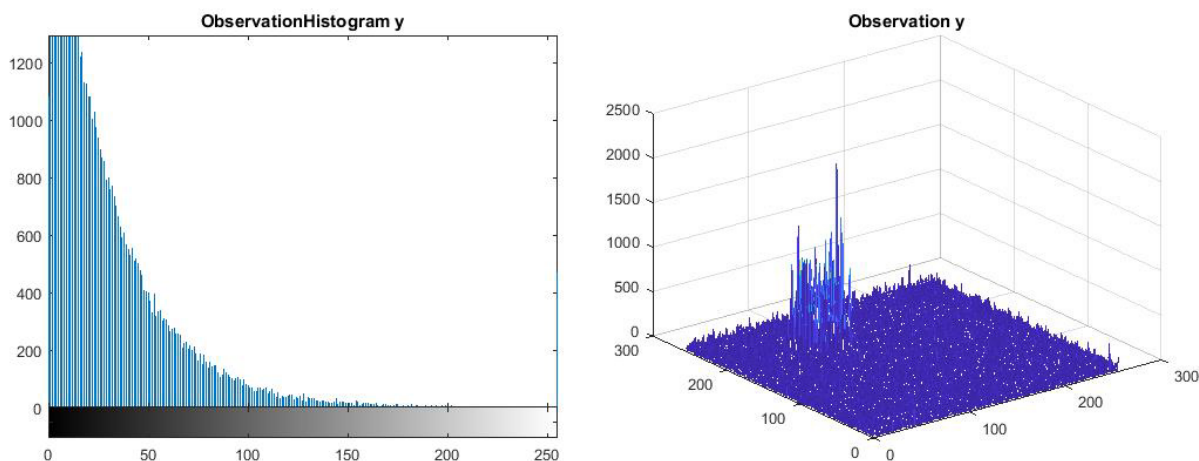


Рис. 7. Гистограммы массива наблюдений с экспоненциальным распределением

На рис. 8 представлен результат фильтрации фоновой области массива z_0 относительно порога thr_0 .

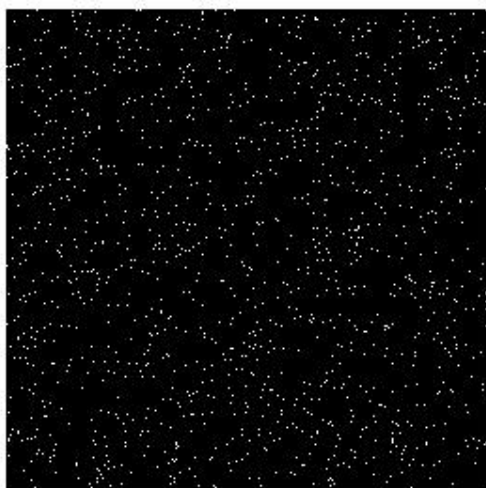


Рис. 8. Результат фильтрации фоновой области массива z_0 относительно порога thr_0

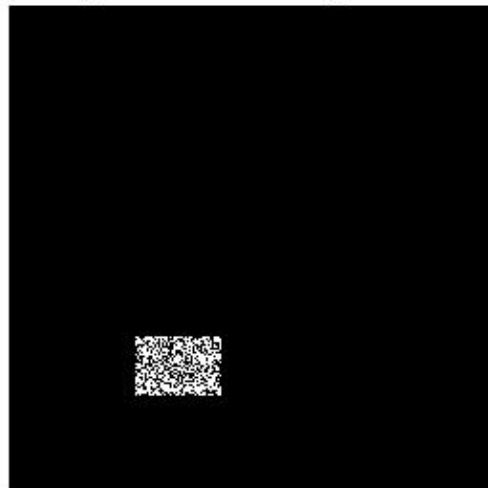


Рис. 9. Результат обработки сигнальной области относительно порога thr_0

После чего производится обработка превышения порога в сигнальной области. Выходная величина функции обработки $D_s=0.5964$ является оценкой вероятности правильного обнаружения по сигнальной области. Результат обработки сигнальной области относительно порога thr_0 представлен на рис. 9.

Завершающим этапом (представлен на рис. 10 (а)) моделирования объекта на фоне шумов и алгоритмов с постоянным порогом является обработка превышения порога по всему изображению. Функция обработки для массива y вычисляет:

- количество превышений порога $C=2038$;
- размер изображения $B=65536$;
- относительное число превышений порога $D=C/B=0,0310$.

В результате обработки массива наблюдения осуществлена фильтрация изображения при помощи алгоритмов с постоянным порогом z_0 при использовании критерия Неймана-Пирсона. На рис. 10 (б) представлен массив наблюдения до обработки и после.

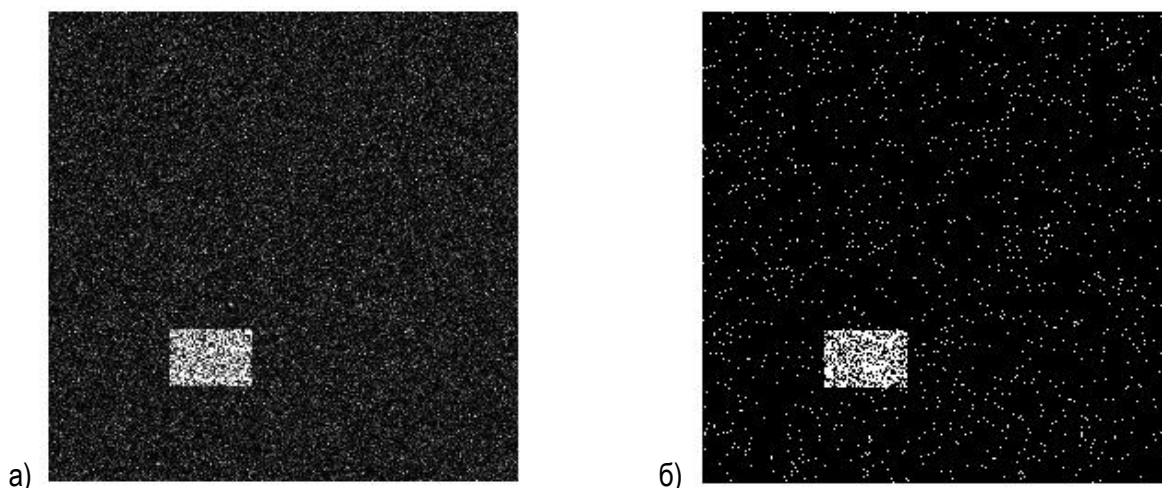


Рис. 10. Массив наблюдения до обработки (а) и после обработки с помощью алгоритма с постоянным порогом (б)

Обработка с постоянным порогом позволила существенно снизить шумы фона, что позволяет увеличить вероятность обнаружения объекта на изображении.

Заключение

Проведенное исследование оптико-электронного мониторинга подтверждает перспективы развития средств мониторинга поверхностей различного класса при помощи МЛА. Представленные методы обнаружения и классификации объектов позволяют на стадии разработки выбрать оптимальный способ для решения конкретной задачи.

Проведенное моделирование объекта на фоне шумов и алгоритма фильтрации с постоянным порогом является простейшим способом фильтрации объекта, который несмотря на простоту существенно улучшил значение полезного сигнала объекта на фоне шума. Для большего увеличения вероятности обнаружения и правильной классификации стоит использовать алгоритмы фильтрации с адаптивным порогом, либо иные способы выделения объектов на фоне шумов, фильтрации и классификации объектов.

Библиографический список

1. Общие виды и характеристики беспилотных летательных аппаратов: справ. пособие / А. Г. Гребеников, А. К. Мялица, В. В. Парфенюк, О. И. Парфенюк, С. В. Удовиченко. URL: www.bookzz.org (дата обращения: 01.04.2020).
2. Обнаружение объектов на изображениях с использованием селекции по площади: статья / В. Ю. Волков.
3. Волков В. Ю. Адаптивные и инвариантные алгоритмы обнаружения объектов на изображениях и их моделирование в MATLAB: учеб. пособие. 2-е изд., доп. СПб.: Лань, 2014. 192 с.

УДК 004.946

И. А. Маханов

студент кафедры проблемно-ориентированных вычислительных комплексов

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

ПЕРСПЕКТИВЫ ВНЕДРЕНИЯ ТЕХНОЛОГИИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ В ПОВСЕДНЕВНУЮ ЖИЗНЬ

Одним из перспективных направлений ИТ-разработок является дополненная реальность. Дополненная реальность (augmented reality, AR — «расширенная реальность») – дополнение физического мира цифровыми данными в режиме online при помощи компьютерных устройств (планшетов, смартфонов и инновационных гаджетов) [1].

Главная идея технологии следующая: в окружающее человека пространство с помощью компьютерных средств добавляются цифровые данные в режиме реального времени, для того чтобы дополнить знания об окружающем нас пространстве или предметах [2].

Задачи дополненной реальности.

- Расширить информационное взаимодействие пользователя с окружением, в виде трёхмерного представления объектов [2].

- Совместить реальный мир и виртуальную реальность с помощью внедрения дополнительных слоев с контекстными изображениями на изображения реальной среды [2].

- Использовать в режиме online информацию, связанную с объектами реального мира.

Рассмотрим предметные области, в которые уже внедрена технология дополненной реальности.

1. Медицина. Клиника Кливленда в Университете Case Western Reserve (<https://case.edu/>) изучает анатомию, используя гарнитуру AR, которая позволяет визуализировать человеческое тело в интерактивном 3Dформате [3].

2. Логистика. Компания DHL (<https://www.logistics.dhl.ru/ru-ru/home.html>) уже внедрила умные AR-очки на некоторых своих складах. Линзы показывают рабочим кратчайший маршрут на складе для поиска и выбора определенных предметов, предназначенных для отправки [3].

3. Строительство и архитектура. AR активно использует компания BNBuilders (<https://www.bnbuilders.com/>) в Сиэтле, чтобы наглядно показать клиентам проект в условиях существующей строительной площадки. Для этого применяются Apple iPad и другие мобильные устройства [4].

В общем случае система дополненной реальности состоит из двух компонентов – сервера и пользовательского приложения [2]. Технологической оснасткой дополненной реальности являются быстрая передача данных от сервера к пользователю, уникальность программного обеспечения, позволяющее в online-режиме получать большой объем необходимой информации.

Для реализации дополненной реальности на устройствах пользователя требуется высокоскоростная передача данных по телекоммуникационным сетям. Поэтому на смену 4G связи приходит связь в формате 5G [3]. Пиковая скорость в таких сетях достигает 20 Гбит/с, а для 4G – 1 Гбит/с. Для подключения широкого класса устройств пользователя, например смартфонов или IoT-компонентов, предусмотрены диапазоны частот 600–6000 МГц и 24–100 ГГц.

По данным AVRA Ассоциации (<http://ar-vr.org/>), сейчас в России более 400 компаний, создающих AR-решения на российском и международном рынке. Размеры некоторых компаний-разработчиков достигают штата в 40-50 человек и обороты в 100 млн руб. За последние 2 года сформировали свои AR-лаборатории такие крупные компании Сбербанк, Ланит, Сибур и др. [4, 5].

Наиболее перспективными направлениями внедрения дополненной реальности в повседневную деятельность следующие.

1. Дизайн интерьера помещения. Дизайнеры уже оценили технологию дополненной реальности, так как она позволяет мгновенно увидеть, как разработка будет смотреться в реальном окружении [1, 6].

2. Развлечения. В настоящее время развлекательные бренды рассматривают AR как отличную маркетинговую возможность для установления более тесных связей между своими героями и аудиторией. На самом деле создатели дополненной реальности PokemonGo скоро планируют выпустить AR-игру на тему Гарри Поттера, с которой фанаты могут взаимодействовать изо дня в день [3, 6].

3. Образование. Дополненная реальность даёт возможность по-другому взглянуть на образование, так как позволяет моделировать реальные ситуации. Например, будущее рабочее место можно спроектировать в учебном классе [6].

4. Ремонт автомобилей. Компания Inglobe Technologies (<https://www.inglobetechnologies.com/>) выпустила приложение, которое помогает автолюбителям определить местонахождение определенных деталей в корпусе автомобиля. Когда эта технология станет более распространенной, даже новичок сможет выполнить базовый ремонт машины, например, проверить уровень жидкости и заменить фильтры [6].

5. GPS-навигация. На армейских самолётах и вертолётах данные выводят на лобовое стекло. Это даёт лётчику возможность постоянно контролировать ситуацию, не отвлекаясь на приборную панель. Эксперты предсказывают, что технология перспективна и в ближайшее время будет внедряться и в гражданских транспортных средствах [6].

Применение технологии AR обещают повысить производительность работы и внести качественные изменения в повседневную жизнь человека. В настоящее время разрабатывают и вводят в эксплуатацию аппаратную поддержку, а также развертывают телекоммуникационные сети в формате 5G для активного внедрения этой технологии через 10-15 лет [1].

Библиографический список

1. Классификация и перспективные направления использования технологии дополненной реальности. URL: <https://cyberleninka.ru/article/n/klassifikatsiya-i-perspektivnye-napravleniya-ispolzovaniya-technologii-dopolnennoj-realnosti/viewer> (дата обращения: 14.05.2020).

2. Digital Trends 2020. URL: <https://www.similarweb.com/corp/reports/2020-digital-trends-report/> (дата обращения: 12.05.2020).

3. Использование дополненной реальности в разных областях. URL: <https://mentamore.com/virtualnaya-realnost/ispolzovanie-dopolnennoj-realnosti.html> (дата обращения: 14.05.2020).

4. Интерфейс будущего – системы дополненной реальности. URL: <https://www.prosoft.ru/cms/f/466284.pdf> (дата обращения: 14.05.2020).

5. В чем разница между 4G и 5G?. URL: <https://habr.com/ru/post/439136/> (дата обращения: 14.05.2020).

6. Тренды AR/VR для бизнеса на 2020 год. URL: https://old.sk.ru/news/b/press/archive/2019/12/27/trendy-ar_2f00_vr-dlya-biznesa-na-2020-god-insayty-s-foruma-avra-days-i-konferencii-mediana.aspx (дата обращения: 14.05.2020).

УДК 004.4.275

В. К. Михайлов

студент кафедры проблемно-ориентированных вычислительных комплексов

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

ПРИМЕНЕНИЕ МНОГОПОТОЧНОГО ПРОГРАММНОГО РАСШИРЕНИЯ DOTS В МЕЖПЛАТФОРМЕННОЙ СРЕДЕ РАЗРАБОТКИ КОМПЬЮТЕРНЫХ ИГР UNITY

Видеоигры являются уникальным видом интерактивного медиаконтента. Высокая скорость работы и отклика на действия пользователя является одним из самых важных показателей качества технического исполнения продуктов этого типа. Для достижения наибольшей производительности для игры крайне важно эффективно использовать ресурсы устройства, на котором она работает. Оптимизация и эффективное использование всех имеющихся ресурсов всегда являлась одной из самых больших проблем при разработке видеоигр. С распространением многоядерных процессоров эта проблема стала еще очевиднее, так как большая часть популярных игровых движков изначально была создана для работы только с одним ядром центрального процессора. Эта проблема до недавнего времени присутствовала и в движке Unity, изначально созданном для создания небольших игр, не требующих использования большого количества аппаратных ресурсов, но со временем ставшим использоваться повсеместно для все более технически продвинутых игр. В последнее время движок начал применяться в приложениях дополненной и виртуальной реальности, где требования к производительности намного выше, чем в обычных играх.

Для упрощения разработки игр и приложений, использующих возможности многоядерных процессоров, компания Unity Technologies разработала новый пакет информационно-ориентированных технологий – DOTS.

Целью работы является обоснование применения технологии UNITY/DOTS при многопоточном программировании.

Таблица 1

Компоненты DOTS

Компонент	Описание
ECS (Entity Component System)	Шаблон проектирования "сущность-компонент-система", основанный на представлении игровых объектов, в виде ограниченного набора данных и последующей манипуляции с ними при помощи систем.
	ECS – это ядро всего пакета DOTS, состоящее из трех основных частей: 1. Сущности (Entities) – это сущности, наполняющие игру или программу. 2. Компоненты (Components) – организованные данные, содержащие информацию о сущностях. 3. Системы (Systems) – логика игры, необходимая для преобразования данных сущностей [1].
C# Job System	Система управления многопоточными вычислениями. Система задач C# предназначена для написания простого безопасного многопоточного кода, взаимодействующего с движком Unity для улучшенной производительности в приложениях [2].
Hybrid Renderer	Инструментарий, необходимый для отображения сущностей на экране. [3] Гибридный рендерер содержит компоненты, необходимые для отображения сущностей на экране, и компоненты, позволяющие преобразовывать стандартные игровые объекты Unity в сущности, что значительно упрощает работу с ними.
Burst Compiler	Компилятор, преобразующий IL/.NET код в высокопроизводительный машинный код, применяя технологии LLVM (Low Level Virtual Machine) [4].

DOTS (Data-Oriented Technology Stack) – это пакет программных средств многопоточных вычислений, разработанный компанией Unity Technologies для межплатформенной среды разработки компьютерных игр Unity.

DOTS предназначен для упрощения создания и оптимизации многопоточных приложений.

DOTS состоит из 4 основных компонентов, представленных в табл. 1.

Основным критерием сравнения с традиционным способом программирования в Unity выбрано время, необходимое на обработку группы объектов в каждый кадр.

Данные сравнения взяты из доклада о переводе Top-down шутера на DOTS с конференции разработчиков Unite Copenhagen 2019 [5].

Скорость работы приложения до применения технологий DOTS представлена в табл. 2.

Таблица 2

Производительность до применения DOTS

	Количество объектов (шт.)	Время (мс)	Увеличение производительности (раз)
без DOTS	2000	9	1

Первым шагом было применение ECS. Для этого было необходимо определить минимальный набор данных, необходимый для представления сущности в игре, и создать архетип для этой сущности. Архетип – это уникальный набор компонентов, характеризующий объект (рис.1).

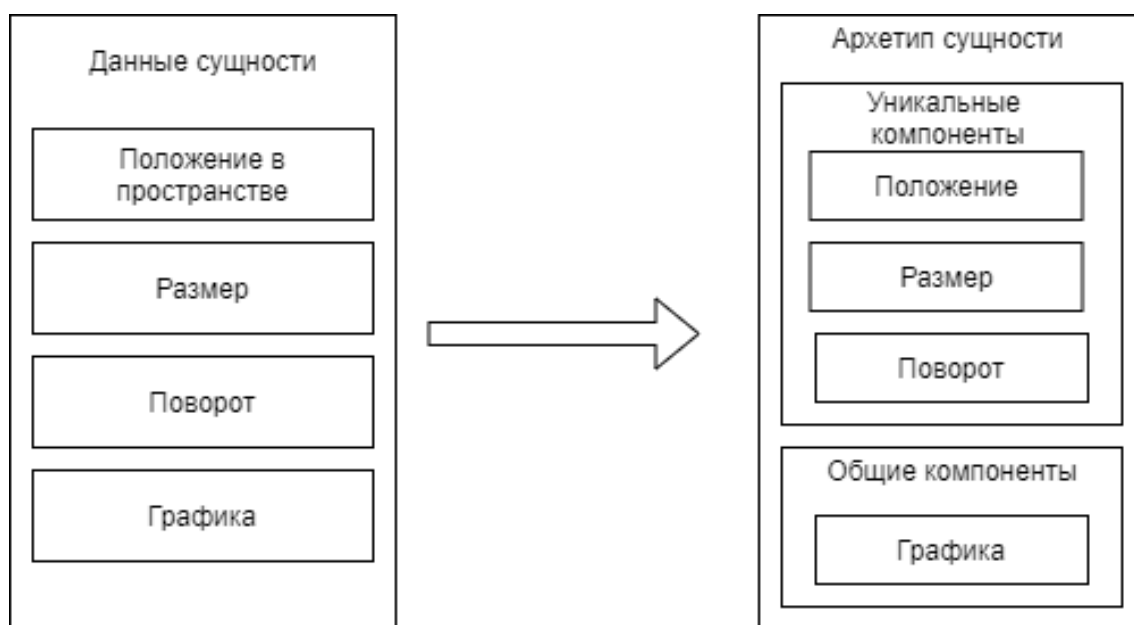


Рис. 1. Архетип сущности

Архетип определяет, где ECS хранит компоненты конкретной сущности. ECS выделяет память кусками – чанками, каждый чанк является объектом архетипа. Каждый такой кусок памяти всегда хранит сущности одного архетипа (рис.2).

Для произведения манипуляций с сущностями необходимо создать систему, содержащую логику поведения сущности, которая соединяется с чанками архетипов через запрос (рис.3).

В сравнении с исходным результатом, внедрение ECS дало прирост производительности в 9 раз (табл. 3).

Следующим шагом было применение системы задач. Для этого было необходимо организовать имеющиеся задачи в очередь задач, которые система задач впоследствии распределяет между ядрами процессора (рис.4.). Стоит также заметить, что прирост производительности зависит от количества ядер центрального процессора, чем больше ядер, тем меньше задачи будут ожидать в очереди.

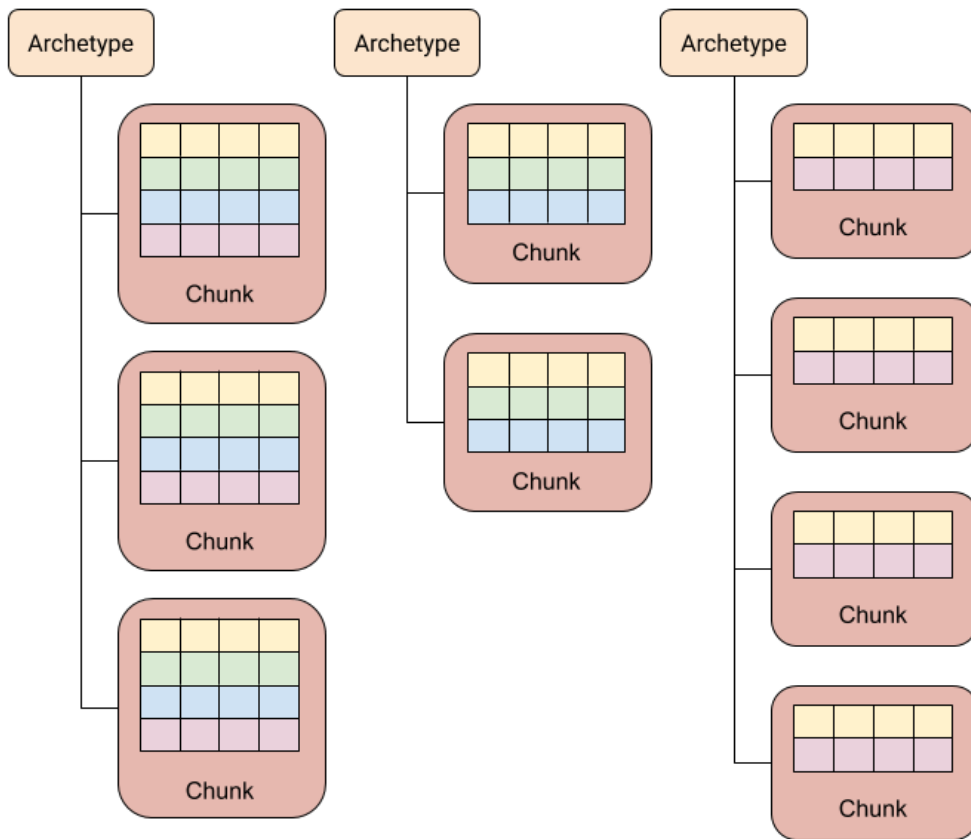


Рис. 2. Выделенная под архетипы память

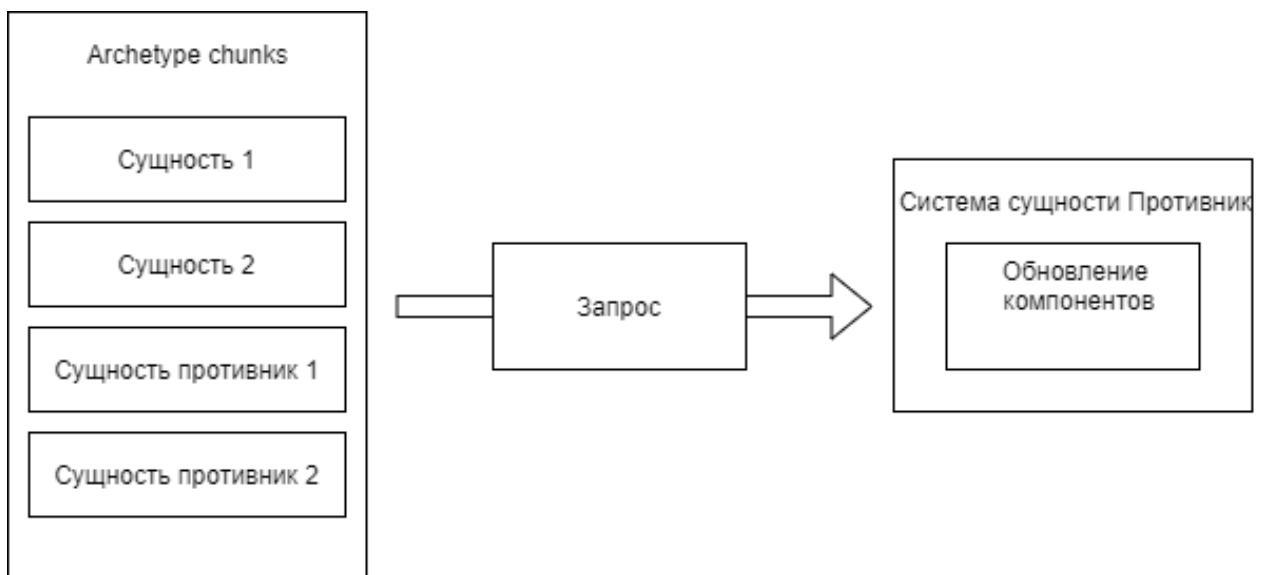


Рис. 3. Система сущности

Таблица 3

Производительность после применения ECS

	Количество объектов (шт.)	Время (мс)	Увеличение производительности (раз)
ECS	2000	1	9

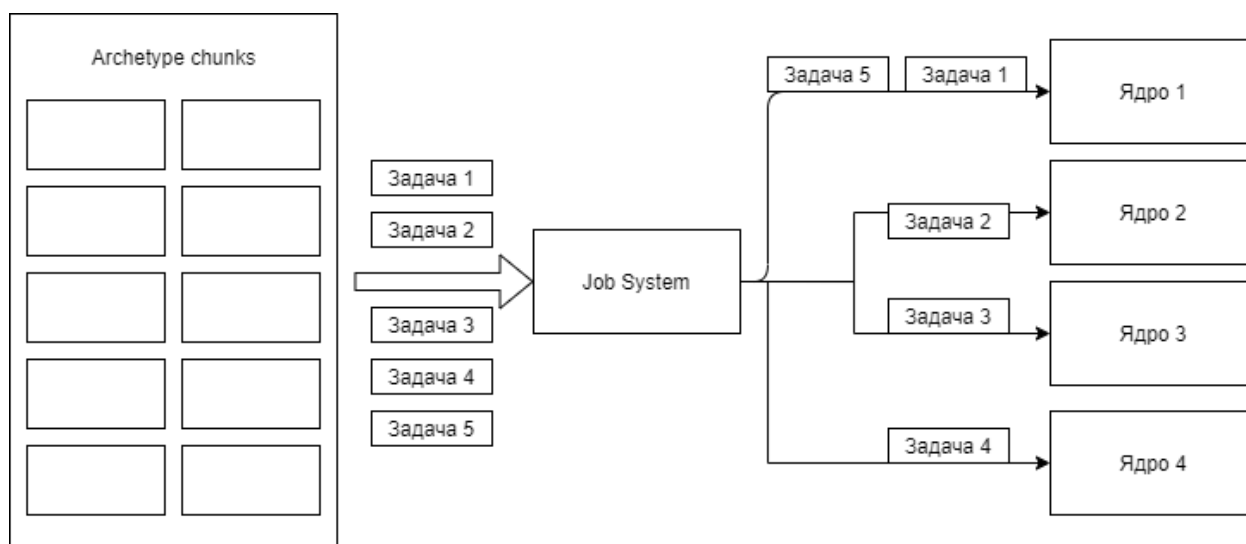


Рис.4. Job System

В сравнении с исходным результатом, применение ECS и системы задач дало прирост производительности в 45 раз (табл. 4).

Таблица 4

Производительность после применения ECS и системы задач

	Количество объектов	Время (мс)	Увеличение производительности (раз)
ECS + Job System	2000	0,2	45

Последним шагом было применения компилятора Burst к коду, использующему ECS.

В сравнении с исходным результатом, применение комбинации ECS , системы задач и компилятора Burst дало прирост производительности в 2250 раз (табл. 5)

Таблица 5

Производительность после применения ECS и системы задач и компилятора Burst

	Количество объектов (шт.)	Время (мс)	Увеличение производительности (раз)
ECS + Job System + Burst Compiler	20000	0,04	2250

Такой прирост производительности достигается за счет способности компилятора создавать код, в котором процессор может выполнять несколько однотипных инструкций сразу. В сочетании с тем, что ECS представляет все объекты как однотипные наборы данных, это дает большой прирост производительности. Финальные результаты применения технологии DOTS представлены в табл. 6.

Таблица 6

Финальные результаты

	Количество объектов (шт.)	Время (мс)	Увеличение производительности (раз)
без DOTS	2000	9	1
ECS	2000	1	9
ECS + Job System	2000	0,2	45
ECS + Job System + Burst Compiler	20000	0,04	2250

Из полученных результатов можно сделать вывод, что применение DOTS позволяет легко получить прирост производительности при обработке очень больших количеств однотипных объектов. Данный результат достигается благодаря тому, что DOTS позволяет очень эффективно оперировать минимально необходимым набором данных и распределять их обработку по разным ядрам процессора. Применение компилятора Burst в связке с ECS с Job system выводит производительность на качественно новый уровень. Также данный прирост производительности может оказаться чрезвычайно полезным для приложений виртуальной реальности, которым необходима очень высокая и стабильная частота кадров. На данный момент польза технологии наиболее велика для видеоигр, но в будущем она может найти свое применение и в других сферах, требующих обработки и вывода на экран большого количества объектов в реальном времени, например, в индустрии кино, где постепенно начинают внедряться технологии рендеринга окружения в реальном времени.

Библиографический список

1. Entity Component System. URL: <https://docs.unity3d.com/Packages/com.unity.entities@0.8/manual/index.html> (дата обращения: 15.03.2020).
2. C# Job System. URL: <https://docs.unity3d.com/2019.3/Documentation/Manual/JobSystem.html> (дата обращения: 15.03.2020).
3. Hybrid renderer. URL: <https://docs.unity3d.com/Packages/com.unity.rendering.hybrid@0.1/manual/index.html> (дата обращения: 15.03.2020).
4. Burst Compiler. URL: <https://docs.unity3d.com/Packages/com.unity.burst@1.3/manual/index.html> (дата обращения: 15.03.2020).
5. Unite Copenhagen 2019. URL: https://www.youtube.com/watch?time_continue=1176&v=yTGhg905SCs&feature=emb_logo (дата обращения: 15.03.2020).

УДК 519.218

К. А. Мозырева

магистрант кафедры проблемно-ориентированных вычислительных комплексов

С. Д. Шапорев – доктор физико-математических наук, профессор – научный руководитель

ПРИМЕНЕНИЕ ВЕТВЯЩИХСЯ ПРОЦЕССОВ

Теория ветвящихся процессов позволяет моделировать эволюцию таких объектов, как гетерозиготы, которые являются носителями мутантного гена, или клиенты, ожидающие в системе очереди, или нейтроны в ядерном реакторе. В простейших ветвящихся процессах время измеряется дискретно во всех поколениях и объекты относятся к одному типу. Каждый объект рассматривается, как живущий в одном поколении, в течении этого периода он производит другие объекты, тем самым внося вклад в следующее поколение. Предполагается, что объекты размножаются независимо по одному и тому же вероятностному закону, при этом взаимодействия между объектами запрещено.

Среди основных сфер применения выделяются следующие:

1. Эпидемии. Ранние стадии распространения эпидемий могут быть легко смоделированы с помощью ветвящихся процессов. Если число инфицированных людей мало, тогда они точно действуют независимо друг от друга. Вероятности $p_k, k = \overline{1, n}$ отражают с какой вероятностью один инфицированный заразит еще больше людей, пока не умрет или полностью не излечится.

2. Реакция нейтронной цепи. В термоядерном реакторе свободный нейтрон начинает цепную реакцию, ударяя и расщепляя ядро. Обычно, это приводит к образованию случайного количества вторичных нейтронов. Эти нейтроны либо поглощаются, либо ударяют другие ядра и высвобождают третичные нейтроны и так далее. Ядерный реактор тщательно моделируется путем отвода избыточных нейтронов для поддержания критического состояния и предотвращения взрыва [1].

3. Сохранение семейной фамилии. Во многих культурах, фамилии передаются по мужской линии. Потомки мужского пола данного мужчины представляют собой ветвящийся процесс. Впервые феномен вымирания фамилий был освещен в работах Гальтона и Ватсона [2]. Получена вероятность исчезновения фамилии благородных семей, которые были основаны в 10-12 веках. Выводом являлось то, что фамилия знатной семьи вымрет с большей вероятностью, чем у обычной.

4. Системы массового обслуживания (СМО). Использование ветвящихся процессов в моделировании СМО позволяет не только находить стационарные характеристики, но и изучать асимптотическое поведение при различных прогнозах. Также возможно получение предельных теорем в условиях различной нагрузки, в том числе в перегруженных системах.

Рассмотрим одну из простейших математических моделей ветвящихся процессов.

Предположим, что в момент времени $t = 0$ имеется одна частица в нулевом поколении $X_0 = 1$,

которая в некоторый момент времени порождает k частиц с вероятностью $p_k, k = \overline{1, n}$, $\sum_{k=0}^{\infty} p_k = 1$. По-

рожденные частицы образуют первое поколение $X_1 = 4$. Каждая новая частица действует независимо от прошлого состояния других частиц. Такой ветвящийся процесс можно представить в виде дерева (рис. 1).

Считаем, что каждая частица живет единицу времени. Пусть имеются независимые последовательности независимых случайных величин $\{Y_n^{(1)}\}, \{Y_n^{(2)}\}, \dots, n \geq 1$, где при всех j и n :

$$P\left(Y_n^{(j)} = k\right) = p_k, \quad k = \overline{0, n}$$

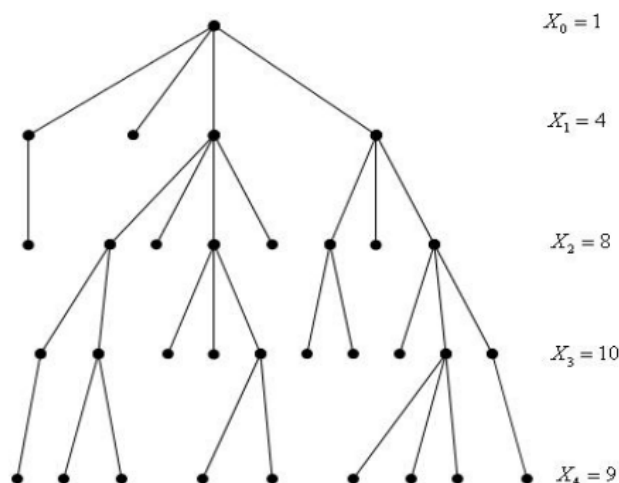


Рис.1. Рост поколения в процессе ветвления

Смысл в том, что случайные величины равны числу потомков частиц из $(j - 1)$ -го поколения, эти потомки формируют j -е поколение. Последовательность X_n можно представить в виде

$$\begin{aligned}
 X_0 &= 1, \\
 X_1 &= Y_1^{(1)}, \\
 X_2 &= Y_1^{(2)} + \dots + Y_{X_1}^{(1)}, \\
 &\dots \dots \dots \\
 X_n &= Y_1^{(n)} + \dots + Y_{X_{n-1}}^{(n)}.
 \end{aligned}$$

Построенная модель называется процессом Гальтона-Ватсона.

Для исследования ветвящихся процессов используются производящие функции. Введем производящую функцию числа частиц в n -м поколении:

$$g_n(z) = \sum_{k=0}^{\infty} z^k P(X_n = k)$$

Основным вопросом при изучении ветвящегося процесса является распределение числа частиц в n -м поколении и вероятность вырождения процесса в какой-то момент времени. Если $X_n = 0$, при некотором n , тогда данный процесс является вырожденным.

Вероятность вырождения r равна наименьшему корню уравнения $z = g(z)$ на отрезке $[0, 1]$ [3].

Приведем пример. Пусть $\{X_n, n \in \mathbb{Z}\}$ – ветвящийся случайный процесс с законом размножения частиц ξ с геометрическим распределением (т.е. $P(\xi = k) = p(1-p)^k, k \in \mathbb{Z}, p \in (0, 1)$). Найти:

- вероятность вырождения;
- $\rho(z) = \lim_{n \rightarrow \infty} g_{X_n}(z)$;
- $P(x = k)$ для $\forall k \in \mathbb{Z}$

Найдем производящую функцию:

$$g_{\xi}(z) = Ez^{\xi} = \sum_{k=0}^{\infty} z^k p(1-p)^k = \frac{p}{1-z(1-p)}$$

Тогда вероятность вырождения равна

$$r = g_{\xi}(r) = \frac{p}{1-r(1-p)} \Leftrightarrow r - r^2(1-p) = p \Leftrightarrow (1-p)(r-1) \left(r - \frac{p}{1-p} \right) = 0$$

Если $\frac{p}{1-p} < 0$, то $r = \frac{p}{1-p}$. Если $\frac{p}{1-p} > 1$, то $r = 1 \Rightarrow$ процесс будет вырожденным.

Найдем $\rho(z)$

$$\rho(z) = zg_{\xi}(\rho) = \frac{z\rho}{1-(q-p)\rho} \Leftrightarrow (1-p)\rho^2 - \rho + zp = 0$$

$$\rho(z) = \frac{1 \pm \sqrt{1-4zp(1-p)}}{2(1-p)} \Leftrightarrow \rho(z) = \frac{1 - \sqrt{1-4zp(1-p)}}{2(1-p)}, \text{ так как } \rho(0) = 0, \rho(1) = r$$

Зная, что $\rho(z) = \sum_{k=0}^{\infty} z^k P(x=k)$ можем разложить функцию $\rho(z)$ в ряд по степеням z . Тогда

$$P(x=k) = -\frac{1}{2(1-p)} (-4p(1-p))^k C_{\frac{1}{2}}^k.$$

Библиографический список

1. Ватурин В. А. Ветвящиеся процессы и их применения. М.: МИАН, 2008. 107 с.
2. Ватутин В. А., Дьяконова Е. Е. Ветвящиеся процессы Гальтона–Ватсона в случайной среде. I: Предельные теоремы // Теория вероятности и ее применение. 2003. Т. 48, № 2. С. 300.
3. Севастьянов Б. А. Ветвящиеся процессы. М.: Наука, 1971. 436 с.

УДК 339.138:004.8

В. А. Орлянская

магистрант кафедры прикладной информатики

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

ТЕХНОЛОГИИ РАЗРАБОТКИ МАРКЕТПЛЕЙСОВ

Современные покупатели научились защищаться от навязчивой рекламы, они записывают интересные шоу и блокируют ненужную информацию, то есть стремятся принимать решения самостоятельно. Согласно статистике 80 % людей хотя бы один раз совершали покупки онлайн, а 5 % делали это больше одного раза. Около 71 % населения убеждены, что заказать и купить товары или услуги выгоднее через Интернет, чем в магазине [1]. Рынок электронной коммерции сегодня является одним из наиболее стремительно растущих, поэтому очень быстро приобрели популярность различные интернет-магазины, торговые платформы, где можно ознакомиться с информацией о товаре и с отзывами предыдущих покупателей. Торговым организациям, чтобы быть в русле времени, надо идти на компромиссы, менять собственные правила, перестраивать принципы ведения бизнеса и осваивать новые пути реализации товаров и услуг.

Современные продавцы, развивая свой бизнес, пытаются создать ресурс в сети Интернет, с помощью которого было бы возможно наладить эффективную коммуникацию с широким кругом клиентов, осуществлять онлайн-платежи, получать оперативную рыночную информацию. Современным методом такого взаимодействия с клиентами является маркетплейс. Маркетплейс (от англ. Marketplace — «рыночная площадь») — это торговая площадка в Интернете, которая помогает продавцам и потенциальным покупателям находить друг друга онлайн и взаимодействовать между собой [3]. Сегодня все чаще бизнес начинает выходить на маркетплейсы, потому что это новые возможности для повышения продаж. Эта тенденция касается даже крупных и уже популярных интернет-магазинов со сложившимся имиджем.

Маркетплейсы развиваются в единстве с маркетинговой теорией, которая, как правило, запаздывает за появлением новых каналов и инструментов в интернет-среде [2].

Целями статьи является исследование понятия «маркетплейс», рассмотрение способов разработки маркетплейсов, а также освещение его основных преимуществ и особенностей.

Задачами статьи является:

- Провести сравнение маркетплейсов;
- Провести классификацию маркетплейсов;
- Рассмотреть технологию разработки маркетплейсов;
- Рассмотреть наиболее распространенные виды парсеров;
- Рассмотреть популярные виды парсеров
- Рассмотреть функционал работы парсеров

Сайты-агрегаторы всегда предлагают необходимые и нужные функции. Обычный пользователь, находясь в выборе услуги или товара просматривает огромное количество сайтов и сравнивает многочисленные предложения. Эти сайты предлагают полноценные каталоги и разнообразные рейтинги, фильтры и другие возможности для пользователя, которые решают его задачу поиска наиболее эффективно.

Систематизация преимуществ маркетплейса важна для таких субъектов деятельности, как:

1) потребители:

- более низкие цены по сравнению с традиционными магазинами,
- доступность информации о товарах, услугах в интернет-магазинах в режиме реального времени.

2) производители:

- снижение объема первоначальных вложений в бизнес, непроизводственных расходов,
- уменьшение численности персонала и фонда оплаты труда,
- сокращение цикла производства и продажи, поддержка бизнес-процессов в режиме онлайн.



Рис. 1. Интернет-торговля в России, прогноз на 2016–2023 гг.

Маркетплейс создается для большого числа поставщиков и потребителей. Участники могут взаимодействовать на равных (когда обе стороны — частные лица или представители бизнеса) либо выстраивать иерархию «предприниматель-клиент». Существует 3 базовые модели (табл.1):

- C2C (customer-to-customer) — торгуют или предлагают услуги обычные люди, не предприниматели.
- B2C (business-to-customer) — бизнес продает людям.
- B2B (business-to-business) — бизнес покупает у бизнеса.

Таблица 1

Базовые модели маркетплейса

Тип	Описание	Пример
C2C (customer-to-customer)	Физические лица, которые взаимодействуют друг с другом на равных	DogVacay, Bla Bla Car
B2C (business-to-customer)	Физические лица, но в роли поставщиков выступают предприниматели	Amazon, Booking.com, ZocDoc
Смешанная модель B2C и C2C	Маркетплейсы, в которых представлены предложения как от физических лиц, так и от предпринимателей	Airbnb
B2B (business-to-business)	Связывает предпринимателей и позволяет им находить удачные решения для своего бизнеса	Indiamart, Capterra

В зависимости от того, где клиент получает услугу (онлайн или офлайн), выделяют 2 типа маркетплейсов (табл. 2).

Основным элементом маркетплейса является парсер, который напрямую работает с БД, наполняя и обновляя её (рис. 2). Таким образом маркетплейс получает сведения о новых товарах и об обновлении цен на старые товары.

Таблица 2

Маркетплейсы онлайн и оффлайна

Online to offline	Online commerce
Маркетплейс привлекает пользователей в сети, а услуга пользователь получает офлайн. Например, вы выбрали товары в интернете, а забрали в магазине; или нашли человека, который согласился прогулять вашу собаку, и он приехал к вам домой.	Все происходит онлайн, в офлайне только доставка товаров, но и ее может не быть. Например, если речь о маркетплейсах по продаже билетов. Сессию можно завершить онлайн, получив на почту электронные квитанции.

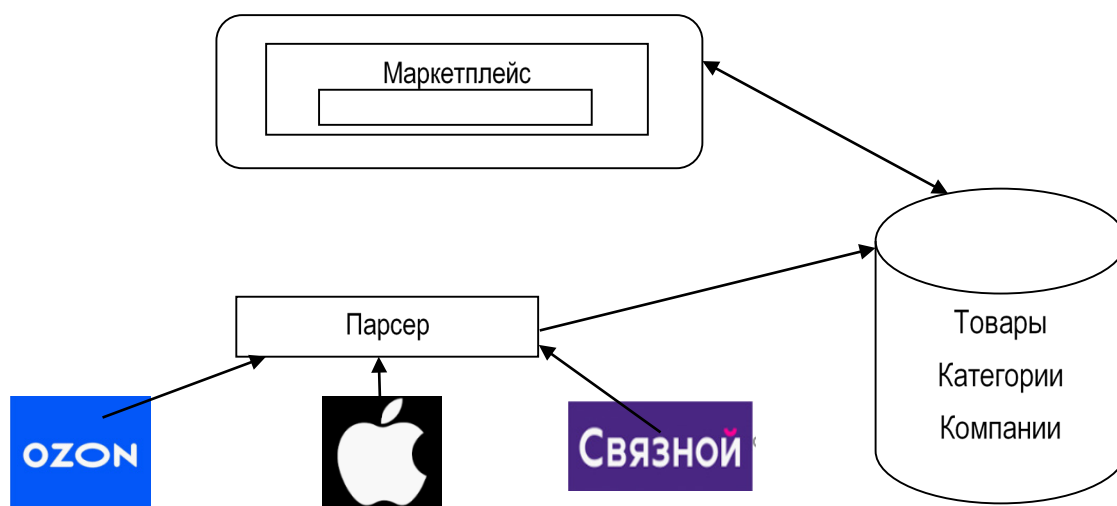


Рис. 2. Автоматизация поиска и заполнения базы данных

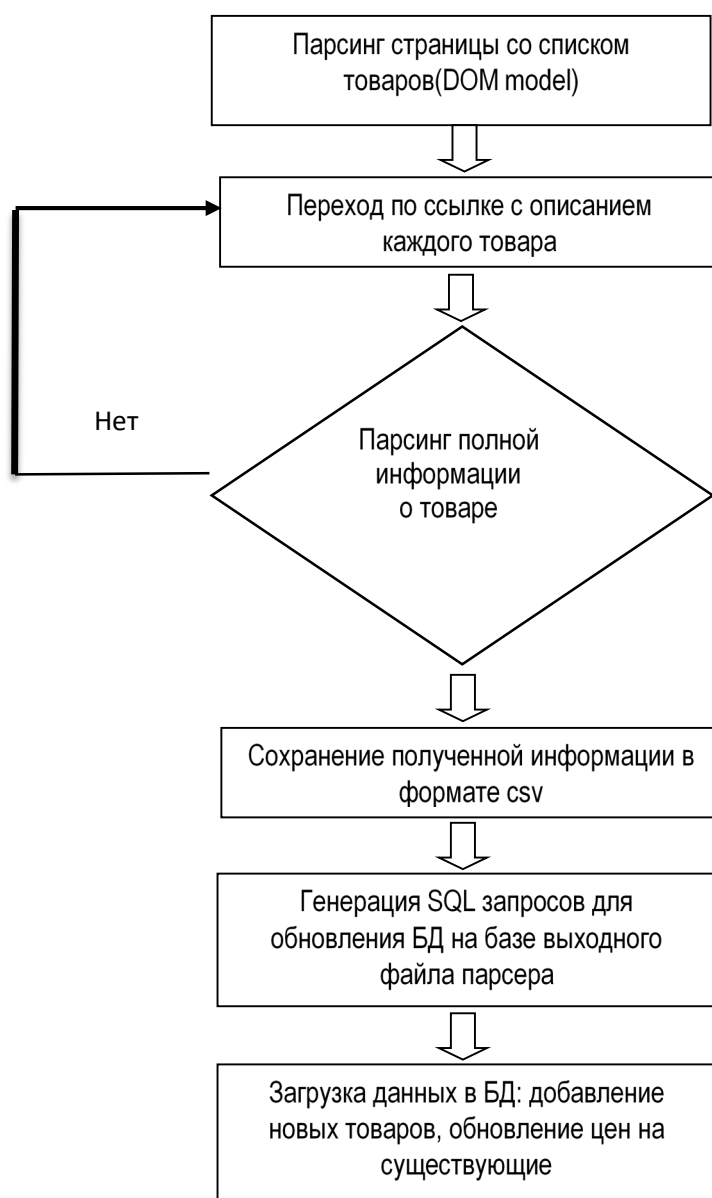


Рис. 3. Алгоритм работы парсера

Под парсингом подразумевается действие, в ходе которого конкретный документ анализируется с точки зрения синтаксиса и лексики. Он преобразовывается; если в нем выявлены искомые сведения, они выбираются для последующего применения.

Для скорой обработки информации применяется парсинг. Так называют поочередную синтаксическую оценку данных, размещенных на интернет-страничках. Данный способ применяется для своевременного обрабатывания и копирования большого количества информации, если ручной труд требует много времени.

Рассмотрим алгоритм добавления информации о новом товаре (рис. 3). Входными данными для компонента «Парсер» являются ссылки на страницы сайтов со списками товаров. На втором шаге работы парсера осуществляется выгрузка документа по ссылке с детальным описанием товара для последующего разбора. Третий шаг алгоритма делает разбор страниц с детальной информацией. На 4 шаге вся полученная информация сохраняется в файле формата csv. На 5 шаге работы алгоритма генерируются sql-запросы для обновления БД. Затем осуществляется поиск такого товара и проверка, был ли обновлен, удален или отсутствует в БД. На последнем шаге алгоритма sql-скрипт запускается, и все данные будут обновлены. Данные базы используются для наполнения контента маркетплейса.

Входными данными для компонента «Парсер» являются ссылки на страницы сайтов со списками товаров. На втором шаге работы парсера осуществляется выгрузка документа по ссылке с детальным описанием товара для последующего разбора. Третий шаг алгоритма делает разбор страниц с детальной информацией. На 4 шаге вся полученная информация сохраняется в файле формата csv. На 5 шаге работы алгоритма генерируются sql-запросы для обновления БД. Затем осуществляется поиск такого товара и проверка, был ли обновлен, удален или отсутствует в БД. На последнем шаге алгоритма sql-скрипт запускается, и все данные будут обновлены. Данные базы используются для наполнения контента маркетплейса.

Сравнение парсеров по поиску и сбору информации в веб-системах показано в табл.3.

Таблица 3

Сравнение парсеров

Парсер	Назначение	Достоинства	Недостатки
Content Downloader	Парсинг товаров интернет-магазина в таблицу CSV. Парсинг статей с картинками и файлами (например, файлами торрентов, флеш-игр) в txt или html форматы.	Большой функционал; не сложный интерфейс для работы с парсером; поддержка разработчиков.	Платный, с разными видами лицензии, нет возможности пробного периода. Сложная настройка, необходимы обучающие уроки.
Datacol	Коммерческий универсальный парсер контента. Является самым популярным и поддерживаемым парсером в интернете. Оконное десктопное приложение.	Большое количество плагинов, за счет которых парсер является универсальным; функционал можно расширить; многопоточность; стабильная поддержка разработчиками полезные функции.	Платная программа; сложная настройка, требует достаточно много времени; запуск из командной строки невозможен. использование парсера доступно только на OS Windows.
Html Agility Pack	Позволяет делать разбор HTML из строки, интернет ресурса, из файла, а также из браузера. интерфейсам DOM.	Универсальный, бесплатный, поддержка XPath.	Ограниченная функциональность.

Парсеры обладают различным функционалом, удобностью и ценой. Выбор парсера зависит от требований заказчика, Рассмотрим базовый функционал парсера [4]:

- Поиск битых ссылок.
- Указание входящих и исходящих ссылок страницы.
- Указание типа и цепочки редиректов.
- Возможность фильтровать страницы.
- Парсинг отдельных URL.
- Выявление дублей Title, Description.
- Указание уровня вложенности.
- Указание заголовков и их количества.
- Код ответа сервера.
- Возможность смены User Agent.
- Title и его длина.
- Description и его длина.
- Keywords и его длина.
- Тип контента.
- Парсинг сайтов, требующих авторизации.

Кроме функциональных возможностей при выборе парсера обращают внимание его производительность и дружелюбность интерфейса по отношению к пользователю.

Таким образом, в статье проведен обзор наиболее востребованных программ по сбору и обработке информации. Рассмотрены технология разработки маркетплейсов и алгоритм автоматического добавления новой информации в базу данных. Раскрыт базовый функционал, присущий парсерам.

Библиографический список

1. Статистика интернет-покупателей. URL: <https://journal.tinkoff.ru/e-commerce-stat/> (дата обращения: 07.04.2020).
2. Язык XML. Объектно-ориентированное программирование. URL: <http://bourabai.kz/xml/parsers.htm> (дата обращения: 09.04.2020).
3. Лучшие бесплатные и платные инструменты для парсинга сайтов и товаров. URL: <https://xmldatafeed.com> (дата обращения: 09.04.2020).
4. Инструменты для парсинга информации. URL: <https://habr.com/ru/post/340038/> (дата обращения: 08.04.2020).

УДК 681.518.3::379.827

П. А. Rogozin

студент кафедры проблемно-ориентированных вычислительных комплексов

В. В. Бульба

студент кафедры проблемно-ориентированных вычислительных комплексов

Н. Н. Григорьева – старший преподаватель – научный руководитель

ИССЛЕДОВАНИЯ ВАРИАНТОВ РЕАЛИЗАЦИИ АВТОМАТИЧЕСКОЙ СИСТЕМЫ ОРОШЕНИЯ РАСТЕНИЙ

Устройство, позволяющее осуществлять полив растений автоматически, без контроля человека, необходимо всегда, а сейчас, во время карантинных мер, особенно актуально. Системы автополива существуют давно, их разновидностей много, но, во-первых, цена изделий далеко не всегда устраивает покупателя, во-вторых, нет большого разнообразия домашних, средних по габаритам электронных устройств. В данной работе рассмотрены варианты реализации средних по габаритам, предназначенных для дома или офиса, систем автополива на базе контроллера.

Порядок работы и функциональная схема работы системы автополива представлены на рис. 1. Эта схема является универсальной, она была использована для проектирования обоих макетов, описанных в данной работе, так как их функции и порядок работы идентичны.

Информацию о влажности почвы объекта полива (ОП) и уровне воды в резервуаре (ИВ) микроконтроллер (МК) получает с помощью датчика влажности (ДВ) и датчика уровня воды (ДУВ) соответственно. По результатам опроса датчиков и с учетом прописанной в МК программы принимается решение о дальнейших действиях: выдаются управляющие сигналы через коммутирующее устройство (КУ) на исполнительное устройство – помпу, либо МК уводится в спящий режим.

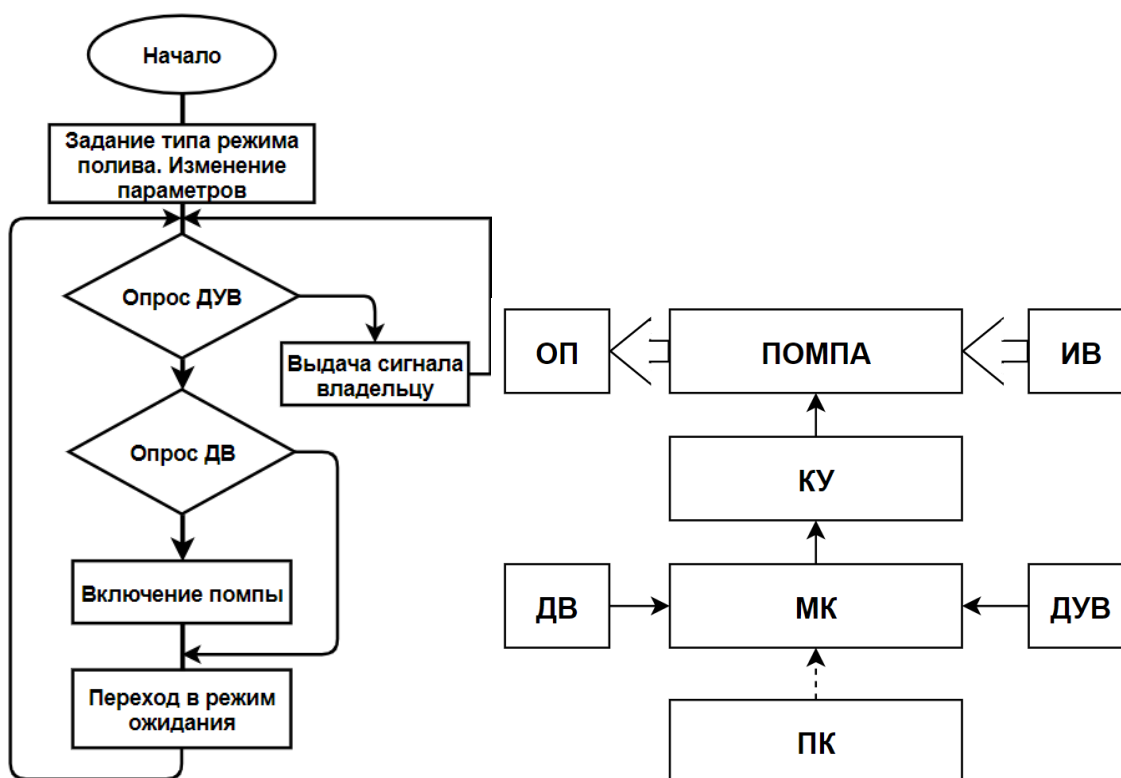


Рис. 1. Функциональная схема и алгоритм работы устройства

Таким образом, МК за один цикл сначала «просыпается», затем проводит опрос датчиков, далее принимает решение – полить растение или нет, и «засыпает» обратно. При недостаточном уровне воды в резервуаре МК посылает соответствующее сообщение хозяину. Под «засыпает» имеется в

виду специальный режим работы МК, при котором энергопотребление значительно уменьшается. При задании режима полива с персонального компьютера (ПК) пользователя производится установка:

- времени сна (частота включения системы), меняется в зависимости от типа растения, которое мы поливаем;
- значений граничных уровней показаний ДВ, определяющих принятие решения о поливе,
- максимального времени полива.

Каждый элемент функциональной схемы может быть реализован разными вариантами компонентов, но нас главным образом интересовали варианты выбора исполнительных устройств – в основном качественно есть смысл менять лишь КУ и помпу.

Выбор элементной базы зависит от поставленной задачи: поливать небольшую площадь (рассада, цветы на подоконнике) или же орошать почву, к примеру, в теплице. В первом случае КУ должен подключать одну помпу, во втором, может – все десять, то есть существенно меняется его мощность.

Первый макет создавался как дешевый прототип устройства для автополива одного-двух однотипных растений, отвечающий требованиям низкой стоимости, низкого энергопотребления, малых габаритов и простоты применения. Второй макет – ориентирован на автополив небольшой теплицы, это более мощное устройство с достаточно высокой производительностью, но все еще малой стоимостью [1].

В первую очередь был выбран контроллер, мы рассматривали МК от STM и Arduino. Первый слишком хорош для такой простой задачи, его возможности избыточны. Второй вполне подходит, так как, во-первых, стоимость МК небольшая, во-вторых, Arduino – достаточно распространенный МК, для которого есть большое количество информации, стандартных решений и дополнительных модулей. Из всего разнообразия Arduino мы остановились на Uno и Nano. После анализа энергопотребления мы пришли к выводу о целесообразности использования Arduino Uno, так как он экономнее [2].

Немаловажным является выбор датчика влажности почвы. Его можно сделать самому или же просто купить готовый, как мы и поступили. Главный минус покупного состоит в том, что его тонкие металлизированные дорожки достаточно быстро подвергаются окислению, вследствие чего датчик выходит из строя. Также у него короткие электроды, из-за этого такой датчик не подойдет для измерения влажности почвы открытого грунта – мы не получим достоверной информации о состоянии земли на глубине.

В качестве исполнительного устройства, помпы, для первого макета была выбрана пятивольтовая помпа, ее габариты и мощность в этом случае вполне достаточны.

Для второго макета была использована помпа от стиральной машины. Недостатков у неё два – это питание от сети 220 Вольт, что требует протягивать электрическую линию для питания помпы, а также потенциальная опасность. В целях повышения безопасности в электрическую цепь питания исполнительного устройства, до помпы, необходимо ставить электрический автомат, который отключит ее питание в случае скачка тока, попадания воды на электронную часть, находящуюся под высоким напряжением, обрыва провода питающего провода или короткого замыкания, вызванного иными случаями антропогенного или биотического характера.

Подключать помпу сразу к микроконтроллеру нельзя, так как высокие токи в момент коммутации исполняющего устройства могут «убить» МК, ну а в случае с мощной помпой, требующей высокого напряжения для питания, это просто невозможно.

Далее мы выбрали соответствующие по мощности коммутирующие устройства для каждого макета: для первого это транзисторный ключ, для второго реле. В первом проекте был использован полевой транзистор типа MOSFET из-за его высоких ключевых характеристик. Он имеет в разы большую скорость переключения, чем биполярный транзистор, и потребляет меньший ток. Высокое значение тока пробоя полевого транзистора позволяет быть уверенным в том, что коммутирующее устройство в ходе работы не выйдет из строя.

Во втором макете в качестве КУ используется реле. Реле бывают разных видов, но мы остановились на электромагнитное реле, так как оно является самым доступным. Это устройство подходит для подключения мощных устройств и с микроконтроллером ничего не случится, так как две электрические цепи: цепь микроконтроллера и цепь нагрузки никак не пересекаются. Стоит отметить, что

можно использовать реле и для коммутации маломощного потребителя, оно будет вполне хорошо справляться с этой задачей.

Габариты получившихся макетов существенно отличаются из-за различий в размерах устройств подачи воды, помпах.

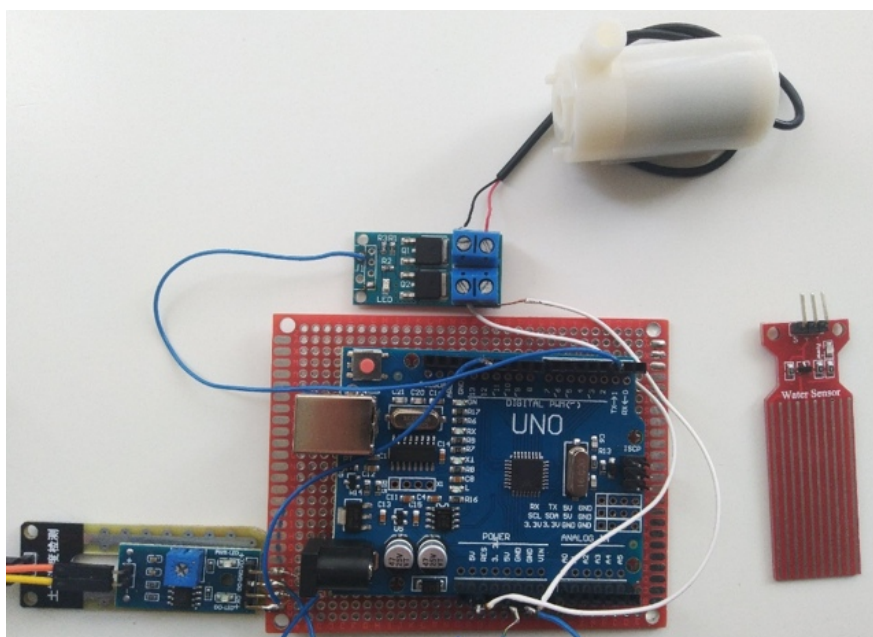


Рис. 2. Внешний вид первого макета

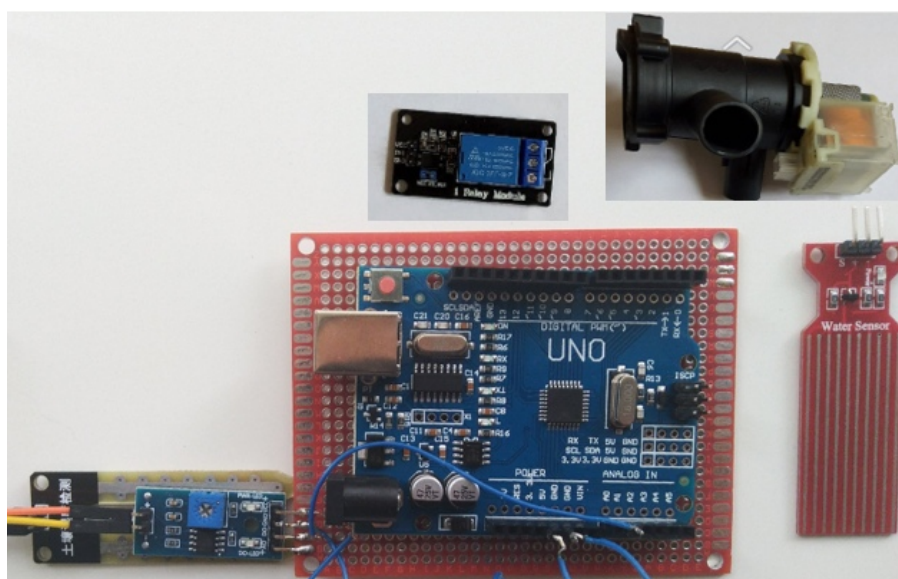


Рис. 3. Компоненты второго макета

Оба макета реализованы, отлажены программы МК, устройства практически показали свою работоспособность. Первый макет отличается бесшумностью, энергоэффективностью и дешевизной, но расплачивается за это малой мощностью и нагревом транзисторного ключа при больших токах. Для второго макета характерны высокая мощность и производительностью, но он имеет большие габариты, со временем происходит обгорание контактов реле, скорость переключения низкая.

Варианты наращивания и перспективы

Рассматриваются возможности таких качественные изменений системы автополива как реализация питания помпы от отдельного источника, организация управления поливом нескольких совершенно разных растений, для чего необходимо будет добавить в систему дополнительные датчики влажности и КУ. Для этого на оба макета есть смысл установить дополнительный модуль (Motor Shield), который позволит подключать больше исполнительных устройств и датчиков. Есть цель сделать устройство более экономным, что актуально для варианта с питанием от батарейного блока (актуально для первого макета). Второй макет нуждается в добавлении «разветвителя» потока воды.

Планируется сделать алгоритм работы системы более гибким, приспособленным к более широкому кругу задач, в частности – повышению удобства использования. Речь идет об установке дисплея и управляющих кнопок, что позволит пользователю без подключения к персональному компьютеру изменять параметры полива, добавлять разного рода оповещения (смс-оповещения об уровне воды в резервуаре, уровне заряда аккумулятора и т.д.). Улучшение программы позволит достичь более качественной работы всей системы автоматического орошения растений.

Подводя итоги, можно сказать, что получилось создать два макета, удовлетворяющие всем требованиям. Первый макет, как и ожидалось, отличается компактностью и малой стоимостью. Второй, несмотря на увеличенные габариты и определенные неудобства использования (из-за проводов не получается сделать систему компактной), также стал высокопроизводительным устройством автоматического орошения растений.

Библиографический список

1. Каталог Arduino. URL: <https://roboshop.spb.ru/#> (дата обращения: 15.03.2020).
2. Энергопотребление arduino. URL: https://www.arduino.md/arduino_power_consumption/ (дата обращения: 15.03.2020).

УДК 004.94

М. В. Русанов

студент кафедры проблемно-ориентированных вычислительных комплексов

Е. П. Виноградова – старший преподаватель – научный руководитель

ПРОБЛЕМЫ ИСПОЛЬЗОВАНИЯ ЧИСЛЕННЫХ МЕТОДОВ ИНТЕГРИРОВАНИЯ В СРЕДЕ MATHCAD

MathCAD предоставляет широкие возможности по вычислению интегралов, однако, как любая программа имеющая дело со сложными математическими вычислениями, при встрече с неопределённостью или не предусмотренным форматом ввода параметров начинает выдавать ошибки. Для понимания того, где кроется ошибка, и так как мы говорим об интегральном исчислении, необходимо понимать, что есть интеграл, а также знать о всех нюансах работы операторов интегрирования в MathCAD.

Применение интегралов находят всюду: в инженерных специальностях, в физике и, конечно, в математике. Так, например, решение дифференциальных уравнений, играющих столь важную роль практически во всех областях жизни и на которых основана современная наука, немыслимо без применения понятия интеграла.

Итак, в MathCAD есть два оператора интегрирования: для вычисления неопределённого интеграла, первообразной, и для определённого [1]. Обозначения операторов представлены на рис. 1. Использовать их предельно просто – необходимо ввести, интегрируемую функцию, переменную, по которой выполняется интегрирование, и, при необходимости, границы интегрирования:

$$\int_{\square}^{\square} \square d\square \quad \text{или} \quad \int \square d\square$$

Рис. 1. Обозначение операторов интегрирования в среде компьютерной алгебры MathCAD

На рис. 2 приведены примеры использования однократных операторов интегрирования.

$$f(x) := \frac{\ln(x)}{x} + \frac{x}{(1+x^2)} \quad w(x) := \int f(x) dx \rightarrow \frac{\ln(x)^2}{2} + \frac{\ln(x^2+1)}{2} \quad \int_0^{\pi} \frac{\sin(x)}{x} dx = 1.852$$

$x1 := 0, 0.1 .. \pi$

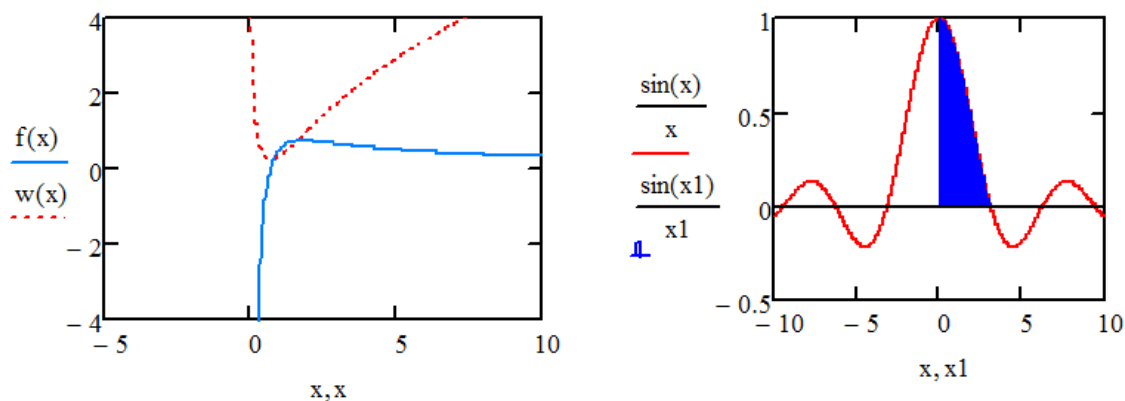


Рис. 2. Пример использования однократных операторов интегрирования

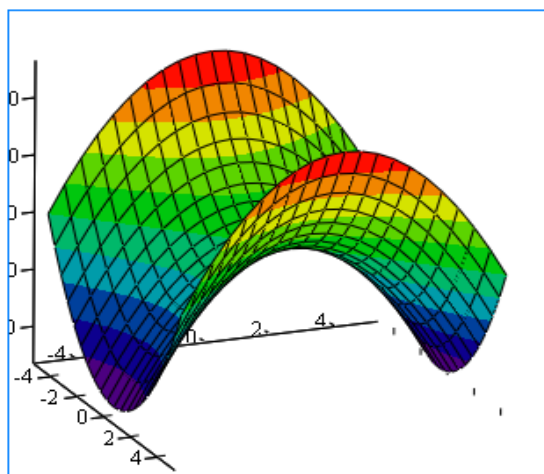
Для вычисления определённого интеграла в MathCAD имеются четыре метода [2]:

- Romberg: представляет метод Ромберга, который заключается в том, что интервал интегрирования делится пополам до тех пор, пока результат не впишется в заданную погрешность;
- Adaptive: метод, подстраивающийся под поведение функции. Наиболее универсальный способ вычисления и подходит для функций быстро изменяющихся на участке интегрирования, например, для функции волны с малым периодом колебания;
- Infinite Limit: метод, учитывающий бесконечные пределы в точках функции и используется программой автоматически всяких раз, когда один из пределов является бесконечным.
- Singular Endpoint: учитывает сингулярность (это те точки, в которых функция, например стремится к бесконечности или не определена) на промежутке интегрирования.

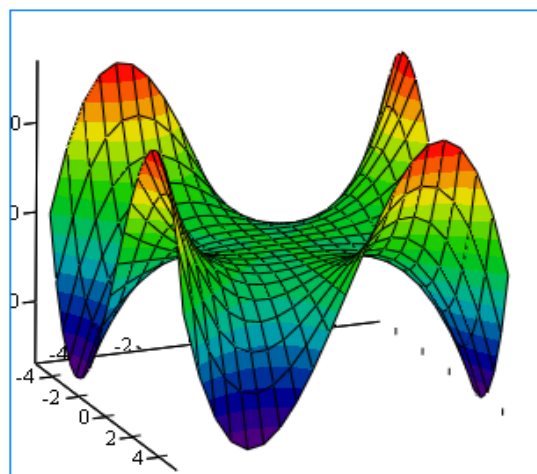
Также с помощью MathCAD можно вычислять двойные, тройные и т. д. интегралы, что следует из их определения – равенство повторному интегралу. Примеры использования повторных операторов интегрирования приведены на рис. 3:

$$\gamma(x,y) := x^2 - y^2$$

$$\text{int}(x,y) := \int_0^x \int_0^y \gamma(x,y) \, dy \, dx \quad \text{int} = f(\text{any1}, \text{any1}) \rightarrow \text{any1}^4$$



γ



int

Рис. 3. Пример использования повторных операторов интегрирования

Теперь, когда нам известно, как работать с интегралами в MathCAD, пора поговорить о проблемах, с которыми возможно столкнуться при интегрировании. Как говорилось ранее, несмотря на то, что за человека все вычисления делает машина, нужно следить за тем, что и как она считает, ведь, в отличие от человека, как мы далее увидим, она не способна справиться даже с элементарным сокращением одинаковых переменных.

В основном, большая часть проблем возникает тогда, когда появляются стандартные неопределённости вида $\frac{0}{0}$, $\frac{\infty}{\infty}$ и прочие, а также при вычислении интегралов с бесконечным нижним или верхним пределом.

Чтобы было проще понять, что и почему происходит при вычислении разными способами, можно рассмотреть несколько примеров. Первый пример представлен на рис. 4.

$$\int_{-\infty}^{-1} \frac{1}{\sqrt{x^3}} dx = 0 \quad (1) \quad \int_{-\infty}^{-3} \frac{1}{\sqrt{x^3}} dx = 0 \quad (2) \quad \int_{-3}^{-1} \frac{1}{\sqrt{x^3}} dx = -0.845i \quad (3)$$

$$\int_{-\infty}^{-1} \frac{1}{\sqrt{x^3}} dx \rightarrow -2i \quad (4) \quad \int_{-\infty}^{-3} \frac{1}{\sqrt{x^3}} dx \rightarrow -\frac{2i\sqrt{3}}{3} \quad (5) \quad \int_{-3}^{-1} \frac{1}{\sqrt{x^3}} dx \rightarrow \frac{2i\sqrt{3}}{3} + -2i \quad (6)$$

$$\int_{-3}^{-1} \frac{1}{\sqrt{x^3}} dx + \int_{-\infty}^{-3} \frac{1}{\sqrt{x^3}} dx = -0.845i \quad (7) \quad \int_{-3}^{-1} \frac{1}{\sqrt{x^3}} dx + \int_{-\infty}^{-1} \frac{1}{\sqrt{x^3}} dx \rightarrow -2i \quad (8)$$

Рис. 4. Первый пример использования различных методов для вычисления интеграла

Как можно видеть, вследствие того, что для вычисления символьным методом (в 4, 5, 6 и 8) и прямыми расчётами (в 1, 2, 3 и 7) Mathcad использует разные алгоритмы обработки данных, результатом являются разные ответы там, где присутствует прямая неопределённость. Можно полагать, что при вычислении оператором аналитического расчета «→» MathCAD сначала вычисляет интеграл в символьном виде, а затем по формуле Ньютона-Лейбница подставляет значения, а в случае использования оператора численного расчета «=» используется метод приближённого вычисления интеграла. В данном случае символьное вычисление даёт точные результаты, в отличие от прямых расчётов.

Далее рассмотрим второй пример, приведенный на рис. 5:

$$\int_1^2 \frac{1}{x} dx \rightarrow \ln(2) \quad (9) \quad \int_1^2 \frac{1}{x} dx = 0.693 \quad (10)$$

$$\int_1^3 \frac{x-1}{x(x-1)} dx = \quad (11) \quad \int_0^3 \frac{x-1}{x(x-1)} dx \rightarrow \int_0^3 \frac{x-1}{x(x-1)} dx \quad (12)$$

$$\int_1^2 \frac{x-\frac{3}{2}}{x(x-\frac{3}{2})} dx \rightarrow \int_1^2 \frac{x-\frac{3}{2}}{x(x-\frac{3}{2})} dx \quad (13)$$

$$\int_1^{1.5} \frac{1}{x} dx + \int_{1.5}^2 \frac{1}{x} dx \rightarrow 0.69314718055994530942 \quad (14)$$

$$\int_1^{\frac{3}{2}} \frac{x-\frac{3}{2}}{x(x-\frac{3}{2})} dx + \int_{\frac{3}{2}}^2 \frac{x-\frac{3}{2}}{x(x-\frac{3}{2})} dx \rightarrow \int_1^{\frac{3}{2}} \frac{x-\frac{3}{2}}{x(x-\frac{3}{2})} dx + \int_{\frac{3}{2}}^2 \frac{x-\frac{3}{2}}{x(x-\frac{3}{2})} dx \quad (15)$$

$$\int_1^{1.5} \frac{x-\frac{3}{2}}{x(x-\frac{3}{2})} dx + \int_{1.5}^2 \frac{x-\frac{3}{2}}{x(x-\frac{3}{2})} dx = \quad (16)$$

$$\int_1^{1.5} \frac{x-\frac{3}{2}}{x(x-\frac{3}{2})} dx + \int_{1.5}^2 \frac{x-\frac{3}{2}}{x(x-\frac{3}{2})} dx \rightarrow -0.039720770839917964126 \quad (17)$$

Рис. 5. Второй пример использования различных методов для вычисления интеграла

Как видно из этих примеров, для верного вычисления интеграла нет универсального способа. Также, чтобы быть уверенным в правильности полученного значения, стоит избавиться от одинаковых членов в числителе и знаменателе, иначе мы получаем лишнюю неопределённость, как в примерах (11), (13), (16) и (17). Помимо всего этого можно заметить, что Mathcad более «охотно» вычисляет интегралы, границы которых указаны не в виде дробей, а как вещественные числа.

Итак, подводя итоги, можно сказать, что для получения точного результата необходимо учесть следующие пункты:

- по мере возможностей, избегать неопределённости на промежутке интегрирования;
- избавиться от равных кратных частей функции в числителе и знаменателе;
- указывать пределы интегрирования, как числа с плавающей точкой, а не в дробном виде;
- при неудачной попытке расчёта через прямое вычисление попробовать вычислить в символьном виде;
 - для разного рода интегралов (с конечными, бесконечными пределами и прочее), использовать наиболее подходящие для выбранной функции методы вычисления (Romberg, Adaptive, Infinite Limit, Singular Endpoint);
 - уметь вычислять интегралы самостоятельно, чтобы следить за корректностью вычислений программы.

Библиографический список

1. Дьяконов В. П. MathCAD 8-12 для студентов. М.: СОЛОН-Пресс, 2005. 632 с.: ил. (Библиотека студента).
2. Интегралы: рук. пользователя MathCAD. URL: http://old.exponenta.ru/soft/MathCAD/usersguide/chapter11/11_4.asp (дата обращения: 10.04.2020).

УДК 311.218

Я. А. Савчук

студент кафедры проблемно-ориентированных вычислительных комплексов

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

СБОР И ПРЕДОБРАБОТКА ДАННЫХ ДЛЯ ПОСТРОЕНИЯ МОДЕЛИ ЦЕНООБРАЗОВАНИЯ НА АВТООБИЛЬНОМ РЫНКЕ

Сейчас автомобили являются незаменимым средством передвижения. На них работают, ездят отдыхать, используют для удобства передвижения по городу.

С темой продажи и покупки автомобиля по статистике сталкивался каждый четвертый человек в стране или каждая вторая семья. Но не каждый автовладелец может адекватно оценить свою машину, проанализировать рынок и поставить соответствующую цену при покупке или продаже авто.

С другой стороны, автопроизводителям приходится так же анализировать рынок, учитывать покупательскую способность граждан, и техническое состояние машин. Особенно это затрагивает компании, занимающиеся продажей б/у автомобилей.

В ходе данной работы были решены следующие задачи:

1. Сбор актуальных данных при помощи парсинга с доски объявлений
2. Обработка данных
3. Визуализировать данные для удобства оценки
4. Выявить зависимости цены от параметров автомобиля

Анализ авторынка

Для формирования датасета (датасет – обработанная и структурированная информация в табличном виде) из реальных данных необходимо выбрать марки самых популярных на вторичном рынке автомобилей. Для этого можно воспользоваться одной из популярных тематических интернет площадок размещения объявлений «Авто.ру» [1].

В результате были просмотрены все марки и выявлены 14 наименований, у которых количество объявлений было более 1000 штук. На рис. 1 изображен перечень марок автомобилей, объявления для которых представлены на сайте.

AMC 1	DW Hower 3	Hummer 19	Mercedes-Benz 2021	SsangYong 238
Acura 23	Dacia 3	Hyundai 1797	Mercury 2	Subaru 387
Adler 2	Dadi 1	Infiniti 284	Mitsubishi 1287	Suzuki 333
Alfa Romeo 26	Daewoo 310	Iran Khodro 3	Nissan 1807	Tesla 4
Alpina 6	Daihatsu 10	Isuzu 12	Oldsmobile 1	Tianye 1
Asia 2	Daimler 2	Jaguar 119	Opel 1334	Toyota 1612
Aston Martin 2	Datsun 83	Jeep 149	PUCH 1	Trabant 1
Audi 1273	Derways 1	Jinbei 1	Peugeot 653	Triumph 1
BMW 2082	Dodge 177	Kia 1742	Plymouth 7	Volkswagen 2535
BYD 15	DongFeng 8	LADA (BA3) 2577	Pontiac 14	Volvo 692
Baltijas Dzips 1	Doninvest 1	Lamborghini 4	Porsche 180	Vortex 21
Bentley 15	FAW 9	Lancia 1	Proton 1	Wanderer 1
Borgward 1	Ferrari 3	Land Rover 508	Ravon 20	Wartburg 2
Brilliance 1	Fiat 101	Lexus 299	Renault 1271	Willys 1
Buick 4	Ford 1969	Lifan 152	Rolls-Royce 6	Zotye 5
Cadillac 101	GMC 10	Lincoln 14	Rover 33	ГАЗ 213
Changan 8	Geely 87	Luxgen 1	SEAT 57	ЗАЗ 66
Changhe 1	Genesis 10	MG 2	Saab 68	ИЖ 38
Chery 189	Great Wall 123	MINI 131	Saturn 11	ЛуАЗ 10
Chevrolet 1328	Haima 2	Maruti 1	Scion 2	Москвич 37

Рис. 1. Количество объявлений по маркам

Ниже приведен список наименований марок автомобилей в порядке убывания количества объявлений с данными автомобилями:

- LADA
- Volkswagen
- BMW
- Mercedes – Benz
- Ford
- Nissan
- Hyundai
- Kia
- Toyota
- Opel
- Chevrolet
- Mitsubishi
- Audi
- Renault

Определившись с марками автомобилей, при помощи парсера необходимо собрать данные с сайта.

Парсер – программный инструмент для извлечения, сбора любой открытой информации с веб-сайта.

Парсинг данных

Парсер написан на языке Python и позволяет собрать данные об актуальных ценах на вторичном рынке города Санкт-Петербурга.

Для данной работы будут собираться следующие данные с каждого представленного на сайте объявления по отобраным маркам:

1. Модель
2. Марка
3. Мощность
4. Топливная система
5. Цена
6. Год производства
7. Пробег
8. Тип коробки передач
9. Кузов
10. Привод
11. Цвет

Это основная информация об автомобилях, которой будет достаточно для оценки их стоимости. Для написания парсера использовались следующие библиотеки, представленные в табл. 1.

Таблица 1

Используемые библиотеки Python

Библиотека	Описание
Requests	Данная библиотека является стандартным инструментом для выполнения HTTP запросов[2].
Csv	Данная библиотека позволяет записать полученные данные в файл с форматом csv.
BeautifulSoup	Позволяет трансформировать сложный HTML-документ в сложное древо различных объектов Python [3]. Это могут быть теги, навигация или комментарии.

Результат работы программы приведен на рисунках 2, 3.

```
In [189]: runfile('C:/Флэшка/вуз/1 ДИПЛОМ/parser.py', wdir='C:/Флэшка/вуз/1 ДИПЛОМ')
Renault Logan II || 389999 || 102 || Бензин || 2014 || 87240 || механика || седан || передний || серебристый
Renault Megane IV || 931000 || 110 || Дизель || 2016 || 77100 || робот || универсал 5 дв. || передний || серый
Renault Talisman || 1235 || 110 || Дизель || 2016 || 96000 || робот || универсал 5 дв. || передний || чёрный
Renault Sandero I || 227000 || 75 || Бензин || 2011 || 209000 || механика || хэтчбек 5 дв. || передний || белый
Renault Duster I || 435000 || 135 || Бензин || 2012 || 126000 || автомат || внедорожник 5 дв. || передний || чёрный
Renault Scenic III Grand || 400000 || 110 || Дизель || 2011 || 186000 || механика || компактвэн || передний || чёрный
Renault Scenic III || 370000 || 110 || Бензин || 2011 || 286477 || механика || компактвэн || передний || серый
Renault Sandero II Stepway || 405000 || 84 || Бензин || 2014 || 74991 || механика || хэтчбек 5 дв. || передний || коричневый
Renault Megane II Рестайлинг || 270000 || 105 || Дизель || 2008 || 270000 || механика || универсал 5 дв. || передний || серый
Renault Megane III || 315000 || 106 || Бензин || 2011 || 79350 || механика || хэтчбек 5 дв. || передний || синий
Renault Duster I Рестайлинг || 665000 || 109 || Дизель || 2017 || 63127 || механика || внедорожник 5 дв. || полный || белый
Renault Sandero I || 335000 || 84 || Бензин || 2012 || 121000 || механика || хэтчбек 5 дв. || передний || чёрный
Renault Logan I || 197000 || 87 || Бензин || 2006 || 150000 || механика || седан || передний || зелёный
Renault Scenic II Рестайлинг || 280000 || 105 || Дизель || 2008 || 220000 || механика || компактвэн || передний || чёрный
Renault Megane III || 349000 || 106 || Бензин || 2012 || 149764 || автомат || хэтчбек 5 дв. || передний || бежевый
Renault Kangoo II Рестайлинг || 509999 || 102 || Бензин || 2014 || 151690 || механика || компактвэн || передний || серебристый
Renault Kaptur I || 770000 || 143 || Бензин || 2017 || 128779 || механика || внедорожник 5 дв. || полный || белый
Renault Duster I || 559900 || 135 || Бензин || 2013 || 94000 || автомат || внедорожник 5 дв. || полный || коричневый
Renault Duster I Рестайлинг || 690000 || 143 || Бензин || 2016 || 80042 || автомат || внедорожник 5 дв. || полный || коричневый
Renault Sandero II Stepway || 585000 || 82 || Бензин || 2018 || 53601 || механика || хэтчбек 5 дв. || передний || синий
Renault Fluence I || 345000 || 110 || Бензин || 2010 || 136377 || автомат || седан || передний || серый
Renault Koleos I Рестайлинг || 675000 || 171 || Бензин || 2013 || 124070 || вариатор || внедорожник 5 дв. || полный || серый
Renault Koleos I Рестайлинг || 615000 || 171 || Бензин || 2012 || 155944 || вариатор || внедорожник 5 дв. || полный || чёрный
Renault Duster I || 479000 || 135 || Бензин || 2012 || 117000 || механика || внедорожник 5 дв. || полный || серый
```

Рис. 2. Промежуточный вывод данных

	A	B	C	D	E	F	G	H	I	J	K
1	Марка	Модель	Мощности	Топливо	Цена	Год произво.	Пробег	Коробка	Кузов	Привод	Цвет
2	Nissan	X-Trail II Рестайлинг	150	Дизель	890000	2012	151509	автомат	внедорожник 5 дв.	полный	серебристый
3	Nissan	Almera Classic I	107	Бензин	300000	2008	62000	автомат	седан	передний	бежевый
4	Nissan	X-Trail II Рестайлинг	141	Бензин	770000	2012	136034	механика	внедорожник 5 дв.	полный	коричневый
5	Nissan	Almera Classic I	107	Бензин	270000	2011	58000	механика	седан	передний	белый
6	Nissan	Teana II	167	Бензин	490000	2010	168000	вариатор	седан	полный	серый
7	Nissan	Qashqai I Рестайлинг	114	Бензин	480000	2011	90464	механика	внедорожник 5 дв.	передний	красный
8	Nissan	Qashqai I Рестайлинг	141	Бензин	545000	2010	143000	вариатор	внедорожник 5 дв.	полный	коричневый
9	Nissan	Tiida I Рестайлинг	110	Бензин	409000	2011	108000	автомат	седан	передний	красный
10	Nissan	Qashqai I	106	Дизель	459000	2008	112000	механика	внедорожник 5 дв.	передний	серый
11	Nissan	X-Trail III	171	Бензин	1570	2018	22342	вариатор	внедорожник 5 дв.	полный	белый

Рис. 3. Данные объявлений в файле

В ходе парсинга данных, была создана база данных из 17 031 реальных объявлений по городу Санкт-Петербург.

Визуализация. Результаты исследования

Перед началом работы с полученной базой данных необходимо почистить ее от пропущенных значений. В итоге на выходе получается 16583 записи объявлений. При помощи языка программирования Python и графической библиотеки Seaborn[4] на основе полученных данных были построены графики, приведенные на рисунках 4–14.

На рис. 4 видно, что самым популярным диапазоном является цена от 200 000 рублей до 500 000 рублей

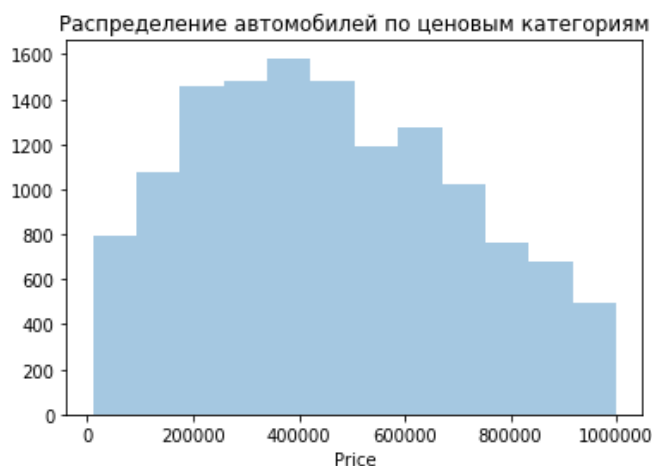


Рис. 4. Диапазон цен на автомобили

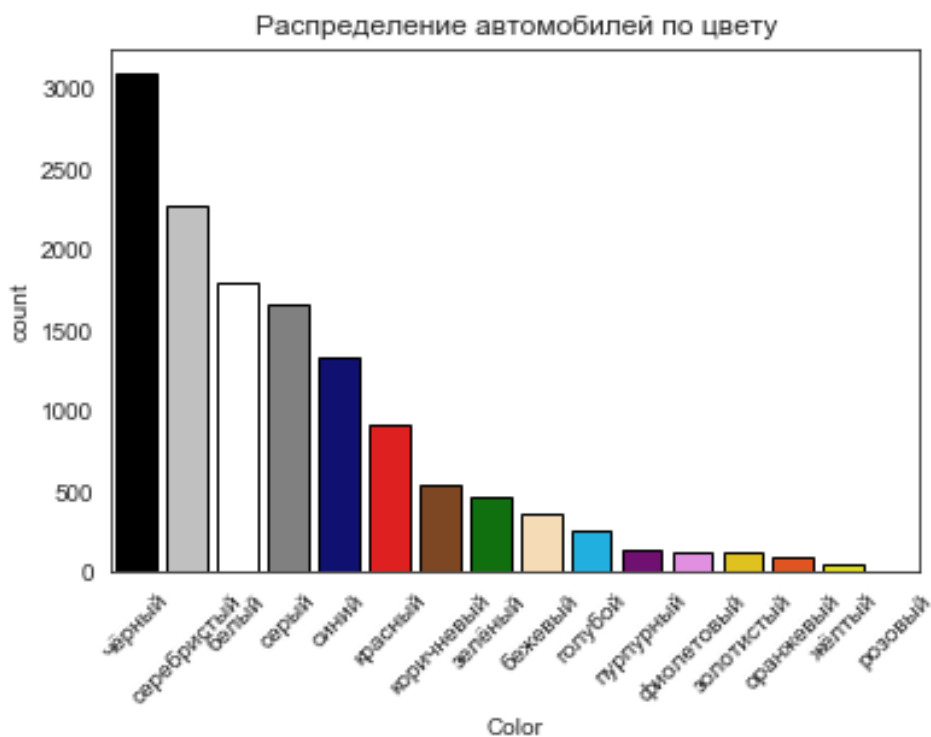


Рис. 5. График цветового диапазона на автомобили

Согласно гистограмме, представленной на рис. 5, самыми часто встречаемыми цветами являются чёрный, серый, белый и серебристый. Так же наблюдается некая зависимость цены от цвета, как видно на рис. 6. В прямоугольнике боксплота содержится 50 % значений выборки, остальные распределены на «усах».

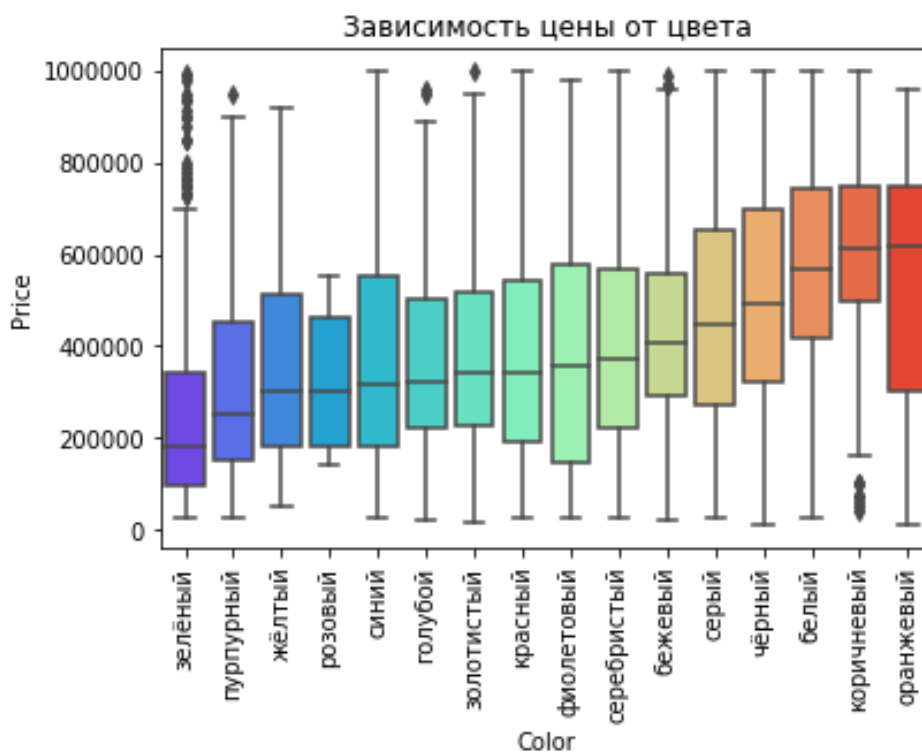


Рис. 6. График зависимости цены от цвета

График зависимости цены от марки (см. рис. 7) позволяет увидеть, что самый низкий ценовой диапазон имеет отечественная марка Лада. А самым дорогими по средней цене являются Mercedes и BMW.

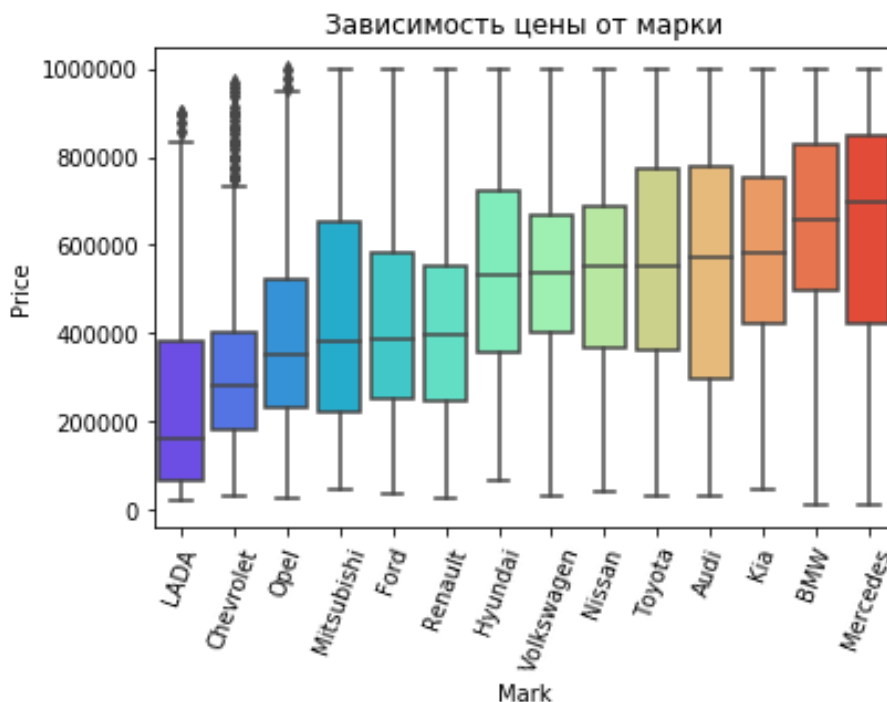


Рис. 7. График зависимости цены от марки

На рис. 8 демонстрируется зависимость вида топлива от цены.

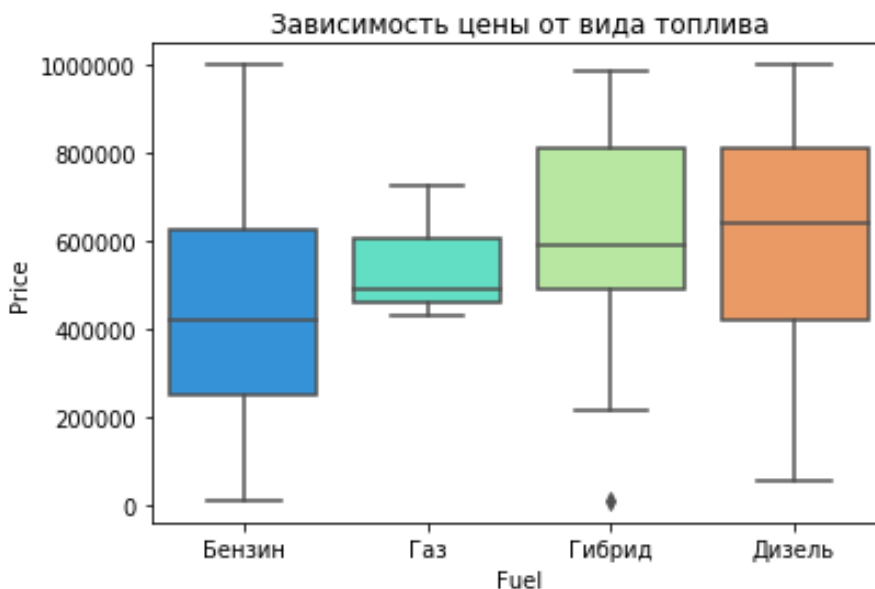


Рис. 8. График зависимости цены от вида топлива

На рис. 9 можно наблюдать регрессию, где цена почти на прямую зависит от года производства машины, чем новее автомобиль, тем выше цена на него.

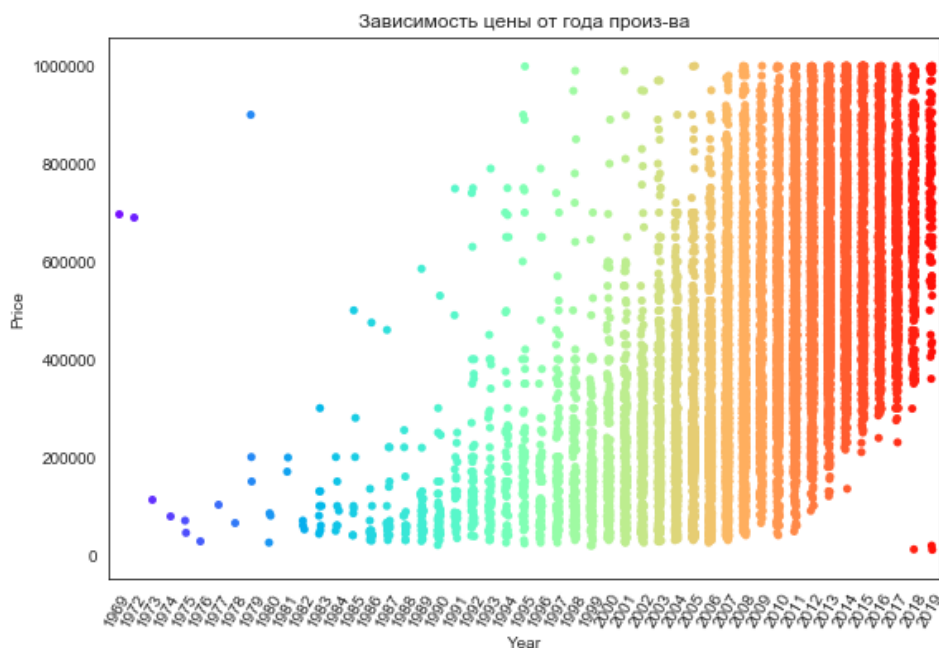


Рис. 9. График зависимости цены от года производства авто

На рис. 10 изображен график зависимости цены от кузова машины, представленный в box-plot. В данном представлении, так же как и в бокслогах, в центральной части содержится 50 % выборки, а оставшиеся 50 % по концам разбиваются на квантили, тем самым давая более подробное представление о распределении цен на «усах».

Заметно, что цены имеют большой диапазон на каждый вид кузова. Самая большая средняя цена наблюдается у четырехдверных хэтчбеков, пикапов и родстеров.

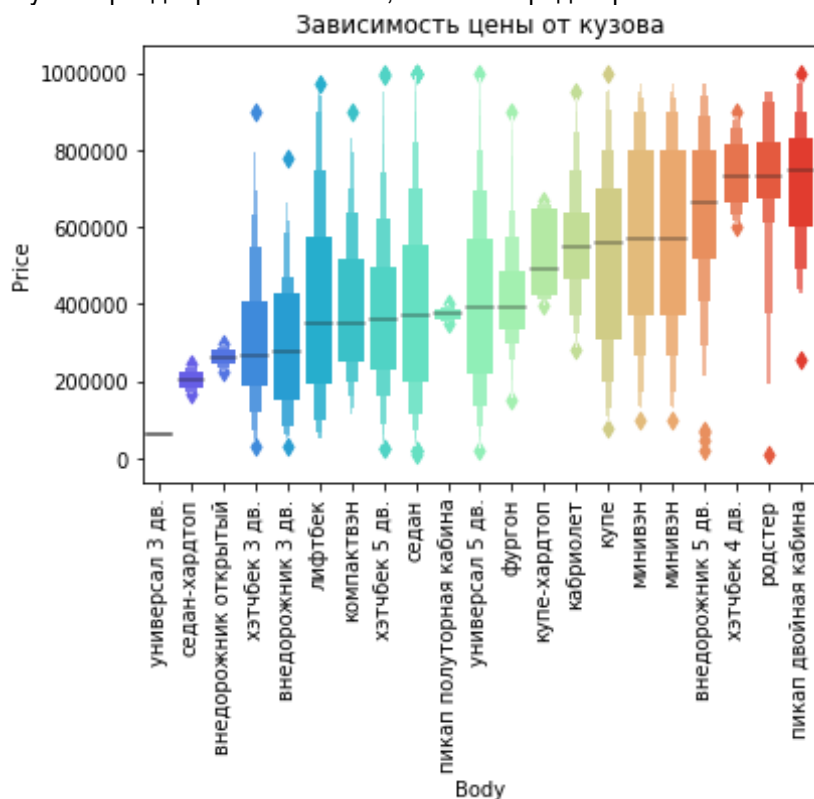


Рис. 10. График зависимости цены от кузова

На рис. 11 представлено разбиение автомобилей по коробке передач и приводу. Наибольшую цену будут иметь автомобили с автоматической и роботизированной коробкой передач при полном приводе.

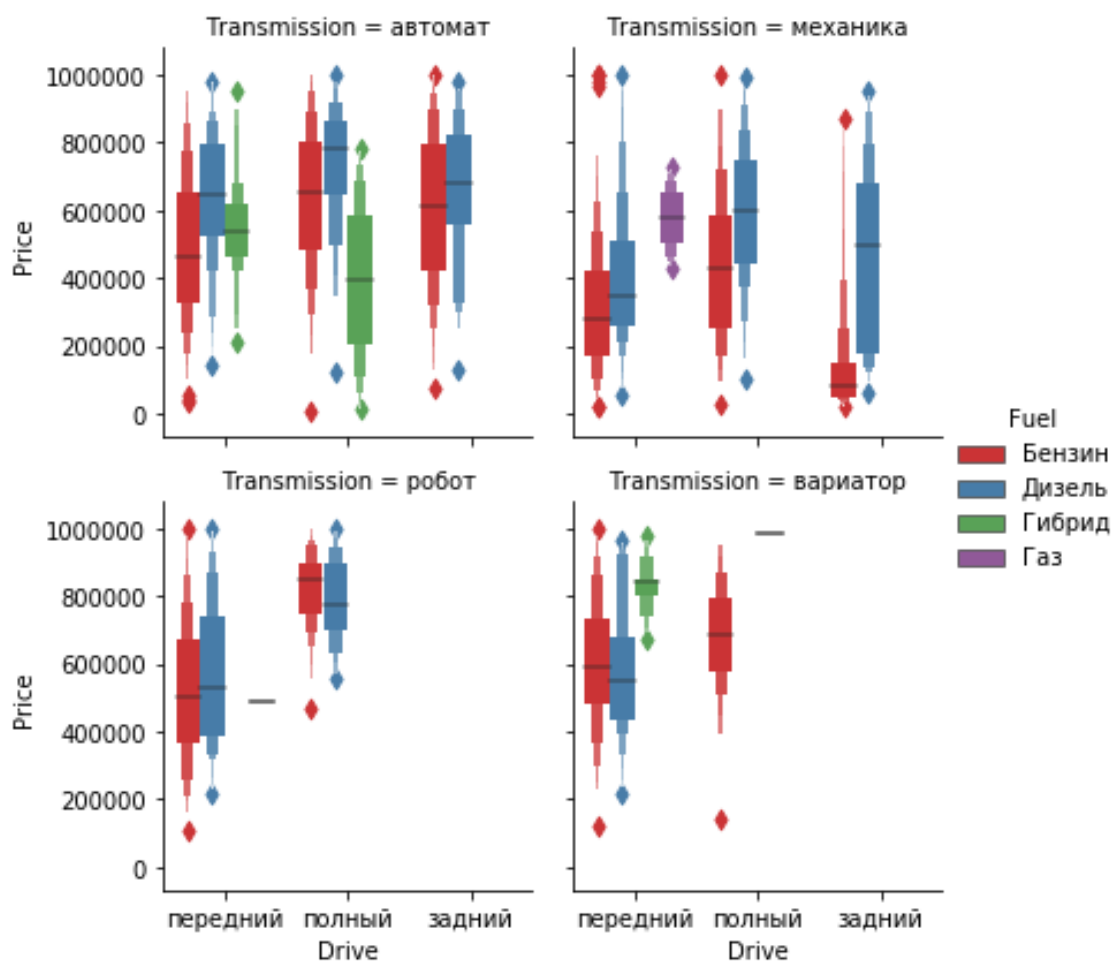


Рис. 11. График зависимости цены от привода

А на графике ниже (см. рис. 12) можно наблюдать прямую зависимость цены от мощности двигателя, растущую вместе с лошадиными силами.



Рис. 12. График зависимости цены от мощности

График зависимости цены от пробега (см. рис. 13) так же показывает большой разброс цен, но тем не менее явно прослеживается динамика снижения цены с увеличением пробега.

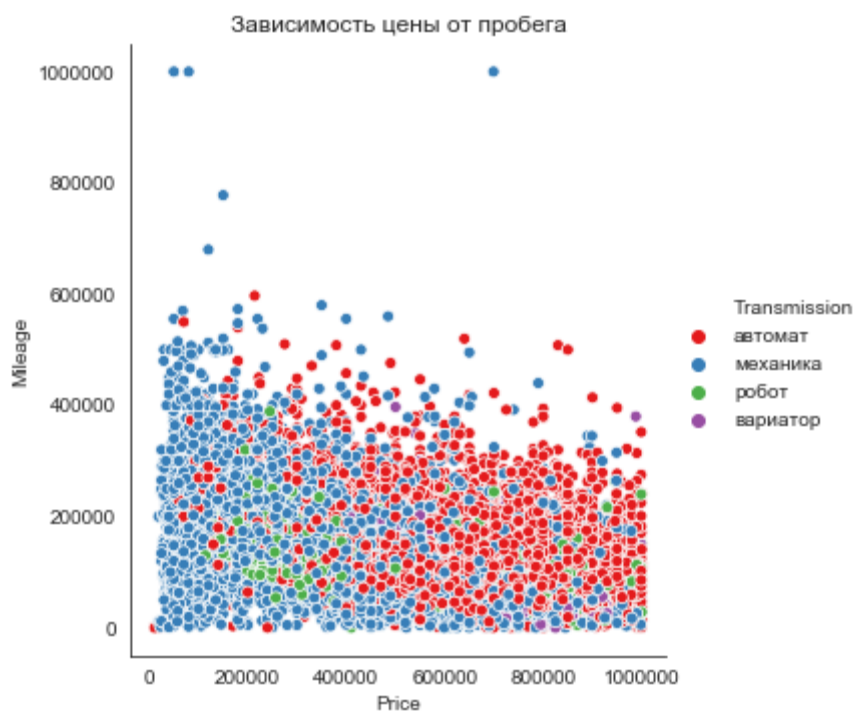


Рис. 13. График зависимости цены от пробега

Рис. 14 показывает в общем одинаковую цену на все коробки передач, кроме механической, она является самым дешевым вариантом по средней цене.

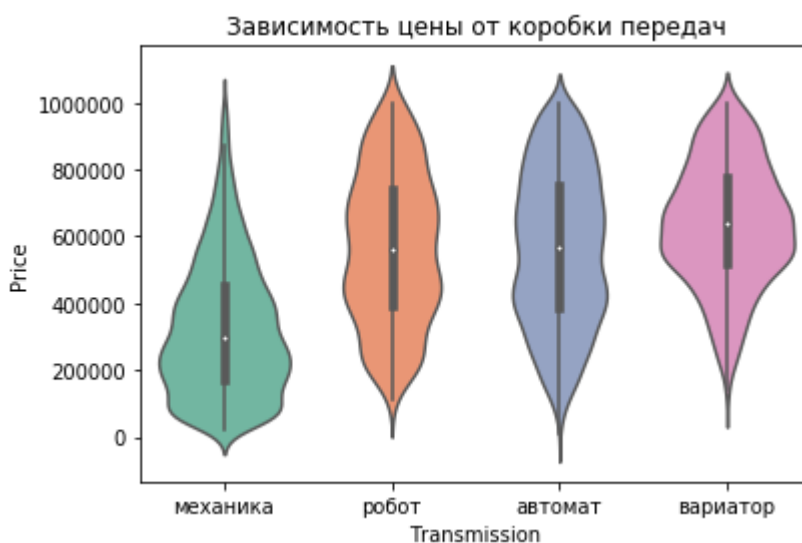


Рис. 14. График зависимости цены от коробки передач

Перспектива развития

В дальнейшем, на основе собранных и обработанных данных планируется обучить и построить модель предсказателя стоимости автомобилей. А также выявить алгоритм с наименьшей ошибкой предсказания, используя методы машинного обучения. Так как все представленные на графиках характеристики явным образом влияют на ценообразование автомобиля, планируется использование всех этих характеристик в алгоритмах машинного обучения.

Выводы

1. Произведен сбор и обработка актуальной базы объявлений.

2. Визуализированы графики параметров автомобилей от цены.
3. Выявлены необходимые параметры для оценки стоимости автомобиля для дальнейшего построения регрессионных моделей.
4. Найдена дальнейшая перспектива развития данной работы.

Библиографический список

1. «Авто.ру» – автомобильный портал. URL: <https://auto.ru> (дата обращения: 11.03.2020).
2. Python HTTP for Humans. URL: <https://pypi.org/project/requests> (дата обращения: 11.03.2020).
3. Seaborn: statistical data visualization. URL: <https://seaborn.pydata.org/index.html> (дата обращения: 11.03.2020).
4. Screen-scraping library. URL: <https://pypi.org/project/beautifulsoup4> (дата обращения: 11.03.2020).

УДК 681.5.011

Е. А. Сердечная

студент кафедры электропривода, автоматики и управления в технических системах

Е. М. Васильев – кандидат технических наук, доцент – научный руководитель

Воронежский государственный технический университет, г. Воронеж, Россия

ОБЕСПЕЧЕНИЕ РОБАСТНОСТИ СИСТЕМ С МОДАЛЬНЫМ УПРАВЛЕНИЕМ

1. Введение

В практических задачах проектирования систем управления приходится сталкиваться с нестационарностью параметров объекта. Для стабилизации характеристик систем управления в этих условиях применяют методы адаптации [1, 2]. Вместе с тем, в последние годы приобретают популярность методы робастного управления, не требующие громоздких алгоритмов адаптации [3, 4]. Настоящая работа посвящена обеспечению робастности систем на основе методов модального управления.

2. Теоретические предпосылки обеспечения робастности

Теоретическую возможность обеспечения робастности методами модального управления продемонстрируем на примере объекта $y^{(n)} + \dots + a_1 \cdot \dot{y} + a_0 \cdot y = k \cdot u$, с характеристической матрицей, представленной в форме Фробениуса:

$$B = \begin{bmatrix} 0 & \vdots & 1 & 0 \\ 0 & \vdots & 0 & 1 \\ \dots & \vdots & \dots & \dots \\ -a_0 & \vdots & -a_1 & -a_{n-1} \end{bmatrix}, \quad (1)$$

где u —управляющее воздействие, y —регулируемая величина, a_0, a_1, a_2 —параметры объекта.

После введения модального регулятора R с коэффициентами $R = [r_1 \ r_2 \ r_3]$, в систему характеристическая матрица системы примет вид:

$$B - N \cdot R = \begin{bmatrix} 0 & 1 & \vdots & 0 \\ 0 & 0 & \vdots & 1 \\ \dots & \dots & \vdots & \dots \\ -a_0 - k \cdot r_1 & -a_1 - k \cdot r_2 & \vdots & -a_{n-1} - k \cdot r_n \end{bmatrix}. \quad (2)$$

Анализ матрицы (2) показывает, что подобрав коэффициенты регулятора таким образом, чтобы выполнялось условие: $k \cdot r_{i+1} \gg a_i$, можно обеспечить робастность системы к изменению параметров a_0, a_1, a_2 .

3. Предлагаемый метод обеспечения робастности

Изложенные теоретические предпосылки указывают лишь на принципиальную возможность обеспечения робастности системы, но не указывают на способ обеспечения этой робастности, который, в данном случае, не является очевидным. Действительно, определение n коэффициентов регулятора осуществляется в результате приравнивания n коэффициентов характеристического полинома $|s \cdot E - B + N \cdot R|$ к n коэффициентам желаемого полинома. Решение этой системы уравнений единственно. Таким образом, в общем случае, обеспечить условие робастности невозможно из-за отсутствия свободных переменных.

Для преодоления этого противоречия воспользуемся тем обстоятельством, что показатели качества системы обеспечиваются небольшим количеством доминирующих корней, и нет необходимости формировать точное расположение всех n корней характеристического полинома объекта. То

есть можно задаться желаемым расположением только части корней: $S_1^*, S_2^* \dots S_m^*, (m < n)$, а расположение оставшихся $n-m$ корней будет определяться мерой их удалённости b от желаемых корней в виде неравенства: $S_j \leq b \cdot S_p^*, (p = \overline{1, m}; j = \overline{m+1, n})$. Тогда полная система условий реализации робастного модального регулятора примет вид:

$$\begin{cases} c_j = d_j, j = \overline{1, n}; \\ S_j < b \cdot S_k^*, k = \overline{1, m}; j = \overline{m+1, n}; \\ k \cdot r_j > q \cdot a_{j-1}, \end{cases} \quad (3)$$

где c_i —коэффициенты полинома замкнутой системы $|s \cdot E - B + N \cdot R|$, (s —оператор Лапласа); d_i —коэффициенты полинома (4):

$$(s - S_1^*) \cdot (s - S_2^*) \cdot \dots \cdot (s - S_m^*) \cdot (s - S_{m+1}) \cdot \dots \cdot (s - S_n), \quad (4)$$

S_j, S_k —корни характеристического полинома; q —коэффициент, определяющий степень (глубину) робастности. Система (3) содержит n уравнений и $n+(n-m)$ неизвестных. Поскольку $m < n$, то $2n-m > n$. То есть число неизвестных больше числа уравнений, и появляется возможность выполнить наложенные ограничения-неравенства

$$\begin{cases} S_j < b \cdot S_k^*, k = \overline{1, m}; j = \overline{m+1, n}; \\ k \cdot r_j > q \cdot a_{j-1}, \end{cases}$$

не единственным образом. Поэтому необходимо ввести некоторый критерий решения задачи (3), обеспечивающий единственность этого решения. Воспользуемся критерием:

$$F(r_1, r_2, \dots, r_n) = \sum_{i=1}^n \left(r_i - \sqrt[n]{r_1 \cdot r_2 \cdot \dots \cdot r_n} \right)^2 \rightarrow \min, \quad (5)$$

требующим близких значений коэффициентов регулятора.

Таким образом, построение робастного регулятора удалось свести к задаче математического программирования.

4. Практический пример

Проверим работоспособность метода на примере объекта третьего порядка со следующими матрицами: B, N и A :

$$B = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -3 & -5 \end{bmatrix}, N = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, A = [1 \ 0 \ 0]. \quad (6)$$

Характеристический полином объекта в изображениях по Лапласу: $s^3+5 \cdot s^2+3 \cdot s+1$, или в матричном виде: $|(s \cdot E) - B|$, где E —единичная матрица.

Введём в систему регулятор, тогда характеристический полином $PS(s)$ системы получит вид:

$$PS(s) = |(s \cdot E) - B + N \cdot R| = s^3 + (r_3+5) \cdot s^2 + (r_2+3) \cdot s + r_1 + 1.$$

Зададимся желаемым полиномом с одним неизвестным корнем: $PG(s) = (s+3)^2 \cdot (s-S_3)$.

Приравняем коэффициенты при s полиномов $PS(s)$ и $PG(s)$:

$$s^3 + (r_3+5) \cdot s^2 + (r_2+3) \cdot s + r_1 + 1 = s^3 + (6-S_3) \cdot s^2 + (9-6 \cdot S_3) \cdot s + 9 \cdot S_3.$$

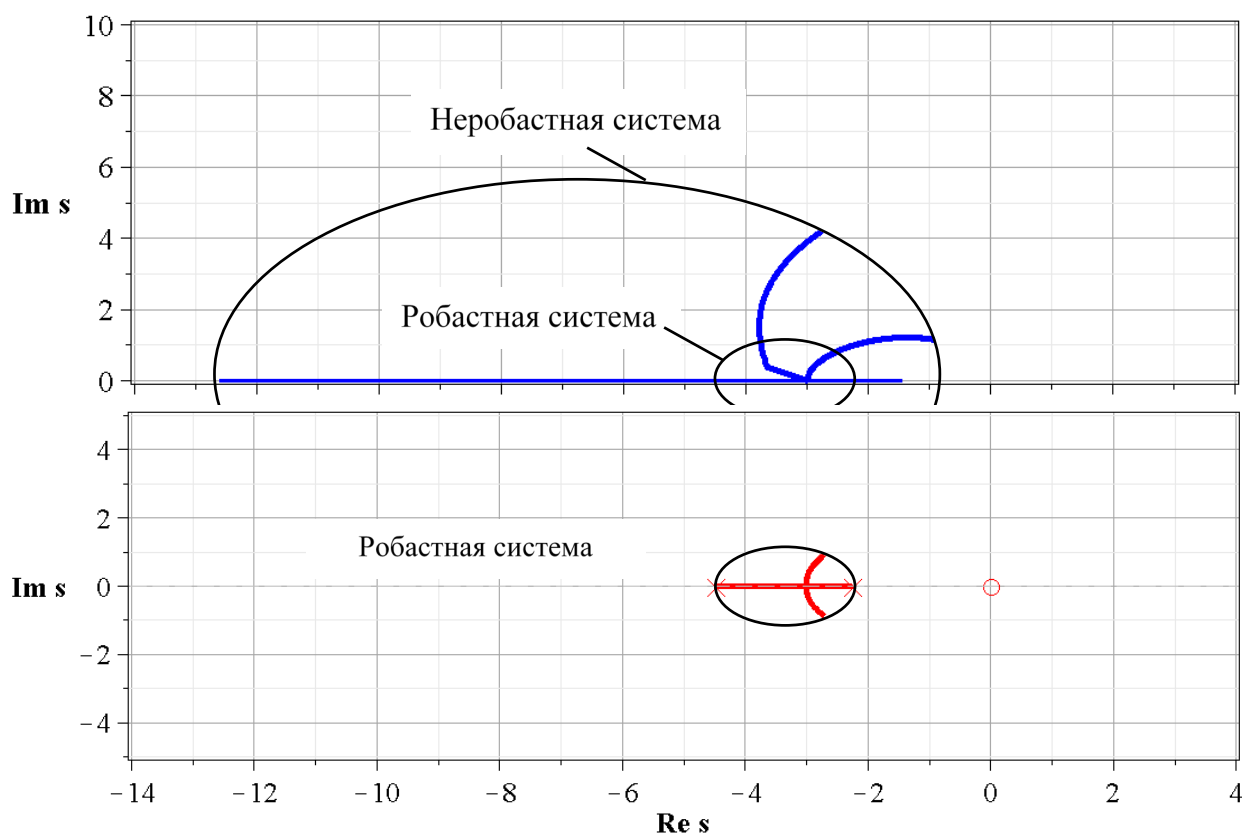
В результате система уравнений (3) с критерием (5) примет вид:

$$\begin{cases} F(r_1, r_2, r_3) = \sum_{i=1}^3 (r_i - \sqrt[3]{r_1 \cdot r_2 \cdot r_3})^2 \rightarrow \min; \\ r_1 + 1 = -9 \cdot S_3; \\ r_2 + 3 = 9 - 6 \cdot S_3; \\ r_3 + 5 = 6 - S_3 \\ k \cdot r_1 \geq 10 \cdot a_0; \\ k \cdot r_2 \geq 10 \cdot a_1; \\ k \cdot r_3 \geq 10 \cdot a_2; \\ S_3 \leq 5 \cdot s_1. \end{cases} \quad (7)$$

Получено решение: $S_3 = -49$, $r_1 = 440$, $r_2 = 300$, $r_3 = 50$.

Проведем анализ робастности полученной системы.

Построим корневой годограф двух систем: с условием робастности и без него.



58

Рис. 1. Корневые годографы: робастной и неробастной систем

Из рис.1 видно, что область локализации значимых корней робастной системы значительно меньше области локализации корневого годографа неробастной системы, то есть цель синтеза регулятора достигнута.

Дополнительно проведем сравнительный анализ областей устойчивости рассматриваемых выше систем в координатах, характеризующих изменение внутренних параметров a_1 , a_2 . На основе критерия Гурвица для третьего порядка построим области устойчивости, рис. 2.

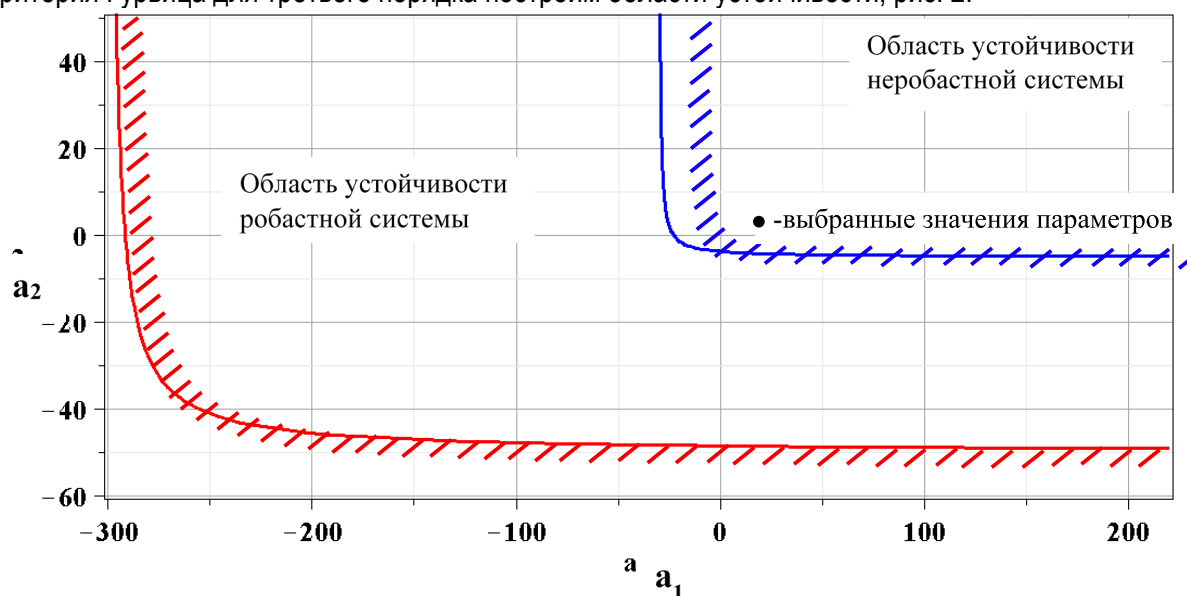


Рис. 2. Сравнение областей устойчивости робастной и неробастной систем

Рис. 2 показывает значительное расширение области устойчивости системы с робастным модальным регулятором. Такая область позволяет сохранять устойчивость системы в большем спектре значений её параметров.

5. Заключение

Существо модального управления создаёт теоретические предпосылки для обеспечения робастности систем регулирования. Однако реализация этих предпосылок затрудняется невозможностью формирования коэффициентов регулятора одновременно по условиям качества системы и по условиям её робастности. Для преодоления этого противоречия в работе предложен метод синтеза модального регулятора путём постановки задачи математического программирования с множеством допустимых решений, обеспечивающих как показатели качества, так и робастность проектируемой системы.

Библиографический список

1. Yue X. Adaptive control for attitude coordination of leader-following rigid spacecraft systems with inertia parameter uncertainties // Chinese Journal of Aeronautics. 2019. Vol. 32, № 3. P. 688–700.
2. Jordehi A. How to deal with uncertainties in electric power systems? A review // Renewable and Sustainable Energy Reviews. 2018. Vol. 9. P. 145–155.
3. Alsaadi F. Robust stability of uncertain fractional order singular systems with neutral and time-varying delays // Neurocomputing. 2020. [Preprint].
4. Wong H. Robust control of the adaptive immune system. Seminars in Immunology. 2018. Vol. 36. P. 17–27.

УДК 681.5

К. Д. Соколова

студент кафедры проблемно-ориентированных вычислительных комплексов

Д. А. Груздев

студент кафедры проблемно-ориентированных вычислительных комплексов

А. В. Сорокин – ассистент – научный руководитель

СОВРЕМЕННЫЕ ПАССИВНЫЕ АКУСТОЭЛЕКТРОННЫЕ ДАТЧИКИ ФИЗИЧЕСКИХ ВЕЛИЧИН НА ПАВ

Современная микроэлектроника базируется на интеграции дискретных элементов электронной техники, при которой каждый элемент схемы формируется отдельно в полупроводниковом кристалле. При этом в основе создания ИМС лежит принцип технологической интеграции, сопровождающейся микроминиатюризацией элементов микросхемы. Функциональная микроэлектроника позволяет реализовать определенную функцию аппаратуры без применения стандартных базовых элементов, основываясь на физических явлениях в твердом теле.

Акустоэлектроника – направление функциональной микроэлектроники, связанное с использованием механических резонансных эффектов, пьезоэлектрического эффекта, а также эффекта, основанного на взаимодействии электрических полей с волнами акустических напряжений в пьезоэлектрическом полупроводниковом материале. В акустоэлектронных устройствах используются звуковые и ультразвуковые волны широкого диапазона частот от единиц до десятков ГГц.

Выделяют особый класс акустоэлектронных устройств, функционирование которых основано на поверхностных акустических волнах. Поверхностные акустические волны являются упругими и распространяются вдоль поверхности твёрдого тела или границы раздела с другими средами.

Для высокочастотного диапазона широко используются как объёмные акустические волны (ОАВ), так и поверхностные акустические волны (ПАВ). Основным преимуществом ПАВ является доступность волнового фронта, что позволяет снимать сигнал и управлять распространением волны и характеристиками сигнала в любой точке [1, 2].

Основным элементом сенсоров на ПАВ является приёмно-передающее устройство на пьезоэлектрическом материале, который служит средой распространения поверхностных акустических волн. На рис. 1 представлена упрощённая схема приёмопередатчика на ПАВ, состоящего из излучателя, приёмника и линии задержки – пьезоэлектрической основы, по которой распространяется волны.

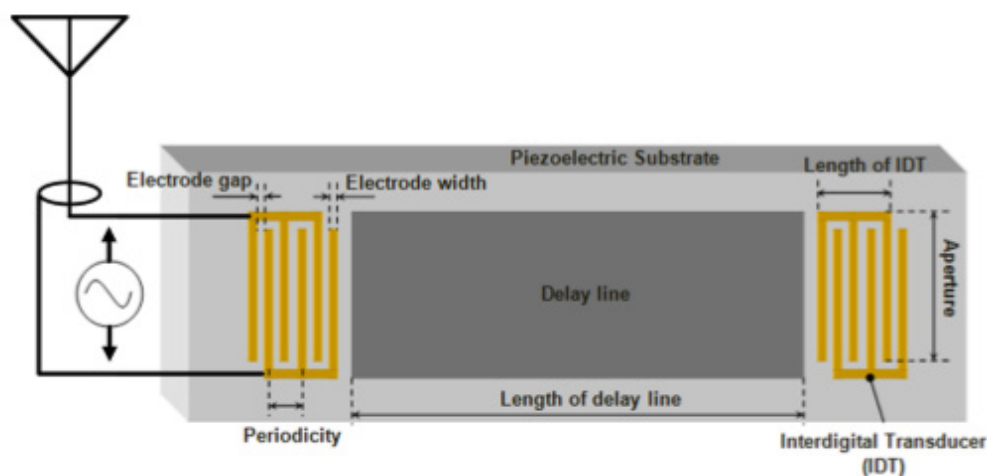


Рис. 1. Структурная схема приёмопередатчика устройства на ПАВ. 1 – антенна; 2 – излучатель; 3 – линия задержки; 4 – встречно-штыревой преобразователь

В качестве звукопровода обычно применяется пластина из пьезоэлектрического материала с тщательно отполированной поверхностью, на которой расположены электромеханические преобразователи. Эти элементы обычно выполняются в виде гребенчатых электродов из металлической пленки толщиной 0,1-0,5 мкм. Их называют встречно-штыревыми преобразователями (ВШП). К входному ВШП подключается источник электрического сигнала и таким образом в звукопроводе возникает ПАВ. А в выходном преобразователе, к которому подключается нагрузка, возникает электрический сигнал. Края на обоих концах пластинки искажаются абсорбционной резиной для подавления отражения в направлении распространения первичной волны. Если на одну из частей ВШП подается высокочастотное напряжение, то за счет обратного пьезоэффекта на поверхности пластинки генерируется ПАВ. Это волна затем распространяется вдоль поверхности пластики до тех пор, пока не попадет на другую часть ВШП, где она преобразуется обратно в высокочастотное напряжение [3].

Гибкий магнитный датчик

В качестве подложки на линии задержки в датчике внешнего магнитного поля с использованием поверхностных акустических волн применяется магнитострикционный материал на тонкой плёнке. Резонансная частота такого устройства изменяется в ответ на изменение внешнего магнитного поля. На этом явлении основан метод детектирования магнитного поля. Точность таких датчиков зависит от следующих факторов – длины и типа антенны, выходной мощности опрашивающего устройства, состояния радиоканала, ослабления сигнала внутри сенсора. Для питания ВШП, служащего опросчиком, требуется около 0.5 мВт. При использовании антенны на 15 дБ диапазон считывания сенсора достигает 18 метров. На рис. 2 представлена схема тонкоплёночного датчика магнитного импеданса на ПАВ. Он содержит в себе поверхностно волновой приёмопередатчик с тремя ВШП, чувствительную часть – тонкий сенсор магнитного импеданса и согласующую цепь.

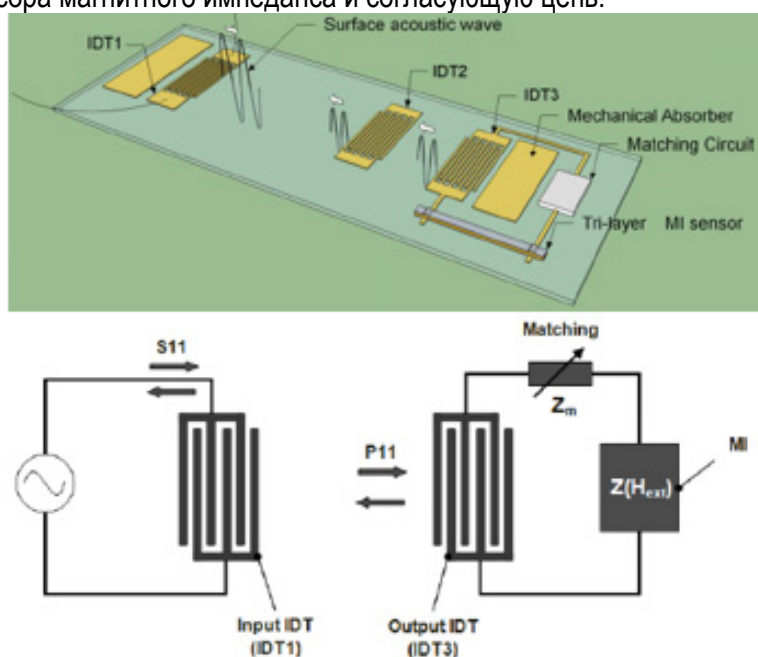


Рис. 2. Гибкий магнитный датчик на ПАВ: а) – структурная схема; б) – функциональная схема

Сенсор магнитного импеданса представляет собой структуру, имеющую один проводящий слой, расположенный между двух слоёв ферромагнетика. Произвести расчет магнитного импеданса возможно по формуле 1:

$$Z = R_{dc} (\xi x_1 + x_2) \left[\frac{\coth(x_1) \coth(x_2) + \xi}{\coth(x_1) + \xi \coth(x_2)} \right] \quad (1)$$

где R_{dc} – значение сопротивления по постоянному току ($R_{dc} = \frac{l}{t_1 w k_1 + 2 t_2 w k_2}$); $\xi = \sqrt{\frac{k_1 \mu_t}{k_2}}$; k_1 и k_2 – проводимость металлического и магнитного слоев; $TCD_i = \frac{1}{\tau_{SAW_i}} \frac{\partial \tau_{SAW_i}}{\partial T} = \frac{1}{L_i} \frac{\partial L_i}{\partial T} - \frac{1}{v_i} \frac{\partial v_i}{\partial T}$, и t_2 – толщина слоев; $x_1 = \frac{(l-1)t_1}{\delta_1}$, $x_2 = \frac{(l-1)t_2}{\delta_2}$; $\delta_1 = \sqrt{\frac{2}{k_1 \omega \mu_0}}$, $\delta_2 = \sqrt{\frac{2}{k_2 \omega \mu_0 \mu_t}}$; l – длина сенсора; ω – ширина сенсора.

Датчик температуры

Пассивный и беспроводной датчик температуры на ПАВ представлен на рис. 4. Принцип его работы заключается в следующем: одиночный пакет несущей радиочастоты испущен к ВШП и производит ПАВ, которая распространяет к рефлекторам, и после этого частично отражена назад. Отражатели расположены в соответствии с двоичным кодом фазового сдвига, и результирующие отраженные импульсы указывают код устройства и измеренную температуру [4].

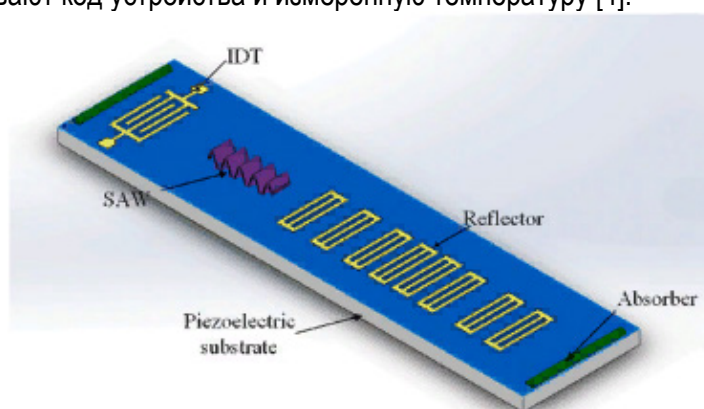


Рис. 4. Упрощенная конструкция беспроводного и пассивного датчика температуры на ПАВ

Для изоляции эхо-сигналов окружающей среды между ВШП и первым отражателем устанавливается исходное пространство для микросхемы размером 2 мм. Длительность импульса рефлектора должна составлять в два раза больше длины ВШП плюс длина рефлектора. Во избежание перекрытия соседних импульсов, минимальный зазор отражателей устанавливается равным:

$$\Delta L = \frac{T_s V}{2} = L_{IDT}, \tag{2}$$

где T_s – разность временных положений соседних отражателей; V – скорость ПАВ; L_{IDT} – скорость ВШП.

Принцип измерения значения температуры выглядит следующим образом: считыватель излучает запросный радиоимпульс и принимает ответный радиоимпульс, отраженный ПАВ структурой. После приема отклика ПАВ структуры считыватель декодирует и выделяет из него температуру ПАВ-датчика температуры и уникальный код. Измерение температуры осуществляется на физическом эффекте изменения скорости распространения акустической волны в ПАВ-кристалле в зависимости от температуры [5,6].

Система детектирования химических испарений на ПАВ

Принцип работы датчика концентрации химических веществ в воздухе основан на фиксации изменений количества частиц, попадающих в устройство. Структурная схема датчика химических испарений представлена на рис. 5.

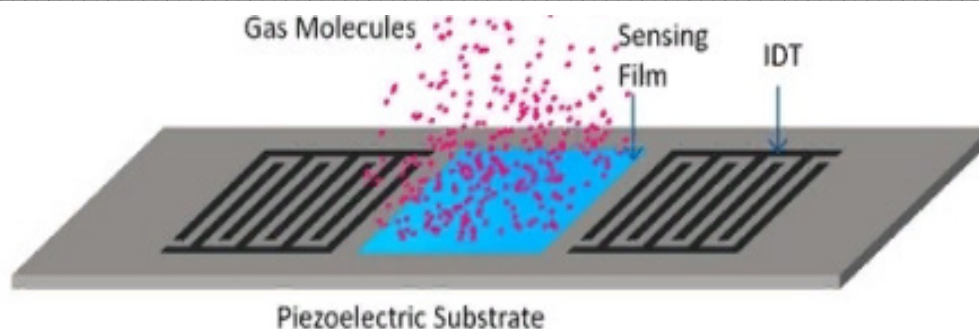


Рис. 5. Датчик концентрации химических веществ

Устройство состоит из двух ВШП, нанесенных на пьезоэлектрическую подложку и предназначенных для ввода и вывода электрического сигнала. Область между ВШП покрыта чувствительным слоем, взаимодействующим с химическими парами или газами. Молекулы химических веществ, попадая на чувствительную плёнку, способны возбудить поверхностные электрические волны в подложке, которые затем будут преобразованы в сигнал напряжения [7].

Беспроводные системы детектирования на ПАВ

В общем виде пассивные беспроводные системы датчиков состоят из трёх частей: беспроводного передающего устройства, посылающего всплески высокочастотных сигналов, датчика на ПАВ, принимающего модулированный сигнал, полученный на основе измеряемой физической величины и приёмного устройства, которое улавливает изменённый сигнал.

Передатчик и приёмник, как правило, объединены в систему, называемую опрашивающим устройством. Опрашивающее устройство связано с датчиками при помощи антенн [8].

Одновременный опрос нескольких пассивных датчиков

Современные мультисенсорные системы используют несколько датчиков для мониторинга объектов. Как правило, в системах интернета вещей IoT и промышленного интернета вещей IIOT считывающее устройство постоянно опрашивает датчики. В системах контроля электроэнергетических объектов [9–11] показано, что применяемые системы для автоматического мониторинга могут использовать несколько одновременно опрашиваемых датчиков температуры для контроля одной ячейки комплектно-распределительного устройства.

Библиографический список

1. *Дмитриев В. Ф.* Устройства интегральной электроники: Акустоэлектроника. Основы теории, расчета и проектирования: учеб. пособие. СПб.: ГУАП, 2006. С. 3–6.
2. *Li B.* Advanced Magnetoimpedance Sensors // Computer, Electrical and Mathematical Sciences and Engineering (CEMSE) Division. 2015. P. 35–38; 48–51; 98–100.
3. *Олинер А., Фарнелл Дж., Джерард Г.* Поверхностные акустические волны / под ред. А. Олинера. М.: Мир, 1981.
4. SAW temperature sensor and remote reading system / X. Q. Bao, W. Burkhard, V. V. Varadan, V. K. Varadan // IEEE Ultrason. Symp. 1987.
5. *Елисеев Н.* Перспективные ПАВ-датчики // Электроника: наука, технологии, бизнес. 2008. С. 43–44.
6. *Malocha C., Gallagher M., Fisher B. et al.* A Passive Wireless Multi-Sensor SAW Technology Device and System Perspectives. Electrical Engineering & Computer Science Department, University of Central Florida, Orlando, FL 32816, USA. 2013. P. 10–14.
7. *Devkota J., Ohodnicki Paul R., Greve David W.* SAW Sensors for Chemical Vapors and Gases. National Energy Technology Laboratory, Pittsburgh, PA 15236, USA. 2017. P. 5–9.
8. *Гуреева О.* Система радиочастотной идентификации на поверхностных акустических волнах // Компоненты и технологии. 2005. № 6.

9. Sorokin A. V., Shepeta A. P. Anti-collision radio-frequency identification system using passive SAW tags // Smart Sensors, Actuators, and MEMS VIII. 2017. Vol. 10246. P. 1024613.
10. Sorokin A., Shepeta A., Wattimena M. Wireless SAW passive tag temperature measurement in the collision case // Journal of Physics: Conference Series. IOP Publishing, 2018. Vol. 1008, № 1. P. 012015.
11. Sorokin A., Shepeta A., Wattimena M. Encoding of Passive Anticollision Radio Frequency Identification Surface Acoustic Waves Tags // Proceeding of the Electrical Engineering Computer Science and Informatics. 2017. Vol. 4. P. 605–609.

УДК 615.47

М. И. Судьин

студент кафедры проблемно-ориентированных вычислительных комплексов

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

НЕСТАНДАРТНЫЙ МЕТОД ОЦЕНКИ ЧАСТОТЫ СЕРДЕЧНЫХ СОКРАЩЕНИЙ И ЧАСТОТЫ ДЫХАНИЯ

Рассматривается нестандартный метод получения сигнала, который позволяет оценить частоту пульса и частоту дыхания человека с помощью пьезоэлектрического датчика, регистрируя микроперемещения тела.

Для того, чтобы оценить данный параметр можно использовать либо контактный метод – фотометрический, либо бесконтактный метод -баллистокардиографический стол Старра. Для изучения выбран бесконтактный метод, а именно баллистокардиография.

В интернет-статье [1] приведено определение баллистокардиографии «Это метод, который заключается в изучении сократительной способности мышцы сердца путем регистрации толчков в теле человека, возникающих в связи с сердечной деятельностью».

На функциональной схеме (рис. 1) представлены пьезоэлектрический датчик(ПЭ), с которого снимаются данные и переходят на операционный усилитель(У), при помощи резистора (R1) и построено резистора (R2) выставляется коэффициент усиления примерно равный 20.

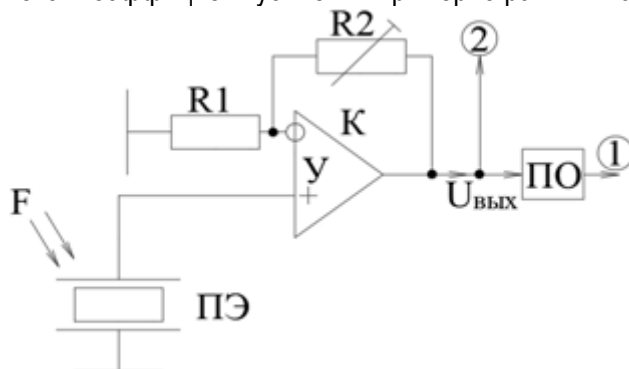


Рис. 1. Функциональная схема макета

С усилителя сигнал поступает на вход АЦП. Для преобразования сигнала в код мы использовали модуль «Е-154», представляющим собой 8-канальный АЦП с максимальной частотой преобразования входного сигнала 120 кГц с разрешением 12 бит, который обеспечивается программой «L-Graph-2», позволяющей визуализировать, сохранять и обрабатывать входные сигналы, как в реальном времени, так и в форме записанных файлов.

Как сказано в уже описанной интернет-статье [1]: «Выброс крови в аорту и легочный ствол сопровождается обратным толчком, «отдачей», передающейся телу». Для регистрации данного явления используется пьезоэлектрический датчик. Его работа указана в интернет-статье [4]: «На поверхностях пластины нанесены электроды, на которых в результате пьезоэлектрического эффекта возникают заряды противоположных знаков».

Когда человек садится на стул и происходит «отдача», то происходит микроперемещение тела, из-за которого сиденье стула деформируется, а на датчике возникает пьезоэлектрический эффект.

Сам датчик устанавливается под сиденьем стула. Из доступных был выбран пьезоэлектрический датчик ЗП-18. Пьезоэлектрические датчики между собой по некоторым характеристикам похожи, все характеристики взяты из интернет источника [2], например их интенсивность звука составляет 75 дБ, но у ЗП-18 есть свои неотъемлемые характеристики: рабочая частота ПЗ-18 равна 4050–4150 Гц, рабочее напряжение составляет 3 В, резонансная частота составляет $(4,1 \pm 0,05)$ кГц.

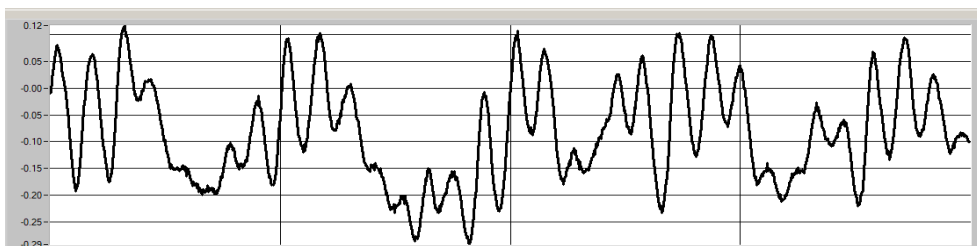


Рис. 2. Исходный сигнал

На исходном сигнале (рис. 2), сложно увидеть чёткие пики ЧСС, но при этом можно увидеть дыхание человека. Так как не со всех людей снимаются чёткие показания, в некоторых случаях возникают «артефакты движения», для фильтрации была использована программа LGraph2 с установленным плагином «digfilters». Для понимания её работы программы была изучено руководство пользователя [3], из которого можно прочесть что: «DigFilters – плагин позволяет осуществлять фильтрацию в реальном масштабе времени одновременно по нескольким каналам с последующей децимацией». Во время экспериментов было выявлено, что лучше всего использовать полосовой фильтр с нижним пределом фильтрации 15 Гц и верхним пределом фильтрации 20 Гц. После фильтрации (рис. 3), можно увидеть более чёткие пики.

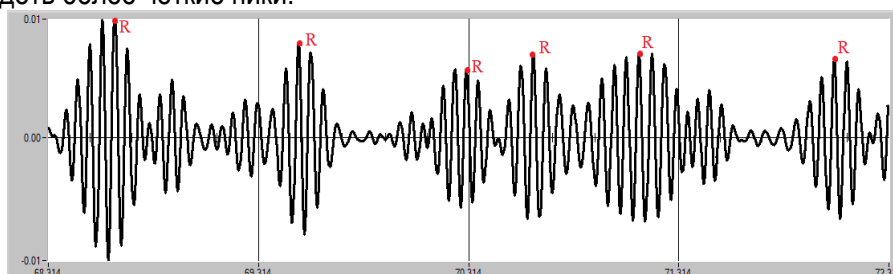


Рис. 3. Обработанный исходный сигнал испытуемого

На данном этапе мы уже можем наблюдать активность сердца и видеть частоту дыхания, более того, наблюдения показали, что оценка частоты пульса зависит от веса испытуемого. В будущем будет проводиться изучение фильтрации сигналов и их улучшение, при которых возможны изменения конструкции макета. Изучение метода в дальнейшем позволит улучшить измерения частоту сердечных сокращений и частоту дыхания человека для подробного изучения его организма.

Библиографический список

1. Баллистокардиография. URL: <http://www.medical-enc.ru/2/ballistokardiografiya.shtml> (дата обращения: 10.05.2020).
2. Пьезозвонки ЗП-19. URL: <https://www.chipdip.ru/product/zp-18> (дата обращения: 15.05.2020).
3. LGraph2. URL: https://www.lcard.ru/download/lgraph2_help.pdf (дата обращения: 15.05.2020).
4. Фрайден Жд. Современные датчики: справочник / пер. Ю. А. Заболотной. М.: Техносфера, 2005. 587 с.

УДК 681.5

А. В. Таволжанский

студент кафедры электропривода, автоматики и управления в технических системах

Е. М. Васильев – кандидат технических наук, доцент – научный руководитель

Воронежский государственный технический университет, г. Воронеж, Россия

ПОСТРОЕНИЕ МОДАЛЬНЫХ РЕГУЛЯТОРОВ С УЧЕТОМ ТРЕБОВАНИЙ ФИЗИЧЕСКОЙ РЕАЛИЗАЦИИ

1. Постановка задачи

При синтезе систем модального управления [1, 2] встречаются ситуации, при которых коэффициенты регулятора находятся в широком диапазоне вещественных чисел, занимающем несколько десятичных порядков. В связи с этим возникает проблема вычисления цифрового сигнала на выходе регулятора [3, 4]. Эта проблема обусловлена тем, что сложение вещественных чисел происходит путем преобразования их к числам одинакового порядка, а затем сложению мантисс числа. Когда выполняется процедура выравнивания порядков происходит сдвиг мантиссы числа меньшего порядка и при этом теряются младшие разряды. При значительной разнице может произойти выход мантиссы числа за разрядную сетку.

Для решения указанной проблемы разработаны различные программные методы (например, длинная арифметика [5], метод суммирования со стохастической арифметикой [6]), они усложняют код микроконтроллера и увеличивают время обработки информации.

В данной работе решается задача построения модальных регуляторов, диапазон коэффициентов которых будет снижен до практически приемлемых значений.

2. Предлагаемый метод построения модального регулятора

Идея метода построения регулятора с компактным расположением своих коэффициентов состоит в том, что от исходной системы координат x , осуществляется переход к другой системе \tilde{x} , в которой порядки коэффициентов регулятора будут локализованы в малой области вещественных чисел.

Рассмотрим исходный объект:

$$\begin{cases} \dot{x} = B \cdot x + N \cdot u; \\ y = A \cdot x, \end{cases} \quad (1)$$

где B – характеристическая матрица, N – матрица управления, A – матрица выхода, x – вектор координат состояния, u – управление, y – выходная координата

Введем регулятор R в систему с исходным объектом, рис. 1.

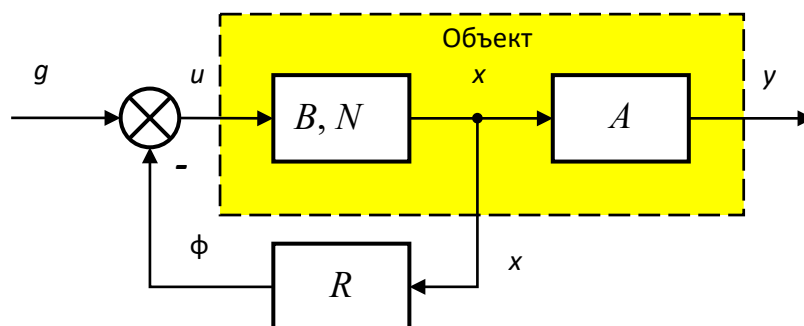


Рис. 1. Модель системы управления в исходной системе координат x

Для этой структуры получим уравнения движения (3), в которых $\phi=R \cdot x$.

$$\begin{cases} \dot{X} = B \cdot x + N \cdot u; \\ y = A \cdot x; \\ u = g - \phi. \end{cases} \quad (3)$$

Перейдем с помощью линейного преобразования T к системе \tilde{x} , где $\tilde{x} = T \cdot x$. Тогда описание объекта примет вид:

$$\begin{cases} \dot{\tilde{x}} = \tilde{B} \cdot \tilde{x} + \tilde{N} \cdot u; \\ y = \tilde{A} \cdot \tilde{x}, \end{cases} \quad (2)$$

где $\dot{\tilde{x}} = T \cdot \dot{x}$, $\tilde{B} = T \cdot B \cdot T^{-1}$, $\tilde{N} = T \cdot N$, $\tilde{A} = A \cdot T^{-1}$.

В пространстве координат \tilde{x} структура системы примет вид представленный на рис. 2.

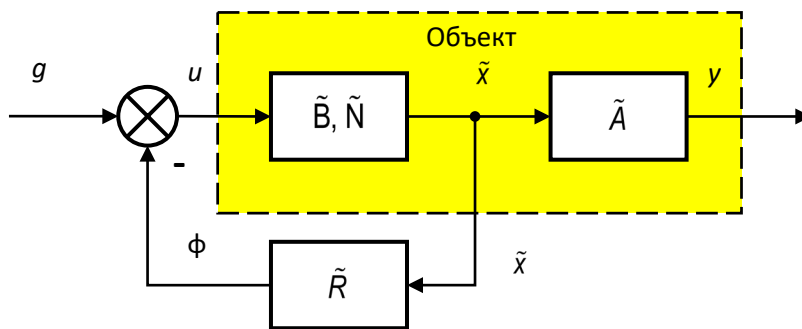


Рис. 2. Модель системы в преобразованной системе координат \tilde{x}

Уравнения движения для системы, показанной на рис.2, где $\phi = \tilde{R} \cdot \tilde{x}$:

$$\begin{cases} \dot{\tilde{X}} = \tilde{B} \cdot \tilde{x} + \tilde{N} \cdot u; \\ y = \tilde{A} \cdot \tilde{x}; \\ u = g - \phi. \end{cases} \quad (4)$$

Так как нам необходимо обеспечить неизменность выхода регуляторов, т.е. равенство ϕ в обеих системах $R \cdot x = \tilde{R} \cdot \tilde{x}$, то:

$$R \cdot x = \tilde{R} \cdot T \cdot x. \quad (5)$$

Анализ уравнения (5) показывает, что при произвольных значения коэффициентов регулятора R можно подобрать такие матрицы \tilde{R} и T , элементы которых располагаются компактно.

Представим взаимосвязь между R и \tilde{R} в виде:

$$R = \tilde{R} \cdot T, \quad (6)$$

и структура системы приобретает вид показанный на рис. 3.

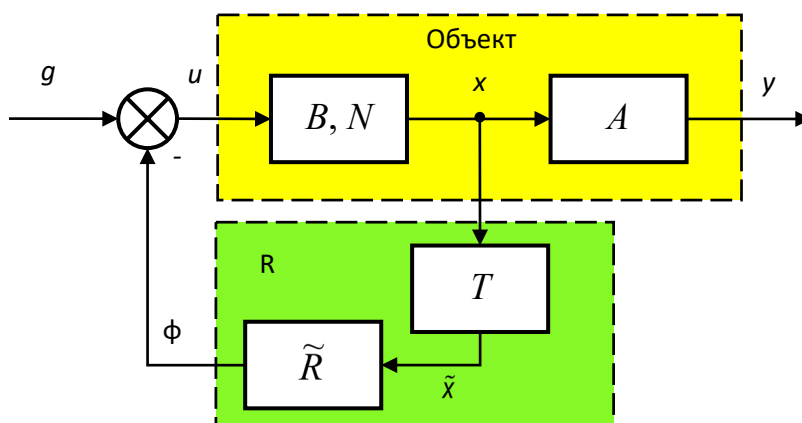


Рис. 3. Модель системы с преобразованным регулятором

Поскольку элементы \tilde{R} можно задать желаемым для нас образом, т.е. заведомо компактно, то для обеспечения равенства (7) необходимо найти n^2 элементов матрицы T , т.е. составить систему из n^2 уравнений. Так как размер матрицы R равен $1 \times n$, то из условия (7) мы получаем только n уравнений. Найдем недостающие $n^2 - n$ уравнений из равенства характеристических полиномов систем:

$$|s \cdot E - B + N \cdot R| = |s \cdot E - \tilde{B} + \tilde{N} \cdot \tilde{R}|, \quad (7)$$

или:

$$B - N \cdot R = \tilde{B} - \tilde{N} \cdot \tilde{R}. \quad (8)$$

Таким образом, синтез регулятора с компактными корнями удалось свести к задаче решения системы уравнений:

$$\begin{cases} R = \tilde{R} \cdot T; \\ B - N \cdot R = \tilde{B} - \tilde{N} \cdot \tilde{R}. \end{cases} \quad (9)$$

3. Экспериментальная проверка метода

Рассмотрим объект:

$$B = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -2,25 & -1,5 \end{bmatrix}, N = \begin{bmatrix} 0 \\ 0 \\ 0,5 \end{bmatrix}, \quad (10)$$

для которого найден регулятор R :

$$R = [15 \quad 0,00316 \quad 0,000221], \quad (11)$$

который обеспечивает монотонный переходный процесс.

Выберем \tilde{R} с компактным расположением элементов:

$$\tilde{R} = [1 \quad 2 \quad 4]. \quad (12)$$

Подставив (10), (11), (12) в (6) и (8) и решив систему (9) получим искомую матрицу T .

$$T = \begin{bmatrix} -0,349 & -0,509 & 0,112 \\ -0,952 & -0,601 & -0,676 \\ 5,751 & 0,571 & 0,414 \end{bmatrix}. \quad (13)$$

Видим, что элементы матриц T и \tilde{R} расположены в пределах двух порядков, в то время как элементы исходной матрицы R занимают пять десятичных порядков.

Цель синтеза достигнута. Оба регулятора обеспечивают монотонный переходный процесс, показанный на рис.4.

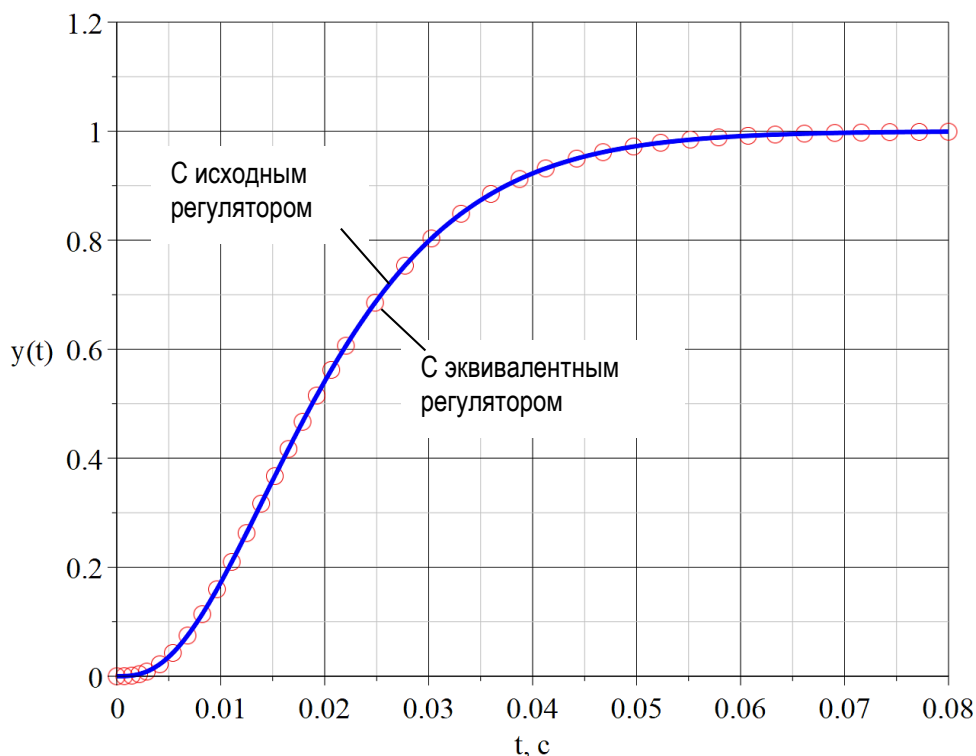


Рис. 4. Переходные процессы на выходе системы с различными регуляторами

Процессы совпадают, следовательно, регуляторы эквивалентны.

4. Заключение

Широкие возможности модального управления зачастую ограничиваются сложностью физической реализации регуляторов, в частности, разрядностью микроконтроллера и временем выполнения цикла программного кода. Известные программные методы, такие как длинная арифметика, различные варианты суммирования приводят к увеличению времени обработки информации.

Предложенный подход к синтезу модального регулятора основан на его построении в новой, искусственно созданной системе координат состояния объекта. Такой переход позволяет решить указанную проблему физической реализации без необходимости расширения разрядности контроллера и усложнения вычислительных алгоритмов.

Библиографический список

1. *Helóí F. G.* A reconfigurable damage-tolerant controller based on a modal double-loop framework // *Mechanical Systems and Signal Processing*. 2017. Vol. 88. P. 334–353.
2. *Погорелов М. Г.* Эталонная модель для синтеза модального регулятора системы автоматического управления // *Известия Тульского гос. ун-та. Техн. науки*. 2019. С. 173–197.
3. *Rony P.* Design and Implementation of Double Precision Floating Point Comparator // *Procedia Technology*. 2016. Vol. 25. P. 528–535.
4. *Тен И. Г.* Информатика: системы с плавающей запятой ограниченной разрядности // *Проблемы современной науки и образования*. 2017.
5. *Eyupoglu C.* Performance Analysis of Karatsuba Multiplication Algorithm for Different Bit Lengths // *Procedia – Social and Behavioral Sciences*. 2015. Vol. 195. P. 1860–1864.
6. *Graillat S.* Numerical Validation of Compensated Summation Algorithms with Stochastic Arithmetic // *Electronic Notes in Theoretical Computer Science*. 2015. Vol. 317. P. 55–69.

УДК 681.5.013

А. Е. Фиргер

студент кафедры электропривода, автоматики и управления в технических системах

Е. М. Васильев – кандидат технических наук, доцент – научный руководитель

Воронежский государственный технический университет, г. Воронеж, Россия

СИНТЕЗ ИНВАРИАНТНОЙ СИСТЕМЫ РЕГУЛИРОВАНИЯ С НАБЛЮДАТЕЛЕМ ВОЗМУЩЕНИЙ

Введение в проблему подавления возмущения

Проблема подавления возмущения является актуальной задачей теории управления [1]. Наиболее известное решение этой проблемы состоит в создании инвариантной системы путём ввода дополнительного канала возмущения [2]. Для создания канала возмущения требуется датчик возмущающего воздействия. Введение в систему такого датчика не всегда возможно, и целесообразно поставить задачу синтеза наблюдателя возмущения. Предлагаемая работа посвящена построению наблюдателя возмущения в рамках теории модального управления.

Условие инвариантности в системах с модальным управлением

Структура системы регулирования с дополнительным каналом ввода возмущения представлена на рис.1.

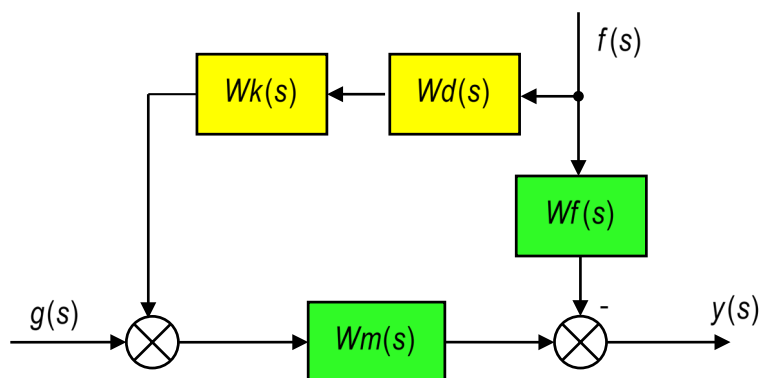


Рис. 1. Структура системы регулирования с дополнительным каналом ввода возмущения

На рис. 1 обозначены: $Wm(s) = B(s)/A(s)$ - передаточная функция замкнутой системы с модальным регулятором по задающему воздействию; $Wf(s) = G(s)/H(s)$ - передаточная функция по возмущающему воздействию; $Wk(s) = C(s)/D(s)$ - передаточная функция корректирующего звена, обеспечивающего инвариантность системы относительно f ; $Wd(s) = Q(s)/P(s)$ - передаточная функция датчика возмущения.

Уравнение движения системы:

$$y(s) = Wm(s) \cdot g(s) + (Wk(s) \cdot Wm(s) \cdot Wd(s) - Wf(s)) \cdot f(s) \quad (1)$$

Условие полной инвариантности относительно f : $Wk(s) \cdot Wm(s) \cdot Wd(s) - Wf(s) = 0$,

$$Wk(s) = \frac{Wf(s)}{Wm(s) \cdot Wd(s)} \quad (2)$$

Условие частичной инвариантности, например, для первого порядка астатизма:

$$Wk(s) = \frac{c_0}{d_0} = \frac{a_0 \cdot g_0 \cdot p_0}{b_0 \cdot h_0 \cdot q_0}.$$

Перейдём к задаче построения наблюдателя.

Построение наблюдателя возмущающего воздействия

Идея наблюдения возмущающего воздействия состоит в том, что возмущение вводится в число координат x состояния и строится наблюдатель, оценивающий все координаты состояний, в том числе и возмущение, рис.2.

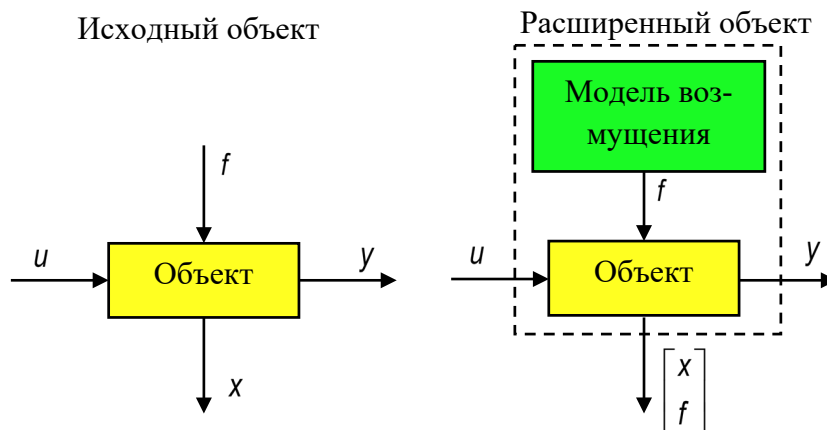


Рис. 2. Иллюстрация введения возмущения в число координат состояния

Для того, чтобы включить возмущение в число координат необходимо в модель объекта ввести модель возмущения, рис.2. Например, для $f = const$ модель возмущения можно записать как $\dot{f} = 0$ или:

$$\begin{cases} \dot{f} = S \cdot f; \\ S = 0. \end{cases} \quad (3)$$

Тогда описание расширенного объекта в пространстве состояний примет вид:

$$\begin{cases} \dot{x} = B \cdot x + F \cdot f + N \cdot u \\ \dot{f} = S \cdot f \\ y = A \cdot x \end{cases} \quad \text{или} \quad \begin{cases} \begin{bmatrix} \dot{x} \\ \dot{f} \end{bmatrix} = \begin{bmatrix} B & F \\ 0 & S \end{bmatrix} \cdot \begin{bmatrix} x \\ f \end{bmatrix} + \begin{bmatrix} N \\ 0 \end{bmatrix} \cdot u, \\ y = A \cdot x \end{cases}$$

где B – характеристическая матрица, N – матрица управления, F – матрица возмущения, A – матрица выхода системы.

Вводя обозначения: $\Lambda = \begin{bmatrix} B & F \\ 0 & S \end{bmatrix}$; $T = \begin{bmatrix} N \\ 0 \end{bmatrix}$; $M = [A \quad P]$; $\gamma = \begin{bmatrix} x \\ f \end{bmatrix}$, получим:

$$\begin{cases} \dot{\gamma} = \Lambda \cdot \gamma + T \cdot u; \\ y = M \cdot \gamma. \end{cases} \quad (4)$$

Выражение (4) представляет собой описание расширенного объекта с координатами γ , включающими неизмеряемое возмущение f . Для полученного объекта уравнение наблюдателя пониженного порядка имеет вид:

$$\dot{\hat{z}} = (\Lambda_{22} - K \cdot \Lambda_{12}) \cdot \hat{z} + (T_2 - K \cdot T_1) \cdot u + (\Lambda_{21} - K \cdot \Lambda_{11}) \cdot y + K \cdot \dot{y}, \quad (5)$$

где $K = [k_2 \quad k_3 \quad k_4]^T$ – матрица ошибок, $\Lambda = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{bmatrix}$; $T = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$; $M = [A]$;

$$\hat{z} = [\hat{x}_2 \quad \hat{x}_3 \quad \hat{x}_4 \quad \hat{f}]^T.$$

Матрица ошибок определяется путём приравнивания коэффициентов характеристического полинома $|\Lambda_{22} - K \cdot \Lambda_{12}|$ к коэффициентам желаемого типового полинома порядка $n - 1$ (предполагается, что выходная координата $y = x_1$ доступна непосредственному измерению).

Практический пример

Рассмотрим объект:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -15.873 \cdot x_1 - 3.333 \cdot x_2 + 25.717 \cdot U - 10.287 \cdot x_3$$

$$\dot{x}_3 = 0$$

$$y = x_1$$

Матричный вид:

$$B = \begin{bmatrix} 0 & 1 \\ -15.873 & -3.333 \end{bmatrix}; N = [0 \quad 25.717]^T; F = [0 \quad -10.287]^T; A = [0 \quad 1].$$

Синтез регулятора: $PJ = (s + 10)^2$ – желаемый полином; $KJ = (100 \quad 20 \quad 1)$ – коэффициенты желаемого полинома, $PF = |s \cdot E - B + N \cdot D \cdot R \cdot K|$ – формируемый полином, $R = [r_1 \quad r_2]$ – матрица регулятора, $KJ = \begin{bmatrix} 1.91 \cdot r_1 + 15.87 \\ 1.91 \cdot r_2 + 3.33 \end{bmatrix}$ – коэффициенты полинома PF .

Приравнявая коэффициенты полиномов PJ и PF , получаем элементы матрицы регулятора: $r_1 = 44.02$; $r_2 = 8.72$.

Построение наблюдателя.

Для расчёта наблюдателя необходимо найти матрицы Λ и T выделить их подматрицы $\Lambda_{11}, \Lambda_{12}, \Lambda_{21}, \Lambda_{22}, T_1, T_2$:

$$\Lambda = \begin{bmatrix} \Lambda_{11} & \vdots & \Lambda_{12} \\ \dots & \vdots & \dots \\ \Lambda_{21} & \vdots & \Lambda_{22} \end{bmatrix} = \begin{bmatrix} 0 & \vdots & 1 & 0 \\ \dots & \vdots & \dots & \dots \\ -15.873 & \vdots & -3.333 & -10.287 \\ 0 & \vdots & 0 & 0 \end{bmatrix}$$

$NPF = |s \cdot E - \Lambda_{22} \cdot K \cdot \Lambda_{12}|$ – формируемый полином -наблюдателя, $KNPF = (-10.29 \cdot k_3 \quad k_2 + 3.33)$ – коэффициенты формируемого полинома, $NPJ = (s + 300)^2$ – желаемый полином наблюдателя, $NKJ = (90000 \quad 60)$. Приравнявая коэффициенты желаемого NPJ и формируемого полинома NPF , получаем значение наблюдателя: $k_2 = 597$; $k_3 = -849$.

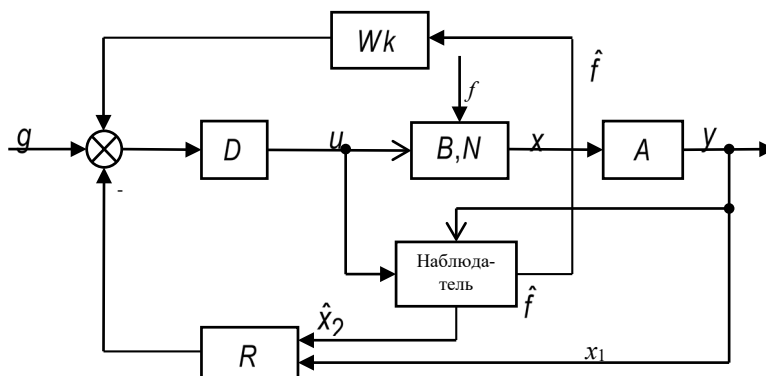


Рис. 3. Схема включения наблюдателя в систему

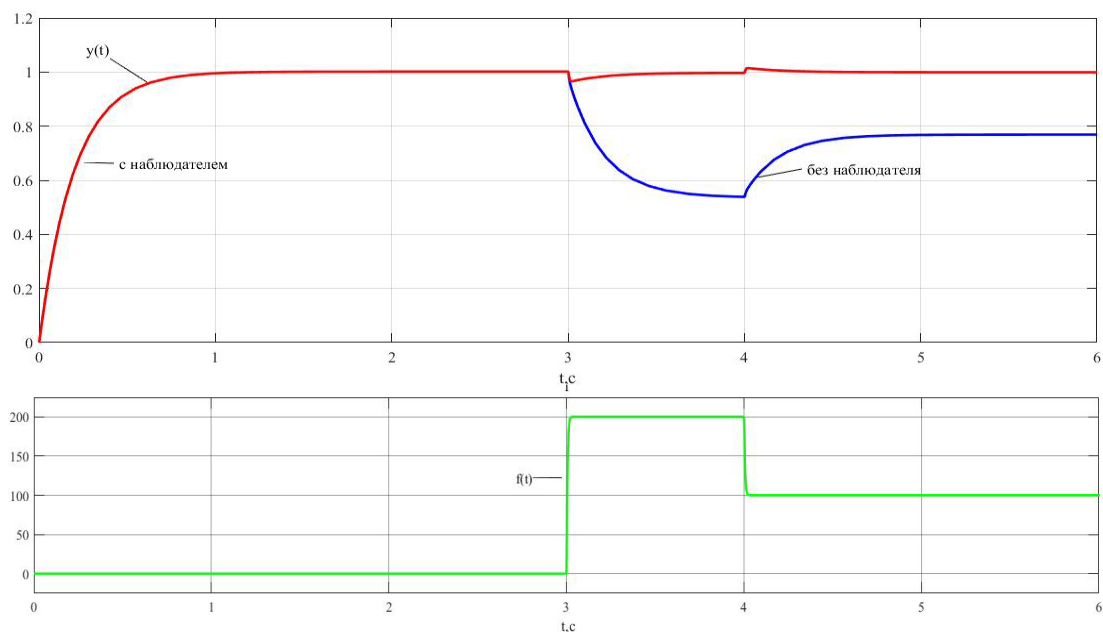


Рис. 4. Отработка систем возмущающего воздействия

Из рис. 4 видно, что построенный наблюдатель правильно восстанавливает значения возмущения $f = 200$ и $f = 100$, и система полностью подавляет возмущающее воздействие.

5. Заключение

Использование аппарата теории модального управления позволяет на строгой теоретической основе синтезировать наблюдатели возмущающего воздействия и строить системы регулирования, инвариантные относительно этого воздействия.

Библиографический список

1. Сазонов А. Е., Сахаров В. В., Чертков А. А. Модальный метод синтез наблюдателя для системы управления курсом судна // Вестн. гос. ун-та морского и речного флота. 2016. № 4 (36). С. 211–223.
2. Гайдук А. Р. Синтез селективно инвариантных систем управления // Вестн. ИГЭУ. 2017. № 1. С. 46–54.

УДК 621.31

А. А. Чеменева

студент кафедры проблемно-ориентированных вычислительных комплексов

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

ИССЛЕДОВАНИЕ СТРУКТУРЫ ТЕХНИЧЕСКИХ СРЕДСТВ ДЛЯ РЕАЛИЗАЦИИ МЕТОДИК ЛЕЧЕНИЯ РАН ОТРИЦАТЕЛЬНЫМ ДАВЛЕНИЕМ

Метод лечения ран отрицательным давлением включает в себя установку герметизирующей повязки над раневой поверхностью с последующим созданием отрицательного давления в объеме раны с помощью внешнего устройства.

Как свидетельствует опыт практической медицины [1-5] применение метода лечения ран отрицательным давлением способствует ускоренному заживлению ран. Современные возможности позволяют реализовать требуемые технические средства на основе микроконтроллера (далее МК), реализующие данные методики с небольшими габаритами, удобным функционалом и простым в использовании.

Общая структурная схема лабораторного устройства изображена на рис. 1.



Рис. 1. Структурная схема лабораторного устройства

Данное устройство, работающее в непрерывном и импульсном режимах, которое подразумевает создание и поддержание требуемого отрицательного давления. Основными элементами, обеспечивающими функционирование разрабатываемого устройства, являются электромагнитные (ЭМ) элементы, такие как компрессор и клапан [6, 7]. В ходе эксплуатации устройства оказалось, что основная проблема заключается в стабильности создания и поддержания требуемого отрицательного давления. Для проверки ЭМ элементов был разработан стенд [8], который позволяет контролировать основные характеристики элементов до установки их в устройство. Результаты испытания приведены для 12 ЭМ элементов каждого типа. Для оценки результата введен показатель K , который отображает спад давления в пневмомагистрали за одну секунду. В таблице 1 представлены результаты исследований для компрессоров типа SC3101PM [6]. Полученные данные коэффициента позволяют отобрать ЭМ комплектующие для устройства. Компрессор под номером 12 показал повышенную утечку в пневмомагистрали и не рекомендуется для использования в устройствах с питанием от автономных источников энергии.

В таблице 2 представлены результаты исследований для клапанов типа 0520D [7] от герметичности которых также зависит качество работы устройства как в непрерывном, так и в импульсном режиме. Из общего количества клапанов, проверенных на стенде, все показали свою работоспособность в импульсном режиме работы и могут использоваться для работы в устройстве.

Таким образом, входной контроль позволяет не допустить в устройство не качественные ЭМ изделия. Далее представлена электрическая принципиальная (далее ЭП) схема устройства, которая

представлена в модульном виде. На рис. 2 представлен модуль индикации и управления, включающий в себя три светодиода и кнопку.

Таблица 1

Таблица 2

Результаты испытаний для компрессоров

Результаты испытаний для клапанов

№	Коэффициент спада давления	№	Коэффициент спада давления
1	0,055	1	0,0033
2	0,0031	2	0,0030
3	0,0025	3	0,0010
4	0,0031	4	0,0019
5	0,0028	5	0,0021
6	0,0074	6	0,0022
7	0,0017	7	0,0030
8	0,0033	8	0,00243
9	0,0034	9	0,0026
10	0,0021	10	0,0043
11	0,008	11	0,0027
12	0,214	12	0,0051

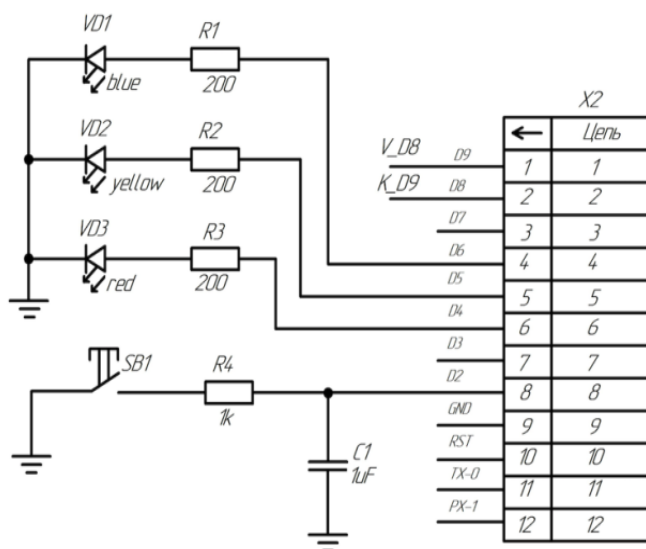


Рис. 2. Схема ЭП модуля индикации и управления

На рис. 3 представлена ЭП схема включения электродвигателя компрессора SC3101PM. Потребляемый ток двигателем составляет около 200мА, подключение к управляющему выходу D9 осуществляется через силовой ключ на полевом транзисторе IRLML6246TRPBF.

На рис. 4 представлена ЭП схема включения электромагнитный клапан 0520D, необходимого для создания атмосферного давления при импульсном режиме работы устройства. Потребляемый ток клапана составляет около 300мА, подключение к управляющему выходу D8 осуществляется через силовой ключ на полевом транзисторе IRLML6246TRPBF.

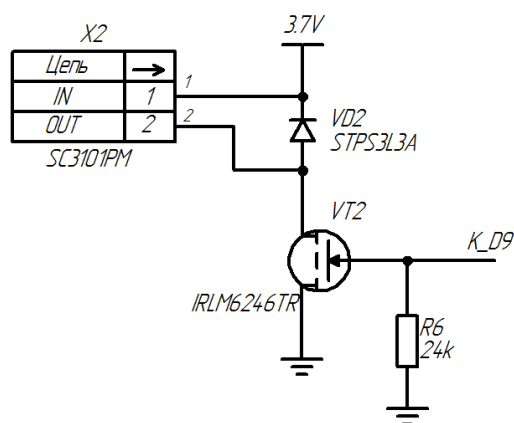


Рис. 3. Схема ЭП включения электродвигателя компрессора

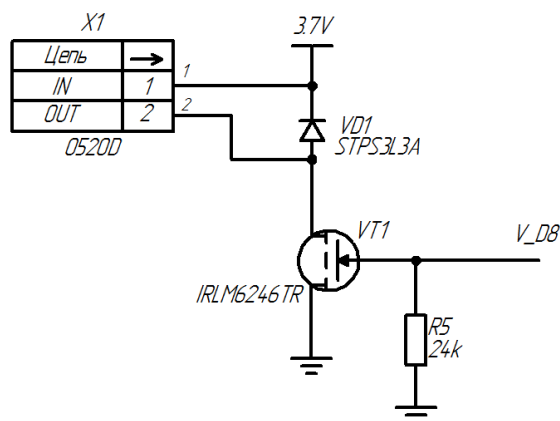


Рис. 4. Схема ЭП включения клапана

На рис. 5 представлена ЭП схема модуля измерителя давления. В качестве сенсора давления используется элемент DA1 типа PSG010R [9]. Дифференциальный выход подается на вход усилителя DA2.A через резисторы R7 и R8. Коэффициент усиления задается величиной резистора обратной связи R10. Для обеспечения балансировки дифференциального усилителя величина резистора R9 должна равняться величине резистора R10. Усилитель на элементе DA2.B необходим для получения требуемой чувствительности датчика давления, его коэффициент усиления определяется соотношением резисторов R13 и R11. Цепочка R12 и C2 служит для сглаживания сигнала от сенсора давления DA1, резистор R12 также служит для балансировки входа усилителя DA2.B. Выходная цепочка R14 и C2 является фильтром низких частот для выходного сигнала датчика давления, сигнал с конденсатора C3 поступает на вход АЦП A1 модуля МК Arduino pro mini.

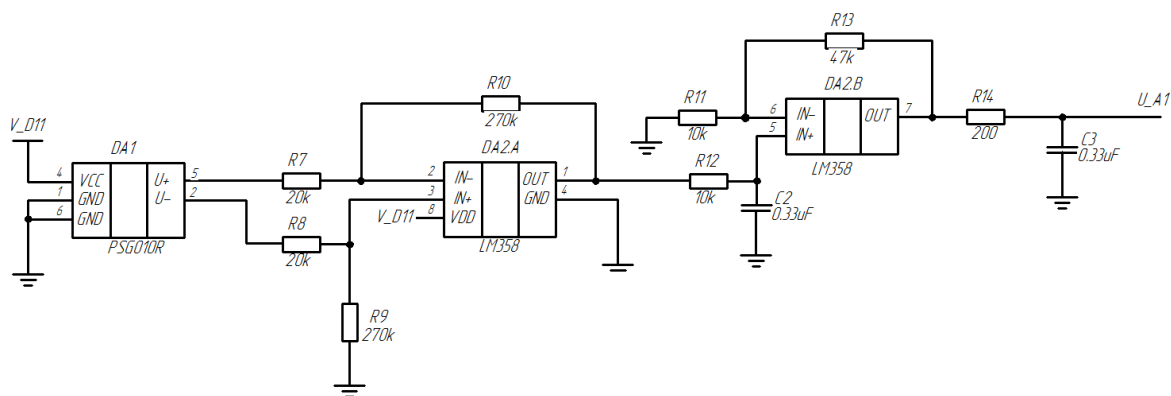


Рис. 5. Схема ЭП модуля измерителя давления

На рис. 6 представлена ЭП схема подключения модуля заряда и контроля аккумулятора к МК. Аккумулятор типа LP103450-PCM [10] B1 подключается к модулю контроля заряда X4 типа TP4056. Контроль напряжения на аккумуляторе B1 осуществляется подачей напряжения с цифрового выхода D10 на вход делителя напряжения на резисторах R14, R15.

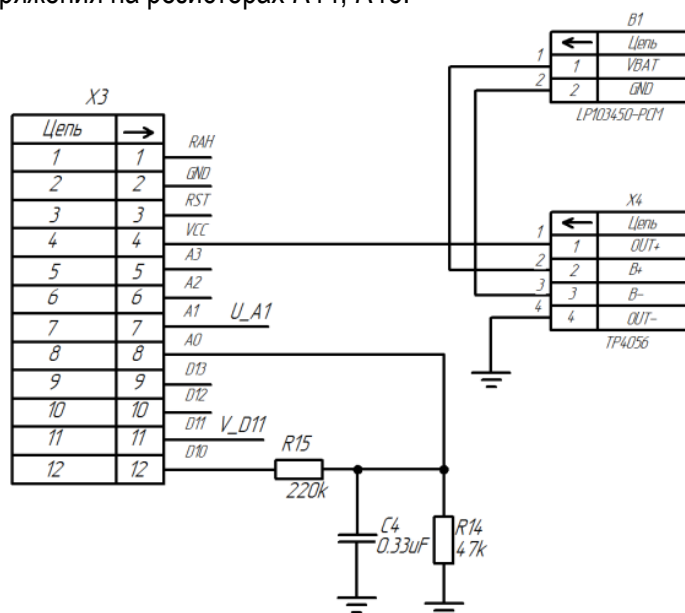


Рис. 6. ЭП схема подключения модуля заряда и контроля аккумулятора

Библиографический список

1. Часнойть А. Ч., Жилинский Е. В. Механизмы действия вакуумной терапии ран. URL: <https://cyberleninka.ru/article/n/mehanizmy-deystviya-vakuumnoy-terapii-ran/viewer> (дата обращения: 15.05.2020).
2. Peinemann F., Labeit A. Negative pressure wound therapy: a systematic review of randomized controlled trials from 2000 to 2017 // J. Evid. Based. Med. 2018. P. 125–132.
3. Терапия раны отрицательным давлением. URL: <https://www.dermnetnz.org/topics/negative-pressure-wound-therapy> (дата обращения: 15.05.2020).
4. Janssen A. H., Mommers E. H., Notter J. Negative pressure wound therapy versus standard wound care on quality of life: a systematic review // J. Wound Care. 2016. № 25. P. 156–159.
5. Бесконтейнерная индивидуальная система для лечения ран отрицательным давлением. URL: <http://zajivlenieran.ru/pico> (дата обращения: 05.05.2020).
6. Компрессор MicroPump. URL: <http://ru.skoocomtech.net/air-pump/positive-pressure-pump/smallest-air-pump-sc3101pm.html>. (дата обращения: 05.05.2020).
7. Клапан нормально закрытый. URL: <http://ru.skoocomtech.net/air-valve/micro-air-valve-for-breast-pump.html>. (дата обращения: 05.05.2020).
8. Климиник В. А., Чекменева А. А. Стенд для проверки электромеханических элементов устройства для создания отрицательного давления // Обработка, передача и защита информации в компьютерных системах. 2020. С. 33–35.
9. Датчик давления. URL: <https://www.joybuy.com/product/650053788.html> (дата обращения: 05.05.2020).

УДК 621.31

А. А. Чекменева

студент кафедры проблемно-ориентированных вычислительных комплексов

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

АЛГОРИТМ РАБОТЫ УСТРОЙСТВА ДЛЯ ЛЕЧЕНИЯ РАН ОТРИЦАТЕЛЬНЫМ ДАВЛЕНИЕМ И СРЕДСТВА ЕГО РЕАЛИЗАЦИИ

При разработке программного обеспечения (далее ПО) для устройства, реализующего методики лечения ран отрицательным давлением, была использована среда разработки «ArduinoIDE» [1]. «ArduinoIDE» представляет собой интегрированную среду разработки на языке C++ и предполагает наличие функциональных возможностей. В выбранной среде разработки был создан проект. Настройка проекта производилась исходя из модели микроконтроллера (далее МК) Atmega 328P реализованного на плате Arduino pro mini [2]. Удобная структура Arduino IDE является преимуществом, так как обеспечивает быстрое написание программы и переход к разработке приложений для Arduino. Для загрузки кода ПО перепрограммируемое запоминающее устройство (ППЗУ) МК типа Atmega 328P требуется воспользоваться рекомендуемыми производителем программирующими устройствами, в частности преобразователи USB/TTL.

Программа предназначена для управления работой устройства для лечения ран отрицательным давлением [3-5] в режимах работы (непрерывным, импульсным), в частности, создание и поддержание заданного отрицательного давления под повязкой. Под непрерывном режимом работы устройства подразумевается поддержание отрицательного давления под повязкой с учетом гистерезиса (как правило ± 10 мм.рт.ст.). Под импульсным режимом работы устройства подразумевается чередование поддержания отрицательного давления с параметрами, аналогичными параметрам работы в непрерывном режиме, и отсутствие отрицательного давления под повязкой (т.е. наличие атмосферного давления под повязкой). Решение этих задач обеспечивается путём измерения давления в пневматическом тракте устройства и поддержание его на заданном уровне в зависимости от режима работы (непрерывный или импульсный).

В состав используемых технических средств входит:

- 8-битный МК со встроенным АЦП;
- сенсор (датчик) давления [6];
- усилитель сигнала сенсора (датчика) давления;
- источник сигнала напряжения на аккумуляторе.

В качестве входных данных программа получает коды сигналов от датчика давления и напряжения на аккумуляторе. В непрерывном режиме работы устройства анализируем сигнал от датчика давления (с учетом заданного гистерезиса) и принимается решение МК о включении компрессора или поддержание его в выключенном состоянии.

В импульсном режиме работы МК ведется учет времени нахождения устройства в состоянии отрицательного давления под повязкой или при атмосферном давлении. При переходе из одного состояния пневмосистемы в другое срабатывает нормально закрытый пневматический клапан.

На основании информации о напряжении на аккумуляторе детектируется текущая стадия заряда, в зависимости от соотношения, требуемого и фактического уровней, формируется управляющая команда для индикации с помощью красного светодиода о необходимости заряда аккумулятора.

Общий алгоритм работы лабораторного образца устройства представлен на рис. 1.

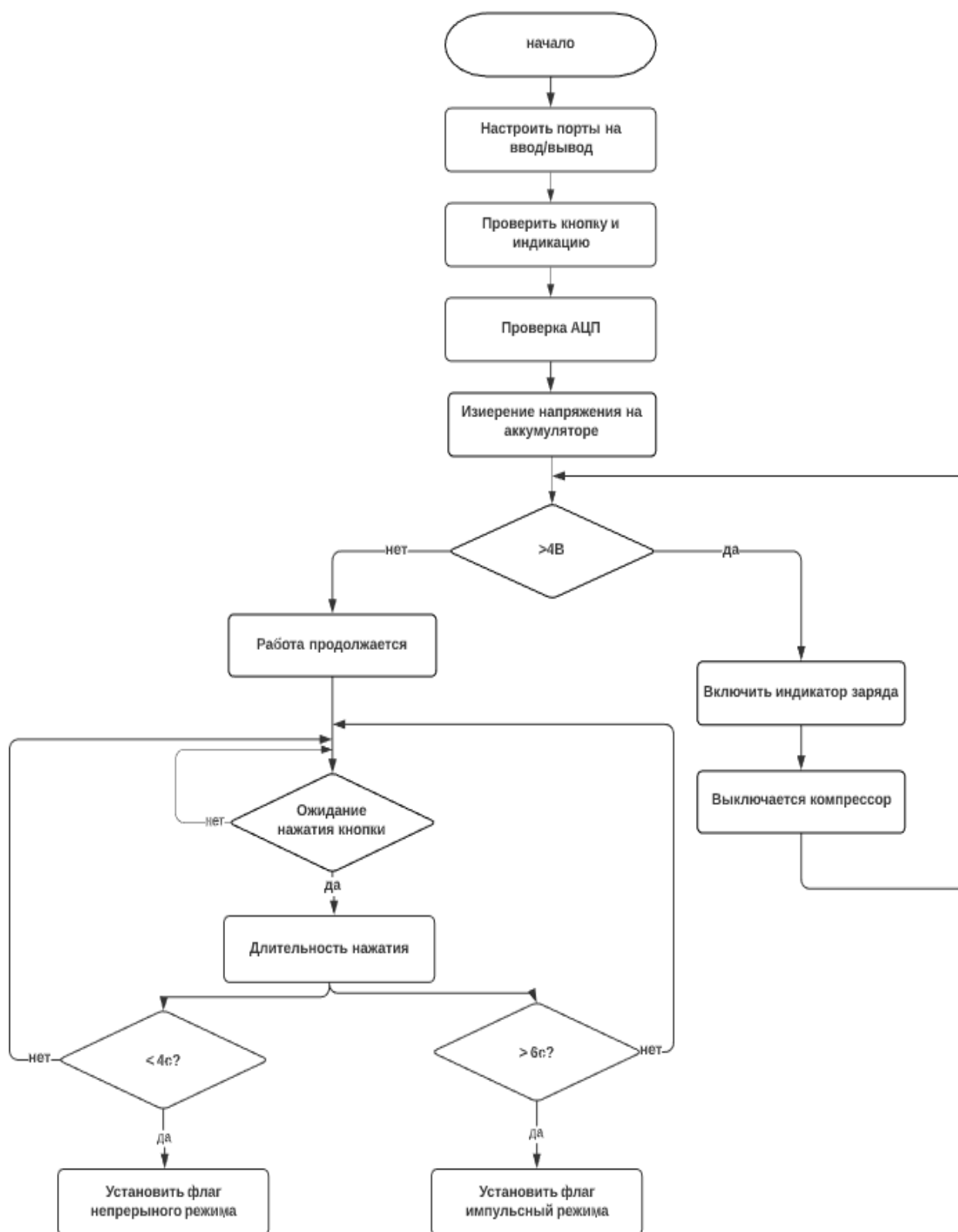


Рис. 1. Общий алгоритм работы лабораторного образца устройства

Выбор режима работы устройства осуществляется путем анализа длительности нажатия кнопки при запуске работы разработанного лабораторного образца устройства. Далее происходит установка флага режима работы – непрерывный или импульсный. На рис. 2 представлен алгоритм работы устройства в непрерывном режиме (длительность нажатия кнопки менее 4 с).



Рис. 2. Алгоритм работы устройства в непрерывном режиме

На рис. 3, 4 представлены алгоритмы работы устройства в импульсном режиме (длительность нажатия кнопки более 6 с). В импульсном режиме работы устройство работает в двух фазах. Работа в фазе 1 аналогична работе устройства в непрерывном режиме (т.е. поддержание отрицательного давления в пневмомагистрале с учетом гистерезиса). Работа в фазе 2 подразумевает сброс давления в пневмомагистрале до атмосферного давления с помощью клапана.

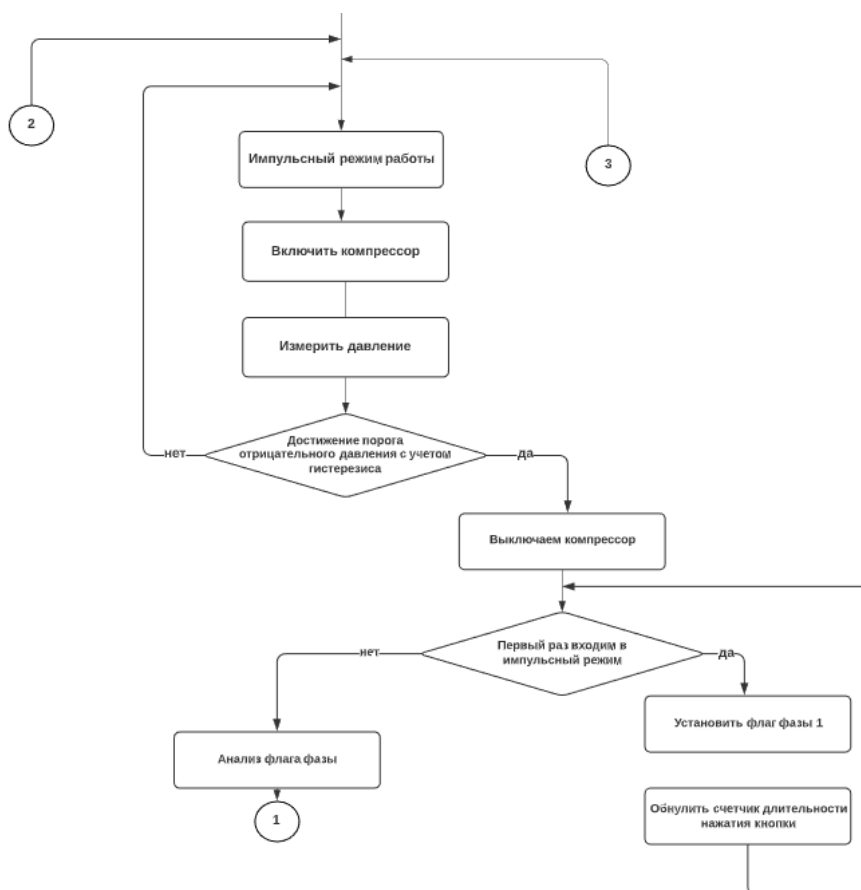


Рис. 3. Алгоритм работы устройства в импульсном режиме (начало)

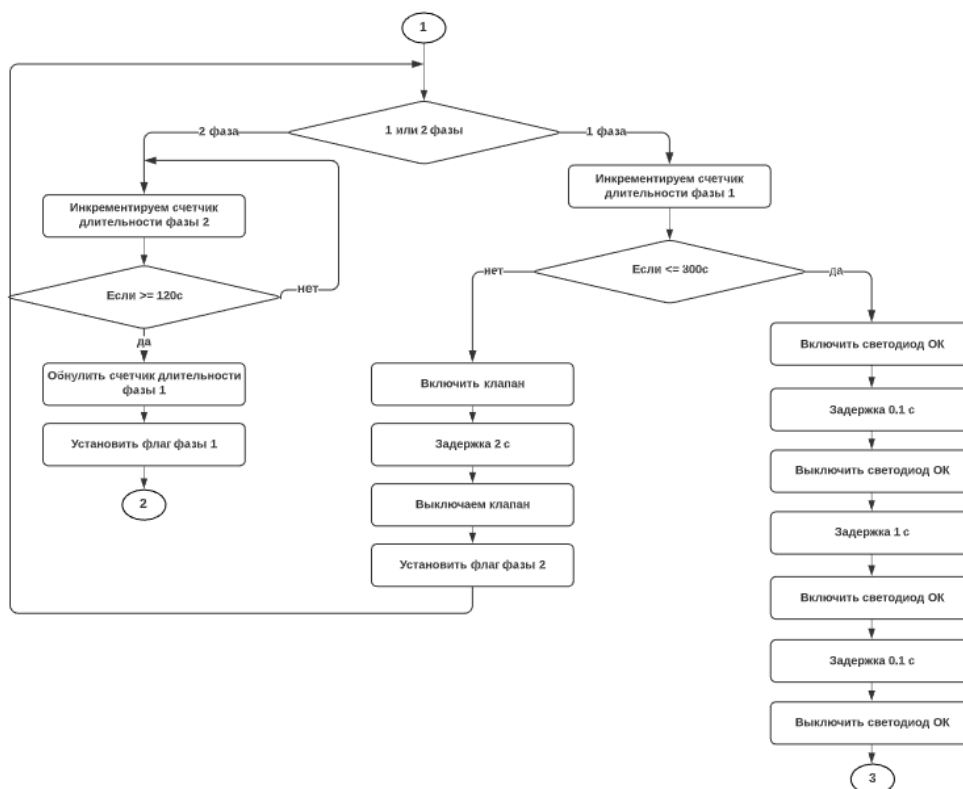


Рис. 4. Алгоритм работы устройства в импульсном режиме (окончание)

Библиографический список

1. ArduinoIDE. URL: <https://www.arduino.cc/en/Main/Software> (дата обращения: 02.05.2020).
2. Arduino pro mini. URL: https://www.delta-n.ru/Info/Arduino_Pro_Mini.pdf (дата обращения: 08.05.2020).
3. Горюнов С. В., Абрамов И. С., Чапарьян Б. А. и др. Руководство по лечению ран методом управляемого отрицательного давления. М.: Апрель, 2013. 130 с.
4. Clinical recommendations and practical guide for negative pressure wound therapy with instillation / S. Gupta, A. Gabriel, J. Lantis, L. Teot // Int. Wound. J. 2016. № 13 (2). P. 159–174.
5. Воробьев А. А., Миронова И. С., Щербаков С. А. Вакуумная терапия ран в повседневной практике отделения гнойной хирургии // Вакуумная терапия ран у детей и взрослых. 2013. С. 20–21.
6. Датчик давления. URL: <https://www.joybuy.com/product/650053788.html> (дата обращения: 05.05.2020).

УДК 004.852

А. А. Шабалин

студент кафедры проблемно-ориентированных вычислительных комплексов

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

ПРОГНОЗИРОВАНИЕ КУРСА КРИПТОВАЛЮТЫ МЕТОДАМИ МАШИННОГО ОБУЧЕНИЯ

В современном обществе, непрерывно развивающегося технически, происходит сдвиг парадигмы в способе совершения платежей денежными средствами. Драйвером этого процесса стало изобретение криптовалют. Первым представителем этого явления стал Биткойн, запущенный в 2008 году, под авторством Сатоши Накамото [1].

Важным аспектом выбора той или иной денежной единицы в качестве средства транзакции или актива является прогноз ее ожидаемой стоимости. С традиционными валютами этот вопрос хорошо изучен, в значительной степени для удержания приемлемого курса существуют специальные государственные контролирующие органы. Криптовалюта в этом вопросе более неоднозначна. Основная проблема в высокой волатильности. Можно в этом убедиться на примере Биткойна. С начала 2017 года по февраль 2018 курс Биткойна резко вырос с 1 тысячи долларов практически до 20 тысяч долларов, а после этого последовал спад до 7 тысяч долларов.

Прогнозирование курса криптовалюты является актуальной задачей, надежные способы прогноза позволяют оперативно реагировать на резкие колебания курса, что в значительной степени укрепит позиции криптовалюты, как повседневное платежное средство.

Постановка задачи

В результате исследования будут решены следующие задачи.

1. Получение и предобработка данных
2. Выбор инструментов машинного обучения
3. Обучение моделей с помощью выбранных методов

Описание исследования

Данные для исследования обучения моделей были взяты с сайта [kaggle.com](https://www.kaggle.com). Набор данных для обучения содержит поминутную стоимость Биткойна с 01.01.2012 по 08.12.2019. Курс Биткойна является временным рядом. Для обучения моделей данные были сокращены с 12.01.2016 по 08.12.2019, в этот временной промежуток наблюдалось сильное колебание курса. Исходный временной ряд был преобразован в производный ряд, усредненный по дням. На рис. 1. представлен подготовленный ряд для обучения моделей прогнозирования.



Рис. 1. Дневной курс криптовалюты Биткойн с 12.01.2016 г. по 08.12.2019 г.

Исследуемый ряд на рис. 1. имеет сильно выраженный тренд, сезонная компонента отсутствует, нерегулярная компонента присутствует. Чтобы дать ответ по циклической компоненте нужно больше наблюдений, обычно циклы делятся от 2 до 10 лет.

Адаптивные методы прогнозирования временных рядов могут успешно применяться для прогнозирования экономических показателей, денежных потоков. Методы адаптивного прогнозирования применяются там, где основной информацией для прогноза являются временные ряды. Они оказываются весьма грубыми при прогнозировании на большое число шагов вперед.

Адаптивные модели изолированных рядов при всей их простоте могут давать более надежные результаты, чем сложные эконометрические системы уравнений. Так, при существенной перестройке некоторой экономической структуры, например, под влиянием научно-технического прогресса, изменений социально-политических условий, эконометрическая модель с постоянными параметрами будет экстраполировать существенно устаревшие зависимости. Адаптивная модель в таких же условиях перманентно приспосабливается и учитывает эти изменения.

Рассматриваемые методы обладают следующими свойствами:

- они применимы для широкого круга задач;
- адаптивное прогнозирование не требует большого объема информации, оно базируется на интенсивном анализе информации, содержащейся в отдельных временных рядах;
- модель, описывающая структуру показателя и его динамику, как правило, отличается ясностью и простотой математической формулировки;
- неоднородность временных рядов и их связей находит отражение в адаптивной эволюции параметров или даже структуры моделей [2].

Адаптивные модели делятся на два вида модели экспоненциального сглаживания и авторегрессионные модели прогнозирования.

Моделью экспоненциального сглаживания, представленной в работе, является модель Хольта.

$$Y = \{y_1, \dots, y_t\}, y_i \in R, \quad (1)$$

$$\hat{Y} = \{\hat{y}_1, \dots, \hat{y}_t\}, \hat{y}_i \in R, \quad (2)$$

$$l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1}) \quad (3)$$

$$b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1}, \quad (4)$$

$$\hat{y}_{t+d} = l_t + db_t, \quad (5)$$

$$\alpha, \beta \in (0, 1), \quad (6)$$

$$d \in \{1, 2, \dots, D\}, \quad (7)$$

где Y – временной ряд, \hat{Y} – прогноз ряда, l_t – прогноз очищенный от тренда, b_t – параметр тренда, \hat{y}_{t+d} – прогноз, α, β – сглаживающие факторы, D – горизонт прогнозирования.

Важной задачей является определение коэффициентов α, β которые отвечают за чувствительность модели. Первый определяет сглаживание ряда вокруг тренда, второй за сглаживание самого тренда. Чувствительная модель способна быстро реагировать на реальные изменения, а нечувствительная не реагирует на шум и случайные отклонения [3].

Коэффициенты α, β определены на основе скользящего контроля.

Скользящий контроль – процедура эмпирического оценивания обобщающей способности алгоритмов, обучаемых по прецедентам [4]. Процесс скользящего контроля представлен на рис. 2.

$$M = \underset{\alpha, \beta \in (0, 1)}{\operatorname{argmin}} \operatorname{MSE}(CV(\operatorname{Holt}(\operatorname{train}, (\alpha, \beta)), \operatorname{test})), \quad (8)$$

где M – оптимальные значения α, β , MSE – среднеквадратическая ошибка, CV – функция последовательной подстановки сгибов, $Holt$ – модель Хольта, $train$ – временной ряд для обучения, $test$ – временной ряд для оценки прогнозов.

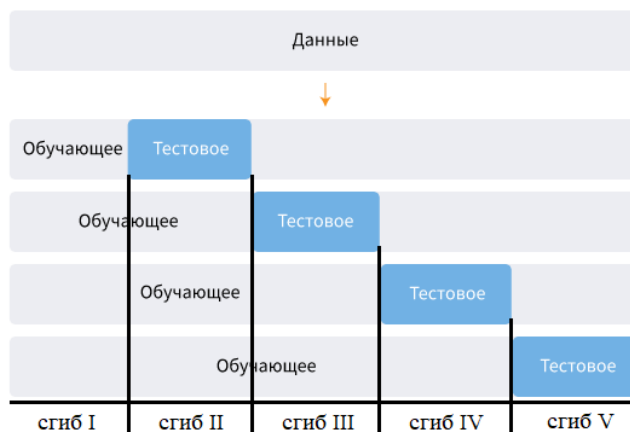


Рис. 2. Схематическое представление процесса скользящего контроля

Сгиб – множество наблюдений за временной отрезок.

Временной ряд разбивается на N равных частей. Постепенно в модель Хольта подставляются значения $\alpha, \beta \in (0,1)$. В первой итерации модель обучается на сгибе I с параметрами α, β близкие к нулю. Прогнозирует сгиб II, и ищет MSE с контрольным сгибом II. MSE и соответствующие α, β запоминается после каждого обучения модели. Далее обучение идет по I и II сгибу, III контрольный. Так процесс идет до последнего N -ого сгиба, где из всех запомненных MSE находится минимальная, дальше подставляются следующие параметры α, β и процесс повторяется. В итоге будут найдены оптимальные для прогнозирования α, β с минимальной MSE .

Авторегрессионная модель прогнозирования, представленная в работе, является модель Бокса-Дженкинса или интегрированная модель авторегрессии скользящего среднего (далее $ARIMA(p, d, q)$).

$$\Delta^d X_t = c + \sum_{i=1}^p \alpha_i \Delta^d X_{t-1} + \sum_{j=1}^q \beta_j \varepsilon_{t-j} + \varepsilon_t, \tag{9}$$

где Δ^d – оператор разности временного ряда, X_t – временной ряд, α_j – коэффициент авторегрессии, β_j – коэффициент скользящего среднего, ε_t – белый шум, p – порядок авторегрессионной составляющей, d – порядок разности дискретной производной, q – порядок скользящего среднего [5].

Для работы с моделью $ARIMA(p, d, q)$ необходимо чтобы ряд был стационарным. С помощью теста Дики-Фулера определяют стационарность ряда. Тест проверяет наличие единичных корней. Ряд, при их отсутствии, считается стационарным.

Избавиться от единичных корней позволяет процедура последовательной разностей исходного временного ряда с рядом сдвинутым на одно наблюдение вперед. Если после последовательной разности тест Дики-Фулера проваливается, то процедуру необходимо повторить. d будет равно числу применённых процедур последовательной разности.

Для поиска p, q и проверки правильности d был написан скрипт, определяющий лучший набор значений параметров в $ARIMA(p, d, q)$, результат работы скрипта – $ARIMA(0,1,1)$.

Работа выполнена на Python 3.7. Библиотеки Numpy, Scipy, Pandas, Statsmodels, Sklearn используются для подготовки данных, построения и обучение моделей, Plotly для визуализации графиков временных рядов.

Результаты исследования

На рис. 3. результат прогнозирования методом Хольта. Чем дальше прогноз заходит в будущее, тем сильнее растет ошибка прогнозирования. Минимальная ошибка наблюдалась на участках стабильного роста, стабильного падения.



Рис. 3. Прогноз, основанный на модели Хольта с параметрами $\alpha = 0.5, \beta = 0.06$

На рис.4. результат работы прогнозирования методом $ARIMA(p,d,q)$. На графике видно, что прогноз при сильном колебании курса не успевает среагировать на изменения в том объёме, чтобы использовать метод, как инструмент прогнозирования точных значений. С другой стороны, модель ошибается на 5 % если во временном ряде не наблюдается резкие скачки или падения.



Рис. 4. Прогноз, основанный на модели $ARIMA(0,1,1)$

Перспектива развития

Полученные результаты во многом свидетельствуют о перспективности выбранных моделей для прогнозирования движения направления временных рядов криптовалюты. Однако стоит отметить, что модели требуют последующего совершенствования.

Материалы статьи будут включены в выпускную квалификационную работу автора, в которой рассматриваются другие методы машинного обучения, способные решать аналогичные задачи.

Выводы

В ходе исследования были получены следующие результаты.

4. Модель Хольта и $ARIMA(p,d,q)$ не подходят для прогнозирования точных значения курса непрерывно по всему временному ряду криптовалюты Биткоин.

5. Модель Хольта и $ARIMA(p, d, q)$ показывают наилучшие результаты прогнозирования точных значений на участках временного ряда без резких колебаний, что делает возможным применять эти методы при прогнозе на фоне стабильного восходящего или нисходящего движения курса.

6. Модель Хольта и $ARIMA(p, d, q)$ являются весьма перспективными для определения направления движения курса криптовалюты Биткоин.

Библиографический список

1. Антонопулос А. М. Осваиваем биткойн. Программирование блокчейн / пер. с англ. М.: ДМК Пресс, 2018. 428 с.

2. Лукашин Ю. П. Адаптивные методы краткосрочного прогнозирования временных рядов. М.: Финансы и статистика, 2013. 416 с.

3. Machinelearning: Модель Хольта. URL: http://www.machinelearning.ru/wiki/index.php?title=Модель_Хольта (дата обращения: 03.04.2020).

4. Воронцов К. В. Комбинаторные подход к оценке качества обучаемых алгоритмов // Математические вопросы кибернетики. Вып. 13. М.: ФИЗМАТЛИТ, 2004. С. 5–36.

5. Трегуб А. В., Трегуб И. В. Методика построения модели ARIMA для прогнозирования динамики временных рядов // Лесной вестн. 2011. Вып. 5. С. 179–183.

УДК 004.043

А. А. Шемякин

студент кафедры проблемно-ориентированных вычислительных комплексов

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

ПРИМЕНЕНИЕ МЕТОДОВ ВИЗУАЛИЗАЦИИ МНОГОМЕРНЫХ ДАННЫХ В МЕДИЦИНСКОЙ СФЕРЕ

Визуализация медицинских данных – мощный инструмент для анализе изучаемых нозологий. Однако, в настоящее время отсутствуют доступные врачам программные средства, способные предоставить графическое отображение сложных данных, но при этом обладающие простым, интуитивно понятным интерфейсом. Разработке подобной программы посвящен данный проект.

В программе содержится несколько методов для анализа и визуализации медицинских данных, которые загружаются в исходном формате .xlsx или .csv сразу по нескольким переменным. Для иллюстрации реализованных методов будут использоваться данные о заболеваемости диабетом, которые содержат результаты анализов крови, индекс массы тела (BMI), толщину кожной складки трицепса (SkinThickness) и некоторые другие данные по которым определяется наличие или отсутствие данного заболевания.

Первым предложенным анализом является метод главных компонент (PCA) – это один из вариантов для уменьшения размерности данных, который, при переходе в новый базис меньшей размерности, сохраняет наибольшее количество информации. Такой тип анализа востребован в медицинских данных, так как зачастую вектора данных, соответствующие пациентам, обладают крайне высокой размерностью [1]. На рис. 1 представлен интерфейс пользователя для метода главных компонент. Перед его применением пользователь при необходимости может провести некоторую предобработку данных, а затем, выбрав только нужные факторы, получить результат.

Метод главных компонент

Файл +

Завершить исследование -

Завершить

Преобразование набора данных -

Удаление факторных и номинативных переменных

Столбцы +

Фильтрация по дате +

Разбиение содержимого ячейки +

Группировка терминов +

Обработка пустых значений +

Замена по словарю +

Уникальный идентификатор записи +

Метод главных компонент -

Выберите нужные столбцы:

Данные -

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	35	168	43.1	2.288	33	1
6	5	116	74	0	0	25.6	0.201	30	0
7	3	78	50	32	88	31	0.248	26	1
8	10	115	0	0	0	35.3	0.134	29	0
9	2	197	70	45	543	30.5	0.158	53	1
10	8	125	96	0	0	0	0.232	54	1

Showing 1 to 10 of 768 entries

Previous 1 2 3 4 5 ... 77 Next

Сводная информация по данным в таблице +

Метод главных компонент -

для получения результата примените метод. ВНИМАНИЕ, для лучшей точности необходимо удалить факторные переменные

Рис. 1. Интерфейс пользователя

Результатом применения этого метода является диаграмма рассеяния, представленная на рис. 2. На графике изображены векторы, которые соответствуют значениям из исходной таблицы, в нашем случае: “SkinThickness”, “Glucose”, “Age”. Этот график наглядно демонстрирует связь между исходными показателями и главными компонентами. Если вектор переменной сонаправлен с осью, то

между ними есть положительная связь, иначе отрицательная. Угол между векторами также является информативным: косинус угла между векторами – это коэффициент корреляции между показателями, соответствующими этим векторам. Также показатели можно сгруппировать по иной переменной, для поиска лучших связей.

Основным преимуществом данного метода является наглядное представление данных эксперимента или результатов теоретического исследования. Также используется для сокращения размерности динамических моделей, при подавлении шума на изображениях, активно используется в биоинформатике для выделения значимой информации.

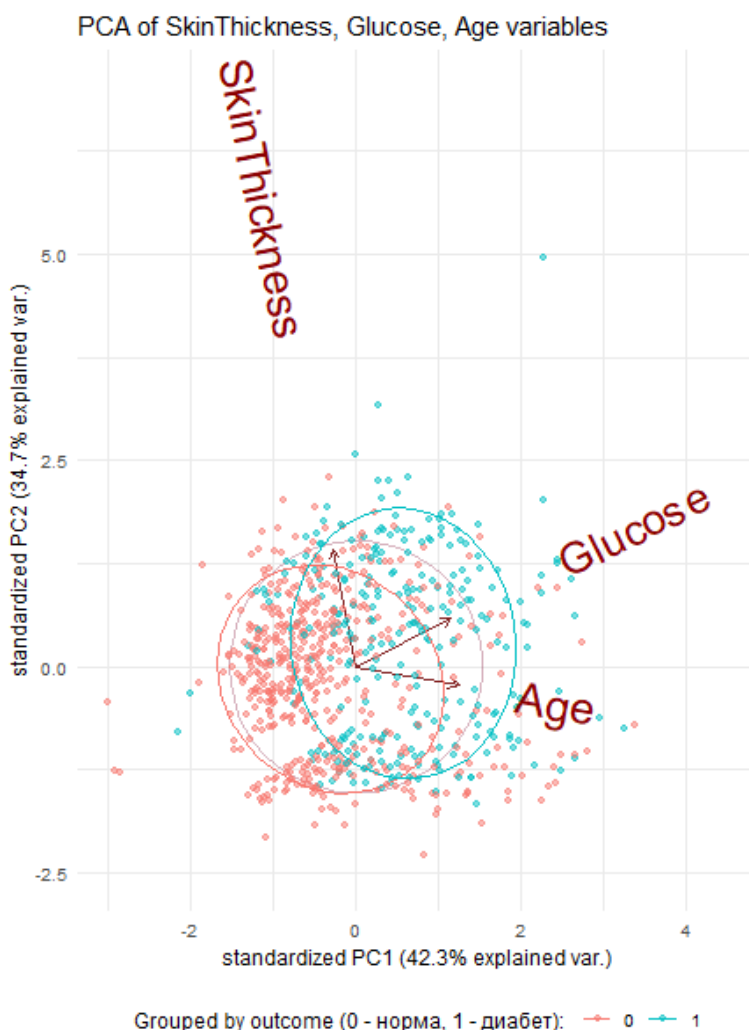


Рис. 2. График зависимости трех наиболее информативных компонент

Следующим методом анализа многомерных данных является многомерное шкалирование (MDS) – используется для визуализации сходства / различий между значениями из исходной таблицы путем построения точек на двумерных графиках. Возвращает оптимальное решение для представления данных в пространстве меньшего размера.[2] С помощью метода k – средних мы можем разбить получаемые точки на определенное количество кластеров. Метод k – средних наиболее популярный метод кластеризации, он стремится минимизировать суммарное квадратичное отклонение точек кластеров от центра этих кластеров. Для примера, представленного на рис. 3, использовались показатели: “SkinThickness”, “Glucose”, “Age” из исходной таблицы и желаемое количество групп для кластеризации равно 5.

Графики

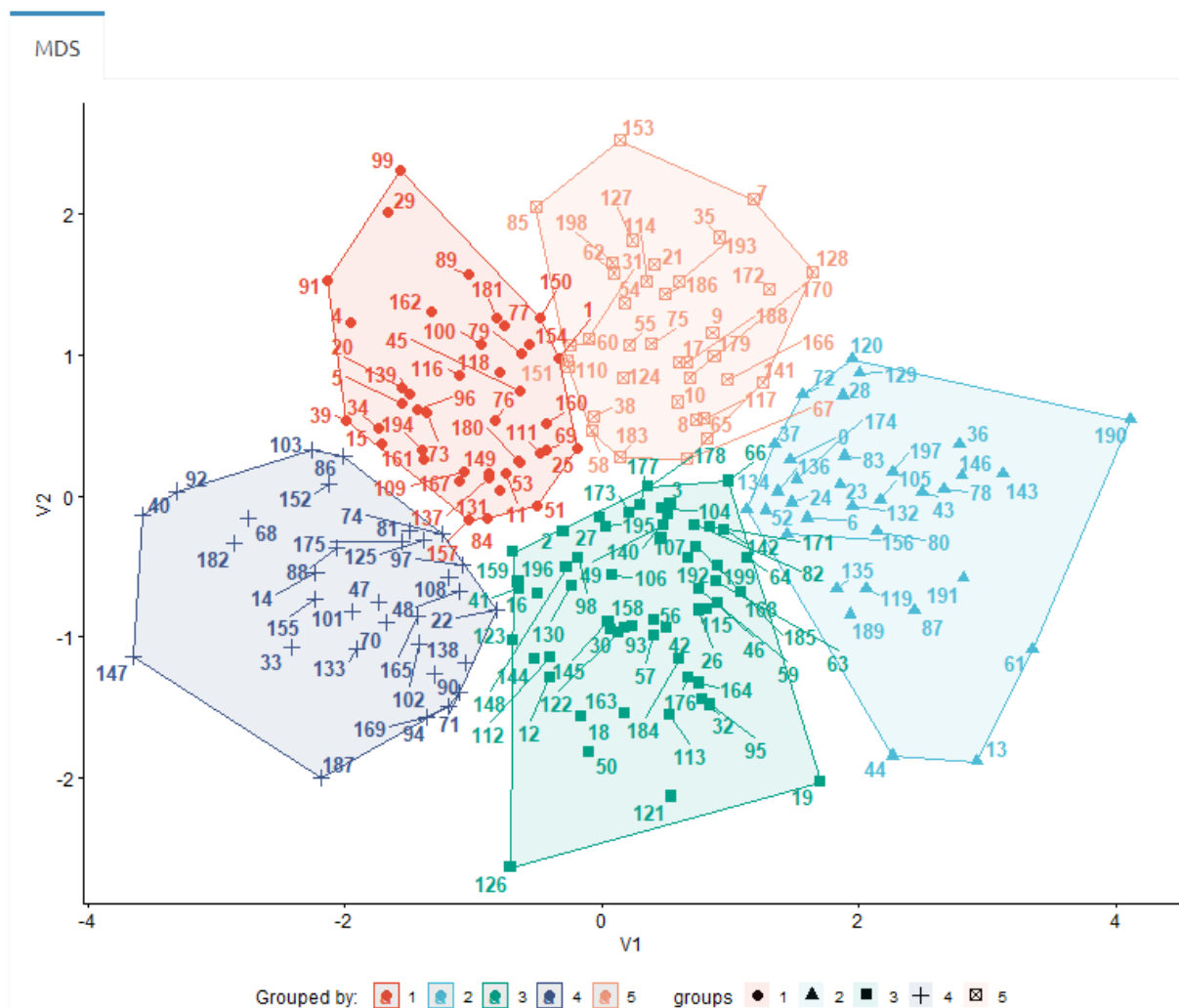


Рис. 3. Результат применения метода многомерного шкалирования

Другой тип визуализации это heatmap – тепловая карта. Это – графическое представление данных, где индивидуальные значения в таблице отображаются при помощи цвета. На рис. 4 изображены представления значений “SkinThickness”, “BMI”, “Age”, “Glucose”. Такое представление позволяет решить задачу о выяснении какие группы пациентов встречаются. Также тепловые карты можно построить таким образом, что порядок переменных и наблюдений будет изменен относительно исходного набора данных. Делается это для вычисления расстояний между каждой парой строк и столбцов и упорядочивания их по сходству, так глаз человека проще определяет зависимости. Обычно, порядок строк определяется путем выполнения иерархического кластерного анализа. Благодаря этому получается размещать похожие строки рядом друг с другом на графике. Порядок следования столбцов определяется аналогичным образом. Например, из рис. 4 видно, что в кластерах, обозначенных, как “1” и “2” наибольшее скопление записей, при которых был диагностирован диабет. Кластер “3” практически не содержит записей, при которых было бы диагностировано это заболевание.

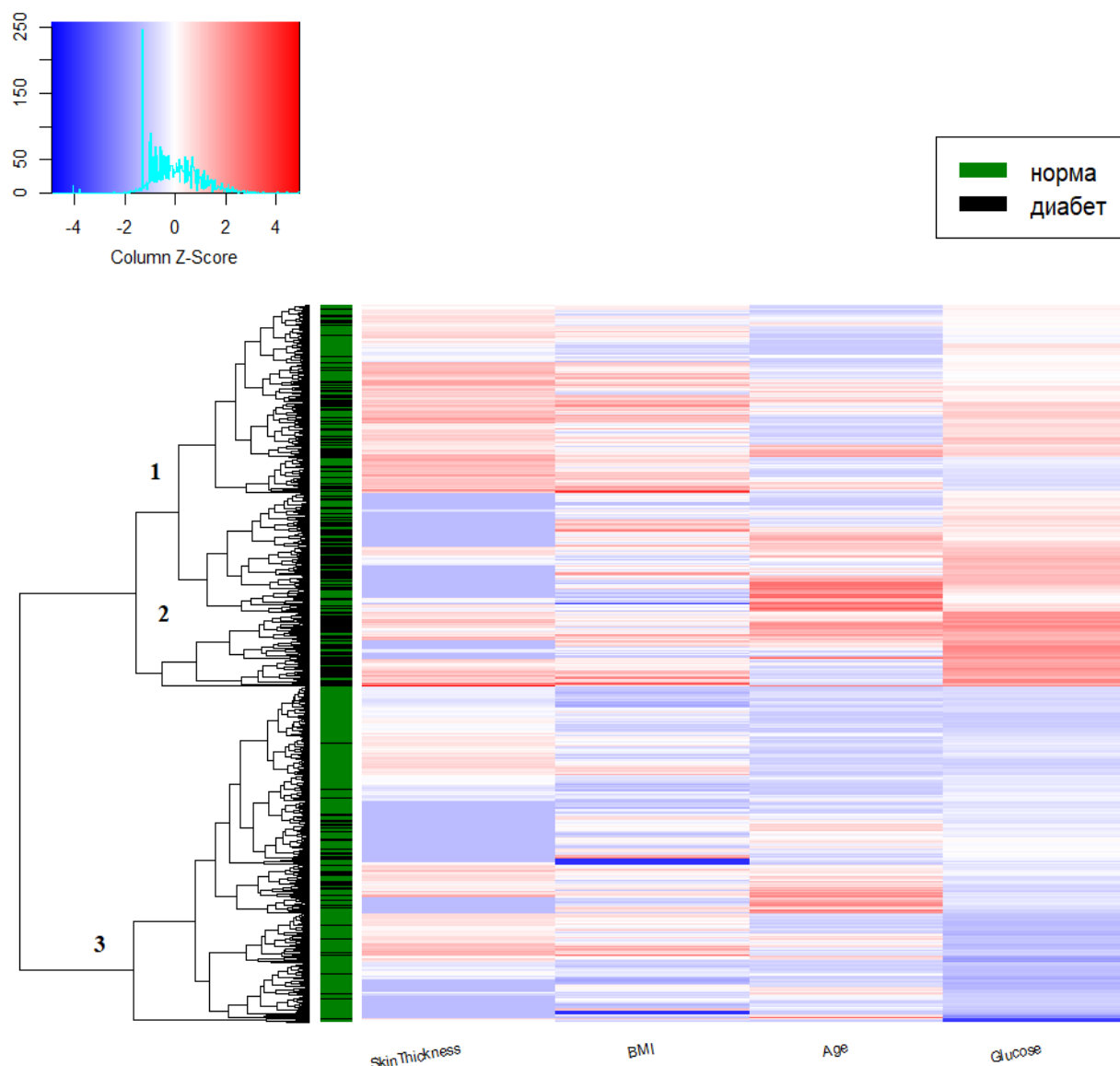


Рис. 4. Пример тепловой карты

При получении новых таблиц с данными бывает полезно отобразить их в наглядном графическом виде для первоначального изучения зависимостей или связей между переменными. Для таких целей очень полезен корреляционный матричный анализ. На графике, представленном на рис. 5 содержится информация о:

- 1) Распределении каждой переменной;
- 2) Диаграммах рассеяния с линиями, которые наилучшим образом приближаются к заданным точкам, которые изображены в нижней треугольной матрице;
- 3) Значениях корреляций с p – уровнем значимости в виде звезд, которые изображены в верхней треугольной матрице.[3]

Каждый уровень значимости связан с символом: p -значения: (0, 0.001, 0.01, 0.05) \Leftrightarrow символы: («***», «**», «*»).

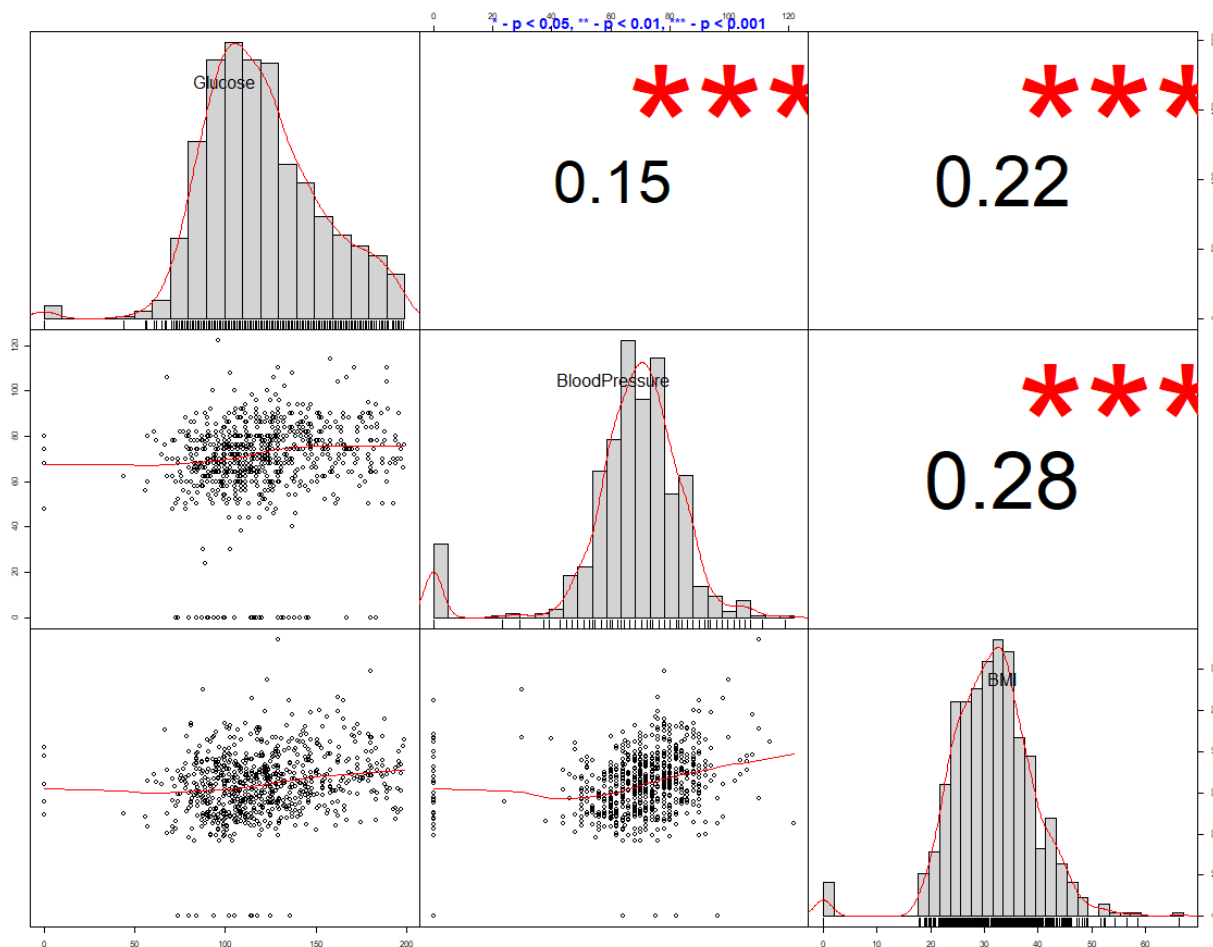


Рис. 5. Корреляционный матричный анализ

Как видно из приведенного обзора, существует широкий выбор средств для работы с медицинскими данными и представлении их в сжатом и наглядном для поиска зависимостей виде. Описанные здесь методы были реализованы в программе для работы с медицинскими данными с графическим интерфейсом пользователя, которая в настоящее время проходит апробацию в ПСПбГМУ им. акад. И. П. Павлова.

Библиографический список

1. Кабаков Р.И. R в действии. Анализ и визуализация данных в программе R. М.: ДМК Пресс, 2016. 588 с.
2. Уикем Хэдли, Гроулмунд Гарретт. Язык R в задачах науки о данных / пер. с англ. СПб.: Диалектика, 2018. 592 с.
3. Золотарюк А. В. Язык и среда программирования R. М.: ИНФРА-М, 2019. 162 с.

УДК 519.252

Л. А. Шубырева

магистр кафедры проблемно-ориентированных вычислительных комплексов

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

АЛГОРИТМЫ СОСТАВЛЕНИЯ КОМПЛЕМЕНТАРНЫХ ПАР ДЛЯ КЛИНИЧЕСКОГО ИССЛЕДОВАНИЯ

Введение

Часто во многих сферах деятельности возникают такие задачи, для решения которых не существует строгих алгоритмов или методов. В клинических исследованиях одной из проблем является подбор испытуемых. Для сравнения препаратов, группы должны быть не только схожи по исходным характеристикам, но и подвергаться схожему лечению и наблюдению во время исследования, за исключением получения испытуемого и контрольного лекарств.

Таким образом, возникает проблема составления клинических групп, сбалансированных по основным факторам. Например, если в группах будет существенно различаться возраст пациентов, будет невозможно различить непосредственную причину наблюдаемого эффекта – возраст или принимаемый препарат. Для контроля такой ситуации используется метод связанных пар (*matched pair analysis*). Это такой метод, при котором из двух множеств составляются пары (один пациент – из первого множества, во второй – из второго) так, чтобы мера схожести пациентов в паре была приемлемой.

Для решения такой задачи, связанной с анализом данных, больше всего подходит язык программирования R.

1. Причины выбора пакета *designmatch*

Пакет *designmatch* [1] реализует функцию построения совпадающих выборок, которые сбалансированы и представляют собой графически сопоставленные пары.

Данные пары:

- могут быть использованы для сравнения в наблюдательных исследованиях с основной группой и контрольной;
- могут быть использованы для разработки рандомизированных экспериментов.

Среди других, *designmatch* может использоваться для сопоставления в исследованиях лечения-контроля, а также в исследованиях случай-контрольного наблюдения; наблюдательных исследованиях с инструментальными переменными и разрывными моделями, а также для планирования рандомизированных экспериментов, например, для сопоставления до рандомизации. В комплект также входят функции для оценки ковариатного баланса в сопоставленных образцах (выборках).

2. Основные функции *designmatch*

Рассматриваемый пакет выпущен в июне 2018 год и обладает максимальным функционалом среди аналогов.

Пакет включает в себя функции для построения сбалансированных выборок, которые и представляют собой графически сопоставленные пары. Среди прочего, эти функции могут быть использованы для сравнения в наблюдательных исследованиях с основной и контрольной группами [2].

Описание *designmatch* включает в себя две функции для построения сопоставленных образцов, которые сбалансированы и представлены графически. Этими двумя функциями являются *bmatch* и *pmatch* для двудольного и не двудольного сопоставления соответственно. Обе функции включают в себя параметры для непосредственного уравнивания средних, моментов более высокого порядка и распределения наблюдаемых ковариат.

3. Функция Nmatch

Функция Nmatch представлена для оптимального двудольного сопоставления в рандомизированных экспериментах и наблюдательных исследованиях, которая непосредственно балансирует наблюдаемые ковариаты. Сопоставление позволяет пользователю применять различные формы ковариатного баланса в сопоставленных выборках, такие как баланс моментов (например, средних значений, дисперсий и корреляций), баланс распределения (например, точное равновесие, почти точное равновесие, прочность-балансировка), согласование и корректировка [3]. Среди прочего, nmatch можно использовать при разработке рандомизированных экспериментов, формирующих перед рандомизацией, и в наблюдательных исследованиях для сопоставления с размерами увеличением инструментальной переменной.

Рассмотрим пример, где задача – минимизировать общее расстояние между подобранными парами (Рис. 1).

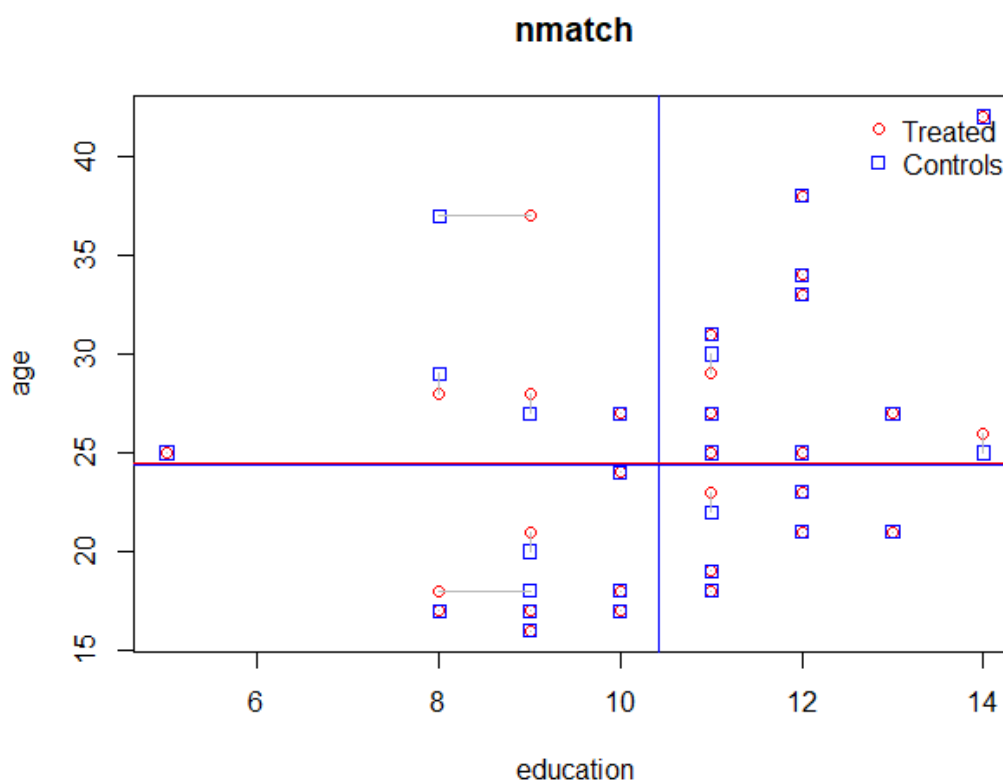


Рис. 1. Nmatch (minimum distance matching)

В этом примере нет требований к ковариатному балансу, то есть остальные параметры не учитываются (горизонтальные и вертикальные линии показывают средние значения обработанных единиц и соответствующих элементов управления для каждого из ковариат).

Оптимальное сопоставление подмножеств преследует две конкурирующие цели одновременно: минимизировать общую сумму ковариатных расстояний при сопоставлении как можно большего количества наблюдений. Компромисс между этими двумя целями регулируется параметром subset_weight. Результат такого компромисса можно увидеть на графике ниже (рис. 2).

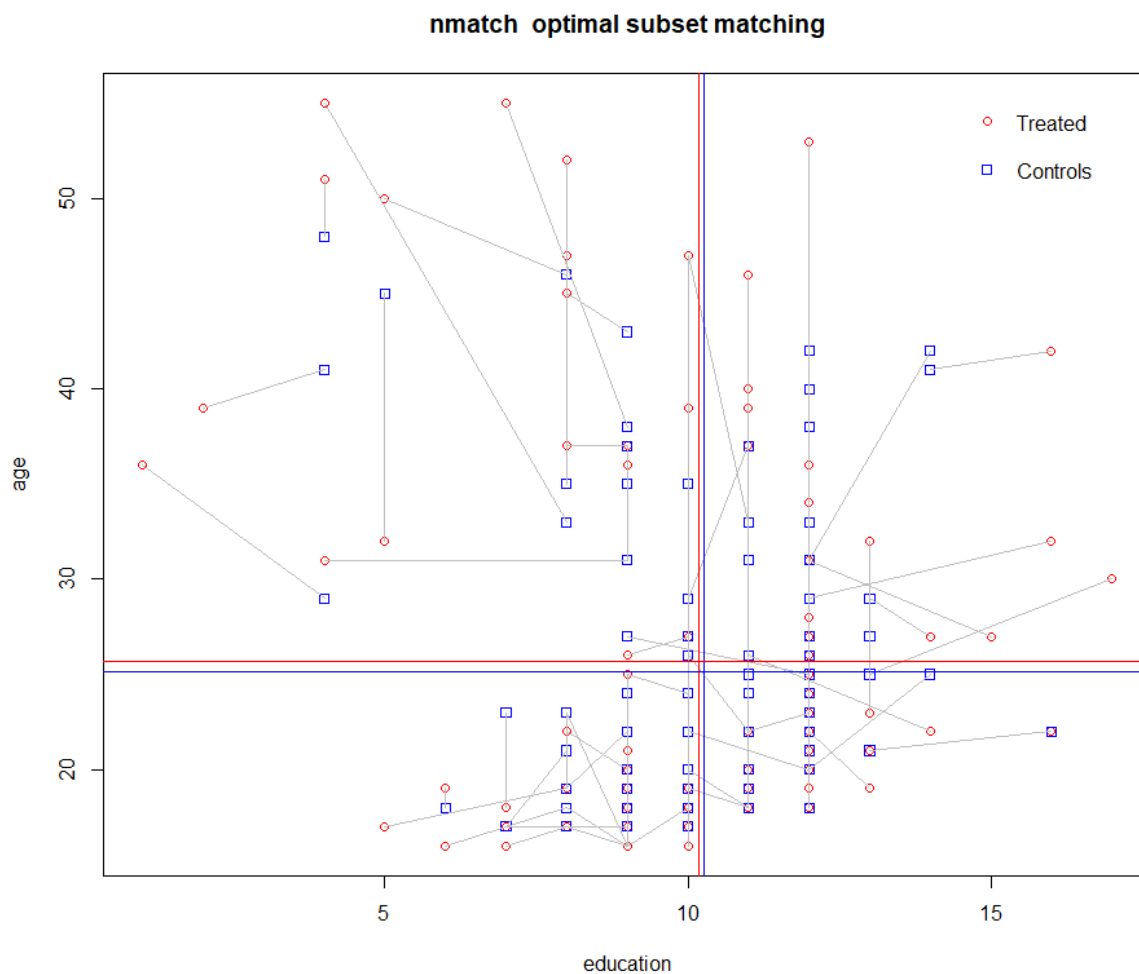


Рис. 2. Nmatch (optimal subset matching)

Здесь требования к балансу – это средний баланс, почти точный баланс и почти точное совпадение для разных ковариат.

4. Функция Bmatch

Bmatch – функция для оптимального недвудольного сопоставления в наблюдательных исследованиях, которая непосредственно балансирует наблюдаемые ковариаты. Bmatch может также использоваться в схемах разрыва путем сопоставления единиц (групп) в окрестности разрыва. Bmatch либо минимизирует общую сумму ковариатных расстояний между совпадающими единицами, максимизирует общее количество совпадающих единиц или оптимизирует комбинацию этих двух при условии соответствия, ковариатного балансирования и ограничений репрезентативности.

В следующем примере (рис. 3) функция находит наибольшую сопоставленную выборку пар, которая удовлетворяет требованиям баланса.

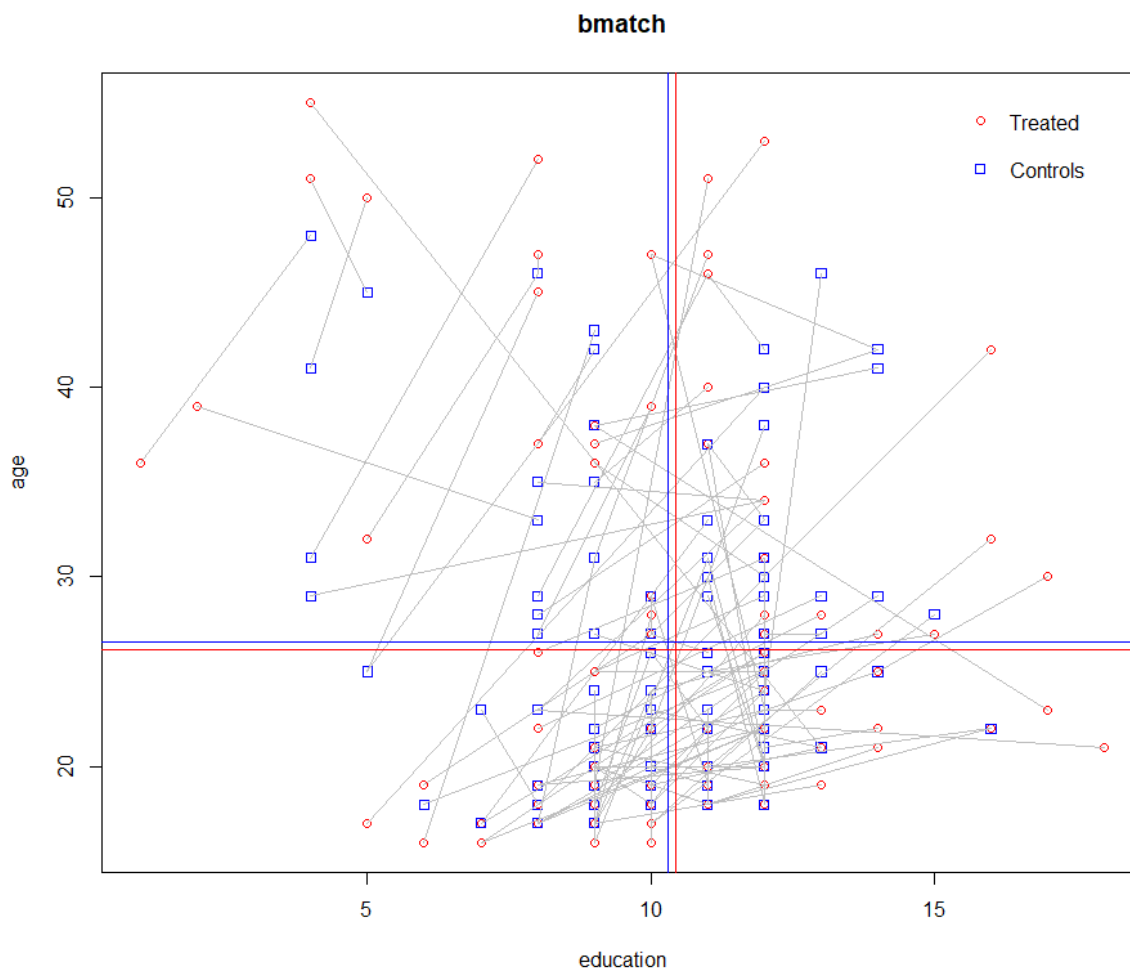


Рис. 3. Bmatch Cardinality matching

Здесь требования к балансу – среднее и максимальное соответствие для различных ковариат. Но для нашей темы такая реализация не совсем уместна, так как нам нужно в результате получить именно двудольное сопоставление.

Заключение

Таким образом, пакет `designmatch` отлично подходит для подбора испытуемых в клинических исследованиях, а именно использование функции `pmatch`. Использование функций пакета на проверочных данных показало хороший результат.

Вместо одновременного сопоставления по всем ковариатам `designmatch` отдельно сопоставляет группы лечения по интересующим параметрам. Для этого накладывается ограничения точного распределения или точного баланса.

Библиографический список

1. Designmatch: Matched Samples that are Balanced and Representative by Design // Cran.r-project. 2018. URL: <https://cran.r-project.org/web/packages/designmatch/index.html> (дата обращения: 25.03.2020).
2. Resa M., Zubizarreta R., Amiriparian S. et al. Snore Sound Classification Using Image-Based Deep Spectrum Features // *Statistics in Medicine*. 2016. Vol. 35. P. 4961–4979.
3. Keller B., Tipton E. Propensity Score Analysis in R: A Software Review // *Journal of Educational and Behavioral Statistics*. 2016. Vol. 41. P. 326–348.

УДК 621.396

А. А. Яковлев

студент кафедры проблемно-ориентированных вычислительных комплексов

В. И. Исаков – кандидат технических наук, доцент – научный руководитель

АЛГОРИТМ МОДЕЛИРОВАНИЯ ОТРАЖЕНИЙ ОТ КРОМКИ «ЗЕМЛЯ-МОРЕ»

В работе рассматривается моделирование помеховых отражений от земной и морской поверхностей. Такие сигналы являются помехой и мешают выделению и обработке информационных сигналов. Для аппроксимации распределения амплитуд сигналов, отраженных подстилающей поверхностью, при исследовании характеристик высокоточных локаторов используют логарифмически-нормальную плотность распределения, плотность распределения Вейбулла, составные распределения и другие плотности распределения.

При импульсном режиме локации сигнал, отраженный от протяженного объекта, представляет собой отрезки случайного узкополосного процесса. Эти импульсы характеризуются плотностью распределения вероятностей амплитуды A или мощности P , а также спектрально-корреляционными характеристиками.

В основном при аппроксимации отражений от морской глади используют логарифмически-нормальную плотность, а при отражении от земной чащи используют распределение Вейбулла, потому что эти распределения не противоречат экспериментальным данным и.

Радиолокационные станции (РЛС), работающие над морем, постоянно сталкиваются с отражением радиолокационных сигналов не только от объектов, которые необходимо обнаружить, но и от самой морской поверхности, что, чаще всего, приводит к помехам различной природы а, следовательно, и к ухудшению характеристик обнаружения. Помехи от морской поверхности нежелательны и могут помешать работе радара. Почти все бортовые радиолокационные станции, работающие над морем, сталкиваются с морскими помехами в независимости от того, используется ли этот радар для наблюдения за кораблями, низколетящими летательными аппаратами или объектами со сверхмалой эффективной поверхностью рассеяния (ЭПР), такими как перископы или телекоммуникационные буи подводных лодок [1].

При изучении морской поверхности, можно сделать вывод, что это не случайная неровная поверхность, но она имеет изменчивую структуру. При высоком разрешении РЛС статистическая модель радиолокационных отражений от взволнованной морской поверхности становится негауссовской.

Для РЛС с низким разрешением, под влиянием эффекта нормализации отраженного сигнала от морских неоднородностей, помехи имеют Гауссовнормальное распределение амплитуды (Рэлеевское распределение огибающей), что не справедливо для РЛС с высоким разрешением, поскольку в ячейку разрешения радара наравне с крупными неоднородностями попадает мелкая рябь, расплывенные частички и пена от перекатывающихся волн. Вследствие чего эффект нормализации плотности распределения вероятности радиолокационных отражений отсутствует, и статистическая модель помех становится негауссовской. Таким образом, для учета физических принципов, которые возникают при увеличении разрешающей способности радиолокатора, необходимо модифицировать Гауссовскую модель помех радиолокационных отражений от моря до составных негауссовских моделей.

Самыми популярными моделями, описывающими статистику огибающей морских помех являются Рэлеевская модель, лог-нормальная модель и модель Вейбулла. Логарифмически-нормальное распределение вероятности амплитуд в нашем случае удобно записать в следующем виде:

$$f(A) = \frac{1}{\sqrt{2\pi\sigma_A A}} \exp\left(-\frac{(\ln A - \ln \bar{A})^2}{2\sigma_A^2}\right),$$

где σ_A и $\ln \bar{A}$ — параметры распределения, связанные с математическим ожиданием, дисперсией и коэффициентом вариации соотношениями

$$M(A) = \bar{A} \exp\left(\frac{\sigma_A^2}{2}\right),$$

$$D(A) = \bar{A}^2 \exp(\sigma_A^2) (\exp(\sigma_A^2) - 1),$$

$$K_A = \sqrt{\exp(\sigma_A^2) - 1}.$$

Это распределение обладает следующим полезным свойством: при возведении случайной величины, распределенной по логарифмически-нормальной плотности, в любую степень, полученная случайная величина также имеет логарифмически-нормальное распределение. Получаем логарифмически-нормальное распределение мощности (1)

$$f(P) = \frac{1}{\sqrt{2\pi}2\sigma_P P} \exp\left(-\frac{(\ln P - \ln \bar{P})^2}{2\sigma_P^2}\right) = \frac{1}{\sqrt{2\pi}2\sigma_A P} \exp\left(-\frac{\left(\ln P - \ln \frac{\bar{A}}{2}\right)^2}{2(2\sigma_A)^2}\right), \quad (1)$$

где σ_P и \bar{P} — параметры распределения, связанные с математическим ожиданием, дисперсией и коэффициентом вариации, при замене в них A (включая индексы) на P . Получаем соотношения, связывающие параметры плотностей распределений мощностей и амплитуд:

$$\sigma_P = 2\sigma_A, \quad \bar{P} = \frac{\bar{A}^2}{2}.$$

Для определения параметров распределений для конкретных условий наблюдения морской поверхности необходимо определить параметры распределений через среднюю мощность отражений \bar{P} , получаем

$$\sigma_P = \sqrt{\ln(1 + K_p^2)},$$

$$\bar{P} = \frac{\tilde{P}}{\ln(1 + K_p^2)}, \quad \sigma_A = \frac{\sigma_P}{2} = \frac{1}{2} \sqrt{\ln(1 + K_p^2)}, \quad \bar{A} = \sqrt{2\bar{P}} = \frac{\sqrt{2\tilde{P}}}{\sqrt{\ln(1 + K_p^2)}}.$$

Средняя мощность отраженного сигнала \tilde{P} определяется по известному уравнению радиолокации, которое позволяет учесть и характеристики локатора, и условия наблюдения поверхности моря [3, 4]. Выражение для коэффициента вариации при различных условиях морской поверхности, ее состояния и параметров локатора приведено в работе [5].

Распределение Вейбулла. Отражения от земной поверхности. Распределение Вейбулла представляет собой одно из наиболее удобных распределений для моделирования помех от земной поверхности при высоком разрешении. Запишем его в следующем виде:

$$f(A) = \alpha_A \lambda_A A^{\alpha_A - 1} \exp(-\lambda_A A^{\alpha_A}),$$

где α_A , λ_A — параметры распределения, связанные с математическим ожиданием, дисперсией и коэффициентом вариации.

Данное распределение обладает следующим свойством: при возведении случайной величины, распределенной по закону Вейбулла в любую степень, полученная случайная величина имеет распределение Вейбулла. Осуществляя преобразование (5), получаем соответствующее распределение мощности:

$$f(P) = \frac{\alpha_A}{2} \left(\lambda_A 2^{\frac{\alpha_A}{2}} \right) P^{\frac{\alpha_A}{2} - 1} \exp\left(-\left(\lambda_A 2^{\frac{\alpha_A}{2}} \right) P^{\frac{\alpha_A}{2}}\right),$$

где α_P и λ_P — параметры распределения, связанные с математическим ожиданием, дисперсией и коэффициентом вариации. Получаем соотношения, связывающие параметры плотностей распределений мощностей и амплитуд:

$$\alpha_P = \frac{\alpha_A}{2}, \quad \lambda_P = \lambda_A 2^{\frac{\alpha_A}{2}}.$$

Для определения параметров распределений для конкретных условий наблюдения земной поверхности необходимо определить параметры распределений через среднюю мощность отражений \tilde{P} и коэффициент вариации мощности K_P . В отличие от логарифмически-нормального распределения, параметры распределения мощности α_P и λ_P получаем в общем случае из решения системы уравнений

$$\frac{1}{\lambda_P^{\alpha_P}} \frac{\Gamma\left(\frac{1}{\alpha_A}\right)}{\alpha_P} = \tilde{P},$$

$$2\alpha_P \frac{\Gamma\left(\frac{2}{\alpha_A}\right)}{\Gamma^2\left(\frac{1}{\alpha_A}\right)} = 1 + K_P.$$

В некоторых частных случаях решение системы можно получить аналитически, выразив в явном виде параметры α_P и λ_P через мощность отражений \tilde{P} и коэффициент вариации мощности K_P .

Отражения от кромки земля-море. Составное векторное распределение

Исключением является ситуация, когда сигналы попадают на бортовую или спутниковую РЛС как с земли, так и с морской поверхности. В таком случае наблюдаемый сигнал будет представлять собой векторную сумму сигналов.

Пусть площадь элемента разрешения РЛС равна S , и в этот элемент попадают фрагменты участков морской S_M и земной S_3 поверхностей, т. е.

$$S = S_M + S_3 = \gamma S + (1 - \gamma)S,$$

где $S_M = \gamma$ и $S_3 = (1 - \gamma)$ — площади соответствующих участков. Тогда средние мощности сигналов, пропорциональны коэффициентам γ и $(1 - \gamma)$.

Алгоритм моделирования отражений от кромки может быть представлен в разных видах. Запишем алгоритм пошагово, [6,7].

Алгоритм моделирования

1. Определяется (или задается) деление элемента разрешения на области отражений от моря и земной поверхности, которое определяет коэффициент γ .

2. Вычисляются по выражениям (17) – (20), с учетом замены \tilde{P} на $\gamma\tilde{P}$, параметры закона распределения (10) $\sigma_A^{(M)}$ и $\bar{A}^{(M)}$, где верхний индекс в скобках определяет отражения от морской поверхности.

3. Вычисляются по выражениям (26) – (28), с учетом замены \tilde{P} на $(1 - \gamma)\tilde{P}$ параметры закона распределения (21) $\alpha_A^{(3)}$ и $\lambda_A^{(3)}$, где верхний индекс в скобках определяет отражения от земной поверхности.

4. Вычисляются $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$ — псевдослучайные, совместно независимые числа, равномерно распределенные на интервале $(0, 1)$.

5. Вычисляется амплитуда $A_{(m)}$ сигнала, обусловленного отражениями от фрагмента морской поверхности:

$$A_{(M)} = \bar{A}^{(M)} \exp\left(\sigma_A^{(M)} \sqrt{2 \ln \varepsilon_1} \cos(2\pi \varepsilon_2)\right),$$

где нижний индекс в скобках определяет отражения от морской поверхности.

6. Вычисляется амплитуда $A_{(3)}$ сигнала, обусловленного отражениями от фрагмента земной поверхности

$$A_{(3)} = \alpha_A^{(3)} \sqrt{\frac{1}{\lambda_A^{(3)}} \ln \varepsilon_3},$$

где нижний индекс в скобках определяет отражения от земной поверхности.

7. Вычисляется амплитуда A сигнала, обусловленного отражениями от кромки земля-море:

$$A = \sqrt{A_{(3)}^2 + A_{(M)}^2 + 2A_{(3)}A_{(M)} \cos(2\pi \varepsilon_4)},$$

при этом полагаем, что разность фаз векторов сигналов, отраженных от фрагментов земной и морской поверхности, равномерно распределена на интервале $(0, 2\pi)$.

Представленный алгоритм моделирования при $\gamma=1$ формирует сигнал, отраженный только от моря, а при $\gamma=0$ — только от земли. Следует отметить, что представленный подход позволяет в алгоритме при соответствующей модификации использовать и другие аппроксимации законов распределения флуктуаций [7].

Таким образом, алгоритм, позволяет моделировать локационные сигналы, отраженные как от кромки земля-море, так и по отдельности.

Библиографический список

1. Валеев В. Г., Милащенко Е. А. Прогнозирование характеристик обнаружения мелких морских целей с учетом негауссовских морских отражений применительно к когерентно-импульсной РЛС. URL: <http://jre.cplire.ru/alt/mar14/2/text.html> (дата обращения: 12.03.2020).
2. Шенета Д. А., Подоплекин Ю. Ф., Исаков В. И. Марковская модель флуктуаций амплитуд и длительностей эхо-сигналов крупных надводных объектов // Морской вестн. 2016. № 3. С. 49–50.
3. Шенета А. П., Сергеев М. Б., Блаунштейн Н. Ш. Прикладные аспекты электродинамики. СПб.: Аграф+, 2016. 272 с.
4. Шенета Д. А., Ненашев В. А., Подоплекин Ю. Ф. Моделирование входных сигналов бортовой РЛС, обусловленных отражениями зондирующего сигнала от подстилающих поверхностей земли и моря // Морской вестн. 2016. № 4. С. 69–71.
5. Тверской Г. Н., Харченко И. П., Терентьев Г. К. Имитаторы эхо-сигналов судовых радиолокационных станций. Л.: Судостроение, 1973. 228 с.
6. Садовский И. Н., Поспелов М. Н., Кузьмин А. В. Микроволновые радиометрические исследования морской поверхности в прибрежной зоне Черного моря. М.: ГРАНП-Полиграф, 2005. С. 103–110.
7. Шенета Д. А., Исаков В. И. Моделирование локационных сигналов, отраженных от кромки земля-море. URL: <https://cyberleninka.ru/article/n/modelirovanie-lokatsionnyh-signalov-otrazhennyh-ot-kromki-zemlya-more> (дата обращения: 10.03.2020).
8. Михайлов В. Н., Кутузов В. М. Методика расчета ЭПР отражений от морской поверхности при оценке зоны видимости морской РЛС. URL: <https://etu.ru/assets/files/nauka/nii/Prognoz/Publikacii/kutuzov-hanoi-2017.pdf> (дата обращения: 15.03.2020).

Компьютерные технологии и программная инженерия

УДК 004.942

А. К. Борисов

студент кафедры компьютерных технологий и программной инженерии

М. Д. Поляк – старший преподаватель – научный руководитель

АВТОМАТИЧЕСКОЕ АННОТИРОВАНИЕ ИЗОБРАЖЕНИЙ НА РУССКОМ ЯЗЫКЕ С ИСПОЛЬЗОВАНИЕМ НЕЙРОСЕТЕВОЙ МОДЕЛИ

Введение

Автоматическое аннотирование изображений есть процесс, преобразующий цифровое изображение в текст, который может быть как осмысленным предложением, так и простым набором из ключевых слов. Оно находит своё применение, в основном, в области компьютерного зрения для организации и поиска изображений в базах данных. Каждый пользователь интернета когда-нибудь искал в поисковых сервисах картинки по текстовому описанию, или запрашивал поиск похожих, на указанное нами, изображений. Такая возможность появилась благодаря кластеризации цифровых изображений в базах данных по определённым меткам, которые обобщают их по определённым признакам. Например, знаменитую картину Леонардо да Винчи “Мона Лиза” визуально можно разбить на следующие фразы: “портрет”, “девушка с тёмными волосами”, “живопись”, “одежда эпохи ренессанса”. С помощью этих фраз картину можно определить в соответствующие секторы с изображениями, объединённых по общему признаку. Именно благодаря подобным обобщениям стал возможен поиск графической информации по её метаданным.

В современном мире, где каждый день генерируется неисчисляемое количество новой информации, львиную долю которой занимает медиа контент, встаёт вопрос о том, каким образом можно автоматизировать аннотирование изображений, описанное выше. Очевидно, что ручной силы людей не хватит, чтобы описать хотя бы тысячную долю от всех изображений, представленных в интернете, более того, все эти аннотации должны со временем дополняться новыми ключевыми словами и обрастать более глубокой детализацией для обеспечения более качественного поиска и, вследствие этого, удовлетворения запроса. Ответом на данный вопрос об автоматизации стали нейронные сети, которые давно перестали быть эфемерной, непостижимой и неподъёмно-ресурсозатратной вещью, окутанной скептицизмом и недоверием. Уже больше 5 лет они находятся в авангарде технического прогресса и служат прикладной науке.

Прекрасным примером их использования на практике и стало автоматическое аннотирование изображений. Нейронные сети решили сразу две вышеперечисленные проблемы: обработка большого количества данных, циркулирующих в интернете, и поддержка более глубокой детализации с течением времени.

Глубина детализации – фактор, определяющий насколько большее количество признаков сможет извлечь из цифрового изображения нейронная сеть, а также насколько точно и подробно эти признаки потом преобразует в текст. Тут важно отметить принципиально важную разницу между извлечением признаков из цифрового изображения, и преобразованием их в текст, поскольку это две разные задачи, которые выполняют две разные нейронные сети. Глубина, с которой нейронная сеть описывает изображение, вещь довольно тонкая, но при этом поддающаяся настройке. Эта своеобразная “метрика” зависит от подобранной архитектуры нейронной сети. Люди, которые так или иначе знакомы с принципами её работы знают, что чем более сложная и развитая архитектура, тем качественнее и детальнее результат её работы. Лучшее качество результата достигается путём долгого перебора всех возможных архитектур, призванных выполнять какую-то конкретную задачу. Всё это осложняется тем, что в данном подходе нет какого-то определённого свода правил или алгоритма для решения всех проблем: по большей части это напоминает охоту на уток, только не в ясную погоду, где

стрельба ведётся прицельным огнём по птице, а в туманную, когда нет никакой видимости и не остаётся ничего иного, как бесцельно стрелять в воздух в надежде попасть хотя бы в кого-нибудь.

В процессе знакомства с архитектурами Н.С. у людей невольно возникает вопрос: “Если качество работы нейронной сети так сильно зависит от её размеров, то почему просто не сделать её как можно более сложной и разнообразной? Ведь в таком случае решается вся эта комплексная проблема с «нейросетевым казино», где все её архитектуры разложены в каком-то странном и неподдающемся логике порядке, известном одной лишь фабрике с киоском, где их заряжают”. В ответе на этот вопрос кроется другая, неочевидная, но критично важная деталь. Чем более масштабная и сложная нейронная сеть, тем, больше признаков она вычленяет из входных данных, это правило является общим для всех Н.С. При, казалось бы, очевидных преимуществах этот фактор может оказывать губительную избыточность для задач, где входные данные имеют небольшой размер или признаки, вычленяемые нейронной сетью, довольно легко определяемы или обособляемы.

Архитектура нейронной сети

Частный случай архитектуры нейронной сети для автоматического аннотирования изображений, как описано выше, вбирает в себя не одну, а сразу две нейронные сети, каждая из которых имеет свою функцию, имеет свои нюансы и имеет свои выходные данные. Одна находит визуальные признаки, другая эти признаки конвертирует в текст.

Первая нейронная сеть является свёрточной. Это широко распространённая архитектура, захватившая пальму первенства в задачах визуального распознавания изображений.

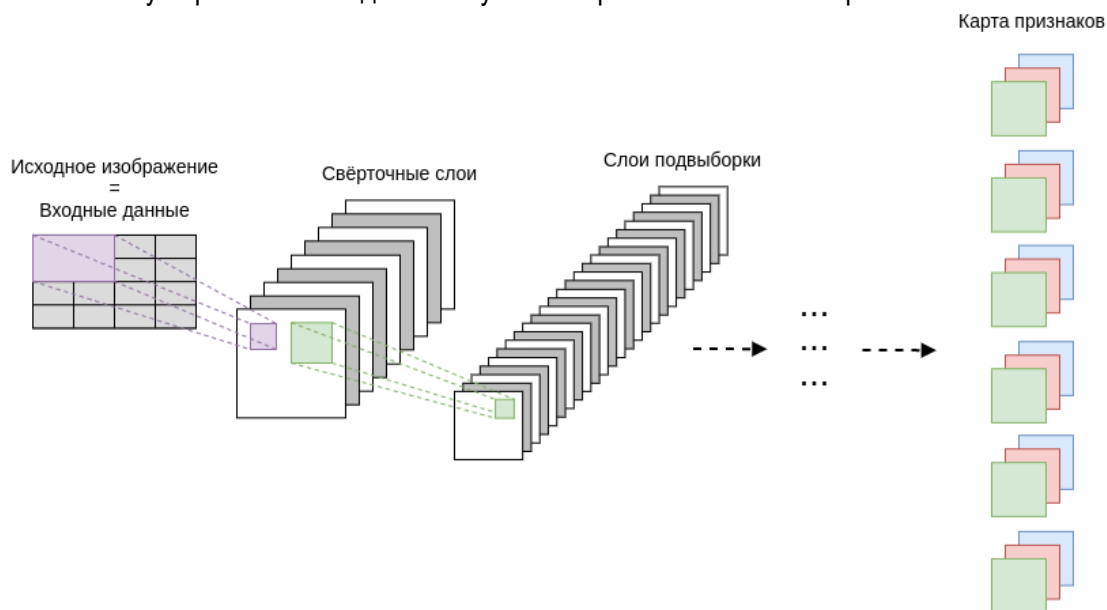


Рис.2. Пример входных данных свёрточных нейронных сетей.

Результатом работы любой свёрточной сети является конечная карта признаков, которую можно использовать в тех или иных целях. В задачах автоматического аннотирования изображений карта признаков будет использоваться в качестве входных данных для второй нейронной сети – рекуррентной.

Рекуррентная нейронная сеть является вторым китом, на котором стоит автоматическая аннотация изображений. Она отвечает за генерацию текста на основе карты признаков получаемой в результате свёртки. Её можно интерпретировать как “переводчика” с машинного, на человеческий, потому что её работа заключается в описании человеческим языком того, что было увидено свёрточной нейронной сетью.

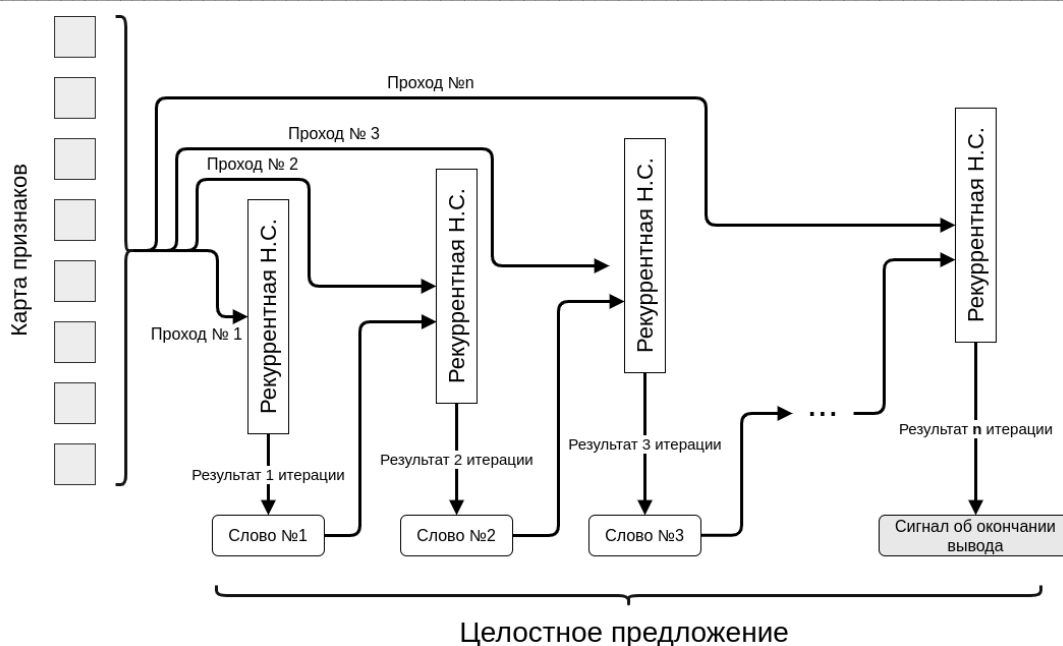


Рис. 3. Схема Рекуррентной нейронной сети.

Автоматическое русскоязычное аннотирование цифровых изображений

Словарный запас рекуррентной нейронной сети определяется размером последнего слоя, т.к. чем больше в нём элементов, тем больше вариация слов. Стоит отметить, что английский язык, технически, наиболее удобен для работы подобной нейронной сети, поскольку ни для кого не секрет, что он является более компактным как со словарной точки зрения, так и с точки зрения его синтаксиса. Это позволяет экономить на количестве нейронов на последнем слое, уменьшается количество связей, соответственно увеличивается скорость работы и ёмкость языка. Но это только то, что плавает на поверхности. Легковесность английского языка позволяет пренебречь экономией на разных формах слова в угоду частоты их использования. Иными словами, при работе с английским языком, в котором простор всевозможных окончаний ограничивается парой возможных вариаций и, вместе с этим, обладающий небольшим количеством исключений форм слов, можно спокойно увеличить общее число выходных нейронов, жертвуя сравнительно небольшими потерями в производительности. При таком подходе очень значительна угода утраты необходимости какой-либо постобработки полученных в конечном итоге результатов.

Сравнение подходов к проектированию Н.С. для англо и русскоязычного автоматического аннотирования можно продемонстрировать на этой интернет-шутке, появившейся в интернете около 10 лет назад:

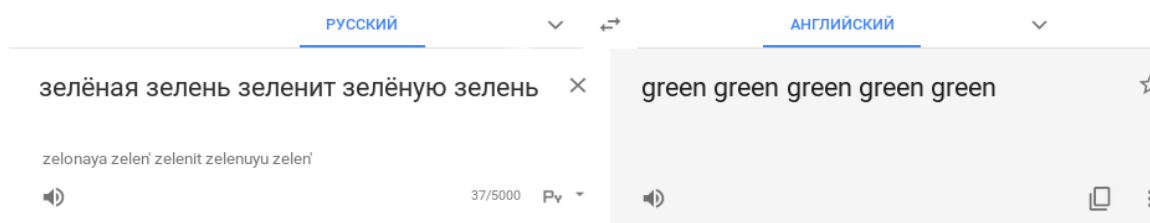


Рис.4. Разница описательной глубины русского и английского языка.

Это яркий пример того, насколько богат русский язык на детализацию изречений. Вместе с этим это яркий пример того, когда великий и могучий русский язык стреляет себе в ногу.

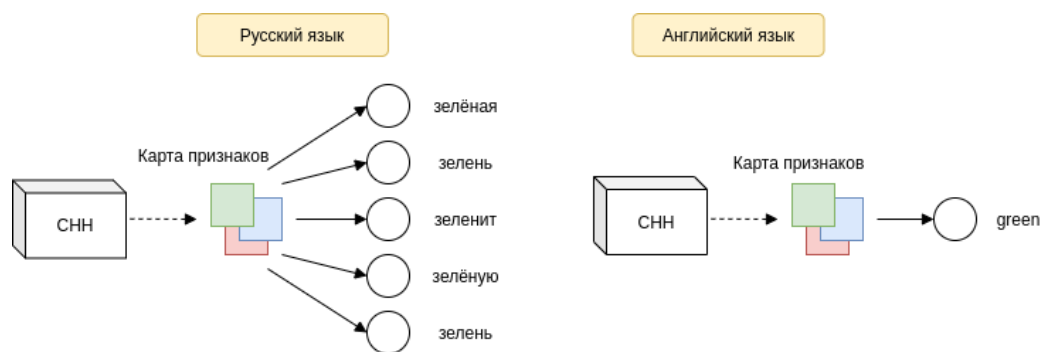


Рис. 5. Разница требований к архитектуре Н.С. англ. и русс. языков

С точки зрения описательной глубины языка – это прекрасный показатель, который позволит русскому языку стать в будущем доминирующим средством кластеризации изображений, обеспечивающим огромные возможности для предоставления более качественного, продуманного и уместного результата поиска. Более того, он будет выдавать более детальные описания карты признаков, полученной со свёрточной нейронной сети.

Однако, с другой точки зрения, а именно реализации системы кластеризации изображений в условиях ограниченности ресурсов можно увидеть, что эта глубина на Рис. 7 оборачивается принудительной, излишней избыточностью, которой никак нельзя пренебречь. Эта проблема стоит особенно остро в так называемой высокоуровневой кластеризации, когда выделяются совсем поверхностные и общие признаки для обеспечения быстрой скорости обработки изображений “вскользь”. Иными словами, в данном случае то, что русский язык описал в пяти формах одного слова английский язык уместил всего в одно. Естественно, с потерей смысла в виде очень грубой формы дискретизации, однако это ситуация, когда всё богатство русского языка является не преимуществом, а недостатком. При работе с английским языком, в случае возникновения необходимости иметь более глубокую детализацию на выходе из Н.С., можно просто дополнить словарь множеством литературных слов и, напротив, при возникновении обратной необходимости – убрать всё лишнее. В русском же языке отсутствует подобная свобода выбора и гибкость в манёврах. Огромное количество форм (среднем от 7 до 10), включая множественные числа, падежи, суффиксы, приставки, окончания, исключения, усложняет самый тривиальный процесс автоматического аннотирования изображений. Если снисходительно установить среднее значение в 5 форм (как в примере выше) то можно увидеть, что требование к количеству нейронов на выходе рекуррентной сети увеличивается в 5 раз (!), по сравнению с такой же нейронной сетью на английском языке.

Последняя и очень важная часть в вопросе Н.С. в задачах русскоязычного аннотирования это поиск и составление тренировочной выборки для глубокого обучения этой нейронной сети. Данную проблему можно решить с помощью двухэтапной подготовки англоязычного сета, достаточно больших размеров: его загрузка, анализ, перевод и его оценка. Роль большого датасета могут на себя взять два известных общедоступных набора размеченных картинок. Одним из них является специальный датасет от MSCOCO, который был размечен специально для обучения нейронных сетей в целях аннотирования изображений. В дополнение к его размерам в ~220 тысяч картинок можно включить датасет от Google, который имеет в своём арсенале более 3 300 000 размеченных изображений. Датасет от Google не находится в каком-то централизованном месте, а разрознен по всему интернету т. к. компания Google просто выпустила на волю свою собственную обученную нейронную сеть, которая проходила по сайтам, находила изображения, аннотировала их, и записывала в таблицу формата .tsv с ячейками пар: “аннотация” – “ссылка на изображение”. Весь датасет целиком занимает около 500GB физического пространства поэтому никто не выкладывает его в общий доступ в цельном виде или, хотя бы, разбив по частям. Для будущих пользователей данного датасета стоит отметить тот факт, что размер данного датасета мельчает с каждым днём, как титаник на дне атлантического океана. Происходит это потому, одни сервера пропадают, другие закрываются, у третьих перекрывается доступ, четвёртые – переезжают и т.д. Естественно, он не пропадёт целиком, но, например, за последние полтора года он уменьшился в размерах почти на пятую часть, см. Рис. 8.

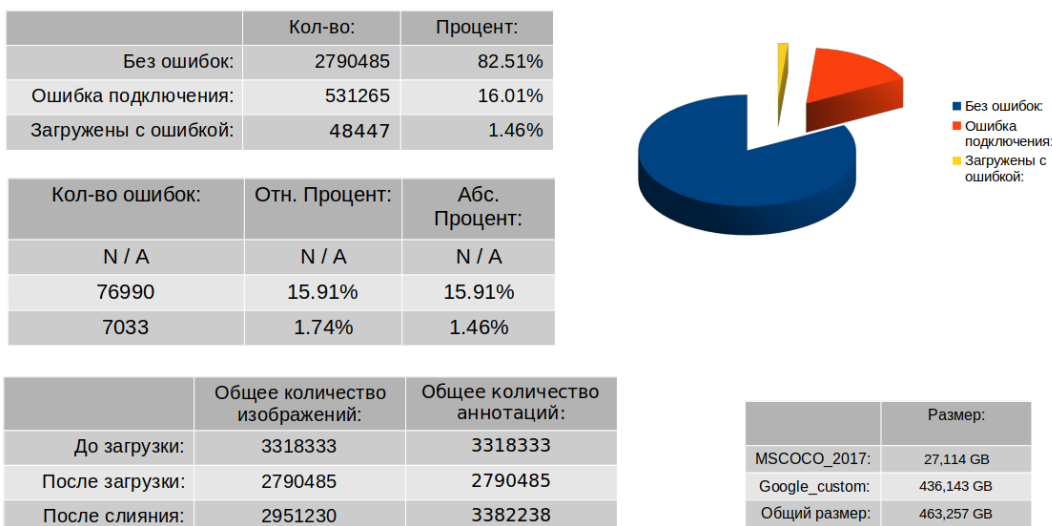


Рис. 6. Анализ данных загруженного трен. датасета от Google.

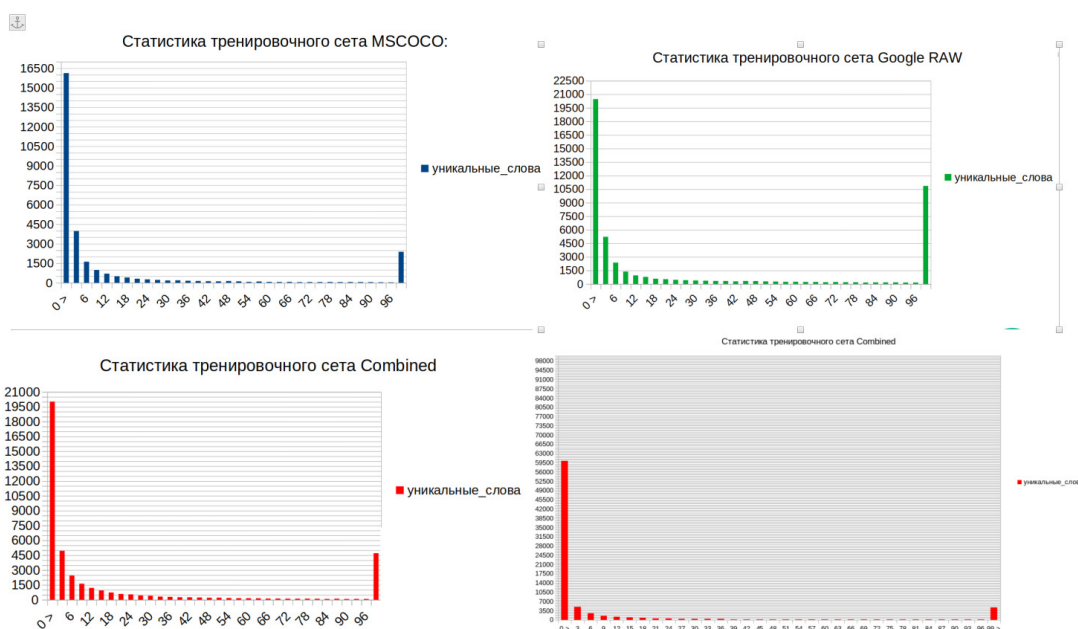


Рис. 7. Общая статистика тренировочных сетов.

В качестве решения задачи перевода можно использовать выпускную квалификационную работу одного из выпускников ГУАПа 2019 года, результатом которой стала обучена нейронная сеть для перевода с английского на русский. Используя данный подход стоит отметить, что словарный запас нейронной сети для автоматического аннотирования изображений приобретает новую прямую зависимость: словарный запас переводчика, который используется для преобразования англоязычных аннотаций в русскоязычные.

На Рисунке 8 представлено количество вхождений уникальных слов в каждую из обучающих выборок. Наибольшую практическую ценность имеют уникальные слова с вхождениями > 100, потому что в конечном итоге будут активно использоваться рекуррентной нейронной сетью. Данные уникальные слова представляют из себя реальный словарный запас нейросети, потому что он оценивается не только количеством нейронов на последнем слое, но и умением нейросети активно к ним обращаться. На четвёртом графике представлена переведённая русскоязычная выборка, на которой можно увидеть разнообразие слов и предварительно оценить требования к архитектуре. Закономерно, что уникальных вхождений в русском языке в несколько раз больше, соответственно словарь,

который нейронная сеть будет использовать для описания будет глубже, однако вместе с этим количество слов, которое на практике использовано не будет, тоже выросло. Отсюда можно сделать вывод, что в сухом остатке нейросетевая модель для аннотирования русскоязычных изображений будет использовать по существу лишь 1/15 долю от всех нейронов на конечном слое, что составляет 1/15 часть от изначального словарного запаса закладываемого в неё на этапе проектирования.

Заключение

В процессе исследования были изучены различные литературные источники, описывающие принципы работы нейронных сетей и их архитектур. В свою очередь, это позволило спрогнозировать возможности будущей системы, оценить целесообразность и провести глубокий анализ данных, прежде чем переходить к более трудо- и ресурсоёмким процессам. На данный момент ведется обучение данной нейронной сети на серверных вычислительных мощностях с целью её дальнейшей прикладной реализации в области баз данных.

Библиографический список

1. *Воронцов К. В.* Математические методы обучения по прецедентам (теория обучения машин). URL: <http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf> (дата обращения: 15.03.2020).
2. Smola A. and S.V.N. Vishwanathan Introduction to Machine Learning. URL: <http://alex.smola.org/drafts/thebook.pdf> (дата обращения: 15.03.2020).

УДК 519.246.8

А. В. Васильева

студент кафедры компьютерных технологий и программной инженерии

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

МОДЕЛИРОВАНИЕ И ПРОГНОЗИРОВАНИЕ СТОХАСТИЧЕСКИХ НЕЛИНЕЙНЫХ МНОГОМЕРНЫХ ВРЕМЕННЫХ РЯДОВ НА ОСНОВЕ МОДЕЛИ ИНТЕГРИРОВАННОЙ АВТОРЕГРЕССИИ И СКОЛЬЗЯЩЕГО СРЕДНЕГО

Решение многих задач экологии, экономики, медицины и других областей связано с составлением прогнозов, точность которых является важным решающим фактором. Для получения необходимой точности прогноза и его своевременности требуется использование эффективных математических методов и алгоритмов. ARIMA является одним из таких алгоритмов. Его точность подтверждается в использовании с одномерными временными рядами. Однако многомерная реализация прогноза методом ARIMA это трудоемкая задача для режима реального времени, а программное обеспечение для решения данных задач в общем виде отсутствует.

Цель исследования – разработать алгоритм, осуществляющий прогнозирование многомерных временных рядов с использованием модели ARIMA. Сопутствующие частные задачи, реализующие решение основной проблемы, следующие:

- 1) выбор многомерных временных рядов для осуществления прогнозирования;
- 2) построение структурной схемы алгоритма ARIMA для решения задачи прогнозирования;
- 3) построение процедуры для проверки значимости полученного прогноза;
- 4) построение прогноза на основании коллектива алгоритмов;
- 5) проверка значимости прогноза коллектива алгоритмов;
- 6) построение решающего правила для получения общего прогноза;
- 7) создание приложения, работающего в режиме, близком к реальному времени.

Метод Бокса-Дженкинса основывается на том, что нестационарный временной ряд путём взятия разностей некоторого порядка (d) можно свести к стационарному, то есть к случаю, для которого разработаны методы анализа и прогнозирования. В общем случае модель обозначается в виде формулы следующим образом: ARIMA(p, d, q), где p – порядок авторегрессии (приближение параметра подбираем по частичной автокорреляционной функции ЧАКФ); d – порядок интегрирования; q – порядок скользящего среднего (приближение параметра подбираем по частичной автокорреляционной функции АКФ). Когда порядок интегрирования получен и исходный ряд имеет стационарный вид, модель приобретает вид ARMA (p, q):

$$X_t = \alpha_0 + \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \dots + \alpha_p X_{t-p} + \varepsilon_t - \beta_1 \varepsilon_{t-1} - \beta_2 \varepsilon_{t-2} - \dots - \beta_q \varepsilon_{t-q},$$

где α – коэффициенты авторегрессионной модели, β – коэффициенты модели скользящего среднего, ε – шумовая компонента [1].

В модели авторегрессии текущее значение процесса X_t представляется в виде линейной комбинации конечного числа предыдущих значений процесса и компоненты шума ε_t , при этом предполагается, что текущее значение шума не коррелировано с лагами X_t .

В моделях скользящего среднего текущее значение процесса X_t имеет представление в виде линейной комбинации текущего и прошлых значений ошибки обладающей свойствами белого шума [2].

В данной статье будет проведен анализ по многомерному временному ряду для прогнозирования загрязнения воздуха, исходные данные, для формирования которого составляют сведения о погоде и уровне загрязнения каждый час в течение пяти лет в Китае, Пекине. Будем использовать эти данные для прогнозирования загрязнения на один отсчет вперед.

Данные включают дату и время, загрязнение, называемое концентрацией PM2.5, и информацию о погоде, включая точку росы, температуру, давление, направление ветра, скорость ветра

и общее количество часов снега и дождя [3]. Полный список функций в необработанных данных выглядит следующим образом:

1. №: номер строки;
2. год: год данных в этой строке;
3. месяц: месяц данных в этой строке;
4. день: день данных в этой строке;
5. hour: час данных в этой строке;
6. PM2,5: концентрация PM2,5;
7. DEWP: точка росы;
8. TEMP: Температура;
9. ПРЕС: Давление;
10. cbwd: комбинированное направление ветра;
11. lws: накопленная скорость ветра;
12. ls: накопленные часы снега;
13. lg: накопленные часы дождя.

На рис.1 представлены графики по основным показателям доступного для наблюдения многомерного ряда.

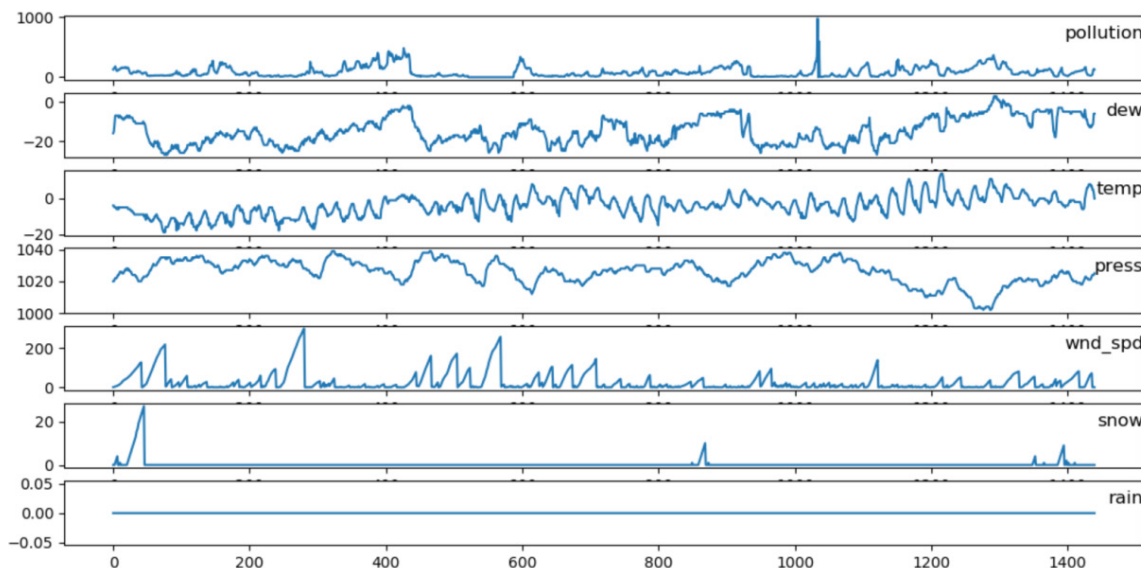


Рис. 1. График по исходным данным

Алгоритм построения модели ARIMA применительно к одномерному временному ряду температуры, он представлен на рис. 2-6. Анализируя график (рис.1) можно сделать вывод, что ряд нестационарен, это подтверждает и критерий Дики-Фуллера. Возьмем первую разность для приведения к стационарности ($d=1$). Ряд все еще не является стационарным, поэтому возьмем еще одну разность, тогда $d=2$ и построим ряд (рис.3), а также его STL-декомпозицию (рис.4), чтобы отследить сезонность и тренд.

Полученный ряд является стационарным. Построим его автокорреляционную (АКФ) и частичную автокорреляционную (ЧАКФ) функции. По ним сделаем приближения параметров модели p и q для алгоритма перебора моделей ARIMA (рис. 5). Параметр приближения для q подберем по виду АКФ, первые два значимо отличающийся от нуля лага дают нам основания взять $q = 2$. Параметр p подберем по виду ЧАКФ возьмем тоже равным двум. Таким образом, начальные приближения всех параметров модели ARIMA готовы. Сделаем их перебор и выберем лучший по информационному критерию Акаике. Для взятого ряда лучшей моделью является ARIMA(2,2,2). Данная часть анализа не учитывала в себе сезонные аспекты модели, а также размеры окна наблюдения для тестовой выборки. Но из примера наглядно видно, что мы можем получить стационарный ряд для прогноза по одномерному ряду путем взятия разностей d .

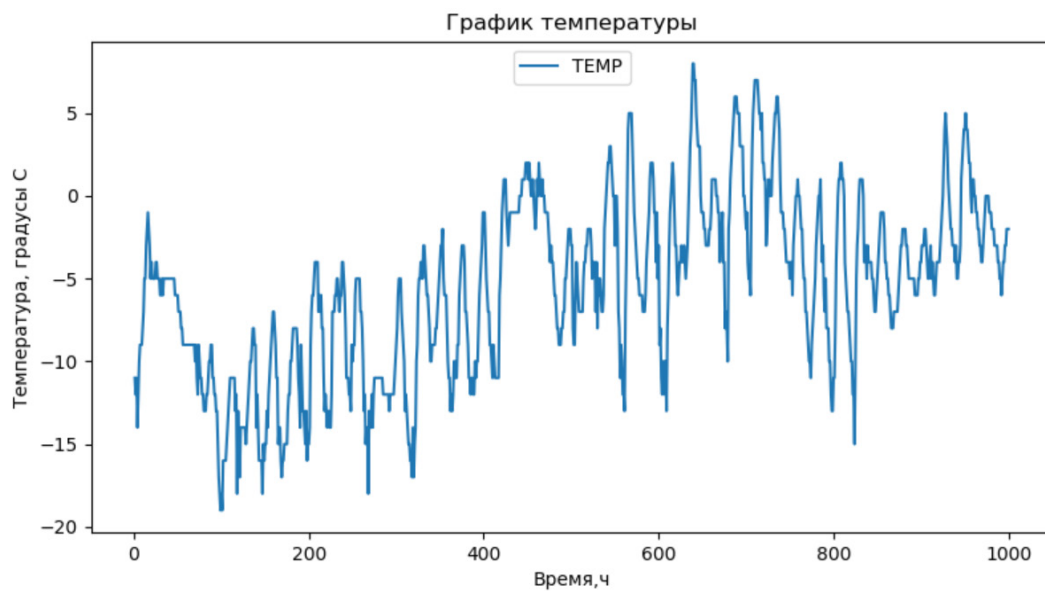


Рис. 2. Временной ряд температуры по исходным данным

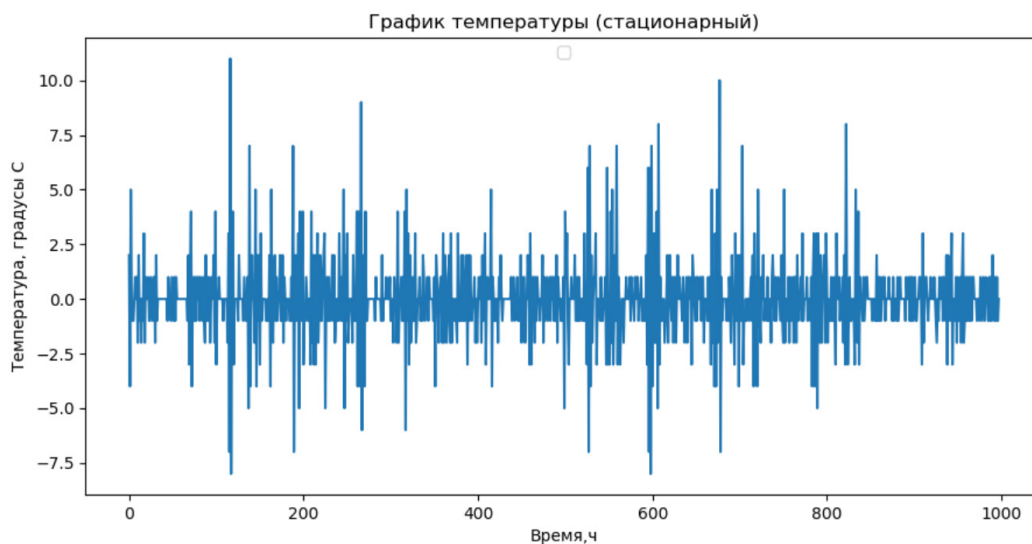


Рис. 3. Временной ряд температуры $d=2$

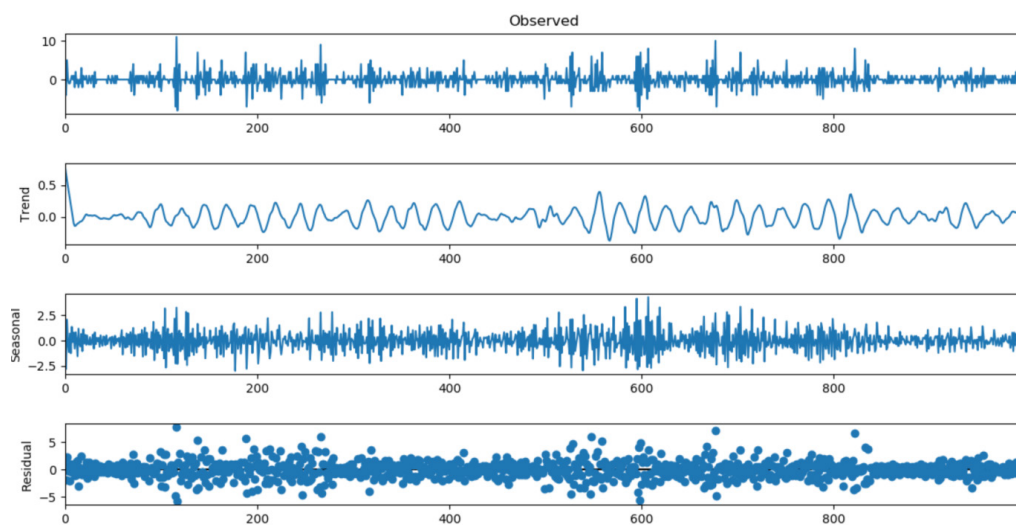


Рис. 4. STL-декомпозиция ряда температуры

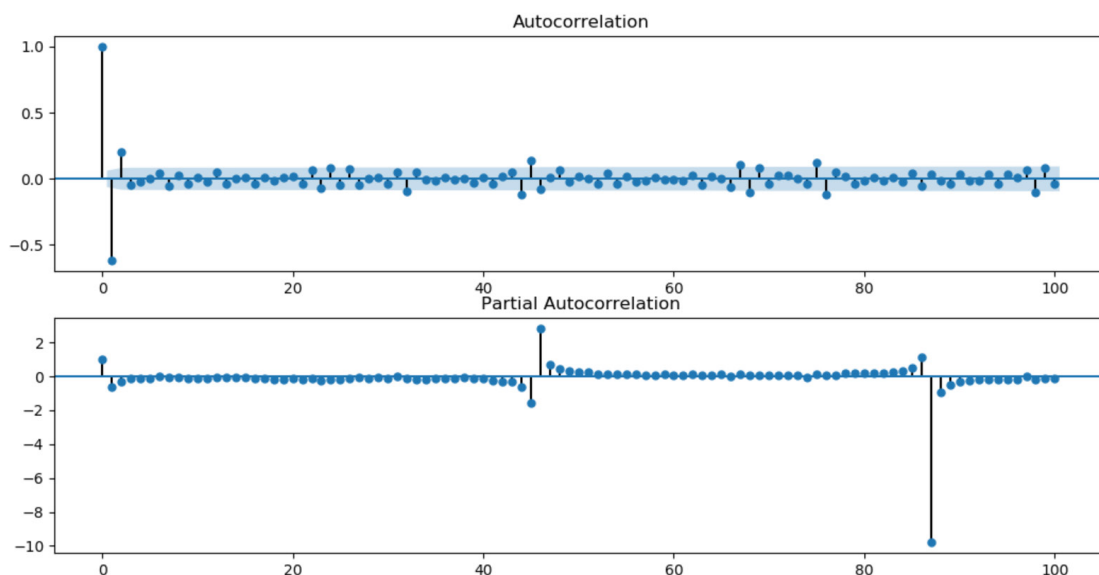


Рис. 5. Графики АКФ и ЧАКФ

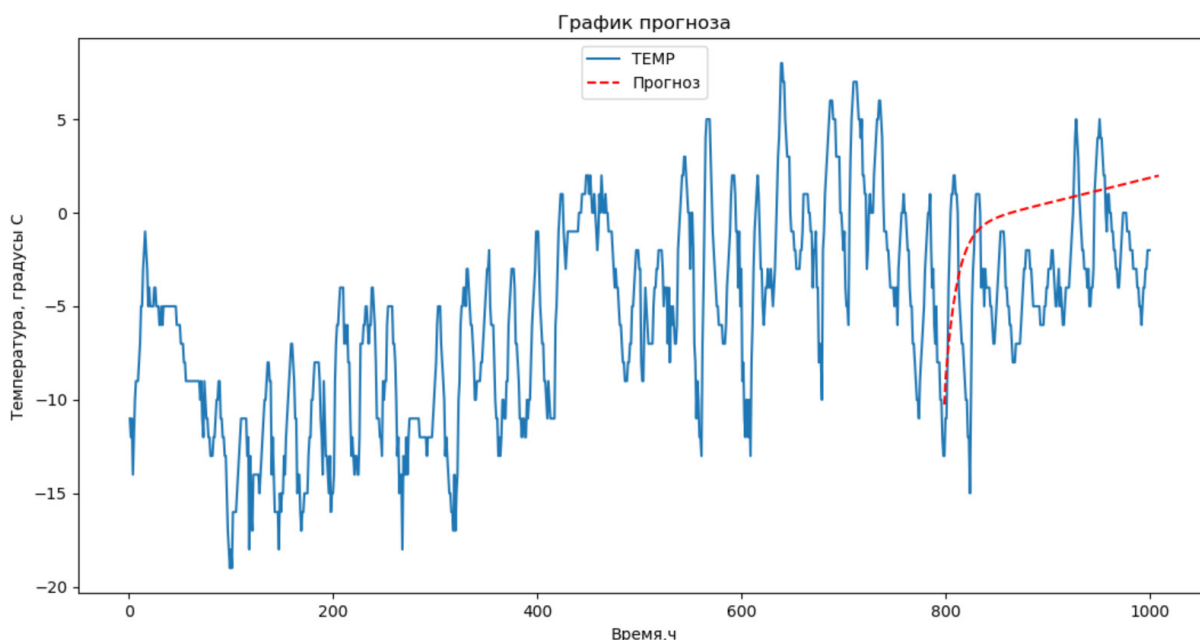


Рис. 6. График прогноза

По графику можно заметить, что выбранная модель строит недостаточно «хороший» прогноз, но она вполне подходит, чтобы приблизительно оценить динамику. Среднеквадратичное отклонение модели составило 4,644. Данная модель хорошо работает на одномерных рядах. Но в многомерных рядах зависимость следующего значения, по одному параметру, зависит не только от предыдущих значений предыдущего «своего» ряда, но и от других составляющих многомерного ряда, имеющих модели ARIMA по отличным от него параметрам. В такой постановке задача крайне непростая для решения, где потребуются использование решающего правила для получения общего прогноза.

На рис. 7 представлена схема обработки многомерного временного ряда для осуществления задачи прогнозирования. Предполагается, что для каждого ряда будет построен отдельный прогноз по оптимальным индивидуальным параметрам, затем на основе специальной свертки прогнозов отдельных рядов будет сформирован общий итоговый результат.

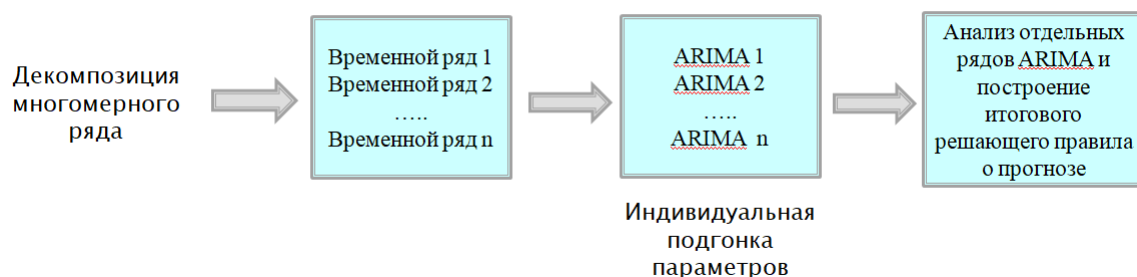


Рис. 7. Структурная схема для использования модели ARIMA в работе с многомерными рядами

Выводы и дальнейшие перспективы исследования.

На данном этапе исследования был выполнен обзор литературы по методам прогнозирования многомерных временных рядов. Поставлена задача исследования. Построена предварительная структура программного обеспечения для многомерного прогнозирования.

В дальнейшем требуется подобрать оптимальное решающее правило для получения общего прогноза на многомерном временном ряде. Проверить качество работы алгоритма ARIMA с различными окнами тестовой выборки и решающего правила на основе EMD. Создать программное обеспечение решающее поставленные задачи.

Библиографический список

1. Бокс Дж., Дженкинс Г. Анализ временных рядов. Прогноз и управление. Вып. 1, 2. М.: Мир, 1974.
2. StatSoft :электр. учеб. по статистике. URL: <http://statsoft.ru/home/textbook/modules/sttimser.html> (дата обращения: 10.05.2020).
3. Multivariate Time Series Forecasting with LSTMs in Keras. URL: <https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras> (дата обращения: 10.05.2020).

УДК 004.934.8*1

А. А. Гандрабура

студент кафедры компьютерных технологий и программной инженерии

М. Д. Поляк – старший преподаватель – научный руководитель

ВЫДЕЛЕНИЕ ГОВОРЯЩИХ В АУДИОПОТОКЕ

Для обозначения задачи выделения говорящих (дикторов) в аудиопотоке также используется термин «диаризация», поэтому дальше для краткости будет использоваться именно он.

Диаризация (или разделение дикторов) — процесс разделения входящего аудиопотока на однородные сегменты в соответствии с принадлежностью аудиопотока тому или иному говорящему [1]. Диаризация повышает качество текстов при автоматическом транскрибировании, а также может использоваться совместно с системой распознавания речи, значительно её улучшая. Используется для ответа на вопрос «Кто сейчас говорит?». Диаризация является сочетанием методов сегментации и кластеризации дикторов. Первый направлен на поиск точек смены диктора, второй — на группирование выделенных в речи диктора речевых сегментов. Соответственно, целью является выделение дикторов в аудиопотоке.

Решая задачу диаризации, как правило, используют следующие инструменты:

- скрытую марковскую модель (СММ);
- модель гауссовой смеси (МГС).

Гауссова смесь – это одна из форм оценки плотности распределения [2]. Т.е, в нашем случае – смесь признаков звучания, характерная для каждого говорящего. Пример гауссовой смеси показан на Рисунке 1 [3]. Во-первых, модели удобны для представления не только статистических характеристик голоса диктора, но и окружающей среды канала звукозаписи. Во-вторых, смеси гауссовых моделей представляют собой удобный способ представления и интерпретации акустических событий речевого сигнала [4].

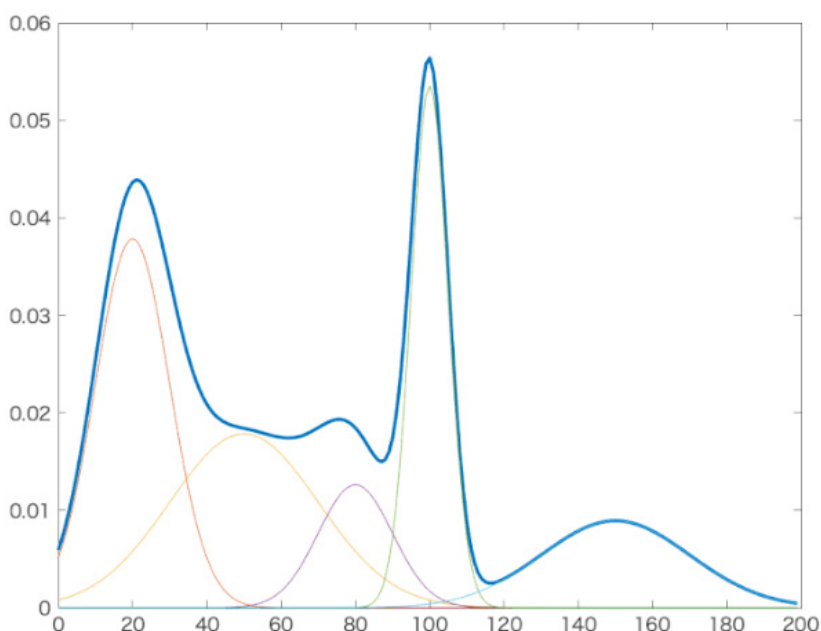


Рис. 1. Пример гауссовой смеси

Модель гауссовых смесей представляет собой взвешенную сумму M компонент и может быть записана выражением:

$$p(\bar{x} | \lambda) = \sum_{i=1}^M p_i b_i(\bar{x}), \quad (1)$$

где \bar{x} — это D -мерный вектор случайных величин; $b_i(\bar{x})$, $i = 1, \dots, M$ — функции плотности распределения составляющих модели и p_i , $i = 1, \dots, M$ — веса компонентов модели. Каждый компонент является D -мерной гауссовой функцией распределения вида:

$$b_i(\bar{x}) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(\bar{x} - \bar{\mu}_i)^T \Sigma_i^{-1}(\bar{x} - \bar{\mu}_i)\right\}, \quad (2)$$

где $\bar{\mu}_i$ — вектор математического ожидания и Σ_i — ковариационная матрица. Веса смеси удовлетворяют выражению: $\sum_{i=1}^M p_i = 1$.

Полностью модель гауссовой смеси определяется векторами математического ожидания, ковариационными матрицами и весами смесей для каждого компонента модели. Эти параметры все вместе записываются в виде:

$$\lambda = \{p_i, \mu_i, \Sigma_i\}, i = 1, \dots, M \quad (3)$$

В задаче распознавания говорящего каждый говорящий представляется моделью гауссовых смесей и ставится со своей моделью λ . Модель гауссовой смеси может иметь несколько различных форм в зависимости от вида ковариационной матрицы. Модель может иметь одну ковариационную матрицу для каждого компонента модели, как определено в формуле (3), одну ковариационную матрицу для всех гауссовых компонент в модели или одну ковариационную матрицу, используемую всеми говорящими во всех моделях [5].

СММ представляет собой статистическую марковскую модель, где моделируемая система считается марковским процессом с ненаблюдаемыми («скрытыми») состояниями [6]. На Рисунке 2 показан принцип работы СММ [7].

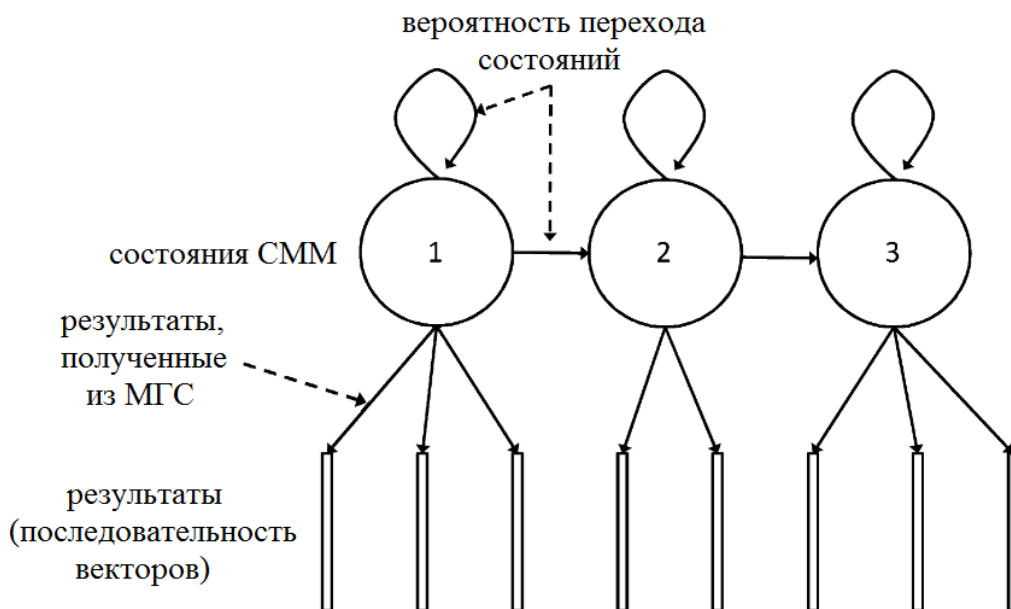


Рис. 2. Наглядная модель взаимодействия СММ

Скрытая марковская модель: оперирует гауссовыми смесями для уточнения модели говорящего; анализирует взаимодействие дикторов; строит модель взаимодействия.

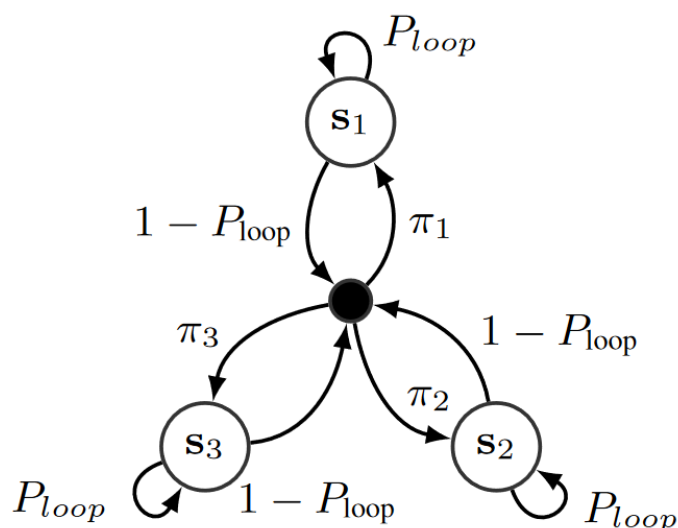


Рис. 3. Модель СММ для 3 говорящих с 1 состоянием на говорящего, с фиктивным начальным состоянием

СММ имеет взаимно-однозначное соответствие между состояниями СММ и говорящими. Модель СММ является эргодической (возможны переходы между всеми говорящими), где начальное число состояний выбирается так, чтобы оно учитывало наибольшее число говорящих, присутствие которых ожидается в разговоре.

Топология СММ и вероятности перехода моделируют длительность смены диктора (см. Рисунок 3): P_{loop} – настраиваемый параметр, который соответствует вероятности нахождения в том же состоянии (говорящего) [8]. Для функций с высокой частотой кадров, обычно устанавливается значение, близкое к единице, для естественного моделирования смены говорящих. Для каждого кадра мы покидаем текущее состояние с вероятностью $1 - P_{loop}$ и переходим в новое состояние диктора s с вероятностью π_s . Вероятности π также управляют выбором исходного состояния СММ. Вероятности π выводятся из входного разговора с использованием итеративных обновлений вариационной байесовской диаризации, и благодаря принципу автоматического определения релевантности [9] значения π_s для любых избыточных состояний говорящего имеют тенденцию сходиться к нулю. Это позволяет вывести количество выступающих во входном разговоре.

Каждое конкретное распределение говорящего (или состояние СММ) моделируется с использованием гауссовой смеси с ограниченным подпространством.

Большинство современных систем диаризации основаны на одном из двух подходов: «снизу вверх» или «сверху вниз» [10]. Оба подхода «снизу вверх» и «сверху вниз», как правило, основаны на скрытых марковских моделях, где каждое состояние представляет собой модель гауссовой смеси и соответствует говорящему.

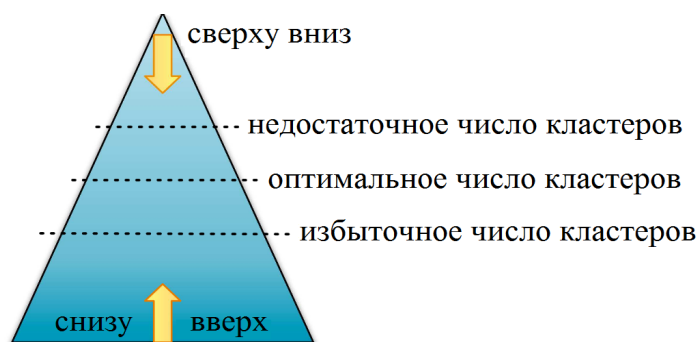


Рис. 4. Схема восходящего и нисходящего подходов

При восходящем подходе система инициализируется с малым количеством кластеров (обычно с одним), в то время как при подходе «снизу вверх» – с большим количеством кластеров (обычно больше, чем число ожидаемых дикторов). В обоих случаях цель – итеративно сойтись к оптимальному количеству кластеров. Если конечное число оказывается больше оптимального, то система считается избыточной. Если оно меньше, то система считается недостаточной. Переходы между состояниями соответствуют смене диктора.

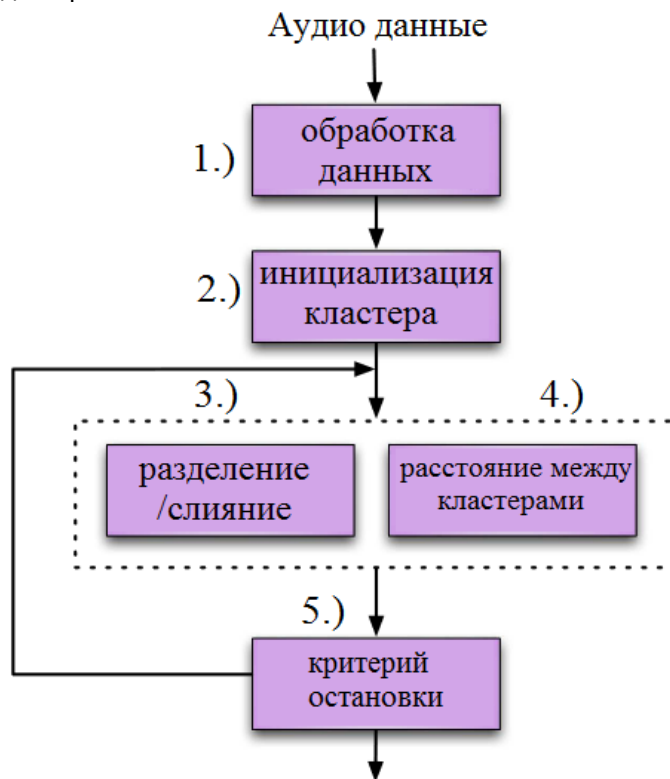


Рис. 5. Блок-схема типовых модулей, которые составляют большинство систем диаризации

Стадия обработки данных на Рисунке 5 (п. 1) имеет тенденцию быть несколько специфичной для домена данных. Т.е. первичная обработка зависит от начального набора данных. Для данных, в случае записи встречи, предварительная обработка обычно включает в себя шумоподавление, многоканальное акустическое формирование луча (метод, позволяющий определить источники звука, основанный на особенностях отраженного сигнала), параметризацию речевых данных в акустические характеристики и обнаружение речевых сегментов с помощью алгоритма обнаружения речевой активности. Т.е., если говорить упрощённо, на этой стадии происходит фильтрация шумов и нормализация уровня громкости.

Инициализация кластера на Рисунке 5 (п. 2) зависит от подхода к диаризации, то есть от выбора начального набора кластеров при кластеризации «снизу вверх» или одного сегмента при кластеризации «сверху вниз».

На Рисунке 5 (п. 3 и п.4) показаны механизм разделения/слияния и расстояние между кластерами. Они используются для итеративного объединения кластеров или для введения новых. Алгоритмы очистки данных могут быть использованы для того, чтобы сделать кластеры более различимыми. Т.е., в этом пункте происходит выделение ключевых характеристик, позволяющих определить говорящего.

Наконец, как показано в пункте 5 на Рисунке 5, критерии остановки используются для определения того, когда было достигнуто оптимальное число кластеров. В этом пункте происходит определение говорящего.

Оптимальное число кластеров – обособление одного набора характеристик от другого. Расстояние между кластерами – разница характеристик.

Таким образом, в этой статье рассмотрены подходы для решения задачи диаризации: восходящий и нисходящий подходы; модель гауссовой смеси; скрытая марковская модель.

Библиографический список

1. Диаризация. URL: <https://ru.wikipedia.org/wiki/Диаризация> (дата обращения: 10.10.2019).
2. adminCraft. Гауссова смесь распределений (GMM). URL: <https://craftappmobile.com/1073-2/> (дата обращения: 02.03.2020).
3. Mixture model. URL: https://en.wikipedia.org/wiki/Mixture_model#Gaussian_mixture_model (дата обращения: 20.05.2020).
4. *Gonina E.* Fast Speaker Diarization Using a SpecializationFramework for Gaussian Mixture Model Training. URL: <https://digitalassets.lib.berkeley.edu/techreports/ucb/text/EECS-2011-128.pdf> (дата обращения: 11.03.2020).
5. *Садыхов Р. Х., Ракуш В. В.* Модели гауссовых смесей для верификации диктора по произвольной речи. URL: <https://cyberleninka.ru/article/n/modeli-gaussovyh-smesey-dlya-verifikatsii-diktora-po-proizvolnoy-rechi/viewer> (дата обращения: 11.03.2020).
6. The subspace Gaussian mixture model — A structured model for speech recognition // *Computer Speech and Language*. 2011. № 25. P. 405.
7. Hidden_Markov_model. URL: https://en.wikipedia.org/wiki/Hidden_Markov_model (дата обращения: 30.11.2020).
8. *Diez M., Burget L., Wang S. et al.* Bayesian HMM based x-vector clustering for Speaker Diarization. URL: https://www.isca-speech.org/archive/Interspeech_2019/pdfs/2813.pdf (дата обращения: 05.04.2020).
9. *Wipf D., Nagarajan S.* A New View of Automatic Relevance Determination. URL: <http://papers.nips.cc/paper/3372-a-new-view-of-automatic-relevance-determination.pdf> (дата обращения: 07.04.2020).
10. *Anguera X., Bozonnet S., Evans N. et al.* Speaker diarization: A review of recent research. URL: https://hal.archives-ouvertes.fr/file/index/docid/733397/filename/IEEE_Transaction2010_Speaker_Diarization_overview_1_.pdf (дата обращения: 15.11.2019).

УДК 004.42

Е. М. Глушкова

студент кафедры компьютерных технологий и программной инженерии

М. Д. Поляк – старший преподаватель – научный руководитель

ПРИМЕНЕНИЕ ЛАТЕНТНОГО РАЗМЕЩЕНИЯ ДИРИХЛЕ ДЛЯ ОПРЕДЕЛЕНИЯ МЕРЫ СХОЖЕСТИ ТЕКСТОВ

Для реализации алгоритмов сравнения текстов используются разные методы, одним из которых является Латентное размещение Дирихле (LDA, от англ. Latent Dirichlet allocation). [1-3]

Для того, чтобы применить метод к коллекции текстов, необходимо преобразовать ее в терм-документную матрицу (term-document matrix), имеющую размеры $N \times W$, где N – количество документов в коллекции, а W – количество уникальных слов, содержащихся в текстах. Для данной матрицы и ранее заданного числа тем T (в этом случае, слово тема определяет некую абстрактную сущность, и редко совпадает с темами, существующими в коллекции текстов) создается два распределения: $N \times T$ – распределение тем по текстам, и $T \times W$ – распределение слов по темам. [4]

Как видим, результаты реализации метода LDA зависят от коллекции текстов и от числа тем, поэтому имея коллекцию из выбранных нами текстов, остается определиться с количеством тем.

Для выбора оптимального числа тем была использована коллекция, состоящая из двух текстов, которые являются статьями на одну тему из разных новостных источников («Israeli election: Netanyahu and Gantz both claim victory» [5] и «Benjamin Netanyahu's re-election plans in limbo as Israeli election too close to call» [6]), позволяло отследить соотношение количества тем и коэффициента схожести.

Далее, для исследования была использована подборка из количества тем от 10 до 100 с шагом 10. Для каждого количества тем было создано такое количество моделей LDA, чтобы можно было примерно определить среднее значение коэффициента схожести двух текстов.

Полученные данные, учитывая при создании модели LDA параметр (random_state), отвечающий за создание отличающихся моделей при каждой реализации алгоритма, показаны на рис. №1:

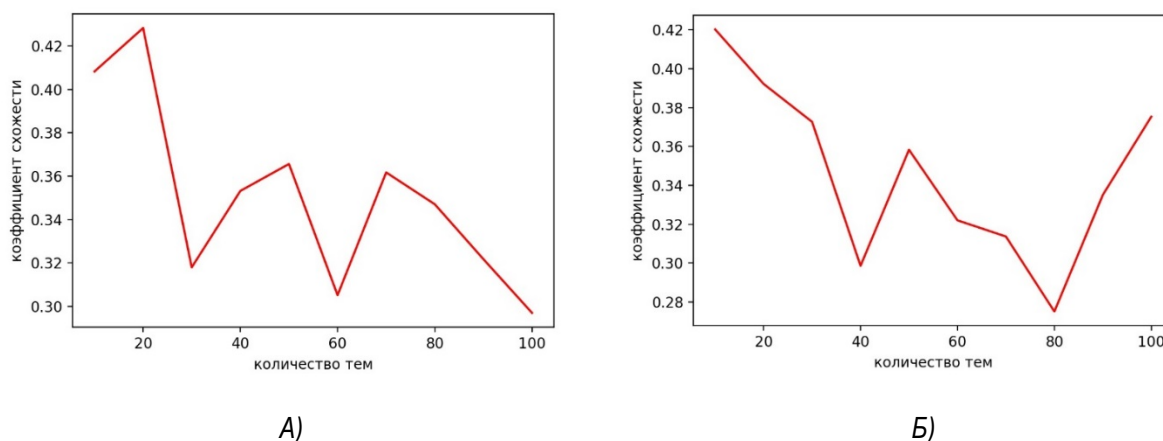


Рис. 1. Результаты реализации метрики lda при двух разных значениях параметра random_state: А – результат первой реализации, Б – результат второй реализации.

Из-за параметра, отвечающего за обновление модели после каждой реализации, получение данных об оптимальном количестве тем потребовало дальнейших расчетов.

Для изучения изменения числа тем были использованы две коллекции, состоящие из двух текстов каждая. В первой – были тексты со схожей тематикой, а во второй – с разной. В качестве текста на отличающуюся тему была взята статья «Coca-Cola reveals how much plastic it uses» [7]. Это необходимо для того, чтобы знать, что данные не пересекались, иначе нельзя будет однозначно сказать являются ли представленные тексты сходными, или же имеют разную тематическую направленность.

Для увеличения точности, для каждого количества тем число созданных моделей LDA было увеличено, и найдено среднее квадратическое отклонение коэффициента схожести, чтобы проверить разброс данных. Результат вычислений отображен на рис. №2:

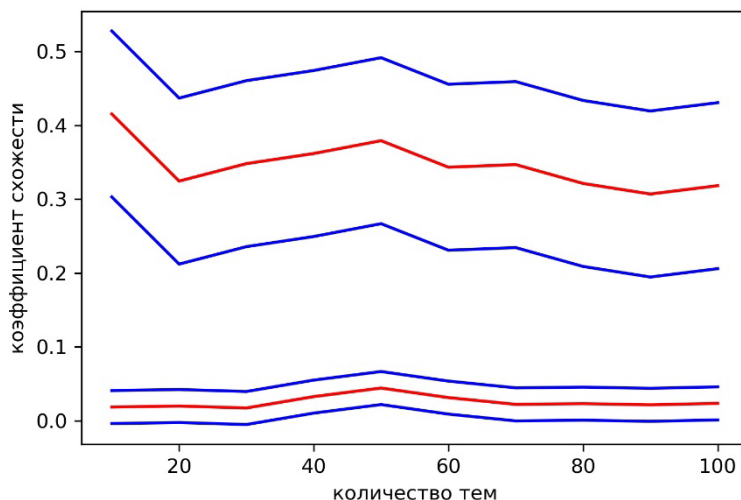


Рисунок 2. Результаты вычислений с использованием метрики *lda* для пар текстов с схожими и разными данными

Можно заметить, что 95 %-доверительные интервалы для коэффициентов схожести одинаковых текстов и текстов на разные темы не пересекаются, что позволяет однозначно определить отношения между текстами. Также из графика видно, что увеличение числа тем оказывает на среднее значение незначительное влияние, из чего можно сделать вывод о том, что повышение количества тем влияет только на точность, но не на определение схожести текстов. Стоит учитывать, что увеличение числа тем, повышает количество созданных моделей LDA, и потому время работы программы, реализующей алгоритм сравнения, растет.

Далее был реализован полный алгоритм, содержащий метод LDA, и, использующий коллекцию из трех текстов, упомянутых выше. Ниже приведены полученные результаты:

Таблица 1

Результаты сравнения текстов с использованием метрики *lda*

	Текст 1	Текст 2	Текст 3
Текст 1	1	0.985	0.194
Текст 2	0.985	1	0.023
Текст 3	0.192	0.024	1

Единичные коэффициенты в таблице указывают на тождество документа самому себе.

Далее, для увеличения точности результатов алгоритма были использованы библиотеки «стоп-слов», которыми называют слова, не несущие смысловой (и, в том числе, тематической) нагрузки, например, 'somewhere', 'and', 'mine', 'too', 'also' и подобные. После реализации сравнения с использованием данных библиотек были получены данные:

Таблица 2

Результаты сравнения текстов с использованием метрики *lda*, уточненные с помощью библиотек «стоп-слов»

	Текст 1	Текст 2	Текст 3
Текст 1	1	0.004	0
Текст 2	0.004	1	0
Текст 3	0	0	1

Можно заметить, что после удаления «стоп-слов» результаты радикально уменьшились, это произошло из-за специфики метода LDA.

Также повышению качества результатов способствовало применение стемминга (от англ. *stem* – стебель) – алгоритма, позволяющего нормализовать слова, т.е. выделять их основу. Такая обработка текстов позволяет определять однокоренные, т.е. близкие по смыслу (и, следовательно, несущие равную тематическую нагрузку), слова как сходные – при том, что в написании они могут существенно различаться.

Работа проводилась со стеммером Портера, который не использует баз основ слов, а лишь отсекает окончания и суффиксы, применяя ряд правил, основанных на особенностях языка. Из-за этого он работает быстрее остальных, но с определенной погрешностью. Ниже приведены результаты сравнения с учетом нормализации данным стеммером [8]:

Таблица 3

Результаты сравнения текстов с использованием *lda*, уточненные с помощью библиотек «стоп-слов» и нормализации

	Текст 1	Текст 2	Текст 3
Текст 1	1	0.971	0
Текст 2	0.971	1	0.236
Текст 3	0	0.235	1

Как видим, все коэффициенты в таблице сильно выросли – за счёт того, что слова, отличающиеся друг от друга лишь окончаниями и суффиксами, были приведены к одной основе и посчитаны как сходные.

Таким образом, в ходе проведенного исследования была доказано, что, используя метод LDA, можно однозначно отличить схожие тексты от различных, показана его совместимость с другими методами, изучено влияние параметра «количество тем» на получаемые данные, и установлено, что увеличение числа задействованных алгоритмов позволяет повысить точность расчетов.

Библиографический список

1. *Breitinger C., Gipp B., Langer S.* Research-paper recommender systems: a literature survey // *International Journal on Digital Libraries*. 2015. № 17 (4). P. 305–338.
2. Семьдесят вторая международная студенческая научная конференция ГУАП: сб. докл.: в 4 ч. Ч. 2: Техн. науки. СПб.: ГУАП, 2019. С. 203–205.
3. *Bulletin of the UNESCO department «Distance education in engineering» of the SUAI: Collection of the papers.* St. Petersburg, Issue 5. SPb.: SUAI, 2020. P. 61–65.
4. *Blei David M., Ng Andrew Y., Jordan Michael I.* Latent Dirichlet allocation // *Journal of Machine Learning Research*. 2003. № 3. P. 993–1022.
5. BBC News. Israeli election: Netanyahu and Gantz both claim victory. URL: <https://www.bbc.com/news/world-middle-east-47874948> (дата обращения: 19.03.2020).
6. CNN World. Benjamin Netanyahu's re-election plans in limbo as Israeli election too close to call. URL: <https://edition.cnn.com/2019/04/09/middleeast/benjamin-netanyahu-israel-election-exit-poll-intl/index.html> (дата обращения: 19.03.2020).
7. BBC News. Coca-Cola reveals how much plastic it uses. URL: <https://www.bbc.com/news/newsbeat-47569233> (дата обращения: 19.03.2020).
8. *Porter M.* An algorithm for suffix stripping // *Program*. 1980. № 14 (3). P.130–137.

УДК 519.61:511-33

М. П. Демчук

студент кафедры компьютерных технологий и программной инженерии

Е. В. Копкин – доктор технических наук, доцент – научный руководитель

РАЗРАБОТКА НОВОГО ФОРМАТА СЖАТИЯ ИЗОБРАЖЕНИЙ НА ОСНОВЕ МАТРИЦ МЕРСЕННА-УОЛША

Актуальность

Потребность сохранять и передавать изображения с каждым годом возрастает. Даже при быстром росте ресурсов вычислительной техники, задача сжатия и передачи фото и видео данных по-прежнему остается открытой. Достижения в области видео технологий и соответствующий рост на мультимедийном рынке, включая телевидение с высоким разрешением, создают спрос на новые и быстродействующие алгоритмы сжатия данных, в том числе изображений.

Сжатие изображений широко используется в таких областях как спутниковые съёмки, архивация данных и медицинская визуализация. Известно, что процедура сжатия – это представление информации в более эффективном виде, влекущее за собой уменьшение объема данных. Существуют два основных подхода к сжатию: без потерь и с потерями. Поскольку первый подход имеет предел сжатия исходного изображения, то будет рассматриваться только второй, как наиболее перспективный, позволяющий получить высокие коэффициенты сжатия, по сравнению с первым. При этом, используя сжатие с потерями возможно достичь компромисса между предъявляемыми требованиями к качеству изображения и степени сжатия, что позволяет разработать алгоритм сжатия для конкретной прикладной задачи.

Основная часть.

Для нахождения матриц Мерсенна-Уолша четвёртого порядка необходимо выполнить ряд последовательных шагов:

Шаг 1. Построить матрицу Адамара, порядок которой равен порядку матрицы Мерсенна-Уолша + 1. Например, для $n = 7$, имеем

$$H_{[n+1]} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}.$$

Шаг 2. Упорядочить столбцы матрицы Адамара, построенной на шаге 1, по частоте (по количеству смен знаков их элементов).

$$H_{[n+1]} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}.$$

Шаг 3. Для определения функции Мерсенна–Уолша второго порядка [1] необходимо в упорядоченной матрице Адамара–Уолша заменить -1 на -b, где b вычисляется по формуле

$$b = \frac{q - \sqrt{4q}}{q - 4}; q = n + 1.$$

Затем полученную матрицу нужно инвертировать по знаку так, чтобы после удаления каймы (первой строки и первого столбца) число положительных элементов в каждом столбце на единицу превышало число отрицательных. Для рассматриваемого примера $b = 0,5858$;

$$\mathbf{HU}_{[n]} = \begin{bmatrix} 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}; \mathbf{MU}_{[n]} = \begin{bmatrix} -b & -b & -b & 1 & 1 & 1 & 1 \\ -b & 1 & 1 & 1 & 1 & -b & -b \\ -b & 1 & 1 & -b & -b & 1 & 1 \\ 1 & 1 & -b & -b & 1 & 1 & -b \\ 1 & 1 & -b & 1 & -b & -b & 1 \\ 1 & -b & 1 & 1 & -b & 1 & -b \\ 1 & -b & 1 & -b & 1 & -b & 1 \end{bmatrix}.$$

Шаг 4. Умножить каждый элемент матрицы Мерсенна–Уолша (ММУ) $\mathbf{MU}_{[n]}$ на соответствующий элемент матрицы шахматного порядка $\mathbf{E}_{[n]}$, состоящей из 1 и -1. В результате получим преобразованную ММУ $\mathbf{MU}_{[n]}^*$. Для рассматриваемого примера

$$\mathbf{E}_{[n]} = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix}; \mathbf{MU}_{[n]}^* = \begin{bmatrix} -b & b & -b & -1 & 1 & -1 & 1 \\ b & 1 & -1 & 1 & -1 & -b & b \\ -b & -1 & 1 & b & -b & -1 & 1 \\ -1 & 1 & b & -b & -1 & 1 & b \\ 1 & -1 & -b & -1 & -b & b & 1 \\ -1 & -b & -1 & 1 & b & 1 & b \\ 1 & b & 1 & b & 1 & b & 1 \end{bmatrix}.$$

В ходе такого решетчатого преобразования низкочастотные и высокочастотные ортогональные функции меняются местами (См. Рис.1 и Рис. 2) и у новой матрицы появляются ещё два уровня [2].



Рис. 1. 2-х уровневая Матрица Мерсенна–Уолша с низкими частотами слева сверху (черный цвет соответствует -b, белый – единице)



Рис. 2. 4-х уровневая Матрица Мерсенна-Уолша с низкими частотами справа снизу

После построения ММУ алгоритм делит изображения на блоки матриц, равных размеру матрицы и производит поблочное умножение ММУ на матрицу изображения и на транспонированную ММУ и полученную матрицу мы умножаем на маску.

Результат.

В результате алгоритма массив изображения сжимается с 10 миллионов чисел (Рис. 3) до 1 миллиона при сжатии изображения в 90 % (Рис. 5) и возвращает изображение в примерно исходный размер, в отличие от JPEG.



Рис. 3. Начальное изображение



Рис. 4; 5. Восстановленное изображение с 50 % сжатием методом ММУ и JPEG

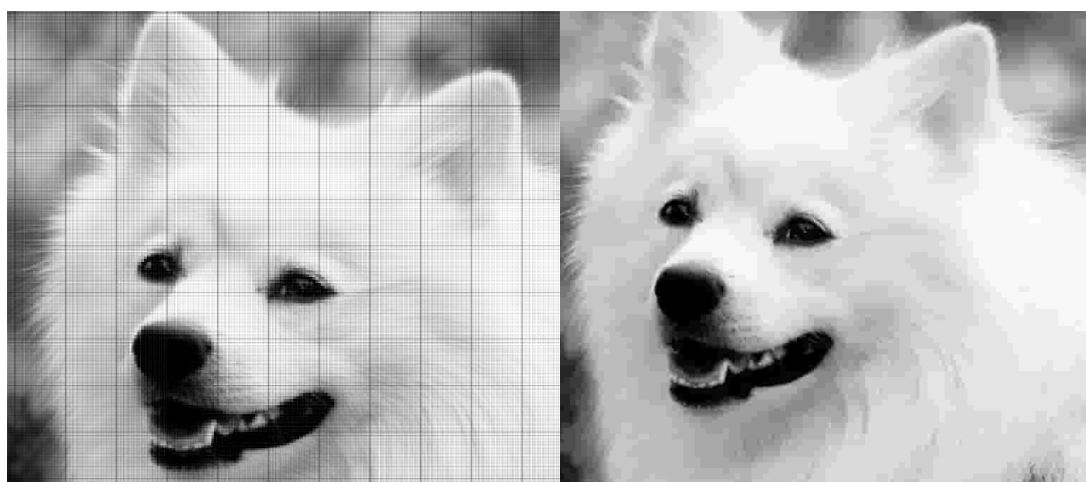


Рис. 6; 7. Восстановленное изображение с 90 % сжатием методом ММУ и JPEG

Заключение

Используя алгоритм сжатия изображений с помощью ММУ, можно прийти к выводу, что при определённом коэффициенте квантования изображения и размере матриц, изображение сохраняет хорошее качество относительно изначального, но имеет некоторые помехи, которые нужно устранить постобработкой.

Библиографический список

1. Вычисление матриц Мерсенна-Уолша. URL: <http://mathscinet.ru/files/2014MersenneWalsh.pdf> (дата обращения: 15.03.2020).
2. Негациклические матрицы и фильтры Мерсенна. URL: http://livelab.spb.ru/pdf/files/2016VKIT_N11.pdf (дата обращения: 15.03.2020).

УДК 004.83

А. А. Кириллова

студент кафедры компьютерных технологий и программной инженерии

М. Д. Поляк – старший преподаватель – научный руководитель

МАЙНИНГ АРГУМЕНТАЦИИ

Аргументация состоит из набора предположений или предпосылок (тезисов), методов рассуждения или логики (аргументов) и заключения (вывода). Входными данными для моделей аргументации служат необработанные или предварительно аннотированные данные в виде монологов (академические эссе, юридические тексты), диалогов и полилогов (контент социальных сетей: форумы и так далее).

Майнинг аргументации может применяться для проверки достоверности информации (что особенно актуально в связи с экспоненциальным ростом потенциально непреднамеренно или с определенными целями искаженных данных на сетевых ресурсах), интеллектуального анализа аргументации, социально-психологических исследований механизма формирования мнений. Данная область включает решение таких задач как анализ дискуссий в интернете, помощь в принятии решений, анализ тональности текста, написание аргументационных текстов, автоматизированная оценка эссе, частично создание диалоговых систем.

System	Task
Eckle-Kohler et al. [2015]	Claim/premise mining
Lippi and Torroni [2015]	Claim mining
Rinott et al. [2015]	Evidence mining
Sardianos et al. [2015]	Boundary detection
Boltuzic and Snajder [2014]	Structure prediction for arguments
Goudas et al. [2014]	Claim/premise mining
Levy et al. [2014]	Claim mining
Stab and Gurevych [2014b]	Claim/premise mining and structure prediction
Cabrio and Villata [2012a]	Textual entailment for arguments
Rooney et al. [2012]	Claim/premise mining
Biran and Rambow [2011]	Evidence mining and structure prediction
Mochales Palau and Moens [2011]	Claim/premise mining and structure prediction
Park et al. [2015]	Premise verifiability categorization
Budzynska et al. [2014]	Dialogical argument mining
Park and Cardie [2014]	Premise verifiability categorization
Houngbo and Mercer [2014]	Rhetorical category sentence classification
Ong et al. [2014]	Automated essay scoring
Peldszus [2014]	Argument graph structure classification
Rosenthal and McKeown [2012]	Opinionated claim mining
Villalba and Saint-Dizier [2012]	Argumentative opinion analysis
Feng and Hirst [2011]	Argument scheme classification

Рис. 1. Системы, решающие различные подзадачи АМ

Майнинг аргументации включает в себя обнаружение тезисов, аргументов, контраргументов, построение структуры аргументации, оценка качества аргументации. Существует достаточно много систем, реализующие те или иные подзадачи в данной области.

Для классификации предложений как части аргументации в работе Моэнса 2007 года используются наивные байесовские классификаторы и модели максимальной энтропии, обученные на лексических и синтаксических признаках, а также статистических данных в виде длины предложения и позиции в тексте. К лексическим признакам относятся n-граммы, части речи, аргументационные маркеры, а к синтаксическим – пунктуация, глубина построенного дерева аргументации и так далее. Обучение в данной работе осуществлялось на данных решений Европейского суда по правам человека и AraucariaDB.

Другим подходом к определению структур аргументации является обучение дерева решений. В работе «Argument extraction for supporting public policy formulation» Florou et al. в качестве признаков использовался вручную составленный список маркеров частей аргументации в комбинации с временными, эмоциональными, лексическими и морфосинтаксическим признаками.

Набор возможных признаков для обучения моделей также включает части речи, их позиция в предложении (n-грамме или абзаце, хотя большинство систем работают с предложениями) и длину, списки ключевых слов, а также сущности, упомянутые в предшествующих предложениях. В работе Goudas et al. эти признаки используются для обучения логической регрессионной модели

Система Stab and Gurevych реализует метод опорных векторов, который базируется на построении разделяющей гиперплоскости, расстояние до которой является показателем средней ошибки. Модель обучена на словах-подсказках из Penn Discourse Treebank в дополнение к структурным, лексическим, синтаксическим признакам и показывает хорошие результаты.

The Excitement Open Platform (EOP) – платформа с открытым исходным кодом, которая, учитывая два текстовых фрагмента, один из которых называется текстом, а другой – гипотезой, решает задачу выводимости гипотезы из текста. На вход эта модель принимает пары «текст-гипотеза», а на выходе дает выводимость гипотезы из текста: «Entailment» или «NonEntailment». Модульная архитектура EOP дает возможность легко заменять или расширять компоненты, например, работать с разными языками: английским, немецким и итальянским.

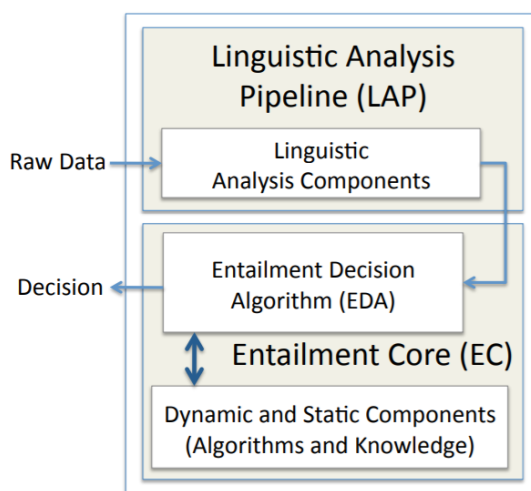


Рис. 2. Архитектура EOP

Linguistic Analysis Pipeline (LAP) состоит из компонентов для обработки естественного языка, основанных на фреймворке Apache UIMA, и отвечает за лингвистический анализ: парсинг, распознавание именованных сущностей, распознавание фраз, определение частей речи и другие задачи. EC принимает решение на основе уже обработанных данных, полученных от LAP, а также на основе статической и динамической информации – синтаксических и лексических знаний. Разделение задач между LAP и Entailment Core (EC) гарантирует, что эти модули могут работать независимо друг от друга.

EC определяет выводимость гипотезы из текста с помощью алгоритмов, реализующих три разных подхода: edit-distance, трансформации между текстом и гипотезой и классификация с помощью признаков, извлекаемых из текста и гипотезы.

Edit-distance алгоритм рассматривает проблему выводимости как сопоставления текста и гипотезы. Сопоставление выполняется в виде последовательности операций редактирования – вставка, удаление, замена – и каждая операция имеет свою цену. Суть подхода в предположении, что вероятность выводимости гипотезы из текста коррелирует с расстоянием между ними. В процессе обучения данный метод создает модель расстояний для данных, включая критическую границу, разделяющую позитивные и негативные примеры в обучающей выборке. Во время тестирования пары с расстоянием ниже границы классифицируются как «Entailment», выше – «NonEntailment».

Основанный на трансформациях алгоритм применяет серию изменений к тексту с целью создания текста, идентичного гипотезе. Учитывая, что каждая трансформация сохраняет изначальный смысл текста, можно сделать вывод о выводимости гипотезы. Базы знаний играют важную роль в распознавании взаимозаменяемых слов и фраз, например, помогая определить, что «born in» и «citizen of» в большинстве случаев имеют одно и то же значение, а «Hawaii» и «America» – связанные друг с другом локации. EOP использует систему Бар-Хейма, вероятностную модель для вычисления уверенности, а также алгоритм, вычисляющий оптимальную последовательность трансформаций.

Алгоритм классификации комбинирует результаты работы нескольких функций оценки, которые извлекают из текста различную лингвистическую информацию – синтаксические и семантические зависимости, именованные сущности, модель «bag-of-words».

Таблица 1

Тестирование EOP

Номер	Позитивные примеры		
	EditDistance	MaxEntClassification	BIUTEE
1	+ (0)	+ (0.96)	- (0.74)
2	- (0.21)	+ (0.88)	- (0.64)
3	+ (0.2)	+ (0.98)	+ (0.7)
	Негативные примеры		
4	+ (0.25)	- (0.66)	+ (0.75)
5	- (0)	- (0.96)	- (0.62)
6	- (0.32)	- (0.98)	- (0.64)

Библиографический список

1. *Manfred Stede, Jodi Schneider, Graeme Hirst. Argumentation Mining (Synthesis Lectures on Human Language Technologies). Morgan & Claypool Publishers, 2018.*
2. *The evolution of argumentation mining: From models to social media and emerging tools / Anastasios Lytos, Thomas Lagkas, Panagiotis Sarigiannidis, Kalina Bontcheva // Information Processing & Management. 2019. Vol. 56, № 6. URL: https://www.researchgate.net/publication/334195420_The_evolution_of_argumentation_mining_From_models_to_social_media_and_emerging_tools (дата обращения: 11.05.2020).*
3. *Теория аргументации. URL: <http://hi-edu.ru/e-books/xbook912/01/part-009.htm> (дата обращения 11.05.2020).*
4. *Raquel Mochales Palau, Marie-Francine Moens. Argumentation mining: the detection, classification and structure of arguments in text // Computer Science. 2009. URL: https://www.researchgate.net/publication/221539331_Argumentation_mining_The_detection_classification_and_structure_of_arguments_in_text (дата обращения: 11.05.2020).*
5. *Eirini Florou, of Linguistics, Stasinou Konstantopoulos, Antonis Kukurikos, Pythagoras Karampiperis. Argument extraction for supporting public policy formulation. URL: https://www.researchgate.net/publication/263995552_Argument_extraction_for_supporting_public_policy_formulation (дата обращения: 11.05.2020).*
6. *Theodosios Goudas, Christos Louizos, Georgios Petasis, and Vangelis Karkaletsis. Argument extraction from news, blogs, and social media. URL: https://link.springer.com/chapter/10.1007/978-3-319-07064-3_23 (дата обращения: 11.05.2020).*
7. *Excitement Open Platform. URL: <http://hlfbk.github.io/Excitement-Open-Platform/> (дата обращения: 11.05.2020).*
8. *Bernardo Magnini, Roberto Zanolini, Ido Dagan, Kathrin Eichler et al. The Excitement Open Platform for Textual Inferences. URL: <https://www.aclweb.org/anthology/P14-5008.pdf> (дата обращения: 11.05.2020).*

УДК 519.246.8

В. В. Малеев

студент кафедры компьютерных технологий и программной инженерии

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

АЛГОРИТМЫ РАСПОЗНАВАНИЯ СОСТОЯНИЙ ДИНАМИЧЕСКОЙ СИСТЕМЫ ПО ВРЕМЕННЫМ РЯДАМ

Целью исследования в статье является выявление возможности применения метода декомпозиции временного ряда для прогнозирования и оценки качества прогнозирования на реальных временных рядах.

Для иллюстрации применимости алгоритма EMD в задаче прогнозирования взяты данные изменения цен на нефть (\$/баррель) за период с 13 мая 2020 года (от 20:00) по 19 мая 2020 года (до 8:00) с динамикой в 4 часа (рис. 1) [3].

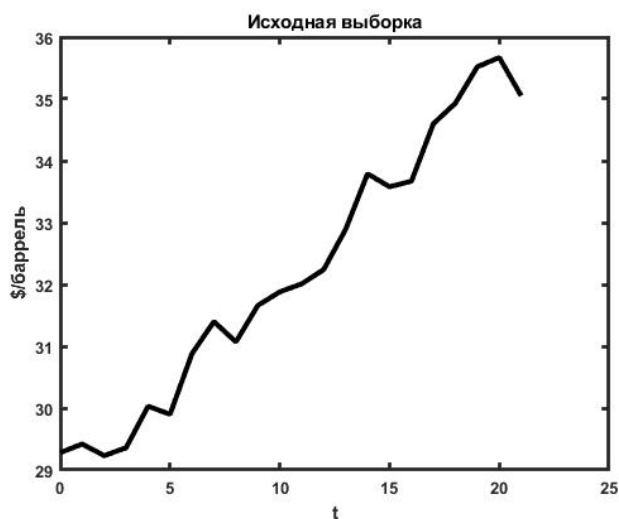


Рис. 1. График зависимости цен на нефть от времени

Поставим задачу разложения имеющегося временного ряда по методу EMD, построения прогноза по совокупности полученных компонентов EMD-разложения, оценивания поведения временного ряда на следующие 4 часа (прогноз цен на нефть за 19 мая в 12:00) и качества правила прогнозирования (сравнение результата с реальными данными как контрольной выборки).

Суть метода EMD заключается в построении итерационной процедуры, результат которой есть совокупность эмпирических мод (внутренних колебаний, англ. *intrinsic mode functions*, IMF) [1]. В отличие от гармонического анализа, где модель (дискретного или непрерывного) сигнала задается заранее, эмпирические моды вычисляются в ходе итерационного вычислительного процесса, останов которого регулируется заранее заданной точностью или указанием количества выделяемых мод.

Разложение на эмпирические моды позволяет анализировать локальные явления (как состояния динамических объектов), поэтому данный метод может быть использован при обработке нестационарных временных рядов (случайных процессов).

Эмпирическая мода, внутреннее колебание или мода (англ. *intrinsic mode functions*, IMF) — это такая функция, которая обладает следующими двумя ограничительными свойствами [4]

- 1) количество экстремумов (и максимумов и минимумов) и количество пересечений нуля не должны отличаться более чем на единицу;
- 2) среднее значение, которое определяется по двум огибающим — верхней и нижней, должно быть равно нулю.

Процедура выделения эмпирических мод называется просеиванием (англ. *sifting*) [4].

Рассмотрим реализацию метода модовой декомпозиции на исходных данных [3, рис. 1].

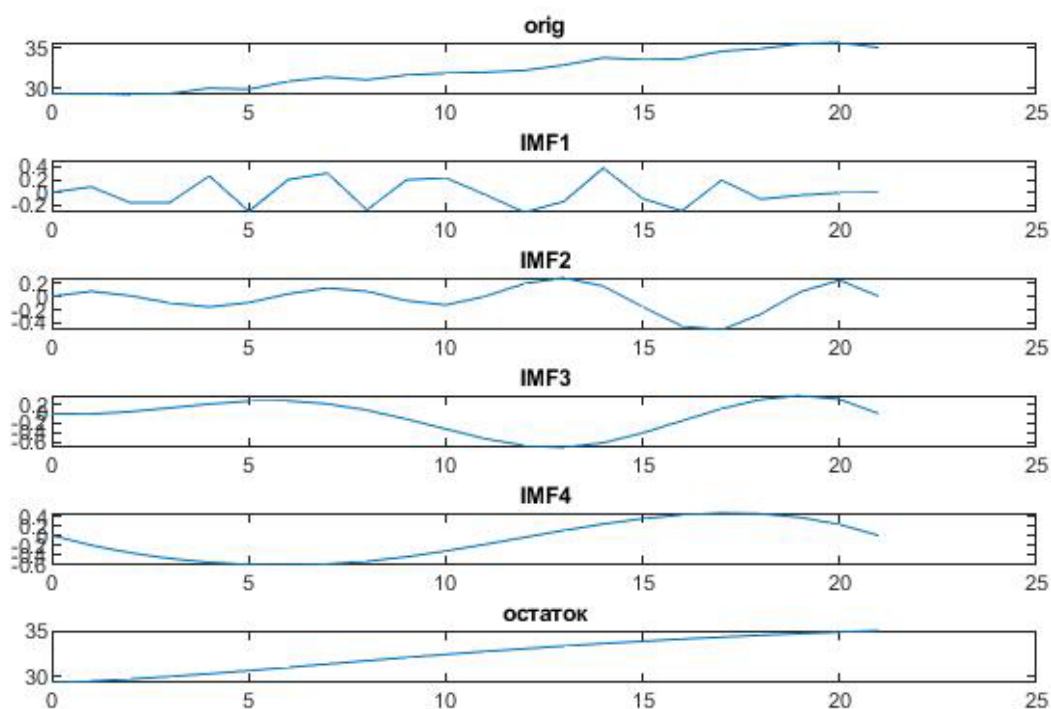


Рис. 2. Исходный график и его компоненты

После процедуры просеивания временного ряда были выявлены его моды (IMF1, IMF2, IMF3, IMF4) и остаток функции, который был принят в качестве тренда графика.

Поскольку поставлена задача построения краткосрочного прогноза, то целесообразно использовать модель Брауна[2], согласно которой формируется прогноз для каждой из полученных компонент, а затем суммируются все прогнозные значения целью получения итогового результата.

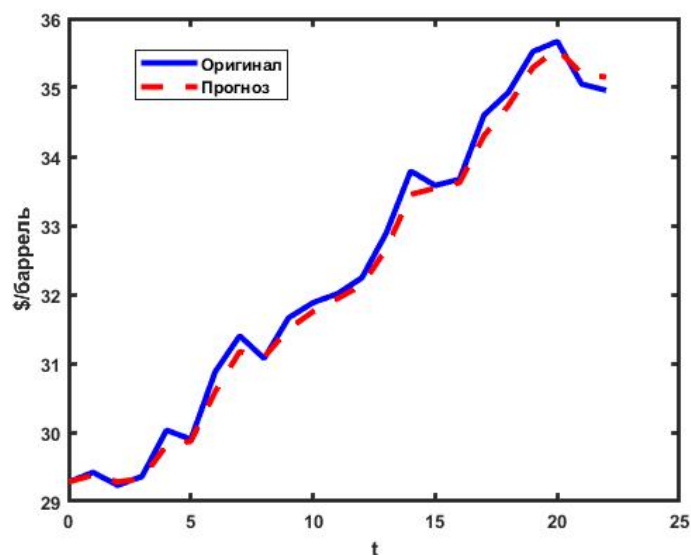


Рис. 3. График сравнения реальных данных и прогнозных

Из графика (рис. 3) видно, что данные, полученные по модели Брауна, слабо расходятся с реальными данными. Стандартная ошибка между фактическим значением и расчётными значениями всего временного ряда равна 0,17. Прогнозируемое значение цены на нефть за искомый период (19 мая 12:00) равно 35,15 \$/баррель, фактическое значение за этот же период равно 34,96 \$/баррель.

Заключение и дальнейшее развитие алгоритма прогнозирования. Результаты численной апробации алгоритма прогнозирования на базе метода EMD дают основание надеяться на создание эффективной процедуры прогнозирования при условии решения проблемы корректного учета «концевых» явлений эмпирических мод (недостаток EMD-алгоритма). В дальнейшем планируется построение алгоритма прогноза и его оценки в зависимости от характера шумовой составляющей и горизонта прогноза.

Библиографический список

1. *Huang N. E., Shen Z., Long S. R. et al.* The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis // Proc. R. Soc. Lond. A. 1998. Vol. 454. P. 903—995.
2. *Brown, Robert G.* Exponential Smoothing for Predicting Demand. Cambridge, Massachusetts: Arthur D. Little Inc. P. 15.
3. Сберометр. Радар валютных новостей: Нефть. URL: <https://www.sberometer.ru/rvn/oil/> (дата обращения: 20.05.2020).
4. Википедия. Empirical Mode Decomposition. URL: https://ru.wikipedia.org/wiki/Empirical_Mode_Decomposition (дата обращения: 08.05.2020).

УДК 004.934

Н. Н. Реутов

студент кафедры компьютерных технологий и программной инженерии

М. Д. Поляк – старший преподаватель – научный руководитель

РАСПОЗНАВАНИЕ РУССКОЙ РЕЧИ ПРИ ПОМОЩИ НЕЙРОСЕТЕВОЙ МОДЕЛИ

Вступление

В декабре 2014 года, компания Baidu представила систему распознавания речи “DeepSpeech”, основанную на сквозном глубоком обучении. Данная система обладала более простой архитектурой, чем другие аналогичные системы и при этом давала значительно лучшие результаты на зашумленных данных. Этот результат достигался за счет тренировки системы на тысячах часов входных данных, которые, к тому же, были дополнены при помощи современных систем модификации звука, что позволило сделать обучающую выборку еще более вариативной. [1].

Важным преимуществом DeepSpeech является возможность обучения другим языкам без кардинальных изменений самой системы. Целью данной работы является составление обучающей выборки для последующего обучения на ней нейросети с подбором оптимальных параметров.

Принцип работы системы

Распознавание текста в системе DeepSpeech состоит из нескольких этапов:

Сначала аудиозапись разбивается на так называемые «окна» – участки длиной несколько миллисекунд и происходит анализ спектрограммы аудиозаписи, в результате которого получается матрица, где строки соответствуют символам, а столбцы – окнам. Данная матрица также называется матрица CTC и представляет собой вероятность того, что в определенном окне звучал определенный символ. Полученная матрица сравнивается с матрицей, представляющей текст, который действительно звучит на аудиозаписи (транскрипт), после чего высчитывается расхождение по CTC [2], отражающее ошибку предсказания и являющееся целевой функцией. Таким образом, система оперирует не фонемами языка, а символами его алфавита, что позволяет обучать ее различным языкам без изменений в архитектуре.

Затем, полученная цепочка символов оптимизируется при помощи префиксного поиска по языковой модели, которая представляет собой набор N-граммов, отражающих наиболее частые последовательности слов, встречающиеся в данном языке. К примеру, получив цепочку символов «сапака_быфайт_кусащей», после обработки префиксным поиском система получит знакомую фразу «собака бывает кусачей»

Исходя из принципов работы системы, становится ясен набор задач, которые необходимо выполнить для достижения цели:

- 1) необходимо создать обучающую выборку, состоящую из записей русской речи и соответствующих им транскриптов, причем эти аудиозаписи должны быть достаточно короткими (до 10 слов), для обеспечения эффективного подсчета CTC
- 2) необходимо создать языковую модель, подобрав при этом оптимальный параметр длины N-граммов

Подготовка данных

Для обучения глубокой нейросети необходим колоссальный объем данных. [3] Минимальный рекомендуемый объем данных для данной системы – от 500 часов. Компания Baidu использовала для обучения наборы данных от Wall Street Journal, Switchboard и Fisher, суммарной длительностью более 7000 часов. [1] К сожалению, для русского языка удалось найти лишь небольшие объемы данных (порядка 20 часов), поэтому обучающая выборка была создана самостоятельно.

Для создания обучающей выборки необходимо найти большой объем аудиозаписей достаточно хорошего качества, который сопровождался бы транскриптом, достаточно точным по таймингу и содержанию. Источником подобных данных является сервис YouTube с его функцией автоматического

субтитрирования. Субтитры появляются на экране ровно во время произнесения текста, а значит обладают большой точностью по времени. Но этот транскрипт не является достаточно точным по содержанию, что неприемлемо при формировании обучающей выборки. Поэтому, для генерации субтитров были использованы аудиокниги, так как их текст строго соответствует тексту оригинала (бумажной книги), а также данные аудиозаписи обладают высоким качеством (хорошая дикция говорящего, отсутствие сторонних шумов). Обработка такого большого количества данных, разумеется, невозможна вручную. Поэтому, было создано несколько скриптов на языке Python для оптимизации процесса загрузки книг, обработки субтитров и нарезки аудиокниг на короткие треки.

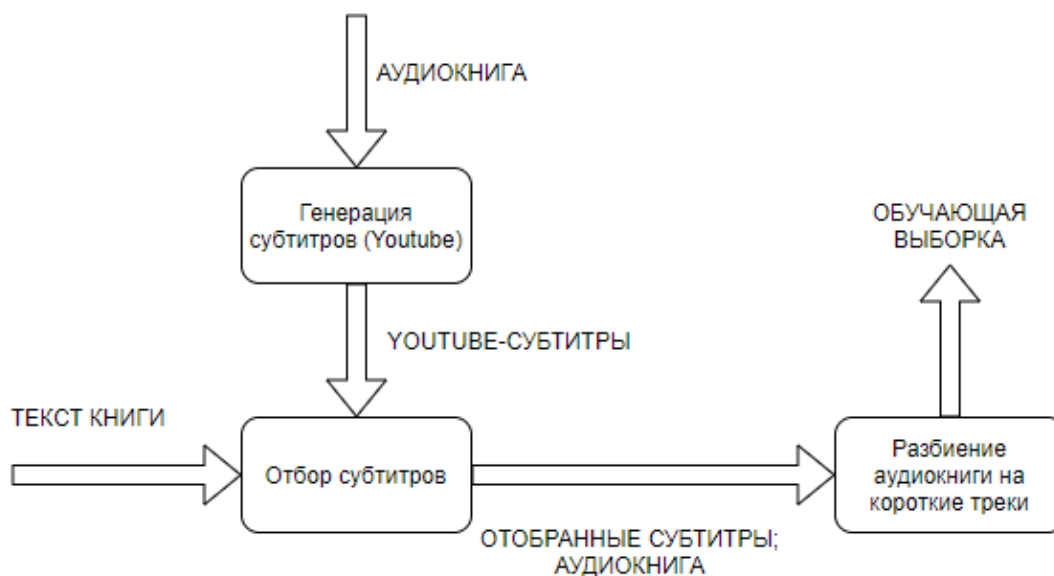


Рис. 1. Схема генерации обучающей выборки

На первом этапе, аудиокнига разбивается на отрезки строго меньше 4 часов (условие генерации субтитров сервисом YouTube) и загружается на сервис YouTube. После обработки записи сервисом, появляются субтитры, представляющие собой короткие фразы длиной до десяти слов, сопровождающиеся временными метками начала и конца фразы в записи.

Второй этап – получение «чистых» субтитров. Для этого, каждая строка субтитров ищется в оригинальном тексте книги, из которого предварительно удаляются все знаки табуляции и пунктуации. При строгом совпадении, дорожка с ее временными метками записывается в файл проверенных субтитров. На данном этапе отсеивается порядка 40-55 % субтитров, в зависимости от качества аудиозаписи.

Третий этап – разбиение файла аудиокниги на короткие треки (длиной порядка 5-7 секунд), занесение имен файлов и транскриптов в таблицу. Во избежание повторений при именовании файлов, каждый файл называется комбинацией из 16 символов латинского алфавита, задаваемой специальным алгоритмом.

Таблица 1

Пример таблицы транскриптов

Path	Sentence
ajaaaaaaaaaaaaae.mp3	звонок обрывается и сейчас же вспыхивает
ajaaaaaaaaaaaaaf.mp3	угрюмый мужчина отбрасывает одеяло и
ajaaaaaaaaaaaaag.mp3	твердо сжатые губы он словно и не спал

Для создания языковой модели был использован проект KenLM [4]. Исходными данными для языковой модели послужил набор из более чем 16 000 книг различных жанров (от научно-популярного до фэнтези) и авторов (как отечественных, так и зарубежных). Из этих данных были сгенерированы несколько языковых моделей, с различной длиной N-грамма (от 3 до 6 слов). Впоследствии, эти модели будут использованы для определения оптимальной для русского языка длины N-грамма эмпирическим путем.

Проблемы

Очевидным недостатком создаваемой выборки является достаточно малое количество различных голосов в ней. Данный недостаток в последствии будет решен в процессе обучения при помощи дополнения набора данных «на лету» средствами модификации звука (изменение тона и скорости речи), предоставляемыми DeepSpeech.

Также, возможным недостатком данного набора данных является его «чистота» – практически полное отсутствие сторонних шумов. Нейросеть, обученная на таких данных, скорее всего будет сильно подвержена влиянию шумов в распознаваемой аудиозаписи. Для решения данной проблемы была разработана система, позволяющая «зашумлять» аудиозаписи из выборки, путем наложения на них типичных фоновых шумов (шум улицы, офиса и так далее). Таким образом, будут получены два набора данных: «зашумленный» и «чистый», что позволит, в последствии экспериментировать с составом обучающей выборки и дообучением нейросети. Эффективность такого подхода доказана в работе Ш. Инь, Ч. Лью, Ж. Джанг, [5]

Узким местом созданной системы является первичная генерация субтитров сервисом Youtube, поскольку сервис ограничивает количество записей, загружаемых в сутки, а также периодически блокирует записи за нарушение авторских прав. Также, Youtube API [6] имеет привязку к Google-аккаунту и жестко ограничивает количество запросов к сервису, что также снижает эффективность

Выводы и дальнейшее развитие

Созданная система позволяет сгенерировать достаточно значительный объем данных, работая при этом в автоматическом режиме. Возможным улучшением является поиск альтернативного способа генерации первичных субтитров.

В дальнейшем предстоит провести ряд экспериментов с процентным соотношением «чистых» и «зашумленных» фрагментов в итоговой выборке, а также длиной N-грамма

Интересным направлением дальнейшей работы также представляется дополнение выборки аудиофрагментами, созданными при помощи технологий синтеза речи. Данные технологии позволяют имитировать характерные особенности произношения и, как показывает работа Дж. Ли и Р. Гейда,[7], данный подход позволяет увеличить процент распознавания текста.

Библиографический список

1. *Hannun A.* Deep Speech: Scaling up end-to-end speech recognition // arXiv:1412.5567. 2016.
2. *Graves A., Fernandez S.* Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks // ICML '06: Proceedings of the 23rd international conference on Machine learning. 2006. P. 369–376.
3. Deep Learning. Nature. 2015. № 521. P. 436–444. Doi: 10.1038/nature14539.
4. *Ken L. M.* Language Model Toolkit. URL: <https://kheafield.com/code/kenlm/> (дата обращения: 21.05.2020).
5. *Yin Sh., Liu Ch., Zhang Zh.* Noisy training for deep neural networks in speech recognition // EURASIP Journal on Audio, Speech, and Music Processing. 2015. Doi: 10.1186/s13636-014-0047-0.
6. YouTube DataAPI. Python. URL: <https://developers.google.com/youtube/v3/quickstart/python> (дата обращения: 21.05.2020).
7. *Li J., Gadde R., Ginsburg B., Lavrukhin V.* Training Neural Speech Recognition Systems with Synthetic Speech Augmentation // arXiv:1811.00707. 2018.

УДК 519.233.24

И. В. Ряполова

студент кафедры компьютерных технологий и программной инженерии

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

О ПОСТАНОВКЕ ЗАДАЧИ СОЗДАНИЯ ПРОГРАММЫ ДЛЯ ОПРЕДЕЛЕНИЯ ИНДИВИДУАЛЬНОЙ ОСОБЕННОСТИ ГОЛОСОВОГО АППАРАТА И ВЫРАБОТКИ РЕКОМЕНДАЦИЙ ПО ЕГО ЦЕЛЕСООБРАЗНОЙ НАСТРОЙКИ

В рамках научного подхода в изучении процессов звукоизвлечения и звуковосприятия за последнее столетие было создано множество акустических, механических, а впоследствии, радиотехнических, в том числе цифровых моделей голосового и слухового трактов, благодаря которым удалось выявить критерии, характеризующие состояние аппарата певца. Но до сих пор остается проблема оценки вокальной одаренности и правильного звукоизвлечения певца.

Разрабатываемая система поддержки принятия решений должна иметь возможность извлекать параметры речи, которые могут быть использованы для диагностики следующих голосовых характеристик.

Основными особенностями голоса профессионального певца являются звонкость, «полетность» и «серебристость» тембра (по терминологии вокальных педагогов). Эти характеристики обусловлены наличием ярко выраженных низкочастотных и высокочастотных составляющих спектра певческого голоса – формантных областей, а также амплитудно-частотной модуляцией звуковых колебаний во времени (вibrato) [1].

Высокая певческая форманта (ВПФ) – это группа усиленных обертонов в спектре певческого голоса, обладающая высокой интенсивностью у профессиональных исполнителей. Частота области высокой (третьей речевой) форманты лежит в пределах от 1800 до 3500 Гц (в зависимости от типа голоса). Также в спектре голоса певца присутствуют область низкой певческой (первой речевой) форманты (200 – 1200 Гц) и область средней певческой (второй речевой) форманты (800 – 1800 Гц) [2, 3].

Наибольшей интенсивностью обладает третья форманта, называемая высокой певческой формантой, которая свойственна профессиональным вокалистам; по этой причине выберем ее в качестве критерия оценки качества голоса.

Заметим, что у профессионалов с большим стажем работы в области высокой певческой форманты сосредоточено до 30-35 % всей звуковой энергии голоса; у начинающих она достигает только 5-7 % [3]. Так, на рисунках 1 и 2 представлены спектры профессиональных певцов Ольги Бородиной и Евгения Нестеренко. В обоих случаях в области высокой певческой форманты сосредоточено больше 40 % всей звуковой энергии голоса. На рис. 3 представлен спектр начинающего вокалиста. В зоне высокой певческой форманты сосредоточено лишь 10 % всей звуковой энергии голоса.



Рис. 1. Амплитудный спектр голоса Евгения Нестеренко.

ЧОТ – частота основного тона, $E = 6.2686e+05$, $E_{pf} = 3.1349e+05$, $ВПФ = 0.5001$



Рис. 2. Амплитудный спектр голоса Ольги Бородиной. ЧОТ – частота основного тона, $E = 1.2381e+07$, $E_{pf} = 6.1887e+06$, $ВПФ = 0.4998$

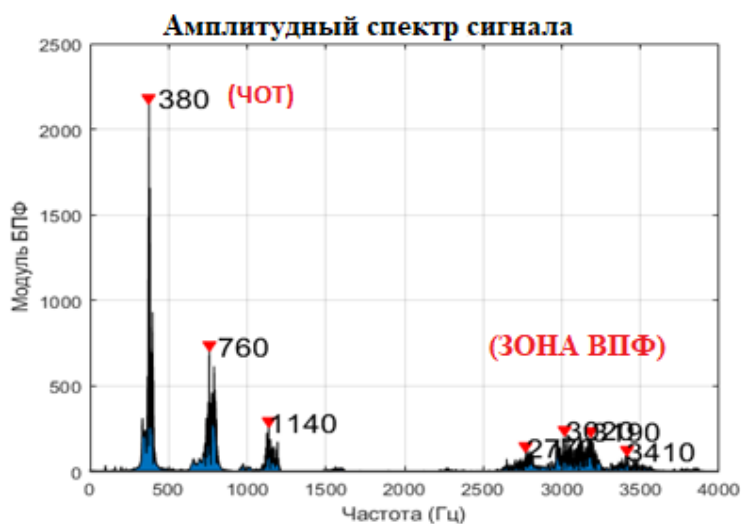


Рис. 3. Амплитудный спектр голоса новичка. ЧОТ – частота основного тона, $E = 3.0892e+07$, $E_{pf} = 2.7083e+06$, $ВПФ = 0.0877$

Разрабатываемая система поддержки принятий решений должна иметь возможность при использовании спектрального анализа извлекать высокую певческую форманту, которая может быть использована для диагностики голосовых патологий.

Блок-схема алгоритма обработки речевого сигнала представлена на рис. 4



Рис. 4. Блок-схема алгоритма обработки речевого сигнала

Непосредственно с системой взаимодействует ученик.

Основные функции, которые реализует данная система:

- общий анализ голоса на выявление дисфоний;
- профессиональный анализ голоса (анализ показателей – высокая певческая форманта, низкая певческая форманта и т. д.).

Данное взаимодействие можно представить в виде диаграммы прецедентов (рис. 5)

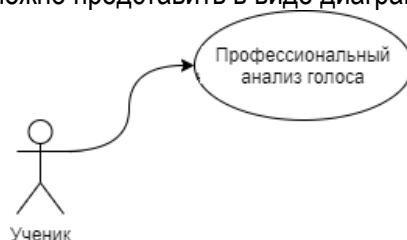


Рис. 5. Диаграмма прецедентов

Основной прецедент характеризуется положениями:

- начало: выбор режима профессионального анализа;
- запись голоса;
- выделение вектора характеристик (признаков) голоса как временного ряда (сигнала);
- сравнение с эталонными значениями по каждому показателю (критерию);
- вывод информация о градации (степени) профессиональности певца;
- завершение процедуры оценивания голоса как временного ряда.

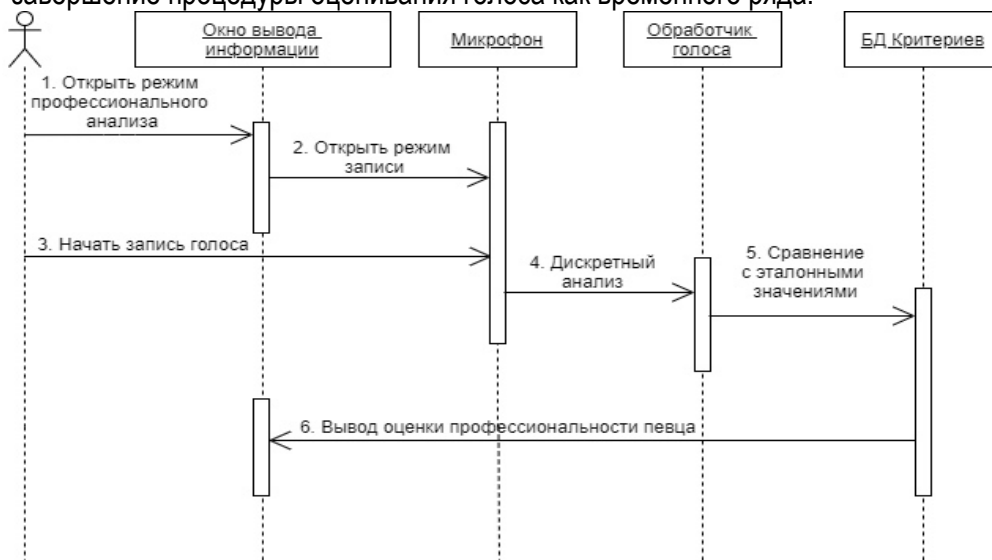


Рис. 6. Диаграмма последовательности событий для прецедента профессиональный анализ голоса

Таким образом, сформированы спецификации требований к планируемому софту системы поддержки принятий решений с рассмотрением требований в предметной области. Смоделирована предметная область, архитектуры программного обеспечения с использованием нотации UML. Сформулированы требования к создаваемому программному продукту.

Библиографический список

1. Морозов В. П. Искусство резонансного пения. Основы резонансной теории и техники. М.: МГК, ИПРАН, 2008.
2. Морозов В. П. Измерение полетности голоса // Голос и речь. 2012. № 3. С. 16–28.
3. Спектральный компьютерный анализ / Ю. С. Василенко, А. П. Мещеркин, О. Г. Павлихин, С. Г. Романенко. URL: <https://nikio.ru/спектральный-анализ-голоса/> (дата обращения: 20.05.2020).

УДК 681.5

Е. Л. Семенухин

студент кафедры компьютерных технологий и программной инженерии

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

НОВАЯ СИСТЕМА УПРАВЛЕНИЯ ЭЛЕКТРОГИДРАВЛИЧЕСКИМ ДИСКОВЫМ ТОРМОЗОМ «SELF-ENERGISING ELECTRO-HYDRAULIC BRAKE» (SEHB) А ПРИНЦИПАХ СИНЕРГЕТИЧЕСКОЙ ТЕОРИИ УПРАВЛЕНИЯ

Цель настоящего исследования – конструирование новой энергосберегающей системы управления тормозной системы для электропоездов на основе определенного функционала качества управления, удовлетворяющего принципам физической теории управления [1], и создание программного средства для полноценного численного моделирования разработанного регулятора.

Задачи, реализующие решение указанной проблемы, следующие:

- 1) представить алгоритм синтеза системы управления для детерминированной (полно определенной) модели нелинейного объекта 4-го порядка (электрогидравлический дисковый тормоз (SEHB));
- 2) разработать программный модуль для численной реализации представленного в п.1;
- 3) разработать новый алгоритм синтеза системы управления для неполно определенной модели нелинейного объекта 4-го порядка (электрогидравлический дисковый тормоз) с неслучайными помехами (функция времени);
- 4) разработать программный модуль для численной реализации представленного в п.3;
- 5) представить алгоритм синтеза системы управления для стохастической дискретизованной модели нелинейного объекта 4-го порядка (система нелинейных стохастических разностных уравнений);
- 6) разработать программный модуль для численной реализации представленного в п.5.

В данной статье будет представлена разработка альтернативной системы управления, обеспечивающую удовлетворительную гладкость переходных процессов для объекта в детерминированных условиях. Объектом является электрогидравлический дисковый тормоз SEHB (рис. 1) [2, 3].

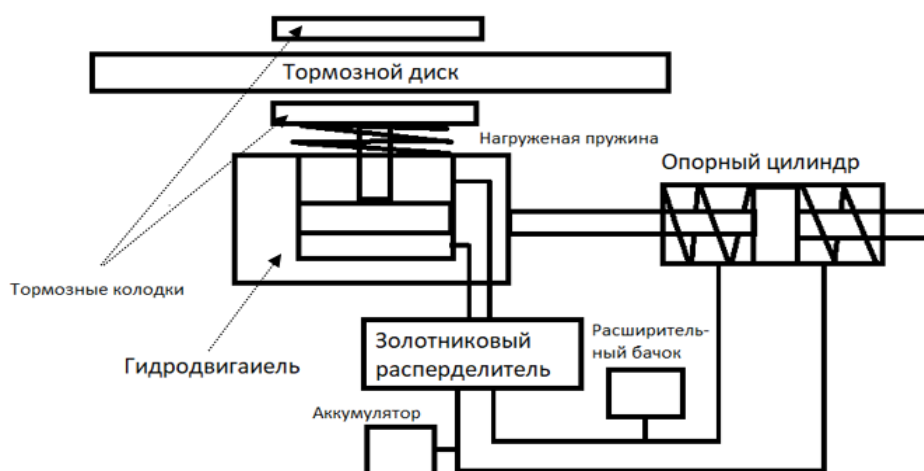


Рис. 1. Схема SEHB

Рассматриваемый объект (рис. 1) является фрикционным тормозом, который использует тормозную силу в качестве источника энергии для генерирования нового тормозного усилия посредством передачи электрогидравлической энергии. Гидравлический цилиндр двойного действия, который будем называть опорным цилиндром, связывает суппорт тормоза с конструкцией тележки, фиксируя поршень гидроцилиндра между двумя столбцами жидкости. В случае торможения сила трения действует на поршень опорного цилиндра, вызывая изменение давления жидкости в его полостях. Давление жидкости в верхней гидролинии практически не меняется из-за установленного расширительного бачка, объем которого много больше, чем совокупный объем линии и полости опорного цилиндра. Золотниковый распределитель связывает гидролинии с полостями гидродвигателя. Если активно прямое соединение гидролиний (случай при увеличении тормозной силы), то происходит нагнетание жидкости в поршневую полость гидродвигателя, тогда как штоковая его полость соединяется с линией слива. При этом выбранное отношение между полной и кольцевой площадями поршня дифференциального цилиндра приводит к самовозбуждающемуся процессу увеличения тормозной силы. Если действует обратное соединение гидролиний (случай при уменьшении тормозной силы), то поршневая полость гидродвигателя соединяется с линией слива, а в штоковую полость поступает жидкость из опорной линии.

Имеющимися аналогами методов управления данным объектом является

- 1) ПИД-регулятор [2], к недостаткам которого относится колебательный характер торможения, что приводит к износу механической части тормоза;
- 2) регулятор, полученный на основе линеаризации обратной связью [3], имеющий затраты на управление в 4-11 раз меньше по сравнению с затратами на управление в системе с ПИД-регулятором, но демонстрирующий тоже неудовлетворительные свойства при критических значениях коэффициента (параметра) торможения.

В качестве исходной математической модели рассмотрена система уравнений 4-го порядка, полученная из редуцированной системы 10-го порядка (показано, что для синтеза управления этого достаточно в силу синергетической взаимосвязанности координат объекта). При этом в силу симметрии уравнений будет рассмотрен только случай при увеличении тормозной силы [2, 3]:

$$\begin{cases} \dot{x}_1 = a_1 x_4 \sqrt{x_2 - x_1 - \alpha p}; \\ \dot{x}_2 = a_2 x_4 \sqrt{x_2 - x_1 - \alpha p}; \\ \dot{x}_3 = -a_3 x_3 - a_4 x_4 + g u; \\ \dot{x}_4 = x_3. \end{cases}$$

Здесь переменные и параметры объекта управления имеют описание:

- x_1 – давление нагрузки гидродвигателя;
- x_2 – давление в опорной линии;
- x_3 – скорость штока плунжера;
- x_4 – перемещение штока в плунжере;
- α – отношение полной и кольцевой площадей гидродвигателя;
- p – давление слива;
- a_1 – введённый коэффициент для нагрузки гидродвигателя;
- a_2 – введённый коэффициент для поршня опорного цилиндра;
- a_3 – коэффициент демпфирования золотникового плунжера, умноженный на собственную частоту колебаний золотникового плунжера;
- a_4 – собственная частота колебаний золотникового плунжера в квадрате;
- g – коэффициент передачи золотникового распределителя, умноженный на собственную частоту колебаний золотникового плунжера в квадрате;

- u – параметр управления.

Как следует из формы уравнений система является нелинейной. Известно, что для нелинейных систем не существует ни общей теории анализа, ни общей теории синтеза регуляторов. Используемый далее новый метод является не только универсальным, но и логически прозрачным и использующий простой математический аппарат принципа минимального действия теоретической механики [4].

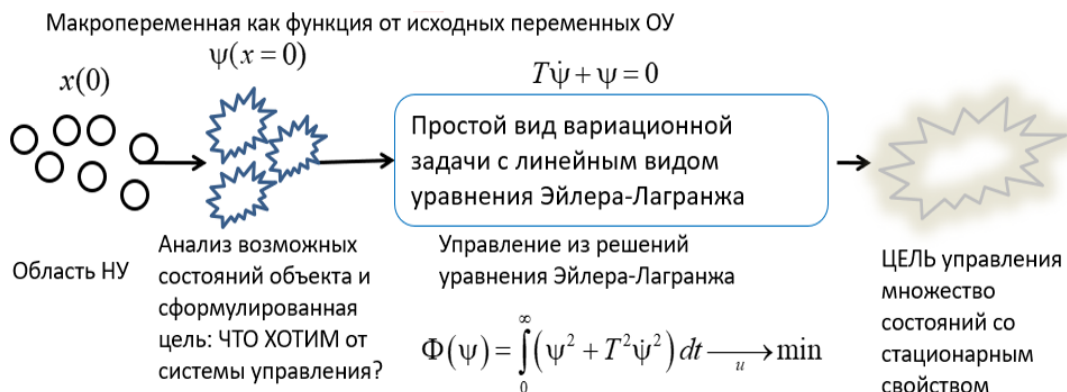


Рис. 2. Иллюстрация содержания метода АКАР для случая скалярного управления

Главная особенность метода, позволяющего относительно простой нелинейный аналитический метод синтеза регуляторов – знание аналитически описанной цели управления в виде $\psi=0$, где $\Phi(\psi)$ представляет собой обобщение классического квадратичного функционала качества при известной функции (макропеременной) ψ . Нетрудно заметить по виду функционала качества (рис. 2), что его минимум не только гарантирует достижение цели, но позволяет регулировать скорость достижения цели за счет параметра $T>0$. Включение макропеременной и ее производной в критерий будет требовать управляемой, максимально энергетически экономной, целенаправленной самоорганизации переменных объекта, что полностью отвечает требованиям физической теории управления (А.А. Красовский) [4].

Конструкция удобна тем, что уравнение Эйлера-Лагранжа для $\Phi(\psi)$ имеет линейный вид, из которого относительно несложно определить переменную управления. Таким образом совокупность описанных условий и предположений гарантирует определение аналитического регулятора.

На первом этапе синтеза необходимо найти функцию внешнего управления u , в соответствии с заданным целевым инвариантом $\psi_1 = 0$, вводится макропеременная $\psi_1 = x_3 - \varphi_1(x_1, x_2, x_4)$, учитывая уравнение Эйлера-Лагранжа относительно устойчивых экстремалей, решаем основное функциональное уравнение $\psi_1 + T_1\dot{\psi}_1 = 0$ и находим выражение для внешнего управления $gu = -\frac{\psi_1}{T_1} + \dot{\varphi}_1 + a_3x_3 + a_4x_4$.

На втором этапе синтеза проводится декомпозиция и поиск внутреннего управления φ_1 в соответствии с заданными целевым инвариантом $\psi_2 = 0$, вводится макропеременная $\psi_2 = x_4 - \varphi_2(x_1, x_2)$ и полагается, что $x_3 = \varphi_1(x_1, x_2, x_4)$, решаем функциональное уравнение, являющееся уравнением Эйлера-Лагранжа $\psi_2 + T_2\dot{\psi}_2 = 0$ и находим выражение для внутреннего управления $\varphi_1 = -\frac{\psi_2}{T_2} + \dot{\varphi}_2$.

На третьем этапе синтеза проводится декомпозиция и поиск внутреннего управления φ_2 в соответствии с заданными целевым инвариантом $\psi_3 = 0$, вводится макропеременная $\psi_3 = x_2 - x_2^*$ и полагается, что $x_4 = \varphi_2(x_1, x_2)$, решаем функциональное уравнение, являющееся

уравнением Эйлера-Лагранжа $\psi_3 + T_3\dot{\psi}_3 = 0$ и находим выражение для внутреннего управления

$$\varphi_2 = -\frac{\psi_3}{T_3 a_2 \sqrt{x_2 - x_1 - \alpha p}}.$$

Собрав вышеуказанные соотношения, получим систему управления в виде:

$$\left\{ \begin{array}{l} \dot{x}_1 = a_1 x_4 \sqrt{x_2 - x_1 - \alpha p}; \\ \dot{x}_2 = a_2 x_4 \sqrt{x_2 - x_1 - \alpha p}; \\ \dot{x}_3 = -a_3 x_3 - a_4 x_4 + gu; \\ \dot{x}_4 = x_3; \\ \psi_1 = x_3 - \varphi_1; \\ gu = -\frac{\psi_1}{T_1} + \dot{\varphi}_1 + a_3 x_3 + a_4 x_4; \\ \psi_2 = x_4 - \varphi_2; \\ \varphi_1 = -\frac{\psi_2}{T_2} + \dot{\varphi}_2; \\ \psi_3 = x_2 - x_2^*; \\ \varphi_2 = -\frac{\psi_3}{T_3 a_2 \sqrt{x_2 - x_1 - \alpha p}}; \\ \dot{\varphi}_1 = \frac{\partial \varphi_1}{\partial x_1} \dot{x}_1 + \frac{\partial \varphi_1}{\partial x_2} \dot{x}_2 + \frac{\partial \varphi_1}{\partial x_4} \dot{x}_4; \\ \dot{\varphi}_2 = \frac{\partial \varphi_2}{\partial x_1} \dot{x}_1 + \frac{\partial \varphi_2}{\partial x_2} \dot{x}_2. \end{array} \right.$$

Численное моделирование сконструированной системы управления реализовано в среде разработки Matlab/Simulink.

На рис. 3-4 представленные изменения переменных замоделированной системы уравнений со значениями, представленными в табл. 1.

Таблица 1

Параметры системы уравнений

Параметр	Значение
A	0.609
P	394760.89
a ₁	17509.9
a ₂	0.026250715
a ₃	314.15
a ₄	394760.89
T ₁	100
T ₂	0.001
T ₃	1
Шаг интегрирования	0.001

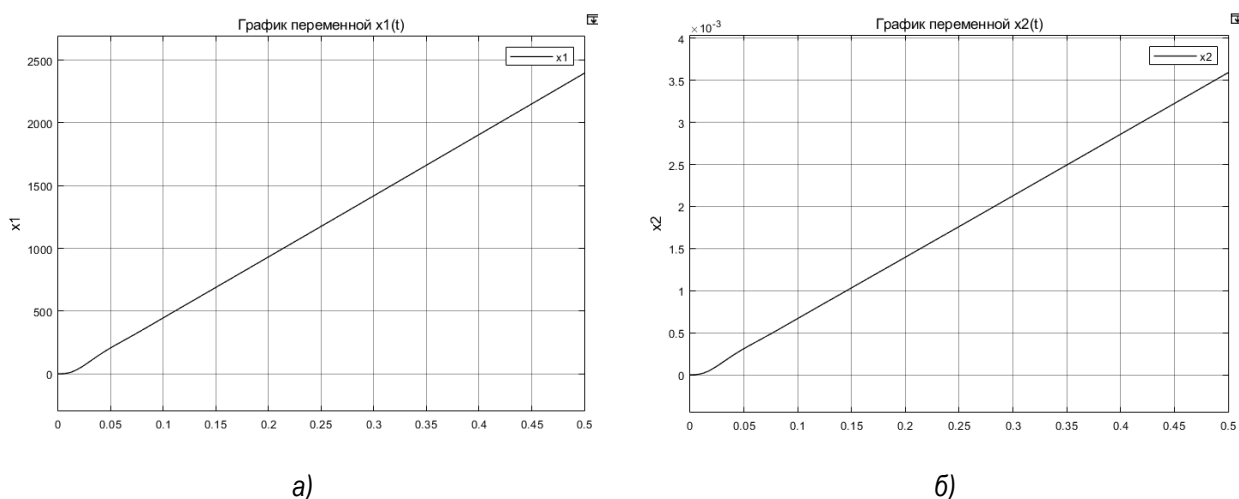


Рис. 3. Графики переменных а) – x_1 ; б) – x_2

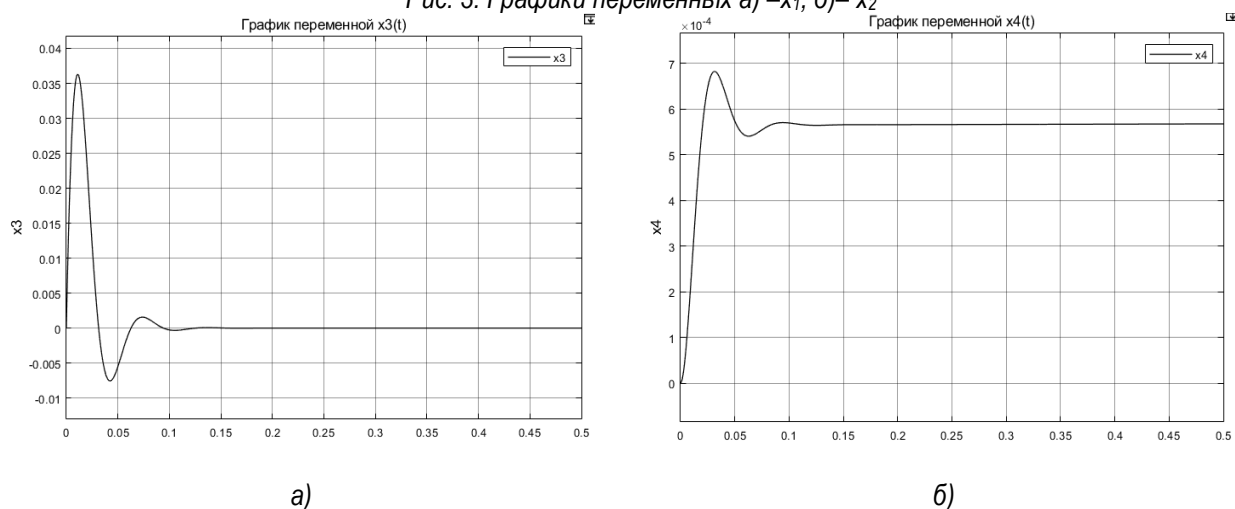


Рис. 4. Графики переменных а) – x_3 ; б) – x_4

Выводы и дальнейшие перспективы исследования.

На данном этапе исследования представлена математическая модель системы уравнений для объекта SEHB в детерминированных условиях. Используя среду разработки Matlab/Simulink было представлена численная апробация разработанной системы управления. Целесообразна разработка программного средства для автоматического подбора параметров регулятора для достижения поставленных целей управления перед объектом управления и проведения сравнения с имеющимися аналогами. Также планируются дальнейшие исследования относительно реализации построения алгоритма нелинейного регулятора для объекта с детерминированными помехами, который основан на новом методе нелинейной адаптации на многообразиях и алгоритм нелинейного дискретного регулятора со стохастическими помехами, которые впервые будет применен для данного объекта SEHB.

Библиографический список

1. Красовский А. А. Статистическая теория переходных процессов в системе управления. М.: Наука, 1968. 240 с.
2. Liermann M. Self-energizing Electro-Hydraulic Brake. //Diss. RWTH Aachen University. Aachen: ShakerVerlag, 2008. 175 p.
3. Старых А. А. Синтез нелинейного регулятора системы управления параметрически неопределенным объектом: дис. ... канд. техн. наук. 2009. 128 с.
4. Колесников А. А. Синергетика и проблемы теории управления. М.: ФИЗМАТЛИТ; Энергоатомиздат, 2004. 504 с.

УДК 519.248

Н. Н. Сергеев

магистрант кафедры компьютерных технологий и программной инженерии

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

АВТОМАТИЗИРОВАННОЕ ОЦЕНИВАНИЕ КВАЛИФИКАЦИОННЫХ НАВЫКОВ И ФОРМАЛИЗАЦИЯ ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ РАБОТНИКА

Известно, что на текущий момент основными позициями разработчика являются junior, middle и senior. Также известно, что граница между указанными градациями весьма размытая, и не существует четких общепринятых «знаний-умений-навыков» для каждого уровня. И тем не менее в каждой компании разработчиков субъективно «назначают» границы навыков для перехода на новый уровень, поэтому здесь в игру вступает человеческий фактор. Таким образом, проблема повышения объективности при выдвижении четких строго соблюдаемых требований к опыту разработки является актуальной.

Цель настоящей работы – проектирование и программная реализация системы «Вектор развития» для частичного решения указанной проблемы с целью установления корректного соответствия совокупности навыков работника определенной позиции, которая будет привязана к зарплате. Есть определенные причины рассматривать такую проблему как задачу ранижирования-выбора с применением имеющихся техник системного анализа и принятия решений (например, [1–4]).

Рассмотрим пример статистики работника в виде «цифровой карточки» (рис. 1). Цифровая карточка представляет из себя набор компетенций работника в числовом эквиваленте, что дает наглядное представление об уровне знаний разработчика.

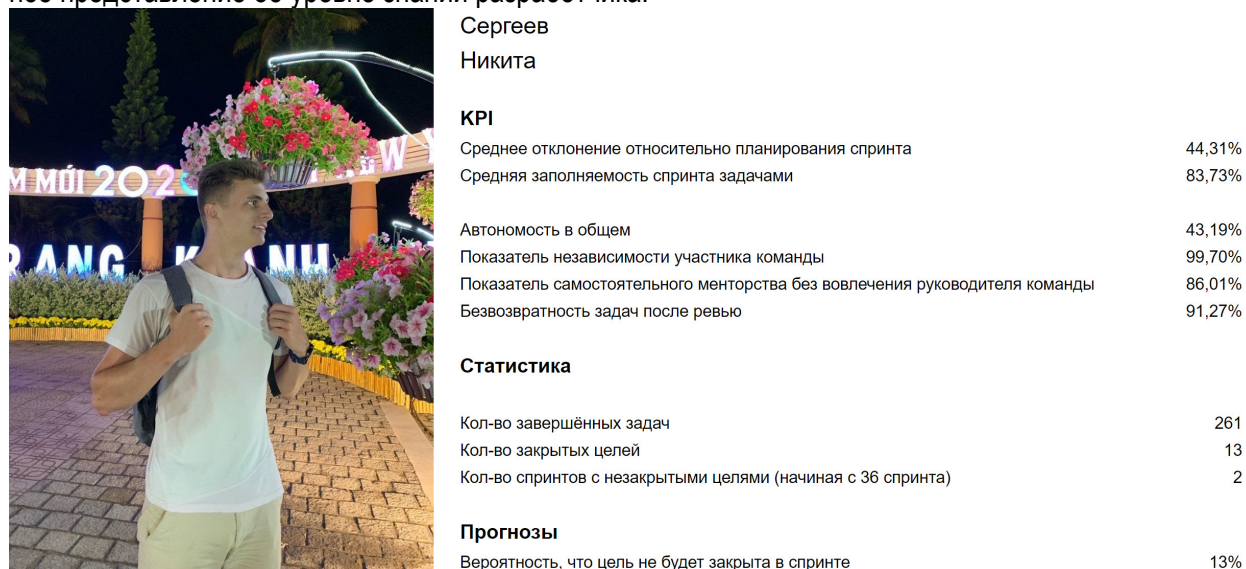


Рис. 1. Пример цифровой карточки

Основными плюсами цифровой карточки являются:

- матрица компетенций с необходимыми навыками на каждом уровне;
- привязка должности к зарплате;
- понятная система развития;
- профессиональные кадры.

Для реализации цели исследования были поставлены следующие сопутствующие задачи:

1. изучение и анализ предметной области, технологий и средств разработки для реализации задач сбора и обработки данных;
2. формализация персональных компетенций разработчиков ПС;
3. перевод навыков работника в цифровую карточку;
4. анализ результатов оценивания квалификационных навыков разработчиков ПС;

5. составление таблицы соответствия уровня квалификации и обработанных данных о сотрудниках;
6. сопоставление результатов оценивания по отношению к заявленным должностям.

Отметим существующие решения данного вопроса на данный момент.

Одной из систем, где имеется матрица развития профессиональных «умений-навыков» является платформа Grow [5], разработанная компанией EPAM. Ранее она использовалась внутри компании и недавно вышла на рынок. Платформа не предоставляет возможности автоматического оценивания разработчика, но предоставляет матрицу компетенций и варианты достижения целей для каждого уровня развития. Выбор индивидуального плана развития подбирается с учетом текущего уровня разработчика.

Другой системой оценивания сотрудника является система E-Stuff[6]. Система направлена на оценивание работников на этапе трудоустройства. Программа формирует приоритизированный список работников, основываясь на резюме и данных из интернета. Основными преимуществами являются:

- обширный функционал;
- хорошая аналитика:
 - на каких этапах собеседования отсеиваются кандидаты;
 - возможность просмотра данных по различным срезам (от какого рекрутера больше всего кандидатов, по руководителю, по стадии, на которой больше всего отсеиваются работников).

Из минусов системы можно выделить:

- устаревший дизайн;
- отсутствия интуитивно понятного интерфейса;
- очень медленная работа;
- платное распространение;
- платное внедрение и дополнительный функционал.

Таким образом, на рынке доступны системы, закрывающие проблему не комплексно. Остается ручная работа.

Реализация. Для осуществления анализа уровня разработчика необходимы данные о его работе. В основном разработчик работает с текстом кода, поэтому в качестве примера первоисточников данных были выбраны система управления и мониторинга работы JIRA [7] и репозиторий проектов GitHub [8].

Основные причины выбора данных систем – это обширная база пользователей. Система JIRA предоставляет данные о задачах и людях, ответственных за них, с возможностью генерировать отчеты о прогрессе разработчика. GitHub необходим для анализа кода разработчика. Оба инструмента имеют API для разработчиков, что предоставляет возможность интеграции с данными инструментами.

Общая схема разрабатываемой системы «Вектор развития» представлена на диаграмме компонентов (рис.2).

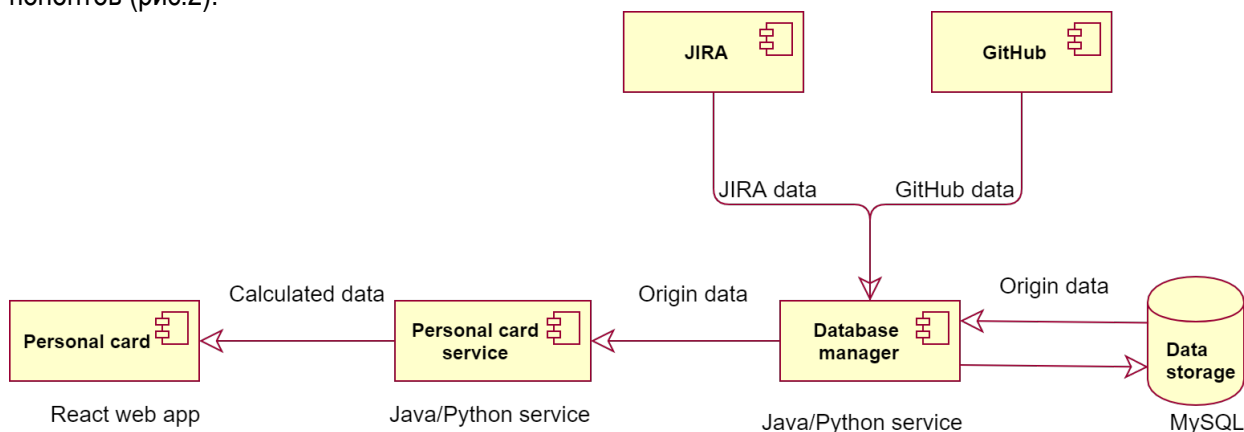


Рис. 2. Общее представление системы «Вектор развития» в виде диаграммы развертывания

Перечислим кратко некоторые реально используемые в работе показатели оценивания выполнения задач для формализации оценивания работы разработчика.

1. **Автономность.** Задачи должны выполняться самостоятельно с минимальным вовлечением других участников. Одним из критериев автономности является безвозвратность задач после проверки. Чем меньше задача переделывается – тем качественнее она выполнена и тем меньше времени других членов команды тратится на ее проверку. Расчет осуществляется по формуле:

$$a_1 = y(n), y = (100\%, 80\%, 50\%, 0\%, 0\%, \dots 0\%), n \in [1, \infty] (1)$$

где: n – количество отметок рецензента в тайм-трекере.

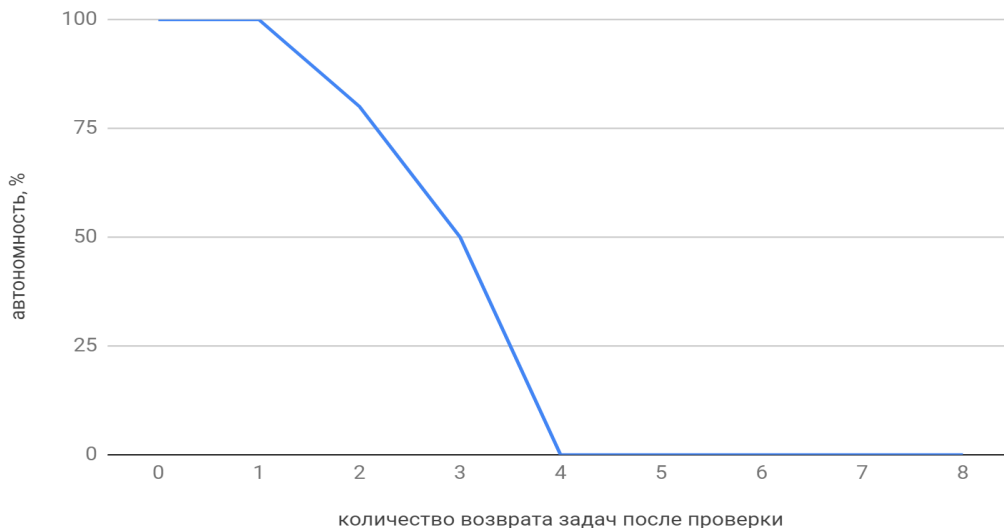


Рис. 3. График показателя

2. **Полнота итерации.** Определяет, насколько полно была выполнена итерация. Расчет осуществляется по формуле:

$$C_f = \frac{T_{pc}}{C_p} \cdot 100\% (2)$$

Где C_p – плановая полнота итерации (норма на которую итерация должна быть укомплектована); T_{pc} – общее плановое время выполненных за итерацию задач T

3. **Относительная оценка времени выполнения работы** показывает, как работник уложился в запланированное время. Расчет осуществляется по формулам:

$$T_{pc} = \sum_{i=1..k} t_{pci}$$

$$T_{fc} = \sum_{i=1..k} t_{fci}$$

$$D_c = \frac{T_{pc} - T_{fc}}{T_{pc}} \cdot 100\%$$

$$A_c = 1 - D_c (3)$$

Где k – количество выполненных задач; T_{pc} – общее плановое время выполненных за итерацию задач; t_{pci} – плановое время на i -ую выполненную задачу; T_{fc} – общее фактическое время выполненных за итерацию задач; t_{fci} – фактическое время на i -ую задачу; D_c – отклонение от оценки по выполненным за итерацию задачам.

Таким образом, основываясь на данных показателях можно сформировать статистику динамики показателей разработчика.

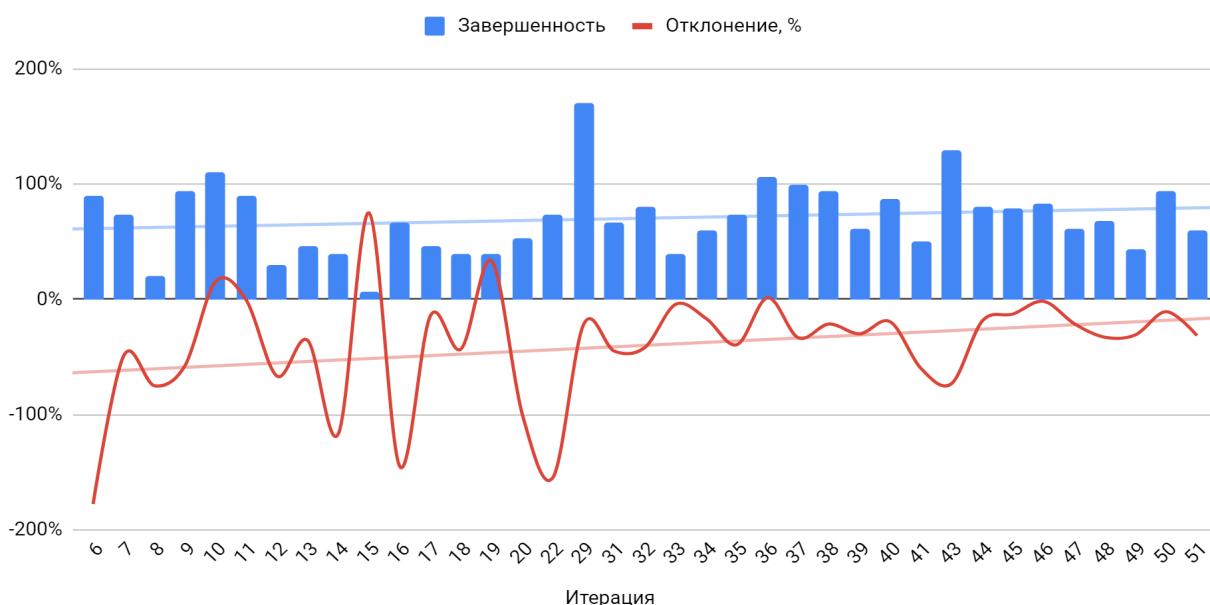


Рис. 4. Статистика динамики показателей отдельного работника (по неделям)

Для плохоформализуемых задач (не имеющих общепринятого формального описания), как правило, используются методы теории принятия решений с целью анализа ситуаций (вариантов возможных решений) и, собственно, принятия решений.

Здесь исследуется возможность применения широко известного метода ELECTRE I [9], предназначенного для решения многокритериальных задач.

Принцип работы указанного метода следующий. Для каждой пары альтернатив A_j и A_k выдвигается предположение (гипотеза) о том, что альтернатива A_j лучше, чем A_k . Затем для каждой пары альтернатив находятся два индекса: индекс согласия (величина, подтверждающая предположение о превосходстве A_j над A_k) и индекс несогласия (величина, опровергающая это предположение). На основе анализа этих индексов выбирается одна или несколько лучших альтернатив («ядро» альтернатив).

Для корректного (здесь – безошибочного на обучающей выборке) решения задачи скаляризации множества оценок следует использовать несколько критериев.

Суть *аддитивного критерия* заключается в следующем. Целевая функция (обобщенный критерий) образуется путем сложения нормированных значений частных критериев (3). Частные критерии имеют, как правило, различную физическую природу и, в этой связи, различную размерность. Поэтому при образовании обобщенного критерия следует оперировать не с «натуральными» критериями, а с их нормированными значениями.

Нормированным критерием называется отношение «натурального» частного критерия к некоторой нормирующей величине, измеренной в тех же единицах, что и сам критерий.

$$F(X) = \sum_{i=1}^n C_i * \frac{F_i(X)}{F_i^0(C)} = \sum_{i=1}^n C_i * f_i(X) \rightarrow \max(\min) \quad (3)$$

Аддитивный критерий имеет ряд недостатков, главный из которых состоит в том, что он требует для корректного его применения независимость альтернатив по предпочтению, что на практике проверить весьма затруднительно.

Следует также учитывать, что при его применении может происходить взаимная компенсация частных критериев (значительное уменьшение одной из критериальных оценок вплоть до нулевого значения может быть компенсировано возрастанием оценки по другому критерию). Для ослабления этого недостатка следует вводить ограничения на минимальные значения оценок по частным критериям и их весовых коэффициентов.

В случае неравноценности оценок по частным критериям вводятся весовые коэффициенты C_i мультипликативный критерий принимает вид (4)

$$F(X) = \prod_{i=1}^n C_i F_i(X) \rightarrow \max(\min). \quad (4)$$

Достоинством мультипликативного критерия является то, что при его использовании не требуется нормировка оценок по частным критериям. К недостаткам критерия относится то, что его использование приводит к компенсации малого значения оценки по одному частному критерию избыточной величиной по другому и имеет тенденцию сглаживать уровни оценки по частным критериям за счет их неравнозначных исходных значений.

Метод анализа иерархий [1] (МАИ) – классический и, по-видимому, самый популярный математический инструмент системного подхода к решению проблем принятия решений. МАИ не предписывает лицу, принимающему решение, какого-либо «правильного» решения, а позволяет ему в интерактивном режиме найти такой вариант (альтернативу), который наилучшим образом согласуется с его пониманием сути проблемы и требованиями к ее решению.

Основным достоинством метода анализа иерархий является высокая универсальность – метод может применяться для решения самых разнообразных задач: анализа возможных сценариев развития ситуации, распределения ресурсов, составления рейтинга клиентов, принятия кадровых решений и др.

Недостатком метода анализа иерархий является ограниченность альтернатив и критериев, так как метод затруднителен в применении в режиме реального времени.

Предлагаемый алгоритм оценивания профессиональных достижений сотрудников:

- 1) на базе метода ELECTRE I осуществляется выбор ядра альтернатив по заданным критериям;
- 2) применяются аддитивный и мультипликативный методы, а также метод анализа иерархий на полученной выборке;
- 3) формируется результирующий результат основываясь на данных каждого метода.

Практический пример. Представим ситуацию, что необходимо выбрать человека для решения определенной задачи. Моделирование осуществлялось на реальных данных компании Arrival по показателям: полнота итерации, относительная оценка времени выполнения работы, автономность. Методом ELECTRE I было отобрано ядро альтернатив с коэффициентами согласия 0.7 и несогласия 0.2. Ядро альтернатив состоит из 4х элементов a_1, a_2, a_3, a_4 (табл. 1).

Таблица 1

Первый этап ранжировки. Формирование выборки методом ELECTRE I

Weight	Completeness, avg	Deviation, avg	Independence, avg
1	1	1	1
a1	79	0.746871	70.2
a2	84	0.835025954	57.8
a3	44	1	94.63
a4	75	0.913580247	86.98
a5	60	0.656232	86.12
a6	71	0.459746	55.18
a7	60	0.8613519	61.71
a8	66	0.623813	61.5
a9	59	0.5834677	76.19
a10	70	0.7195323	68.51

Далее на полученной выборке был применен каждый из методов принятия решений и получена ранжировка работников по каждому методу.

Таблица 2

Результаты аддитивного критерия

Место	Альтернатива	Значение
1	a4	0.9085
2	a3	0.8413
3	a2	0.8153
4	a1	0.8097

Таблица 3

Результаты мультипликативного критерия

Место	Альтернатива	Значение
1	a4	53.69
2	a3	37.51
3	a1	37.31
4	a2	36.52

Таблица 4

Результаты метода анализа иерархий

Место	Альтернатива	Значение
1	a4	0.2694
2	a3	0.2492
3	a2	0.2411
4	a1	0.24

Как видно из полученных ранее результатов, альтернативы a1 и a2 в разных методах попадают на разные позиции в ранжировке. Это обусловлено тем, что не существует универсального метода оценивания.

Заключение. В работе представлена актуальная и не вполне решенная на данный момент задача выбора результирующего ранжированного множества на основе нескольких алгоритмов принятия решений с целью применения оценки профессиональных навыков работника. Результаты работы будут основой системой принятия решений системы «Вектор развития».

Библиографический список

1. Саати Т. Л. Принятие решений. Метод анализа иерархий. М.: Радио и связь, 1989. 311 с.
2. Подиновский В. В., Ногин В. Д. Парето-оптимальные решения многокритериальных задач. Новосибирск: Наука, 2007.
3. Самохвалов Ю. Я. Особенности применения метода анализа иерархий при оценке проблем по метрическим критериям // Кибернетика и системный анализ. 2004. № 5. С. 15–19.
4. Колесникова С. И. Системный подход к оцениванию взаимного влияния признаков в тестовом распознавании // Кибернетика и системный анализ. 2009. № 3. С. 127–135.
5. Платформа индивидуального плана развития – «Grow». URL: <https://grow.telescopeai.com/> (дата обращения: 20.05.2020).
6. Система автоматического подбора персонала по навыкам – «E-Staff». URL: <http://www.e-staff.ru/> (дата обращения: 20.05.2020).
7. Система управления и мониторинга работы – «JIRA». URL: <https://www.atlassian.com/software/jira> (дата обращения: 11.05.2020).
8. Веб-сервис для хостинга IT-проектов – «GitHub». URL: <https://logister.su/> (дата обращения: 11.05.2020).
9. Rogers M., Bruen M. Choosing realistic values of indifference, preference and veto thresholds for use with environmental criteria within ELECTRE // European Journal of Operational Research. 1998. P. 542–551.

УДК 574.522+004.942

Д. В. Соболев

студент кафедры компьютерных технологий и программной инженерии

М. Д. Поляк – старший преподаватель – научный руководитель

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ В ЭКОЛОГИИ

Введение

В современном мире одним из важных вопросов для человечества является экология. Появляются различные методы борьбы с загрязнением природы: экологичные вещи, продукты, переработка материалов и так далее. Метод, который может предложить миру математика – это построение математических моделей для прогнозирования и моделирования поведения той или иной экосистемы на планете. Это помогает исследовать, понять и принять какие-то меры в отношении конкретных, точечных проблем.

Например, одной из таких точечных проблем является увеличение цветения в водоемах, как пресных, так и морских. Цветение водорослей также иногда происходит в питьевой воде и прекрасно развивается под толстым ледовым покровом. Но в основном подобный феномен наблюдается в пресных стоячих водоемах, характеризуется болезнетворными микроорганизмами и стойким неприятным запахом. В любом из случаев, вне зависимости от типа водной экосистемы, в которой развивается цветение, вода занимает большую часть нашей планеты и жизненно важна для большинства организмов, обитающих на ней.

Цветение воды происходит из-за развития фитопланктона, которое вызывает изменение цвета воды. Оно появляется из-за быстрого размножения водорослей в водоёме. Окраска воде придается в связи с высокой концентрацией пигментированных клеток. Вода зачастую становится зелёного цвета, хотя и существуют исключения в виде желто-коричневого или красного цвета. Окраска воды тем или иным цветом зависит от вида водорослей. На рис. 1 и рисунке 2 представлены снимки того, как выглядит цветение водорослей, а также сравнение с нормальным функционированием водной экосистемы.

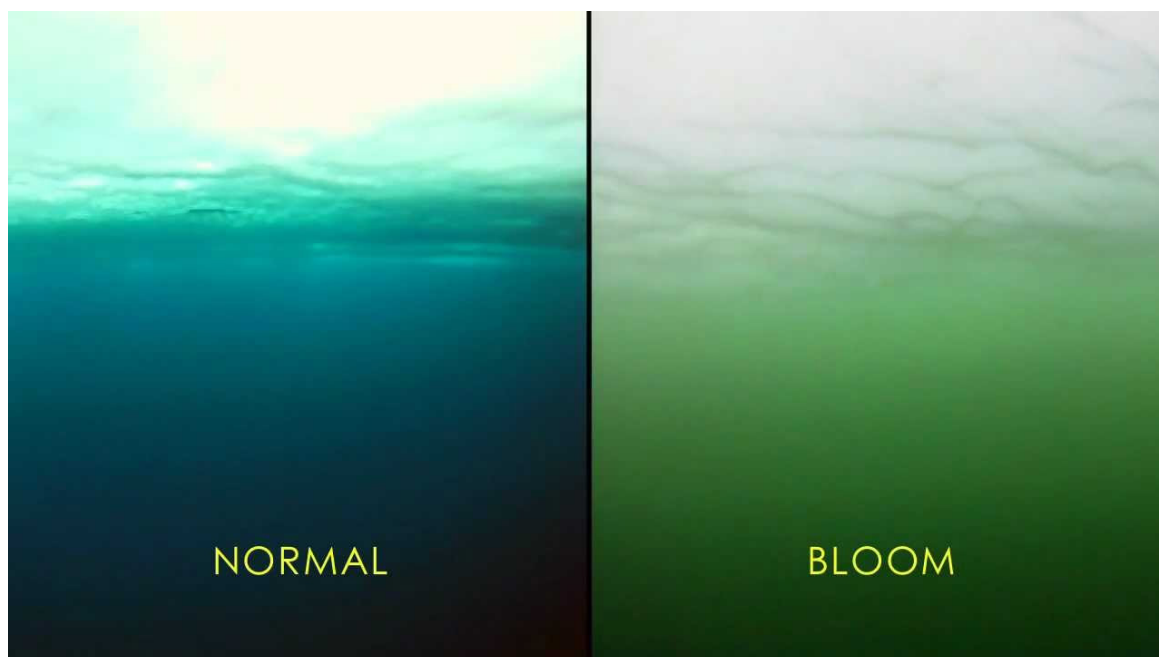


Рис. 1. Сравнение водной экосистемы в норме и водной экосистемы с цветением водорослей.

Фитопланктон является одним из важнейших компонентов водной экосистемы, быстро реагирующим на любые изменения условий окружающей среды. Основными компонентами его развития

служат: свет, углерод, фосфор и азот. Влияет на численность фитопланктона, в основном зоопланктон, который также является звеном пищевой цепи и употребляется многими рыбами и млекопитающими.

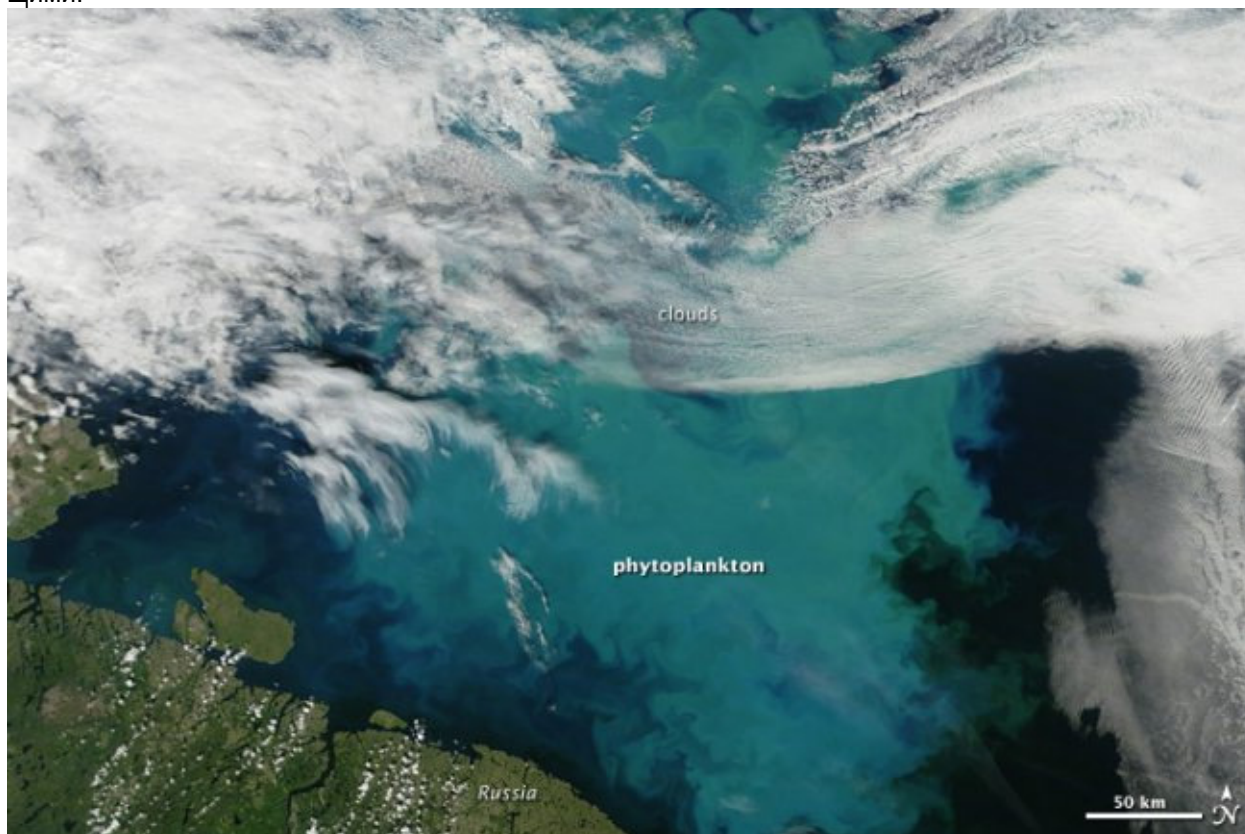


Рис. 2. Цветение водорослей на снимках со спутника

Математическое моделирование

Одним из средств решения проблемы цветения является прогнозирование. Для того, чтобы спрогнозировать развитие фитопланктона в водоемах, следует построить модель. Модель – это копия объекта, которая допускает манипуляции во времени и в пространстве. В целом, модели можно разделить на три класса: биологические, аналоговые и знаковые.

Биологической моделью является модель, которая имеет одни и те же физические, химические или биологические процессы, различен лишь масштаб происходящих явлений. Данные модели имеют свои преимущества, например, достоверность воспроизведения процессов, присущих живому организму, но в то же время, модели являются достаточно громоздкими, поскольку работают в реальном масштабе времени. Такие модели также называют реальными моделями.

Модель называется аналоговой в том случае, когда процессы происходящие в оригинале и в модели, описываются одними и теми же дифференциальными уравнениями [1].

Знаковая модель – это модель, в которой элементы и процессы оригинала описываются в модели абстрактными символами и операциями над ними. Данные модели значительно лучше по своим возможностям, чем биологические или аналоговые модели, так как они почти не связаны ограничениями биологической или физической реализации. Частным случаем таких моделей являются математические модели. Математические модели описывают целый класс процессов или явлений, которые обладают сходными свойствами, или являются изоморфными. Подробнее с описанием определенных типов моделей и примерами моделирования их поведения можно ознакомиться в [2, 3].

В настоящей работе будут рассматриваться только математические модели популяций и сообществ. Это не означает, что аналоговые и, в частности, биологические модели являются не пригодным средством для изучения динамики популяций планктона. Данные виды моделирования имеют свои достоинства, но математические модели обладают наибольшей гибкостью и широтой применения.

Существуют простые математические модели, которые поддаются аналитическому исследованию и обладают свойствами, позволяющими описывать целый спектр природных явлений. Такие модели также называют базовыми. После того, как досконально математически изучена суть процессов на базовой модели и ее модификациях, по аналогии становятся понятными явления, происходящие в гораздо более сложных, реальных системах.

Несмотря на существование огромного разнообразия экосистем, можно выделить некоторые важнейшие присущие им качественные свойства такие как:

- рост
- самоограничение роста
- способность к переключениям
- биоритмы
- пространственная неоднородность
- квазистохастичность

Все эти свойства можно продемонстрировать на сравнительно простых нелинейных динамических моделях. С описанием всех изученных базовых моделей можно ознакомиться в [4].

Для моделирования динамики популяции фитопланктона в данной работе было принято решение использовать модифицированную модель Лотки-Вольтерра – хищник-жертва с питанием. В нашем случае были приняты следующие обозначения:

- жертва – фитопланктон $x_1(t)$,
- хищник – зоопланктон $x_2(t)$,
- питание – азот или фосфор (питание жертвы) $x_3(t)$.

Опишем общий вид системы:

$$\begin{cases} \frac{dx_1}{dt} = f_1(x_1(t), x_2(t), x_3(t)) \\ \frac{dx_2}{dt} = f_2(x_1(t), x_2(t), x_3(t)) \\ \frac{dx_3}{dt} = f_3(x_1(t), x_2(t), x_3(t)) + U(t) \end{cases} \quad (1)$$

Описание частной модели:

Пусть

$$\begin{aligned} x_3(t) &= \alpha_1(t) \\ f_1(x_1(t), x_2(t), x_3(t)) &= \alpha_1(t) x_1 - \beta_1 x_1 x_2 \\ f_2(x_1(t), x_2(t), x_3(t)) &= -\alpha_2 x_2 + \beta_2 x_1 x_2 \\ f_3(x_1(t), x_2(t), x_3(t)) &= 0 \end{aligned}$$

Тогда (1) принимает вид:

$$\begin{cases} \frac{dx_1}{dt} = \alpha_1(t) x_1 - \beta_1 x_1 x_2 \\ \frac{dx_2}{dt} = -\alpha_2 x_2 + \beta_2 x_1 x_2 \\ \frac{d\alpha_1(t)}{dt} = U(t) \end{cases} \quad (2)$$

где $\alpha_2, \beta_1, \beta_2 - const, x_1(t), x_2(t), \alpha_1(t), U(t) = U(x_1(t), x_2(t), \alpha_1(t))$ - некоторые функции времени.

В модели хищник-жертва с питанием ключевой особенностью является то, что она является немарковской, то есть задается зависимость от условий жизни в прошлом. В нашем случае рождаемость жертвы зависит от их питания в течение некоторого времени t . Данная особенность является немаловажной для моделирования системы в данной работе, так как именно от избытка питания зависит развитие фитопланктона в водных экосистемах, что, в свою очередь, вызывает цветение водоемов. Выбор модели Лотки-Вольтерра был обусловлен несколькими преимуществами: данная модель хорошо изучена; по ней имеется большое количество научной литературы; модель не усложнена вычислениями, как, например, модель А.Д. Базыкина [5]; также, она и по сей день является эталонной, поскольку на основании нескольких очевидных предположений объясняет одну из причин периодичности изменения численности популяций.

Постановка задачи управления

Для объекта (1) построить управление $U(t)$, выводящее изображающую точку (ИТ) $(x_1(t), x_2(t), \alpha_1(t))$ системы (2) на целевое многообразие вида:

$$\psi(t) = x_1(t) - x_1^* \xrightarrow[t \rightarrow \infty]{} 0, \tag{3}$$

где x_1^* - заданная постоянная целевая (для $x_1(t)$) величина, доставляющая глобальный минимум функционалу качества управления:

$$\Phi = \int_0^{\infty} ((\psi(t))^2 + \omega^2 \dot{\psi}^2) dt \rightarrow \min \tag{3^*}$$

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

Как известно, основой классического метода АКАР, является следующий факт:

Утверждение 1. Уравнения $\dot{\omega}\psi + \phi = 0$ есть уравнение Эйлера-Лагранжа для функционала:

$$\Phi = \int_0^{\infty} (\phi^2(\psi(t)) + \omega^2 \dot{\psi}^2) dt \rightarrow \min$$

где

- 1) $\phi(\psi)$ – однозначная, непрерывная, дифференцируемая функция для всех ψ ;
- 2) $\phi(0) = 0$;
- 3) $\phi(\psi)\psi > 0, \forall \psi \neq 0$.

Доказательство утверждения следует из результатов теоретической механики [9] для того, чтобы соотношение $\psi(x(t)) = 0$ было инвариантным многообразием системы, необходимо и достаточно, чтобы выполнялось условие:

$$\frac{d\psi}{dt} = \sum_{i=1}^n \frac{\partial \psi(x_1, \dots, x_n)}{\partial x_i} R_i(x_1, \dots, x_n) = \phi(\psi, x_1, \dots, x_n)$$

где $R_i(x_1, \dots, x_n)$ – правая часть i -го уравнения при n -мерном описании исходной системы, $\phi(0, x_1, \dots, x_n) = 0$. Далее непосредственной проверкой можно убедиться, что в задаче (Φ, ψ) с

учетом обозначения $F(t, \psi, \dot{\psi}) = \phi^2(\psi) + \omega^2 \psi^2$ решения уравнение Эйлера-Лагранжа $\frac{\partial F}{\partial \psi} - \frac{d}{dt} \frac{\partial F}{\partial \dot{\psi}} = 2\phi(\psi) \frac{\partial \phi}{\partial \psi} - 2\omega^2 \psi = 0$ являются решениями уравнения $\dot{\omega}\psi + \phi = 0$ и обратно.

Заключение

В процессе исследования были изучены различные литературные источники, описывающие методы математической биофизики и синергетики. Что, в свою очередь, позволило выбрать и построить математическую модель для прогнозирования развития фитопланктона в различных водных экосистемах. На данный момент ведется тестирование программной реализации описанной модели в пакете прикладных программ MATLAB. В ходе этого тестирования планируется получить параметры, дающие устойчивое решение и показательный график динамики развития популяции фитопланктона.

Библиографический список

1. *Ольсон Г.* Динамические аналогии. М.: Гос. изд-во иностранной лит., 1947. 224 с.
2. *Ризниченко Г. Ю.* Лекции по математическим моделям в биологии: учеб. пособие для студ. биол. спец. высш. учеб. заведений. М.; Ижевск: R&C Dynamics: РХД, 2002.
3. *Меншуткин В. В.* Математическое моделирование популяций и сообществ водных животных. Л.: Наука, 1971. 196 с.
4. *Ризниченко Г. Ю.* Математические модели в биофизике и экологии. М.; Ижевск: Ин-т компьютерных исслед., 2003. 184 с.
5. *Базыкин А. Д.* Математическая биофизика взаимодействующих популяций. М.: Наука, 1985.
6. *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. Doi: 10.1134/S0005117918090023.
7. *Князева Е. Н. Курдюмов С. П.* Основания синергетики. СПб.: Алетейя, 2002. 414 с.
8. *Holling C. S.* The Functional Response of Predators to Prey Density and its Role in Mimicry and Population Regulation // Mem. Entomol. Soc. Canada, 1965. № 45.
9. Синергетика и проблемы теории управления: сб. науч. тр. / под ред. А. А. Колесникова. М.: ФИЗМАТЛИТ, 2004.

УДК 004.942

А. А. Щеголева

студент кафедры компьютерных технологий и программной инженерии

М. Д. Поляк – старший преподаватель – научный руководитель

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ БИОЛОГИЧЕСКОЙ СИСТЕМЫ «ХИЩНИК-ЖЕРТВА С ПИТАНИЕМ»

Введение

В окружающем мире есть множество процессов, которые описываются различными моделями. Рассмотрим модель Лотки-Вольтерра, описывающую изменение популяции двух конкурирующих видов. Данную модель можно применить при описании изменения популяции двух видов, изолированных от окружающего мира. Однако она не является в достаточной мере показательной, так как не учитывает изменение количества питания, которое зависит от количества жертв, хищников. Питание в процессе сосуществования хищников и жертв меняется во времени и фактически является функцией.

Модель Лотки-Вольтерра

Созданная в начала 20-го века модель Лотки-Вольтерра [1] описывает изменение популяции двух видов: хищников и жертв. Она имеет вид системы двух дифференциальных уравнений, описанных в формуле (1), где x_1 и x_2 – популяция жертв и хищников соответственно, $\alpha_1, \alpha_2, \beta_1, \beta_2$ – коэффициенты, влияющие на изменение популяций [2]:

$$\begin{cases} \frac{dx_1}{dt} = \alpha_1 x_1 - \beta_1 x_1 x_2 \\ \frac{dx_2}{dt} = -\alpha_2 + \beta_2 x_1 x_2 \end{cases} \quad (1)$$

Решение этой системы представлено на графике (рис. 1) при $\alpha_1 = \frac{2}{3}, \alpha_2 = \frac{4}{3}, \beta_1 = 1, \beta_2 = 1$.

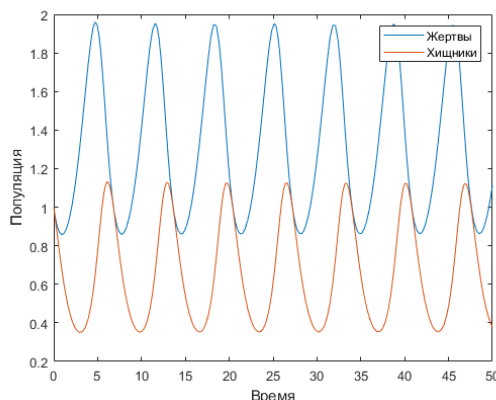


Рис. 1

Система (1) не может быть решена аналитически, поэтому для ее решения используются численные методы. Сравним два способа решения данной системы: методом Рунге-Кутты и методом Эйлера, см. рис. 2. Получим следующие данные о различиях между решениями, вычисленными разными численными методами: $M(x_1) = -0.3189$ (математическое ожидание отличий в решениях для жертвы), $M(x_2) = 0.2731$ (математическое ожидание отличий в решениях для хищника), $\sigma(x_1) = 0.3818$ (среднее квадратическое отклонение отличий в решениях для жертвы),

$\sigma(x_2) = 0.2709$ (среднее квадратическое отклонение отличий в решениях для хищника). Анализируя эти данные, можно сделать вывод, что оба метода имеют примерно одинаковые решения, усредненная величина различий между ними («ошибка») не существенна. В дальнейшем при решении систем будет применяться метод Эйлера.

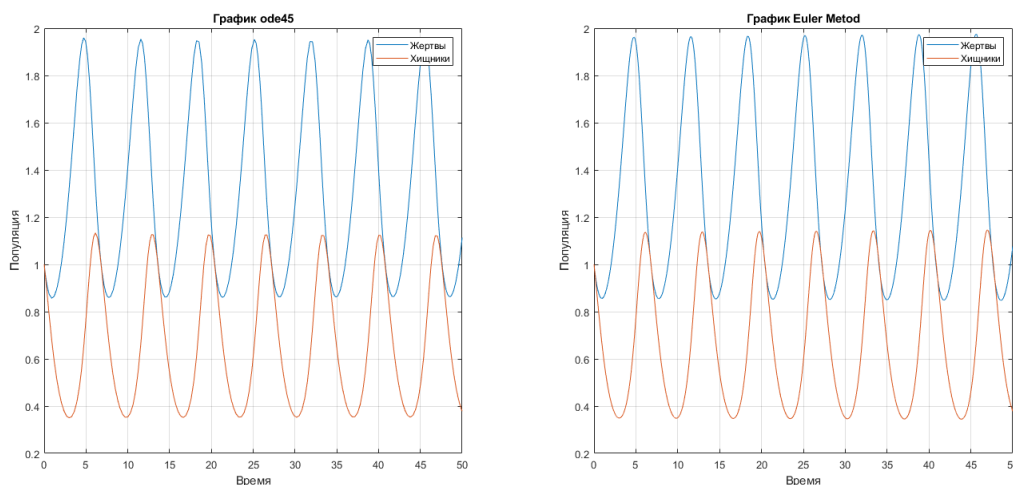


Рис. 2

Система Лотки-Вольтерра с питанием

Чтобы система учитывала изменение питания необходимо ввести соответствующую функцию, меняющуюся с течением времени. Для этого добавим функцию $\alpha_1(t)$, влияющую на изменение количества жертв. Тогда система приобретет следующий вид:

$$\begin{cases} \frac{dx_1}{dt} = \alpha_1(t)x_1 - \beta_1x_1x_2 \\ \frac{dx_2}{dt} = -\alpha_2x_2 + \beta_2x_1x_2 \\ \frac{d\alpha_1(t)}{dt} = U(t) \end{cases}$$

где, $U(t)$ некоторая функция, зависящая от времени, которая будет являться функцией питания. В разные периоды времени популяция жертв будет меняться в зависимости от функции $U(t)$ и постоянного коэффициента β_1 .

Управление питанием в системе

Постановка задачи управления

Чтобы управлять изменением значения функции $U(t)$ введем функцию

$$\psi(t) = x_1 - x_1^* \rightarrow 0, \quad t \rightarrow \infty$$

где x_1^* некоторая переменная, являющаяся целью, к которой стремится x_1 .

Определение структуры АКАР-управления

Введем промежуточную макропеременную:

$$\psi^{(l)}(t) = \alpha_1(t) - \varphi(x_1(t), x_2(t))$$

Тогда

$$U(t) = -\frac{\psi^{(1)}}{T_1} - f_3 + \frac{d\varphi}{dt}, \quad \varphi(x_1, x_2) = \beta_1 x_2 - \frac{1}{T_2} + \frac{x_1^*}{T_2 x_1} \quad \frac{d\varphi(x_1, x_2)}{dt} = -\frac{x_1^*}{T_2 x_1^2} f_1 + \beta_1 f_2$$

Тогда система с включенным питанием примет следующий вид

$$\left\{ \begin{array}{l} \frac{dx_1}{dt} = f_1 = \alpha_1(t)x_1 - \beta_1 x_1 x_2 \\ \frac{dx_2}{dt} = f_2 = -\alpha_2 x_2 + \beta_2 x_1 x_2 \\ \frac{d\alpha_1}{dt} = f_3 + U(t) \\ f_3 = 0 \\ U(t) = -\frac{\psi^{(1)}}{T_1} - f_3 + \frac{d\varphi}{dt} \\ \psi^{(1)}(t) = \alpha_1(t) - \varphi(x_1(t), x_2(t)) \\ \varphi(x_1, x_2) = \beta_1 x_2 - \frac{1}{T_2} + \frac{x_1^*}{T_2 x_1} \\ \frac{d\varphi(x_1, x_2)}{dt} = -\frac{x_1^*}{T_2 x_1^2} f_1 + \beta_1 f_2 \end{array} \right.$$

Решение системы Лотки-Вольтерра с питанием

Для решения системы применим метод Эйлера (ранее описаны возможные отклонения этого метода от более точного метода Рунге-Кутты). Для решения системы введем некоторые ограничения, при которых система будет стабильной: $\varepsilon_1 = 25$ (критерий для устойчивости системы), $\varepsilon_2 = 0.2 * x_1^*$ (критерий для достижения цели). Время существования системы $t = 0 \dots 50$.

Следующей задачей было определение коэффициентов $\alpha_2, \beta_1, \beta_2, T_1, T_2$ и шага h для решения уравнений. В ходе расчетов и проведения тестов было выяснено, что оптимальным является значение шага $h = 0.3$. При более высоких значениях шага система становится менее устойчивой. Последующие графики будут построены с этим шагом.

Далее было определено, что система является стабильной при значениях $T_1 < 1, T_2 > 50$ (при соблюдении критериев устойчивости). Ниже представлены графики, доказывающие это:

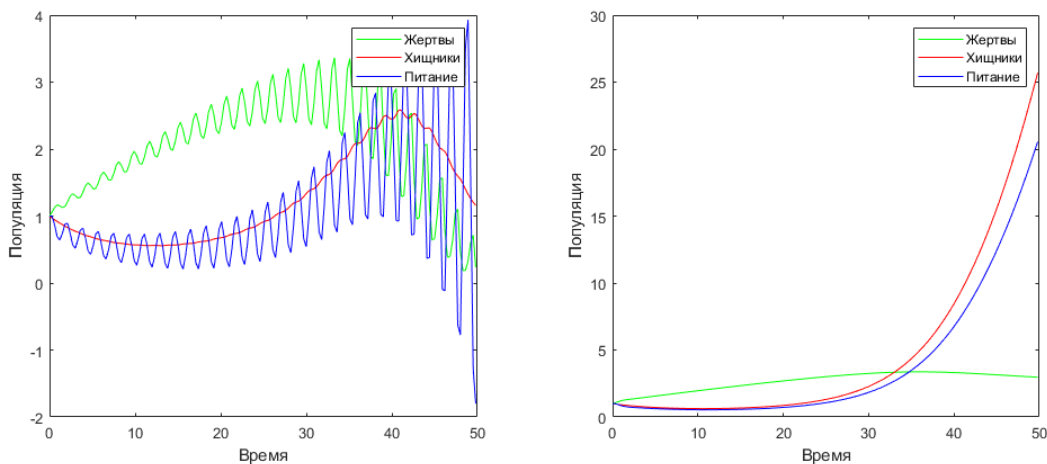


Рис. 3. а) $\alpha_2 = 0.2, \beta_1 = 0.8, \beta_2 = 0.1, T_1 = 0.3, T_2 = 100$, б) $\alpha_2 = 0.2, \beta_1 = 0.8, \beta_2 = 0.1, T_1 = 1.0, T_2 = 100$

При проверке всех значений было выяснено, что наиболее стабильной система является при $T_1 = 0.3$. Если брать более высокие значения, как видно из графика 3.б, система показывает экспоненциальный рост, что недопустимо для биосообществ. Однако при значениях $T_1 = T_2 = 1$ найдены значения коэффициентов ($\alpha_2 = 0.4, \beta_1 = 0.1, \beta_2 = 0.04$), при которых система является стабильной (см. рис. 4).

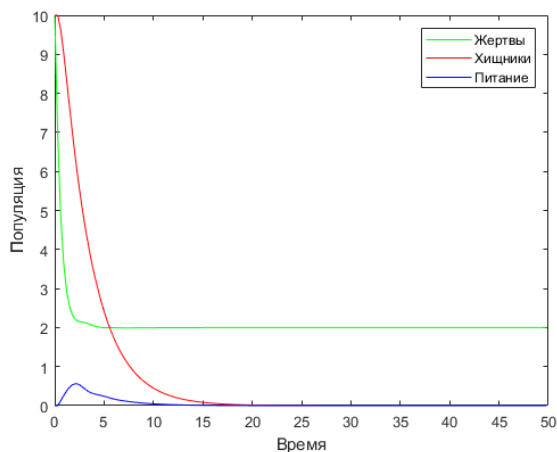


Рис. 4 $\alpha_2 = 0.4, \beta_1 = 0.1, \beta_2 = 0.04, T_1 = T_2 = 1$.

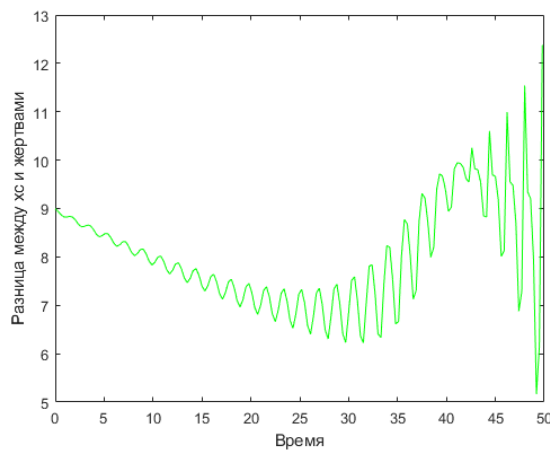


Рис. 5 $\alpha_2 = 0.2, \beta_1 = 0.8, \beta_2 = 0.1, T_1 = 0.3, T_2 = 80$.

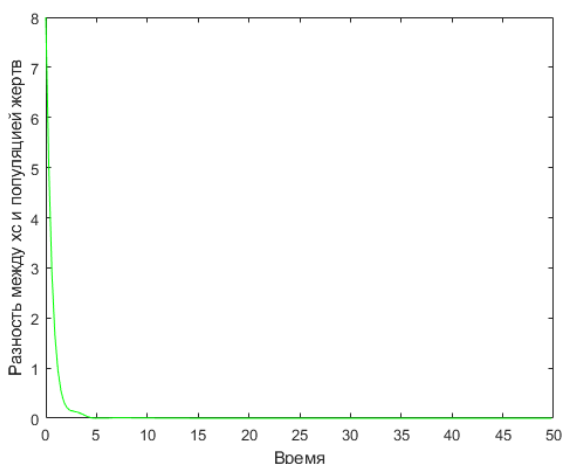


Рис. 6 $\alpha_2 = 0.4, \beta_1 = 0.1, \beta_2 = 0.04, T_1 = T_2 = 1$.

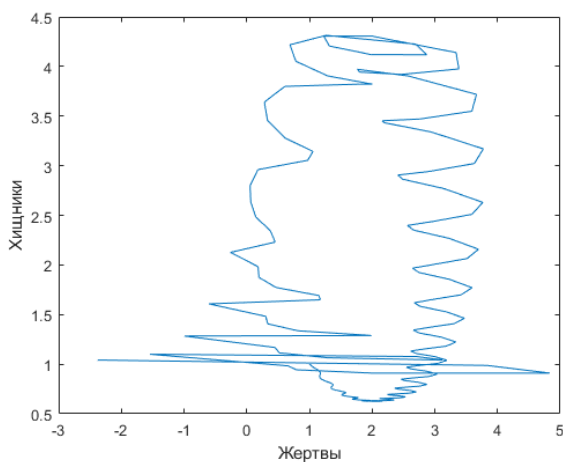


Рис. 7. Фазовый портрет решения системы уравнений «хищник-жертва с питанием» $\alpha_2 = 0.2, \beta_1 = 0.8, \beta_2 = 0.1, T_1 = 0.3, T_2 = 80$

Ранее была введена целевая переменная x_1^* , к этому числу должна стремиться популяция жертв. Для значений $T_1 = T_2 = 1$ возьмем $x_1^* = 2$, на рис. 4 представлен график, на котором видно, что заданная цель достигается, однако популяция хищников и питания к моменту времени 20 становятся равными нулю, что неестественно для биосообществ. При нулевых значениях популяции хищников и количества питания популяция жертв не изменяется, поскольку нет прироста за счет питания и убывания за счет хищников, а естественную смертность данная система не учитывает, и остается равной целевой. Это также видно на рис. 3.б, на графике хищники и питание растут экспоненциально с примерно равными скоростями прироста, таким образом компенсируя друг друга.

Рассмотрим каждый график детально. Возьмем значения $x_1^* = 10$. График на рис. 5 показывает изменение во времени разности между целевой переменной и популяцией жертв. Видно колебательное изменение этого значения при постоянном уменьшении на промежутке от 0 до 30. Однако при

$t > 30$ разность вновь идет вверх, достигая изначальной разности и снова быстро убывает при $t \geq 45$, достигая почти нулевого значения (т.е. цель почти достигнута). Это поведение можно интерпретировать как одно из возможных поведений системы, в том случае, когда цель не достигается. Далее возьмем $x_1^* = 2$. График на рис. 6 показывает изменение во времени разности между целевой переменной и популяцией жертв. Видно, что разность между целевым значением и популяцией жертв экспоненциально уменьшается с течением времени, в итоге становясь равной нулю и оставаясь в таком положении все оставшееся время. Т. е. цель достигается, и популяция сохраняет это значение в течение всего периода времени.

Для дальнейшего анализа построим фазовые портреты для двух пар значений коэффициентов. На графике рис. 7 представлена зависимость популяции хищников от жертв для первой пары коэффициентов. Как видно между ними нет прямой зависимости или какого-то определенного закона. Однако есть некоторая колебательная зависимость.

На рис. 8 представлена зависимость популяции хищников от жертв для второй пары коэффициентов. Как видно зависимость между популяциями имеет гиперболический вид и останавливает рост на значении 10, поскольку популяции жертв и хищников со временем принимают постоянные значения.

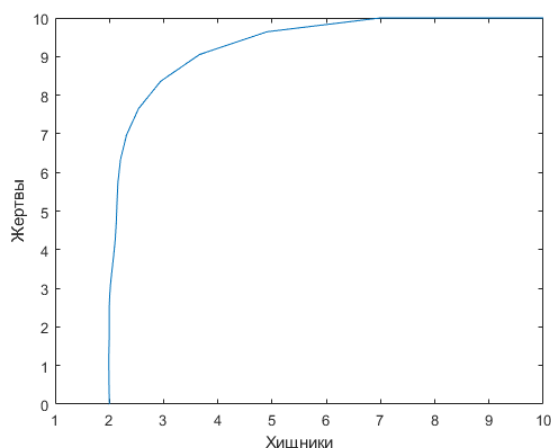


Рис. 8. Фазовый портрет решения системы уравнений «хищник-жертва с питанием»
 $\alpha_2 = 0.4, \beta_1 = 0.1, \beta_2 = 0.04, T_1 = T_2 = 1$

Заключение

В работе сформулирована постановка задачи управления для модели хищник-жертва с питанием. Проведены расчеты и получена новая система дифференциальных уравнений. Однако возникли значительные сложности при решении системы. Во время исследования было получено много новых данных о поведении системы при различных значениях коэффициентов, эти данные необходимо верно интерпретировать и дать математическое объяснение. Были найдены значения коэффициентов, при которых заданная цель достигается, однако популяция хищников и количество питания являются неестественными для биосообществ. Работа по этой теме будет продолжена.

Библиографический список

1. Lotka A. J. Contribution to the Theory of Periodic Reaction // The Journal of Physical Chemistry A. 1910. № 3. P. 271–274.
2. Takeuchi Y. Global Dynamical Properties of Lotka-Volterra Systems. Singapore: World Scientific Pub Co Inc, 1996. 316 p.
3. Kolesnikov A. A., Kolesnikov Al. A., Kuz'menko A. A. The ADAR Method and Theory of Optimal Control in the Problems of Synthesis of Nonlinear Control Systems // Мехатроника, Автоматизация, Управление. 2016. № 10. P. 657–669.
4. Колесникова С. И. Синтез управления нелинейным объектом второго порядка с неполным описанием // Автоматика и телемеханика. 2018. № 9. С. 18–30.

УДК 51-77

В. С. Юрьева

студент кафедры компьютерных технологий и программной инженерии

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

СИНЕРГЕТИЧЕСКИЙ АЛГОРИТМ ОПТИМИЗАЦИИ В ПРОСТЕЙШЕЙ БАЛАНСОВОЙ МОДЕЛИ ЗАПАСОВ В УСЛОВИЯХ СЛУЧАЙНЫХ ПОМЕХ

В статье дано понятие алгоритма метода управления аналитического конструирования агрегированных регуляторов (АКАР), а также реализация метода на примере управления складскими запасами.

Ключевые слова: управление запасами, линейная модель, метод аналитического конструирования агрегированных регуляторов.

Теория управления запасами занимается разработкой методов вычисления объема производства, установлением уровня запасов, необходимых для удовлетворения будущего спроса. Ее важность заключается в том, что рациональное управление запасами и правильное планирование производства позволяют сократить на предприятии издержки, связанные с хранением запасов и потерями из-за неудовлетворенного спроса.

В области теории запасов было проведено немало исследований (например, [1, 2]). Так, в работе [3] рассматриваются управление с учетом транспортных запаздываний и площади склада методом локально-оптимального слежения [1-2]. В статье [4] для решения задачи управления используется теория следящих систем. В данной работе для управления запасами будет рассмотрен классический метод аналитического конструирования агрегированных регуляторов (АКАР) научной школы А.А.Колесникова [5] и его дискретного стохастического расширения [6, 7].

Синергетический подход реализует направленный переход от хаоса (непредсказуемого поведения системы) к управляемому движению вдоль желаемых инвариантных многообразий, к которым будет подстраиваться все другие переменные динамической системы.

Реализации алгоритма управления методом АКАР будет осуществляться на примере управления запасами однопродуктового склада.

Объект управления представим в следующем виде:

$$x(t+1) = x(t) + u(t) - s(t+1) \quad (1)$$

$$s(t+1) = as(t) + b + c\xi(t) + \xi(t+1)$$

где $x(t)$ — количество товаров на складе в момент времени t ; $u(t)$ — объем поставок в момент t ; $s(t)$ — величина спроса в момент t ; a, b, c — параметры модели; $\xi(t)$ — нормальный случайный процесс с нулевым средним и дисперсией равной σ^2 .

Уравнения (1) описывают динамику количества товаров на складе и спроса. Для описания стохастического характера спроса был введен случайный процесс. Для объекта (1) ставится задача нахождения закона управления, обеспечивающего максимально желаемый уровень запасов на складе. Задается целевое многообразие в виде:

$$\psi(t) = x(t) - x^* \rightarrow 0, t \rightarrow \infty \quad (2)$$

где $\psi(t)$ — целевая макропеременная; x^* — максимально заданный целевой уровень количества запасов (рис.1).

Итоговое выражение для управления запасами согласно стохастическому методу АКАР [6, 7]:

$$u^{As}(t) = -(\omega + c + 1)\psi(t) + as(t) + b - c\omega\psi(t-1) \quad (3)$$

Алгоритм вывода управление (3) осуществлялся согласно схеме, представленной на рис. 1.



Рис. 1. Алгоритм вывода закона управления по методу АКАР

Также для уменьшения транспортных затрат и учета грузоподъемности транспортного средства предполагается, что поставки осуществлялись только при снижении количества товаров на складе до заданного значения:

$$u(t) = \begin{cases} 0, & \text{если } u^{As}(t) < U_{\min} \\ u^{As}(t), & \text{если } U_{\min} \leq u^{As}(t) \leq U_{\max} \\ U_{\max}, & \text{если } u^{As}(t) > U_{\max} \end{cases} \quad (4)$$

где $u(t)$ — объём реальных поставок с учетом ограничений; U_{\min} — минимальное значение, введенное для уменьшения затрат, связанных с транспортировкой; U_{\max} — максимальная грузоподъемность транспорта.

Моделирование системы управления запасами по методу АКАР осуществлялось в MATLABc параметрами $a = 0,9$, $a = 1,2$, $c = 0,2$, $\omega = 0,6$, $\sigma = 5$, $z = 180$, $U_{\min} = 180$, $U_{\max} = 200$. В качестве параметров случайного процесса использованы значения математического ожидания $m = 0$ и среднеквадратического отклонения $\sigma = 5$. В качестве начальных условий количества товаров и спроса были выбраны значения $x(0) = 160$ и $s(0) = 5$.

В результате моделирования были получены графики количества товаров на рис. 2, спроса на рис. 3, размера поставок с ограничениями на рис. 4. Как следует из графика на рис. 4, поставки за 100 временных тактов осуществлялись 7 раз.

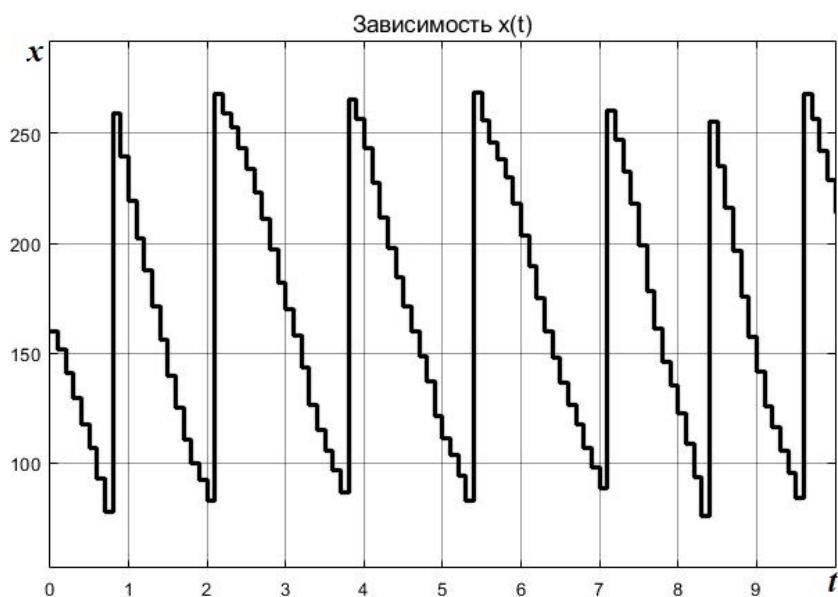


Рис. 2. График зависимости количества товаров от времени

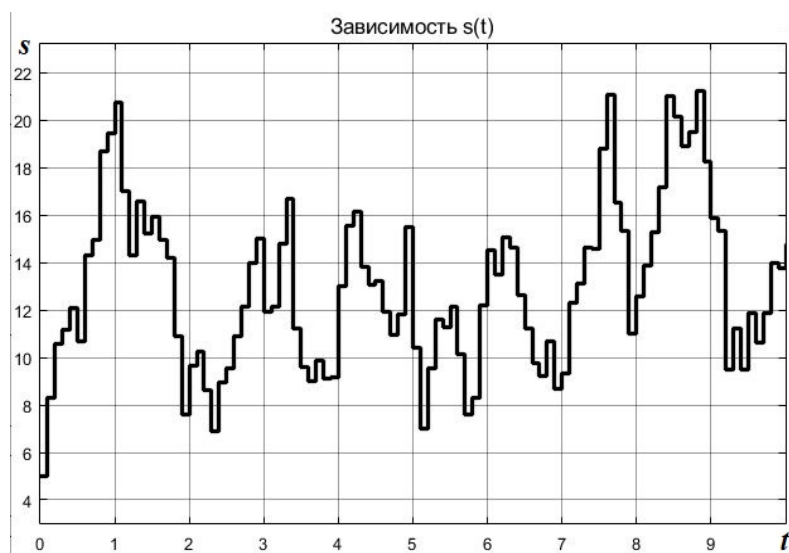


Рис. 3. График зависимости спроса от времени

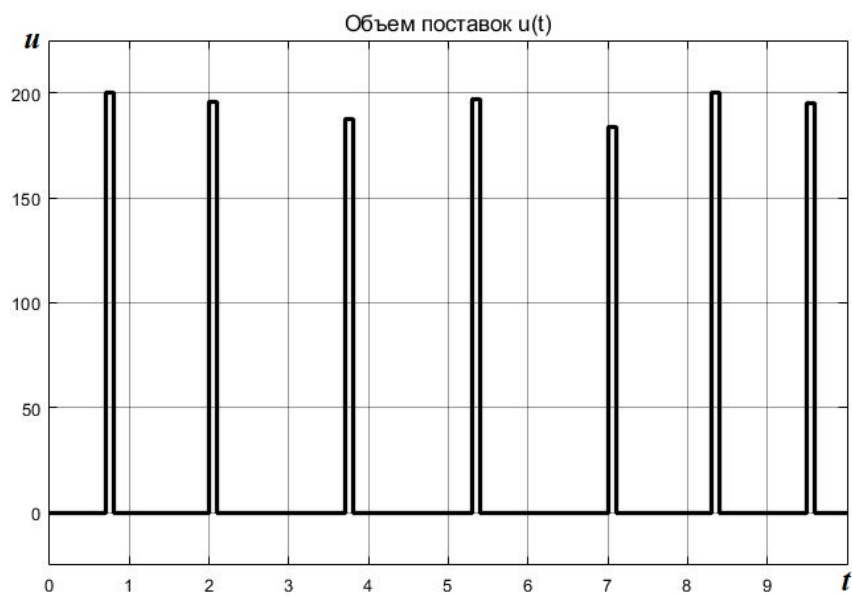


Рис. 4. График зависимости объема поставок от времени

Таким образом, был реализован алгоритм управления запасами методом АКАР на примере однопродуктового склада с целью дальнейшего усложнения модели для исследования более сложного ряда задач.

Библиографический список

1. *Zipkin P. H.* Foundations of inventory management. Burr Ridge, IL, Irwin, 2000.
2. *Simchi-Levi D.* Stochastic Inventory Models // The Logic of Logistics. Springer Series in Operations Research and Financial Engineering / D. Simchi-Levi, X. Chen, J. Bramel. New York: Springer, 2014.
3. *Smagin V. I., Paraev Yu. I.* Synthesis of servo control systems by quadratic criteria / Publishing house Tom. University. Tomsk, 1996.
4. *Саймон Г. А.* О применении теории следящих систем для изучения процессов регулирования производства // Процессы регулирования в моделях эконом. систем. 1961. С. 290.
4. *Колесников А. А.* Основы синергетической теории управления. М.: ИСПО-Сервис, 2000.
5. *Kolesnikova S. I.* A multiple-control system for nonlinear discrete object under uncertainty. Optimization Methods and Software. Doi: 10.1080/10556788.2018.1472258, 2018.
6. *Kolesnikova S. I.* Stochastic Discrete Nonlinear Control System For Minimum Dispersion of the Output Variable // Cybernetics and Automation Control Theory Methods in Intelligent Algorithms. Springer Nature Switzerland AG. CSOC. 2019. P. 1–7.

УДК 004.921, 004.77

А. А. Антипова

студент кафедры вычислительных систем и сетей

Н. Н. Решетникова – кандидат технических наук, доцент – научный руководитель**ИНТЕРАКТИВНАЯ 3D-МОДЕЛЬ ЭКСПОНАТА ИЗ АРХЕОЛОГИЧЕСКОЙ КОЛЛЕКЦИИ**

С незапамятных времен люди увлекаются изучением истории. Они стараются составлять полную картину жизни предков, восстанавливая не только события минувших лет, но и изучая быт и культуру народов. В этом им помогают раскопки, производимые археологами. Музеи проводят выставки, где презентуются экспонаты, или же оставляют их на постоянную экспозицию. Одни и те же предметы не могут находиться в нескольких местах одновременно, чтобы с ними ознакомилось как можно большее количество людей, а выставочные залы собирают большие очереди, отстаивать которые приходится достаточно долго. Если же экспонаты перевозят, есть шанс того, что в процессе транспортировки или же непосредственно на выставке их могут повредить, тем самым увеличив затраты на дальнейшее их содержание и реставрацию. В двадцатом веке, где компьютерные технологии стали неотъемлемой частью жизни, появляются новые средства для демонстрации экспонатов, найденных в археологических экспедициях. С помощью компьютерного моделирования и 3D сканеров становится реальным получить образы экспонатов в цифровом виде, и даже доработать их, если некоторые части были утрачены. Следует отметить, что разработка интерактивных 3D моделей не только позволит сохранить или восстановить экспонаты, но также позволит ознакомиться с ними посетителям в режиме онлайн, не покидая дома, что становится актуальным в наши дни.

Трехмерное представление экспонатов за последние годы активно применяется в археологии, так как является наглядным для человеческого восприятия, а интерактивность делает такое представление наиболее полным для аналитической работы, ведь появляется возможность не только рассматривать предмет со всех сторон, но и взаимодействовать с ним. Кроме того, это более эффективный метод изучения различных аспектов, когда освоенной теории и воображения недостаточно для полного восприятия.

В данной работе рассмотрены два метода получения 3D моделей археологических экспонатов: с помощью 3D сканера, а также создание модели в 3D редакторе на основе фотографии, выявлены преимущества и недостатки каждого рассмотренного метода. Приведены иллюстрации предмета из археологической коллекции и результаты создания его 3D модели.

Понятие реконструкции часто используют как синоним слова – «восстановление», так как одно из определений гласит, что реконструкция – восстановление чего-нибудь по сохранившимся остаткам, описаниям [1]. Несмотря на то, что основной материал для её проведения добывают археологи, с помощью компьютерных технологий возможно вывести этот процесс на новый уровень. В современном мире появилось понятие трехмерной реконструкции, которое обозначает процесс получения формы и облика реальных объектов. Оно может выполняться как проектированием вручную человеком, так и с помощью специального оборудования, такого как сканеры, камеры или другие методы трехмерного сканирования.

Проектирование 3D объекта предполагает использование специализированных программ для разработки необходимых 3D объектов и сцен, таких как Blender или 3Ds Max и выполняется вручную человеком. При создании модели проектировщик сам разрабатывает его, основываясь на собственном восприятии объекта, прямое взаимодействие с объектом в большинстве случаев отсутствует. Этот метод позволяет получить как конкретную модель, так и различные абстракции, которые существуют в воображении человека, дорабатывающего образ объект (что не является лишним, если его часть по факту отсутствует). Можно выделить следующие этапы построения 3D-модели:

1. Построение полигональной модели, называемой каркасом. Достигается путем восстановления геометрической формы и размеров объекта моделирования;

2. Выбор составляющего материала. Например, применение к ювелирному изделию текстуры золота;
3. Настройка освещения;
4. Настройка сцены: расположение камер для наблюдателя.

3D реконструкция, полученная с помощью набора изображений, полученных с камеры, предполагает использование последовательного ряда изображений объекта с разных сторон, при этом требуемый процент наложения двух смежных кадров должен превышать 50 %, а минимальное количество перекрывающихся кадров должно быть равно трем. При грамотном выполнении этих условий можно получить достаточно качественный образ, с последующим редактированием лишь размера объекта. Схема получения модели объекта с помощью съемки двумя камерами представлена на рис. 1, где P – точка пространства, проецируемая на две камеры P_L и P_R соответственно, O_L и O_R – их центры.

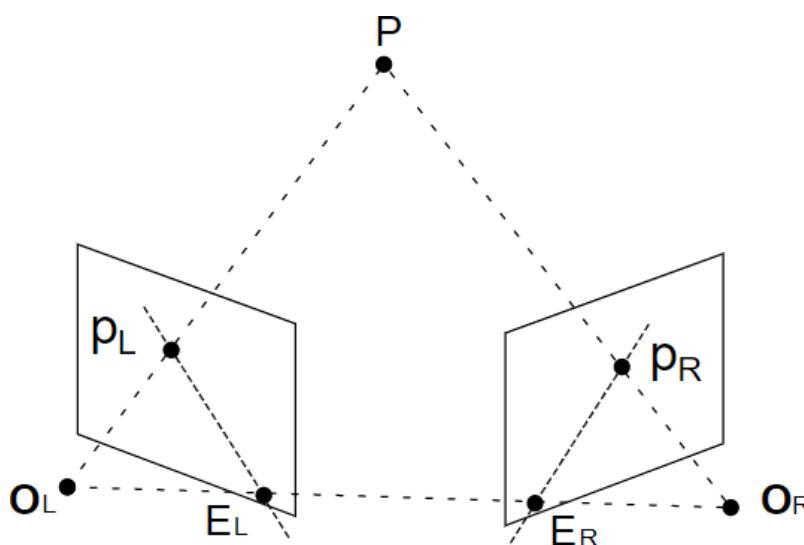


Рис. 1. Общая схема съемки объекта

Алгоритм работы приведенного метода можно представить следующим перечнем этапов:

1. Съемка исследуемого объекта и импорт полученных изображений;
2. Поиск особых точек и решение системы уравнений, полученной на основании множества данных точек;
3. Определение параметров камеры;
4. Поиск «одинаковых» точек на различных наборах смежных изображений объекта;
5. Вычисление координат точек относительно «базового» изображения объекта;
6. Приведение точек к системе координат, наиболее удобной для анализа объекта и наложения структуры [2].

3D-сканер относится к типу устройств, анализирующих объект при построении его 3D модели. Процедура 3D-сканирования сложнее, чем обычная фотосъемка. По методу сканеры делятся на два типа: контактные и бесконтактные. Первый предполагает использование специального щупа, который исследует точки выбранные оператором. Точность получаемой информации зависит от датчиков, расположенных на щупе, которые схожи с датчиками оптико-механической мыши. Данный тип сканеров предполагает работу с объектами небольшого размера и простой формы. Бесконтактные 3D-сканеры имеют более сложную структуру и оперируют ультразвуком или различными лазерными датчиками и фотокамерами: происходит комбинирование данных самих изображений и результатов, полученных от других используемых датчиков.

Можно отметить, что создание модели в 3D редакторе вручную, позволяет достроить по их описаниям утраченные части найденных археологами предметов, а также с высокой точностью моделировать малейшие детали. Человек обладает воображением, что добавляет ему преимущество перед

машиной. Существенный недостаток моделирования заключается в том, что для того, чтобы достигнуть хороших и достоверных результатов требуется много времени.

Изучим объект, с которым будем работать далее. На рис. 2 изображена скифская бронзовая бляха из кургана Кулаковского. Скифский памятник позднеархаического времени в Крыму курган, раскопанный профессором Киевского университета Ю. А. Кулаковским в бывшем Симферопольском уезде в 1895 г., содержал погребение скифского воина. Найденные в погребении уникальные бронзовые предметы, выполненные в скифском зверином стиле, практически сразу были введены в научный оборот.

Кулаковская бляха, была отлита в бронзе на основе формы, сделанной из восковой пластины. Мастер изготавливал её с помощью техники выдавливания, а с помощью резьбы вырезал на лицевой стороне все детали изображения хищников, так как мастера скифского времени понимали, что воск можно не только лепить как глину, но и резать как дерево, а пластичные свойства воска использовать для сгибания вырезанных деталей [3]. Эта вещь уникальна, так как методы, применяемые для её изготовления, являлись на тот момент новаторскими и являлись прорывными не только в ремесле, но и в искусстве.



Рис. 2. Скифская бронзовая бляха из кургана Кулаковского

Процесс создания модели археологического предмета в 3D редакторе на основе фотографии начнем с отрисовки эскиза (рис. 3а), а затем переведем его в программе Inkscape в вектор с помощью кривых Безье для дальнейшей работы (рис. 3б). Так как некоторые части бляхи на фотографии затемнены и трудноразличимы, было принято решение их доработать.



Рис. 3а. Эскиз бляхи Кулаковского

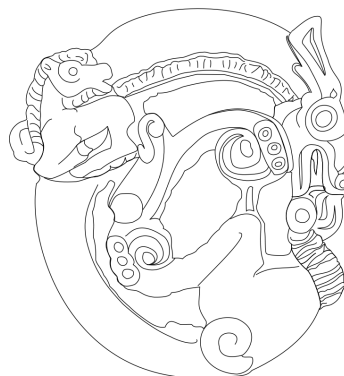


Рис. 3б. Векторное представление

Далее импортируем векторное изображение в формате *.dxf для работы с ним в программе 3Ds Max 2018. Получаем линии в виде Editable Spline (редактируемые сплайны). Начнем разработку с основных частей бляхи, для этого к каждому сплайну из группы Editable Spline применяем модификатор Extrude, затем для деталей применяем модификатор Lattice. Далее берем примитив Box, ковертируем в редактируемые полигоны и во вкладке Modify с помощью Paint Deformation делаем рельеф и накладываем его сверху. Заготовка модели приведена на рис. 4.



Рис. 4. 3D модель бляхи Кулаковского

Рассмотрим отсканированную модель этого же объекта (рис. 5). При её получении использовался 3D-сканер Artec Space Spider, характеристики представлены в табл. 1 [4].

Таблица 1

Характеристики 3D-сканера Artec Space Spider

Разрешение камер	1.3 Мп
Возможность захвата текстур и передачи цвета	Есть
3D разрешение	0.5 мм
Точность сканирования	В зависимости о расстояния, до 0.03 % на 100 см
Скорость сканирования	1 000 000 точек в секунду
Площадь сканирования	от 90x70 мм до 180x140 мм
Рабочее расстояние до объекта	от 90x70 мм до 180x140 мм
Угол триангуляции (поле видимости камер)	30x21 градусов
Технология многоядерной обработки данных	Предусмотрена
Время экспонирования	0.0005 с
Частота видеосъёмки	до 7.5 кадров в секунду



Рис. 5. Отсканированная модель бляхи

Можно заметить, что даже высокоточный 3D-сканер при обработке поверхности бронзового предмета отразил далеко не все детали и модель получилась нечеткой. В данном случае, можно сделать вывод о преимуществах ручного моделирования при детализации элементов поверхности сложного объекта, в сравнении с результатом, полученным при сканировании.

Пакет Autodesk 3Ds Max включает редактор материалов, который позволяет создать материал с эффектом бронзовой поверхности объекта и применить его к разработанной 3D модели (рис.6)

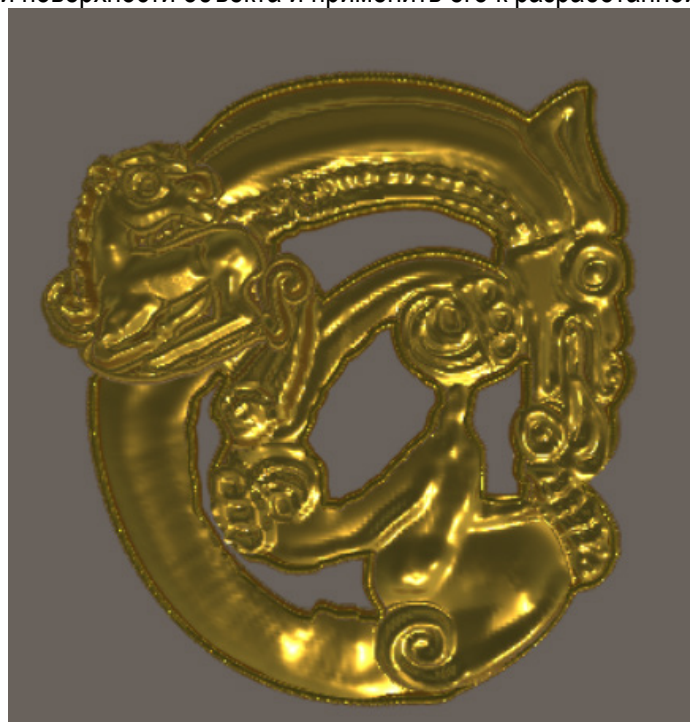


Рис. 6. Итоговая 3D модель скифской бронзовой бляхи Кулаковского

Создание интерактивных 3D-реконструкций отдельных археологических предметов, включающих наряду с фиксацией формы, размеров и текстуры, отдельные семантические фрагменты сложного образа дает возможность подробно рассмотреть и получить информацию о каждом из них. Это

делает разбор композиции по элементам более наглядным, что чрезвычайно важно для оценки исследователем статуса элементов изображения. Поэтому основной задачей при разработке интерактивной 3D модели является возможность взаимодействовать с ней и рассматривать семантические элементы, составляющие сложный образ экспоната со всех возможных сторон и ракурсов. Для выполнения этой задачи созданная 3D модель разбирается на части, что позволяет рассмотреть каждую составляющую по отдельности (рис.7).

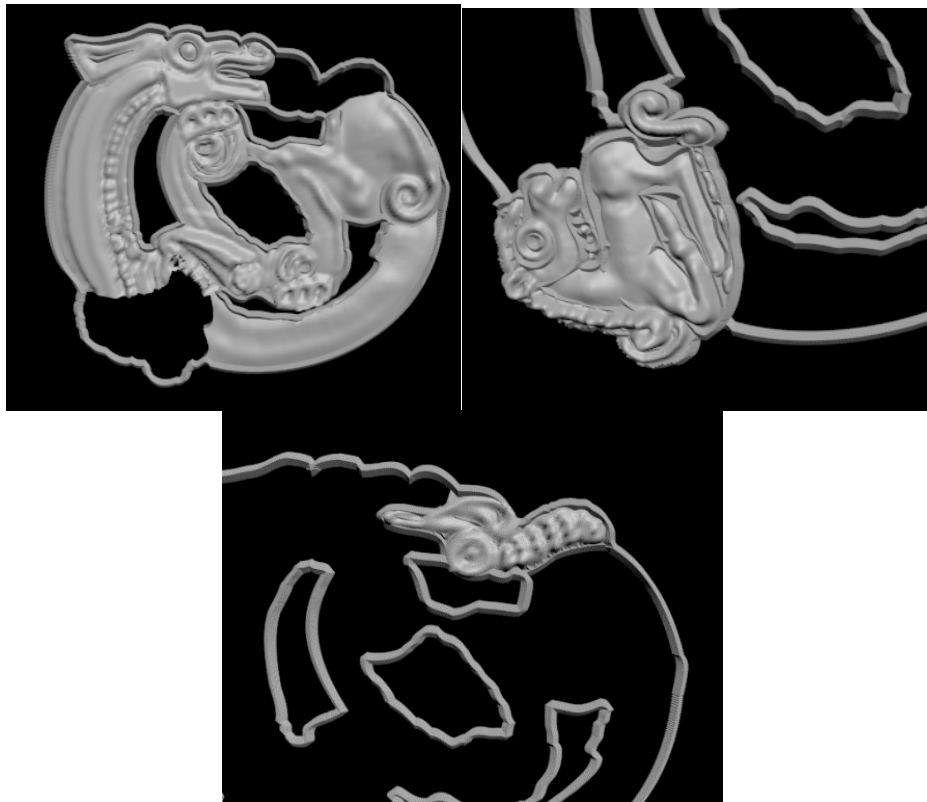


Рис. 7. Отдельные семантические фрагменты 3D модели

В заключение следует отметить, что созданная интерактивная 3D модель может найти применение в современной музейной практике, поскольку существует потребность подачи вспомогательного пояснительного материала с использованием информационных технологий, позволяющих наглядно и доходчиво разъяснить суть экспозиции и представленных на ней артефактов.

Библиографический список

1. Толковый словарь Ожегова. URL: <https://slovarozhegova.ru/word.php?wordid=26937> (дата обращения: 27.03.2020).
2. Лекция проективная геометрия в компьютерном зрении. URL: <https://www.intuit.ru/studies/courses/10622/1106/lecture/18024?page=5> (дата обращения: 28.03.2020).
3. Минасян Р. С. Металлообработка в древности и средневековье. СПб.: Изд-во Государственного Эрмитажа, 2014. 472 с.
4. Официальный сайт Artec 3D. URL: <https://www.artec3d.com/ru/portable-3d-scanners/artec-spider-v2#specifications> (дата обращения: 01.04.2020).

УДК 004.052.2

А. А. Антонов

студент кафедры вычислительных систем и сетей

Н. А. Балонин – доктор технических наук, профессор — научный руководитель

ИССЛЕДОВАНИЕ ГРАФИКОВ АМПЛИТУД ВЕТВЕЙ АТТРАКТОРОВ РИМАНА

Целью данной работы является определение закономерностей и систематизация данных, позволяющих приблизиться к решению гипотезы Римана, на основе изучения графиков Дзета-функции. Актуальность данной задачи велика, поскольку ее решение имеет существенное значение для теоретической и прикладной математики.

Определение: Дзета-функция Римана – это функция от комплексной переменной s , вычисляемая с помощью ряда Дирихле, и являющаяся обобщением ряда Эйлера:

$$\zeta(s) = \frac{1}{1^s} + \frac{1}{2^s} + \frac{1}{3^s} + \dots = \sum_{t=1}^{\infty} t^{-s} = \prod_p \frac{1}{1-p^{-s}} \quad (1)$$

По Эйлеру произведение берется по всем простым числам p . Отрицательные четные числа, такие как $-2, -4, -6, \dots$ называются тривиальными нулями. Также Дзета-функция имеет систему самосопряженных нулей, симметричных относительно прямой $s = a + ib$, $a = 1/2$. Согласно гипотезе Римана, мнимая часть всех нетривиальных нулей Дзета-функции равна $0,5$. Поэтому, если гипотеза верна, бесконечное число нетривиальных нулей лежат на критической прямой $\text{Re } s = 0,5$, которая состоит из комплексных чисел $0,5 + ib$ [1].

Первой итерацией при решении поставленной задачи является понимание графического представления Дзета-функции. Трехмерные графики функции при a равном $0,2, 0,3, 0,5$ и b в диапазоне от 0 до 200 с шагом $0,1$ изображены на рис. 1.

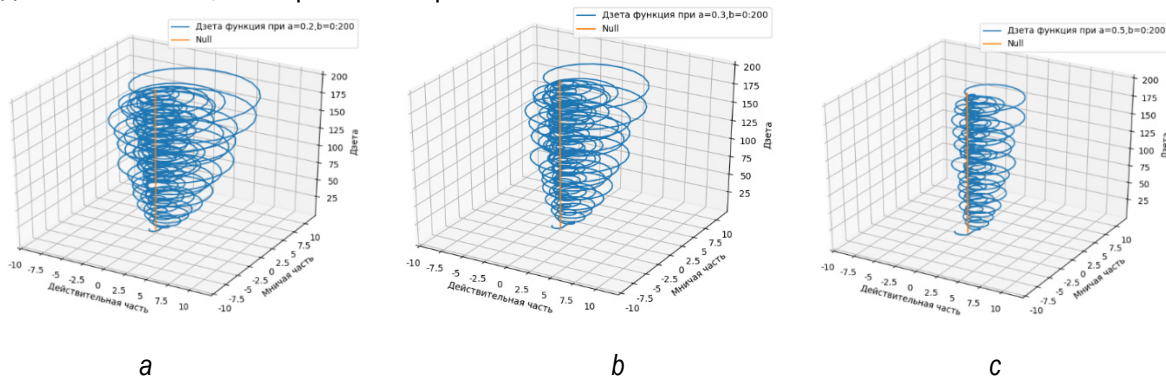


Рис. 1. Графики Дзета-функции при $a = 0,2$ (a), $a = 0,3$ (b), $a = 0,5$ (c)

Из приведенного рисунка видно, что витки аттрактора наиболее прилегают к нулевой оси при $a = 0,5$.

Место прилегания ветви аттрактора к нулевой оси является нулем Дзета-функции. Для определения областей нулей функции используем приведенный на рис. 2 график зависимости расстояний ветвей до аттрактора при b от 0 до 200 и различных a .

Анализ графика показывает, что:

- область первого нуля расположена примерно между 8 и 15 единицами b , область второго нуля от 15 до 24 и т.д. В области $b = 2$ расположен недоноль.
- при увеличении a , расстояние до аттрактора не уменьшается.

Для более детального разбора приведем график зависимости минимальных расстояний при a от $-0,5$ до $1,5$ в разных диапазонах b . Эта зависимость изображена на рис. 3.

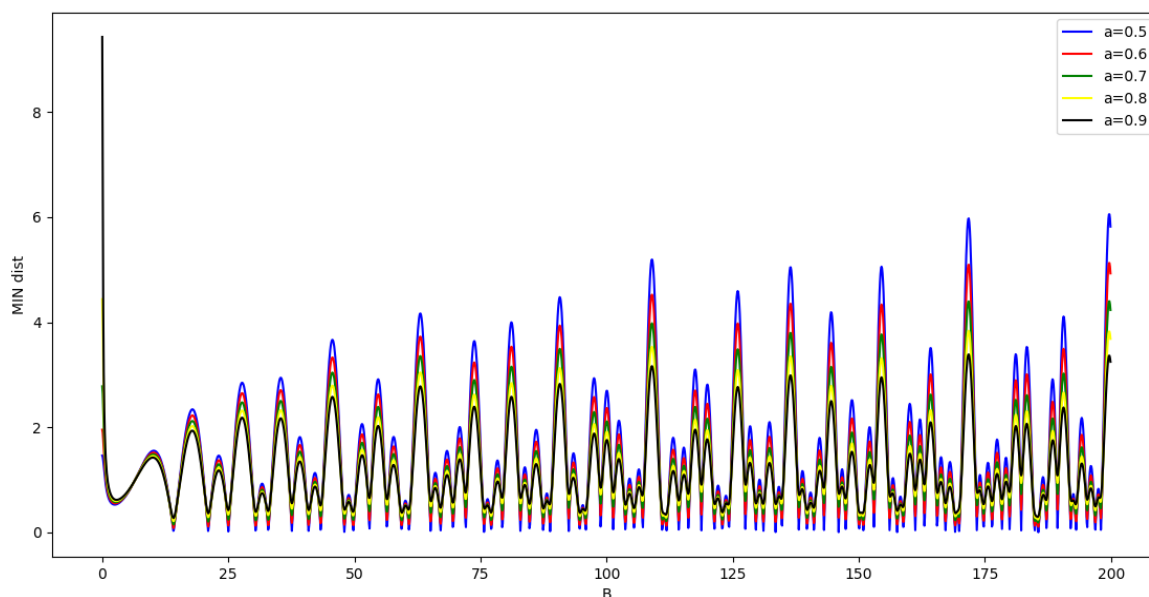


Рис. 2. График расстояний ветвей аттракторов до нулевой оси

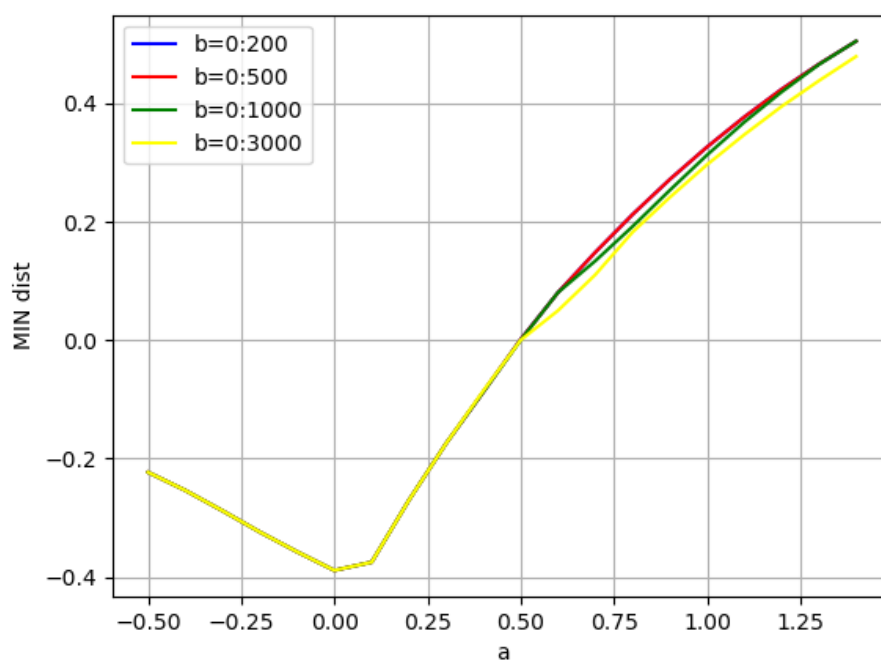


Рис. 3. График минимальных расстояний между ветвями аттракторов и нулевой осью

На рис. представлены графики для трех диапазонов b (от 0 до 200, от 0 до 500, от 0 до 1000) и диапазона от 0 до 3000, из которых видно, что при увеличении диапазона вид графика почти не изменился. Отсюда можно сделать вывод, что на его вид влияют не последующие нули, а нули в начальных областях [2].

Построим графики минимумов в области первого нуля и в области 0 до 200, который приведен на рис. 4. На этом рисунке видно, что в конкретный момент графики полностью накладываются, следовательно, первый минимум сохраняет лидерство на всем пути при a больше 0.1 [3]. Основываясь на проведенных исследованиях можно сделать вывод, что минимальное расстояние, сохраняемое на всем участке, зависит от выбранной области b .

Все исследование и построение графиков проведено с использованием высокоуровневого языка Python. Из-за низкой скорости вычислений, графики были построены с невысокой точностью, но этого достаточно, чтобы решить поставленную задачу.

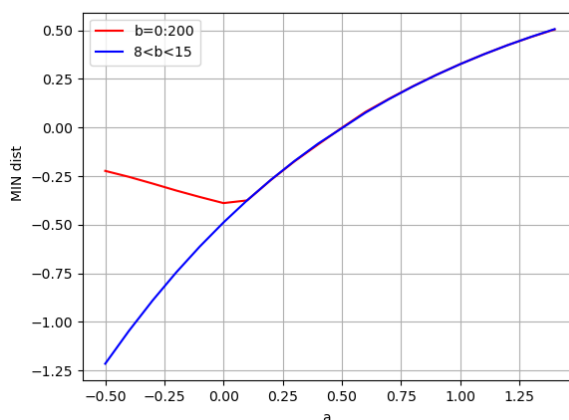


Рис. 4. Влияние первого нуля на график минимумов при b от 0 до 200

Постановщиками математического соревнования (Mathematical worldskills, 2020) программы исследований ГУАП являются профессор Н. А. Балонин, профессор М. Б. Сергеев от ГУАП и заслуженный профессор Дженифер Себерри (Emeritus Professor Jennifer Seberry) от University of Wollongong.

Работа выполнена при поддержке Минобрнауки РФ при проведении научно-исследовательской работы в рамках проектной части государственного задания в сфере научной деятельности по заданию № 2.2200.2017/4.6.

Библиографический список

1. Там Д. О нулях Дзета-Функции $\zeta(S)$, лежащих на почти всех коротких промежутках критической // Чебышев. сб. 2016. № 17. С. 71–89.
2. Балонин Н. А. Облачные вычисления «MATHSCINET.RU». URL: <http://www.mathscinet.ru> (дата обращения: 10.04.2020).
3. Карацуба А. А. Венский доклад: о количестве нулей дзета-функции Римана на коротких промежутках критической прямой // Чебышев. сб. 2015. № 16. С. 19–31.

УДК 004.032.24

О. С. Глотова

студент кафедры вычислительных систем и сетей

А. В. Гордеев – доктор технических наук, профессор – научный руководитель**БАЗОВЫЕ СРЕДСТВА ПАРАЛЛЕЛЬНОГО ПРОГРАММИРОВАНИЯ ОС «ЭЛЬБРУС»
ДЛЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С МИКРОПРОЦЕССОРОМ «ЭЛЬБРУС-4С»**

Многопроцессорные и многоядерные компьютеры актуальны для параллельных задач, так как одновременно на каждом процессоре или ядре в один и тот же момент времени могут выполняться сразу несколько операций по обработке данных. Это полезно в случае, если разделить алгоритм по выполнению задачи на независимые части и распределить их вычисление по процессорам (или ядрам). Или же применить один и тот же алгоритм к большому числу однотипных данных, распределив их параллельное выполнение по доступным ресурсам, описанным выше.

Микропроцессор «Эльбрус-4С» (в ходе работы над проектом использовалось рабочее название «Эльбрус-2S») – многоядерный универсальный высокопроизводительный микропроцессор, который содержит 4 вычислительных ядра, 3 канала межпроцессорного обмена и канал ввода-вывода, 3 контроллера памяти и кэш-память 2-го уровня общим объемом 8 Мегабайт. Рабочая тактовая частота составляет 800 МГц, кристалл выполнен по технологической норме 65 нм [1]. Данный микропроцессор построен в соответствии с архитектурой «Эльбрус».

Принципиальной особенностью архитектуры «Эльбрус», в отличие от традиционных архитектур типа RISC или CISC (x86, SPARC, ARM), является возможность при компиляции каждого фрагмента программы предопределить максимальное распараллеливание вычислительного процесса по всем доступным аппаратным ресурсам, которая базируется на использовании широкого командного слова. То есть, архитектуру «Эльбрус» можно отнести к категории VLIW (Very Long Instruction Word). Широкое командное слово содержит набор операций (с их адресными, литеральными и функциональными параметрами), которые одновременно дешифруются и параллельно выполняются, каждая в своем отдельном конвейере. Это принципиальный фактор реализации параллелизма, свойственного данному программному коду. Микропроцессор «Эльбрус» имеет шесть каналов для выполнения арифметико-логических операций. Помимо них в одном широком командном слове могут быть заданы операции и других типов с фиксированными временами выполнения, что позволяет статически, во время трансляции, планировать параллельную работу исполнительных устройств [2].

В архитектурах RISC и CISC анализ зависимостей, планирование, распределение регистров выполняет аппаратура процессора, а не компилятор. Но динамический анализ зависимостей и поддержка внеочередного исполнения имеет свои ограничения: лучшие современные процессоры способны анализировать и запускать гораздо меньше команд за такт, чем в архитектуре «Эльбрус» (для примера: в рассматриваемом микропроцессоре «Эльбрус-4С» каждое ядро декодирует и отправляет на исполнение до 23 операций за такт), к тому же соответствующие блоки внутри процессора потребляют заметное количество энергии.

В архитектуре «Эльбрус», как уже было сказано ранее, задачи анализа зависимостей между операндами и планирования ресурсов процессора возложены на компилятор. В результате, аппаратура процессора может быть проще и экономичнее. Здесь компилятор способен анализировать исходный код гораздо тщательнее, чем аппаратура RISC/CISC процессора, и находить больше независимых операций. Вот почему в архитектуре «Эльбрус» больше параллельно работающих исполнительных устройств, чем в традиционных архитектурах, и на многих алгоритмах она демонстрирует очень высокую скорость вычислений.

Еще одной особенностью процессоров архитектуры «Эльбрус» являются методы работы с памятью. Кроме наличия кэша, позволяющего оптимизировать время доступа в память, ими поддерживаются методы предварительной подкачки данных, которые позволяют прогнозировать обращения в память и производить подкачку данных в кэш или другое специальное устройство за некоторое время до их использования. Процессоры архитектуры Эльбрус поддерживают программно-аппаратный метод подкачки. Это означает, что аппаратная часть микропроцессора включает в себя специальное

устройство для обращения к массивам (Array Access Unit, AAU), в то время как необходимость подкачки определяется компилятором, генерирующим специальные инструкции для AAU. Использование устройства подкачки эффективнее помещения элементов массива в кэш, поскольку элементы массивов чаще всего обрабатываются последовательно и редко используются более одного раза. В целом, микропроцессоры с архитектурой «Эльбрус» поддерживают несколько видов параллелизма помимо параллелизма на уровне команд: векторный параллелизм, параллелизм потоков управления, параллелизм задач в многомашинном комплексе [3].

В работе изложены результаты проведенных исследований производительности микропроцессора «Эльбрус-4С» на основе результатов выполнения программы, написанной с помощью имеющихся средств для параллельного программирования в ОС «Эльбрус», а также сравнение полученных результатов с последовательной реализацией данной программы.

Было установлено, что в ОС «Эльбрус» для параллельного программирования предусмотрен пакет MPICH – одна из самых первых и наиболее распространенных реализаций стандарта MPI [4]. К тому же, в системе было обнаружено сразу две версии: библиотека MPICH2, основанная на стандарте MPI-2, а также MPICH, базирующаяся на стандарте MPI-3. Для тестирования использовалась версия MPICH с поддержкой стандарта MPI-3.

MPI (Message Passing Interface, интерфейс передачи сообщений) — стандарт интерфейса обмена данными в параллельном программировании, который позволяет обмениваться сообщениями между процессами, выполняющими одну задачу. Используется при разработке программ для кластеров и суперкомпьютеров. Это объясняет, почему в рассматриваемой ОС сразу установлено средство стандарта MPI, ведь данная архитектура разрабатывалась, в том числе, для работы кластеров и суперкомпьютеров.

Однако, в практическом применении следует тщательно анализировать все преимущества, а также и ограничения по производительности, которые могут возникнуть в результате использования MPI в определенной вычислительной среде, и только после этого принимать решение, следует ли использовать средства данного стандарта. Например: в многопроцессорных системах с общей памятью (симметричных системах) предпочтительнее будет вариант многопоточного распараллеливания программ (OpenMP), так как MPI больше ориентирован на работу в вычислительных системах с распределенной памятью, где процессоры работают независимо друг от друга, каждый под управлением своей операционной системы, и нужно распределять вычислительную нагрузку, а также организовать передачу данных между компьютерами (затраты на передачу данных могут быть достаточно большими). То есть, на сервере «Эльбрус-4.4 1У», в составе которого находятся 4 процессора «Эльбрус-4С», каждый из которых имеет по 4 ядра, использование средств MPI уже не будет являться лучшим решением. Однако, если сервер входит в состав кластера, тогда уже можно использовать совместно MPI и средства для многопоточного распараллеливания, чтобы оптимально распределять нагрузку на имеющихся ресурсах.

В данной работе приводятся результаты исследования с использованием алгоритма умножения матрицы на вектор, в параллельной реализации основанного на представлении матрицы горизонтальными полосами строк, так как в задачах распараллеливания с более объемными и трудоемкими вычислениями данный алгоритм может фигурировать достаточно часто. При таком способе разделения данных в качестве базовой подзадачи может быть выбрана операция скалярного умножения одной строки матрицы на вектор.

Главная функция параллельной программы, а также подпрограммы, написанные на языке Си++ в MPICH, имеют вид:

```
int main(int argc, char* argv[]) {
    double start, end;
    int size;
    double* rezultat;
    double* isxMatrica;
    double* isxVector;
    double* rezProc;
    double* strokProc;
```

```

int NumRow;
MPI_Init(&argc, &argv);
MPI_Comm_size(MPI_COMM_WORLD, &NumProc);
MPI_Comm_rank(MPI_COMM_WORLD, &RankProc);

InitProc(isxMatrica, isxVector, rezultat, strokProc, rezProc, Size, NumRow);

RaspredelDannix(isxMatrica, strokProc, isxVector, Size, NumRow);

start = MPI_Wtime();
Ymnozheniye(strokProc, isxVector, rezProc, Size, NumRow);
end = MPI_Wtime();

SborRezult(rezProc, rezultat, Size, NumRow);

MPI_Barrier(MPI_COMM_WORLD);

if (RankProc == 0) {
VivodRezult(isxMatrica, isxVector, rezultat, strokProc, rezProc, Size);
printf(" Vremya vipolneniya progr: %f\n", end – start);
}

MPI_Finalize();

return 0;
}

void InitProc (double* &isxMatrica, double* &isxVector, double* &rezultat, double* &strokProc, double*
&rezProc,
int &Size, int &NumRow) {
int RestRows;
int i;
if (RankProc == 0) {
do {
printf("\nVvedite razmer matrici: ");
scanf(" %d", &Size);
if (Size < NumProc) {
printf("Размер матрицы должен превышать количество процессов! \n ");
}
}
while (Size < NumProc);
}
MPI_Bcast(&Size, 1, MPI_INT, 0, MPI_COMM_WORLD);
RestRows = Size;
for (i=0; i<RankProc; i++)
RestRows = RestRows-RestRows/(NumProc-i);
NumRow = RestRows/(NumProc-RankProc);
isxVector = new double [Size];
rezultat = new double [Size];
strokProc = new double [NumRow*Size];
rezProc = new double [NumRow];
if (RankProc == 0) {

```

```

isxMatrica = new double [Size*Size];
InitDannix(isxMatrica, isxVector, Size);
}
}

```

```

void RaspredeIDannix(double* isxMatrica, double* strokProc,
double* isxVector, int Size, int NumRow) {
int *OtpravlNum;
int *OtpravlInd;
int RestRows=Size;
MPI_Bcast(isxVector, Size, MPI_DOUBLE, 0, MPI_COMM_WORLD);
OtpravlInd = new int [NumProc];
OtpravlNum = new int [NumProc];
NumRow = (Size/NumProc);
OtpravlNum[0] = NumRow*Size;
OtpravlInd[0] = 0;
for (int i=1; i<NumProc; i++) {
RestRows -= NumRow;
NumRow = RestRows/(NumProc-i);
OtpravlNum[i] = NumRow*Size;
OtpravlInd[i] = OtpravlInd[i-1]+OtpravlNum[i-1];
}
MPI_Scatterv(isxMatrica, OtpravlNum, OtpravlInd, MPI_DOUBLE, strokProc,
OtppravlNum[RankProc], MPI_DOUBLE, 0, MPI_COMM_WORLD);
delete [] OtppravlNum;
delete [] OtppravlInd;
}

```

```

void Ymnozheniye(double* strokProc, double* isxVector,
double* rezProc, int Size, int NumRow) {
int i, j;
for (i=0; i<NumRow; i++) {
rezProc[i] = 0;
for (j=0; j<Size; j++)
rezProc[i] += strokProc[i*Size+j]*isxVector[j];
}
}

```

```

void SborRezult(double* rezProc, double* rezultat,
int Size, int NumRow) {
int *PoluchNum;
int *PoluchInd;
int RestRows=Size;
int i;
PoluchNum = new int [NumProc];
PoluchInd = new int [NumProc];
PoluchInd[0] = 0;
PoluchNum[0] = Size/NumProc;
for (i=1; i<NumProc; i++) {
RestRows -= PoluchNum[i-1];
PoluchNum[i] = RestRows/(NumProc-i);
PoluchInd[i] = PoluchInd[i-1]+PoluchNum[i-1];
}
}

```

```

    }
    MPI_Allgather(rezProc, PoluchNum[RankProc], MPI_DOUBLE, rezultat, PoluchNum, PoluchInd,
MPI_DOUBLE, MPI_COMM_WORLD);
    delete [] PoluchNum;
    delete [] PoluchInd;
    }

```

Последовательный алгоритм умножения матрицы на вектор, также написанный на языке Си++, имеет следующий вид:

```

for (int i=0; i<Size; i++){
    Result[i]=0;
    for (int i1=0; i1<Size; i1++){
        Result[i] += Matrix[i][i1]*Vector[i1];
    }
}

```

Также приведены результаты тестов по сравнению времени выполнения вычислений параллельной реализации алгоритма с использованием разного количества процессов, и времени выполнения вычислений последовательной реализации алгоритма. Тестирование параллельной программы проводилось на кластере из двух рабочих станций «Эльбрус 401-РС» на базе процессоров «Эльбрус-4С» под управлением операционной системы «Эльбрус». Результаты тестов приведены в табл. 1.

Таблица 1

Результаты тестирования

Последовательная реализация алгоритма			Параллельная реализация алгоритма				
Объем кода	Размер матрицы	Время, с	Объем кода	Количество процессоров «Эльбрус-4С»	Размер матрицы	Количество процессов	Время, с
66	3	0.000004	211	2	3	2	0.000003
						4	-
						8	-
	6	0.000005			6	2	0.000004
						4	0.000004
						8	-
	9	0.000008			9	2	0.000005
						4	0.000005
						8	0.000004
	12	0.000011			12	2	0.000006
						4	0.000006
						8	0.000003
19	0.000023	19	2	0.000032			
			4	0.000007			
66	19	0.000023	211	2	19	8	0.000004
	26	0.000039			26	2	0.000019

Последовательная реализация алгоритма			Параллельная реализация алгоритма						
Объем кода	Размер матрицы	Время, с	Объем кода	Количество процессоров «Эльбрус-4С»	Размер матрицы	Количество процессов	Время, с		
					33	4	0.000010		
						8	0.000005		
						2	0.000030		
	4	0.000017							
	8	0.000012							
	33	0.000062							
	48	0.000131					48	2	0.000093
								4	0.000035
								8	0.000021

По полученным результатам для большей наглядности составлен график производительности последовательной и параллельной реализаций алгоритма, представленный на рис. 1.

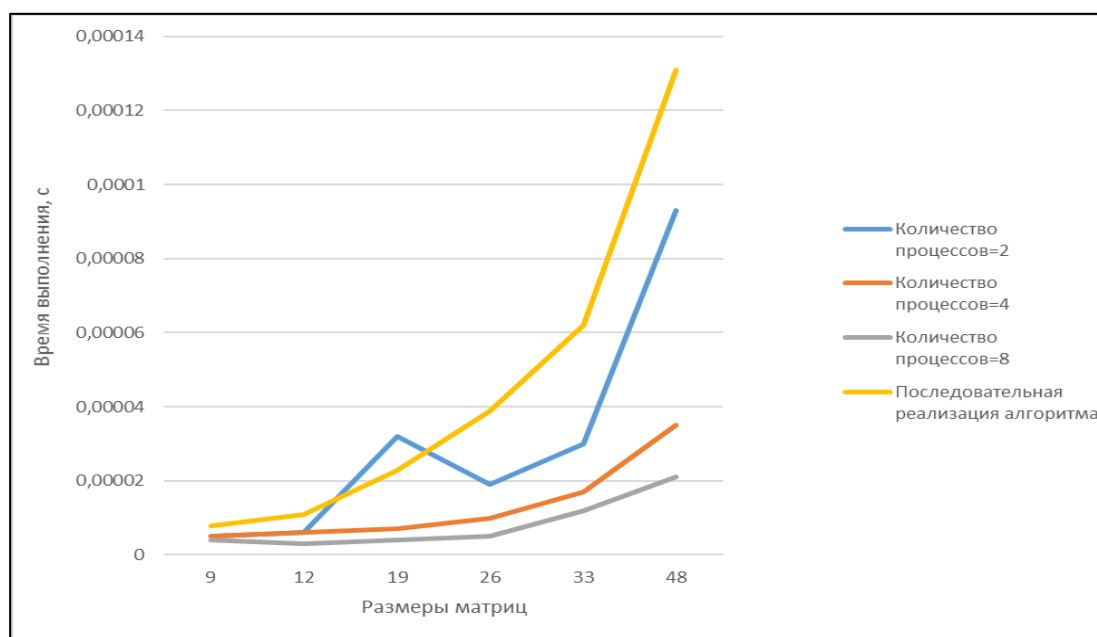


Рис. 1. Результаты тестирования последовательной и параллельной реализации алгоритма для количества процессоров, равного 2, 4 и 8

Заключение по результатам исследования

Как видно на рис. 1, параллельная программа выполняется гораздо быстрее, чем последовательная. При числе процессоров = 8 времени на выполнение алгоритма уходит меньше, чем при числе процессоров = 4, и гораздо меньше, чем при числе процессоров = 2. То есть, можно сделать вывод, что использование MPI в данном кластере с распределенной памятью является верным решением, так как затраты времени на передачу данных не оказывают большого влияния на скорость выполнения задачи, и распределение вычислительной нагрузки также осуществлено достаточно оптимально. Также стоит отметить, что даже не смотря на небольшую тактовую частоту процессора, при грамотном распараллеливании программы с помощью MPI, а также благодаря своей архитектуре, «Эльбрус» демонстрирует существенное сокращение времени выполнения программы. То есть, данная

архитектура оправдывает свою направленность на задание явного параллелизма операций, и можно уверенно сказать, что ключевыми чертами архитектуры «Эльбрус» являются энергоэффективность и высокая производительность.

Библиографический список

1. Официальный сайт коллектива разработчиков микропроцессора «Эльбрус-4С» АО МЦСТ. URL: <http://www.mcst.ru/elbrus-4c> (дата обращения: 09.01.2020).
2. Ким А. К., Перекатов В. И., Ермаков С. Г. Микропроцессоры и вычислительные комплексы семейства «Эльбрус». СПб.: Питер, 2013. 272 с.
3. Лимонова Е. Е., Бочаров Н. А., Парамонов Н. Б. и др. Оценка быстродействия системы распознавания на VLIW архитектуре на примере платформы Эльбрус // Программирование. 2019. № 1. С. 15–21.
4. Официальный сайт MPICH. URL: <https://www.mpich.org> (дата обращения: 09.01.2020).

УДК 004.657

Е. Е. Доминов

магистрант кафедры вычислительных систем и сетей

А. В. Аксенов – старший преподаватель – научный руководитель

НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ БИБЛИОТЕКИ ОБЪЕКТНО-РЕЛЯЦИОННОГО ОТОБРАЖЕНИЯ МОДИФИЦИРОВАННЫМ МЕТОДОМ ТРС-С

При написании приложений на Python, для работы с базами данных часто используются объектно-реляционные мапперы (ORM). Примерами ORM для Python являются SQLAlchemy, PonyORM и объектно-реляционный маппер, входящий в состав Django. При выборе ORM довольно важную роль играет её производительность.

В интернете можно найти большое количество тестов производительности. В основном данные тесты оценивают скорость выполнения различных типов запросов за определенный промежуток времени. В качестве примера таких тестов можно привести бенчмарк от Tortoise ORM [1]. Данный бенчмарк анализирует скорость работы шести ORM для одиннадцати различных видов SQL-запросов. В целом бенчмарк от tortoise хорошо позволяет оценить скорость выполнения запросов при использовании разных ORM, но у такого подхода к тестированию существует одна проблема. ORM зачастую используют в веб приложениях, где одновременно несколько пользователей могут посылать различные запросы, но не было найдено ни одного бенчмарка, оценивающего работу ORM при таких условиях. Вследствие этого было решено написать свой тест, который бы оценивал работу ORM при таком сценарии.

Компания TPC с 1988 года разрабатывает тесты, направленные на обработку данных. Они давно стали индустриальным стандартом и используются почти всеми вендорами оборудования на различных образцах аппаратного и программного обеспечения. Главная особенность этих тестов состоит в том, что они нацелены на тестирование при огромной нагрузке в условиях, максимально приближенных к реальным.

TPC-C симулирует работу сети складов. Он включает в себя комбинацию из одновременно выполняемых транзакций пяти различных типов и сложности. База данных состоит из девяти таблиц с большим количеством записей. Производительность в тесте TPC-C измеряется в транзакциях в минуту.

Описание теста

В написанном мной тесте сначала создается и наполняется база данных, которая представляет из себя базу сети складов. Схема БД представлена на рис. 1

База данных состоит из восьми отношений:

1. Warehouse – склад
2. District – участок склада
3. Order – заказ
4. OrderLine – строка заказа (позиция заказа)
5. Stock – количество определенного товара на определенном складе
6. Item – товар
7. Customer – клиент
8. History – История платежей клиента.

В ходе теста, обрабатываются транзакции, посылаемых одновременно от лица нескольких виртуальных пользователей. Каждая транзакция состоит из нескольких запросов. Всего в данном тесте существует пять видов транзакций, которые подаются на обработку с разной процентной вероятностью:

1. new_order (создание нового заказа) 45 %
2. payment (оплата клиентом заказа) 43 %
3. order_status (возвращает состояние последнего заказа клиента) 4 %

4. delivery (доставка заказов) 4 %
5. stock_level (возвращает остаток на складе заказанных предметов) 4 %

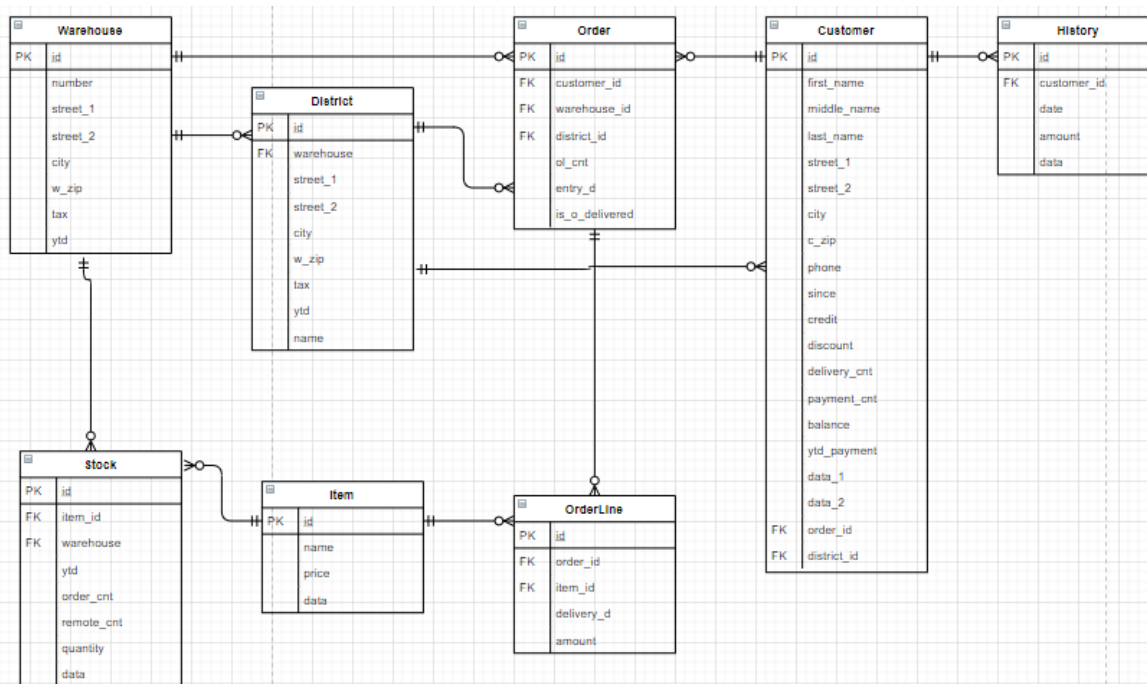


Рис. 1. Схема тестовой базы данных

Вероятность появления транзакций такая же, как и в оригинальном тесте TPC-C.

По сравнению с оригинальным тестом TPC-C данный тест несколько упрощен, в связи с техническими ограничениями и тем, что стоит цель проверить производительность ORM, а не устойчивость железа к нагрузке. Оригинальный тест проводится на вычислительных машинах, имеющих не менее 64 гигабайт оперативной памяти.

Основные различия:

1. Тест запускается с меньшим количеством виртуальных пользователей, чем в оригинальном тесте. В данный момент оборудование позволяет запустить от 2 до 5 виртуальных пользователей

2. Меньше количество записей в таблицах. Например: количество записей в отношении Stock в оригинальном тесте рассчитывается по формуле $100\,000 * W$, где W – это количество складов, а в предлагаемом тесте: $100 * W$

3. В оригинальном тесте некоторые из 5 транзакций имеют несколько вариантов запроса данных из базы. Например в транзакции Payment с одной вероятностью клиент будет запрашиваться из базы по ID, с другой по фамилии и имени. На данный момент в предлагаемом тесте в подобных ситуациях вызов производится только по ID, в дальнейшем планируется реализовать и второй вариант.

4. В схеме данного теста отсутствует одна таблица, которая есть в оригинальном тесте. В оригинальном тесте, когда создается заказ, то он добавляется и в таблицу Order, и в таблицу NewOrder. После доставки заказа, он удаляется из таблицы NewOrder. Это ускорит работу при огромном количестве транзакций в минуту, но так как у в данном тесте меньше потоков и меньше одновременных транзакций, обращающихся к базе, то это излишне. Вместо этого в таблице Order был добавлен boolean атрибут "is_o_delivered", который будет равен False, до тех пор, пока заказ не доставят.

Далее будет кратко описано, что делает каждая транзакция.

Транзакции

New Order

Краткое описание алгоритма:

1. Транзакции подается два аргумента: id клиента и id склада
2. Из базы запрашиваются склад и клиент по переданным id
3. Случайным образом берется один из участков склада
4. Генерируется случайное число строк(позиций) заказа
5. Создается объект заказа
6. В цикле создаются объекты для позиций данного заказа. На каждой итерации цикла из базы данных случайным образом берется товар из таблицы Item
7. Для каждого товара в заказе, в базе данных изменяется его доступное количество на складе

Payment

Краткое описание алгоритма:

1. Транзакции подается два аргумента: id клиента и id склада
2. Из базы данных запрашиваются склад и клиент по переданным id
3. Случайным образом берется один из участков склада и сумма оплаты
4. Баланс склада и отдельного участка увеличивается на сумму оплаты
5. Баланс клиента уменьшается на сумму оплаты
6. Счетчик количества оплат у клиента увеличивается на 1
7. Суммарная сумма, полученного от данного клиента, увеличивается на сумму оплаты
8. Создается объект истории платежей.

Order Status

Краткое описание алгоритма:

1. Транзакции подается id клиента
2. Из базы данных берутся клиент и его последний заказ
3. Из заказа берутся его статус (доставлен он или нет) и позиции заказа

Delivery

Краткое описание алгоритма:

1. Транзакции подается id склада
2. Из базы запрашиваются склад по id и все его участки
3. Для каждого участка берется самый старый из не доставленных заказов. В каждом из них статус доставки меняется на True
4. Из базы данных берутся пользователи, чьи заказы были доставлены в ходе данной транзакции, и у каждого из них увеличивается счетчик доставок

Stock Level

Краткое описание алгоритма:

1. Транзакции подается id склада
2. Из базы данных запрашивается склад по id
3. Из базы данных запрашиваются последние 20 заказов этого склада
4. Для каждой позиции этих заказов из базы данных запрашиваются кол-во остатка товара на складе

Результаты тестирования

В тестировании участвуют два ORM:

1. SQLAlchemy. На графиках изображен синей линией
2. PonyORM. На графиках изображен желтой линией

Ниже приведены результаты запуска теста на 10 минут с 2 параллельными процессами, обращающимися к базе. Процессы реализованы с помощью модуля multiprocessing. В качестве СУБД используется PostgreSQL. На рисунках ниже изображены в виде графиков результаты запуска теста на каждой из пяти типов тестовых транзакций и для смеси пяти транзакций. На графиках ниже ось X обозначает время в минутах, ось Y – количество транзакций выполненных за минуту.

Запуск со всеми транзакциям

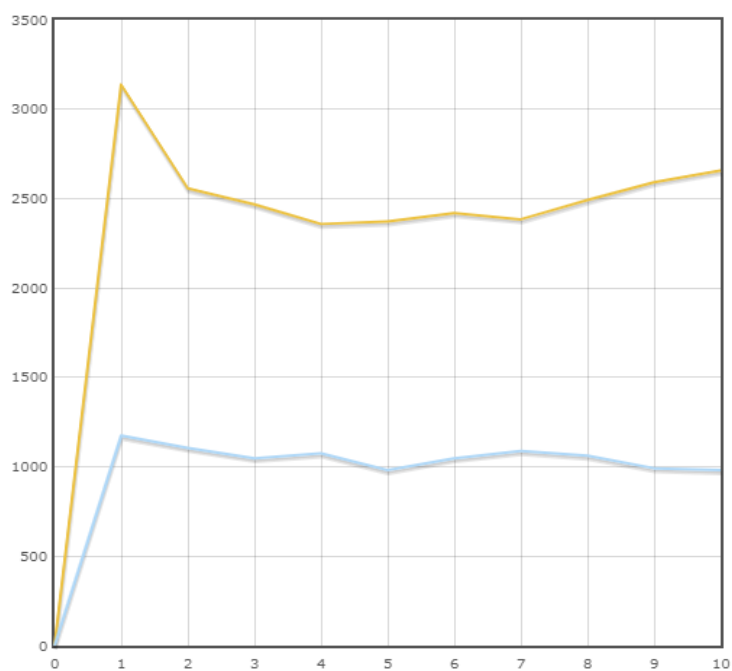


Рис. 2. Запуск со всеми транзакции

Средняя скорость:

1. PonyORM – 2543 тран/мин
2. SQLAlchemy – 1055.8 тран/мин

Транзакция «New Order»

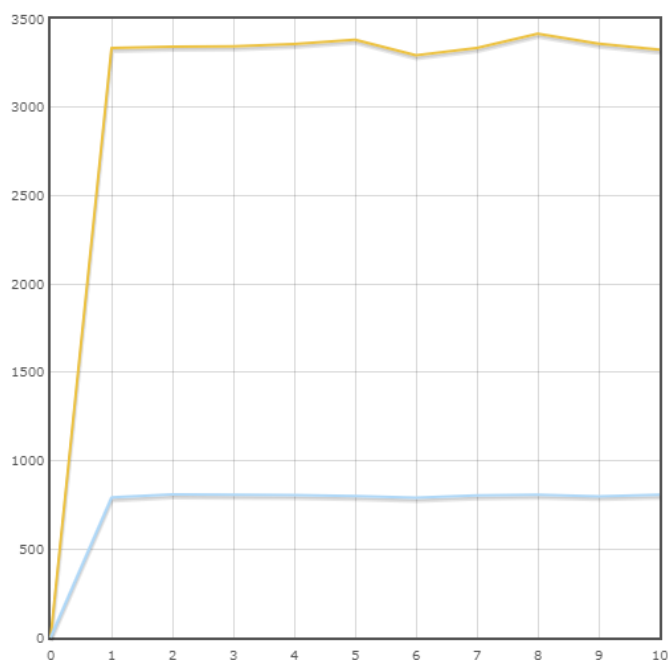


Рис. 3. Транзакция «New Order»

Средняя скорость:

1. PonyORM – 3349.2 тран/мин
2. SQLAlchemy – 802.1 тран/мин

Транзакция «Payment»

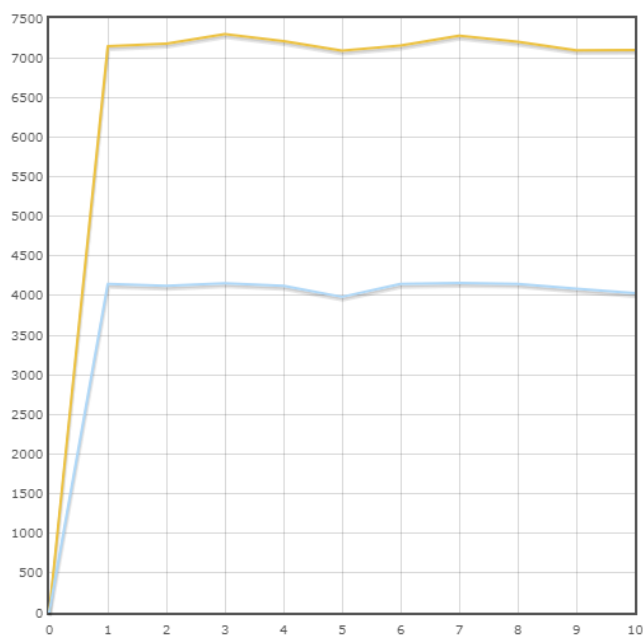


Рис. 4. Транзакция «Payment»

Средняя скорость:

1. PonyORM – 7175.3 тран/мин
2. SQLAlchemy – 4110.6 тран/мин

Транзакция «Order Status»

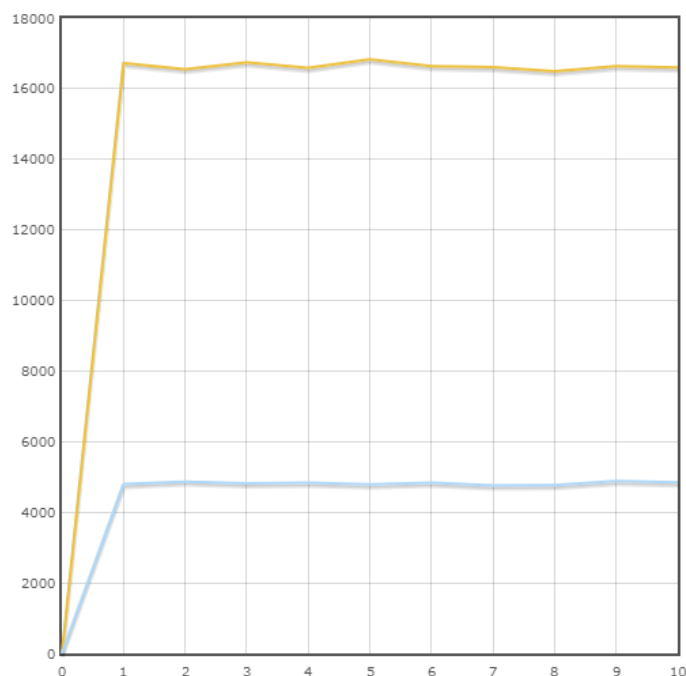


Рис. 5. Транзакция «Order Status»

Средняя скорость:

1. Pony – 16645.6 тран/мин
2. SQLAlchemy – 4820.8 тран/мин

Транзакция «Delivery»

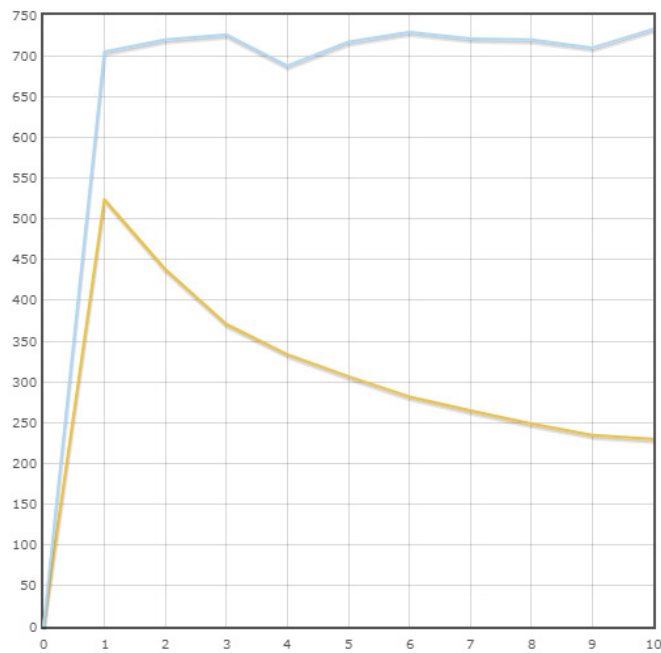


Рис. 6. Транзакция «Delivery»

Средняя скорость:

1. SQLAlchemy – 716.9 тран/мин
2. PonyORM – 323.5 тран/мин

Транзакция «Stock Level»

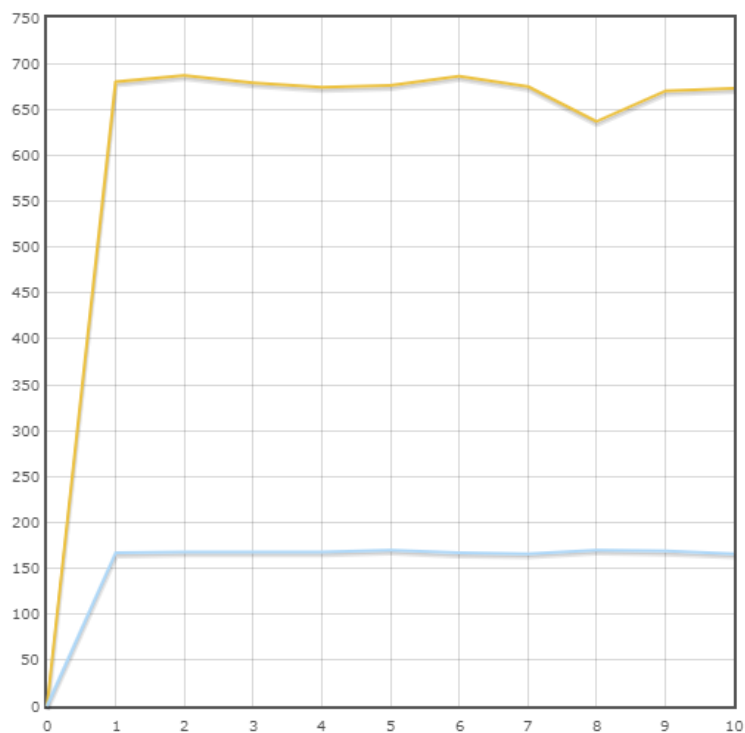


Рис. 7. Транзакция «Stock Level»

Средняя скорость:

1. PonyORM – 677.3 тран/мин
2. SQLAlchemy – 167.9 тран/мин

Анализ результатов тестиров:

Далее были проанализированы результаты тестирования и получены следующие выводы:

1. В 4 из 5 транзакций PonyORM оказалась быстрее. После анализа различий в работе между данными ORM-библиотеками выяснилось, что при генерации SQL кода PonyORM запоминает результат трансляции выражений Python в язык SQL, и не выполняет трансляцию заново при повторном выполнении запроса, в то время как SQLAlchemy вынуждена генерировать текст SQL при каждом выполнении запроса. Можно предположить, что именно по этой причине PonyORM оказывается быстрее в большинстве случаев.

2. По результатам теста SQLAlchemy выполняет транзакции типа Delivery быстрее. После анализа генерируемых SQL запросов, выяснилось, что SQLAlchemy умеет объединять несколько операций UPDATE, применяемых к разным объектам, в единую команду. PonyORM в таких случаях генерирует отдельные SQL запросы для каждого отдельно взятой операции UPDATE. Вот пример такого запроса, как он записан в логах SQLAlchemy:

```
INFO:sqlalchemy.engine.base.Engine:UPDATE order_line SET delivery_d= %(delivery_d)s WHERE
order_line.id = %(order_line_id)s
INFO:sqlalchemy.engine.base.Engine:(
{'delivery_d': datetime.datetime(2020, 4, 6, 14, 33, 6, 922281), 'order_line_id': 316},
{'delivery_d': datetime.datetime(2020, 4, 6, 14, 33, 6, 922272), 'order_line_id': 317},
{'delivery_d': datetime.datetime(2020, 4, 6, 14, 33, 6, 922261)})
```

В данном примере видно, что SQLAlchemy генерирует один SQL запрос, в котором производится UPDATE трех строк отношения OrderLine. PonyORM в такой же ситуации сгенерировал бы три запроса.

Вывод

Данное тестирование показало, что Pony работает быстрее в 2 – 4 раза при выборке из базы данных, а SQLAlchemy в некоторых случаях может с заметно большей скоростью производить запросы типа Update.

В дальнейшем таким методом можно проанализировать и другие ORM-библиотеки для Python, что позволит сравнить их друг с другом, увидеть их преимущества и недостатки, и выбирать максимально подходящий ORM, в зависимости от нужд проекта.

Данный тест справился со своей задачей, смог проанализировать работу двух ORM библиотек для Python и выявить, в каких ситуациях лучше использовать ту или иную библиотеку.

Репозиторий теста

URL: <https://github.com/DominovTut/Python ORM Benchmark>

Библиографический список

1. Tortoise ORM Benchmark. URL: <https://github.com/tortoise/orm-benchmarks> (дата обращения: 20.03.2020).
2. Документация SQLAlchemy. URL: <https://docs.sqlalchemy.org/en/13/> (дата обращения: 20.03.2020).
3. Документация PonyORM. URL: <https://docs.ponyorm.org/> (дата обращения: 20.03.2020).
4. Спецификация теста TPC-C. URL: http://www.tpc.org/tpc_documents_current_versions/pdf/tpc-c_v5.11.0.pdf (дата обращения: 20.03.2020).
5. Волков А. А. Тесты TPC. URL: <https://www.osp.ru/news/articles/1995/0402/13031418> (дата обращения: 20.03.2020).

УДК 004.43: 519.682

Е. Е. Доминов

магистрант кафедры вычислительных систем и сетей

А. В. Гордеев – доктор технических наук, профессор – научный руководитель

СБОРКА ИНТЕРАКТИВНОЙ СРЕДЫ РАЗРАБОТКИ ДЛЯ ЯЗЫКА KOTLIN НА ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ С АРХИТЕКТУРОЙ ЭЛЬБРУС

На данный момент всё большую популярность активно набирает язык программирования Kotlin. Kotlin — это статически типизированный язык программирования от компании JetBrains, работающий поверх JVM. Наиболее широко этот язык применим в написании мобильных приложений. Kotlin заимствовал многое от языков JAVA и Scala. Основные его плюсы – это лаконичность, типобезопасность, быстрая компиляция и простота разработки.

Основной областью данного языка является разработка приложений под платформу Android. В данный момент язык Kotlin поддерживается компанией Google и признан официальным языком разработки под Android, наравне с языком Java.

В 2017 году в докладе на Jpoint Андрей Бреслав, возглавляющий проект «Kotlin», перечислил области применения Kotlin, которые кажутся ему перспективными. Помимо мобильной разработки, он ещё указал такие направления, как разработка кроссплатформенных компьютерных игр и embedded-приложений. Также Kotlin нацелен на максимальную кроссплатформенность и в мобильной, и в серверной и в десктопной разработке. Авторы языка признаются, что не пытались придумать что-то кардинально новое. Язык специально задумывался максимально прагматичным и удобным в использовании для разработчиков. Ниже приведены основные особенности данного языка [9]:

1. Extension functions
2. Null Safety
3. String templates
4. Properties
5. Data classes
6. Range expressions

Null safety — проверка переменных на nullable type проверяется еще на этапе компиляции. Это довольно удобно и помогает избежать большого количества багов.

Extension Functions позволяют расширять классы дополнительным функционалом, не наследуясь от этого класса и не используя паттерны декораторов. Это позволяет персонализировать для своего проекта классы, подключаемые к проекту в виде модулей. Во многих случаях это может сделать разработку быстрее и удобнее.

String template позволяет создавать текстовые шаблоны, что довольно удобно для передачи приложению текстовых значений на этапе тестирования и при работе с запросами к базам данных.

Иногда нам необходимо получить доступ к типу, переданному в качестве параметра. В таких случаях inline-функции могут принимать *параметры вещественного типа* (reified type parameters). Тип параметра определяется с помощью ключевого слова reified, после этого он доступен внутри функции почти так же, как и обычный класс. Так как функция встроена, то для работы таких операторов как !is и as рефлексия не нужна. Хотя рефлексия может быть не нужна во многих случаях, мы всё ещё можем использовать её с параметром вещественного типа. Тип, который не имеет представление во времени исполнения (например, параметр не вещественного или фиктивного типа вроде Nothing), не может использоваться в качестве аргумента для параметра вещественного типа.

Также Kotlin упрощает работу с полями классов. Обращаться к полям можно, как к обычным переменным, при этом будет вызван геттер или сеттер нужного поля.

МЦСТ «ЭЛБРУС

«Эльбрус 401-PC» – основная рабочая станция, выпускаемая АО «МЦСТ», распространяется с предустановленной операционной системой «Эльбрус» — Debian дистрибутивом, портированным под архитектуру ELBRUS. Эльбрус 401-PC поддерживает выполнение команд архитектуры x86, но эффективнее выполняет команды родной архитектуры Эльбрус.

Микропроцессор Эльбрус-4С (1891ВМ8Я) – многоядерный универсальный высокопроизводительный микропроцессор, построенный в соответствии с улучшенной архитектурой «Эльбрус». Каждое ядро процессора декодирует и отправляет на исполнение до 23 операций за такт.

Эльбрус-4С представляет собой систему на кристалле, содержащую 4 вычислительных ядра, кэш-память 2-го уровня общим объемом 8 Мегабайт, 3 контроллера памяти, 3 канала межпроцессорного обмена и канал ввода-вывода

Рабочая тактовая частота микросхемы составляет 800 МГц. Кристалл выполнен по технологической норме 65 нм, средняя рассеиваемая мощность составляет 45 Вт. Имеются средства для значительного снижения рассеиваемой мощности.

Основная сфера применения микропроцессоров «Эльбрус-4С» – серверы, настольные компьютеры, мощные встраиваемые вычислители, предназначенные для работы в сферах с повышенными требованиями к следующим аспектам:

- информационная безопасность
- рабочий диапазон температур
- длительность жизненного цикла продукции

Особенности архитектуры «Эльбрус» позволяют эффективно применять процессор Эльбрус-4С в системах цифровой интеллектуальной обработки сигналов, в математическом моделировании, научных расчётах и других сферах с повышенными требованиями к вычислительной мощности [2].

Платформа Эльбрус оснащена JVM, что теоретически делает возможным написание и запуск программ на языке Kotlin. Исследование и создание возможности писать программное обеспечение на языке Kotlin на вычислительных машинах с Эльбрус-архитектурой могут позволить писать программное обеспечение для устройств на платформе Android, используя отечественные компьютеры. В дальнейшем это может привести к созданию мобильных устройств и планшетов, оснащенных ОС Android и процессорами Эльбрус. На данный момент этот вопрос особенно актуален, так как готовится массовый выпуск планшетов на основе микропроцессора «Эльбрус-1С+». Данный планшет является совместной разработкой ИНЭУМ и МЦСТ.

Основные характеристики микропроцессора «Эльбрус-1С+» [10]:

1. «Эльбрус-1С+» 64-разрядный микропроцессор
2. Частота – 1 ГГц
3. Совмещен с графическим ядром с аппаратным ускорением 3D графики
4. Уровень электропотребления под нагрузкой не больше 10 Вт
5. 6 сдвоенных арифметико-логических устройств
6. Присутствует динамическое управление частотой и напряжением питания ядра

Так как исходный код Kotlin IDE хранится в открытом доступе, можно вести работы по адаптации и сборке Kotlin IDE на Эльбрус архитектуре. Могут возникнуть сложности в связи с тем, что JVM Эльбруса имеет ряд отличий от JVM x86. В этом случае нужно будет адаптировать исходный код под JVM Эльбруса.

Библиографический список

1. Процессор «Эльбрус-4С» . URL: <http://www.mcst.ru/mikroprocessor-elbrus4s> (дата обращения: 20.03.2020).
2. Рабочая станция «Эльбрус 401-PC» . URL: <http://www.ineum.ru/arm-elbrus401> (дата обращения: 20.03.2020).
3. Программное обеспечение «Эльбрус». URL: <http://mcst.ru/programmnoe-obespechenie-elbrus> (дата обращения: 20.03.2020).
4. Архитектура отладчика Java Platform. URL: <https://docs.oracle.com/javase/6/docs/technotes/>

guides/jpda/architecture.html (дата обращения: 20.03.2020).

5. Удаленное выполнение кода в Java Debug Wire Protocol. URL: <https://www.securitylab.ru/analytcs/454945.php> (дата обращения: 20.03.2020).

6. Официальный сайт языка Kotlin. URL: <https://kotlinlang.org/> (дата обращения: 20.03.2020).

7. Документация Kotlin. URL: <https://github.com/JetBrains/kotlin-website/tree/master/pages/docs/reference> (дата обращения: 20.03.2020).

8. Официальный репозиторий Kotlin с исходным кодом компилятора. URL: <https://github.com/JetBrains/kotlin> (дата обращения: 20.03.2020).

9. Жемеров Д., Исакова С. Kotlin в действии. С. 22–27.

10. Процессор «Эльбрус-1С+». URL: <http://www.mcst.ru/elbrus-1c-plus> (дата обращения: 20.03.2020).

УДК 004.451.9

А. Д. Ёлкин

магистрант кафедры вычислительных систем и сетей

А. В. Гордеев – доктор технических наук, профессор – научный руководитель

ВЫБОР АРХИТЕКТУРЫ СЕТЕВОГО РЕПОЗИТОРИЯ ПАКЕТОВ ОС «ЭЛЬБРУС»

Базовой операционной системой для Вычислительного Комплекса Эльбрус 401-РС [1] является ОС (операционная система) «Эльбрус». Она построена на основе ядра Linux и поддерживает множество приложений с открытым исходным кодом. Имеются средства для разработки программного обеспечения, тесты для самодиагностики аппаратуры. ОС обеспечивает многозадачный и многопользовательский режимы работы. Для неё разработаны особые механизмы управления процессами, виртуальной памятью, прерываниями, сигналами, синхронизацией, поддержка тегированных вычислений.

Средства поддержки графического пользовательского интерфейса содержат базовые компоненты графической системы Xorg, а также набор различных вспомогательных библиотек, в том числе GTK+ (GIMP Toolkit) и Qt.

Для поддержки программ, работающих в защищённом режиме, поставляется компактная библиотека libmcst, обеспечивающая функции работы с памятью и поддержку ввода-вывода на уровне базовой библиотеки libc.

На данный момент существуют следующие версии дистрибутивов [2]:

- ОПО «Эльбрус» — общее программное обеспечение;
- ОС «Эльбрус» — портированная версия Debian 8.11;
- PDK «Эльбрус» — (Platform Development Kit, набор разработчика) та же ОС, но с возможностями разработки. Как утверждается, это самая современная версия ОС. Она предназначена для скачивания и установки на компьютеры с процессорами российского производства;
- PDK «Эльбрус» для архитектуры x86 — ОС для процессоров с системой команд x86. При этом версия пакетов ОС «Эльбрус» для микропроцессоров с системой команд Elbrus сохранена.

Но для обновления пакетов пользователь вынужден воспользоваться образом системы, скачанным из интернета. Это увеличивает трудоёмкость процесса, ведь нам нужно найти в интернете подходящий образ системы, скачать его, монтировать и начать установку. Или, если он монтирован на одном из компьютеров в сети, можно попытаться подтянуть обновления по локальной сети.

Решением данной проблемы является создание общего сетевого репозитория для пакетов ОС Эльбрус по примеру репозитория для Linux – подобных систем. Это позволит просто и эффективно обновлять систему. Также централизованный репозиторий позволит удобно хранить в одном месте все пакеты, предназначенные для ОС Эльбрус.

Современный сетевой репозиторий пакетов должен удовлетворять пяти требованиям [3]:

1. Распределенный контроль

Есть риск монополизации небольшим кругом участников системы научных коммуникаций определённых научных ресурсов. Распределённые сети в меньшей мере подвержены этому.

2. Инклюзивность

При создании репозитория нужно учитывать различные нужды и условия разных регионов и отраслей науки.

3. Интеллектуальная открытость и доступность

Научные ресурсы по возможности должны быть представлены в удобном формате и быть доступны всем, что повысит их ценность и позволит широко применять в интересах как научного сообщества, так и всего социума.

4. Устойчивость

Учреждения и исследовательские организации должны стать основными участниками глобальной сети, способствуя долгосрочной устойчивости ресурсов.

5. Функциональная совместимость

Репозитории должны следовать общим направлениям развития, функциональным возможностям и стандартам, обеспечивающим взаимодействие между институтами и позволяющим им совместно использовать внешних поставщиков услуг.

ОС Эльбрус является Debian — подобной системой и использует формат deb – пакетов, поэтому в качестве примера рассмотрим структуру сетевого репозитория для системы Debian GNU/Linux.

Официальные репозитории Debian [4] — общие для всех дистрибутивов семейства, и располагаются они по адресу: <https://deb.debian.org/debian/>. Это — «головное» хранилище пакетов, имеющее многочисленные региональные зеркала, принадлежность которых к стране указывается стандартным двухсимвольным префиксом, например, <http://ftp.ru.debian.org/debian/> [5] — российское зеркало. Сайты с первичным зеркалом имеют хорошее подключение к Интернет и синхронизируются напрямую с внутренней сетью Debian syncroху. Сайты с вторичным зеркалом могут содержать неполный архив (из-за нехватки места). Причисление сайта к вторичным зеркалам необязательно говорит о его медлительности или редкой обновляемости по сравнению с первичным сайтом, но если оно находится ближе к пользователю, чем первичное, и содержит нужную ему архитектуру, то такой вариант предпочтительнее.

Некоторые разработчики программ или разработчики Debian предоставляют дополнительные репозитории, которые можно использовать, помимо официальных. Это может быть вызвано различными причинами, например, программы ещё не находятся в официальном репозитории, могут быть проблемы в "правовом поле" (например, deb-multimedia), или потому что программа имеет закрытые исходные коды (например, Oracle, Opera).

Посмотрим поближе на устройство сетевого репозитория. Релизы Debian [6] названы в честь персонажей из фильмов «История игрушек» (wheezy, jessie, stretch, ...). У кодовых имен есть псевдонимы, так называемые наборы (stable, oldstable, testing, unstable). Релиз разделен на несколько компонентов. В Debian они называются main, contrib и non-free и указывают условия лицензирования программного обеспечения, которое они содержат. Релиз также содержит пакеты для различных архитектур (amd64, i386, mips, powerpc, s390x, ...), а также пакеты, не зависящие от исходного кода и архитектуры.

Корневой каталог репозитория имеет каталог dists, в котором, в свою очередь, есть каталог для каждого выпуска и набора, последний — обычно символические ссылки на первый, но браузер не покажет вам разницу. Каждый подкаталог выпуска содержит криптографически подписанный файл выпуска и каталог для каждого компонента. Внутри находятся каталоги для разных архитектур, называемые бинарными <arch> и источниками. И в этих файлах находятся пакеты, которые представляют собой текстовые файлы, содержащие метаданные пакетов.

Сами пакеты находятся в /pool в корневом каталоге репозитория. Там снова есть каталоги для всех компонентов, и в них находятся каталоги с именами 0, ..., 9, a, b, ... z, liba, ..., libz. И в них находятся каталоги, названные в честь программного пакета, который они содержат, и эти каталоги наконец содержат фактические пакеты, то есть файлы .deb. Имя не обязательно является именем самого пакета, пакет *bsdutils*, например, находится в каталоге *pool/main/u/util-linux* — это имя источника, из которого создан пакет. Один исходящий источник может генерировать несколько двоичных пакетов, и все они окажутся в одном и том же подкаталоге ниже пула. Дополнительные однобуквенные каталоги — это всего лишь хитрость, позволяющая избежать слишком большого количества записей в одном каталоге, с чем у многих систем традиционно возникают проблемы с производительностью.

В подкаталогах /pool обычно есть несколько версий пакета, и информация о том, к каким выпускам относится каждая версия, находится исключительно в индексах. Таким образом, одна и та же версия пакета может принадлежать нескольким выпускам, но использовать дисковое пространство только один раз, причем без использования жестких или символических ссылок, поэтому зеркалирование довольно простое и даже работает с системами, которые не имеют этих концепций. В директории /indices содержатся файлы переопределения.

Для каждого дистрибутива и компонента должен быть один файл, обычно называемый *override.distribution.component*. Данные файлы [7] переопределяют местонахождение двоичных пакетов, которые есть в репозитории. Структура файла переопределения проста: каждая строка

содержит разделенную пробелами запись о каждом двоичном пакете в выпуске и компоненте. Столбцы – это имя пакета, приоритет (обязательный, важный, стандартный, необязательный или дополнительный) и раздел (admin, base, devel и т. Д.).

Последний элемент также является подкаталогом, в котором находится двоичный пакет. Директория /doc содержит документацию для Debian, /project – экспериментальные пакеты проекта и другие разные файлы, а /tools – инструменты для создания загрузочных дисков и загрузки Linux.

Для размещения репозитория в сети Интернет нужна машина с легальным IP адресом и доменным именем. На ней должен быть установлена серверная операционная система и она должна иметь содержать работающий WEB – сервер (Apache, Nginx и т.д.). В корневой папке сервера должен быть создан репозиторий [8]. Для заполнения репозитория можно использовать стандартную утилиту dpkg-scanpackages или, к примеру, aptly. Для создания репозитория с помощью aptly [9] нужно выполнить команду:

```
aptly repo create -comment="comment"
-distribution="distribution " architectures="architectures"
-component="component" repo_name
```

Или можно использовать короткую команду добавить нужные параметры вручную:

```
aptly repo create repo_name
aptly repo edit -comment="comment"
-distribution="distribution " architectures="architectures"
-component="component" repo_name
```

Просмотр существующих репозиторияев осуществляется командой:

```
aptly list
```

Добавление пакетов в репозиторий осуществляется командой:

```
aptly repo add repo_name path/to/deb_package_or_folder_of_packages
```

Публикация репозитория выполняется командой:

```
aptly publish repo repo_name
```

Удаление публикации

```
aptly publish drop repo_name
```

Ещё aptly есть поддержка снимков состояния, создания зеркал, перемещения пакетов между репозиториями, поддержка зависимостей. Эта утилита имеет встроенный HTTP сервер для работы с опубликованными репозиториями.

В комплекс мер по обслуживанию репозитория входят поиск и удаление пакетов, перемещение пакетов между репозиториями, обновление опубликованных репозиторияев. Все перечисленные действия могут быть осуществлены с помощью утилит aptly и dpkg-scanpackages.

В связи с размещением репозитория в сети интернет существует вероятность подмены репозитория злоумышленником на свой, содержащий модифицированные пакеты. Поэтому для обеспечения безопасности репозиторий должен иметь защиту от подмены. Она должна быть реализована при помощи сверки цифровых подписей репозитория и клиента, где доверенным будет считаться тот репозиторий, которые имеет цифровую подпись, а пользовательский компьютер содержит открытый ключ для этого репозитория.

По мере повышения нагрузки на машину, поддерживающую репозиторий, возможно потребуется развёртывание кластера из нескольких машин. Кластеризация повысит уровень доступности репозитория, ведь в случае выхода из строя одного из экземпляров другие остаются доступными для работы.

Также следует предусмотреть возможность использования в репозитории следующих технологий:

- применение унифицированных идентификаторов ресурса;
- размещение информации об используемой лицензии;
- навигация для облегчения процедуры поиска;
- взаимодействие с пользователями (аннотирование, комментирование, рецензирование);
- передача контента;
- сервисы дискавери;
- статистические метрики посещаемости ресурса;

- идентификация пользователей;
- аутентификация пользователей;
- стандартизованные метрики использования ресурса;
- архивирование.

В ходе анализа архитектуры репозитория Debian было установлено, что данная реализация полностью удовлетворяет требованиям современного репозитория, а именно: обеспечивает распределённый контроль за счёт зеркалирования и создания дополнительных репозиториев от разработчиков, обеспечивает инклюзивность, обеспечивает инклюзивность, интеллектуальную открытость и доступность научных ресурсов, является устойчивым и поддерживает функциональную совместимость пакетов. Также она полностью подходит для хранения и распространения пакетов для ОС Эльбрус.

Библиографический список

1. Рабочая станция «Эльбрус 401-PC». URL: <http://www.ineum.ru/arm-elbrus401> (дата обращения: 02.03.2020).
2. Программное обеспечение «Эльбрус». URL: <http://mcst.ru/programmnoe-obespechenie-elbrus> (дата обращения: 03.03.2020).
3. Рекомендации по построению репозиториев. URL: https://openrepository.ru/images/docs/Metod_Schwarzman.pdf (дата обращения: 04.03.2020).
4. Типы репозиториев Debian. URL: <http://myubuntu.ru/faq/typy-repozitoriev-v-ubuntu> (дата обращения: 03.03.2020).
5. Зеркало официального репозитория Debian. URL: <https://ftp.ru.debian.org/debian> (дата обращения: 03.04.2020).
6. Структура репозиториев Debian – подобных систем и устройство deb-пакетов. URL: <https://www.alv.me/debian-i-ubuntu-pakety-i-repozitorii/> (дата обращения: 04.03.2020).
7. Документация по репозиториям Debian. URL: <https://www.debian.org/doc/manuals/repository-howto/repository-howto.en.html> (дата обращения: 03.03.2020).
8. Рекомендации по созданию репозитория. URL: <https://wiki.debian.org/ru/CreateLocalRepo> (дата обращения: 06.03.2020).
9. Официальная документация утилиты aptly. URL: <https://www.aptly.info/doc/> (дата обращения: 06.03.2020).

УДК 004.4

И. А. Житников

магистрант кафедры вычислительных систем и сетей

А. В. Гордеев – доктор технических наук, профессор – научный руководитель**ПЕРЕНОС СИСТЕМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ MOODLE
НА КЛАСТЕР ВЫСОКОЙ ДОСТУПНОСТИ НА БАЗЕ СЕРВЕРОВ «ЭЛЬБРУС»**

Moodle (англ. modular object-oriented dynamic learning environment – модульная объектно-ориентированная динамическая учебная среда) — это свободная распространяемая система управления обучением, ориентированная на организацию взаимодействия между преподавателем и учениками, она подходит как для организации традиционных дистанционных курсов, так и для поддержки очного обучения. Является свободным (распространяется по лицензии GNU GPL) веб-приложением, предоставляющим возможность создавать сайты для онлайн-обучения. Первая версия была написана 20 августа 2002 года [1].

Благодаря Moodle пользователь может создавать курсы, наполняя их содержимым в виде текстов, вспомогательных файлов, презентаций, опросников и т.п. Все курсы в Moodle состоят из тем, темы в свою очередь могут иметь в себе лекции, тесты и scorm-пакеты (международный стандарт для курса системы дистанционного обучения, его использование гарантирует поддержку данного курса любой из существующих систем дистанционного обучения) [2]. Для создания лекции в Moodle предусмотрен внутренний редактор Authoring Tool [3]. Помимо инструментов создания учебных материалов, в системе реализованы приложения для контроля знаний обучающихся, в частности анкеты и опросы.

Moodle имеет встроенную систему аналитики, которая позволяет создавать отчёты по активности пользователей системы, например просмотры курсов, оценки тестирования, комментариев, входы и выходы [4].

Для того, чтобы использовать Moodle, необходимо всего лишь иметь любой web-браузер, что делает использование этой учебной среды удобной как для преподавателя, так и для студентов. По результатам выполнения учениками заданий, преподаватель может выставлять оценки и давать комментарии, т.е. вести полноценный учёт успеваемости учеников. Moodle является центром создания учебного материала и обеспечения интерактивного взаимодействия между участниками учебного процесса.

Важной особенностью системы являются модули, расширяющие возможности системы; примером подобного модуля может служить модуль «Электронный деканат», позволяющий управлять учебным процессом и упростить документооборот внутри учебного заведения. Существует большое количество подобных модулей, реализующих самые разные функции для администрирования курсов, тестирования обучающихся, подсистемы аналитики и т. д. Это позволяет производить более гибкую настройку учебного процесса.

Moodle поддерживает мобильные браузеры, что позволяет увеличить охват платформ, с которыми работает система. Также реализовано мобильное приложение Moodle Mobile для платформ Android и iOS [5].

Большим плюсом данной системы является возможность интеграции с существующими сервисами, например с системами вебинаров для организации лекции в реальном времени, или с платёжными системами для подключения оплаты за курсы.

АО «МЦСТ» – российская компания, занимающаяся разработкой универсальных микропроцессоров, микроконтроллеров и управляющих вычислительных комплексов. В линейке процессоров компании присутствуют многоядерные модели, такие как «Эльбрус-4С» и «Эльбрус-8С». Партнёр компании МЦСТ – Институт электронных управляющим машин имени И.С. Брука производит вычислительную технику на базе процессоров Эльбрус, в частности рабочие станции и серверы [6]. Таким образом, на базе процессора «Эльбрус-4С» была разработана рабочая станция «Эльбрус 401-РС» и сервер «Эльбрус-4.4 1U».

Компания создала ОС Эльбрус, которая является портированной версией системы Linux, конкретно дистрибутива Debian.

ОС «Эльбрус» разработана компанией МЦСТ для компьютеров с архитектурой SPARC (от англ. “scalable processor architecture” – масштабируемая процессорная архитектура) и «Эльбрус» на базе ядра Linux [7]. Операционная система обеспечивает многозадачный и многопользовательский режимы работы. Для неё разработаны оригинальные механизмы управления процессами, виртуальной памятью, прерываниями, сигналами, синхронизацией, реализована поддержка тегированных вычислений.

Целью работы является портирование системы курсов Moodle на компьютеры с процессором Эльбрус-4С и последующий запуск этого приложения на кластере высокой доступности, созданного на серверах Эльбрус 4.4.

Работа предполагает две части. Во-первых, необходимо из исходных кодов собрать и установить саму Moodle, а также пакеты, от которых зависит данная система. Во-вторых, нужно настроить сетевую конфигурацию для кластерной работы системы из двух серверов.

По результатам проведённой на данный момент работы, были получены предварительные конфигурации для запуска системы на сервере Эльбрус, в частности стало известно, какие из установленных пакетов необходимо обновить, с какими из пакетов возникают конфликты.

Первым делом было проверено наличие системы контроля версий Git, так как это основной инструмент для получения исходных кодов программ. В ОС Эльбрус Git не предустановлен, поэтому подключался диск с дополнительным программным обеспечением, откуда было установлено данное программное обеспечение.

Сначала было принято решение произвести установку Moodle со стандартными зависимостями [8]. Установлено, что для работы Moodle версии 3.5 необходим интерпретатор языка PHP версии не ниже 7.0.0. В ОС Эльбрус была предустановлена PHP 5.6.40, эта версия интерпретатора не поддерживается системой Moodle 3.5, поэтому необходимо её обновить. Старая версия была удалена с машины, для установки новой был с помощью Git клонирован репозиторий PHP, устанавливаемой версией было выбрана 7.4.0. Несмотря на то, что вывод команды «make test» говорит об отсутствии ошибок, команда «make install» заканчивается сообщением об ошибке. Тем не менее, интерпретатор успешно запускает и выполняет php-скрипты. Нельзя точно утверждать, что при работе Moodle интерпретатор будет работать корректно, поэтому необходимо исследовать данную ошибку, найти её причины и устранить их. Либо инсталлировать интерпретатор других версий.

Стандартной СУБД (системой управления базами данных) для Moodle является MySQL. После клонирования репозитория с исходным кодом MySQL Server, было установлено, что большинство программ, от которых зависит сервер СУБД, либо отсутствуют в ОС Эльбрус, либо имеют устаревшую версию. Все необходимые программы и пакеты являются open-source проектами и располагают исходный код в свободном доступе. Была обновлена утилита CMake, компилятор GCC, установлены библиотеки GMP, MPFR, MPC и Boost C++. Некоторые пакеты не пришлось обновлять (например Bison, ncurses, Make). На данном этапе наибольшие проблемы вызвала библиотека OpenSSL. В системе уже была установлена эта библиотека, инсталлятор Moodle выдавал сообщение о несоответствии необходимой и установленной версии. Пакет последней версии был собран из исходных кодов и установлен, но сообщение не пропало. Вскоре выяснилось, что инсталлятору необходим пакет для разработчиков. После инсталляции данного пакета сообщения об ошибке пропали.

Процесс установки запустился, по достижении прогресса в 70 % завершился ошибкой. Причиной ошибки являлась устаревшая библиотека zlib, в последствии также обновлена путём сборки из исходных кодов и установки. Тем не менее, даже после этого установка не завершилась успешно. На 95 % прогресса процесс завершился с ошибкой. Сообщение об этой ошибке было найдено на баг-трекере MySQL, разработчики пообещали решить эту проблему в следующем релизе.

Ввиду того, что Moodle поддерживает и другие СУБД, была произведена попытка инсталляции PostgreSQL. В отличие от MySQL, PostgreSQL имеет меньшее количество зависимостей, поэтому инсталляция происходит быстрее и проще. Данная СУБД была собрана из исходных кодов, загруженных с официального сайта и успешно инсталлирована PostgreSQL 11-ой версии.

Веб-сервер Apache, предустановленный в ОС Эльбрус, не соответствует минимальным

требованиям Moodle. Ввиду этого было предпринято решение обновить веб-сервер. Репозиторий Apache был клонирован с GitHub, полученные оттуда исходные коды были собраны с помощью команды `stake` и установлены. Установка прошла успешно, но для Moodle понадобилось ещё два пакета с официального сайта Apache. После установки данных пакетов должна была появиться возможность установки Moodle из веб-браузера, но по неизвестной причине это невозможно, ведётся поиск решения. Далее было решено выполнить установку из интерфейса командной строки. В течение данного процесса была выявлена следующая ошибка — программа установки не обнаружила драйвер базы данных для интерпретатора языка PHP. Были найдены исходные коды пакета `php-pgsql` для версии PHP 7.3.0, на сайте пакетов для дистрибутива Debian. Данный пакет был собран и установлен, но программа установки так и не смогла обнаружить СУБД. В целом, остаётся устранить проблему соединения установщика с базой данных. Решений несколько – смена СУБД (например, возврат к MySQL), смена версии интерпретатора языка PHP.

Достижение поставленных целей позволит использовать имеющиеся серверы для изучения архитектуры Эльбрус. Впоследствии можно создать на базе Moodle обучающий курс по вычислительным машинам данной платформы. В рамках данного курса рассмотреть особенности этой архитектуры, например систему команд процессора или процесс динамической двоичной трансляции, а также предоставить возможность студентам удалённо запускать собственные программы.

Библиографический список

1. About Moodle – Moodle Docs. URL: https://docs.moodle.org/38/en/About_Moodle (дата обращения: 30.03.2020).
2. Moodle — обзор самой популярной системы электронного обучения. URL: <https://lmslist.ru/free-sdo/obzor-moodle/> (дата обращения: 30.03.2020).
3. Система электронного обучения и тестирования Moodle: обзор возможностей URL: <https://www.ispring.ru/elearning-insights/moodle> (дата обращения: 31.03.2020).
4. Moodle, как платформа организации eLearning и дистанционного обучения. URL: <https://habr.com/ru/post/139629/> (дата обращения: 31.03.2020).
5. Moodle app. URL: <https://download.moodle.org/mobile/> (дата обращения: 01.04.2020).
6. Партнёры МЦСТ. URL: <http://www.mcst.ru/partnery-mcst> (дата обращения: 01.04.2020).
7. Программное обеспечение «Эльбрус». URL: <http://mcst.ru/programmnoe-obespechenie-elbrus> (дата обращения: 02.04.2020).
8. Installing Moodle. URL: https://docs.moodle.org/38/en/Installing_Moodle#Software (дата обращения: 02.04.2020).

УДК 004.451.9

И. А. Житников

магистрант кафедры вычислительных систем и сетей

А. В. Гордеев – доктор технических наук, профессор – научный руководитель

УСТАНОВКА ОПЕРАЦИОННЫХ СИСТЕМ «ЭЛЬБРУС» И ALTLINUX НА КОМПЬЮТЕРЫ С ПРОЦЕССОРАМИ ЭЛЬБРУС-4С

В 2019 году кафедра вычислительных систем и сетей ГУАП получила в своё распоряжение рабочие станции “Эльбрус-401PC” и серверы “Эльбрус-4.4 1U”. В процессе изучения и освоения данной аппаратной платформы появилось много интересной информации о которой можно рассказать. Данная статья освещает вопрос установки операционных систем на рабочие станции.

На сегодняшний день, для платформы “Эльбрус” существует несколько операционных систем, созданных под различные процессоры данной архитектуры. Большинство из этих операционных систем подразумевают специальное применение, но есть и системы общего назначения.

Ярким примером операционной системы особого применения (тем не менее относится к классу ОС (операционная система) общего назначения) является МСВС (Мобильная Система Вооруженных Сил) [1]. Данная ОС была создана для решения задач построения автоматизированных стационарных защищённых систем. МСВС является многозадачной многопользовательской сетевой операционной системой, она поддерживает многопроцессорные конфигурации и такие механизмы разграничения доступа как ролевая модель, мандатное управление доступом и списки контроля доступа [2]. Система работает на таких аппаратных платформах как Intel x86, SPARC(Эльбрус-90микро), IBM System/390 и MIPS. Также ОС имеет сертификаты по требованиям безопасности информации Министерства обороны Российской Федерации. Следующим этапом развития ОС МСВС является ОС “Заря”, основанная на дистрибутиве Red Hat Enterprise Linux. На данный момент информации о портировании ОС “Заря” на платформу “Эльбрус” нет.

Операционные системы специального назначения представлены несколькими дистрибутивами: Astra Linux, QNX, “Нейтрино” и “Нейтрино-Э”, “БагрОС-4000”.

Astra Linux является операционной системой основанной на ядре Linux и дистрибутиве Debian. Данная ОС применяется для комплексной защиты информации и построения защищённых автоматизированных систем. Релиз ОС Astra Linux для вычислительных комплексов на базе архитектуры “Эльбрус” носит название Leningrad. Система обеспечивает высокую степень защиты обрабатываемой информации вплоть до уровня государственной тайны “особой важности”. Также как и ОС МСВС, она имеет сертификацию Министерства обороны РФ [3]. В состав данного дистрибутива входит множество пакетов с открытым исходным кодом, например пакет офисных приложений LibreOffice, графический редактор GIMP, медиапроигрыватель VLC, веб-браузер Mozilla Firefox, и многие другие. Astra Linux работает с пакетами на базе формата .deb (Debian).

Операционные системы реального времени на платформе “Эльбрус” разработанные ООО “СВД Встраиваемые Системы” представлены тремя ОС: ЗОСРВ (защищённая операционная система реального времени) “QNX”, ЗОСРВ “Нейтрино” и ЗОСРВ “Нейтрино-Э” [4]. Данные системы предназначены для отказоустойчивого и предсказуемого управления ресурсами вычислительных комплексов на базе микропроцессоров “Эльбрус” в режиме реального времени. Также они могут использоваться при создании автоматизированных систем, имеющих класс защищённости до 1Б включительно. “Нейтрино-Э” позволяет решать данные задачи на многопроцессорных/многоядерных и многомашинных вычислительных комплексах.

Операционная система реального времени БагрОС-4000 разработана для применения в авионике, она является актуальной высоконадёжной, высокоэффективной и мобильной ОСРВ (операционной системой реального времени) [5]. БагрОС-4000 предназначена для управления вычислительными ресурсами, организации вычислительных процессов и функционирования программного обеспечения в режиме реального времени. БагрОС-4000 поддерживает такие архитектуры как Эльбрус, MIPS64, PowerPC, ARMv7, Intel x86. Операционная система также поддерживает функционирование многомодульных (многопроцессорных, многоядерных) вычислительных систем.

Стоит отметить, что БагрОС-4000 имеет средства обработки ошибочных ситуаций, содержит средства восстановления работы приложений при сбоях. БагрОС-4000 умеет обрабатывать такие ошибки как: сбои и отказы аппаратуры; исключительные ситуации (например, недопустимый адрес команды, деление на 0 и т. п.); переполнение стека памяти; ошибки, выявленные прикладной программой (недопустимые данные, превышение допустимого интервала времени и т. п.). Обработка некорректных ситуаций производится как на уровне прикладной программы, так и на уровне операционной системы.

ОС общего назначения для платформы “Эльбрус” представлены двумя системами: стандартной для рабочей станции “Эльбрус-401РС” ОС Эльбрус и разработанной “Базальт СПО” Alt Linux 9.0. Рассмотрим каждую из них подробнее.

Операционная система “Эльбрус” разработана АО МЦСТ для процессоров архитектуры “Эльбрус 2000” и “Эльбрус-90микро”, основана на ядре Linux, в целом базируется на дистрибутиве Debian [6]. В связи с тем, что для оригинальной архитектуры Эльбрус-2000 необходимы соответствующие механизмы управления прерываниями, сигналами, виртуальной памятью, процессами, синхронизацией – то есть почти всеми основными механизмами ОС, было принято решение о разработке данной системы. Вместе с системой поставляется около полутысячи пакетов различного ПО, в том числе серверного. В системе инсталлирован офисный пакет LibreOffice, веб-браузер Firefox и почтовый клиент Thunderbird и другие.

Перед описанием процесса установки систем на рабочую станцию, следует рассказать об особенностях программы начальной загрузки данной аппаратной платформы.

Программа начальной загрузки решает задачу базовой самодиагностики вычислительной машины с последующим запуском операционной системы [7]. На данной платформе интерфейс программы управления конфигурацией не похож на BIOS x86, а взаимодействие организовано в текстовом режиме. С момента включения рабочей станции и до появления первых символов на экране дисплея проходит примерно 20-25 секунд, после чего можно видеть детали процесса инициализации. Есть возможность просмотреть этот процесс с самого начала – нужно лишь иметь ещё один компьютер, соединённый с рабочей станцией посредством интерфейса RS-232.

Для того, чтобы прервать автоматическую загрузку ядра системы, необходимо нажать клавишу “Пробел” при появлении сообщения [8]:

```
Autoboot in 03 sec. PRESS SPACE TO DISABLE IT.
```

Далее появятся следующие сообщения:

```
Key pressed. Autoboot canceled.
```

```
CPU#00: Starting menu.
```

BOOT SETUP

```
Press command letter, or press 'h' to get help
```

```
:
```

Данное меню позволяет просмотреть и изменить параметры загрузчика. Нажатие клавиши “~”, переводит пользователя в командную строку, где возможно скорректировать различные настройки машины, от установки системного времени до задания режимов работы периферийных контроллеров и системной шины. Все произведённые изменения сохраняются автоматически, а для сброса параметров можно использовать перемычку (джампер) на материнской плате рабочей станции.

Нажатие клавиши “h” выведет подсказку в виде списка доступных команд:

- d** — show Disks and partitions (показать диски и разделы);
- c** — Change boot parameters (изменить параметры загрузки);
- u** — show cUrrent parameters (показать текущие параметры);
- m** — save params to NVRAM (сохранить параметры в NVRAM);
- b** — start Boot.conf menu (запустить меню Boot.conf).

При нажатии на клавишу “d” будет выведена конфигурация запоминающих устройств машины, например в следующем виде:

```
CPU#00: Drive [2]: SATA – PCI BUS[1]:DEV[3]:FUNC[0], MCST SATA COMBINED Port
```

```
[0] – KINGSTON SMS200S3120G
```

```
CPU#00: Partition [0]: Linux EXT2;
          U:246194e7-0512-4db3-a821-cbcbe3c92c38 L:'''
CPU#00: Partition [1]: Linux swap
CPU#00: Partition [3]: Extended
CPU#00: Partition [4]: Unknown file system type
CPU#00: Drive [10]: ATAPI device
```

Таким образом, консоль отображает устройства, с которых можно продолжить загрузку. При нажатии клавиши "с" будет выведено меню изменения параметров загрузки:

CHANGE BOOT PARAMETERS

```
Current Settings:
drive_number: '2'
drive label:  '*'
partition_number: '0'
file system id: '07bde958-ec62-492e-933c-17334bb02da2'
command_string:
filename:
initrdfilename:
autoboot in:  '10'
```

To advance to next setting press ENTER. To skip setting press ESC

В данном меню можно изменить параметры загрузки операционной системы. Для указания места расположения программы установки ОС необходимо сменить номер `drive_number` и `partition_number` (в том случае, если установочный образ записан на определённый раздел накопителя). Остальные настройки не заполняются.

После настройки загрузки в основном меню необходимо последовательно нажать клавиши "b", "Tab" и "Enter", таким образом начнётся загрузка инсталлятора. В процессе выводятся сообщения в следующем виде:

```
boot# install
CPU#00: Label 'install' found, loading parameters
CPU#00: Search drive and partition by label or uuid succeed

Trying to load and start image with following parameters:
drive_number:  '10'
drive label:
partition_number: '0'
file system id:
command_string: 'hardreset fastboot live automatic=method:cdrom'
filename:      '/alt0/vmlinux.0'
initrdfilename: '/alt0/full.cz'
```

Далее управление передаётся программе установки ОС.

Процесс установки ОС Эльбрус уже описан на многих ресурсах, поэтому в рамках данной статьи этот процесс описываться не будет.

Второй системой общего назначения рассматривается Alt Linux 9.0. Данный дистрибутив был разработан для серверов «Эльбрус-4.4» и персональных компьютеров «Эльбрус 401-PC» и «Эльбрус 801-PC» [9]. Дистрибутив содержит операционную систему и набор прикладных программ, как для серверов, так и для рабочих станций. Программы из этого набора работают в основном режиме процессора, этим обеспечивается высокая производительность.

Основой разработки выступил отечественный репозиторий Sisyphus, который обеспечивает полный жизненный цикл дистрибутивов, созданных на его базе. Текущая ветка данного репозитория (p9_e2k) содержит более 11700 исходных пакетов, среди них рабочие окружения Xfce, LXQt, MATE, офисный пакет LibreOffice, системы инициализации sysvinit и systemd, различные средства разработки, серверные пакеты от openssh и nginx до samba-DC и PostgreSQL, программное

обеспечение для образовательных целей и многое другое. Репозиторий Sisyphus расширяется и обновляется силами разработчиков. Это положительное отличие Alt Linux от ОС Эльбрус, поскольку для второй подобного репозитория не существует.

Установка данного дистрибутива осуществляется с помощью внешнего DVD-ROM привода, т.е. предварительно необходимо записать установочный образ на оптический диск и поместить его в привод. Далее в загрузчике найти номер привода оптических дисков в списке устройств и настроить загрузку с него.

Если в процессе записи образа всё прошло корректно, то после выбора загрузки с DVD-ROM, управление будет передано программе установки. В отличие от ОС Эльбрус, инсталлятор Alt Linux 9.0 имеет графический интерфейс.

В начале предлагается выбрать основной язык программы установки и устанавливаемой операционной системы. Также на этом этапе выбирается раскладка клавиатуры. Далее необходимо выбрать страну и часовой пояс, присутствует опция хранения времени в BIOS по Гринвичу, текущее время устанавливается автоматически, но можно изменить его вручную. Так как Alt Linux устанавливается второй системой чекбокс “Хранить время в BIOS по Гринвичу” снимается.

Следующее окно предлагает настроить таблицу разделов. Ввиду того, что на твердотельном накопителе установлена ОС Эльбрус, создадим разделы на жестком диске. Удаляем размеченный для ОС Эльбрус раздел /home который занимал весь жесткий диск, первым разделом создается /boot для Alt. Разметим корневой раздел, на оставшемся месте создадим раздел /home на всё оставшееся на диске место.

После подтверждения настроек таблицы разделов, программа установки предложит выбрать дополнительные приложения для установки, среди которых присутствуют офисные приложения, приложения для работы с сетями и Интернетом, мультимедиа и средства виртуализации.

Далее начинается инсталляция операционной системы. По окончании установки имеется возможность настроить сеть. В последующих шагах настраиваются пароль системного администратора (суперпользователя), его можно создать автоматически, и пароль пользователя. После этого установка завершается.

Приведём таблицу разделов после установки обеих ОС:

```
Disk /dev/hdc: 16.0 GB, 15988654080 bytes
215 heads, 63 sectors/track, 2305 cylinders, total 31227040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x05c21836
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hdc1		2048	31227039	15612096	83	Linux

```
Disk /dev/sda: 120.0 GB, 120034123776 bytes
255 heads, 63 sectors/track, 11593 cylinders, total 234441610 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x1fb40722
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	2048	2099200	1048576+	83	Linux
/dev/sda2		2099201	10487889	4194304+	82	Linux swap
/dev/sda3		10487810	192494509	91003350	83	Linux
/dev/sda4		192494510	234441647	20973596	5	Extended
/dev/sda5		192496558	213468078	10485760+	83	Linux
/dev/sda6		213470127	234441647	10485760+	83	Linux

Disk /dev/sdb: 1000.2 GB, 1000201006016 bytes
 255 heads, 63 sectors/track, 121601 cylinders, total 1953525160 sectors
 Units = sectors of 1 * 512 = 512 bytes
 Sector size (logical/physical): 512 bytes / 4096 bytes
 I/O size (minimum/optimal): 4096 bytes / 4096 bytes
 Disk identifier: 0x5d1dfd9f

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1	*	2048	4194303	2096128	83	Linux
/dev/sdb2		4194304	20973567	8389632	82	Linux swap
/dev/sdb3		20973568	87687167	33356800	83	Linux
/dev/sdb4		87687168	601733119	257022976	5	Extended
/dev/sdb5		87689216	601733119	257021952	83	Linux

Разделы /dev/sda1 /dev/sdb1 являются /boot разделами для обеих систем. /dev/sda2 и /dev/sdb2 – swap-разделы. /dev/sda3 и /dev/sdb3 представляют корневые разделы. Ранее раздел /home для ОС Эльбрус занимал весь жесткий диск, поэтому на оставшемся после установки Alt Linux месте разметим раздел /home и подключим его к обеим системам.

Теперь возникает вопрос о возможности удобного переключения между установленными системами. К сожалению, в текущем варианте загрузки не предусмотрено механизма, подобного GRUB, но выход есть. В файле конфигурации загрузчика (boot.conf) для ОС Эльбрус создаётся запись с параметрами загрузки системы, добавив в этот файл параметры для ОС Alt Linux 9.0 получим возможность выбирать ОС в которую хотим загрузиться.

Создадим метку с конфигурацией для Alt Linux. Для этого допишем в файл boot.conf ОС Эльбрус информацию об Alt Linux. Укажем название нашей метки в переменной label (например, "alt"), введём номер запоминающего устройства (номер 6) и номер раздела /boot (номер 0) второй системы. Далее необходимо заполнить имена файлов образа и initrd. Найти их можно следующим образом: загрузиться в ОС Эльбрус, с помощью команды mount подключить /boot раздел Alt Linux, открыть данный раздел в терминале и просмотреть его содержимое. Конечный вид файла boot.conf может быть следующим:

```
default=elbrus
timeout=3
```

```
label=elbrus
    drive=4
    partition=0
    image=/image-4.9.0-2.7-e2s
    initrd=/initrd-4.9.0-2.7-e2s.img
    cmdline=root=/dev/sda3 console=ttyS0, 115200 iommu=0
```

```
label=alt
    drive=6
    partition=0
    image=/image-4.9.170-elbrus-def-alt4.3.1
    initrd=/initrd-4.9.170-elbrus-def-alt4.3.1.img
    cmdline=root=/dev/sdb3 console=ttyS0, 115200 iommu=0
```

Теперь в процессе запуска рабочей станции можно прервать автоматическую загрузку, в меню Boot Setup нажимаем кнопку "b", вводим имя метки (для приведённого выше файла либо "elbrus", либо "alt") и загружаемся в соответствующую метке ОС. Изменив параметр default можно выбрать метку

для загрузки по умолчанию. Параметр timeout отвечает за количество секунд, в течении которых можно прервать автоматическую загрузку.

Проведение описанных манипуляций, позволило получить на рабочей станции две операционных системы, которые можно относительно удобно запускать. В связи с этим, появилась возможность изучать новую операционную систему для данной платформы.

Библиографический список

1. ОС MCBC. Мобильная система вооруженных сил. URL: <https://www.softdefence.ru/item/6-msvs> (дата обращения: 28.03.2020).
2. Мандатная модель ОС MCBC 3.0. URL: <https://habr.com/ru/post/133247/> (дата обращения: 28.03.2020).
3. Astra Linux – Astra Linux® Special Edition. URL: <https://astralinux.ru/products/astra-linux-special-edition/> (дата обращения: 28.03.2020).
4. «СВД Встраиваемые Системы» Продукты. URL: <https://www.kpda.ru/products/> (дата обращения: 29.03.2020).
5. ОСПВ «БагРОС-4000». URL: <https://www.imaxai.ru/zosrv-bagros-4000> (дата обращения: 29.03.2020).
6. Обзор и сравнительное тестирование ПЭВМ «Эльбрус 401-РС». Часть вторая — программное обеспечение. URL: <https://habr.com/ru/post/371723/> (дата обращения: 29.03.2020).
7. Эльбрус/пнс. URL: <https://www.altlinux.org/Эльбрус/пнс> (дата обращения: 29.03.2020).
8. Эльбрус/загрузчик. URL: <https://www.altlinux.org/Эльбрус/загрузчик> (дата обращения: 29.03.2020).
9. Альт. Рабочая станция 9. URL: https://www.altlinux.org/Альт_Рабочая_станция_9 (дата обращения: 29.03.2020).

УДК 004.92

Т. Р. Мустафин

студент кафедры вычислительных систем и сетей

Н. Н. Решетникова – кандидат технических наук, доцент – научный руководитель

О СПОСОБАХ СКАНИРОВАНИЯ ПРЕДМЕТОВ ДЛЯ ПОЛУЧЕНИЯ 3D-МОДЕЛЕЙ

Сканирование реальных объектов для получения их 3D моделей используется во многих сферах, таких как, образование, архитектура, медицина, игровых индустриях и многих других. Существует достаточное количество разновидностей сканеров и применяемых в них технологий. Например, технология RealSense от компании INTEL, портативные сканеры-приставки к смартфонам, различное профессиональное оборудование, а также технологии фотограмметрии в виде программного обеспечения для смартфона.

Актуальность проведения эксперимента заключается в оценке качества полученной 3D модели предмета и возможностей оборудования, применяемого для сканирования. Стоит ли покупать дорогостоящее оборудование для 3D сканирования или можно обойтись обычным смартфоном с камерой, имеющей высокое разрешение (например, 12Мп)?

Для сканирования предметов с целью получения их 3D моделей были выбраны производственный сканер Artec Space Spider, работающий под управлением программы ArtecStudio 11 [1], и смартфон Xiaomi Mi Note3 [2], который для получения 3D модели взаимодействовал с программой Qlone открытого доступа [3].

Сканер Artec Space Spider и программа ArtecStudio 11 это набор инструментов для систем автоматизированного проектирования, способный передавать в 3D сложную геометрию поверхности с достаточно большой точностью и высоким качестве. Предназначен для работы с предметами небольшой величины, например, компьютерные платы, монетки или ключи. Готовую 3D модель можно экспортировать в разные форматы для дальнейшего использования, например, редактирования и сокращения размерности 3D модели в редакторе Autodesk 3DsMax и дальнейшего использования при разработки интерактивных приложений. Встроенного процессора нет, используется процессор персонального компьютера. Рекомендуемые процессоры серии Intel Core i5 или i7.

Характеристики: разрешение камеры сканера – 1.3 Мп; масса сканера: 0,8 кг; основные получаемые из программы ArtecStudio форматы файлов: STL, OBJ, PLY, BTX и другие; величина сканируемого предмета от 5 мм;

Смартфон Xiaomi Mi Note3 – это мобильное устройство, которое используется каждый день. Программа Qlone использует для ориентации акселерометр телефона и специальную подложку QloneMat.

Характеристики: чипсет смартфона Qualcomm Snapdragon 660 с сопроцессором Adreno 512; разрешение камеры – 12 Мп; вес смартфона: 163 грамма; получаемые из программы Qlone форматы файлов: OBJ, STL, X3D, PLY, возможность загрузки на сайт 3D моделей SketchFab[...]. Габариты объекта сканирования ограничиваются размерами подложки QloneMat, чьи размеры составляют 20x20 см, а так же размерами виртуального купола. В качестве предмета для сканирования выбрана модель SubaruWRX, представленная на рис. 1.

Съемка объекта на ArtecSpider несложна. Для начала необходимо подключить сканер с помощью двух кабелей из комплекта компьютеру и электросети. Далее создается новый проект в программе ArtecStudio, открывается окно предпросмотра, где и происходит отрисовка данных, получаемых с камеры сканера. Диапазон-шкала дальномера (рис. 2) устанавливается в зеленую зону, и только тогда можно начинать записывать координаты вершин объекта сканирования. Кнопка начала записи находится на ручке сканера и продублирована кнопкой рабочем окне программы ArtecStudio.

После того как процесс записи запущен, нужно поворачивать модель на вращающейся подставке, так же удерживая значения на шкале в зеленой зоне. Результат сканирования модели автомобиля представлен на рис. 3. Затем в окне редактирования программы ArtecStudio следует совместить полученные сканы.



Рис. 1. Выбранная модель

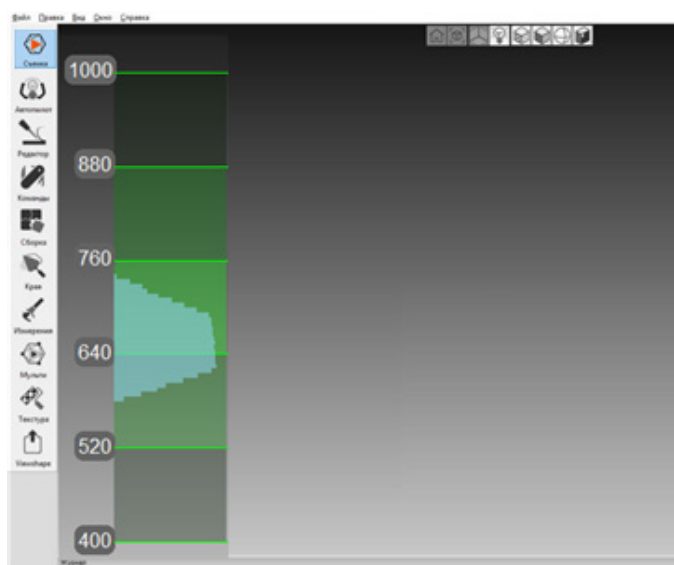


Рис. 2. Диапазон-шкала дальномера в окне программы ArtecStudio

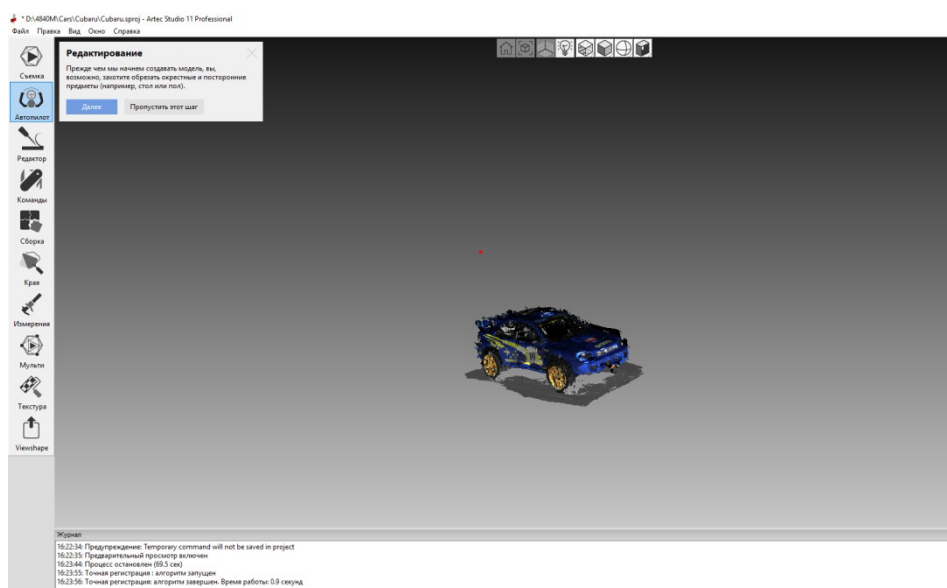


Рис. 3. Результат записи объекта

Далее можно запустить мастер обработки отсканированной 3D модели, называемый «Автопилотом». Он в автоматическом режиме удалит большую часть мусора в виде бликов, отражений и

прочих артефактов (рис.4). После завершения процесса нужно вручную удалить подложку под 3D моделью автомобиля.



Рис. 4. 3D-модель автомобиля после обработки в окне программы ArtecStudio

Далее 3Dмодель, отделенная от вспомогательной подложки экспортируется в один из доступных форматов для дальнейшей обработки в 3D-редакторе.

Для сканирования объекта на смартфон необходимо скачать программное приложение Qlone[3] из магазина Google Play Store. Приложение бесплатное и находится в свободном доступе. Так же нужно распечатать подложку QloneMat с калибровкой. Взять ее можно на официальном сайте приложения.

После открытия приложения появляется главный экран, на котором будут выводиться все сканированные объекты. В правом нижнем углу экрана нужно нажать белую кнопку «+». Открывается окно предпросмотра, смартфон следует направить на подложку с установленной на нее моделью автомобиля и вокруг объекта сканирования появляется виртуальный купол. Чтобы начать запись координат объекта необходимо нажать кнопку в правом верхнем углу экрана.

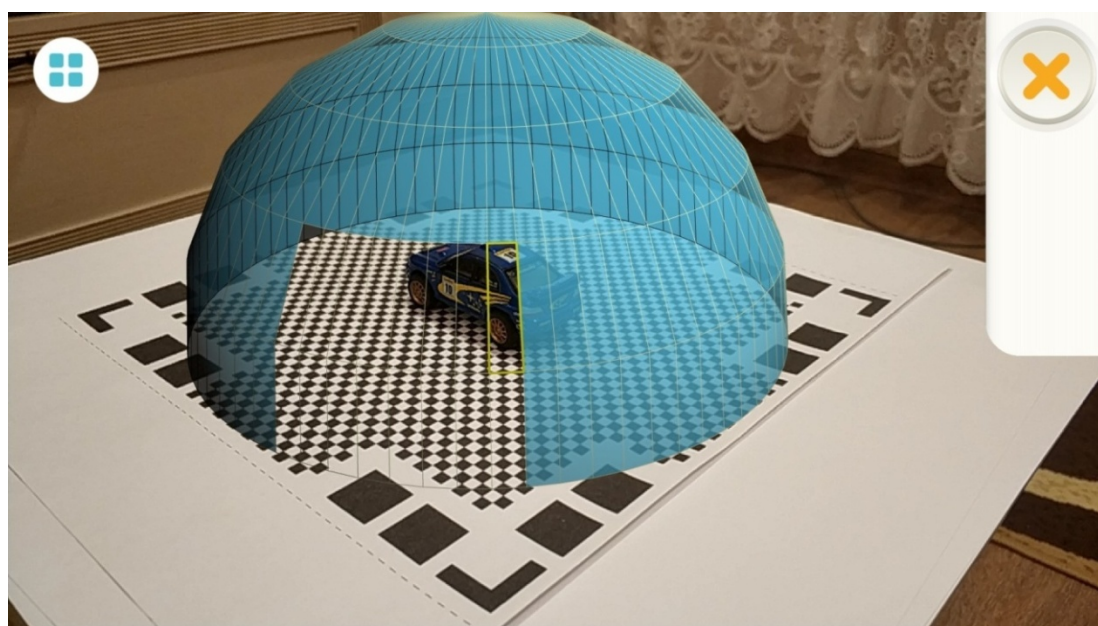


Рис. 5. Сканирование в программе Qlone

В процессе сканирования нужно водить курсор с помощью смартфона по фрагментам виртуального купола. С каждым перемещением курсора приложение обесцвечивает отсканированный фрагмент, параллельно делая снимки модели автомобиля с разных ракурсов. После того как обесцветятся последние фрагменты, запись остановится автоматически. Приложение начнет обработку фотоснимков, после чего скомпонирует и выведет на экран 3D модель объекта сканирования (модель автомобиля).

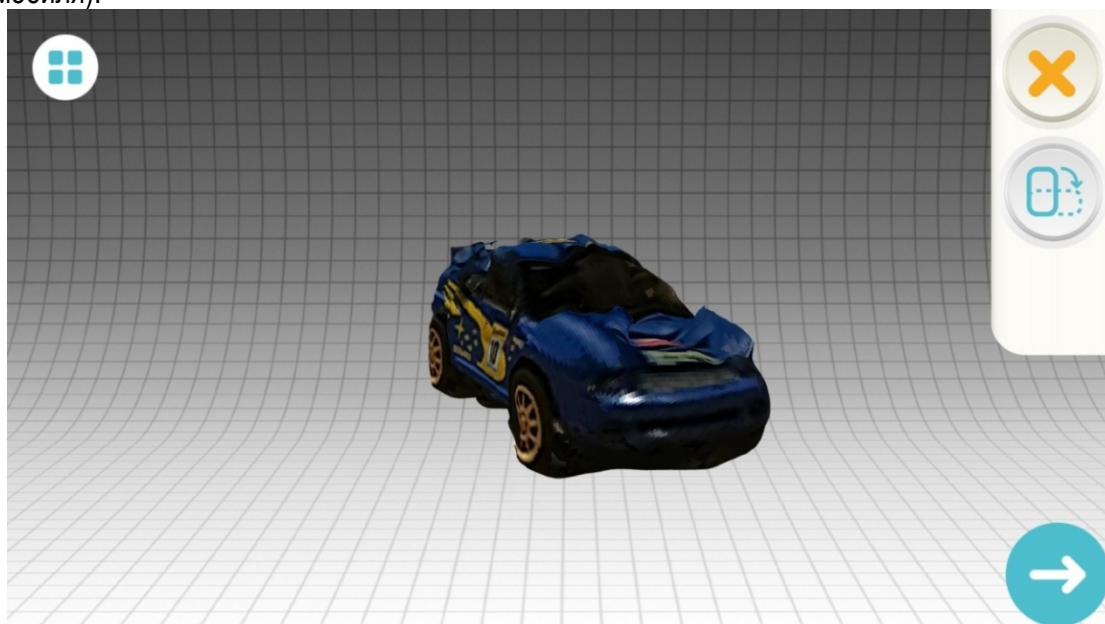


Рис. 6. Составленная 3D-модель автомобиля



Рис. 7. Результат обработки двух этапов сканирования

Для увеличения точности 3D модели, можно отсканировать ее еще раз в другом режиме, перевернув объект на бок. После окончания процесса, приложение сначала обработает только что снятый материал, и затем будет объединять оба этапа сканирования.

Далее результат можно, экспортировать в один из нескольких форматов или загрузить на сайт 3D моделей Sketchfab.com [4].

Результат проведенного эксперимента показал, что 3D модель, подготовленная с помощью сканера Artec Space Spider, получилась качественнее, чем на смартфон. Вызвано это может быть множеством факторов при использовании смартфона, таких как блики, неровный свет, дергание камеры из-за нестабильного положения рук. Все это можно частично откорректировать, используя студийный свет, электронный стабилизатор для телефона. Съемка модели на смартфон скорее больше подходит для домашнего использования.

Плюсы и минусы обоих способов

Плюсы съемки на телефон: невысокая стоимость, удобство, высокая мобильность.

Минусы: не высокое качество модели; трудности переноса на компьютер.

Плюсы съемки на Artec Spider: высокое качество модели; хорошая точность съемки; возможности обработки модели сразу после съемки в программе ArtecStudio.

Минусы: высокая стоимость оборудования; мобильность ограничена длиной проводов; высокие требования к конфигурации компьютера.

Библиографический список

1. Artec 3D: сайт. URL: <https://www.artec3d.com/ru> (дата обращения: 11.04.2020).
2. Xiaomi Mi Note3: сайт. URL: <https://mobile-review.com/review/xiaomi-mi-note-3-fl.shtml> (дата обращения: 11.04.2020).
3. Qlone: сайт. URL: <https://www.qlone.pro/> (дата обращения: 11.04.2020).
4. Sketchfab: сайт. URL: <https://sketchfab.com/features/> (дата обращения: 11.04.2020).

УДК 004.058:004.93

М. Э. Назорных

студент кафедры вычислительных систем и сетей

А. А. Антонов

студент кафедры вычислительных систем и сетей

А. Н. Быков

студент кафедры вычислительных систем и сетей

Н. В. Кучин – кандидат технических наук, доцент – научный руководитель

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ УПРАВЛЕНИЯ ПОЛЬЗОВАТЕЛЬСКИМ ИНТЕРФЕЙСОМ КОМПЬЮТЕРА И ПРИЛОЖЕНИЙ С ПОМОЩЬЮ ЖЕСТОВ КИСТЕЙ РУК

Введение

В последние десятилетия наблюдается бурное развитие в сфере искусственного интеллекта, его использования в научной среде и повседневной жизни. Это обусловлено возможностью выполнять функции, которые считаются прерогативой человека, ускоряя и упрощая процесс работы той или иной системы. Именно поэтому большое число стартап-проектов используют данную технологию.

Одним из перспективных направлений является человеко-машинное взаимодействие. Во многих научно-фантастических литературных и кинематографических произведениях еще прошлого века уделяется внимание перспективам управления компьютером посредством жестов кистей рук. Но сегодня эта идея уже не кажется фантастикой и может иметь практическую реализацию с широким спектром применения. Например, для управления робототехническими изделиями или пользовательским интерфейсом компьютера. Это позволяет человеку без использования периферийных устройств, таких как мышь и клавиатура, производить аналогичные функции, используя собственные руки. Что, в свою очередь, позволяет ускорить работу, делая ее более комфортной.

Зачастую такие технологии основаны на использовании дорогостоящей системы Kinect, сложнонастраиваемых стереокамер и других замысловатых технических систем. Именно поэтому была поставлена задача, обойдя эти ограничения, реализовать программное обеспечение которое при минимальных технических затратах и без использования специального оборудования, способно выполнять все задачи компьютерного зрения по поиску кистей рук на изображении и классификации их жестов. Полученный опыт будет использован при разработке программного обеспечения для распознавания речи глухонемых.

Анализ методов поиска и классификации объектов на изображении

Поставленная задача решалась постепенно путем анализа различных методов и их сравнения. Сначала, необходимо было “научить” компьютер искать кисть руки и распознавать ее жест. Первым шагом на пути к этому был выбор языка программирования, который остановился на Python [1]. А для обработки изображений была выбрана библиотека компьютерного зрения с открытым кодом – OpenCV.

Совместное использование Python и OpenCV позволяет быстро реализовать базовые функциональности для анализа и работы системы в целом. Python позволяет быстро разрабатывать прототипы программных продуктов, а также дает возможность легко подключать и использовать библиотеки. OpenCV же имеет огромное количество алгоритмов для обработки и предобработки изображений и видео потока.

Для начала работа проводилась с выделенной областью экрана. Для этого захватывался задний фон, и с помощью алгоритмов OpenCV производилась бинаризация полученного изображения. После чего, в выделенную область помещалась рука и также бинаризовалась. Далее, производилось сравнение фона и самого изображения с рукой, посредством вычитания изображений. В результате получалось бинаризованное изображение руки, которое в дальнейшем подвергалось анализу.

На полученном изображении находились, так называемые, дефекты выпуклости, которые отмечаются точками. После этого рассчитывалось расстояние между ними, их количество, и на основе полученных данных классифицировался жест руки.

Результаты оказались неудовлетворительными: был получен низкий уровень классификации жестов. Наличие других динамических объектов в области распознавания сводил результат к минимуму. Также был сделан вывод, что данный метод не позволяет производить поиск руки и ее обнаружение вне выделенной области, что в свою очередь критично для поставленной задачи. Поэтому было необходимо искать новые пути решения.

Для устранения проблемы классификации было решено использовать метод опорных векторов (SVM) [2], который позволяет с более высокой точностью классифицировать объекты, а для поиска руки на всем изображении – каскад Хаара.

Каскад Хаара – один из основополагающих методов детектирования объектов с большой скоростью работы. При обучении на вход классификатора подается набор позитивных изображений с предварительно выделенной областью, содержащей искомый объект, и набор негативных изображений. Далее, выполняется перебор выборки и расчет значений признаков Хаара. Полученные данные сохраняются в файле в формате xml. После, полученный классификатор можно использовать для обнаружения объекта на изображении в реальном времени.

SVM – это набор алгоритмов обучения с учителем, использующихся для классификации объектов. В качестве «Учителя» выступает набор изображений, являющихся примером жестов кистей рук. На основе исходных данных формировались признаки для классификации объектов. Они были получены путем перевода исходных векторов в пространство более высокой размерности и поиском разделяющей гиперплоскости.

Совместное использование вышеупомянутых методов, позволяет с высокой точностью классифицировать объект. Однако, для получения требуемого уровня точности поиска и классификации, необходимо составить очень большую выборку. В данном случае она составила около двадцати тысяч примеров. Также полученная система сильно зависит от внешних факторов: свет, движение камеры, угол поворота объекта и его масштаб, что не позволяет производить детектирование в достаточной мере.

Пример работы программы, основанной на связке признаков Хаара и SVM продемонстрирован на рис. 1.

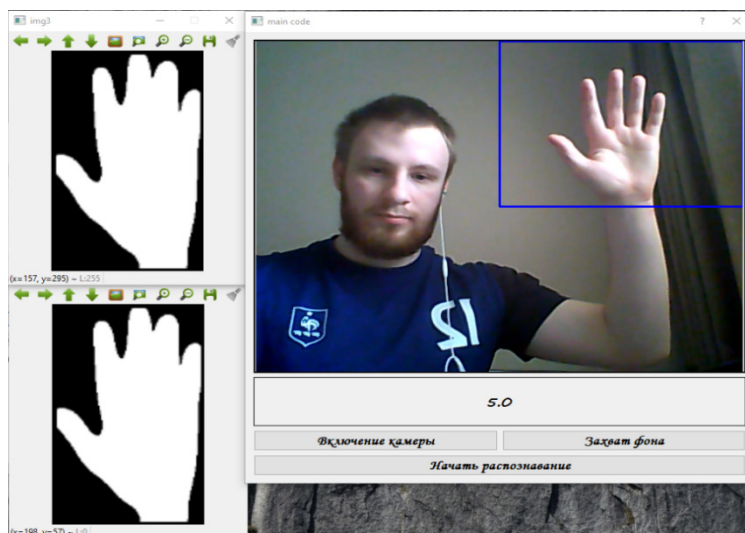


Рис. 1. Работа программы, основывающейся на использовании признаков Хаара и SVM

Рассмотренные методы не смогли дать необходимые результаты, из-за чего было выбрано прибегнуть к использованию нейросети [3]. На данный момент существует множество библиотек для машинного обучения. Каждая из них имеет ряд преимуществ, в зависимости от области применения. Для данной задачи была выбрана библиотека с открытым кодом Tensorflow. Она известна своим удобством использования и надежностью.

После этого, был составлен файл с параметрами обучения нейронной сети и подготовлена выборка для ее тренировки. Для начала была взята выборка в количестве 37 фотографий руки. После обучения, даже при столь малой выборке, нейронная сеть показывала высокий уровень детектирования объекта. Это и являлось определяющим фактором дальнейшей работы с использованием методов глубокого обучения посредством TensorFlow.

Следующим шагом было необходимо удостовериться в том, что помимо качественного детектирования, нейросеть также способна хорошо классифицировать объекты.

Для этого была составлена новая, более объемная, выборка для определения 10 жестов кистей рук. На каждый жест приходилось по 60 фотографий, то есть в общей сумме – 600 примеров.

Результатом стал файл, содержащий весовые коэффициенты нейронной сети, способной работать при разных освещениях с несколькими объектами в кадре. Новый подход полностью решил проблемы поиска и распознавания рук, при этом использовалась гораздо меньшая выборка.

Пример работы приложения, использующего обученную нейросеть продемонстрирован на рис. 2.

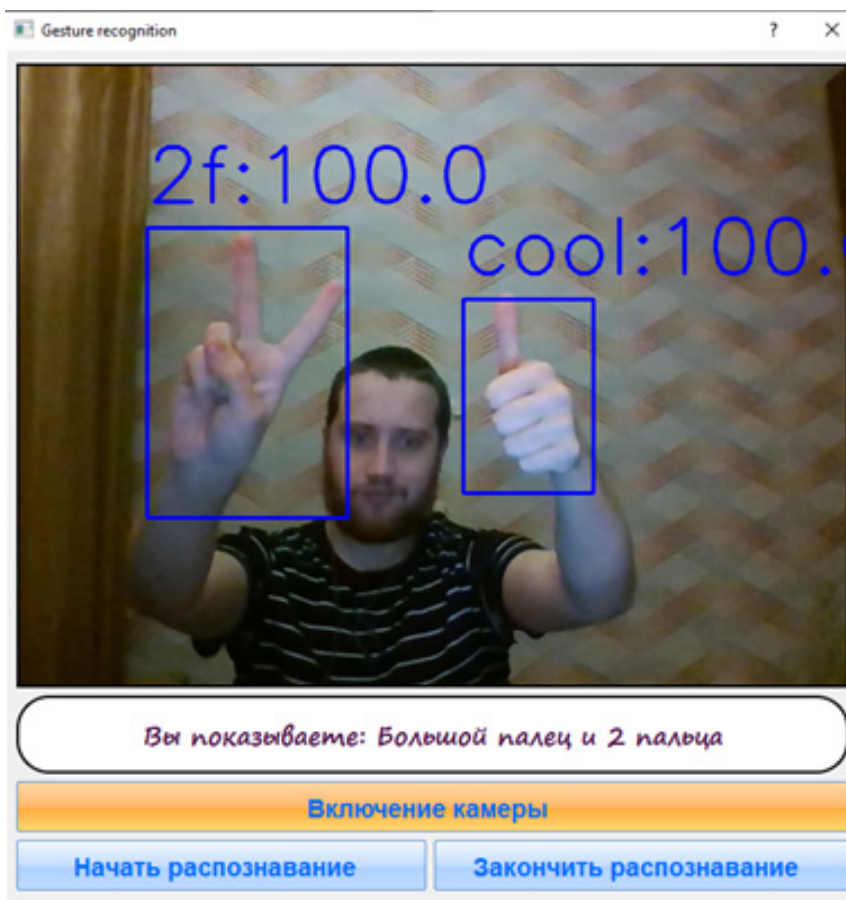


Рис. 2. Работа разработанного приложения, использующего нейросеть

Разработка программного обеспечения для управления пользовательским интерфейсом

Получив обученную нейронную сеть, можно было переходить к выполнению основной задачи. Для демонстрации работы было принято использовать программу Sony Vegas, где управление интерфейсом программы могло производиться с помощью жестов и движений рук. Это позволяет заменить использование периферийных устройств ввода. Библиотека компьютерного зрения OpenCV обеспечивает поиск отдельных элементов интерфейса программы, это значительно ускоряет работу с ней. Процесс поиска необходимой функции основывался на анализе экрана и поиска на нем заранее заданных изображений иконок функций-шаблонов.

Пример работы приложения продемонстрирован на рис. 3, на котором можно видеть, что при показе жеста “2 пальца” завершается перемотка видео.

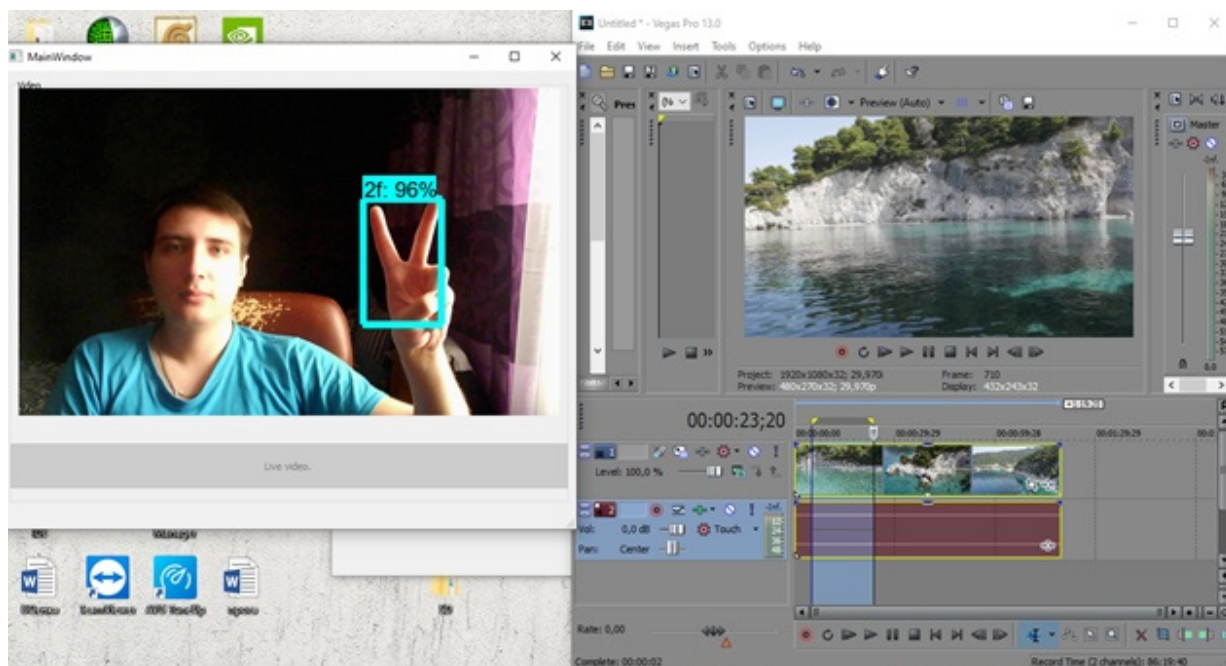


Рис. 3. Работа разработанного приложения

В конечном счете, была достигнута поставленная цель – приложение способное производить анализ изображения на наличие кистей рук в кадре и классификацию их жестов, что позволяет управлять любой пользовательской программой.

Заключение

Разработанное программное обеспечение соответствует поставленным требованиям и для его работы достаточно обычной веб-камеры, имеющейся у большинства пользователей. Также следует отметить универсальность разработки – каждый может легко подстроить программу для управления интересующим приложением.

Полученный опыт, при разработке данного приложения, будет использован для развития и модернизации данного проекта. Так, например, следующим шагом является распознавания речи глухонемых.

Библиографический список

1. Саммерфилд М. Программирование на Python 3: подробное рук. М.: Символ, 2016. 608 с.
2. Вьюгин В. В. Математические основы теории машинного обучения и прогнозирования. М.: 2013. 387 с.
3. Комарцова Л. Г., Максимов А. В. Нейрокомпьютеры. М.: МГТУ им. Н. Э. Баумана, 2004. 400 с.

УДК 621.389

С. А. Ненашев

студент кафедры вычислительных систем и сетей

Е. О. Лукашова

магистрант кафедры проблемно-ориентированных вычислительных комплексов

М. Б. Сергеев – доктор технических наук – научный руководитель

СПОСОБ ОПЕРАТИВНОГО МОНИТОРИНГА С ПОМОЩЬЮ ДВУХПОЗИЦИОННОЙ ОПТИЧЕСКОЙ СИСТЕМЫ МАШИННОГО ЗРЕНИЯ

Введение

Оптико-электронные локационные системы в настоящее время применяются во многих сферах современной жизни, например, в медицинской технике, системах контроля дорожного движения, в спутниковых системах, системах мониторинга окружающего пространства, системах безопасности муниципальных учреждений, а также в системах экологического и оперативного мониторинга и др.

Способы экологического и оперативного мониторинга, реализуемые на Беспилотный летательный аппарат (БПЛА) с целью предупреждения о чрезвычайных ситуациях. Реализуемые способы могут быть применены для мониторинга охраняемых объектов занимаемые большую площадь наблюдения, а также труднодоступные зоны или зоны представляющие опасность для жизни (с повышенным уровнем радиации, а также химических и биологических веществ).

С каждым годом наблюдается рост к требованиям подобных современных систем с высоким разрешением порядка 12К, 16К и более. Кроме того, следует увеличить скорость обработки в рассматриваемых системах, реализация чего становится все более затруднительной в особенности при частоте считывания кадров порядка 50 кадров/с и более. Это налагает ряд требований к реализации подобных систем и в первую очередь это требования к производительности графического процессора и др. Кроме того, подобные системы должны обеспечивать работу в реальном масштабе времени.

В связи с этим актуальна задача разработки и совершенствования моделей и алгоритмов обработки видеoinформации для решения локационных задач [1, 2].

Для выполнения вышеуказанных требований и отладки режимов функционирования данной двухпозиционной системы машинного зрения была реализована соответствующая имитационная модель, позволяющая с одной стороны задавать положение, скорость, направление движение объектов, а с другой осуществлять обнаружение и определение координат объектов. Данная модель представляет собой программно-аппаратный комплекс, который состоит из двух видеокамер высокого разрешения, которые подключены к высокопроизводительной рабочей станции на базе NVIDIA Tesla K20X и NVIDIA Quadro K2000 для математических вычислений в Matlab.

Для реализации системы машинного зрения в программной части данного комплекса используются способы, реализующие процессы фильтрации и бинаризации сформированных изображений. Для реализации системы движения объектов реализован программный модуль имитирующий траекторию объектов.

Постановка задачи

При измерении расстояния могут применяться несколько оптико-локационных систем в комплексе [1].

Преимуществом данных методов является высокая скорость обработки, что тем самым повышает оценку принятия решения в критических ситуациях, и также точность измерений до объектов.

Недостатками оптических систем являются шумы, помехи, перекрытие объекта частичные или полное, точность измерений зависит от корректной калибровки камеры и ее установки относительно другой камеры, некоторые методы обнаружения неспособны реагировать в ситуациях не имеющих шаблона объекта, конкретный метод работает под определенную задачу.

Но основной проблемой при обнаружении объектов является обусловленным смещением, поворотом, масштабом, изменением яркости, изменение положения камеры.

При разработке подобных систем особое внимание следует уделить следующим основным этапам обработки:

- обнаружение интересующих объектов в потоке видеоданных;
- измерение координат позиции обнаруженного объекта в кадре;
- измерение координат объекта в следующем кадре;
- определение скорости объекта в потоке видеоданных;
- сопровождение в потоке видеок кадров;
- классификация объекта
- ошибка положения из-за несоответствия пикселей
- ошибка отслеживания камеры

Обычно алгоритм видео детектирования состоит из следующих шагов (рис.1):

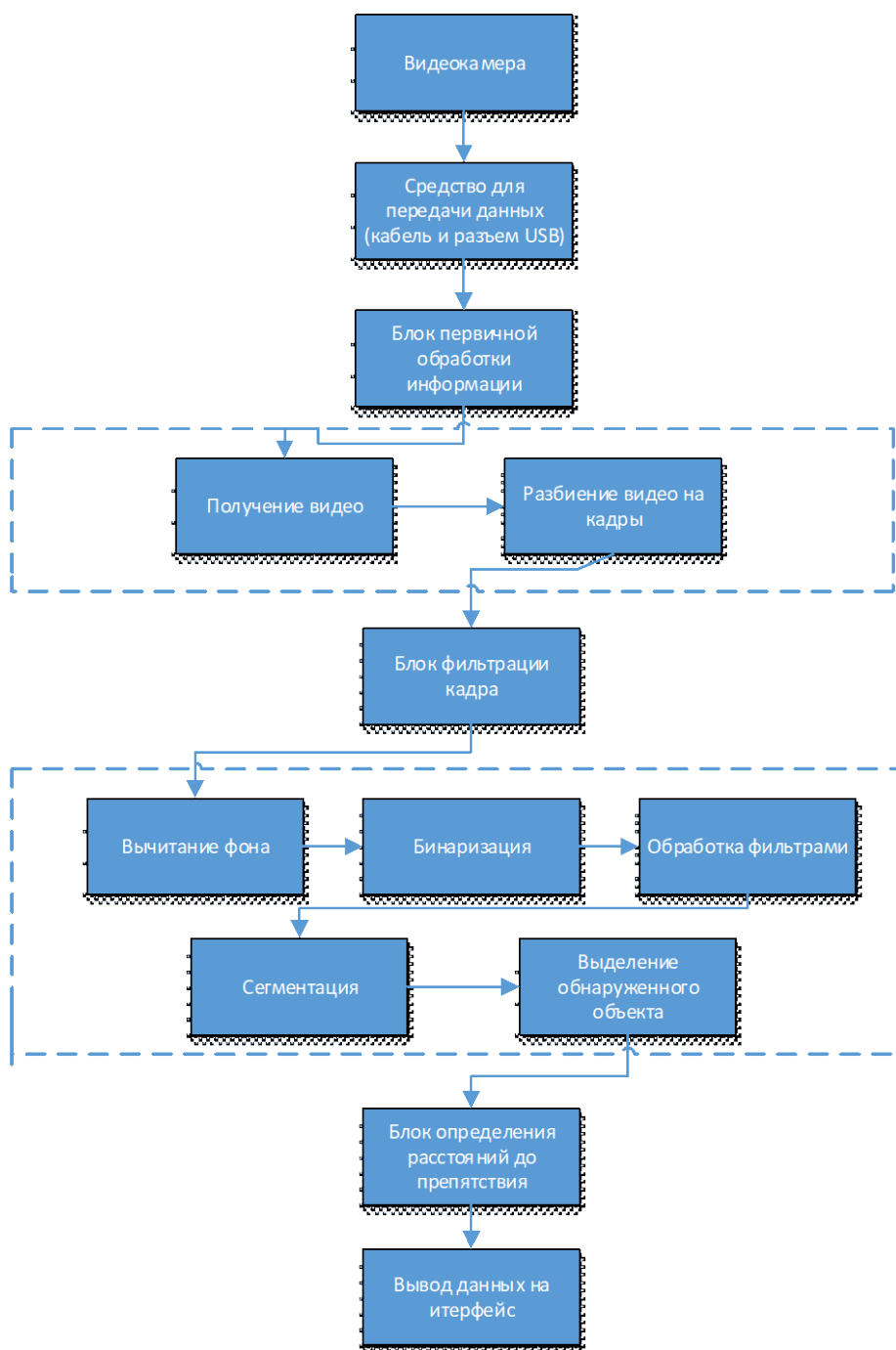


Рис. 1. Схема алгоритма оптической системы машинного зрения

Применив данный алгоритм [2–7] к каждому потоку данных с левой и правой камеры на выходе получим отметки координат объекта. Для того чтобы реализовывать поставленные задачи требуется как минимум два источника информации или так называемая стереопара.

Стереопара

Стереопара представляет информацию о глубине изображения, в частности расстояние до объекта, а так же составить трехмерную картину окружающего мира. Зачастую используются две камеры работающие синхронно, установлены строго параллельно друг другу с половинным перекрытием совместной зоны обзора.

В основе стереозрения лежит бинокулярное, т.е. то которое использует человек для оценки пространства и ее глубины. Слияние зрительных образов полученных с каждого глаза, происходит в сложнейшем физиологическом механизме зрения.

Так же измерять расстояния возможно таким геометрическим явлением как параллакс. Оно основывается на изменении угла обзора по отношению его дальности. Двигаясь влево и вправо, смотря вперед, близлежащие объекты меняют свой угол по отношению к тому, что находится прямо перед наблюдателем, быстрее, чем объекты, расположенные дальше. Из скорости изменения угла при его движении — из параллакса движения — наблюдатель может понять, насколько далеко расположен объект.

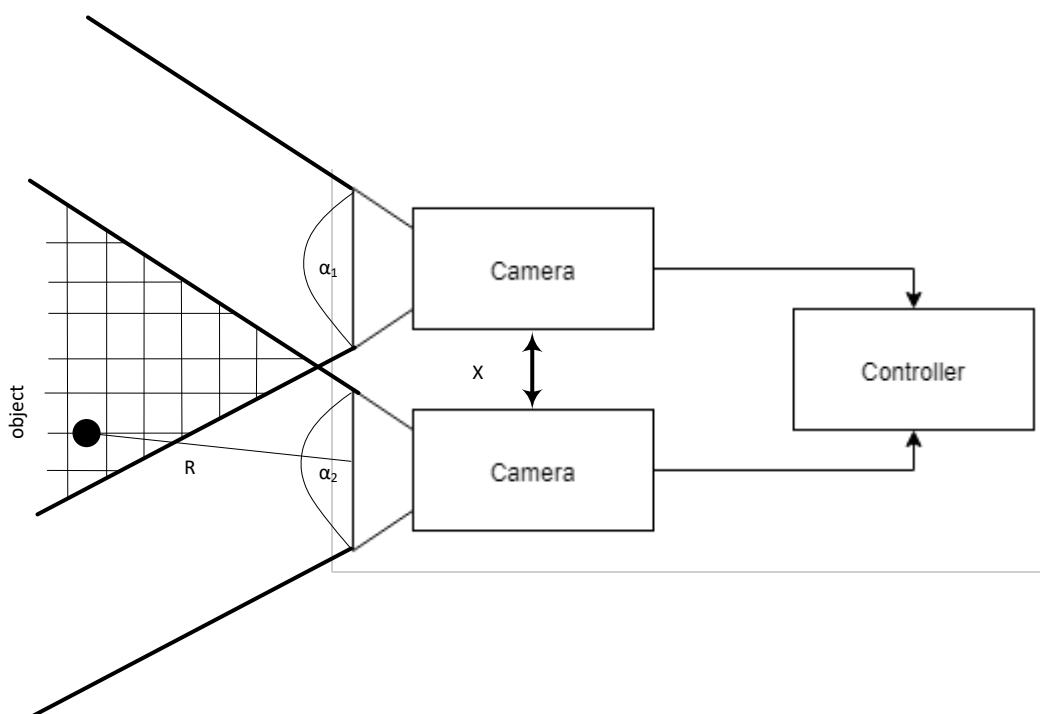


Рис. 2. Схема параллельного расположения двухпозиционно-оптической системы

Итак, зная расстояние между двумя стереопарами X и угол обзора $\alpha_{1,2}$ можно определить и расстояние до объекта R .

Большое преимущество стереопары в том, что возможно получить данные глубины без применения дорогостоящих датчиков измерения расстояния [7–9].

Измерение камеры

Измерения на основе стереокамеры выполняются путем сопоставления положения объекта (точки) на двух изображениях, а затем его использования в сочетании со знанием положений камеры для вычисления положения объекта [10]. Если сопоставление выполнено неправильно, предполагаемое положение объекта будет также неправильным. Это показано на рис. 3, здесь показана ошибка

в положении, когда положение объекта в двух камерах занимает одно место больше элемента разрешения. Результирующая погрешность, зависит от расстояния между датчиком и элементом, и эта погрешность больше в радиальном направлении, чем в тангенциальном. Эти типы несоответствий характеристик могут возникать случайно из-за временных эффектов, однако они также могут систематически возникать, если две различные функции неверно идентифицируются как одинаковые в системе камер.

Случайные несоответствия могут усредняться по времени, в то время как систематические погрешности не могут, так как они зависят от физических атрибутов наблюдаемого объекта.

Ошибка положения из-за несоответствия пикселей Ошибка положения из-за несоответствия пикселей

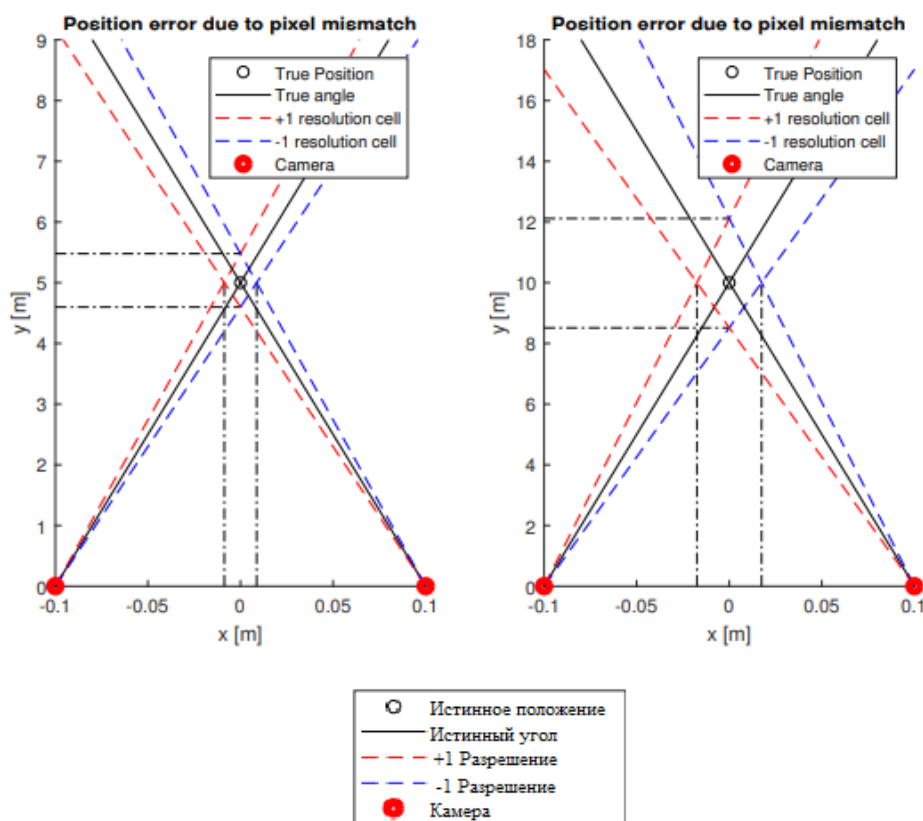


Рис. 3. Теоретическая ошибка положения, когда пиксельное соответствие между изображениями неверно

Измеренное положение объекта определяется пересечением линий, соответствующих согласованным пикселям каждой камеры. На левом графике показана ошибка положения, которая возникает, если камера не соответствует элементу с одной ячейкой разрешения на расстоянии 5 м, а справа – то же самое, когда объект находится на расстоянии 10 м.

Ошибка отслеживания камеры

Ошибка в треках моделируемой камеры предназначена для того, чтобы вести себя как реальная реализация, однако трудно оценить производительность трекера на основе стереокамеры. Поскольку влияние ошибок измерения будет уменьшено за счет фильтрации и использования нескольких объектов на целях для оценки состояния. Поэтому для треков была использована простая и легко модифицируемая модель ошибок, которая добавляет ошибки к предполагаемому положению, курсу и протяженности объекта. Эти ошибки были настроены для генерации того, что мы нашли как разумные ошибки.

Погрешность положения в радиальном направлении моделируется как функция r , радиального расстояния между датчиком и целью и k , количества последовательных обнаружений. Используемая функция

$$e_r(k,r) = br + s_{1,r}r\eta_{1,r}(k) + s_{2,r}r\eta_{2,r}(k)e^{-\xi_r k}$$

где $s_{1,r}, s_{2,r}$ и ξ_r – масштабные константы, которые используются в качестве расчетных параметров, b – постоянная смещения, а $\eta_{1,r}(k)$ и $\eta_{2,r}(k)$ – случайные величины. Константа смещения b используется для учета возможности систематических несовпадений признаков в системе триангуляции камеры. Следовательно, она уникальна и постоянна для каждой цели и берется из $N(0, \sigma_b^2)$, при первом обнаружении объекта. Случайные величины определяются как

$$\eta(k) = \begin{cases} q(k) & \text{если } k = 0 \\ (1-\alpha)\eta(k-1) + \alpha q(k) & \text{если } k \geq 1 \end{cases}$$

где α – расчетный параметр и $q(k) \sim N(0,1)$. Эти термины используются для учета случайного характера любой системы отслеживания. Используя эту модель, можно рассчитать дисперсию ошибки, и она должна соответствовать оценке дисперсии, которую реальная система отслеживания оценила бы как дисперсию.

Ошибка в тангенциальном положении моделируется аналогичным образом, однако член смещения исключается, и расчетные константы имеют более низкие значения для повышения точности оценки. Модель

$$e_t(k,r) = s_{1,t}r\eta_{1,t}(k) + s_{2,t}r\eta_{2,t}(k)e^{-\xi_t k}$$

Эти две ошибки преобразуются в декартову систему координат, которое используется в представлении состояний позиций. Это сделано в соответствии с формулой:

$$\begin{aligned} e_x(k,r) &= e_r \cos(\theta) - e_t \sin(\theta) \\ e_y(k,r) &= e_r \sin(\theta) + e_t \cos(\theta) \end{aligned}$$

где θ – угол к объекту от датчика.

Оценка пути курса объекта оценивается так же, как и тангенциальная ошибка,

$$e_\varphi(k,r) = s_{1,\varphi}r\eta_{1,\varphi}(k) + s_{2,\varphi}r\eta_{2,\varphi}(k)e^{-\xi_\varphi r}$$

где константы были масштабированы, чтобы соответствовать отслеживанию угла.

Кроме того, оценка систем камер экстенда целей содержит одно из этих «стандартных» представлений об ошибках, однако оно комбинируется с дополнительным коэффициентом масштабирования, который зависит от количества ячеек разрешения, которые объекта занимает в датчике. Таким образом, полное выражение для ошибки оценки экстенда

$$\begin{aligned} e_l &= (s_{1,l}\eta_{1,l}(k) + s_{2,l}\eta_{2,l}(k)e^{-\xi_l k})re^{-\zeta\theta_{span}} \\ e_w &= (s_{1,\omega}\eta_{1,\omega}(k) + s_{2,\omega}\eta_{2,\omega}(k)e^{-\xi_\omega k})re^{-\zeta\theta_{span}} \end{aligned}$$

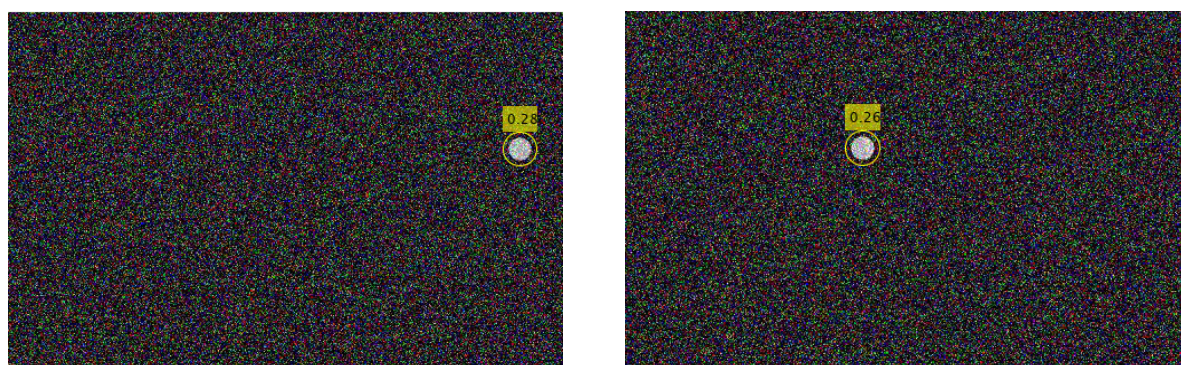
где e_l – ошибка длины, а e_w – ошибка ширины. θ_{span} – это угловой интервал, который цель занимает в датчике, а ζ – это постоянная, используемая в качестве расчетного параметра.

Имитационная модель

Рассмотрим имитационную модель двухпозиционной системы оптической локации, когда данные поступают с оптоэлектронных устройств на блок обработки, далее принимается решение, имеется ли в данном видеопотоке движущийся объект или нет, и затем осуществляется решение задач: оценки его траекторных координат, сопровождения, распознавания, а также задача разрешения двух крупных близкорасположенных объектов [11].

Прежде всего, в разработанной модели, положение локационного датчика следует задать таким образом, чтобы объект находился длительное время в области его видимости и при этом не заходил за препятствия [12]. В модели предполагается, что фокусировка источника видеоданных, а также еще других внутренних параметров системы видеонаблюдения уже осуществлена.

В модели, возможно, добавлять случайный белый гауссовский шум, при работе с кадрами в градациях серого – рис. 4. Имеется возможность задавать, количество кадров в секунду, количество объектов, их начальное положение, скорость и размер, а также траекторию их движения.



а) Кадр видеопотока по времени t левой камеры

б) Кадр видеопотока по времени t правой камеры

Рис. 4. Индикатор двухпозиционной оптической системы локации

На рис. 4а и 4б показано, как программный модуль детектирует движущийся объект на фоне шумов, а также определяет его текущее положение, пройденный путь и скорость в режиме реального времени.

Итак, оптические системы локации без специальных алгоритмов обработки видеоданных позволяют успешно решать задачи обнаружения, сопровождения, определения траекторных координат, однако для решения задач распознавания и разрешения объектов в группах, при условии удовлетворительных показателей сигнал/шум требуются более сложные алгоритмы обработки [13].

Моделирование стереопары

Рассматриваемая стереопара генерирует траектории(объектов) на основе алгоритма слежения, основанного на алгоритмах машинного зрения. Однако никакие реальные данные не использованы в этом тезисе, вместо траектории моделируются на основе обусловленного движущегося объекта рис.5.

Треки стереокамеры показывают предполагаемое положение (x, y) , направление (ϕ) и протяженность (w, l) обнаруженных объектов. Точность треков рассчитана на то, чтобы быть похожей на реальную систему слежения за стереокамерой. Это означает, что есть несколько факторов, которые влияют на точность пути. В этой работе учтены следующие факторы:

- расстояние между датчиком и объектом
- количество последовательных обнаружений объектом
- угловой диапазон, занимаемый объектом в датчике
- структурное несоответствие характеристик объектов

Применяя алгоритм обработки (рис.1), продемонстрируем работу стереопары.

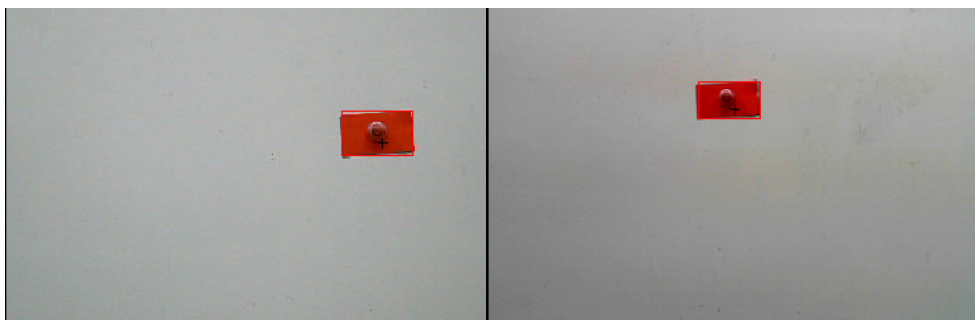


Рис. 5. Исходные изображения, полученные с левой и правой камер



Рис. 6. Изображения с выделенным контуром

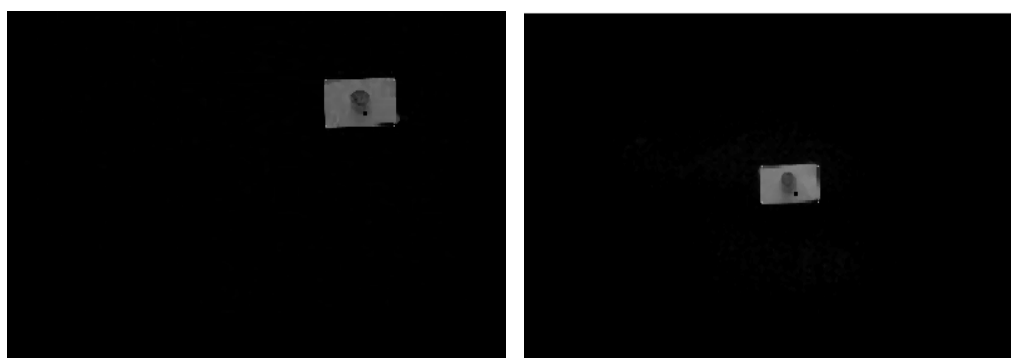


Рис. 7. Изображения с применением медианного фильтра



Рис. 8. Изображение с применением блока бинаризации

На основе формул изложенных в главе – "Ошибки отслеживания камеры", произведено в лабораторных условиях моделирование движущегося объекта на стенде. Вычислены его основные параметры, дальность, погрешность измерения камеры, а так же наложены кадры с двух параллельно стоящих камер [14, 15].

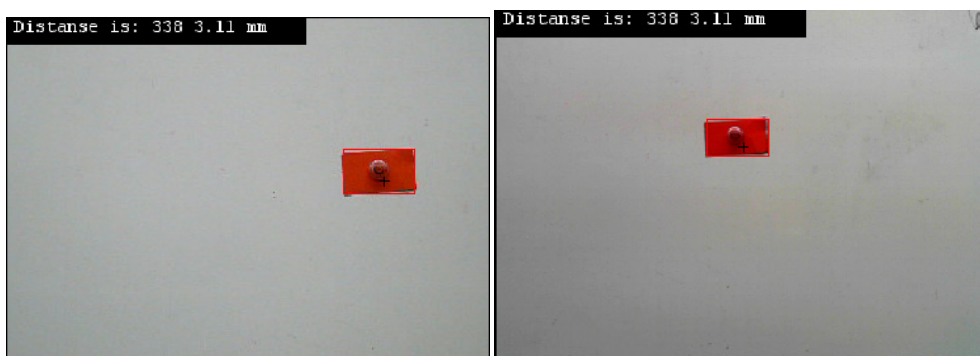


Рис. 9. Вычисление параметров

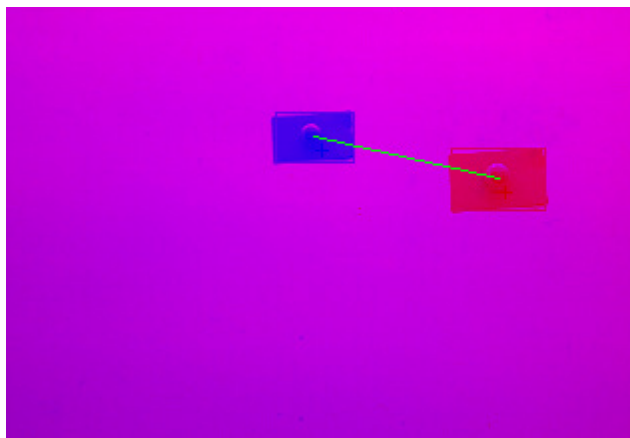


Рис. 10. Совмещенное изображение с левой и правой камер

Используя программный пакеты инструментов среды Matlab, произведен расчёт дистанции равном 338 мм до имитационного объекта, и погрешность равную, 3.11 мм. Данная погрешность удовлетворяет требованиям и согласуется с теоретическим рассчитанной.

Заключение

Результатом данной работы является реализованная система машинного зрения на основе двухпозиционной оптической локационной системы. С учетом того что каждый год требования к обработке данных повышается, была спроецирована имитационная модель система состоящей из двух веб-камер расположенных параллельно друг другу, на установленной базе, а на рабочем стенде поэтапно представлено обнаружение объекта, с учетом уменьшения шумов и помех. Данная имитационная модель увеличит скорость обработки данных. В работе используются методы распределенных систем оптических систем локации, методы машинного зрения, а также методы слияния данных.

Также на основе стереометрии были вычислены дистанции объектов. В работе представлена оценка ошибок положения камеры. Данный метод поможет оценить точность рассчитанной дальности до объекта и исключить в дальнейшей работе погрешности путем калибровки камеры.

Зная в точности все параметры объекта такие как, скорость, геометрические значения, дальность и координаты на карте местности, в чрезвычайной ситуации повышается оценку принятие решения.

Библиографический список

1. Локтев Д. А., Алфимцев А. Н. Измерение расстояния до движущегося объекта с помощью комплексной системы видеомониторинга // Инженерный журн.: наука и инновации. 2013. № 11 (23). С. 4.

2. Волков В. Ю. Адаптивные и инвариантные алгоритмы обнаружения объектов на изображениях и их моделирование в MATLAB: учеб. пособие. 2-е изд., доп. СПб.: Лань, 2014. 192 с.
3. Гребеников А. Г., Мялица А. К., Парфенюк В. В. и др. Общие виды и характеристики беспилотных летательных аппаратов: справ. пособие. URL: www.bookzz.org (дата обращения: 01.04.2020).
4. Волков В. Ю. Обнаружение объектов на изображениях с использованием селекции по площади.
5. Гай В. Е., Утробин В. А., Викулова Е. Н. и др. Ляхманов. Метод обнаружения объектов на изображении с позиций теории активного восприятия // Труды НГТУ им. П. Е. Алексеева. 2018. № 1 (120). Р. 9–15.
6. Object Detection Using Deep Learning. URL: https://ww2.mathworks.cn/examples/computer-vision/mw/deeplearning_shared-ex69834039-object-detection-using-deep-learning (дата обращения: 03.04.2020).
7. Kalal Zd., Mikolajczyk Kr., Matas J. Tracking-Learning-Detection // IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. 2010. Vol. 6, № 1.
8. Левинский А. Т., Селюто Н. М., Родионов И. Д. Отслеживание движения и оценка скорости объекта в видеопотоке. М.: МГТУ им. Н. Э. Баумана, 2017. 12 с.
9. Вахитов Л. Т., Гуревич Л. С., Павленко Д. В. Обзор алгоритмов стереозрения // Стохастическая оптимизация в информатике. Вып. 4 / под ред. О. Н. Граничина. СПб.: СПбГУ, 2008. 299с.
10. Волкович А. Н. Расчет расстояния до объекта и карт диспаратности на основе разностных исходных данных // Решетневские чтения. 2017. № 21–2. Р. 322–323.
11. Бейненсон Л. Методы сопровождения объектов. URL: <https://delta.course.org/docs/delta7/Day7/Delta7D7L2.pdf> (дата обращения: 01.04.2020).
12. Шенета А. П., Ненашев В. А., Григорьев Е. К. и др. Обнаружение и оценка скорости подвижных объектов в оптико-локационных системах // Научная сессия ГУАП: сб. докл.: в 3 ч. Ч. II: Техн. науки. СПб.: ГУАП, 2019. – С. 455–459.
13. Морозов А. А., Сушкова О. С. Анализ видеоизображений в реальном времени средствами языка Акторный Пролог // Компьютерная оптика. 2016. Т. 40, № 6. Р. 947–957. Doi: 10.18287/2412-6179-2016-40-6-947-957.
14. Фам Конг Тханг, Данг Нгок Хоанг Тхань, Нгуен Ван Хьеу. Автоматическая сшивка изображений на основе характерных точек // Техн. науки – от теории к практике. 2014. № 36. С. 23–29.
15. Толкачев Д. С. Формирование панорамного изображения с учетом параллакса при известной модели окружающего мира // Инженерный вестн. Дона. 2013. Т. 26, № 3. С. 116.
16. Анисимов А. Г., Араканцев К. Г., Горбачев А. А. Исследование погрешности контроля дистанции в симметричном внутрибазовом канале двухкоординатной оптико-электронной системы контроля смещений // Научно-техн. вестн. информ. технологий, механики и оптики. 2006. № 34. С. 213–218.

УДК 004.92

Д. С. Поляков

студент кафедры вычислительных систем и сетей

Н. Н. Решетникова – кандидат технических наук, доцент – научный руководитель

ИНТЕРАКТИВНАЯ 3D-РЕКОНСТРУКЦИЯ СРЕДНЕВЕКОВОЙ КРЕПОСТИ С ИСПОЛЬЗОВАНИЕМ СПЕЦЭФФЕКТОВ

Стремительное развитие технологий в последнее десятилетие привело к такому же быстрому росту в области компьютерной техники и программного обеспечения. Еще совсем недавно незначительный по сегодняшним меркам графический эффект в 3D игре вызывал бурю восторга и обсуждений. Сегодня спецэффекты стали обыденным явлением. В современных 3D играх встречается множество элементов, которые по своей природе не имеют четкой формы и изменяются в реальном времени, например, огонь и дым, дождь и снег, для их моделирования применяются инструменты, известные как системы частиц (Particle System).

В статье рассматривается применение системы частиц при создании интерактивного приложения «Средневековье» на базе игрового движка Unity3D. Unity3D [1] — один из самых популярных игровых движков, в котором компонент Particle System может применяться для симуляции многих типов эффектов, так как имеет большое количество настроек и вариантов использования. Интерактивность в приложении подразумевает, что пользователь может не только наблюдать за происходящим, но и перемещаться в виртуальном пространстве с возможностью взаимодействовать с 3D объектами. Для повышения реалистичности сцены в приложении «Средневековье» присутствуют спецэффекты огня факела и дыма при выстреле из пушки, реализованные с использованием модуля Particle System, а также физика объектов.

Процесс создания интерактивного приложения состоит из следующих этапов:

- Создание моделей сложных объектов в 3D редакторе (Autodesk 3Ds Max);
- Перевод моделей в интерактивную среду игрового движка Unity3D;
- Моделирование физики объектов;
- Текстурирование моделей;
- Создание спецэффектов на основе Particle System;
- Постановка освещения;
- Разработка скриптов для управления сценой пользователем.

Рассмотрим подробнее этапы создания интерактивного приложения.

Трехмерные модели крепости (рис. 1) и пушки (рис. 2 а,б) создавались в пакете Autodesk 3Ds Max [2].

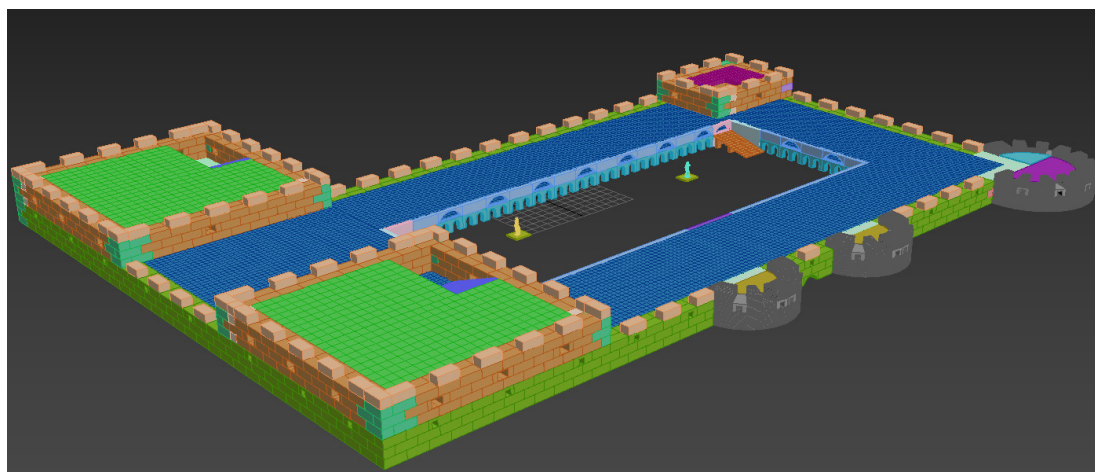


Рис. 1. Модель крепости в Autodesk 3Ds Max

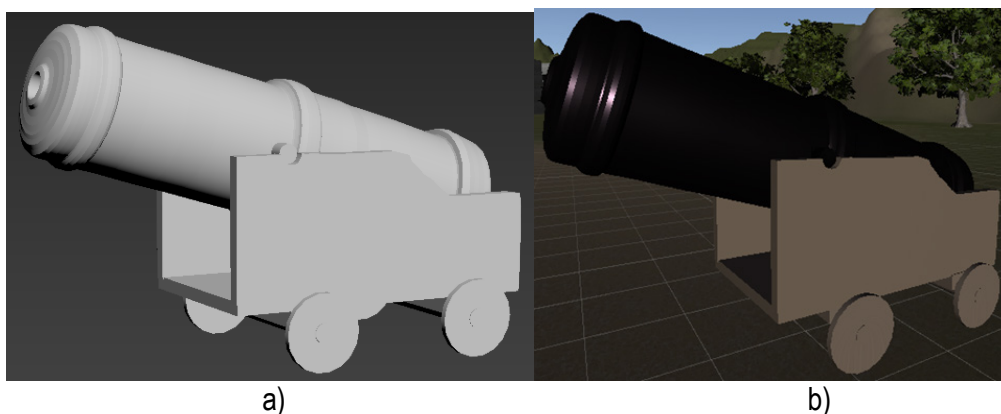


Рис. 2. а). Модель пушки: в Autodesk 3Ds Max; б). Текстурированная модель пушки в Unity 3D

При создании этих моделей использовались различные способы и инструменты, например, для создания арок и окон, использовались булевы операции, для вычитания проемов в стенах крепости. Для создания стен крепости, состоящих из примитивов Box (Прямоугольник), использовался инструмент Array (Массив). При моделировании пушки использовались контуры с изображения реальной пушки и модификатор Extrude (Выдавливание), с помощью которого выдавливались контуры и получались детали конечной модели.

Перед использованием моделей в этом игровом движке необходимо их экспортировать из Autodesk 3Ds в формате *.fbx, а затем поместить модели в папку созданного проекта в Unity. При переводе моделей в игровой движок Unity они станут Prefab (шаблон игрового объекта). Для редактирования Prefab'a его нужно распаковать (Unpack Prefab Completely) в Hierarchy Window (окно иерархии).

Текстурирование моделей необходимо для увеличения реалистичности сцены. При создании интерактивного приложения потребовались текстуры травы, песка, бетона, дерева, чугуна.

Моделирование физики объектов. Для реалистичности интерактивной сцены введена физика для каждого объекта. Физика к объекту добавляется с помощью применения к объектам компонентов Rigidbody и одного из видов Collider (например, Box Collider, Mesh Collider...). Для перекрытий крепости также применяется компонент Fixed Joint, позволяющий перекрытиям падать при применении к ним определенной силы.

Rigidbody – компонент, позволяющий GameObjects (Объектам Сцены) действовать под контролем физики. Любой объект с компонентом Rigidbody может получать силу и крутящий момент, чтобы двигаться реалистично.

Collider – компонент, определяющий форму объекта в целях физических столкновений.

Fixed Joint – компонент, ограничивающий движение определенного объекта, с возможностью связывания с другим объектом.

Для моделирования физики арки внутренней стены крепости потребовалось применить пять(5) компонентов Box Collider, создающих вместе форму для физических взаимодействий. Как и к любому объекту, который должен взаимодействовать по законам физики, к ним добавляем компонент Rigidbody (рис. 3).

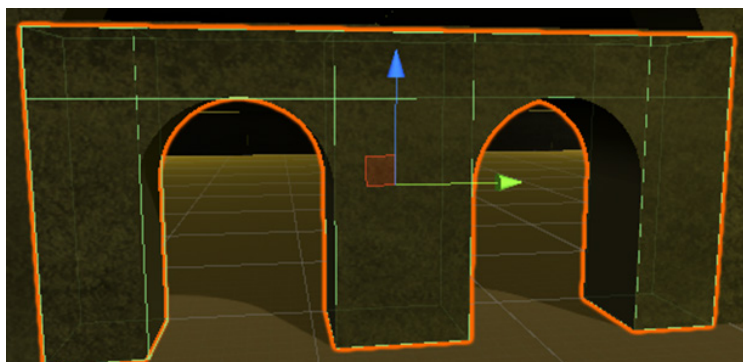


Рис. 3. Пять компонентов Box Collider для проемов в стенах крепости

Для придания физики пушке также использовались компоненты Box Collider и Rigidbody, и кроме того были написаны скрипты, реализующие перемещение пушки по сцене, стрельбу из пушки, настройку мощности выстрела, выбор ядра, которым стреляет пушка, отдачу при выстреле.

Специальные эффекты на основе Particle System. Для имитации огня факела и дыма при выстреле из пушки использовался компонент Particle System. Перед тем, как имитировать огонь и дым в игровом движке необходимо провести анализ физической природы этих явлений.

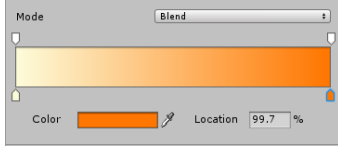
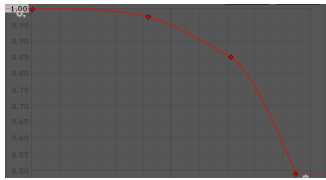
Проведем анализ физической природы огня:

- Пламя вытянутое;
- Сосредоточено вокруг горячего места факела;
- Высокая скорость стремления пламени вверх;
- Цвет пламени меняется от желтого цвета у основания пламени к оранжевому цвету у кончиков пламени.

На основе проведенного анализа огня создан спецэффект огня с помощью Particle System. Основные параметры Particle System, использованные при создании спецэффекта огня, и их значения представлены в табл. 1.

Таблица 1

Параметры Particle System для эффекта огня

Название	Описание	Значение
Duration	время, в течении которого будет работать система	5.00
Looping	циклическая работа системы	True
Start Delay	задержка до излучения частиц после запуска системы частиц	0
Start Lifetime	продолжительность жизни частиц после запуска системы	1.5
Start Speed	начальная скорость частицы при ее создании	35
Start Size	начальный размер частицы	40
Gravity Modifier	коэффициент гравитации для частиц	-0.3
Max Particles	максимальное количество частиц, одновременно находящихся в системе	10000
Emission	модуль, обрабатывающий число, время излучения частиц, задающий тип работы частиц: непрерывный поток или внезапный выброс	Rate over Time – 20 Rate over Duration – 0
Shape	модуль, который управляет поведением частиц в заданной форме (например, в форме конуса, сферы, прямоугольника)	Shape – Cone Angle – 25 Mode – Random
Color over Lifetime	модуль, позволяющий задавать цвет частицы в зависимости от ее времени жизни	
Size over Lifetime	модуль, позволяющий задавать размер частицы в зависимости от времени ее жизни	
Render	модуль, позволяющий определять вид частиц	Billboard Material – материал с текстурой огня из Standart Asset

В результате настройки и применения параметров системы частиц, приведенных в табл. 1, моделирование эффекта огня факела выглядит следующим образом (рис. 4).

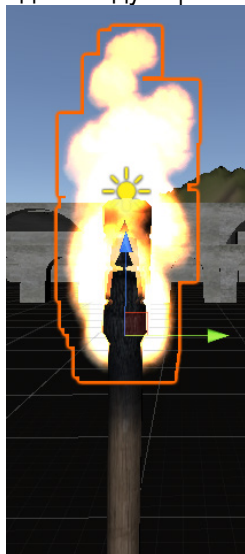
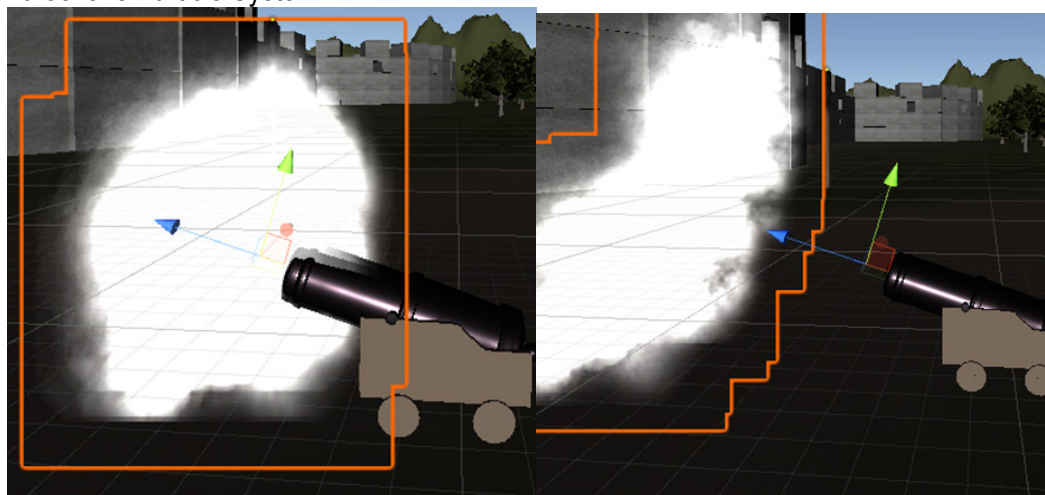


Рис. 4. Эффект огня в Particle System

Полученный спецэффект на рис. 4 реалистично повторяет пламя огня: пламя вытянутое и стремится вверх; цвет пламени меняется от желтого оттенка к оранжевому
Спецэффект дыма. Проведем анализ физической природы дыма [3]:

- В момент выстрела пушки из дула с высокой скоростью выходит густой дым;
- Далее дым по инерции устремляется вперед и рассеивается.

На основе проведенного анализа дыма при выстреле из пушки создан спецэффект дыма (рис. 5 а), b)) на основе Particle System.



a)

b)

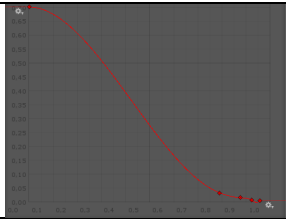
Рис. 5. а). Момент выстрела пушки; б). Момент после выстрела из пушки

Основные параметры Particle System, использованные при создании спецэффекта дыма, и их значения представлены в табл. 2.

Полученный спецэффект реалистично повторяет дым при выстреле из пушки: серый и густой дым с высокой скоростью выходит из дула; по инерции дым рассеивается и устремляется вперед.

Постановка освещения. Освещение в интерактивном приложении представлено двумя видами: Directional Light (Направленный свет) и Point Light (Точечный свет). Направленный источник света, расположенный над сценой, используется для имитации солнечного света, а точечный источник – для имитации света от пламени факела.

Параметры Particle System для эффекта дыма

Название	Описание	Значение
Duration	время, в течении которого будет работать система	0.20
Looping	циклическая работа системы	False
Start Delay	задержка до излучения частиц после запуска системы частиц	0
Start Lifetime	продолжительность жизни частиц после запуска системы	30
Start Speed	начальная скорость частицы при ее создании	4
Start Size	начальный размер частицы	15
Gravity Modifier	коэффициент гравитации для частиц	0
Max Particles	максимальное количество частиц, одновременно находящихся в системе	10000
Emission	модуль, обрабатывающий число, время излучения частиц, задающий тип работы частиц: непрерывный поток или внезапный выброс	Rate over Time – 50 Rate over Duration – 0
Shape	модуль, который управляет поведением частиц в заданной форме (например, в форме конуса, сферы, прямоугольника)	Shape – Cone Angle – 25 Mode – Random
Size over Lifetime	модуль, позволяющий задавать размер частицы в зависимости от времени ее жизни	
Render	модуль, позволяющий определять вид частиц	Render Mode – Billboard Material – используется материал с текстурой дыма из Standart Asset

Разработка скриптов. Для взаимодействия пользователя с приложением используются скрипты, написанные на языке программирования C# [4]. В разработанном приложении используются следующие скрипты:

- Ogon.cs – запуск имитации дыма из дула при стрельбе (используя Particle System);
- Gun.cs – управление мощностью выстрела и выбор ядра, которым стреляет пушка, назначение клавиши выстрела из пушки;
- Move.cs – перемещение пушки по сцене;
- UI.cs – для нужд меню и загрузки сцен.

В заключении следует отметить, что скрипты не влияют на стабильность работы интерактивного приложения, стабильно поддерживая FPS на постоянном уровне. Интерактивное приложение было создано для погружения в эпоху средневековья. Сцена выглядит реалистично благодаря подключению инструментов моделирования физики 3D объектов, а также имитации эффектов огня и дыма, которые по своей природе не имеют чёткой формы, с использованием возможностей детальной настройки параметров системы частиц (Particle System) игрового движка Unity.

Библиографический список

1. Free Student Software Downloads | Autodesk Education Community // Autodesk. URL: <https://www.autodesk.com/education/free-software/featured> (дата обращения: 08.12.2019).
2. Unity Manual – Unity User Manual (2019.2) // Unity Documentation. [электронный ресурс] URL: <https://docs.unity3d.com/Manual/index.html> (дата обращения: 08.12.2019).
3. Выстрел пушки на острове Русский // Youtube. URL: <https://www.youtube.com/watch?v=DMaKsHYACBo> (дата обращения: 08.12.2019).
4. *Либерти Дж.* Программирование на C# / пер. с англ. СПб.: Символ, 2003. 688 с.

УДК 004.451.54

Р. С. Халиков

студент кафедры вычислительных систем и сетей

С. Л. Прокопьева

студент кафедры вычислительных систем и сетей

К. А. Спесивцева – ассистент – научный руководитель

ПРОГРАММА, ФОРМИРУЮЩАЯ УЧЕБНОЕ РАСПИСАНИЕ ДЛЯ СТУДЕНЧЕСКИХ ГРУПП

Одна из задач любого учебного заведения – наиболее удобное и подходящее прохождение учебного процесса в стенах ВУЗа. Поэтому все более актуальным становится вопрос об автоматизации этого процесса.

Если использовать при организации процесса обучения ЭВМ, в частности различного рода программные продукты, то можно сделать этот процесс гораздо более точным и быстрым, лишенным множества накладок, которые возникают при его организации вручную.

Задачу составления расписания не стоит рассматривать только как некую программу, реализующую функцию механического распределения занятий в начале семестра, на которой ее (программы) использование и заканчивается. Экономический эффект от более эффективного использования трудовых ресурсов может быть достигнут только в результате кропотливой работы по управлению этими трудовыми ресурсами. Расписание здесь является лишь инструментом такого управления, и для наиболее полного его использования необходимо, чтобы программа сочетала в себе не только средства для составления оптимального расписания, но и средства для поддержания его оптимальности в случае изменения некоторых входных данных, которые на момент составления расписания считались постоянными.

Цель данной статьи состоит в том, чтобы отразить суть программы, позволяющей сформировать расписание занятий для групп высших учебных заведений.

Расписание занятий в высшем учебном заведении служит для сведения в единую взаимосвязанную систему учащихся, преподавателей и мест проведения занятий (аудиторий). Оптимизация расписания занятий является одним из основных факторов, способных существенно оптимизировать учебный процесс. В частности, организация учебного процесса, представляя собой один из важнейших этапов на пути развития и эффективного функционирования вуза, является совокупностью взаимосвязанных задач, решаемых различными подразделениями учебного заведения. Одной из таких задач является составление расписания. От того, насколько хорошо составлено расписание зависит эффективность работы преподавателей, усвоение учебного материала студентами, рациональное использование интеллектуальной и материальной баз вуза. Традиционный способ представляет собой следующие этапы, выполняющиеся вручную:

Разработка и утверждение учебных планов.

Выборка из планов перечня дисциплин и закрепление их за кафедрами.

Составление учебных поручений кафедрам других факультетов на закрепление учебной нагрузки за преподавателями.

Распределение нагрузки на кафедрах.

Составление проходки на семестр, которая содержит информацию о дисциплинах, видах занятий, часами по каждому из видов и преподавателе.

Учёт занятого и свободного времени преподавателей, работающих по совместительству.

Составление расписания.

Теперь обратимся к самой программе, а именно к функциональным назначениям, описанию логической структуры и вызову и загрузки.

Программа написана на языке Python и состоит из окна выбора необходимой нам группы и их расписанием. Данная программа предназначена для более удобного и быстрого процесса составления расписания для групп высших учебных заведений.

Программа написана на языке Python с подключением базы данных SQLite. Данная программа открывается с помощью терминала (командной строки).

После того, как она откроется – появится окно выбора между группами, расписание которых вы составили. (В нашем случае, мы составляли расписание для всего нашего потока).

С помощью базы данных SQLite есть возможность редактирования, удаления и добавления того или иного фильтра, позволяющего определить предмет, преподавателя, красную/синюю неделю, группу.

Вызов и загрузка программы осуществляется в следующей последовательности: открыть терминал (командную строку), ввести "127.0.0.1:5000"

Далее открывается окно с выбором группы (Рисунок 1).

При выборе группы на экране выводится её расписание (Рисунок 2).

Открытие базы данных:

Требуется наличие дополнительной программы "DB Browser for SQLite".

В ней вносится код программы для последующей работы с данными.



Рис. 6. Окно выбора группы

понедельник
вторник
5 пара ЛР - Основы программирования - 24-05 - 4842 Преподаватель: Куртяник Д.В.(старший преподаватель)
6 пара ▲ ПР - Социология и политология - 24-10 - 4842 Преподаватель: Исаев Б.А.(профессор д-р социол.наук)
среда
3 пара ▼ ЛР - Электротехника - 14-06 - 4842 Преподаватель: Шишлаков Д.В.(доцент,канд. техн.наук)
4 пара ▼ ЛР - Электротехника - 14-06 - 4842 Преподаватель: Шишлаков Д.В.(доцент,канд. техн.наук)
четверг
2 пара ▲ ЛР - Численные методы и вариационное исчисление - 22-09 - 4842 Преподаватель: Шинтяков Д.В.(доцент,канд. техн.наук)
3 пара пр - Нормативная документация - 21-01 - 4842 Преподаватель: Попов В.П.(доцент,канд. техн.наук)
4 пара ЛР - Численные методы и вариационное исчисление - 22-09 - 4842 Преподаватель: Шинтяков Д.В.(доцент,канд. техн.наук)
пятница
4 пара ▼ ЛР - Физика - 32-06 - 4842

Рис. 7. Пример расписания для группы 4842 с указанием названия предмета, преподавателя, должности преподавателя, времени пары, красной/синей недели

Условия применения данной программы просты: необходим Python версии 3.0 и реляционная база данных SQLite, технические средства должны обеспечивать вывод результатов вычислений на экране.

Заключение

Таким образом, в данной статье мы рассмотрели основные аспекты программы, формирующей учебное расписание.

Наши данные свидетельствуют о том, что нам еще многое предстоит сделать, чтобы довести сделанную работу до более высокого уровня, однако уже на этом этапе развития данная программа может оказаться полезной.

Библиографический список

1. Каким может быть расписание вуза. URL: <https://habr.com/ru/post/194212/> (дата обращения: 16.11.2019).
2. Работа с данными из базы данных SQL. URL: <https://pythonworld.ru/obrabotka-dannyh/pandas-cookbook-8.html> (дата обращения: 18.11.2019).
3. Компилирование SQLite. URL: <https://www.sqlite.org/howtocompile.html> (дата обращения: 20.12.2019).
4. Руководства работы по SQLite. URL: <https://proglib.io/p/sqlite-tutorial> (дата обращения: 20.12.2019).

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ЗАЩИТА ИНФОРМАЦИИ

УДК 004.622

А. Д. Ахраменко

студент кафедры инфокоммуникационных систем

А. М. Тюрликов – доктор технических наук, профессор – научный руководитель

СИСТЕМА СБОРА ДАННЫХ ОБ ОЦЕНКЕ СВОЕГО СОСТОЯНИЯ ЗДОРОВЬЯ ПАЦИЕНТАМИ

Введение

При обычном образе жизни человек, обнаруживший у себя какие-либо нетипичные симптомы, склонен обратиться к врачу, который осмотрит пациента и поставит диагноз. В таком случае человек должен прийти до больницы, общаться с медперсоналом и потратить часть своего времени, но что делать если у человека нет возможности посетить врача. Такие ситуации могут случаться как у отдельно взятого человека постоянно, например, если пациент не имеет возможности передвигаться без посторонней помощи либо он слишком занят и не успевает посетить врача, так и у многих людей сразу, как например, в ситуации, происходящей на сегодняшний день, в период пандемии и самоизоляции, когда каждый выход на улицу нежелателен, не говоря о посещении больницы. Тогда встает вопрос, каким образом человек может обследоваться и получить предварительный диагноз. Совокупность этих фактов приводит к мысли о разработке системы, которая способна поставить диагноз на основе самочувствия человека дистанционно. Помимо трудности создания такой системы встает вопрос о создании исходных данных, поступающих в эту систему в качестве обучающей выборки. Данные должны из себя представлять симптомы, их интенсивности и диагнозы, поставленные по определенному набору симптомов и интенсивностей. Именно поэтому появляется необходимость в создании системы, которая соберет необходимую информацию о пациенте, обработает, структурирует и подготовит для дальнейшего машинного обучения. Такой набор данных можно будет набрать как из медицинских баз, так и с помощью системы, опрашивая пациента.

В настоящее время в области автоматизации здравоохранения наблюдается тенденция, когда учреждения здравоохранения самостоятельно покупают информационную систему либо самостоятельно разрабатывают ее. Один из способов объединения медицинских данных описан в работе [1], однако он больше направлен на объединение общей информации в лечебном учреждении, а не на сбор и анализ данных об одном пациенте. Более схожим решением является интеграционная система Validic [2], которая позволяет получать доступ к данным, ежедневно генерируемым пациентами и медицинскими организациями, и повышать их качество. Такие данные формируются в единый поток информации и могут передаваться в любую медицинскую информационную систему. При этом данные до передачи в медицинскую или иную систему проверяются на корректность.

В этой компании была разработана система PGHD – patient-generated health data (данные о состоянии здоровья пациента). Идея продукта состоит в возможности контроля за здоровьем пациента путем постоянного сбора биометрических данных. В работе А.В. Гущина [3] описывается интернет-системы для сбора и обработки данных о связанном со здоровьем качестве жизни (СЗКЖ) населения.

Основной идеей такой системы является формирование концепции длительного, систематизированного измерения СЗКЖ широких слоев населения по средствам сети Интернет. А.В. Гущин описывает алгоритм, с помощью которого обычные пользователи (пациенты) могут оценить эффективность лечебно-диагностических мероприятий. Таким образом задача сбора данных об оценке своего состояния здоровья пациентами остается актуальной. Данная система является первым шагом для создания системы постановки диагноза, с использованием машинного обучения.

Исходные данные, их анализ и введение в систему

Проблема заключается в том, что нельзя начать создавать данные на основе произвольных текстов, записанных со слов пациентом. Каждый человек по-разному описывает свое состояние здоровья, используя разные термины, поэтому обработка таких текстов является сложно выполнимой задачей. В связи с этим удобнее и логичнее взять уже существующие тексты, приведенные к какому-то определенному формату. В качестве исходных данных была взята выборка жалоб и анамнезов 100 пациентов, которые были записаны со слов пациентов врачами. Каждый текст представлял из себя перечисление жалоб пациента (например, «На слабость, нестабильные цифры АД, головные боли, головокружения, дискомфорт в грудной клетке, редко – перебои в работе сердца, одышка»). Соответственно, каждый текст было необходимо обработать, выделить симптомы и интенсивности, и далее объединить полученные данные в список, где у каждого симптома есть градация интенсивностей с числовым значением. Таким образом были обработаны все 100 пациентов и получившийся список содержал 42 симптома, где у каждого было от 2 до 5 видов интенсивностей. Данный список был сформирован самостоятельно и не отвечал требованиям точности и достоверности, поэтому на следующем этапе врач дополнительно скорректировал список и пересформировал соотношения симптом-интенсивность, а также расположил интенсивности от самой легкой до самой тяжелой. В итоге, конечный список содержал 32 симптома и их интенсивности. В качестве примера приведен симптом головокружение с интенсивностями:

Головокружение	1	Нет головокружения
	2	Ощущение вращения окружающих предметов, и/или собственного тела
	3	Чувство дурноты, предобморочное состояние, «потемнение» в глазах

Хранение пар симптом – интенсивность

Следующим шагом является проектирование и реализация базы данных, в которой будут храниться пары симптом-интенсивность, данные о пользователе и результаты прохождения диагностики. После таких шагов как определения сущностей и атрибутов, выполнения метода синтеза, определения зависимостей и построения диаграммы зависимости, была получена результирующая декомпозиция отношений, то есть 6 таблиц.

Для хранения пар симптом – интенсивность было использовано 3 таблицы. В первой таблице хранятся порядковый номер симптома и название симптома, во второй таблице хранятся абсолютно все интенсивности, а именно порядковый номер интенсивности и название интенсивности, третья таблица отвечает за создание пар симптом – интенсивность и хранит в себе пары порядковых номеров симптомов и интенсивностей и числовое значение интенсивности.

Создание и реализация алгоритма диагностики пользователя

Получаемые данные после опроса пациента должны иметь как можно более высокую точность и достоверность. Именно поэтому был предложен алгоритм, состоящий из двух этапов. Первый этап представляет из себя тест, проходя который человек выбирает симптомы и интенсивности, соответствующие его самочувствию. Во втором этапе человек заполняет свободную форму, описывая состояние здоровья своими словами. По окончании диагностики будут сформированы два текста, один упорядоченных по парам симптом – интенсивность, второй – свободный текст. Пример строки, получаемой после первого этапа: «April 21, 2020, 11:50 pm; Шум в ушах: Шум (2); Головные боли : Головные боли (2); Головокружение: Ощущение вращения окружающих предметов и/или собственного тела. (2); Стабильность артериального давления: нестабильные цифры (2); Характер боли в грудной клетке: Колющие (4); Аритмии: «Перебои» в работе сердца постоянно (3);». Получение двух видов одной и той же информации облегчит дальнейшую работу по обучению нейронной сети.

Интерфейс системы был реализован как веб-сайт, что значительно облегчает процесс сбора данных, так как потенциальному пользователю нет необходимости устанавливать какое-либо дополнительное программное обеспечение. В системе присутствует процесс регистрации, что позволяет

уже на данном этапе использовать систему для прохождения диагностики и отсылать получившийся результат своему лечащему врачу.

Заключение

Предложен метод сбора данных о субъективном состоянии здоровья пациентов, а также представлен процесс обработки исходных данных и их представление в систематизированном виде. На данный момент система выполняет свою основную функцию – сбор данных. Далее эти данные планируется использовать в качестве входных данных для обучения модели.

Библиографический список

1. *Копаница Г. Д., Силич В. А.* Разработка системы сбора и анализа данных медицинской статистики на основе стандарта передачи медицинских данных ISO 13606 // Доклады ТУСУР. 2011. № 2 (24). Ч. 2. С. 290–294.
2. Обзор системы PGHD от Validic. URL: <https://validic.com/> (дата обращения: 25.04.2020).
3. *Гущин А. В.* Здоровье и качество жизни населения: концепция интернет-системы сбора и обработки данных // Социолог. исслед. 2016. № 7. С. 119–123.

УДК 004.51**Е. И. Гордеева**

студент кафедры инфокоммуникационных систем

М. Р. Егорова

студент кафедры инфокоммуникационных систем

Е. С. Суворова

студент кафедры инфокоммуникационных систем

М. М. Афанасьев – ассистент – научный руководитель

РАЗРАБОТКА ТАБЛО ДЛЯ ЭЛЕКТРОННОЙ ОЧЕРЕДИ SMART SCREEN**Введение**

Сегодня электронные табло используются в каждой сфере нашей жизни. Мы видим их утром, когда едем в транспорте, они показывают нам температуру на улице и сообщают о скорой остановке, так же встречаем их в обед, когда ждем очередь своего заказа, порядок отражается на дисплее, и даже вечером, когда идем домой, тоже на глаза попадают электронные табло, на которых красуется реклама магазинов. Но если купить дисплей с бегущей строкой не представляет проблем, сейчас на рынке их представлено достаточно много и цена из-за конкуренции становится всё ниже, то с покупкой табло для электронной очереди возникают проблемы.

На данный момент на рынке нет электронных табло, которые можно было бы быстро адаптировать под любую написанную систему очереди, поэтому при покупке необходимо будет приобрести так же и готовую электронную очередь. Минусом такого решения помимо самой стоимости, ведь эта программа всегда идет за дополнительную плату, является так же то, что её код закрыт и вы не сможете в нем ничего изменить. Если необходимо будет перейти на другую систему электронной очереди, единственным решением будет приобрести другие табло, отправив предыдущие на утилизацию.

Большинство дисплеев имеет сложные в управлении меню для клиентов или не имеет их вовсе. Именно из анализа рыночной ситуации и появилась сама идея нашего дисплея.

Связь экранов электронной очереди в большинстве случаев осуществляется с помощью отдельной программы с закрытым кодом, что ограничивает пользователя в действиях.

ПОСТАНОВКА ЗАДАЧИ

Разработать электронное табло без привязки к какой-либо системе электронной очереди, с возможностью управления экранами по Wi-fi, с минимальной стоимостью, с удобным пользовательским интерфейсом, с открытым кодом, с возможностью быстрой интеграции с любой сторонней системой электронной очереди.

ВЫБОР КОМПЛЕКТУЮЩИХ

Для создания нашего устройства нам понадобились следующие комплектующие: интерьерный светодиодный модуль QIANGLI, микроконтроллер ESP-32, провода, кнопка включения/выключения, пластиковый корпус.

1. Интерьерный светодиодный модуль QIANGLI – Прежде всего, выбор этого светодиодного модуля обусловлен его техническими характеристиками, такими как разрешение в 40x80 RGB пикселей, хорошей яркостью и размером самого дисплея в 320x160 мм с шагом пикселей 4 мм, что позволяет выводить на нем как текстовую, так и графическую информацию разных цветов на расстоянии до 3-4 метров, что, по нашему мнению, является оптимальным расстоянием для систем электронной очереди внутри помещений. Также у данного светодиодного модуля есть возможность соединения нескольких таких же в один экран побольше, что также является плюсом при дальнейшем развитии проекта. Производителем заявлена продолжительность работы до 10 лет.



Рис. 1. Интерьерный светодиодный модуль QIANGLI

2. Микроконтроллер ESP-32 – Выбор этого микроконтроллера был обусловлен его невысокой стоимостью и всеми необходимыми для нашего проекта техническими характеристиками, сокращение стоимости устройства играет для нас одну из ключевых ролей. Эта плата имеет низкое энергопотребление, с интегрированными Wi-fi и Bluetooth контроллерами и антеннами, что является для нас самым главным требованием, так как именно это позволит организовать веб-интерфейс для управления по беспроводному каналу и поможет осуществить связь экранов с электронной очередью. Так же плюсом этого микроконтроллера является его маленький размер 5х3 см и наличие внутренней памяти, что позволит нам облегчить работу пользователя, так как все настройки можно будет сохранить в плате и отключение питания, даже экстренное, не удалит из устройства сохраненные данные. Так же полезной характеристикой является поддержка большого диапазона рабочей температуры от -40°C до $+125^{\circ}\text{C}$, а значит наши экраны можно разместить даже на улице и в самые сильные морозы с ними ничего не случится. (при внесении дополнительных изменений в конструкцию)

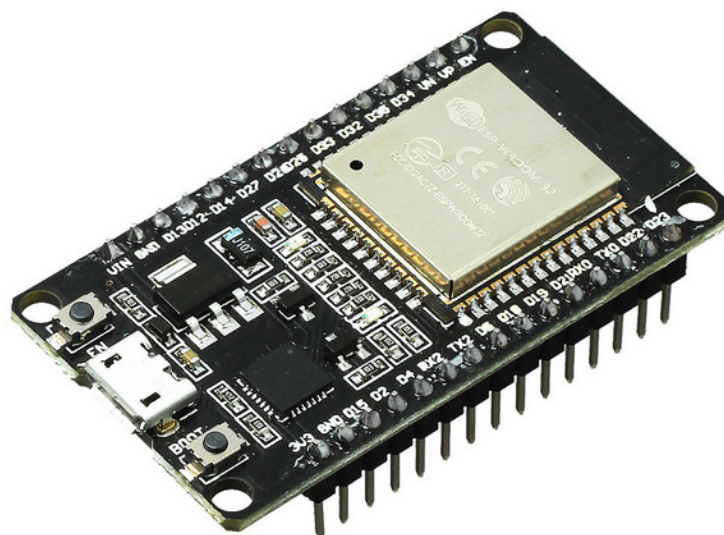


Рис. 2. Микроконтроллер ESP-32 DevKit V1

3. Блок питания – Исходя из характеристик самого экрана и платы напряжение питания в 5 вольт будет достаточным, а их общий ток достигает до 4.6 ампер при максимальном энергопотреблении (то есть когда весь экран горит белым цветом, потому что в этом случае у каждого пикселя горят все три компонента R, G и B). Блок питания был подобран согласно всем этим характеристикам, также мы старались выбрать блок питания небольшого размера и с наименьшей ценой.



Рис. 3. Блок питания RS-25-5

В качестве протокола обмена сообщениями был выбран протокол MQTT. Этот протокол был изначально разработан для взаимодействия между устройствами Интернета Вещей и развивается уже 20 лет. Поддерживается большим количеством языков программирования (C/C++, Java, Python, PHP и многими другими). Взаимодействие между устройствами происходит по принципу издатель-подписчик. На рис. 4 в центре сто можно видеть центральный элемент этого протокола – MQTT брокер, задача которого заключается в том, чтобы принимать сообщения от подключенных устройств и пересылать их другим. Каждое сообщение состоит из топика (заголовка) и самого сообщения (текстовая строка). Каждое устройство после включения подключается к брокеру и сообщает ему множество интересующих его топиков (это называется подписывается на топики), и как только брокер получит очередное новое сообщение с топиком из этого множества, то это сообщение будет переслано этому устройству. Простота этого протокола заключается в том, что в любой момент можно добавить новое устройство и начать сразу же им управлять без каких-либо сложных настроек системы.

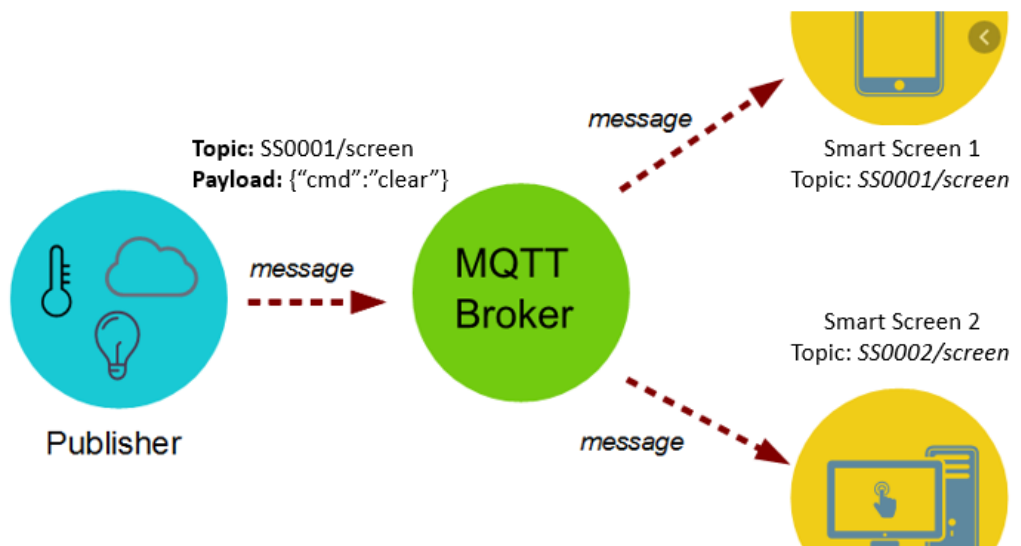


Рис. 4. Схема работы

Выбором этого протокола мы решаем задачу простоты интеграции нашего устройства в любую существующую систему электронной очереди (и не только), за счет того, что практически для каждого языка программирования существует библиотека клиента MQTT.

Программная часть состоит из потока обработки команд управления MQTT, потока обработки запросов веб сервера и потока обновления дисплея. Выбранный нами микроконтроллер ESP32 имеет два вычислительных ядра, и мы как программисты, можем решать какой поток на каком ядре запускать. Из-за особенности работы дисплея, для создания стабильной картинки (не мерцающей в глазах)

необходимо соблюдать одинаковые интервалы между обновлениями строчек экрана, и, если в процессе работы многозадачной операционной системы микроконтроллера (FreeRTOS) произойдет переключение контекста на другую задачу, это практически сразу приведет к появлению видимых глазу мерцаний. Поэтому, после рассмотрения различных способов компоновки потоков между ядрами, было принято решение выделить самое свободное второе вычислительное ядро под задачу обновления экрана, установив один из самых больших приоритетов. Потоки обработки команд MQTT и веб сервера установили на первое ядро, где они разделяют вычислительные ресурсы ядра с служебными задачами обеспечивающие работу стека Wi-Fi, но из-за низкой нагрузки наших потоков они не сказываются на качестве работе Wi-Fi.

На рис. 5 вы можете видеть пример сообщений, передаваемых по протоколу MQTT. Так как за одно сообщение может потребоваться передача многих различных параметров описываемых в свою очередь различным числом полей, нами было принято решение использовать формат представления данных JSON, который предназначен для представления структурированных данных любой степени вложенности. В свою очередь, JSON еще более популярный формат представления практически в любом языке программирования, поэтому наше это решение не сказывается на легкости интеграции нашего устройства в другие системы.

Пример команды управления

1. Очистить экран

```
{"cmd":"clear"}
```

2. Установка картинки

При установке картинки можно одной командой установить необходимые параметры (указываются только нужные):

```
{  
  "cmd": "bitmap1",  
  "data": "...", /*0123456789ABCDEF*/  
  "fps": "100",  
  "brightness": "3",  
  "gammaCorrection": "1",  
  "colorDepth": "1"  
}
```

Рис. 5. Пример сообщений, передаваемых по протоколу MQTT

РАЗРАБОТКА ПЕЧАТНОЙ ПЛАТЫ

В первоначальном варианте сборки нашего устройства, мы использовали большое количество проводов для соединения микроконтроллера со светодиодным экраном. Но такой способ сборки является ненадежным и имеет множество недостатков таких как: если отойдет всего 1 провод, то нарушится работа всего экрана; правильно подключить 16 проводов к 1 устройству ещё можно, но если мы хотим запустить производство небольшой серии, то возникнут большие сложности; к тому же провода занимают слишком много места в корпусе. Поэтому было решено разработать печатную плату, которая устранил все эти недостатки и позволит установить на плату тактовую кнопку для переключения режимов работы устройства.

Цена платы получилась даже меньше, чем цена проводов, что также является большим плюсом.

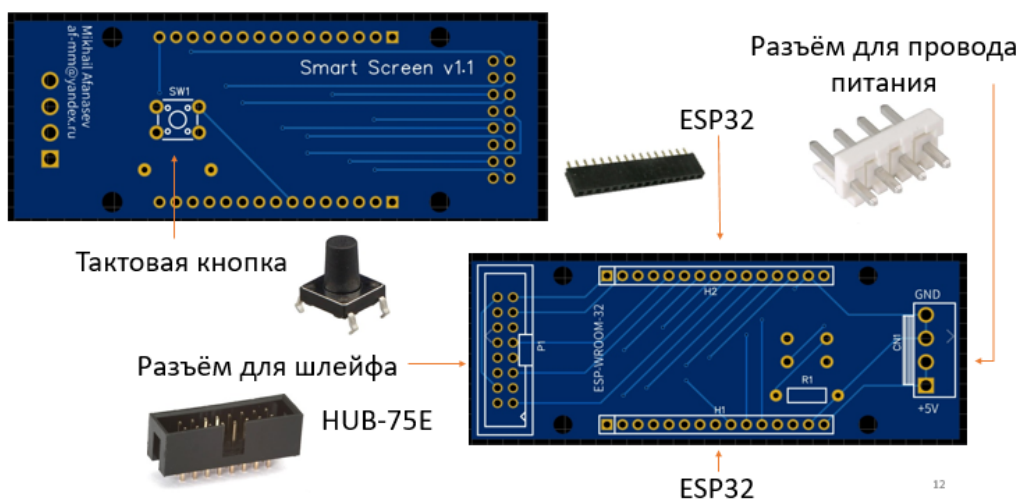


Рис. 6. Разработка печатной платы

РАЗРАБОТКА ПЛАСТИКОВОГО КОРПУСА

При разработке корпуса, мы должны были учесть то, каким способом будет изготовлен корпус, материал и способ создания модели и чертежа.

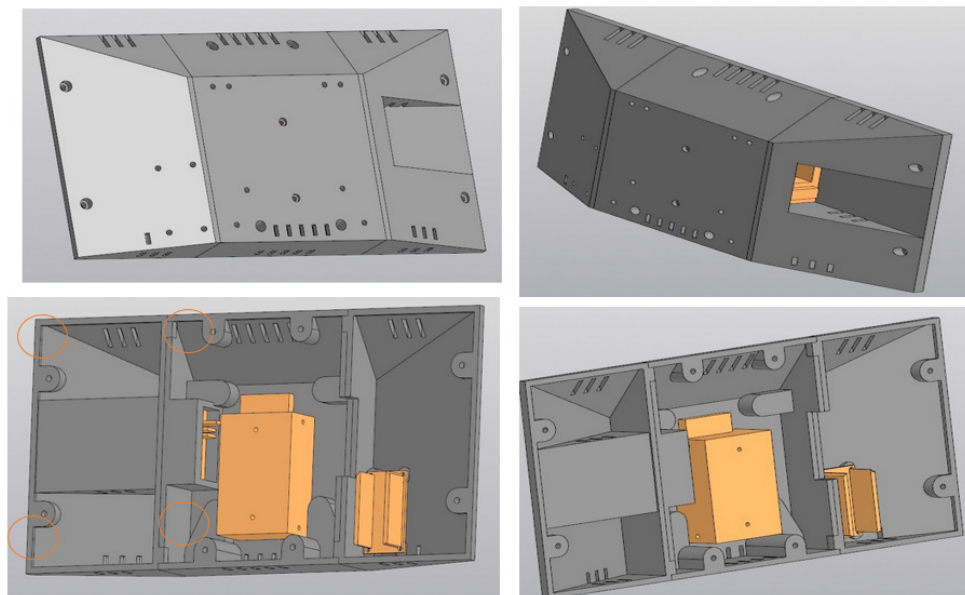


Рис. 7. Разработка корпуса

Для разработки чертежа была использована программа «Компас 3D». В ней мы создали 3D модель будущего корпуса, которую в дальнейшем распечатали на 3D принтере, используя черный ABS-пластик. У нас получилось сделать корпус далеко не с первого раза. Во время изготовления мы столкнулись со следующими проблемами:

1. Целиковый корпус не влезал в принтер. Нам пришлось разбить его на 3 примерно части, между которыми мы предусмотрели механизмы сборки и стыковки.
2. На печать одного корпуса уходит очень много времени. Когда печать происходит неправильно, мы теряем еще дополнительные часы на возобновление печати данной детали. Представьте, что одна часть печаталась 3 часа и вы увидели дефект. Тогда придется начинать печать заново.
3. Самая главная трудность, с которой мы столкнулись – выбор пластика. С самого начала у нас на руках был ABS пластик. У него есть усадка. Пластик в процессе остывания искривляется и отклеивается на поверхности стола

С трудом поборов эту проблему, нам удалось сделать первый прототип корпуса, но он получился далеко не идеальным, с шершавостями и небольшими дефектами. В наши планы входит рассмотрение варианта производства путем литья пластика. Но для этого нам нужно создать идеальную мастер-модель. Для этой цели мы хотим использовать фотополимерный 3д принтер. Мы приобрели новый пластик PLA, но из-за сложившейся ситуации с коронавирусом еще не успели его протестировать.

Разработанный корпус позволяет не только строго зафиксировать внутреннее содержимое, что исключает возможность отсоединения проводов внутри и свободного перемещения всех компонентов, но также позволяет слегка утяжелить нижнюю часть корпуса. Это не повлияет на установку корпуса на вертикальную поверхность, но значительно облегчает его эксплуатацию на горизонтальной.

Также мы создали макет другого корпуса, который будет выступать в качестве демонстрационно-лабораторного стенда. Он поможет студентам в изучении работы дисплея. Задача заключалась в том, чтобы сделать его безопасным и удобным для работы с экраном. Данный корпус будет выполнен из фанерных листов с помощью лазерного станка. Все электронные компоненты, которые могут навредить здоровью, будут надежно спрятаны внутри, а те, которые понадобятся для изучения и не несут опасности, будут находиться в свободном доступе.

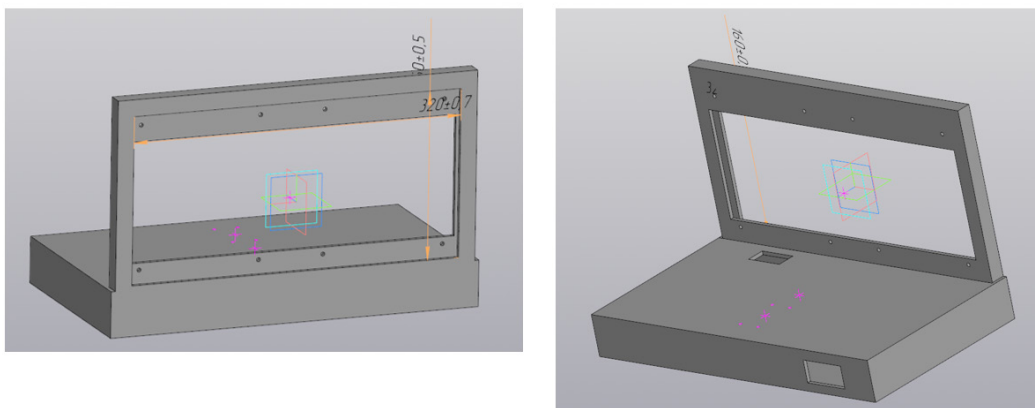


Рис. 8. Корпус для студенческих работ

ФИНАЛЬНАЯ СБОРКА ПЕРВОГО ПРОТОТИПА

После получения всех комплектующих, мы произвели сборку первого прототипа.



Рис. 9. Собранное устройство, вид сзади



Рис. 10. Собранное устройство, вид спереди

ЗАКЛЮЧЕНИЕ

В ходе нашей проектной работы, мы смогли разработать устройство, удовлетворяющее всем нашим требованиям. Экран действительно не имеет привязки к электронной очереди и поддерживают быструю интеграцию в любую программу, стоимость получилась в ниже конкурентов, а код и команды управления для клиентов мы, конечно, оставим открытым. Устройство имеет удобный веб-интерфейс, где все необходимые пользователю действия вынесены в отдельные кнопки.

Объем проделанной работы достаточно большой, и рассказать подробно про каждый отдельный аспект в рамках одной работы не представляется возможным, поэтому описали только наиболее важные и ключевые моменты выполненные в ходе нашей работы.

Наши дальнейшие планы:

1. Создание корпуса с помощью метода литья пластика в силиконовые формы
2. Создание учебного демонстрационно-лабораторного стенда
3. Интеграция возможности широкополосной передачи данных по Wi-Fi на несколько Smart Screen-ов для снижения нагрузки на канал передачи данных

Библиографический список

1. Бакалов В. П., Игнатов А. Н., Крук Б. И. Основы теории электрических цепей и электроники : учебник для высших учебных заведений. М. : Радио и связь, 1989.
2. Каган Б. М., Сташин В. В. Основы проектирования микропроцессорных устройств автоматики: производств. изд. М.: Энергоатомиздат, 1987.
3. Хвоц С. Т., Попов Е. А., Варлинский Н. Н. Микропроцессоры и микроЭВМ в системах автоматического управления: справочник. Л.: Машиностроение. Ленингр. отд-ние, 1987.
4. Васильев А. Е. Микроконтроллеры. Разработка встраиваемых приложений: учеб. пособие / С.-Петербург. гос. политехн. ун-т. СПб.: СПбГПУ, 2003.
5. Александров Е. К., Грушвицкий Р. И., Куприянов М. С. и др. Микропроцессорные системы: учеб. пособие / ред. Д. В. Пузанков. СПб.: Политехника, 2002.

УДК 004.85

А. Д. Драноцук

студент кафедры инфокоммуникационных систем

А. И. Веселов – кандидат технических наук, доцент – научный руководитель

ИССЛЕДОВАНИЕ ВЛИЯНИЯ ПРЕДОБУЧЕНИЯ СВЕРТОЧНЫХ СЕТЕЙ В ЗАДАЧЕ ПОИСКА ЧЕЛОВЕКА НА НЕСКОЛЬКИХ КАМЕРАХ

Рассмотрена задача поиска человека в выборе из большого числа людей по его изображению. Дано описание одного из самых популярных баз данных в задаче поиска человека. Приведены модели сверточных сетей для решения данной задачи, а также рассмотрен способ по повышению точности распознавания и скорости обучения.

В наше время, когда уже сложно уследить за потоком людей в общественных местах, с каждым днем растет необходимость в автономном поиске одного и того же человека на разных камерах. Эта, как казалось, очень простая задача, которую может решить даже ребенок, дается довольно тяжело компьютеру. Большим шагом в сторону решения данной проблемы стало появление сверточных нейронных сетей, но и они не всегда дают точный ответ и требуют много времени для обучения. В данной работе на примере моделей *OSNet*, *mobilenet* рассмотрим влияние предобучения на итоговую оценку точности сети и скорость ее обучения в задаче поиска человека на нескольких камерах.

Предобучение это обучение модели на не целевой базе данных перед основным обучением. На данный момент существует несколько подходов решения данной задачи:

- 1) Использующие калибровку камеры и геометрию комнаты.
- 2) Не использующие калибровку камеры и геометрию комнаты

Для того чтобы поиск человека не зависел от информации о внешней среде и геометрии пространства используют второй подход. Он заключается в активной или пассивной генерации признаков человека, на основе которых можно построить метрику и различать людей. В данной статье будет рассматриваться только активный способ генерации отличительных признаков с помощью сверточных нейронных сетей, а поиск человека будет осуществляться евклидовой метрикой по K ближайшим соседям из выборки.

Для оценки качества сгенерированных, признаков обычно используют большие базы данных. *Market1501* является одной из самых популярных баз данных среди тех, кто решает задачу поиска человека. Она содержит в себе изображения людей в разных локациях и позах.



Рис. 1. Пример изображений Market1501

Данную базу данных часто используют, потому что в ней 32668 изображений 1501 человека в разрешении 128 на 64 пикселя в формате jpg, что делает ее одной из самых больших и удобных баз данных для обучения и проверки построенных моделей.

Для генерации признаков были обучены две модели:

- 1) OSNet [3]
- 2) Mobilenet [1]

Архитектура сверточных сетей OSNet и Mobilenet представлена в таблицах 1 и 2 соответственно.

Таблица 1

Слой	Размер выхода слоя	OSNet
Сверточный слой №1 Субдискретизация по максимуму	(128x64, 64) (64x32, 64)	Размер свертки (7x7), сдвиг 2 Размер (3x3), сдвиг 2
Сверточный слой №2	(64x32, 256) (64x32, 256)	OS блок OS блок
Переходный слой №1 Субдискретизация по среднему значению	(64x16, 256) (32x16, 256)	Размер свертки (1x1) Размер (2x2), сдвиг 2
Сверточный слой №3	(32x16, 384) (32x16, 384)	OS блок OS блок
Переходный слой №2 Субдискретизация по среднему значению	(32x16, 384) (16x8, 384)	Размер свертки (1x1) Размер (2x2), сдвиг 2
Сверточный слой №4	(16x8, 512) (16x8, 512)	OS блок OS блок
Сверточный слой №5	(16x8, 512)	Размер свертки (1x1)
Субдискретизация	(1x1, 512)	Субдискретизация по среднему значению
Полносвязный слой	(1x1, 512)	

Таблица 2

Архитектура MobileNet

Слой	Размер входа слоя	Размер фильтра/сдвиг
Сверточный слой №1	(224x224x3)	(3x3x3x32), сдвиг 2
Глубинный сверточный слой №1	(112x112x32)	(3x3x32), сдвиг 1
Сверточный слой №2	(112x112x32)	(1x1x32x64), сдвиг 1
Глубинный сверточный слой №2	(112x112x64)	(3x3x64), сдвиг 2
Сверточный слой №3	(56x56x64)	(1x1x64x128), сдвиг 1
Глубинный сверточный слой №3	(56x56x128)	(3x3x128), сдвиг 1
Сверточный слой №4	(56x56x256)	(1x1x128x128), сдвиг 1
Глубинный сверточный слой №4	(56x56x128)	(3x3x128), сдвиг 2
Сверточный слой №5	(28x28x128)	(1x1x128x256)
Глубинный сверточный слой №5	(28x28x256)	(3x3x256)
Сверточный слой №6	(28x28x256)	(1x1x256x256)
Глубинный сверточный слой №6	(14x14x512)	(3x3x256)
5 x глубинный сверточный слой 5 x сверточный слой	(14x14x512) (14x14x512)	(3x3x512) (1x1x512x512)
Глубинный сверточный слой №7	(7x7x512)	(3x3x512)
Сверточный слой №7	(7x7x1024)	(1x1x512x1024)
Глубинный сверточный слой №8	(7x7x1024)	(3x3x1024)
Сверточный слой №8	(7x7x1024)	(1x1x1024x1024)
Субдискретизация	(7x7x1024)	(7x7)

Данные сети являются одними из самых легких и точных сетей по генерации признаков.

Без предобучения *OSNet* и *Mobilenet* получают следующие оценки: Средняя точность определения по К ближайшим соседям составила 81.0 и 67.3 процентов соответственно, а точность определения по 1 ближайшему соседу 93.6 и 85.6 процентов.

Таблица 3

Результаты без предобучения

Название сети	Кол-во параметров (10^6)	Вычислительная сложность (GFlops)	mAP (%)	Top1 (%)
OSNet	2.2	0.98	81.0	93.6
Mobilenet	4.3	0.4	67.3	85.6

Для обучения *OSNet* использовался оптимизатор *sgd* с установленным шагом обучения 0.065, который затухал в 10 раз после 150, 225, 300 эпох. Обучение длилось 350 эпох на наборах из 64 изображений.

Для обучения *Mobilenet* использовался оптимизатор *AMSGrad* с установленным шагом обучения 0.0006. Обучение длилось 80 эпох на наборах из 64 изображений.

Рассмотренные модели получают неплохие результаты, но, так как задача генерации признаков требует много данных, а возможность расширить выборку новыми людьми отсутствует, их можно попробовать улучшить с помощью предобучения на другой базе данных, не связанной с задачей поиска человека [4]. Для предобучения был использован *imagenet* [2]. Это огромная база данных 14 000 000 изображений по 20000 категориям.

После предобучения на *imagenet* *OSNet* и *Mobilenet* получают следующие оценки: Средняя точность определения по К ближайшим соседям составила 82.7 и 68.5 процентов соответственно, а точность определения по 1 ближайшему соседу 94.8 и 87.0 процентов.

Таблица 4

Результаты с предобучением

Название сети	Кол-во параметров (10^6)	Вычислительная сложность (GFlops)	mAP	Rank1
OSNet	2.2	0.98	82.7	94.8
Mobilenet	4.3	0.4	68.5	87.0

Для обучения *OSNet* использовался оптимизатор *sgd* с установленным шагом обучения 0.065, который затухал в 10 раз после 100, 175, 250 эпох. Обучение длилось 300 эпох на наборах из 64 изображений.

Для обучения *Mobilenet* использовался оптимизатор *AMSGrad* с установленным шагом обучения 0.0006. Обучение длилось 80 эпох на наборах из 64 изображений. С 1 по 10 эпохи сверточные слои у обеих сетей были заморожены и обучался только классификатор, это было сделано для того чтобы ошибка классификатора не портила признаки сформированные на *imagenet*.

Анализ показал, что в ряде случаев нейронной сети может не хватить данных для обучения, тогда перед тем как обучать ее генерировать признаки, имеет смысл предобучить ее на другой базе данных с большим количеством данных. В таком случае предобучение может дать небольшой прирост в точности модели и значительно ускорить процесс обучения.

Для рассмотренной задачи прирост средней точности определения по К ближайшим соседям составил 1.7 и 1.2 процентов соответственно, а прирост точности определения по 1 ближайшему соседу 1.2 и 1.4 процентов при тех же или меньших затратах по времени на обучение.

Библиографический список

1. Andrew G. Howard, Menglong Zhu, Bo Chen et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications / Cornell University. 2017.
2. Ross Girshick, Piotr Dollár. Rethinking ImageNet Pre-training Kaiming / Cornell University. 2018.

3. *Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro et al.* Learning Generalisable Omni-Scale Representations for Person Re-Identification / Cornell University. 2019.
4. *Olga Russakovsky, Jia Deng, Hao Su et al.* ImageNet Large Scale Visual Recognition Challenge / Cornell University. 2014.

УДК 004.056.2

Д. А. Зыков

студент кафедры безопасности информационных систем

А. В. Афанасьева – старший преподаватель – научный руководитель

УНИВЕРСАЛЬНЫЙ СТЕНД ДЛЯ ТЕСТИРОВАНИЯ ПРОИЗВОДИТЕЛЬНОСТИ СЕРВИСОВ

В настоящее время происходит непрерывный рост услуг, предоставляемых при помощи веб-приложений. Среди организаций набирает популярность размещение своих сервисов на облачных серверах, которые предлагают инфраструктуру в виде услуги [1].

В связи с ростом популярности облачных серверов широкое распространение получила технология контейнеризации, позволяющая решить проблему зависимостей в разных окружениях для приложений, при этом имеющая большую скорость работы и меньшее потребление ресурсов, чем виртуальные машины [2].

Несмотря на используемые технологии и архитектуру приложений, важнейшим этапом разработки программного обеспечения является тестирование. Одним из главных этапов в тестировании веб-приложений является тестирование производительности.

Для тестирования контейнеризированных приложений в среде близкой к используемой на облачных серверах был разработан тестовый стенд, удовлетворяющий следующим требованиям:

1. Работа и сбор показателей в реальном времени.
2. Масштабируемость.
3. Отказоустойчивость.

Разработанный тестовый стенд позволяет моделировать взаимодействие между сервисом, предоставляющим услуги, и клиентами. Под сервисом в данном случае понимается любое контейнеризированное веб-приложение, предоставляющее свои услуги клиентам. На рис. 1 приведена общая структурная схема моделируемой системы.

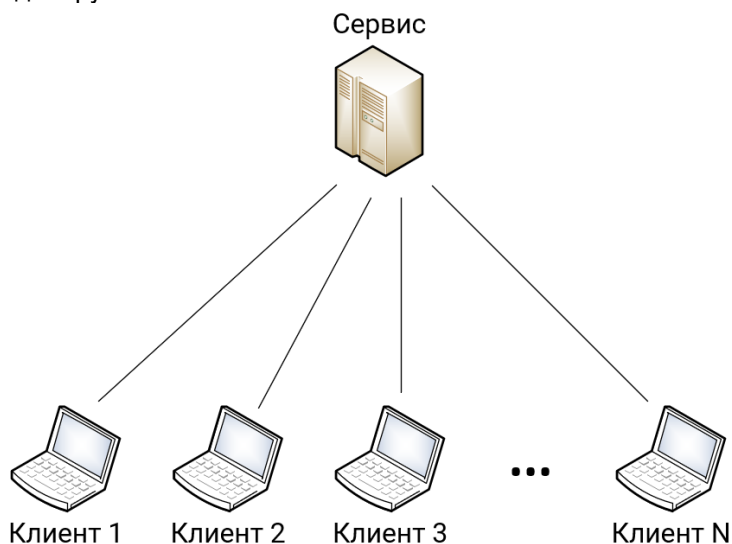


Рис. 1. Схема моделируемой системы

Для реализации возможности масштабирования клиент-серверного взаимодействия, а также для отказоустойчивости в работе тестового стенда, был использован контейнер-оркестратор Kubernetes, позволяющий конфигурировать необходимое количество единиц запущенных приложений, а также реализующий отказоустойчивую кластерную архитектуру [3].

В качестве инфраструктуры на базе которой разворачивается Kubernetes кластер, используются виртуальные машины, развёрнутые при помощи гипервизора KVM.

Схема инфраструктуры тестового стенда приведена на рис. 2.

В качестве основной операционной системы, используемой в основе тестового стенда, был использован наиболее популярный linux дистрибутив: Ubuntu 18.04 LTS. Данный дистрибутив был выбран в связи с высокими показателями производительности и совместимостью с используемым аппаратным обеспечением.

В качестве гипервизора для разворачивания виртуальных машин и виртуальных сетей был выбран гипервизор с открытым исходным кодом KVM, так как данное решение имеет наибольшую эффективность в использовании аппаратного обеспечения, а также имеет набор средств для управления из командной строки, который позволяет автоматизировать работу с инфраструктурой при помощи shell-скриптов.

Основные проведённые этапы конфигурирования хост-системы:

1. Проверка поддержки виртуализации аппаратным обеспечением.
2. Установка необходимых пакетов и их зависимостей.
3. Разметка дискового пространства для хранения данных виртуальных машин.
4. Выделение дискового пространства для системы хранения данных ScaleIO.

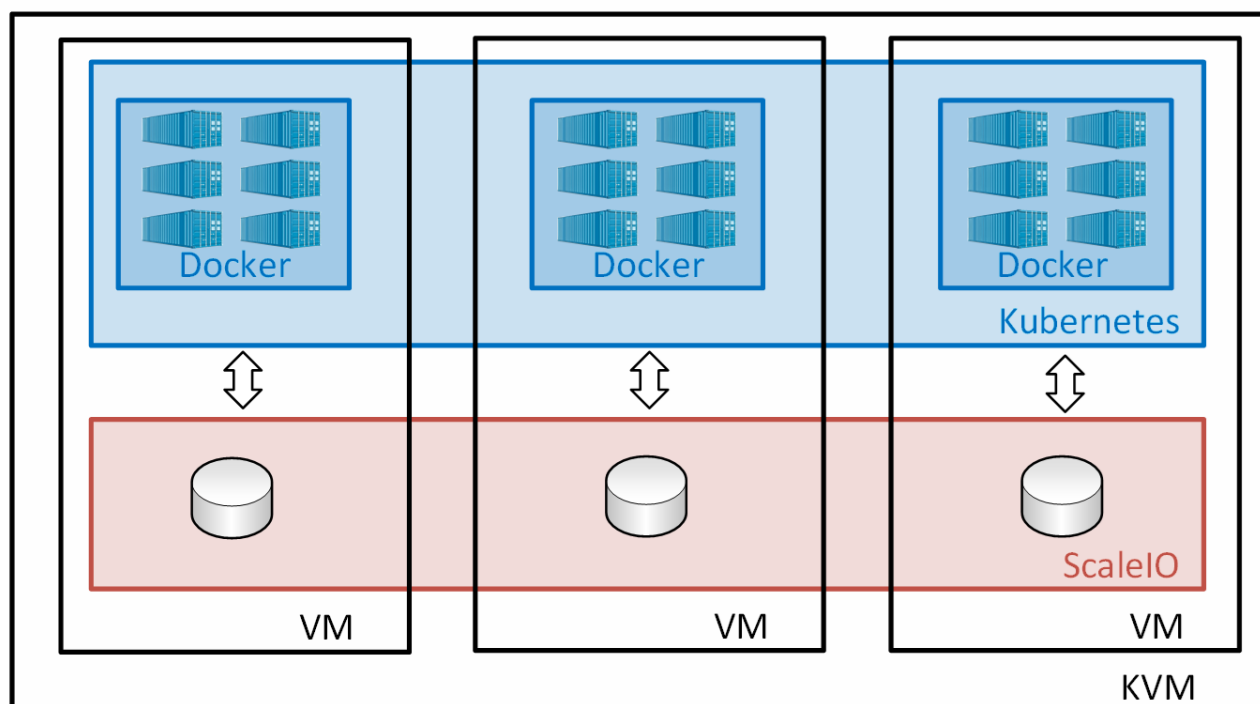


Рис. 2. Схема тестового стенда

Для упрощения разворачивания инфраструктуры тестового стенда были автоматизированы следующие этапы работы с гипервизором:

1. Создание и клонирование виртуальных машин.
2. Создание виртуальных сетей.
3. Создание виртуального дискового пространства для данных виртуальных машин.
4. Подключение к виртуальным машинам дискового пространства для системы хранения данных ScaleIO.

Для визуализации метрик, полученных в ходе тестирования приложений, реализован веб-сервис "SUAI-Dashboard" на языке Python с использованием веб-фреймворка Flask, веб-сервера Gunicorn и библиотек: Plotly для отображения графиков и Bootstrap для стилизации графического интерфейса. Интерфейс реализованного приложения приведён на рис. 3. Реализованный сервис отображает данные в виде линейных, круговых либо столбчатых диаграмм. Реализованный веб-интерфейс имеет адаптивный дизайн и два стиля оформления (светлое и тёмное), что позволяет использовать его на различных устройствах с отличающимися параметрами. Наборы метрик, отображаемые сервисом, хранятся в базе данных MySQL. В качестве системы хранения данных, предоставляющей дисковое пространство для базы данных, используется ScaleIO – программно-определяемая отказоустойчивая

сеть хранения данных, позволяющая при необходимости масштабировать необходимое пространство.

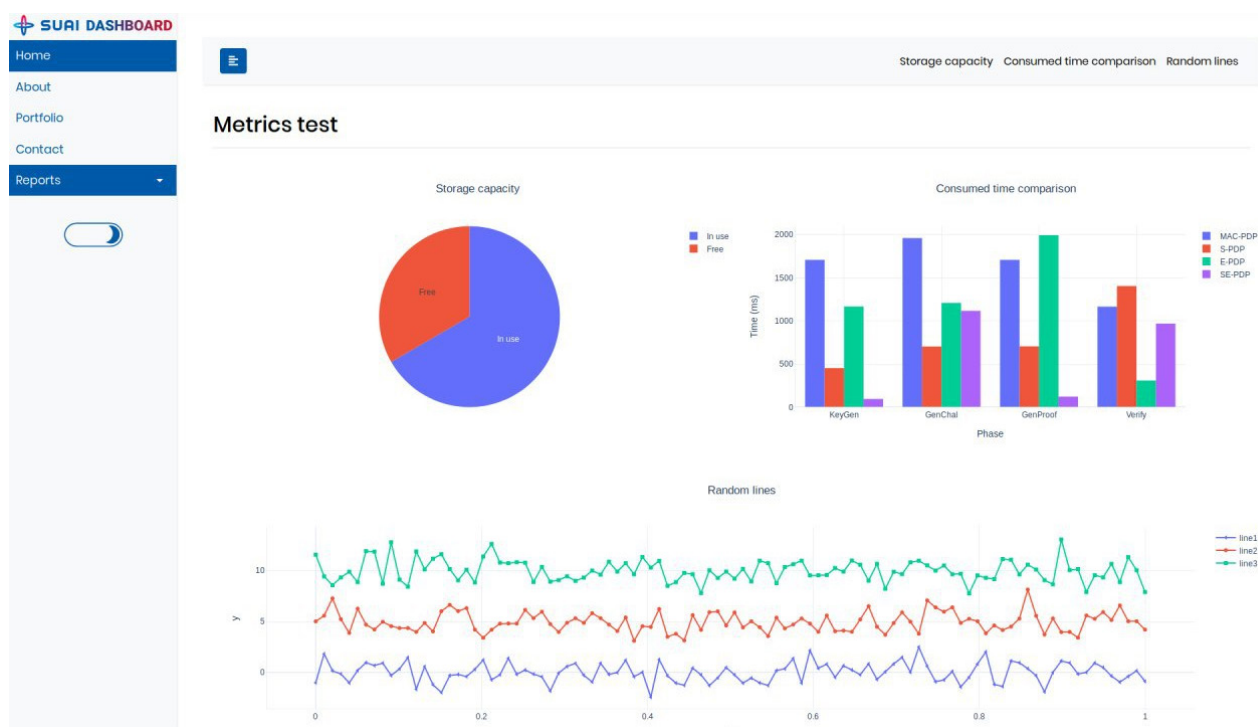


Рис. 3. Интерфейс реализованного веб-приложения

Для сбора показателей производительности в реальном времени, в Kubernetes кластере была развернута система мониторинга с открытым исходным кодом Prometheus. Данная система позволяет собирать следующие виды метрик:

1. Загрузка процессора.
2. Объем используемой оперативной памяти.
3. Использование сетевых интерфейсов.

Система Prometheus позволяет гибко настраивать множество параметров таких как: частота сбора статистики, объем хранимых данных, а также собираемые метрики, что позволяет при необходимость добавить необходимые пользователю характеристики. Важной особенностью системы мониторинга Prometheus является встроенный веб-интерфейс, позволяющий отображать необходимые показатели производительности в формате графиков.

Для корректной работы системы хранения данных были проведены следующие предварительные шаги конфигурирования системы:

1. Обновление ядра linux на виртуальных машинах до поддерживаемого в версии ScaleIO 2.5.0.
2. Установка необходимых для системы хранения данных пакетов и их зависимостей
3. Проверка конфигурации и работоспособности ранее созданной виртуальной сети.
4. Подготовка CSV файла, содержащего топологию будущей системы хранения.

Основными шагами установки системы стали:

1. Установка ScaleIO Gateway – программного обеспечения позволяющего проводить установку, масштабирование и конфигурирование сети хранения данных, а также предоставляющего Restfull API для управления установленной системой.

2. Установка системы по заранее подготовленной топологии в виде CSV-файла.

3. Установка на виртуальные машины драйверов необходимых для компонента ScaleIO SDC (ScaleIO Data Client), реализующего взаимодействие пользовательских приложений с томами данных.

Функционал разработанной тестовой системы был испытан опытным путём на сервисах, предоставляющих услугу доказательства обладания данными. Было протестировано три приложения, реализующие различные протоколы доказательства обладания данными. В результате тестирования

были получены следующие метрики: среднее время выполнения основных операций протоколов доказательства обладания данными, объём используемой оперативной памяти сервисом и клиентами, процессорное время, затрачиваемое сервисом и клиентами.

Разработанный тестовый стенд позволяет проводить тестирование производительности контейнеризированных приложений и наглядно отображать собираемые характеристики.

Библиографический список

1. Cloud Native Architectures: Design high-availability and cost-effective applications for the cloud / T. Laszewski // Packt Publishing. 2018.
2. Using Docker / A. Mouat. O'Reilly Media, Inc., 2015.
3. Cloud Native DevOps with Kubernetes: Building, Deploying, and Scaling Modern Applications in the Cloud / J. Arundel, J. Domingus // O'Reilly Media, Inc., 2019.

УДК 004.02

К. Д. Камышников

студент кафедры инфокоммуникационных систем

П. Е. Бернанс

студент кафедры безопасности информационных систем

И. А. Пастушок – кандидат технических наук – научный руководитель

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ПОЗИЦИОНИРОВАНИЯ УСТРОЙСТВА НА ОСНОВЕ ТЕХНОЛОГИИ BLUETOOTH

Введение

В настоящее время существует необходимость в определении местоположения внутри помещения. Это может потребоваться в местах, с ограниченным доступом в определенные комнаты и помещения (на каких-нибудь предприятиях), на военных объектах либо на объектах с классификацией уровня безопасности.

Для создания такого приложения возможно использование трех различных решений: Видеотрекинг, технология Ultra-Wide Band и технология BLE.

Видеотрекинг [1] – определение местоположения движущегося объекта при помощи камер. Преимуществом данного метода позиционирования является то, что для него не нужны маячки или иные устройства на отслеживаемом предмете или человеке, но при этом для применения данного метода требуется установка камер на всей наблюдаемой территории. Также для данного метода позиционирования требуются значительные вычислительные ресурсы и сложные алгоритмы анализа кадров видео.

Ultra-Wide Band (UWB) [2, 3] – это беспроводная технология связи, использующая короткие импульсы с малой энергией в широком диапазоне частот. Применение данной технологии позволяет добиться высокой точности позиционирования. Однако значительным недостатком данного способа позиционирования является высокая стоимость необходимого оборудования.

Технология BLE (Bluetooth Low Energy) [4]. Достаточно распространенная технология передачи данных между устройствами (которая может быть использована в нашем проекте). Достоинствами данной технологии являются низкая стоимость оборудования и широкое распространение технологии, что позволяет интегрировать наше приложение с мобильным телефоном. Недостатком данной технологии является низкая точность определения местоположения.

Целью работы является создание масштабируемого решения для определения местоположения на основе технологии BLE с мобильным приложением.

Описание решения

Для решения поставленной задачи мы выбрали решение на основе технологии BLE.

Технология BLE это усовершенствованная технология Bluetooth, используемая для передачи данных между устройствами на небольших расстояниях. Технология была выпущена в 2009 году, а её наиболее существенным достоинством является сверхмалое энергопотребление. Данная технология есть во многих современных смартфонах, а Bluetooth-маячки стоят дешево. С помощью BLE можно определить местоположение в помещении с точностью до комнаты, этого нам вполне достаточно.

Наша программа делится на две части, мобильное приложение, устанавливаемое на смартфон, и серверное приложение. Для разработки мобильного приложения использовался язык программирования Java 8 и IDE-Android Studio[5][6]. Приложение поддерживается на всех Android-устройствах с версией Android 7.0 и выше. Для разработки серверного приложения использовался язык программирования Java 11 и IDE-Eclipse. Приложение поддерживается на ОС Linux.

Структура и устройство нашей программы представлены на C4 диаграмме (Рис. 1, Рис 2, Рис 3, Рис 4, Рис. 5).

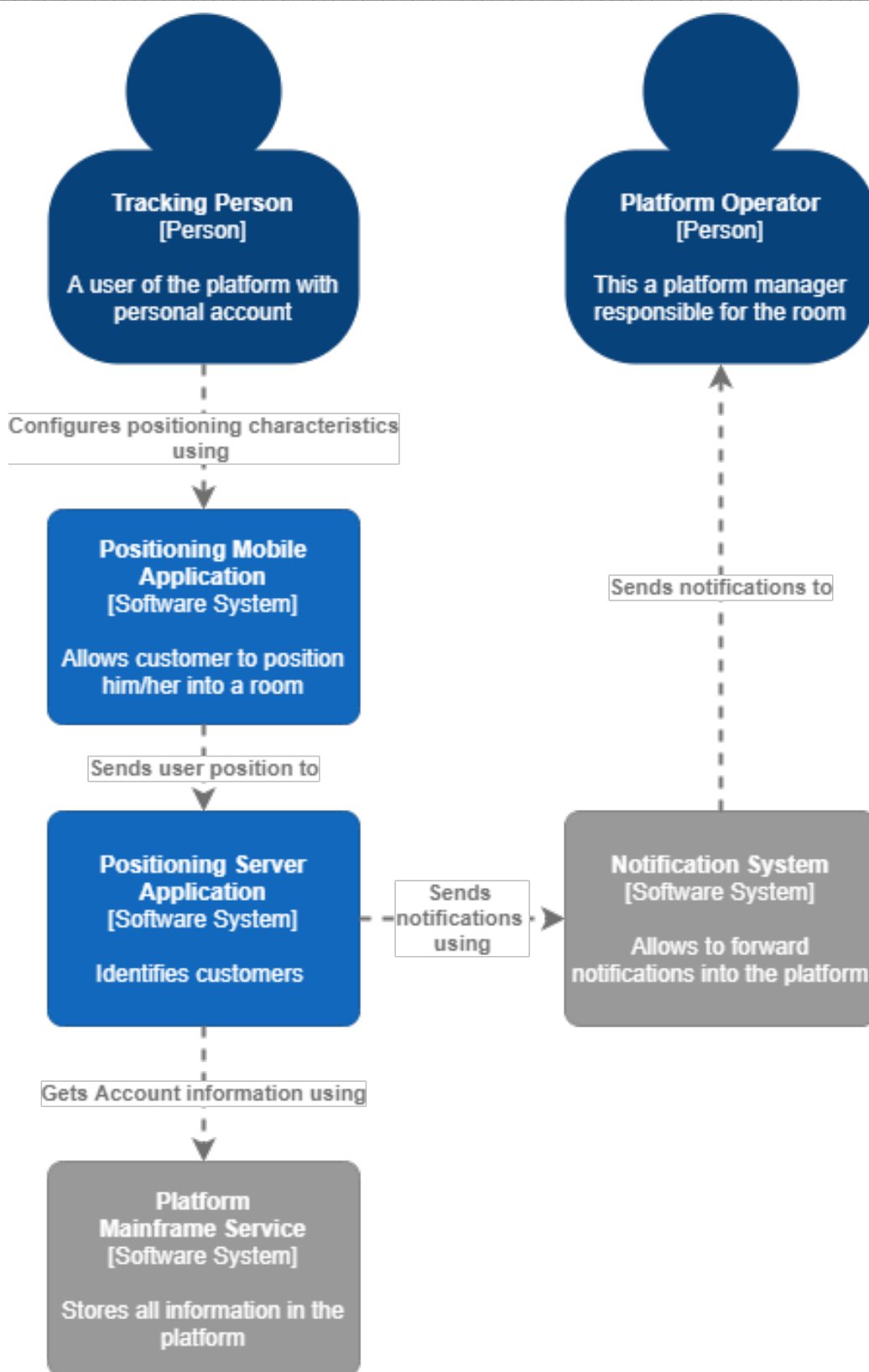


Рис. 1. Уровень C1. Context.

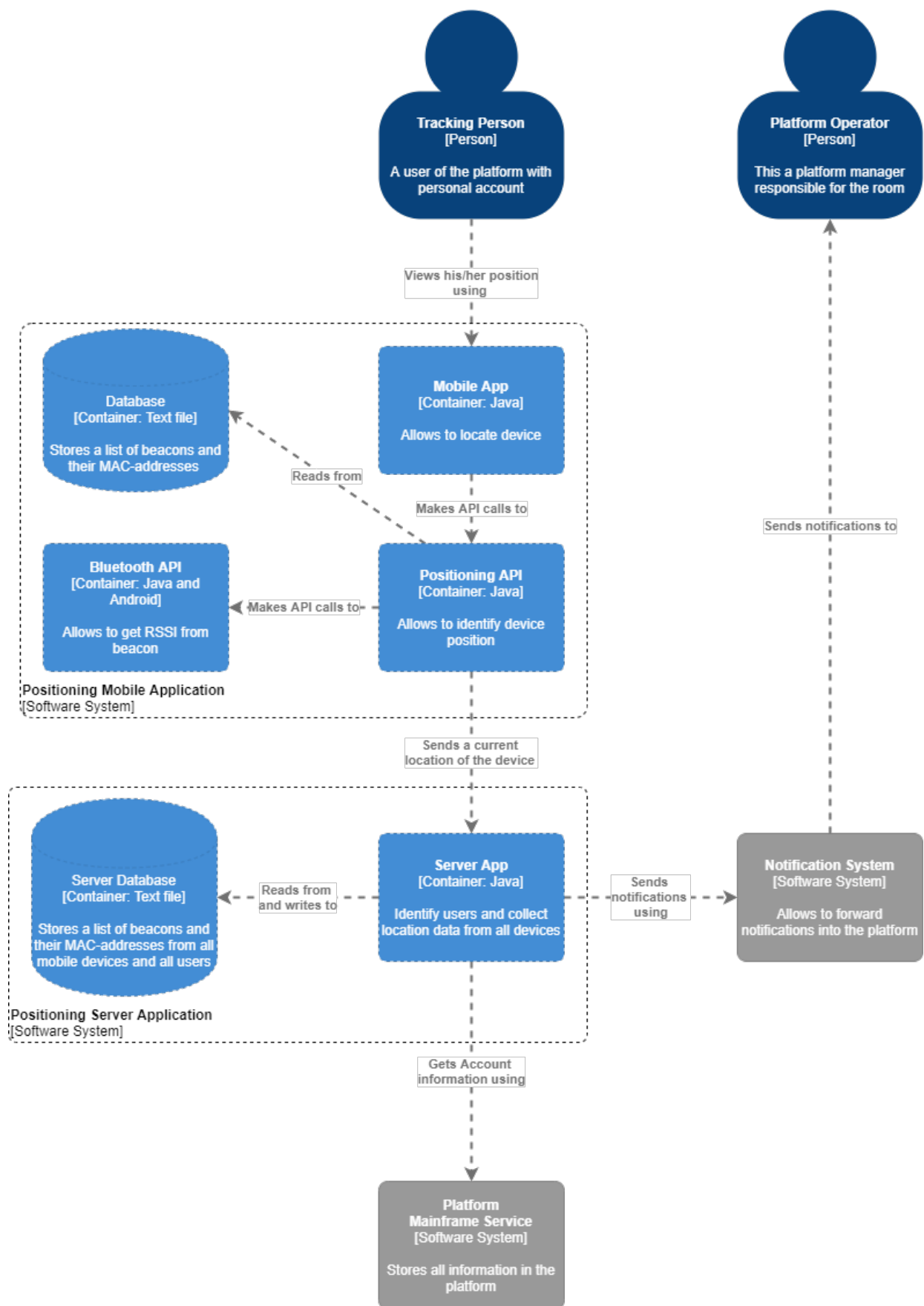


Рис. 2. Уровень C2. Containers.

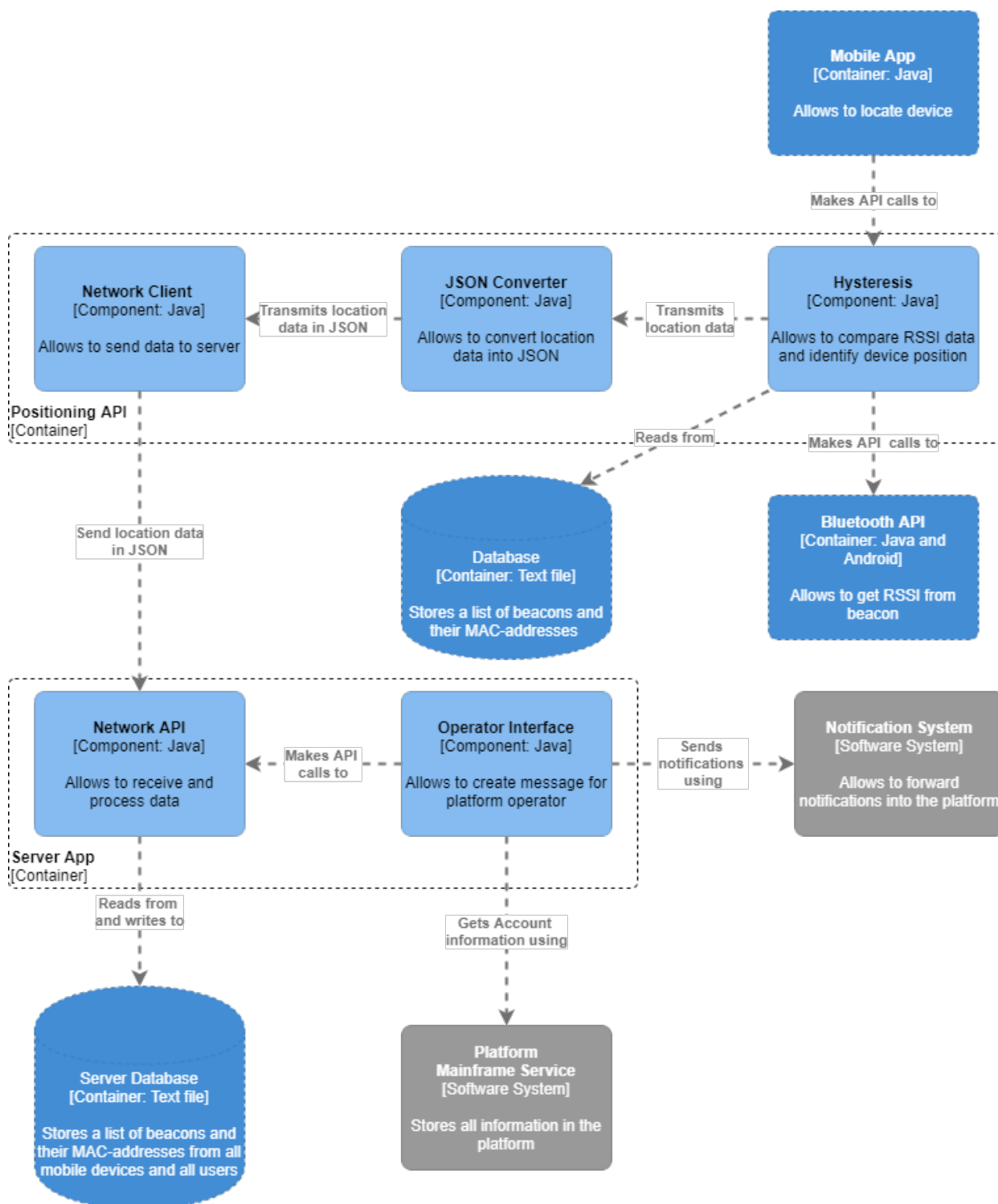


Рис. 3. Уровень C3. Components

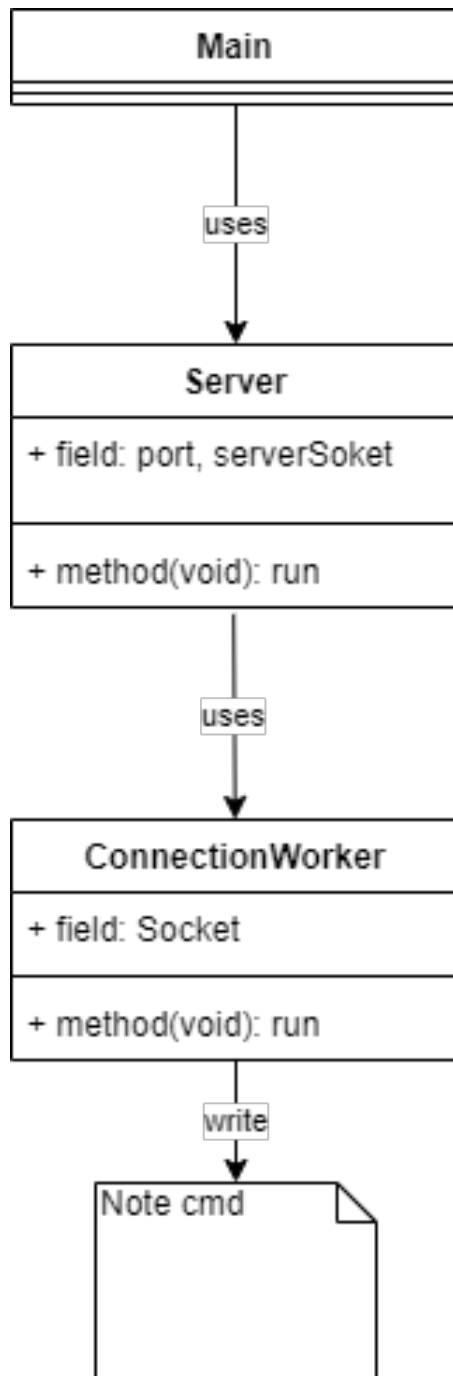


Рис. 4. Уровень C4. Code. Серверное приложение

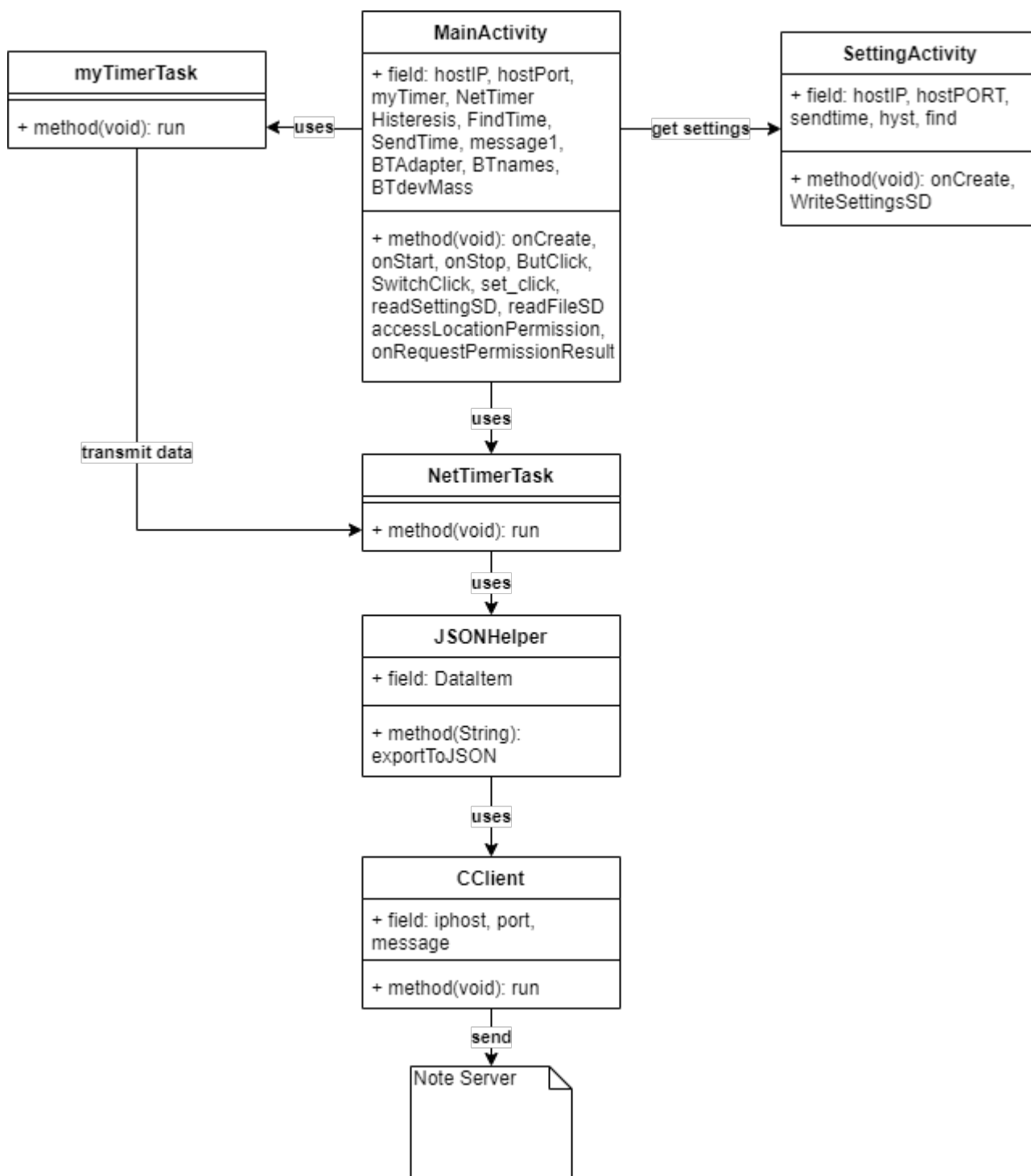


Рис. 5 Уровень C4. Code. Мобильное приложение

Алгоритм гистерезиса в нашей программе используется для предотвращения ложных переключений между маячками. Принцип работы алгоритма показан на рис. 6. Алгоритм запускается при старте поиска устройств и дальше работает по схеме, представленной на рис. 7.

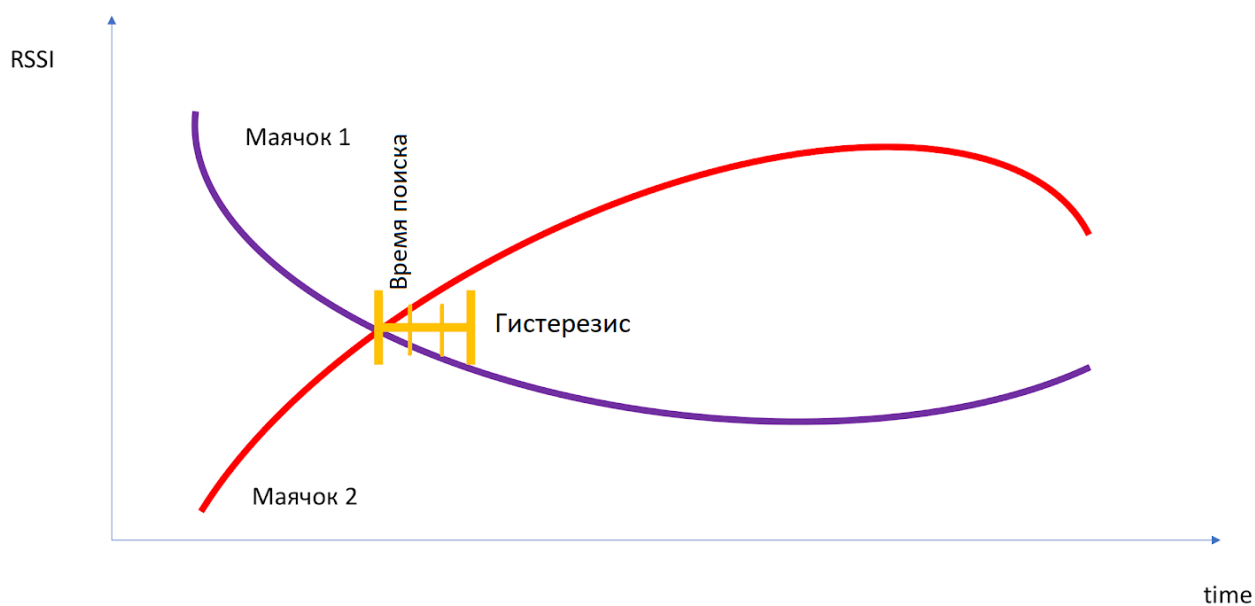


Рис. 6. Принцип работы алгоритма гистерезиса

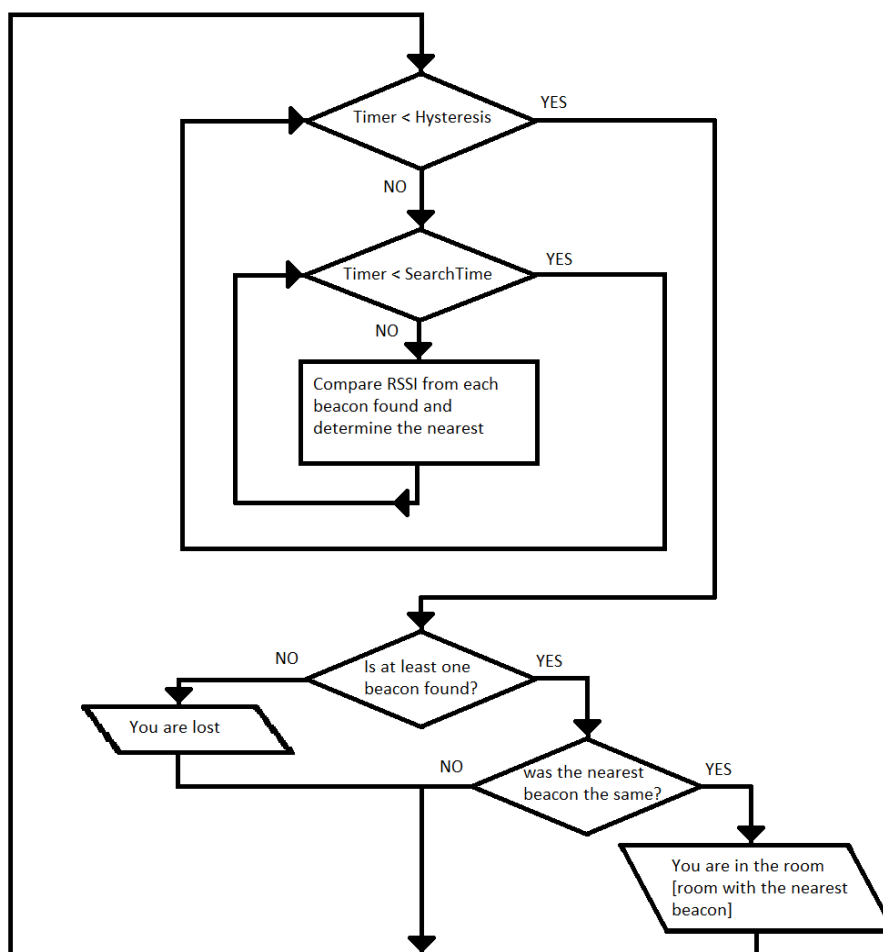


Рис. 7. Схема работы алгоритма гистерезиса

Тестовые сценарии

Для тестирования мобильного приложения использовались следующие сценарии:

1. Сценарий 1 – переход от одного маячка до другого по прямой;

2. Сценарий 2 – циклический обход всех маячков;
3. Сценарий 3 – определение местоположения на границе между маячками.

Рис. 8. Настройки приложения

Таблица настроек:

Для настройки приложения мы добавили экран настроек (Рис. 8).

1. ServerIP – IP адрес сервера;
2. Port – порт, открытый на сервере, для серверной части приложения;
3. Send net message – флажок, установлен для отправки сообщения на сервер;
4. Количество комнат(маячков) – количество используемых для теста маячков – 4;
5. send time – время отправки сообщения на сервер – 15 секунд;
6. your device number – номер устройства;
7. hysteresis – время гистерезиса – для теста выставлялись значения 15, 20, 25, 30;
8. search time – время поиска устройств – для теста выставлялись значения 5, 10.

Результаты тестирования

В ходе тестирования мы убедились в работоспособности приложения и определили оптимальные настройки. Результаты тестирования представлены ниже в табл. 1–3.

Таблица 1

Результаты эффективности приложения в Сценарии 1

Сценарий 1 – время переключения, с.		Время поиска, с.	
		5	10
Время гистерезиса, с.	15	30	-
	20	45	40
	25	65	-
	30	60	150

Таблица 2

Результаты эффективности приложения в Сценарии 2

Сценарий 2 – время переключения, с.		Время поиска, с.	
		5	10
Время гистерезиса, с.	15	30	-
	20	40	60
	25	45	-
	30	60	120

Таблица 3

Результаты эффективности приложения в сценарии 3

Сценарий 3 – время переключения, с.		Время поиска, с.	
		5	10
Время гистерезиса, с.	15	20	-
	20	20	30
	25	25	-
	30	30	40

При увеличении времени поиска и гистерезиса увеличивается и время определения нового местоположения, но уменьшается количество ложных срабатываний. Из таблиц видно, что оптимальные настройки гистерезиса и времени поиска – 15 и 5 секунд соответственно. Во всех случаях задержка сообщения на сервере составляла менее 1 секунды.

Заключение

Разработано мобильное приложение для Android. Приложение устанавливается и запускается на Android-устройствах с версией Android 7.0 и выше.

Разработано серверное приложение для ОС Linux. Приложение контейнеризовано с помощью Docker.

Библиографический список

1. Тершуков И. А. Разработка интеллектуальной системы видеонаблюдения. URL: https://real-trac.com/ru/company/blog/razrabotka_intellektualnoj_sistemy_videonablyudeniya/ (дата обращения: 18.05.2020).
2. Лекнин В. Технологии идентификации и позиционирования в режиме реального времени. URL: <https://habr.com/ru/post/157619/> (дата обращения: 18.05.2020).
3. Технологии позиционирования в реальном времени. URL: <http://www.rtlsnet.ru/technology/view/4> (дата обращения: 18.05.2020).
4. Как выбрать лучший протокол Bluetooth для вашего приложения. URL: <https://elenergi.ru/kak-vybrat-luchshij-protokol-bluetooth-dlya-vashego-prilozheniya.html> (дата обращения: 18.05.2020).

5. Руководство по Android Studio. URL: <https://developer.android.com/guide> (дата обращения: 18.05.2020).
6. Bluetooth overview. URL: <https://developer.android.com/guide/topics/connectivity/bluetooth> (дата обращения: 18.05.2020).

УДК 004.7

А. В. Михайлова

студент кафедры инфокоммуникационных технологий и систем связи

Т. М. Татарникова – доктор технических наук, доцент – научный руководитель**О ПОСТРОЕНИИ ПСИХОЛОГИЧЕСКОГО ПОРТРЕТА ПО ТЕХНОЛОГИИ TEXT MINING**

Человек является одним из самых популярных объектов познания. Изучить и понять каждого человека – это сложная задача, с которой сталкиваются самые разные специалисты. В науке – это психологи, социологи, в практике – кадровые работники при подборе персонала, руководители при распределении видов работ, в обыденной жизни – молодые люди при выборе профессии и многое другое.

Одним из способов изучения человека – прослеживание эмоциональных изменений и реакций на основе речевого поведения. Речевое поведение человека служит индикатором его общей эрудиции, особенностей интеллекта, мотивации поведения и эмоционального состояния. По нему можно также определить эмоциональную напряженность человека, которая проявляется в выборе слов и стиле построения фраз.

В статье предложен алгоритм обработки естественного языка, построенный на базе нейронной сети, который позволяет определить психологический портрет человека.

Обработка естественного языка

Обработка естественного языка (NLP – Natural Language Processing) – это технология на стыке искусственного интеллекта и компьютерной лингвистики, включающая способы общения посредством речи, текста, виртуального общения и обмена сообщениями [1].

Большая часть обработки посвящена машинному переводу с одного языка на другой. Это самый распространенный сценарий использования NLP.

Другой не менее известный сценарий NLP – классификация текстов. С помощью обработки естественного языка можно классифицировать текст по темам, адресам, переданному настроению и многим другим меткам.

Классификатор текста строится на следующих элементах [2]:

1. Набор тренировочных текстов (data set) – это исходные тексты, на которых происходит обучение модели классификации.

2. Вектор текста – это вектор параметров, характеризующих классы текстов.

3. Метки классов – это predetermined categories (классы), к которым будет отнесен тот или иной текст, поступающий на обработку.

4. Алгоритм машинного обучения – это алгоритм построения модели классификации текстов, в данной работе применяется рекуррентная нейронная сеть (Recurrent neural network, RNN).

5. Прогнозирующая модель – это модель, которая обучается на наборе тренировочных текстов и может выполнять предсказания меток для новых текстов.

Общая схема взаимосвязи элементов классификатора текстов приведена на рис. 1.

Особенности реализации классификатора текстов

Входные данные. При формировании data set использована речь, переведенная в текстовый формат. Data set содержит 4 папки, названные одноименно с метками в соответствии с классификацией психотипов: холерики, меланхолики, сангвиники и флегматики. Такое разделение людей по темпераменту существует в психологии, где каждый тип характеризуется своими чертами поведения, характера, стремлениями и желаниями. Каждая папка data set содержит тексты из Твиттера, речь, переведенную в текстовый формат знакомых людей, а также тексты знаменитых людей, принадлежащих к данному психотипу. Поскольку в психологии доказано влияние типа темперамента на речь и определены конкретные части речи и выражения, которые использует каждый психотип, то с помощью построенной модели можно также проследить за этими закономерностями.

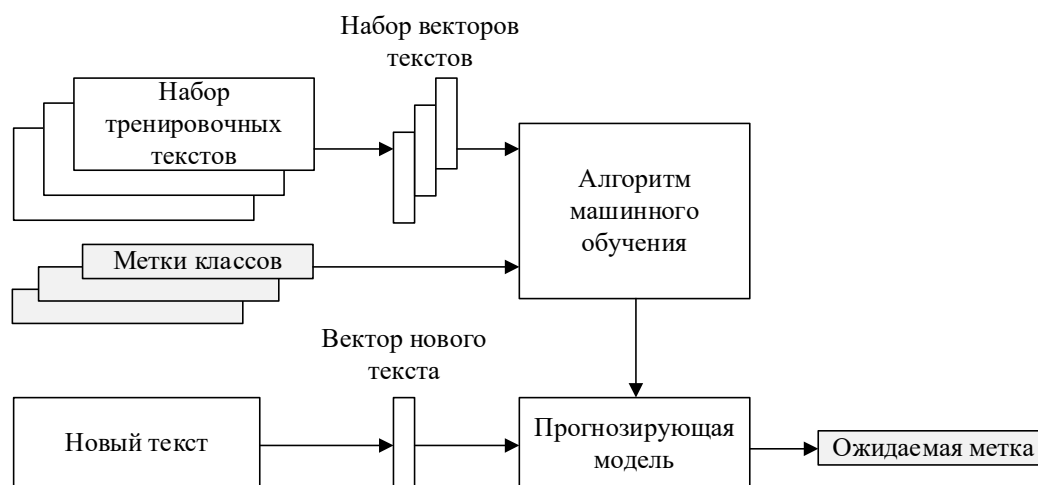


Рис. 1. Схема классификации текста

Формирование вектора текста. Формирование вектора текста происходит в последовательно-сти этапов Text Mining: токенизация по словам, фильтрация от стоп-слов, нормализация, преобразование регистра слов. В целом Text Mining позволяет преобразовать свободный (неструктурированный) текст в нормализованные, структурированные данные, пригодные для анализа в соответствии с алгоритмом машинного обучения [2, 3].

Алгоритм машинного обучения [4, 5]. В работе использована LSTM-модель. LSTM – это особый вид рекуррентной нейронной сети, способной изучать долгосрочные зависимости в данных. Модель принимает на вход векторизацию текста. Часто встречаемые слова образуют словарь, который ограничивается максимальным значением. Длина каждого вектора фиксирована, ее выбор учитывает объем анализируемого текста. Если на вход алгоритма обучения будет подан текст меньшего размера, чем ограничение, то вектор дополняется нулями.

Формирование меток. В соответствие каждому типу темперамента ставится число, которое отвечает за этот тип: 0 – холерик, 1 – сангвиник, 2 – меланхолик, 3 – флегматик.

Прогнозирующая модель. Data set разделен на два множества: обучающее и тестовое в соотношении 80%:20% соответственно.

В модели использована функция активации – нормализованная экспонента, которая предназначена для мультиклассовой классификации, функция потерь – категориальная перекрестная энтропия CCE библиотеки Keras:

$$CCE = -\sum_{i=1}^n (x_i \log(y_i)),$$

где y_i – прогнозируемое значение; x_i – истинное значение; n – размер вектора $x_i(y_i)$.

Модель имеет вид:

```

model = Sequential()
model.add(Embedding(num_words, embed_dim, input_length=num_words))
model.add(Dropout(0.2))
model.add(LSTM(lstm_out, recurrent_dropout=0.2, dropout=0.2))
model.add(Dropout(0.2))
model.add(Dense(4, activation='softmax'))
model.compile(loss = 'categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    
```

где num_words – количество слов в словаре;

embed_dim – длина векторов для представления каждого слова;

lstm_out – количество блоков памяти в LSTM слое;

Dense – количество меток классов.

Полученные результаты

Обучение проводилось 20 эпох, за которые были получены результаты, приведенные на рис. 2, 3.

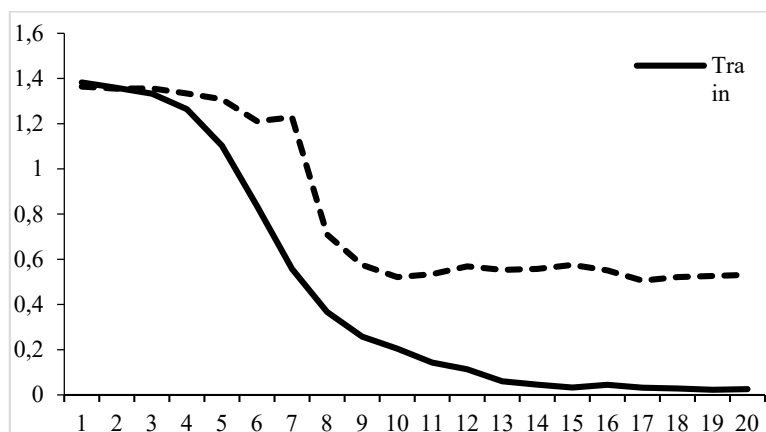


Рис. 2. Изменение функции потерь

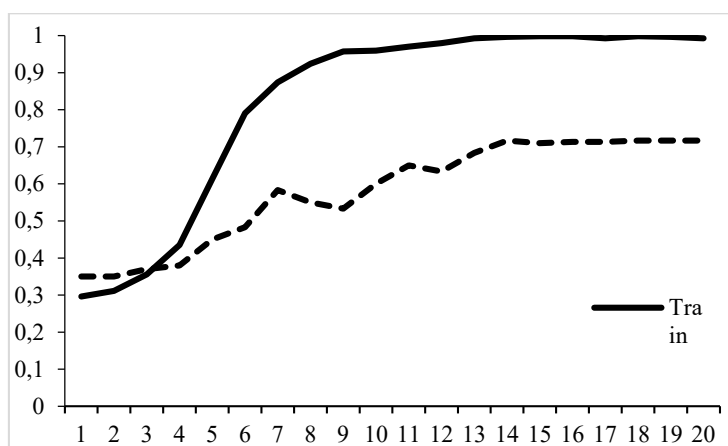


Рис. 3. Изменение функции точности

Заключение

Продемонстрирована возможность применения технологии NLP для извлечения информации о психологическом портрете из текстового описания. Сформированные метки класса были сопоставлены с известными результатами. Точность на данный момент составила более 70%. В перспективе возможно повышение точности, т.к. планируется добавление параметров, увеличение объема data set и другие шаги. Предложенный подход демонстрирует возможность автоматического решения задачи определения психотипа человека при наличии записи его речи в разных жизненных ситуациях.

Библиографический список

1. Большакова Е. И., Клышинский Э. С., Ландэ Д. В. и др. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика. М.: МИЭМ, 2011. 272 с.
2. Советов Б. Я., Татарникова Т. М., Яшин А. И. Use of Technology Textmining for Identifying Hidden Threats in Communications Distributed by Social Networks / Известия СПбГЭТУ ЛЭТИ. 2019. № 4. С. 26–32.
3. Татарникова Т. М. Анализ данных. СПб.: СПбГЭУ, 2018. 85 с.
4. Алпайдин Э. Машинное обучение: новый искусственный интеллект. М.: Альпина Паблшер: Точка, 2017. 208 с.
5. Вьюгин В. Математические основы теории машинного обучения и прогнозирования. МЦМНО, 2013. 390 с.

УДК 004.056.5

П. П. Недошивин

студент кафедры безопасности информационных систем

А. А. Овчинников – кандидат технических наук, доцент – научный руководитель**ИССЛЕДОВАНИЕ КОДОВЫХ ЦИФРОВЫХ ПОДПИСЕЙ**

Для реализации процедур криптографии с открытым ключом необходимо наличие задачи, которая является экспоненциально сложной как массовая задача и имеет частные задачи полиномиальной сложности. Такие задачи должны относиться к классу NP-полных, то есть решаемых на детерминированной машине Тьюринга за экспоненциальное число шагов. В асимметричной криптографии используется несколько типов задач, из них наиболее известны два – теоретико-числовые и кодовые задачи. Теоретико-числовые задачи лежат в основе большинства современных несимметричных криптосистем: задача факторизации числа – в основе системы RSA, задача дискретного логарифмирования – в основе системы Эль-Гамала. В 1970-1980-е гг. появились также системы, использующие кодовые задачи, например, система МакЭлиса и система Нидеррайтера, не снискавшие популярности вследствие большого объема ключевых данных и высокой стоимостью ресурсов памяти в то время. Однако в отличие от многих теоретико-числовых задач задача декодирования алгебраического кода оказалась более стойкой к квантовым атакам.

Одной из важнейших проблем криптографии, начиная с 1980-х гг., является проблема аутентификации информации. Решением стало внедрение электронной цифровой подписи (ЭЦП), которая использует асимметричное шифрование и хеширование данных. Вследствие популярности криптосистем с открытым ключом, использующих теоретико-числовые задачи, большая часть ЭЦП основана на этом классе задач.

Существующие кодовые подписи либо не устойчивы к определенным атакам, либо имеют высокую сложность формирования и большой объем ключевых данных. Поэтому необходимость исследования кодовых ЭЦП сейчас стоит наиболее остро.

Одним из самых известных и популярных до сих пор алгоритмом кодовой подписи является схема, предложенная в 2001 г. Николя Куртуа, Матье Финиашем и Николя Сендре, получившая название по инициалам ее создателей – CFS. Эта схема основана на криптосистеме Нидеррайтера, использует многие ее параметры.

Для решения проблемы небольшой мощности множества декодируемых слов авторами CFS был предложен метод полного декодирования, который повышает корректирующую способность с целью найти ближайшее (по минимуму расстояния Хэмминга) кодовое слово с высокой вероятностью.

Главным принципом подписи CFS является применение к исходному сообщению M хеш-функции, преобразующей его в синдром, который нужно декодировать. Особенность CFS – хеширование надо проводить многократно, при этом увеличивая каждый раз добавленный счетчик, что позволит рано или поздно получить нужный синдром. Стойкость схемы к подлогу подписи основана на сложности решения задачи синдромного декодирования. Секретный ключ, используемый отправителем для определения вектора ошибок, должен помочь декодеру с решением этой задачи для некоторой доли кодовых слов. Получившийся вектор ошибок и последнее значение счетчика (при котором был получен декодируемый синдром) составляют пару подписи и отправляются вместе с сообщением.

Алгоритм. Подпись CFS**Генерация ключей:**

- Выбрать случайную проверочную матрицу \tilde{H} двоичного (n, k) – кода Гоппы, исправляющего t ошибок с алгоритмом декодирования γ .

- Случайно выбрать $(n - k) \times (n - k)$ обратимую матрицу Q над полем \mathbb{F}_q .

- Случайно выбрать $n \times n$ перестановочную матрицу P над полем \mathbb{F}_q .

Открытый ключ: $H = Q\tilde{H}P$ **Закрýтый ключ:** $(P, \tilde{H}, Q, \gamma)$

Подпись сообщения:
Вход: сообщение M

1) $i \leftarrow i + 1$

2) $x' = \gamma(Q^{-1}h(h(m) || i))$

 3) Если x' не обнаружен, вернуться к шагу 1

Выход: подпись $(i, x'P)$
Проверка подписи:

 - Вычислить $s' = Hx'^T$ и $s = h(h(m) || i)$.

 - Подпись действительна, если $s = s'$.

Для генерации ключей не требуется использование диагональной матрицы D , так как применяются двоичные линейные коды.

Алгоритм декодирования должен по принятому синдрому $s = (s_0, s_1, \dots, s_{n-k-1})$ либо принять решение об отказе от декодирования, либо выделить вектор ошибок $e = (e_0, e_1, \dots, e_{n-1})$ и вычислить кодовое слово $c = (c_0, c_1, \dots, c_{n-1})$. Процедура формирования подписи и инкрементация счетчика продолжается до тех пор, пока не наступит событие «успех» на выходе декодера.

При выполнении алгоритма формирования подписи по схеме CFS значение $h(h(m) || i)$ представляется в виде синдромной последовательности $s = (s_0, s_1, \dots, s_{n-k-1})$. Результат хеш-функции также соответствует вектору ошибок $e = (e_0, e_1, \dots, e_{n-1})$, что также представимо в виде $Y = (e, i) : H \cdot e^T = (h(h(m) || i))^T$.

Задача вычисления вектора $e = (e_0, e_1, \dots, e_{n-1})$ по известному вектору $h(h(m) || i)$ сопряжена с решением задачи декодирования (n, k, d) кода. Для уполномоченного пользователя (знающего секретный ключ) – это вычислительно простая задача (полиномиальной сложности). Для злоумышленника (знающего только открытый ключ) – это вычислительно сложная задача декодирования случайного кода (относящаяся к классу сложности NP – полных задач).

При верификации (проверки подлинности подписи $Y(e, i)$ сообщения m) проверяется, является ли результат хеширования $h(h(m) || i)$ синдромной последовательностью, вычисленной по вектору $e = (e_0, e_1, \dots, e_{n-1})$ (который, как указано выше, является вектором ошибок).

В алгоритме CFS ключевым моментом является перебор векторов до получения декодируемого синдрома. Стойкость системы зависит от того, сколько попыток потребуется атакующему для подбора нужного синдрома определенного веса. Был проведен эксперимент по определению среднего количества попыток, которые требуются до получения декодируемого синдрома, и определению спектра вектора ошибок (вероятности появления вектора заданного веса).

Каждое кодовое слово имеет $\sum_{i=0}^t C_n^i$ векторов длины n , в которых можно исправить до t оши-

бок, и получить это кодовое слово. Так как количество кодовых слов 2^k , а векторное пространство имеет 2^n векторов, следовательно, вероятность того, что полученный после передачи по каналу век-

тор удастся декодировать равна $P_y = \frac{2^k \sum_{i=0}^t C_n^i}{2^n} = \frac{\sum_{i=0}^t C_n^i}{2^r}$. Эта величина должна соответствовать гео-

метрическому распределению, в котором рассматриваемая величина равна номеру первого «успеха» или количеству «провалов» до первого «успеха». Математическое ожидание равно $\frac{1}{P_y}$ или $\frac{1-P_y}{P_y}$

соответственно (в моделировании рассматривался второй случай).

Как сказано выше, каждое кодовое слово имеет $\sum_{j=0}^t C_n^j$ декодируемых синдромов. Так как ко-

личество кодовых слов веса i длины n равно C_n^i , то вероятность декодирования вектора конкрет-

$$P_i = \frac{C_n^i}{\sum_{j=0}^t C_n^j}.$$

Было проведено имитационное моделирование, которое должно было доказать истинность приведенных выше формул на практике. В схеме подписи авторы рекомендуют использовать коды Гоппы, но система работает и при использовании кодов БЧХ. Коды Гоппы можно декодировать теми же способами, что и коды БЧХ (при моделировании использовался декодер Берликэмпа-Мэсси). В ходе эксперимента генерировался случайный (равномерное распределение с вероятностью «1», равной 0.5) вектор длины n (в схеме подписи CFS происходит подбор синдрома длины $r = n - k$, но так как множество векторов длины n однозначно отображается во множество синдромов, то это не должно влиять на результат). Этот вектор декодируется; в случае «успеха» – записывается количество исправленных ошибок, в случае «отказа» – повторение генерации и декодирования. Эксперимент проводился при различных параметрах n и t . Полученные графики представлены на рис. 1 – 4.

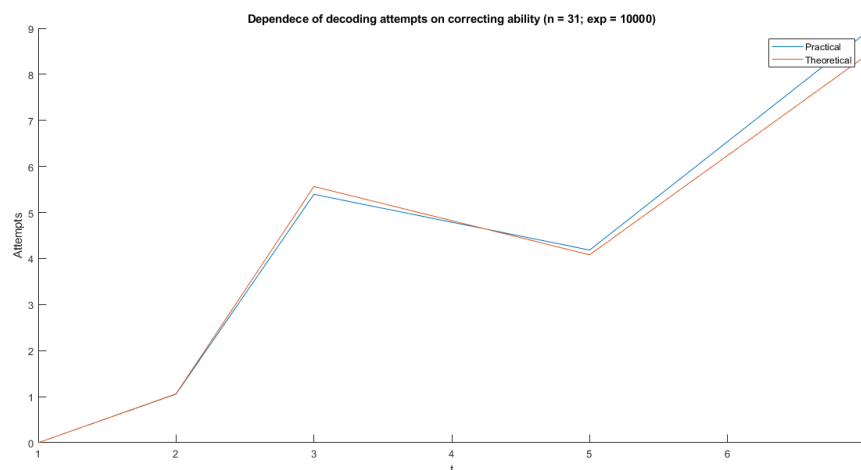


Рис. 1. Среднее количество попыток в зависимости от t ($n = 31$)

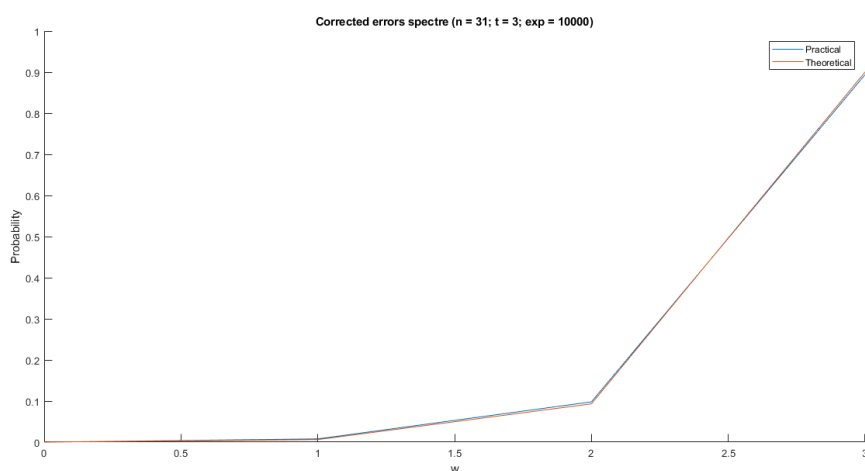


Рис. 2. Спектр векторов ошибок ($n = 31$; $t = 3$)

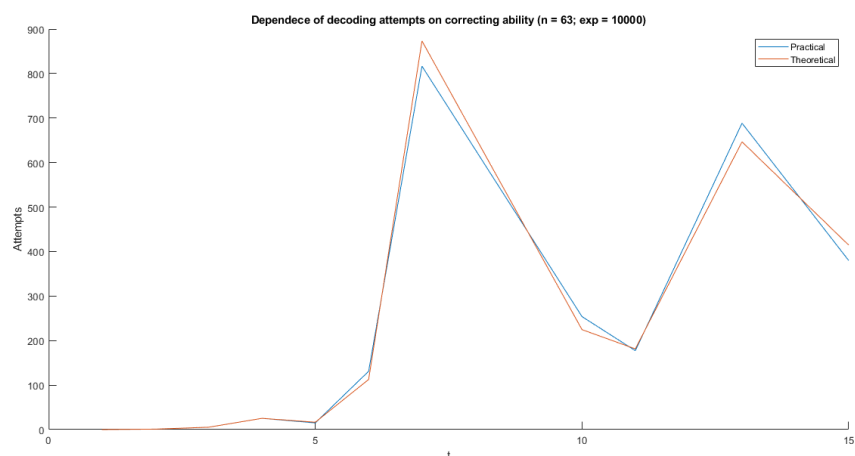


Рис. 3. Среднее количество попыток в зависимости от t ($n = 63$)

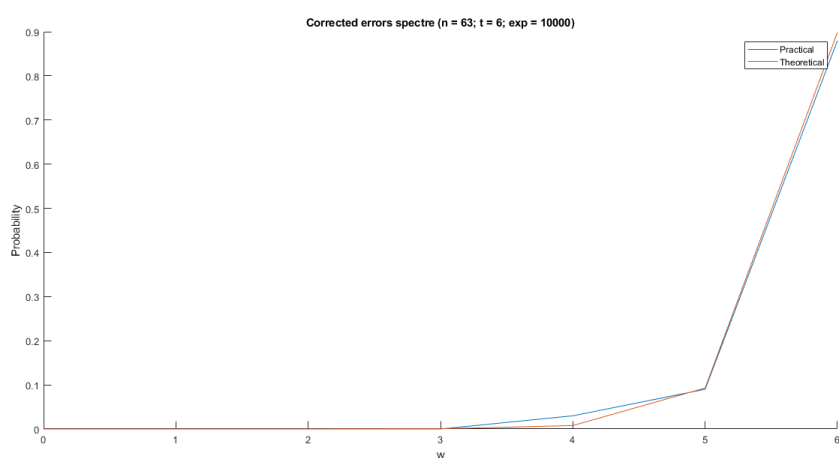


Рис. 4. Спектр векторов ошибок ($n = 63$; $t = 6$)

Заключение

В настоящее время в связи с появлением квантовых алгоритмов стали востребованы системы электронной цифровой подписи, основанные на задаче декодирования линейных кодов. В статье рассматривалась схема CFS, стойкость которой основана на сложности поиска декодируемого синдрома определенного веса. Было проведено имитационное моделирование, доказавшее, что и на практике среднее количество попыток декодирования является случайной величиной, распределенной по геометрическому закону.

Библиографический список

1. *Cayrel P.-L., Mezzani M.* Post-Quantum Cryptography: Code-Based Signatures. 2010.
2. *Courtois N. T., Finiasz M., Sendier N.* How to Achieve a McEliece-Based Signature Scheme. 2011.
3. *Baldi M.* QC-LDPC Code-Based Cryptography. New York: Springer. 2014.
4. *Finiasz M.* Parallel-CFS. Strengthening the CFS McEliece-Based Signature Scheme. 2011.
5. *Крук Е. А.* Комбинаторное декодирование линейных блочных кодов. СПб.: ГУАП, 2007.

УДК 004.738.2

А. Д. Хорошавин

студент кафедры инфокоммуникационных технологий и систем связи

А. М. Тюрликов – доктор технических наук, профессор – научный руководитель

СИСТЕМА ПЕРЕДАЧИ ДАННЫХ О РАСПОЛОЖЕНИИ ОБЪЕКТОВ С НИЗКИМ ЭНЕРГОПОТРЕБЛЕНИЕМ

Введение

Интернет вещей (Internet of things) – концепция, базирующаяся на объединении некоторого количества различных устройств (сенсоров, датчиков и т.п.) с целью создания единой сети этих устройств, в которой они смогут анализировать информацию и обмениваться ею.

На сегодняшний день возможно создать целую инфраструктуру из таких устройств, позволяющую облегчить жизнь человеку.

Для беспроводной передачи данных особо важную роль в построении «интернета вещей» играет такое качество, как энергоэффективность. Поскольку большая часть таких устройств использует в качестве источника питания различные аккумуляторы и батареи, основной интерес здесь представляет технология LoRa.

LoRa (Long Range) – технология, реализующая метод модуляции (первый уровень модели OSI), который обеспечивает большую дальность связи (зону покрытия) по сравнению с альтернативами при той же мощности передатчика. Протокол канального уровня получил название LoRaWAN. Задачей LoRa является быстрая доставка малых сообщений на большие расстояния с минимальными энергозатратами в ущерб скорости передачи. Ширина полосы канала составляет 125 кГц, минимальная скорость передачи по каналу составляет 0.018 кб/с и достигает 50 кб/с, при использовании GFSK-модуляции (Gaussian Frequency-Shift Keying). Дальность передачи в условиях открытой местности – до 15 км, в городской обстановке до 7км. Технология LoRa позволяет осуществлять демодуляцию сигналов с уровнями на 19,5 дБ ниже уровня шумов. [1]

В настоящее время, используя данную технологию связи, становится возможна разработка энергоэффективных устройств. Сенсоры и датчики, пользуясь данной технологией передачи способны экономить значительно количество энергии источника питания, что положительно скажется на времени работы от аккумулятора.

Создание устройства на базе модуля LoRa

Для разработки конечного устройства была получена плата разработчика AcSip S76Gxb, предназначенная для тестирования и отладки LoRa-модуля S76G.

Характеристики модуля S76G:

- ядро ЦП: ARM 32 Cortex M0,
- частота ЦП: 24 МГц,
- флэш-память: 8КБ,
- напряжение питания: 2.1 – 3.6В,
- разъемы отладки: JTAG, GROVE, SWD, UART со встроенным переходом на USB 2.0,
- частота LoRa-модуля: 863 МГц,
- дальность LoRa-модуля: 8 – 15 км в зависимости от местности,
- скорость передачи: до 37,5 Кбит/с.

Модуль S76G с поддержкой LoRa и GPS/ГЛОНАСС оснащён встроенным энергоэффективным процессором 32-bit Cortex M0+ MCU (STM32L073x) и предназначен для применения в беспроводных системах безопасности, датчиках дыма, управлении освещением, умных домах, умных счётчиках и любых других устройствах, где требуется дальний радиус действия, низкое энергопотребление и GPS, ГЛОНАСС позиционирование. [3]

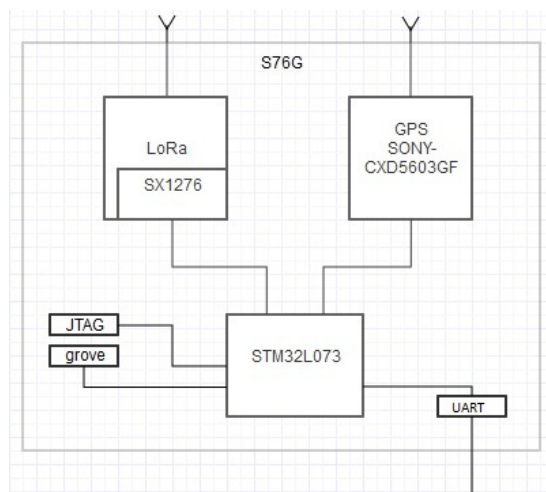


Рис. 1. Функциональная схема устройства S76Gxb

На рис. 1 представлена функциональная схема платы S76Gxb, где STM32L073 – центральный микроконтроллер, SX1276 – модуль LoRa, GPS SONY-CXD5603GF – модуль GPS-приемника.

К недостаткам данного устройства следует отнести закрытое ПО. В случае изменения прошивки восстановление до заводского состояния невозможно. Управление осуществляется командами в терминальном режиме, поступающими с внешних устройств (напр., компьютера). В качестве автономного внешнего управляющего устройства был взят микроконтроллер STM32F103C6T8.

STM32F103C6T8 – микроконтроллер на базе процессора ARM Cortex-M3

Характеристики микроконтроллера STM32F103C8T6:

- ядро ЦП: ARM 32 Cortex-M3,
- частота ЦП: 72 МГц,
- флэш-память: 20 КБ,
- память программ: 64 КБ,
- напряжение питания: 2.1 – 3.6В,
- разъемы отладки: SWD.

Плата характеризуется низким энергопотреблением и малыми размерами. Наличием готовых библиотек в среде Arduino IDE существенно облегчает процесс разработки ПО. [2][5]

На рис. 2 приведена функциональная схема собранного устройства, состоящего из двух плат, подключенных друг к другу по интерфейсу UART.

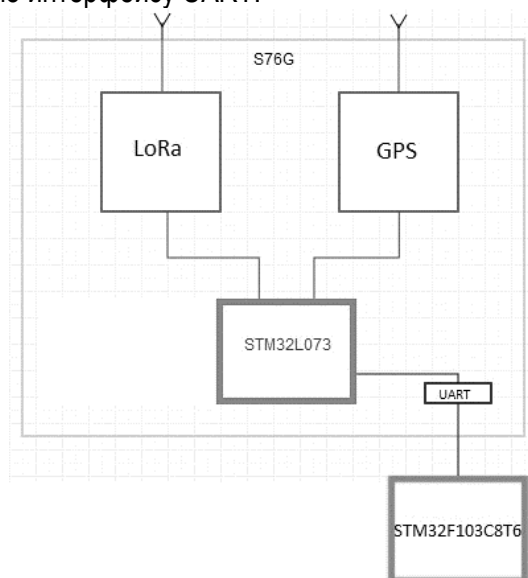


Рис. 2. Функциональная схема конечного устройства

Описание основного алгоритма работы устройства

Исходя из того, что главным требованием к устройству является его энергоэффективность, оно большую часть времени будет находиться в режиме «сна».

1. При первичном запуске устройство инициализирует GPS-модуль, определяет и записывает свои координаты и уходит в режим «сна».

2. По завершении заданного периода T (входной параметр алгоритма) устройство заново инициализирует GPS-модуль, получает координаты и анализирует их.

3. Если координаты не изменились, устройство уходит в режим «сна» до следующего пробуждения.

4. Если координаты изменились, устройство инициализирует модуль LoRa и начинает передавать свои координаты на базовую станцию. Как только от станции придет положительная квитанция, устройство перезаписывает новые координаты вместо старых и переключается в режим «сна» до следующего пробуждения.

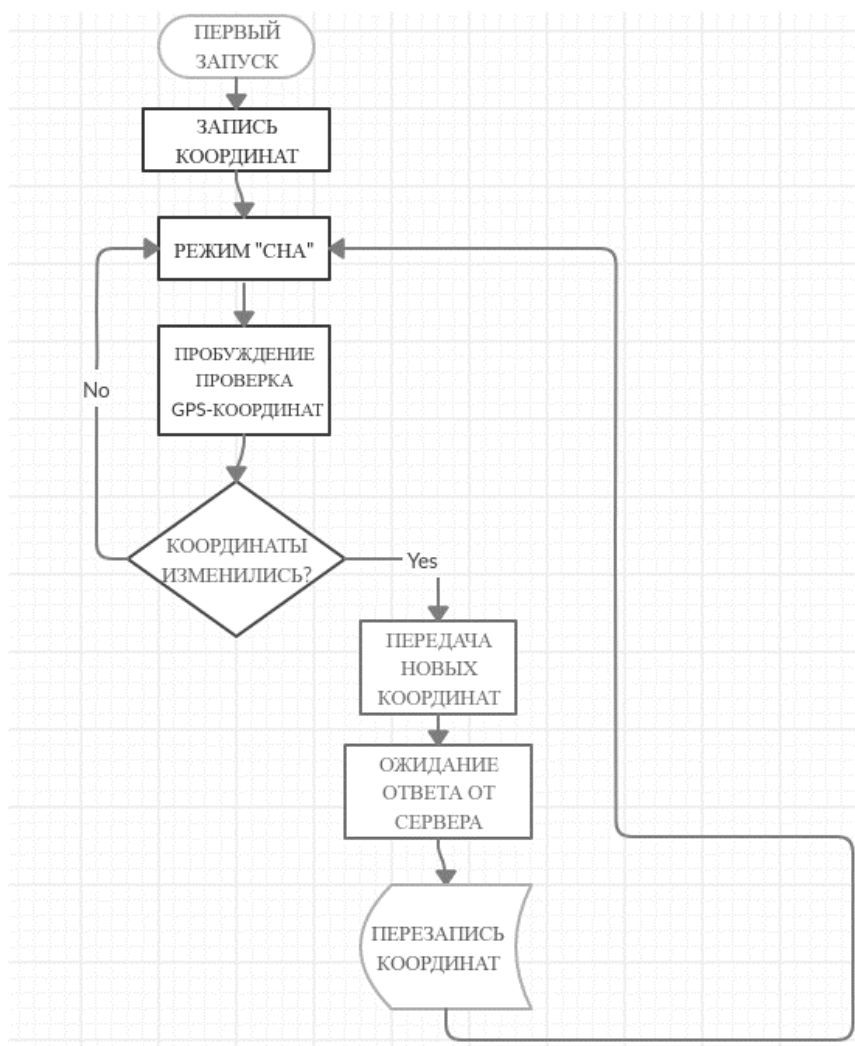


Рис. 3. Блок-схема алгоритма работы устройства

В алгоритме также должны быть заданы:

- *погрешность* – максимально допустимое расстояние, на которое устройство может переместиться и не сообщить об этом;

- *таймер форсированной передачи* – время, через которое устройство будет отсылать свои координаты и уровень заряда батареи вне зависимости от результатов анализа местоположения.

Заключение

Разработана функциональная схема устройства. Собрано конечное устройство на базе модуля LoRa. Разработан и реализован алгоритм приема, анализа и отправки GPS координат с использованием стека протоколов LoRa. В алгоритме имеется возможность изменения параметров пробуждения, благодаря этому устройство можно настроить для работы в различных режимах.

Библиографический список

1. Wireless-e. Технология LoRa в вопросах и ответах URL: <https://wireless-e.ru/standarty/tehnologiya-lora-v-voprosah-i-otvetah> (дата обращения: 26.04.2020).
2. HWman. Дешевая STM32 плата + Arduino IDE. URL: <https://habr.com/ru/post/395577/> (дата обращения: 01.03.2020).
3. Euromobile. Плата разработчика S76Gxb. URL: <https://www.euromobile.ru/produkcija/lora-moduli/s76g> (дата обращения: 21.11.2019).
4. ChipDip. Плата разработчика STM32F103C8T6. URL: <https://www.chipdip.ru/product/stm32f103c8t6> (дата обращения: 25.04.2020).

СОДЕРЖАНИЕ

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ	3
Проблемно-ориентированные вычислительные комплексы	3
<i>Алирзаев Р. З., Шевяков А. А.</i> АЛГОРИТМЫ ШИФРОВАНИЯ ХРАНИМЫХ ДАННЫХ В MYSQL	3
<i>Балезин Л. Н., Берёза В. В.</i> УЧЕБНЫЙ КОМПЛЕКС НА ОСНОВЕ СЕТИ МИКРОКОМПЬЮТЕРОВ RASPBERRY PI	6
<i>Балыков А. Г.</i> ОСНОВНЫЕ МЕХАНИЗМЫ ИНТЕГРАЦИИ 1С С ДРУГИМИ СИСТЕМАМИ	11
<i>Борисов Д. А.</i> ОСОБЕННОСТИ И ХАРАКТЕРИСТИКИ ТВЕРДОТЕЛЬНЫХ УСТРОЙСТВ ФОРМАТА M.2.....	14
<i>Велюга В. А.</i> ИСПОЛЬЗОВАНИЕ ИНСТРУМЕНТОВ DEEPSPECTRUM И AUDEEP ДЛЯ АНАЛИЗА РЕЧИ	18
<i>Ветошев А. А., Кукушкина Н. С.</i> ОСОБЕННОСТИ РЕАЛИЗАЦИИ ETL-ПРОЦЕССА ПРИ КОНСОЛИДАЦИИ ДАННЫХ В СИСТЕМЕ ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ	22
<i>Вихров В. В.</i> ДЕСКРИПТИВНЫЙ АНАЛИЗ ФАКТОРОВ, ВЛИЯЮЩИХ НА СТОИМОСТЬ НЕДВИЖИМОСТИ	25
<i>Герасимов С. М.</i> ОПРЕДЕЛЕНИЕ ДЕФЕКТА АКУСТИЧЕСКИМ МЕТОДОМ.....	31
<i>Герасимов С. М.</i> ИССЛЕДОВАНИЕ МЕТОДОВ АНАЛИЗА РЕЧЕВЫХ СИГНАЛОВ.....	36
<i>Давидович Б. В., Хасанов Б. Р.</i> ПОВЫШЕНИЕ УРОВНЯ ЗАЩИТЫ БЕСПРОВОДНОЙ ПЕРЕДАЧИ ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ БИОМЕТРИЧЕСКИХ ДАННЫХ	42
<i>Заболотская А. А.</i> СИНТЕЗ УПРУГОЙ ЭЛЕКТРОМЕХАНИЧЕСКОЙ СИСТЕМЫ С НЕПОЛНЫМ МОДАЛЬНЫМ УПРАВЛЕНИЕМ	47
<i>Зорина Е. Д.</i> АНАЛИЗ ПРОГРАММНЫХ ШАБЛОНОВ РЕАЛИЗАЦИИ ИНТЕРАКТИВНЫХ ВЕБ-ПРИЛОЖЕНИЙ.....	51
<i>Клычкова Е. А.</i> АНАЛИЗ МНОГОКОМПОНЕНТНОЙ ТРАНСПОРТНОЙ ЗАДАЧИ С ОГРАНИЧЕНИЯМИ НА ПРОПУСКНУЮ СПОСОБНОСТЬ	57
<i>Кондратьев И. Р.</i> ВОПРОСЫ ОБРАБОТКИ ИЗОБРАЖЕНИЙ В MATLAB	61
<i>Корепин П. А.</i> УСТРОЙСТВО ОПРЕДЕЛЕНИЯ ПАДЕНИЯ ЧЕЛОВЕКА.....	68
<i>Крылова В. Е.</i> ПРИМЕНЕНИЕ АЛГОРИТМОВ НЕЧЕТКОГО ПОИСКА ДЛЯ УСТРАНЕНИЯ ИНФОРМАЦИОННОЙ НЕОПРЕДЕЛЕННОСТИ	71
<i>Лукашова Е. О., Ненашев С. А.</i> ИССЛЕДОВАНИЕ РЕЖИМОВ ОПТИКО-ЭЛЕКТРОННОГО МОНИТОРИНГА, РЕАЛИЗУЕМЫХ НА БАЗЕ МАЛЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ	74
<i>Маханов И. А.</i> ПЕРСПЕКТИВЫ ВНЕДРЕНИЯ ТЕХНОЛОГИИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ В ПОВСЕДНЕВНУЮ ЖИЗНЬ	81
<i>Михайлов В. К.</i> ПРИМЕНЕНИЕ МНОГОПОТОЧНОГО ПРОГРАММНОГО РАСШИРЕНИЯ DOTS В МЕЖПЛАТФОРМЕННОЙ СРЕДЕ РАЗРАБОТКИ КОМПЬЮТЕРНЫХ ИГР UNITY	83
<i>Мозырева К. А.</i> ПРИМЕНЕНИЕ ВЕТВЯЩИХСЯ ПРОЦЕССОВ	88
<i>Орлянская В. А.</i> ТЕХНОЛОГИИ РАЗРАБОТКИ МАРКЕТПЛЕЙСОВ	91
<i>Рогозин П. А., Бульба В. В.</i> ИССЛЕДОВАНИЯ ВАРИАНТОВ РЕАЛИЗАЦИИ АВТОМАТИЧЕСКОЙ СИСТЕМЫ ОРОШЕНИЯ РАСТЕНИЙ	96
<i>Русанов М. В.</i> ПРОБЛЕМЫ ИСПОЛЬЗОВАНИЯ ЧИСЛЕННЫХ МЕТОДОВ ИНТЕГРИРОВАНИЯ В СРЕДЕ MATHCAD	100
<i>Савчук Я. А.</i> СБОР И ПРЕДОБРАБОТКА ДАННЫХ ДЛЯ ПОСТРОЕНИЯ МОДЕЛИ ЦЕНООБРАЗОВАНИЯ НА АВТОМОБИЛЬНОМ РЫНКЕ	104
<i>Сердечная Е. А.</i> ОБЕСПЕЧЕНИЕ РОБАСТНОСТИ СИСТЕМ С МОДАЛЬНЫМ УПРАВЛЕНИЕМ ..	113

Соколова К. Д., Груздев Д. А. СОВРЕМЕННЫЕ ПАССИВНЫЕ АКУСТОЭЛЕКТРОННЫЕ ДАТЧИКИ ФИЗИЧЕСКИХ ВЕЛИЧИН НА ПАВ	117
Судьин М. И. НЕСТАНДАРТНЫЙ МЕТОД ОЦЕНКИ ЧАСТОТЫ СЕРДЕЧНЫХ СОКРАЩЕНИЙ И ЧАСТОТЫ ДЫХАНИЯ.....	122
Таволжанский А. В. ПОСТРОЕНИЕ МОДАЛЬНЫХ РЕГУЛЯТОРОВ С УЧЕТОМ ТРЕБОВАНИЙ ФИЗИЧЕСКОЙ РЕАЛИЗАЦИИ	124
Фиргер А. Е. СИНТЕЗ ИНВАРИАНТНОЙ СИСТЕМЫ РЕГУЛИРОВАНИЯ С НАБЛЮДАТЕЛЕМ ВОЗМУЩЕНИЙ	128
Чекменева А. А. ИССЛЕДОВАНИЕ СТРУКТУРЫ ТЕХНИЧЕСКИХ СРЕДСТВ ДЛЯ РЕАЛИЗАЦИИ МЕТОДИК ЛЕЧЕНИЯ РАН ОТРИЦАТЕЛЬНЫМ ДАВЛЕНИЕМ.....	132
Чекменева А. А. АЛГОРИТМ РАБОТЫ УСТРОЙСТВА ДЛЯ ЛЕЧЕНИЯ РАН ОТРИЦАТЕЛЬНЫМ ДАВЛЕНИЕМ И СРЕДСТВА ЕГО РЕАЛИЗАЦИИ.....	136
Шабалин А. А. ПРОГНОЗИРОВАНИЕ КУРСА КРИПТОВАЛЮТЫ МЕТОДАМИ МАШИННОГО ОБУЧЕНИЯ.....	141
Шемякин А. А. ПРИМЕНЕНИЕ МЕТОДОВ ВИЗУАЛИЗАЦИИ МНОГОМЕРНЫХ ДАННЫХ В МЕДИЦИНСКОЙ СФЕРЕ	146
Шубырева Л. А. АЛГОРИТМЫ СОСТАВЛЕНИЯ КОМПЛЕМЕНТАРНЫХ ПАР ДЛЯ КЛИНИЧЕСКОГО ИССЛЕДОВАНИЯ	151
Яковлев А. А. АЛГОРИТМ МОДЕЛИРОВАНИЯ ОТРАЖЕНИЙ ОТ КРОМКИ «ЗЕМЛЯ-МОРЕ»	155
Компьютерные технологии и программная инженерия	159
Борисов А. К. АВТОМАТИЧЕСКОЕ АННОТИРОВАНИЕ ИЗОБРАЖЕНИЙ НА РУССКОМ ЯЗЫКЕ С ИСПОЛЬЗОВАНИЕМ НЕЙРОСЕТЕВОЙ МОДЕЛИ.....	159
Васильева А. В. МОДЕЛИРОВАНИЕ И ПРОГНОЗИРОВАНИЕ СТОХАСТИЧЕСКИХ НЕЛИНЕЙНЫХ МНОГОМЕРНЫХ ВРЕМЕННЫХ РЯДОВ НА ОСНОВЕ МОДЕЛИ ИНТЕГРИРОВАННОЙ АВТОРЕГРЕССИИ И СКОЛЬЗЯЩЕГО СРЕДНЕГО.....	165
Гандрабура А. А. ВЫДЕЛЕНИЕ ГОВОРЯЩИХ В АУДИОПОТОКЕ.....	170
Глушкова Е. М. ПРИМЕНЕНИЕ ЛАТЕНТНОГО РАЗМЕЩЕНИЯ ДИРИХЛЕ ДЛЯ ОПРЕДЕЛЕНИЯ МЕРЫ СХОЖЕСТИ ТЕКСТОВ.....	175
Демчук М. П. РАЗРАБОТКА НОВОГО ФОРМАТА СЖАТИЯ ИЗОБРАЖЕНИЙ НА ОСНОВЕ МАТРИЦ МЕРСЕННА-УОЛША	178
Кириллова А. А. МАЙНИНГ АРГУМЕНТАЦИИ	182
Малеев В. В. АЛГОРИТМЫ РАСПОЗНАВАНИЯ СОСТОЯНИЙ ДИНАМИЧЕСКОЙ СИСТЕМЫ ПО ВРЕМЕННЫМ РЯДАМ.....	185
Реутов Н. Н. РАСПОЗНАВАНИЕ РУССКОЙ РЕЧИ ПРИ ПОМОЩИ НЕЙРОСЕТЕВОЙ МОДЕЛИ .	188
Ряполова И. В. О ПОСТАНОВКЕ ЗАДАЧИ СОЗДАНИЯ ПРОГРАММЫ ДЛЯ ОПРЕДЕЛЕНИЯ ИНДИВИДУАЛЬНОЙ ОСОБЕННОСТИ ГОЛОСОВОГО АППАРАТА И ВЫРАБОТКИ РЕКОМЕНДАЦИЙ ПО ЕГО ЦЕЛЕСООБРАЗНОЙ НАСТРОЙКИ	191
Семенухин Е. Л. НОВАЯ СИСТЕМА УПРАВЛЕНИЯ ЭЛЕКТРОГИДРАВЛИЧЕСКИМ ДИСКОВЫМ ТОРМОЗОМ «SELF-ENERGISING ELECTRO-HYDRAULIC BRAKE» (SENB) А ПРИНЦИПАХ СИНЕРГЕТИЧЕСКОЙ ТЕОРИИ УПРАВЛЕНИЯ	194
Сергеев Н. Н. АВТОМАТИЗИРОВАННОЕ ОЦЕНИВАНИЕ КВАЛИФИКАЦИОННЫХ НАВЫКОВ И ФОРМАЛИЗАЦИЯ ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ РАБОТНИКА	199
Соболь Д. В. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ В ЭКОЛОГИИ	205
Щеголева А. А. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ БИОЛОГИЧЕСКОЙ СИСТЕМЫ «ХИЩНИК-ЖЕРТВА С ПИТАНИЕМ».....	210

Юрьева В. С. СИНЕРГЕТИЧЕСКИЙ АЛГОРИТМ ОПТИМИЗАЦИИ В ПРОСТЕЙШЕЙ БАЛАНСОВОЙ МОДЕЛИ ЗАПАСОВ В УСЛОВИЯХ СЛУЧАЙНЫХ ПОМЕХ.....	215
Вычислительные системы и сети.....	219
Антипова А. А. ИНТЕРАКТИВНАЯ 3D-МОДЕЛЬ ЭКСПОНАТА ИЗ АРХЕОЛОГИЧЕСКОЙ КОЛЛЕКЦИИ.....	219
Антонов А. А. ИССЛЕДОВАНИЕ ГРАФИКОВ АМПЛИТУД ВЕТВЕЙ АТТРАКТОРОВ РИМАНА....	225
Глотова О. С. БАЗОВЫЕ СРЕДСТВА ПАРАЛЛЕЛЬНОГО ПРОГРАММИРОВАНИЯ ОС «ЭЛЬБРУС» ДЛЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С МИКРОПРОЦЕССОРОМ «ЭЛЬБРУС-4С»....	228
Доминов Е. Е. НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ БИБЛИОТЕКИ ОБЪЕКТНО-РЕЛЯЦИОННОГО ОТОБРАЖЕНИЯ МОДИФИЦИРОВАННЫМ МЕТОДОМ TRC-C.....	235
Доминов Е. Е. СБОРКА ИНТЕРАКТИВНОЙ СРЕДЫ РАЗРАБОТКИ ДЛЯ ЯЗЫКА KOTLIN НА ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ С АРХИТЕКТУРОЙ ЭЛЬБРУС	242
Ёлкин А. Д. ВЫБОР АРХИТЕКТУРЫ СЕТЕВОГО РЕПОЗИТОРИЯ ПАКЕТОВ ОС «ЭЛЬБРУС» ...	245
Житников И. А. ПЕРЕНОС СИСТЕМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ MOODLE НА КЛАСТЕР ВЫСОКОЙ ДОСТУПНОСТИ НА БАЗЕ СЕРВЕРОВ «ЭЛЬБРУС».....	249
Житников И. А. УСТАНОВКА ОПЕРАЦИОННЫХ СИСТЕМ «ЭЛЬБРУС» И ALTLINUX НА КОМПЬЮТЕРЫ С ПРОЦЕССОРАМИ ЭЛЬБРУС-4С	252
Мустафин Т. Р. О СПОСОБАХ СКАНИРОВАНИЯ ПРЕДМЕТОВ ДЛЯ ПОЛУЧЕНИЯ 3D-МОДЕЛЕЙ	258
Нагорных М. Э., Антонов А. А., Быков А. Н. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ УПРАВЛЕНИЯ ПОЛЬЗОВАТЕЛЬСКИМ ИНТЕРФЕЙСОМ КОМПЬЮТЕРА И ПРИЛОЖЕНИЙ С ПОМОЩЬЮ ЖЕСТОВ КИСТЕЙ РУК	263
Ненашев С. А., Лукашова Е. О. СПОСОБ ОПЕРАТИВНОГО МОНИТОРИНГА С ПОМОЩЬЮ ДВУХПОЗИЦИОННОЙ ОПТИЧЕСКОЙ СИСТЕМЫ МАШИННОГО ЗРЕНИЯ	267
Поляков Д. С. ИНТЕРАКТИВНАЯ 3D-РЕКОНСТРУКЦИЯ СРЕДНЕВЕКОВОЙ КРЕПОСТИ С ИСПОЛЬЗОВАНИЕМ СПЕЦЭФФЕКТОВ.....	276
Халиков Р. С., Прокопьева С. Л. ПРОГРАММА, ФОРМИРУЮЩАЯ УЧЕБНОЕ РАСПИСАНИЕ ДЛЯ СТУДЕНЧЕСКИХ ГРУПП	282
ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ЗАЩИТА ИНФОРМАЦИИ	285
Ахраменко А. Д. СИСТЕМА СБОРА ДАННЫХ ОБ ОЦЕНКЕ СВОЕГО СОСТОЯНИЯ ЗДОРОВЬЯ ПАЦИЕНТАМИ.....	285
Гордеева Е. И., Егорова М. Р., Суворова Е. С. РАЗРАБОТКА ТАБЛО ДЛЯ ЭЛЕКТРОННОЙ ОЧЕРЕДИ SMART SCREEN.....	288
Дранощук А. Д. ИССЛЕДОВАНИЕ ВЛИЯНИЯ ПРЕДОБУЧЕНИЯ СВЕРТОЧНЫХ СЕТЕЙ В ЗАДАЧЕ ПОИСКА ЧЕЛОВЕКА НА НЕСКОЛЬКИХ КАМЕРАХ.....	295
Зыков Д. А. УНИВЕРСАЛЬНЫЙ СТЕНД ДЛЯ ТЕСТИРОВАНИЯ ПРОИЗВОДИТЕЛЬНОСТИ СЕРВИСОВ.....	299
Камышников К. Д., Бернанс П. Е. РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ПОЗИЦИОНИРОВАНИЯ УСТРОЙСТВА НА ОСНОВЕ ТЕХНОЛОГИИ BLUETOOTH.....	303
Михайлова А. В. О ПОСТРОЕНИИ ПСИХОЛОГИЧЕСКОГО ПОРТРЕТА ПО ТЕХНОЛОГИИ TEXT MINING	313
Недошивин П. П. ИССЛЕДОВАНИЕ КОДОВЫХ ЦИФРОВЫХ ПОДПИСЕЙ	316
Хорошавин А. Д. СИСТЕМА ПЕРЕДАЧИ ДАННЫХ О РАСПОЛОЖЕНИИ ОБЪЕКТОВ С НИЗКИМ ЭНЕРГОПОТРЕБЛЕНИЕМ	320

Научное издание

СЕМЬДЕСЯТ ТРЕТЬЯ
МЕЖДУНАРОДНАЯ СТУДЕНЧЕСКАЯ
НАУЧНАЯ КОНФЕРЕНЦИЯ ГУАП
(13–17 апреля 2020 г.)

Часть 3

ТЕХНИЧЕСКИЕ НАУКИ

Сборник докладов

ISBN: 978-5-8088-1507-0



9 785808 815070

Ответственный за выпуск:

А. В. Бобович

Редакторская подготовка *В. А. Черниковой*
Компьютерная верстка *А. Н. Колешко*

Подписано к печати 21.09.20. Формат 60x84 1/16.
Усл. печ. л. 20,4. Уч.-изд. л. 38,0. Тираж 150 экз. Заказ № 304.

Редакционно-издательский центр ГУАП
190000, Санкт-Петербург, ул. Б. Морская, д. 67