

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

---

Всемирному дню авиации и космонавтики  
и 100-летию гражданской авиации России посвящается

СЕМЬДЕСЯТ ШЕСТАЯ  
МЕЖДУНАРОДНАЯ СТУДЕНЧЕСКАЯ  
НАУЧНАЯ КОНФЕРЕНЦИЯ ГУАП  
(17–21 апреля 2023 г.)

Часть 3

ТЕХНИЧЕСКИЕ НАУКИ

Сборник докладов



Санкт-Петербург  
2023

УДК 001  
ББК 72  
С30

С30 Семьдесят шестая Международная студенческая научная конференция ГУАП (17–21 апреля 2023 г.): сб. докл.: в 4 ч. Ч. 3: Технические науки. – СПб.: ГУАП, 2023. – 412 с.

ISBN 978-5-8088-1858-3 (Ч. 3)

ISBN 978-5-8088-1855-2

Представлены доклады Семьдесят шестой Международной студенческой научной конференции ГУАП, прошедшей в апреле 2023 г.

Научные работы студентов ГУАП посвящены актуальным проблемам авиационного и аэрокосмического приборостроения, киберфизических систем, систем управления, робототехники, электроэнергетики, проблемам развития новых информационных технологий, радиотехники, электроники и связи, а также 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

*А. С. Будагов*, доктор экономических наук, доцент

*Д. В. Ворновских*, кандидат юридических наук, директор Ивангородского филиала ГУАП

*К. В. Лосев*, доктор экономических наук, профессор

*Н. Н. Майоров*, доктор технических наук, доцент

*М. О. Самоловов*, директор РИЦ

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

*Т. М. Татарникова*, доктор технических наук, профессор

*Е. А. Фролова*, доктор технических наук, доцент

*С. Л. Поляков*, кандидат технических наук

Ответственный секретарь редакционной коллегии *О. Ю. Багиева*

ISBN 978-5-8088-1858-3 (Ч. 3)

ISBN 978-5-8088-1855-2

© Санкт-Петербургский государственный  
университет аэрокосмического  
приборостроения, 2023

# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММИРОВАНИЕ

Прикладная информатика

УДК 004

**С. П. Барболин**

студент кафедры прикладной информатики

**А. С. Автушко**

студент кафедры прикладной информатики

**А. Н. Бубнов**

студент кафедры прикладной информатики

**К. С. Жаданова**

студент кафедры прикладной информатики

**К. А. Порядина**

студент кафедры прикладной информатики

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

## РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ КОММУНИКАЦИИ УЧАСТНИКОВ РЫНКА УСЛУГ В СФЕРЕ ОРГАНИЗАЦИИ МЕРОПРИЯТИЙ

### Введение

В наше время организация event-мероприятий становится все более популярной и востребованной. Количество компаний, специализирующихся на проведении event-услуг, растет с каждым годом [1, 2]. Однако для использования таких услуг часто необходимо общаться с множеством поставщиков, искать информацию о доступных услугах, выбирать подходящие даты и время и многое другое, что может быть трудоемким и занимать много времени. Поэтому нужна автоматизированная система, которая помогает клиентам быстро и удобно организовывать event-мероприятия.

### Описание системы

Разработано web-приложение, которое помогает клиентам находить и бронировать event-услуги. Ключевой функцией web-приложения для поиска event-услуг является возможность быстрого и удобного поиска и заказа услуг, связанных с организацией event-мероприятий. Приложение предоставляет возможность поиска по категориям и критериям, таким как тип мероприятия, дата, местоположение.

### Проектирование приложения

Первым шагом, необходимым для успешного выполнения поставленных задач, является проектирование информационной системы. На рис. 1 представлена блок-схема предлагаемой системы.

I. Серверная часть.

Указана информация о базе данных, построенной на платформе Firebase. Структура базы заложена на следующих концепциях:

1. Компонент Authorization – отражает блок, содержащий таблицу с первичной информацией о пользователях.
2. Cloud storage – это хранилище файлов с индексацией данных.
3. База данных No SQL содержит таблицы, абстрагированные от SQL-запросов, реализующие получение данных на более высоком уровне.

4. API – это компонент, который обеспечивает создание, редактирование и получение данных из No SQL DB, а также предоставляет ответы на соответствующие запросы на сервере.
5. Хранилище Audio – это хранилище для аудиофайлов.

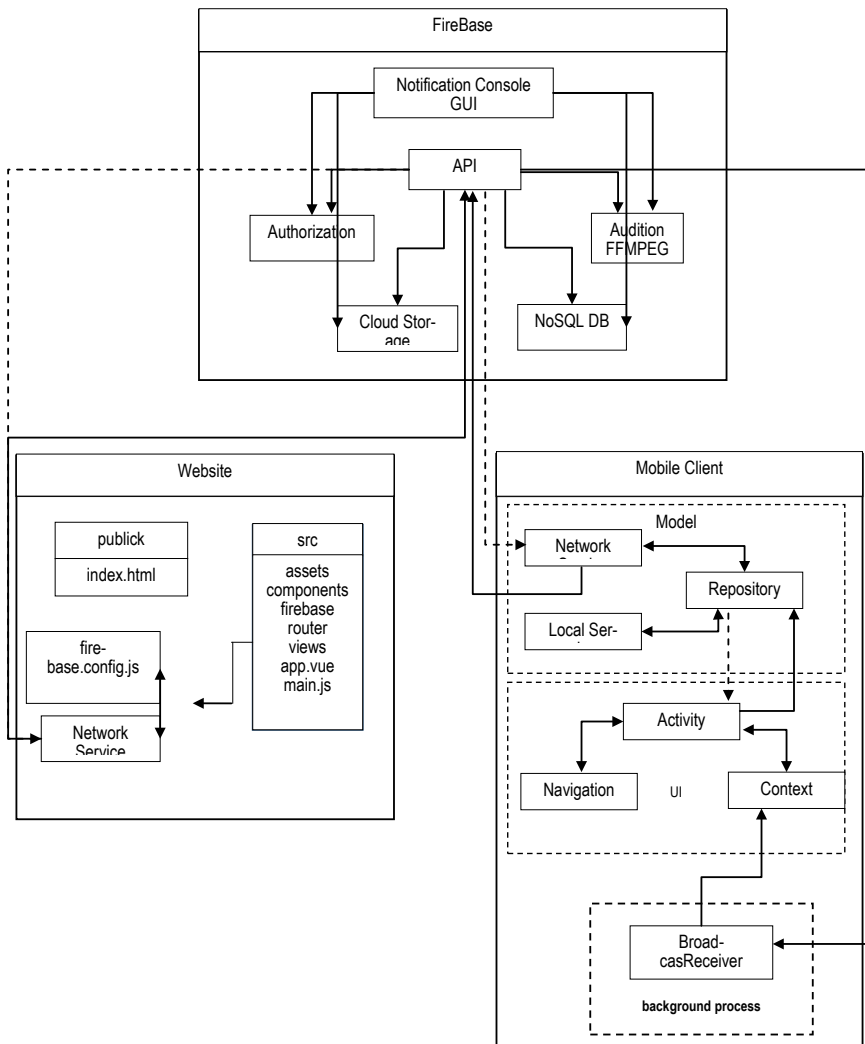


Рис. 1. Блок-схема предлагаемой event-системы

База данных No SQL содержит таблицы: Регионы, Города, Услуги, Категории, Пользователи.

Данная структура обеспечивает хранение и управление данными, указанными в каждой из таблиц базы данных Firebase, и позволяет облегчить процесс работы с данными, улучшить эффективность и надежность их обработки.



### II. Клиентская часть/

На этапе проектирования приложения необходимо определить структуру и способ взаимодействия компонентов, чтобы достичь высокой степени модульности, гибкости и масштабируемости кода. В данном случае, для построения архитектуры приложения рекомендуется использовать паттерн проектирования MVVM.

Паттерн MVVM состоит из трех компонентов: Model, View и ViewModel.

Модель (Model) – это компонент, отвечающий за данные и бизнес-логику приложения. В этом компоненте определяются методы для получения и обработки данных, а также логика работы с базами данных или сетевыми источниками данных.

Представление (View) – это компонент, отвечающий за отображение пользовательского интерфейса. Он отвечает за взаимодействие с пользователем и обновление интерфейса при изменении данных, полученных из модели.

Модель представления (ViewModel) – это компонент, который управляет передачей данных между моделью и представлением. ViewModel содержит бизнес-логику приложения и предоставляет инструменты для управления данными, обновления представления и связи с архитектурой модели данных.

### Стек технологий

После достижения полного понимания данной информационной системы, ее компонентов и подразделов необходимо перейти к ее реализации. Для решения данной задачи был выбран следующий стек технологий.

### III. Android.

Jetpack Compose – это новый инструмент для создания пользовательского интерфейса (UI) для приложений Android. Firebase – это облачная платформа Google, предоставляющая различные решения для разработки мобильных приложений. Для реализации приложения на Firebase с Jetpack Compose можно использовать следующие технологии:

1. Kotlin: язык программирования, на котором можно писать приложение для Android.
2. Jetpack Compose: инструмент для создания пользовательского интерфейса на Kotlin.
3. Android Studio: IDE для разработки приложений для Android.
4. Firebase: облачная платформа для создания и развертывания приложений.
5. Firebase Authentication: сервис для аутентификации пользователей.
6. Firebase Realtime Database: база данных в режиме реального времени, в которой можно хранить и обмениваться данными.
7. Firebase Cloud Messaging: сервис для отправки уведомлений на мобильное устройство.
8. Firebase Crashlytics: инструмент для мониторинга ошибок в приложении и отчетов об ошибках.
9. Firebase Analytics: инструмент для сбора данных о поведении пользователей и мониторинга эффективности приложения.
10. Retrofit: библиотека для работы с сетевыми запросами и API.

Это только некоторые из технологий, которые можно использовать для создания приложения на Firebase с Jetpack Compose. Но, конкретный стек технологий может отличаться в зависимости от требований проекта и предпочтений разработчика.

### IV. Реализация web-клиента.

Vue.js – это JavaScript-фреймворк для разработки пользовательских интерфейсов. Чтобы создавать сайты на Vue.js, можно использовать следующие технологии:

1. HTML и CSS: для создания структуры и стиля веб-страницы.
2. JavaScript: язык программирования, на котором написан Vue.js.
3. Vue.js: сам фреймворк.
4. Vue CLI: инструмент командной строки для быстрого старта проектов на Vue.js и управления конфигурацией.
5. Vue DevTools: расширение для браузера, которое облегчает отладку приложений на Vue.js.
6. Vuex: библиотека для управления состоянием приложения во время его работы.
7. Vue Router: библиотека для управления маршрутизацией веб-приложения.

8. Axios: библиотека для работы с AJAX запросами и API.

9. Webpack: модульный сборщик для Vue.js, который упаковывает и сжимает файлы JavaScript, CSS и другие ресурсы.

10. Babel: транспилятор, который преобразует новый синтаксис JavaScript в старый, поддерживаемый старыми браузерами.

### **Вывод**

В результате проведенной работы на текущий момент реализован функционал просмотра, создания, поиск услуг на сайте и отправка заявок на предложение о работе.

Внедрение IT-технологий – очень важный шаг в любой сфере деятельности. Сфера event-организации расширяется с каждым днем, появляется все больше проблем и все больше различных решений, и многие решения связаны с web-разработкой различных приложений, доказывая, как важна разработка web-приложений для event-рынка и event-сферы [3]. Web-приложение для организации и бронирования event-услуг помогает пользователям находить, бронировать и оплачивать услуги в одном месте. Проектируемая event-система позволит быстро и эффективно организовать публичное мероприятие.

### **Библиографический список**

1. Организация мероприятий: на что обратить внимание? URL: <https://www.eventnn.ru> (дата обращения: 10.05.2023).

2. Проблемы и тенденции развития отрасли Event&MICE в России. URL: <https://adindex.ru/publication/analytics/search/2017/08/24/161917.phtml> (дата обращения: 10.05.2023).

3. 10 полезных приложений для event-проектирования. URL: <https://event.ru/overviews/10-poleznyih-prilozheniy-dlya-vedeniya-event-proektov/> (дата обращения: 10.05.2023).

УДК 004.4.274

**Д. Ф. Виноградов**

студент кафедры информационных систем и технологий

**Б. К. Акопян** – ассистент – научный руководитель

## РЕШЕНИЕ ПРОБЛЕМ, ВОЗНИКАЮЩИХ ПРИ КОМПЬЮТЕРНОЙ ПОСТОБРАБОТКЕ ВИДЕОДАНЫХ

### Введение

В настоящее время важную роль при создании видео играет этап компьютерной постобработки снятых видеоданных. Современный уровень аппаратно-программного обеспечения позволяет существенно упростить процедуру видеомонтажа. Однако, даже при использовании довольно мощных компьютеров у пользователей возникает ряд проблем, таких, как проблемы совместимости данных и производительности программы обработки.

В качестве такой программы рассматривается Adobe Premiere Pro.

### Формулировка проблемы

Пользователи Adobe Premiere Pro часто встречаются с проблемой низкой скорости воспроизведения видеофайлов в данной программе. В предложенном ниже материале мы рассмотрим несколько методов увеличения быстродействия Premiere Pro.

### Рекомендации

В первую очередь, работать рекомендуется в английской версии программы, так как большинство обучающих материалов рассчитано на использование этой версии. Устанавливать Premiere Pro желательно на самый быстрый диск, желательно на SSD (Solid State Drive, твердотельный накопитель).

### Настройки

Начинать оптимизацию целесообразно с настроек программы, для этого нажимаем слева сверху edit (изменить), далее preferences (установки):

Здесь необходимо выбрать Media cache (кэш медиаданных) и в нем установить локацию временных файлов (рис. 1). Далее необходимо выбрать самый быстрый диск, который есть в системе. Желательно для этих целей иметь SSD, но если в компьютере установлен медленный жесткий диск под систему, то не следует туда ставить временные файлы, потому что сама система может начать работать медленнее. Настройки, указанные на рис. 2, отвечают за то, когда будут удаляться временные данные, устанавливаем их исходя из загруженности диска.

Затем нужно перейти во вкладку memory (память), установить минимальное значение в RAM reserved for other application (ОЗУ, зарезервированное для других приложений), тем самым выделив программе максимальное значение оперативной памяти. В параметре Optimize rendering for (оптимизировать рендеринг для) необходимо выбрать пункт performance (производительность), что значит, что программа не будет экономить оперативную память вопреки производительности.

При создании проекта в программе во вкладке General (общие) есть пункт renderer (средство рендеринга). При наличии подходящего видеоадаптера желательно выбирать Mercury Playback Engine GPU Acceleration (Аппаратное GPU-ускорение ядра Mercury Playback) (рис. 3).

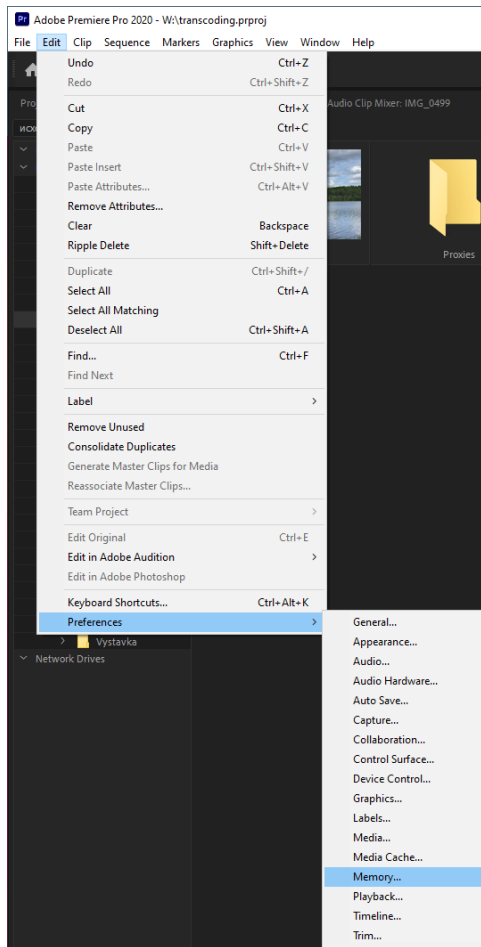


Рис. 1. Панель редактирования

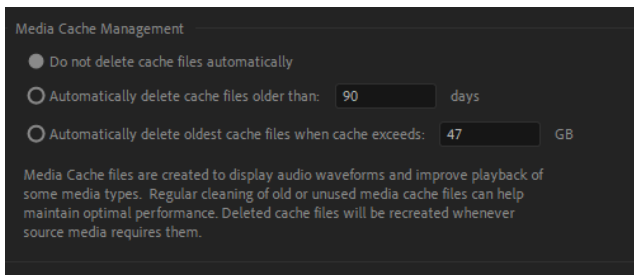


Рис. 2. Панель настроек

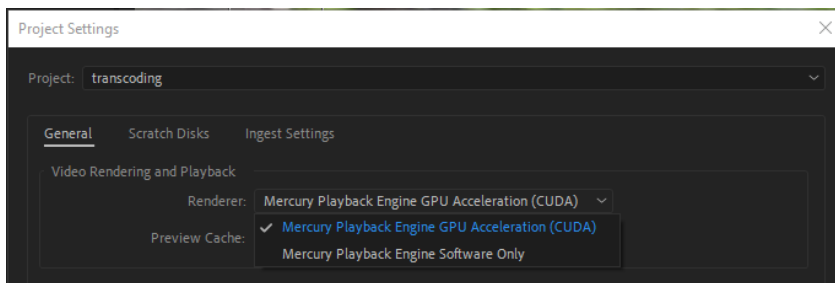


Рис. 3. Настройка проекта

### Транскодирование

В данном разделе будет рассмотрен процесс транскодирования видео. Транскодирование – это перевод видео из одного кодека в другой, в данном случае из просмотрowego кодека (такие как h.264,h.265) в монтажный (например Apple ProRes), потому что кодеки в которых снимают телефоны больше оптимизированны на размер файла, а не производительность при его редактировании.

Для данной операции обязательно понадобится Adobe Media Encoder [1].

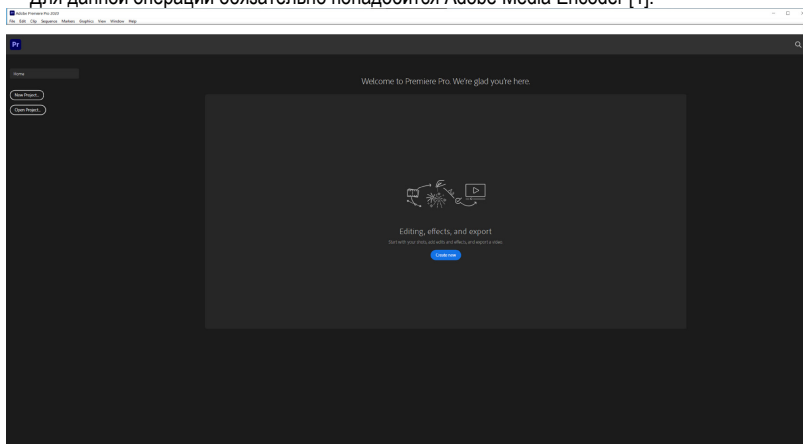


Рис. 4. Начальный экран программы

Для осуществления процесса транскодирования, необходимо создать новый проект (рис. 4), зайдя во вкладку Ingest settings (Настройки вставки), нажать галочку Ingest (вставить) и напротив галочки выбрать Transcode (перекодирование), рис. 5.

Во вкладке preset (стиль) целесообразно использовать Apple ProRes 422 LT, если Битрейт выше 100 мб/сек можно поставить Apple ProRes 422 (рис. 5) Во вкладке Primary destination (основное место назначения) можно выбрать расположение файлов.

Далее необходимо импортировать нужные нам видео в Premiere Pro, сделать это можно несколькими способами, например, выбрать в проводнике нужные нам файлы и перетащить на таймлайн, после этого автоматически откроется Adobe Media Encoder (рис. 6) который начнет автоматическое транскодирование наших видео, это может занять некоторое время.

Для примера возьмем видео с кодеком hevc(h.265) который часто встречается в современных мобильных телефонах, но имеет проблемы с быстродействием в Premiere Pro.

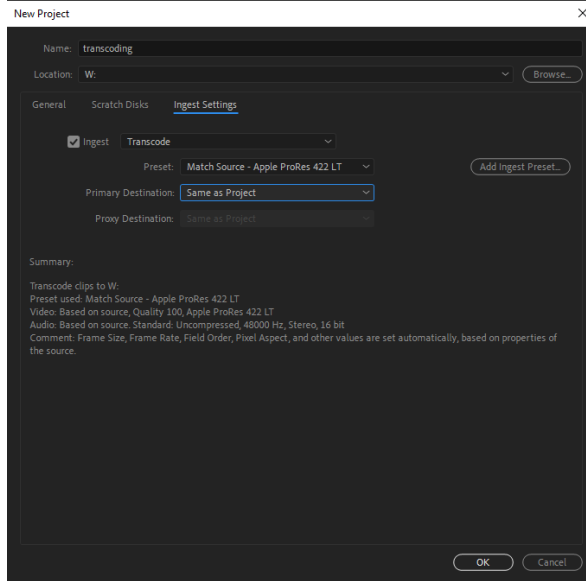


Рис. 5. Настройка вставки

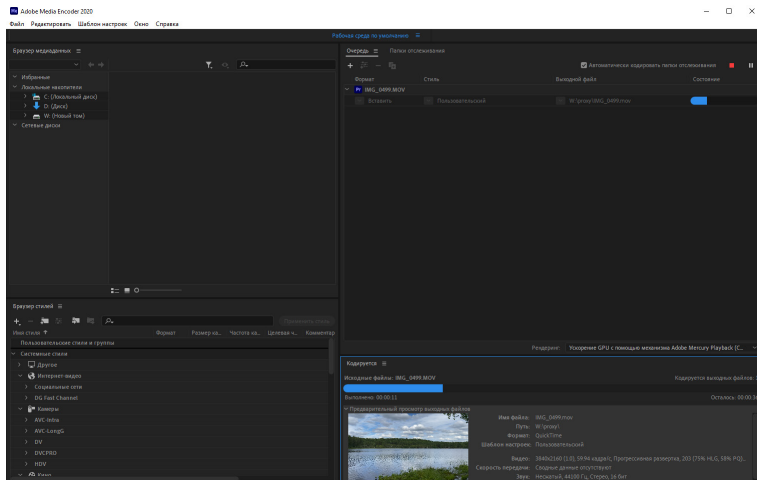


Рис. 6. Adobe Media Encoder 2020

Транскодирование выполнено, теперь на таймлайне уже есть видео в монтажном кодеке [2].

### Настройки предпросмотра и пререндер

С помощью уменьшения масштаба рендера предпросмотра можно облечить компьютеру воспроизведение секвенции на таймлайне (рис. 7).

Эту настройку можно найти в окне предпросмотра, full (полное) обозначает, что предпросмотр будет обрабатываться в полном разрешении, а 1/4, например, будет обрабатывать только одну четвертую от исходного разрешения. Уменьшая масштаб, мы увеличиваем скорость, но при этом просматриваем картинку в худшем качестве.

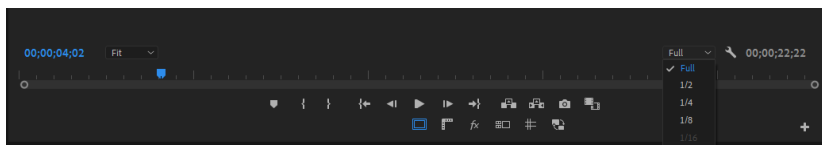


Рис. 7. Таймлайн

Еще одним приемом, который может облегчить воспроизведение, является пререндер. Когда к видео добавляют эффекты, таймлайн становится красным (рис. 8) и воспроизведение становится прерывистым. Чтобы решить эту проблему мы должны нажать на окно таймлайна левую кнопку мыши (лкм), затем клавишу enter [3].

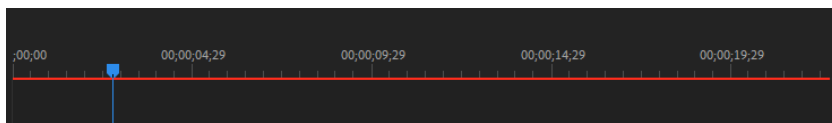


Рис. 8. Таймлайн, после вставки эффектов

После процесса пререндера (рис. 9), таймлайн станет зеленого цвета (рис. 10).

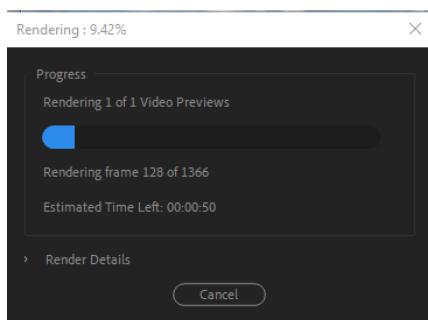


Рис. 9. Рендеринг видео

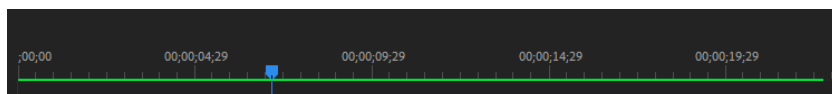


Рис. 10. Таймлайн, после пререндера

Но, как только изменится видео, например, в результате обрезки, полоса опять станет красной, и возникнет необходимость делать пререндер заново, всегда следует это учитывать.

Пожалуй, последним инструментом (кнопкой) будет Global FX Mute (Глобальное приглушение эффектов), который полностью убирает эффекты и цветокоррекцию на тот момент, когда нажата данная кнопка. Это может понадобиться, когда уже сделана цветокоррекция и наложены эффекты на видео, но

при этом необходимо, например, его обрезать. Чтобы добавить кнопку надо нажать на плюсики справа снизу окна предпросмотра (редактор кнопок), рис. 7.

И, зажимая лкм, перенести иконку  $f_x$  (рис. 11) на панель инструментов, теперь, нажимая на нее, можно просматривать видео быстрее за счет отключения эффектов [4].

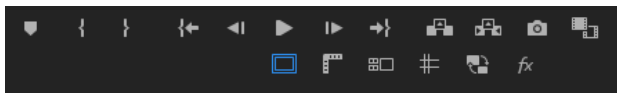


Рис. 11. Панель управления

### Вывод

С помощью грамотной оптимизации настроек программы Premiere Pro можно решить целый ряд проблем, связанных с компьютерной обработкой видеоданных.

### Библиографический список

1. Media Encoder. Конвертация и рендеринг видео для любых экранов. URL: <https://www.adobe.com/ru/products/media-encoder.html> (дата обращения: 27.04.2023).
2. Как работать в Adobe Premiere Pro. URL: <https://adobe-premiere.pro/kak-polzovatsya/> (дата обращения: 27.04.2023).
3. Adobe Premiere Pro для новичков: как смонтировать простое видео. URL: <https://yandex.ru/turbo/lifehacker.ru/s/adobe-premiere-pro/> (дата обращения: 27.04.2023).
4. 20 лучших уроков по использованию Premiere Pro. URL: <https://wondershare.com.ru/adobe-premiere/premiere-pro-for-beginners.html> (дата обращения: 28.04.2023).



УДК 004

**А. Р. Гарипов**

студент кафедры прикладной информатики

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

## МИКРОФРОНТЕНДЫ В СОВРЕМЕННОЙ WEB-РАЗРАБОТКЕ

### Введение

В контексте современной разработки веб-приложений наблюдается необходимость адаптации к постоянным изменениям технологического прогресса и растущей сложности разрабатываемого кода. Если серверная разработка находится в постоянном состоянии развития и обогащения архитектурными подходами, то клиентская разработка только недавно стала объектом внимания. Однако, некоторые архитектурные подходы, ранее применявшиеся только для серверной разработки, стали применимы и в клиентской разработке, включая микросервисную архитектуру, которая успешно применяется для создания независимых компонентов веб-приложения. Учитывая эти факты, становится ясно, что современная веб-разработка предъявляет к разработчикам высокие требования к адаптивности и гибкости, чтобы успешно справляться с растущей сложностью и изменениями в технологической среде.

При разработке клиентской части веб-приложений на сегодняшний день используются множество различных инструментов: Vue, React, Angular, Svelte и другие. Все они предоставляют удобные и простые решения для создания архитектуры клиентской части, что позволяет сократить время и усилия, затрачиваемые на разработку приложений. Однако по мере разработки приложения код превращается в монолит, что со временем может привести к проблемам по нескольким причинам. Во-первых, приложение может стать слишком сложным и разросшимся, что затрудняет его поддержку и масштабирование. Во-вторых, в монолитном приложении изменения в одной его части могут затронуть другие, что усложняет разработку и тестирование. Кроме того, монолитный подход затрудняет внедрение новых технологий и инструментов, а также ограничивает возможности расширения функционала. Все это способно привести к ухудшению производительности и качества приложения в целом.

Однако данную проблему можно избежать, если применить микросервисный подход для разработки клиентской части [1]. Микросервисный подход – это такой подход к разработке программного обеспечения, который предлагает разбить большое приложение на множество небольших, слабо связанных, независимых сервисов. Каждый сервис выполняет конкретную задачу и все взаимодействие происходит через API. Это позволяет работать с меньшими, более легко управляемыми частями кода, а также упрощает процесс развертывания и масштабирования приложения. Кроме того, использование микросервисов позволяет более эффективно использовать время разработчиков, так как каждый разработчик может работать над своим сервисом параллельно с другими.

На основе микросервисного подхода главная идея микрофронтенда заключается в том, чтобы взаимодействовать с веб-приложением как с набором отдельных приложений, которые работают совместно и могут принадлежать различным командам [2]. Каждая команда отвечает за определенную бизнес-логику, о которой она заботится и на которой специализируется. При этом команда является кросс-функциональной и работает над своим приложением от начала до конца, от базы данных до пользовательского интерфейса.

Для примера, рассмотрим архитектуру обычного интернет-магазина. У веб-приложения есть три составляющие: клиентская часть (frontend) для взаимодействия с пользователем через интерфейс, серверная часть (backend) для обработки логики работы с данными и база данных (database) для безопасного хранения данных. Обычно каждый интернет-магазин состоит из следующих модулей:

- домашняя страница для отображения рекомендованных продуктов;
- страница корзины;
- страница оформления заказа;
- страница оплаты.

Рассмотрим для начала традиционные подходы к разработке приложений. К ним можно отнести монолитную архитектуру и архитектуру клиент-сервер. В данных подходах внутренняя и внешняя

логика, как и другие задачи, генерируются в одной кодовой базе. Схема данных подходов изображена на рис. 1. Такой подход сильно упрощает разработку, так как в процессе реализации необходимого функционала разработчику не приходится задумываться над проблемами коммуникации логических компонентов.

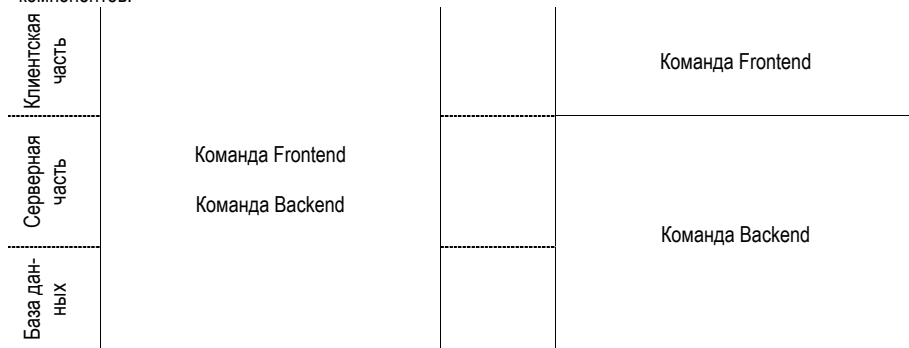


Рис. 1. Монолитный подход и архитектура клиент-сервер

В свою очередь микросервисная архитектура позволяет работать с данными как с набором отдельных приложений. Хотя микро-интерфейсы и микросервисы часто используются вместе, они не являются взаимозависимыми. Можно использовать микро-интерфейсы без микросервисов на бэкенде, и наоборот. Все микро-приложения будут существовать как независимые малосвязанные между собой приложения (рис. 2).

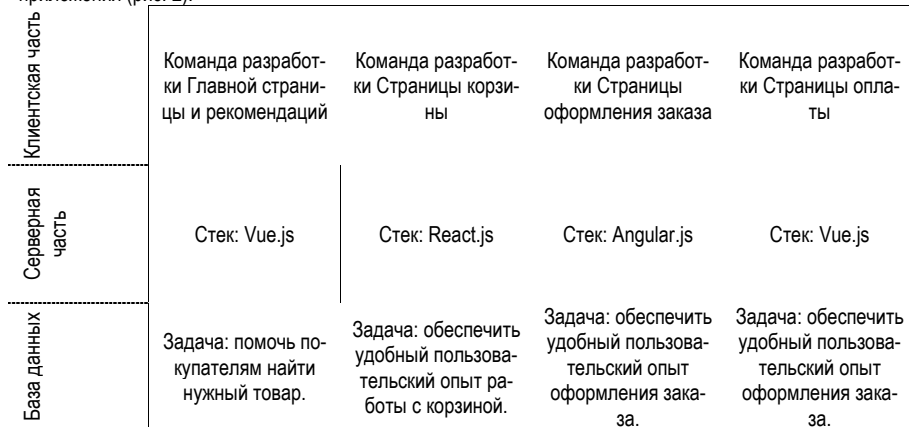


Рис. 2. Микросервисный подход реализации интернет-магазина

Также стоит отметить, что каждое из мини-приложений может быть реализовано на том технологическом стеке, к которому привыкла команда разработки. На данной схеме каждая команда разработки занимается определенной бизнес-задачей, которая независима от других задач. При этом технологический стек каждой команды различается. Рассмотрим преимущества и недостатки микрофронтендов.

### Преимущества:

1. Предоставляет возможность непрерывной доставки и развертывания сложных приложений, что сокращает время выхода продукта на рынок и обеспечивает высокий уровень надежности услуг компании в соответствии с современными ожиданиями.
2. Микро-интерфейсы получаются небольшими и простыми в обслуживании.
3. Независимое масштабирование сервисов друг от друга.
4. При использовании микрофронтендов разработчики могут работать над отдельными микро-сервисами, что позволяет им использовать различные технологии и языки программирования, а также делает возможным разделение обязанностей между командами.
5. Переиспользование кода. Микрофронтенды могут использоваться в разных приложениях, что позволяет повторно использовать код и ускорить процесс разработки.
6. Обеспечивает возможность экспериментировать и внедрять новые технологии в различные сервисы независимо друг от друга.
7. Утечка памяти или другие проблемы в работе одного сервиса не будут влиять на работу других сервисов, что обеспечивает улучшенную изоляцию проблем и минимизирует негативные последствия от отказов.

### Недостатки:

1. Использование микрофронтендов требует более сложного процесса разработки и координации между командами, что может быть сложно для небольших проектов.
2. Тестирование микрофронтендов может быть сложным из-за необходимости тестировать каждый микросервис отдельно и в контексте всего приложения.

Таким образом, на основе преимуществ и недостатков данной архитектуры можно сделать вывод, что микро-интерфейсы не столько уменьшают сложность системы, сколько делают ее более управляемой [3].

Существует несколько подходов в построении микрофронтендов [4]. Рассмотрим наиболее популярные из них:

1. WebPack Module Federation. Это подход, который позволяет создавать микрофронтенды на основе webpack, который является популярным инструментом для сборки фронтенд-приложений. Webpack Module Federation предоставляет механизмы для изоляции кода микрофронтендов друг от друга, что повышает безопасность и устойчивость приложения, а также определяет, как приложения зависят друг от друга и совместно используют общие библиотеки.

2. Import-map. Это механизм загрузки и разрешения зависимостей для JavaScript-модулей, который позволяет использовать модули из разных источников, в том числе из разных микрофронтендов. Одним из фреймворков, который использует подход import-map для реализации микрофронтендов, является single-spa. Single-spa это фреймворк для создания приложений на основе микрофронтендов, который позволяет интегрировать различные фронтенд-приложения, написанные на разных фреймворках и языках программирования.

3. Web Components. Использование подхода Web Components в архитектуре микрофронтенда позволяет разбить пользовательский интерфейс на более мелкие, переиспользуемые части, каждая из которых может быть реализована в виде отдельного микросервиса. Компонент может быть написан на любом языке программирования и использовать любые фреймворки или библиотеки.

4. Iframes. Суть подхода заключается в том, что каждый компонент приложения представлен в виде отдельной HTML-страницы, которая загружается в iframe на странице главного приложения.

### Заключение

В заключение, стоит отметить, что микрофронтенды – это не универсальное решение для всех сценариев разработки веб-приложений. Однако, они могут быть эффективным инструментом для решения проблем с переписыванием и расширением программного обеспечения, а также для обеспечения независимого деплоя приложений. Окончательное решение о применении данного подхода зависит от размера команды разработчиков и конкретных потребностей проекта, а использование микрофронтендов может повысить эффективность работы и улучшить процесс разработки.

**Библиографический список**

1. Micro Frontends – extending the microservice idea to frontend development // Micro Frontends. URL: <https://micro-frontends.org/> (дата обращения: 23.03.2023).
2. Yang C., Liu C., Su Z. Research and Application of Micro Frontends // IOP Conf. Ser. Mater. Sci. Eng. 2019. Vol. 490. P. 062082. URL: [https://www.researchgate.net/publication/332386596\\_Research\\_and\\_Application\\_of\\_Micro\\_Frontends/](https://www.researchgate.net/publication/332386596_Research_and_Application_of_Micro_Frontends/) (дата обращения: 24.03.2023).
3. Прокофьев А. П. Особенности Архитектуры Микрофронтендов // Инновации Наука Образование. 2022. № 50. URL: [https://www.elibrary.ru/download/elibrary\\_48046202\\_55569589.pdf](https://www.elibrary.ru/download/elibrary_48046202_55569589.pdf) (дата обращения: 24.03.2023).
4. Sultan A.S. Efficiency of micro frontend architecture: performance and convenience // Вопросы Устойчивого Развития Общества. 2022. № 5. URL: [https://www.elibrary.ru/download/elibrary\\_48690125\\_70269296.pdf](https://www.elibrary.ru/download/elibrary_48690125_70269296.pdf) (дата обращения: 24.03.2023).

УДК 004

**В. В. Господынюк**

студент кафедры прикладной информатики

**Е. Л. Турецкая** – кандидат технических наук – научный руководитель**ИНСТРУМЕНТЫ И ПОДХОДЫ К ТЕСТИРОВАНИЮ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ****Введение**

Тестирование программного обеспечения — это процесс, который включает в себя оценку качества программного продукта с помощью различных методов, техник и инструментов. Целью является выявление ошибок и дефектов в программном продукте, чтобы обеспечить его безопасность, надежность, эффективность, удобство использования и соответствие требованиям заказчика.

Тестирование программного обеспечения проводится на всех этапах жизненного цикла разработки, включая тестирование на начальном этапе проектирования, тестирование при разработке кода, интеграционное тестирование при сборке всех компонентов программы, и тестирование после выпуска продукта.

В целом, тестирование программного обеспечения является неотъемлемой частью процесса разработки программного продукта, которая помогает выявлять ошибки и дефекты, улучшать качество продукта, повышать его надежность и безопасность, и обеспечивать соответствие требованиям заказчика.

Тестирование программного обеспечения может быть проведено в различных формах, включая ручное тестирование и автоматизированное тестирование.

Ручное тестирование является важной частью процесса разработки программного обеспечения и оценки его качества. Одной из основных причин, по которым данный вид тестирования продолжает оставаться важным, является то, что люди все еще могут замечать тонкие детали и неожиданные поведения, которые могут быть упущены при автоматизированном тестировании. Человеческий фактор является ключевым при рассмотрении пользовательского опыта и оценки удобства использования продукта. Например, проверка юзабилити всегда осуществляется вручную, поскольку она требует непосредственного взаимодействия с реальными пользователями и оценки их поведения и отзывов. Тестирование юзабилити позволяет убедиться, что продукт удобен и понятен для пользователей. Хорошо спроектированный интерфейс может существенно повысить удобство использования продукта и увеличить его популярность.

Тем не менее, ручное тестирование также имеет свои ограничения и недостатки. Недостатки ручного тестирования включают в себя ограниченность его масштабируемости, трудоемкость, необходимость квалифицированного персонала и повторяемость процесса. Кроме того, ручное тестирование может быть скучным и монотонным процессом для тестировщика, что может привести к снижению эффективности и уровню мотивации.

Для повышения эффективности и качества процесса тестирования, ручное тестирование обычно дополняется автоматизированным тестированием.

Автоматизированное тестирование ПО (Automation Testing) — это процесс верификации программного обеспечения, при котором основные функции и шаги теста, такие как запуск, инициализация, выполнение, анализ и выдача результата, выполняются автоматически при помощи инструментов для автоматизированного тестирования. Автоматическое тестирование по заданному сценарию не требует участия человека – система сама тестирует программный продукт во всех нужных режимах. Вмешательство человека требуется только для пополнения библиотеки сценариев, изучения отчетов и поддержки тестов в актуальном состоянии [1].

Для определения того, что следует автоматизировать, необходимо сначала оценить целесообразность автоматизации тестирования в рамках проекта. Если автоматизация действительно нужна, то следует разработать план, учитывая требования объекта тестирования, по которому будут создаваться автоматизированные тесты. При разработке такого плана необходимо ясно определить, что и как нужно автоматизировать [2]. Автоматизация тестирования обычно позволяет ускорить и удешевить процесс тестирования, но есть условия, при которых она особенно эффективна:

- Рутинные операции, например, проверка форм, в которых необходимо заполнять большое количество полей.
- Труднодоступные места в системе, такие как запись в базу данных, бекэнд процессы, логирование.
- Валидационные сообщения (проверка появления валидации при некорректном заполнении поля).
- Проверка корректного поиска данных.
- Проверки, требующие точных математических расчетов.
- Длинные end-to-end сценарии.

При выборе инструментов для автоматизированного тестирования следует учитывать множество факторов.

Первым важным фактором является цель тестирования. Некоторые инструменты лучше подходят для функционального тестирования, другие – для тестирования производительности или безопасности.

Второй фактор – масштаб проекта. Не все инструменты подходят для больших проектов. Необходимо выбрать инструмент, который сможет автоматизировать тестирование на всех этапах разработки.

Третий фактор – доступность инструмента. Некоторые инструменты требуют дополнительных расходов, связанных с обучением или лицензированием.

Четвертый фактор – опытность команды. Не все инструменты одинаково просты в использовании. Важно, чтобы команда тестировщиков была опытна в работе с выбранным инструментом.

Существует множество инструментов автоматизированного тестирования [3].

Некоторые из них:

1. Selenium: инструмент для автоматизированного тестирования веб-приложений.
2. Appium: фреймворк для автоматизации тестирования мобильных приложений.
3. TestComplete: платформа для автоматизации функционального, нагрузочного и UI тестирования.
4. JMeter: инструмент для нагрузочного тестирования веб-приложений.
5. SoapUI: инструмент для тестирования API.
6. Postman: инструмент для тестирования API и создания запросов.
7. Pytest: фреймворк для написания автоматических тестов на языке Python.
8. Cucumber: инструмент для создания автоматизированных функциональных тестов на основе языка Gherkin.
9. Testim: автоматизированное тестирование на основе машинного обучения.
10. Cypress: инструмент для автоматизированного тестирования веб-приложений на основе JavaScript.

В целом, при выборе инструментов автоматизированного тестирования необходимо учитывать все эти факторы и выбирать инструмент, который лучше всего подходит для конкретного проекта.

### Заключение

Таким образом, можно сделать вывод, что тестирование является одним из основных способов обеспечения качества программных продуктов. Стремительно развивающаяся сложность проектов требует все большего количества тестов, а проверки после внесения любых изменений в системе могут быть обеспечены построением сценариев автоматизированного тестирования.

Ручное тестирование является неотъемлемой частью процесса тестирования, человеческий фактор является ключевым при рассмотрении пользовательского опыта и оценки удобства использования продукта, в то время как автоматизированное тестирование ускоряет и удешевляет процесс тестирования, особенно в условиях повторяемости и масштабности тестовых сценариев. Переход к автоматизированному тестированию должен быть рассмотрен с учетом требований объекта тестирования и разработан соответствующий план.

В заключение можно сказать, что ручное тестирование продолжает оставаться важным элементом процесса тестирования программного обеспечения, однако автоматизированное тестирование

может значительно улучшить эффективность и качество процесса. Комбинация ручного и автоматизированного тестирования может помочь повысить качество продукта.

**Библиографический список**

1. Винниченко Илья. Автоматизация процессов тестирования. СПб.: Питер, 2005 г. 203 с.
2. Элфрид Дастин, Джефф Рэшка, Джон Пол. Автоматизированное тестирование программного обеспечения. М.: Лори, 2019. 567 с.
3. Top 15 Automation Testing Tools | Latest Update in 2023. URL: <https://katalon.com/resources-center/blog/automation-testing-tools> (дата обращения: 05.04.2023).

УДК 004.75

**А. В. Душеба**

студент кафедры прикладной информатики

**Н. Н. Григорьева** – старший преподаватель – научный руководитель**СКАНЕР BLOCKCHAIN ТРАНЗАКЦИЙ ДЛЯ СЕТЕЙ НА БАЗЕ EVM****Введение**

Ethereum – это децентрализованная блокчейн-платформа, которая позволяет создавать смарт-контракты и децентрализованные приложения. Транзакции являются фундаментальными строительными блоками блокчейна Ethereum. Транзакции используются для передачи эфира (ETH) и выполнения смарт-контрактов. Каждая транзакция имеет уникальный идентификатор, называемый хешем транзакции, который можно использовать для отслеживания транзакции в блокчейне Ethereum. EVM (Ethereum Virtual Machine) – среда выполнения кода внутри блокчейна. На данный момент существует множество блокчейн-сетей, работа которых основана на технологии EVM [1].

Сканирование транзакций — это процесс анализа и мониторинга транзакций Ethereum, который используется для работы с данным, обнаружения подозрительных транзакций и выявления потенциальных угроз безопасности. Существует несколько методов сканирования транзакций, однако они не подходят для мониторинга транзакций в реальном времени [2].

**Описание задачи**

Задача – сканирование блокчейна на базе виртуальной машины Ethereum для предобработки и сохранения данных транзакций в базе данных (БД) в реальном времени; работа с данными через REST API.

**Постановка задачи**

Для того чтобы выполнить сканирование транзакций в EVM сети, необходимо: знать адрес смарт-контракта, для которого выполняется сканирование, его двоичный программный интерфейс (ABI), а также иметь доступ к узлу блокчейна, например, используя RPC [3].

В момент вызова события, обработчик событий должен оценить, является ли транзакция новой или уже обработанной, выполнить предобработку и сохранить данные в соответствующую таблицу. Для последующей работы с данными могут быть созданы необходимые конечные точки REST API.

**Описание программы**

Для решения задачи была написана программа на языке JavaScript. В качестве блокчейна был использован Ethereum, работа с которым осуществлялась при помощи библиотеки web3.js и RPC-сервиса Infura. Для создания API была использована библиотека Express. Для сохранения данных была выбрана БД MongoDB, что связано с динамической структурой схемы данных. На рис. 1 показана схема данных, которая включает в себя таблицу EventSyncStatus, содержащую информацию о синхронизируемых контрактах и их состоянии. В таблицы Class 1, Class 2, ..., Class N записывается обработанная информация, полученная с блокчейна.

Блок-схема алгоритма предобработки транзакции показана на рис. 2. Она включает: инициализацию сканера (активация слушателя событий), основной рабочий цикл. При получении события выполняется проверка на наличие данных о событии в таблице EventSyncStatus (ESS), проверяется номер блока полученной транзакции. В случае успешного выполнения условий, информация о синхронизации в таблице ESS обновляется, вызывается функция – обработчик данных.

На рис. 3 показано содержание таблицы, в которую сохранены полученные в результате сканирования блокчейна данные событий goomCreated, включая хэш блока и хэш транзакции, адрес контракта, дату события, а также полезную нагрузку.



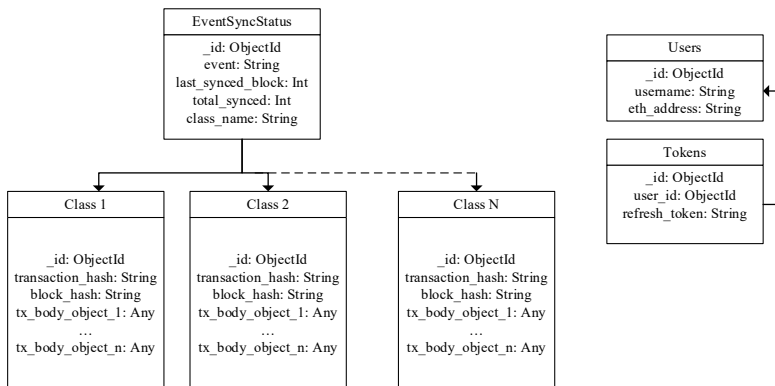


Рис. 1. Схема данных

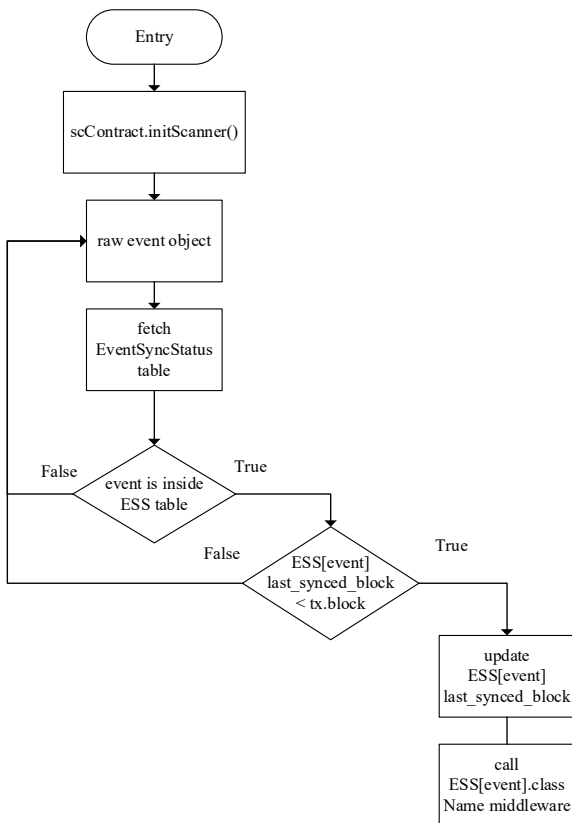


Рис. 2. Блок-схема алгоритма

## Artrooms.roomCreated

Documents Aggregations Schema Explain Plan Indexes Validation

Filter Type a query: { field: 'value' }

ADD DATA EXPORT COLLECTION

```

_id: ObjectId('633fe9f78c56f33918fa7f96')
room: Array
  0: "1"
  1: "0"
  2: "0"
  3: "Test Gallery"
  4: "0x881a32Dfd99C2AAAd18cd25A658B2beDc582385b"
  5: "0x881a32Dfd99C2AAAd18cd25A658B2beDc582385b"
  6: "150"
  7: "370"
  8: "300"
  9: "150"
  10: "15000000000000000000"
  11: "0"
  12: "desc"
  13: "https://multygallery.mypinata.cloud/ipfs/QmaN4oQd2s3bFdsVvuHA8ZA3gsd5g..."
  14: ""
createdAt: 2022-10-07T08:57:27.142+00:00
transaction_hash: "0x4a04353a8a2eb0e85d77471895b72b9e9c9761bc9b12d98a19d0c48491e7e622"
block_hash: "0xecd7c8864cfd0045c9774246ceff38267481cf35d9afaa7a1aa0d77c8ca994"
address: "0x13ef99992c57896fcef2ff33729bc55c5e5f02d7"
    
```

Рис. 3. Содержимое таблицы roomCreated

### Заключение

Сканирование блокчейна через подпись на события позволяет обрабатывать и сохранять данные вне блокчейна, работать с ними по API в реальном времени. Представленный алгоритм может быть использован для создания backend-сервисов для работы web3-приложений.

### Библиографический список

1. Huang Yuxin & Wang, Ben & Wang, Yinggui. Research and Application of Smart Contract Based on Ethereum Blockchain // ISCM 2020 Journal of Physics: Conference Series. № 1748. 2021. P. 283–297.
2. Blockchain-Enabled Transaction Scanning Method for Money Laundering Detection / A. Oad, A. Razaque, A. Tolemyssov, M. Alotaibi [et al]. Preprints, 2021. P. 1–8.
3. A Model for Detecting Cryptocurrency Transactions with Discernible Purpose / H. Baek, J. Oh, C. Y. Kim, K. Lee // Eleventh International Conference on Ubiquitous and Future Networks (ICUFN). Zagreb, Croatia. 2019. P. 713–717.

УДК 004.77

**А. В. Душеба**

студент кафедры прикладной информатики

**Н. Н. Григорьева** – старший преподаватель – научный руководитель

## WEB-AВТОРИЗАЦИЯ НА ОСНОВЕ ТЕХНОЛОГИИ BLOCKCHAIN

### Введение

Важным аспектом веб-безопасности является веб-аутентификация — процесс проверки личности пользователя, обращающегося к веб-приложению или службе. Аутентификация гарантирует, что только авторизованные пользователи смогут получить доступ к конфиденциальным данным или выполнять действия от имени пользователя. Однако традиционные методы веб-аутентификации, такие как имена пользователей и пароли, имеют ряд недостатков, включая повторное использование паролей, подверженность фишинговым атакам и централизованное управление идентификацией.

MetaMask – это расширение для браузера, которое предоставляет пользователям безопасный и децентрализованный способ управления своей цифровой идентификацией и взаимодействия с приложениями на основе блокчейна. Одной из ключевых особенностей MetaMask является его способность аутентифицировать пользователей с помощью подписи сообщений. Этот метод позволяет пользователям подтвердить свою личность, не раскрывая свои закрытые ключи или пароли сторонним службам. Используя аутентификацию через MetaMask, веб-разработчики могут предложить своим пользователям безопасный и простой в использовании метод аутентификации, который не зависит от централизованного управления идентификацией и не требует от пользователей запоминания сложных паролей [1].

### Описание задачи

Задача – создать веб-сервис для авторизации пользователей через подпись сообщения по принципу асимметричного шифрования с выдачей пары ключей access – refresh.

### Решение задачи

Для решения задачи был выбран Node.js и Express. Работа с web3 осуществлена через библиотеку web3.js.

Аутентификация пользователя включает в себя следующие этапы:

- Подпись сообщения, генерируемого на клиенте и подписываемого закрытым ключом пользователя.
- Отправка подписанного сообщения (хэша) вместе с адресом пользователя и оригинальным сообщением на бэкэнд.
- Расшифровка сообщения на бэкэнде с помощью открытого ключа (адреса пользователя).

Схема алгоритма авторизации показана на рис. 1.



Рис. 1. Схема алгоритма авторизации

Для авторизации на клиентской стороне необходимо вызвать метод “eth\_requestAccounts”, что может быть реализовано при использовании MetaMask следующим образом:

```

window.web3 = new Web3(window.ethereum);
window.ethereum.request({ method: "eth_requestAccounts" });
    
```

При вызове функции `eth_requestAccounts` MetaMask открывает всплывающее окно с подписываемым сообщением, пример показан на рис. 2.

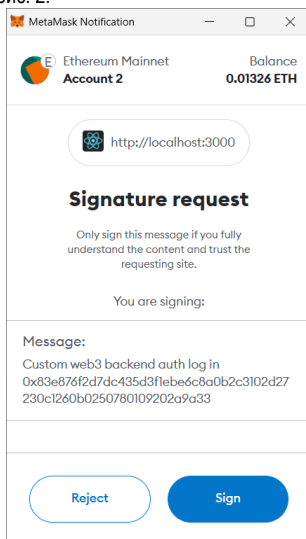


Рис. 2. Окно подписи сообщения

Для проверки сообщения на бэкенде можно использовать функцию `recoverMessage`:

```
recoverMessage(message: string, address: string, signature: string): boolean {
  const recovered_address = this.web3.eth.accounts.recover(message, signature);
  return recovered_address.toLowerCase() === address.toLowerCase();
}
```

После проверки подписи алгоритм повторяет сценарий обычной авторизации с использованием пары токенов `access – refresh`. На рис. 3 показан ответ от сервера при авторизации пользователя, который включает в себя пару токенов и метаданные пользователя – его адрес и пользовательское имя. Особенностью работы такого алгоритма является то, что для аутентификации пользователя достаточно сохранить в базе данных его публичный адрес [2].

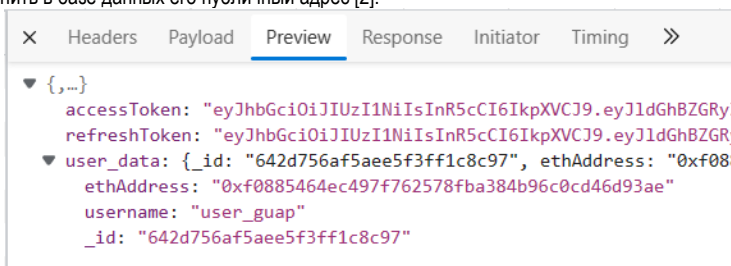


Рис. 3. Ответ от сервера

Использование описанного метода аутентификации позволяет не бояться утечки данных, так как стандартная схема авторизации предполагает сохранение на серверной части приложения логина и пароля, а предложенная схема позволяет хранить в базе данных лишь публичный адрес.

### **Заключение**

Используя авторизацию через подпись сообщения с помощью закрытого ключа на базе blockchain-кошелька, пользователи могут подтверждать свою личность, не раскрывая свои закрытые ключи или пароли, что снижает риск фишинговых атак и других уязвимостей системы безопасности.

### **Библиографический список**

1. Adrian Petcu, Bogdan Pahontu, Madalin Frunzete, Dan Alexandru Stoichescu. A Secure and Decentralized Authentication Mechanism Based on Web 3.0 and Ethereum Blockchain Technology // Multidisciplinary Digital Publishing Institute. Applied Sciences. Vol. 13. № 2231. 2023. P.1–17.
2. Van Giang Phan Mai. A Blockchain-based User Authentication Model Using MetaMask // URL: [https://www.researchgate.net/publication/370339058\\_A\\_Blockchain-based\\_User\\_Authentication\\_Model\\_Using\\_MetaMask](https://www.researchgate.net/publication/370339058_A_Blockchain-based_User_Authentication_Model_Using_MetaMask) (дата обращения 10.03.2023).

УДК 004.032.26

**И. А. Зорин**

студент кафедры прикладной информатики

**Д. В. Жданов**

студент кафедры прикладной информатики

**Н. Н. Григорьева** – старший преподаватель – научный руководитель

## НЕЙРОННАЯ СЕТЬ ДЛЯ ОБРАБОТКИ ИДИОМАТИЧЕСКИХ ВЫРАЖЕНИЙ ЕСТЕСТВЕННОГО ЯЗЫКА

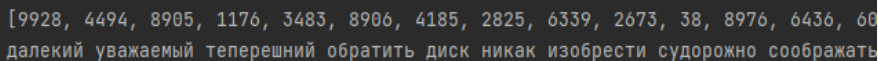
В настоящее время количество обсценной лексики, используемой в публичных диалогах сети интернет, резко увеличилось, однако программное обеспечение для модерирования данного вида лексики практически не изменилось, из-за чего возникает огромное количество простых способов обхода данного вида программ. Поэтому существует потребность в реализации удобной и универсальной системы, которая способна качественно модерировать пользовательские сообщения различной сложности [1].

Для обнаружения нецензурной лексики нет необходимости следить за основным содержанием сообщения, поэтому набор обрабатываемых данных можно представить как большое количество несвязанных между собой слов. Для работы был собран набор из нецензурной и обычной лексики, каждому слову было присвоено значение, указывающее на принадлежность к одному из двух классов: нецензурные слова (помечены единицей), остальные слова (помечены нулем). Для формирования предположений, на которых будет обучаться нейронная сеть, был создан алгоритм, который случайным образом формирует последовательности и присваивает им метку класса: 1 – когда последовательность содержит табуированную лексику, 0 – когда не содержит. Всего было сгенерировано 300 тысяч пар последовательность – метка, которые будут использоваться для обучения модели.

Данные для обучения были представлены в двух форматах.

1. Каждое слово последовательности представляется в числовом виде, список числовых представлений слов поступает на входной слой нейронной сети.

Весь текст представляется в числовом виде, для этого каждому слову последовательности присваивается числовой индекс; чем чаще слово встречалось в наборе данных, тем меньше будет его индекс. Пример числового представления текста показан на рис. 1. После присвоения каждому слову числового значения последовательности приводятся к размеру в 150 слов (если предложение длиннее 150 слов, первые  $n$  слов удаляются, если слов меньше, то в начало предложения добавляются нули). В результате предварительной обработки каждое предложение представлено в виде числового массива, содержащего 150 элементов.



[9928, 4494, 8905, 1176, 3483, 8906, 4185, 2825, 6339, 2673, 38, 8976, 6436, 60...]  
далекий уважаемый теперешний обратить диск никак изобрести судорожно соображать

Рис. 1. Представление данных в числовом виде

Основной недостаток данного метода заключается в том, что нейронная сеть работает с целыми словами, из-за чего при отсутствии в обучающем наборе данных слов во всех его возможных формах, нейронная сеть будет классифицировать данные слова как нейтральные. При наличии в предложении нецензурного слова в форме, которой не было в обучающем наборе, нейронная сеть не классифицирует данное предложение как содержащее нецензурную лексику. По этой причине было рассмотрено также представление исходных данных в виде триграмм.

2. Каждое слово предложения разделяется на триграммы, каждой триграмме присваивается свое числовое значение, список числовых представлений подается на входной слой нейронной сети.

Триграмма – элемент из трех идущих подряд символов. Предложения последовательно проходят через фильтр, который выделяет три идущих подряд символа, с шагом в один символ, в результате каждое слово предложения разделяется на  $n$  трехбуквенных элементов. Например, слово Носочки разделится на триграммы “Нос”, “осо”, “соч”, “очк”, “чки”. Предварительная обработка данных аналогич-

на первом методу представления данных. Триграммы позволяют решить проблему, которая возникает при первом виде представления данных, однако возникает проблема в идентичности триграмм в различных словах, за счет чего будут возникать неопределенности, когда нейронная сеть будет считать нейтральную последовательность триграмм за нецензурную лексику [2].

```
[1751, 669, 2001, 927, 2723, 1983, 2178, 493, 20, 367,
жил ило лой отс тсу сут утс тст ств тви вие дой ойт йти
```

Рис. 2. Представление данные в числовом виде при использовании три-грамм

Для реализации модели распознавателя использовался язык программирования Python и три библиотеки: Keras – для создания архитектуры модели, NumPy – для обработки данных, Matplotlib – для визуализации результатов нейронной сети. В качестве входного слоя выступает слой Embedding, данный слой преобразует каждое число в плотный вектор, представляющий собой массив из двух ненулевых числовых значений (рис. 3). Первым скрытым слоем является слой Flatten, который преобразует многомерный массив данных в одномерный, второй скрытый слой – слой Dropout, который с заданной вероятностью отключает случайные нейроны, чтобы предотвратить переобучение модели, третий скрытый слой – полносвязный слой Dense, состоящий из 60 нейронов с функцией активации Relu.

Выходным слоем также является слой Dense, который на выходе выдает одно число от 0 до 1, которое является вероятностью того, что подаваемая последовательность содержит obscene лексик [3, 4, 5, 6].

```
предложение и число
[[-0.03448546 -0.08665063]]
[ 0.02561544 -0.02272348]
[ 0.12550747  0.01532848]]
```

Рис. 3. Представление слов в виде плотных векторов

Для обучения модели был выбран метод обратного распространения ошибки. Метод основан на изменении весов нейронной сети после определенного количества итераций модели. Коррекция весов направлена на уменьшение значения ошибки предсказания, в модели в качестве ошибки используется бинарная кросс-энтропия, которая является показателем расстояния между вероятностными распределениями двух элементов; чем меньше данное расстояние, тем сильнее элементы похожи друг на друга. В качестве метода оптимизации градиента был выбран Adam, который в зависимости от близости к минимуму градиента меняет величину изменения весов, за счет чего на ранних этапах обучение проходит намного быстрее, чем на поздних [7, 8].

Результаты обучения модели при представлении набора данных в первом виде показаны на рис. 4. Распределение векторов слов показано на рис. 5

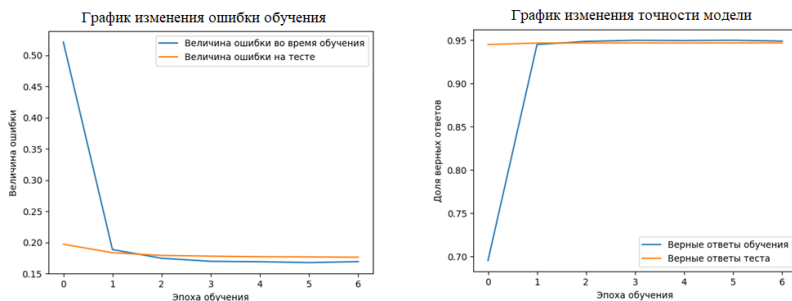


Рис. 4. Величина ошибки и точности модели (первая модель данных)

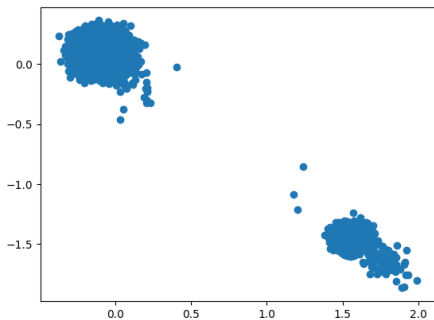


Рис. 5. Распределение слов по кластерам

7. Результаты обучения модели при представлении данных в виде триграмм показаны на рис. 6 и

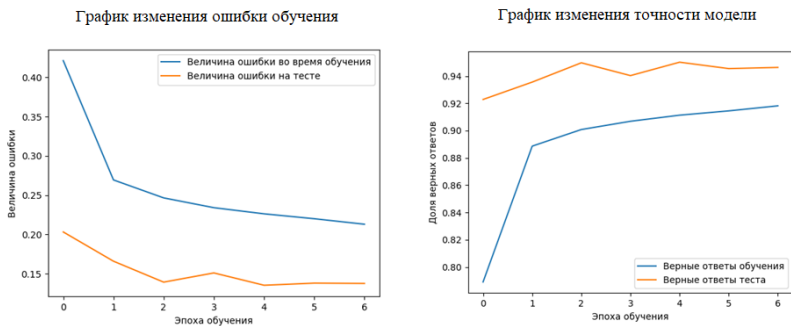


Рис. 6. Величина ошибки и точности модели (второй метод представления данных)

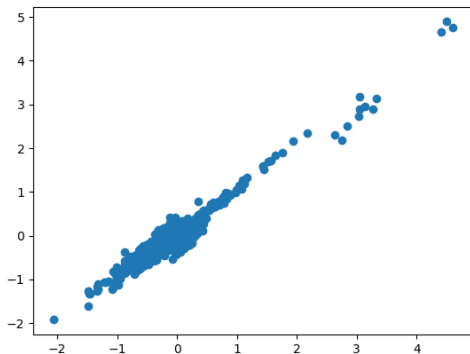


Рис. 7. Распределение триграмм по кластерам



Из результатов обучения видно, что при первом методе представления данных модель распознавателя показывает лучшие результаты, нежели при втором; также при первом методе модель в явном виде разделила слова на два кластера (рис. 5). За счет этого классификация проходит точнее, нежели при втором методе, так как векторные представления слов уже разделены на кластеры.

На основе полученных результатов можно сделать вывод, что для исходных данных, представленных в виде триграмм, лучше использовать архитектуру нейронной сети со связью с предыдущими элементами. Таким образом будет преодолена неоднозначность распределения триграмм по классам, так как распределение будет делаться на основе рассмотрения не только конкретного три-грамма, но и его соседей. Полносвязная модель лучше работает с первым методом представления данных, так как в этом случае связь между словами практически не играет роли, из-за чего модель с высокой точностью классифицирует текст.

### Библиографический список

1. Как пользователи соцсетей матерились в 2022-м году. URL: <https://vc.ru/social/530333-kak-polzovateli-socsetey-materilis-v-2022-m-godu> (дата обращения: 20.01.2023).
2. N-граммы для машинного обучения. URL: [https://qudata.com/ml/ru/ML\\_NGrams.html](https://qudata.com/ml/ru/ML_NGrams.html) (дата обращения: 12.03.2023).
3. Embedding layer. URL: [https://keras.io/api/layers/core\\_layers/embedding/](https://keras.io/api/layers/core_layers/embedding/) (дата обращения: 12.03.2023).
4. Dropout layer. URL: [https://keras.io/api/layers/regularization\\_layers/dropout/](https://keras.io/api/layers/regularization_layers/dropout/) (дата обращения: 12.03.2023).
5. LSTM layer. URL: [https://keras.io/api/layers/recurrent\\_layers/lstm/](https://keras.io/api/layers/recurrent_layers/lstm/) (дата обращения: 10.03.2023.)
6. Dense layer. URL: [https://keras.io/api/layers/core\\_layers/dense/](https://keras.io/api/layers/core_layers/dense/) (дата обращения: 12.03.2023).
7. Probabilistic losses. URL: [https://keras.io/api/losses/probabilistic\\_losses/](https://keras.io/api/losses/probabilistic_losses/) (дата обращения: 10.03.2023).
8. Adam. URL: <https://keras.io/api/optimizers/adam/> (дата обращения: 10.03.2023).

УДК 004.4

**М. В. Ильиных**

студент кафедры прикладной информатики

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

**АВТОМАТИЗАЦИЯ РАБОТЫ В ГРАФИЧЕСКИХ РЕДАКТОРАХ**

Даже самые творческие профессии не исключают факт наличия монотонно повторяющихся действий в работе. Человек, заинтересованный в продуктивности своей деятельности, рано или поздно задумывается о том, как бы он мог оптимизировать энергозатраты. Такая необходимость привела к формированию целого подхода по управлению человеческими ресурсами. В его основу лег принцип сокращения человеческого участия и внедрения программных технологий [1].

Со временем вопрос автоматизации встал и в сфере графического дизайна. Использование графических редакторов и их распространенность помогает довольно быстро решить рутинную работу при построении дизайн-макетов [2]. Современные программы имеют достаточное количество инструментов и шаблонов для формирования визитки или презентации, но не всегда такой функционал закрывает все наши потребности. Даже выполнение одного и того же действия, в особенности, последовательного, занимает много времени. В таких ситуациях становится актуально использование сценариев.

Сценарии, они же макросы или скрипты – небольшие программы, выполняющие определенную последовательность действий. Удобство в том, что для написания этих программ можно запустить запись макроса, которая перехватит все выполняемые команды. При необходимости можно вручную дописать макрос или написать свой, если языком программирование это будет проще.

Перечисленные выше возможности сейчас дают такие популярные редакторы как Adobe Illustrator, Corel Draw, Figma и др. В каждой из программ можно записать макросы и запускать их при необходимости. Помимо разницы в самих редакторах, отличия есть и в записи сценариев (табл. 1).

Таблица 1

**Сравнение процесса написания сценариев**

	CorelDRAW [3]	Adobe Illustrator [4]	Figma [5]
Язык хранимых сценариев	VBA (Visual Basic for Application)	JavaScript	JavaScript
Запись сценария через захват действий	+	+	-
Необходимость наличия редактора программного кода	Редактор программного кода встроен в ПО	Необходим редактор программного кода	Необходим редактор программного кода
Возможность редактирования (без вмешательства в программный код)	+	-	-
Сложность использования	низкая	средняя	высокая

Иногда на создание макроса уходит пару минут, а иногда пара недель, в зависимости от его сложности. Но всегда перед написанием собственного большого макроса стоит проверить, нет ли его на просторах всемирной паутины. Многие сценарии есть в свободном доступе, какие-то продаются в магазинах [6]. Есть те, о существовании которых даже не догадываются, но они решают важные проблемы.

К востребованным сценариями автоматизации относят:

1. Макрос Kwart для CorelDRAW, генерирующий сетку квартального календаря на основе заданного шаблона.

2. Макрос Contour для CorelDRAW, создающий контуры для всех выбранных объектов
3. Скрипт Tree-like для Adobe Illustrator, генерирующий деревья.
4. Скрипт Metaball для Adobe Illustrator, с помощью которого круги разного диаметра объединяются акkuratными дугами.
5. Скрипт Points Move Random для Adobe Illustrator, который перемещает в случайном диапазоне точки фигур.
6. Скрипт ExtUngroup для Adobe Illustrator, отменяющий все группы и удаляющий обтравочные маски.

Для наглядности результат запуска макроса Contour представлен на рис. 1. На выполнение макроса для такого количества стикеров ушло 2 секунды, а вот сама программа содержит в себе более 300 строчек кода [7].

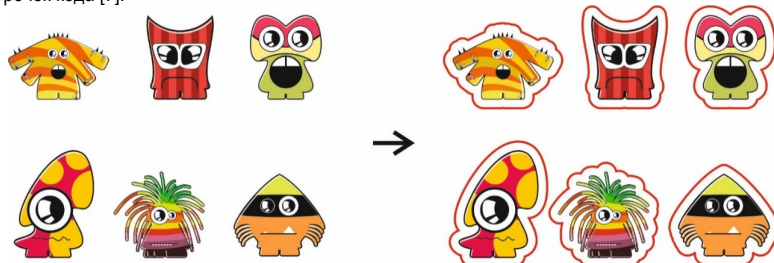


Рис. 1. Добавление контура реза для стикеров при помощи макроса

Просмотреть и отредактировать содержимое можно макроса можно в редакторе макросов (рис. 2). Макрос Contour помимо программного кода на языке VBA содержит в себе библиотеку с классами и функциями, а также две формы, позволяющими задавать настройки отрисовки контура.

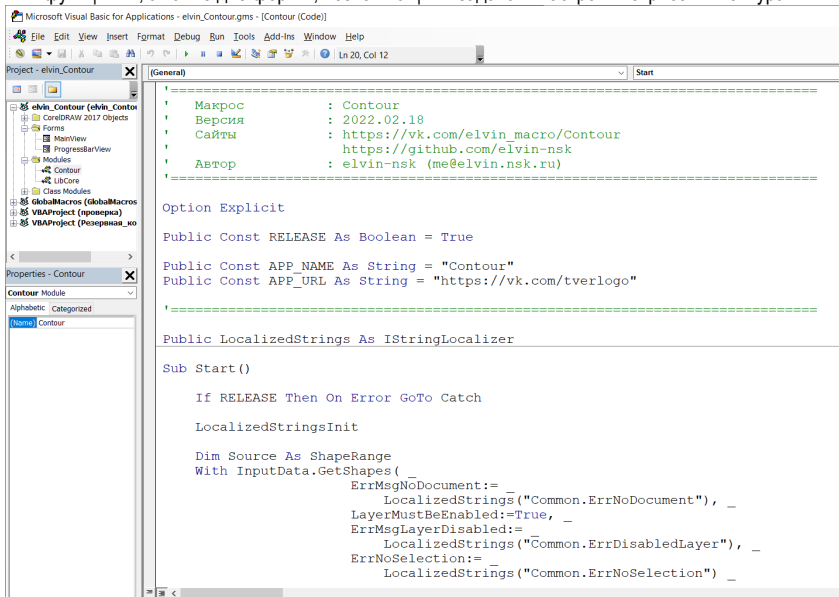


Рис. 2. Редактор макросов в CorelDRAW

Убедиться в необходимости использования сценариев для сокращения времени работы с заказом можно на примере работы в типографии. Допустим у нас есть 17 наклеек различной формы, чтобы собрать их в единый стикерпак, дизайнер потратит 5 минут на раскладку и 40 минут на отрисовку контуров реза для каждой наклейки. Если же запустить макрос по отрисовке контуров, то суммарно дизайнер потратит вместо 45 минут всего 15 минут. В качестве второго примера возьмем календарный блок – листы с месяцами и днями недели, где отмечены выходные и праздничные дни. На формирование такого календаря при отсутствии шаблонов может уйти не один час, однако с подготовленным заранее макросом на это уйдет от 15 до 30 минут.

Такие результаты дают нам понять, что всегда есть возможность упростить задачу. При работе с программным обеспечением необходимо помнить, что любое актуальное в своей сфере ПО будет обновляться, а появление новых инструментов в нем будет оправдано удобством применения и функционалом. Каждый такой механизм в графическом редакторе можно рассматривать как когда-то написанный и не столь популярный макрос, но теперь это уже полноценно работающий официально включенный в руководство инструмент.

### Библиографический список

1. Джон Будро, Равин Джесутасан. Реинжиниринг бизнеса. М.: Альпина Паблишер, 2019. 180 с.
2. Виктор Папанек. Дизайн для реального мира. 6-е изд. М.: Аронов, 2018. 416 с.
3. Комолова Н. В. Самоучитель CorelDRAW 2021. СПб.: БХВ-Петербург, 2022. 432 с.: ил.
4. Тучкевич Е. И. Самоучитель Adobe Illustrator CS6. СПб.: БХВ Петербург, 2014. 352 с.: ил.
5. Александр Окунев. Руководство по Figma. URL: <https://assets.super.so/83bfff20-a177-485b-a5ba-afe3fc16ebf6/files/8cf1c829-3edf-4e15-9d4c-af0868c6055b.pdf> (дата обращения: 10.04.2023).
6. AISCRIPTS: Авторские скрипты для Иллюстратора. URL: <https://ais.sergosokin.ru/page/2>. (дата обращения: 10.04.2023).
7. CorelDRAW Справка: Автоматизация задач с помощью макросов. URL: <http://product.corel.com/help/CorelDRAW/540223850/Main/RU/Documentation/wwhelp/wwhimpl/common/html/wwhelp.htm#href=CorelDRAW-Automate-tasks.html&single=true>. (дата обращения: 09.04.2023).

УДК 004.032.26

**П. В. Кирчевская**

студент кафедры прикладной информатики

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

## НЕЙРОННАЯ СЕТЬ ДЛЯ ОПРЕДЕЛЕНИЯ СТЕПЕНИ ЦВЕТОВОЙ СОЧЕТАЕМОСТИ ЭЛЕМЕНТОВ ОДЕЖДЫ

### Введение

В основе исследования лежит реализация модели нейронной сети, способной различать цветовые решения в одежде человека. Необходимость точно определять цвет выбранных элементов изображения является целеполагающим аспектом продукта, который в свою очередь позволит эффективно и быстро обрабатывать входные данные и на основе полученных результатов выносить вердикт о сочетаемости элементов одежды, предложенной пользователем [1].

### Анализ запросов целевой аудитории

Цвета и различные их комбинации влияют на психику человека, а значит, во многом определяют его поведение. Внешний вид цветовых сочетаний был и остается одним из неотъемлемых пунктов оценки не только образа человека, но и многих продуктов и услуг. Описываемая модель останется актуальным инструментом решения многих задач как бытового плана, так и оформительских работ, предлагая альтернативу готовому шаблону, либо готовые цветовые комбинации, способные облегчить работу графическим дизайнерам, специалистам сферы мод и рядовому потребителю.

### Принцип сравнения и анализа

Удачные сочетания цветов строятся на нескольких принципах, опираясь на круговую цветовую палитру. Первый – цвета соседи. Это различные оттенки, которые будут смотреться всегда хорошо. Второй – треугольник цветов. Случай, когда точки на палитре соединяются в равносторонний треугольник, является самым сочетаемым вариантом. Третий – противоположные оттенки. Причем не только отдельные цвета, но и некоторые диапазоны будут сочетаться.

Для сравнения и анализа соответствия цветов по первому принципу необходимо сначала определить главный цвет в оттенке, затем, при получении нового проверять, является ли тот же цвет главным в оттенке. При анализе значений байтов, определяем лидирующее. Как видно по значениям байтов RGB для истинно зеленого цвета с параметрами (0,256,0), оттеночные цвета зеленого с параметрами (10,142,10) и (146,165,49) будут являться сочетаемыми [2].

Второй принцип сравнения и анализа состоит в том, чтобы сдвигать значения каждого байта RGB кодировки влево или вправо для получения координат равностороннего треугольника. Например, первый цвет у нас будет фиолетовый со следующими параметрами RGB: 179, 5, 243. Затем сдвинем байты влево так, что значение байта  $R$  попадает в  $B$ ,  $G$  в  $R$  и  $B$  в  $G$ . Для получения следующего цвета сделаем то же самое. Для сравнения и анализа по третьему принципу необходимо соблюдение следующего выражения:

$$C = ((|B_R - 255|); (|B_G - 255|); (|B_B - 255|)),$$

где  $C$  – новый оттенок, соответствующий противоположному данному цвету,  $B_R$ ,  $B_G$ ,  $B_B$  соответствуют байтам кодировки RGB.

### Модель

Модель нейронной сети будет функционировать на основе библиотек NumPy, Pandas и OpenCV. Для корректной работы программы и распознавания формата цветовой кодировки внутри модели необходимо определить глобальные переменные, представляющие собой положение цветовой точки по трем координатам, например,  $M3(r; g; b)$  [3].

### Формирование обучающего набора

В основе обучения нейронной сети лежит цветовая модель RGB представляющая собой формат кодировки цветовой информации, популярной в фотографии, телевидении, и компьютерной графике. Данная модель кодировки представляется наиболее удобной так как описывает собой трехмерные координаты радиусом 255 единиц, где точка с координатами  $M1 (0;0;0)$  будет определять отсутствие цвета – т. е. черный, а точка  $M2 (255;255;255)$ , являющаяся наложением красного, синего и зеленого цвета – как белый цвет. Набор данных для обучения модели будет содержать координаты цветов и их названия в формате CSV списка, который по мере роста требований целевой аудитории сможет дополняться и редактироваться. Используемый для обучения список данных уже является базовой кодировкой объектов, поэтому в предварительной обработке данных не нуждается.

### Перспективность исследования

В процессе анализа запросов и требований целевой аудитории была выявлена как практическая, так и информационная ценность проводимого исследования. Перспективы практического использования описываемой идеи заключаются в формировании на ее основе веб-продуктов с перспективой монетизации, мобильных приложений AR-формата, локальных инструментов на базе сторонних приложений [4].

### Библиографический список

1. Иоханнес Иттен. Искусство цвета. М.: Д. Аронов, 2004. С. 4–8.
2. Таблица цветовых кодов RGB. URL: [https://www.rapidtables.org/ru/web/color/RGB\\_Color.html](https://www.rapidtables.org/ru/web/color/RGB_Color.html) (дата обращения: 23.03.2023).
3. Никитин А. А., Лиманова Н. И. Процесс распознавания изображения нейронной сетью // Молодой ученый. 2020. № 47 (337). С. 23–25. URL: <https://moluch.ru/archive/337/75420/> (дата обращения: 17.04.2023).
4. Фаустова К. И. Нейронные сети: применение сегодня и перспективы развития // Территория науки. 2017. № 4. С. 83–87. URL: <https://cyberleninka.ru/article/n/neyronnye-seti-primenenie-segodnya-i-perspektivy-razvitiya/viewer> (дата обращения: 17.04.2023).

УДК 004.942

**А. Н. Коновалов**

студент кафедры аэрокосмических компьютерных и программных систем

**Н. Н. Григорьева** – старший преподаватель – научный руководитель

### ЦИФРОВАЯ РЕАЛИЗАЦИЯ МЕХАНИЗМА ШИФРОВАЛЬНОЙ МАШИНЫ «ЭНИГМА»

«Энигма», по своей сути, механизм замены символов. Имея заранее определенные правила по замене, этот механизм получает на вход один символ, выводит другой, а затем, за счет вращения роторов, изменяет изначальные правила [1, 2]. Для понимания того, как была написана программа необходимо описать, как работает физический оригинал механизма [3, 4] (схема показана на рис. 1):

1. Пользователь нажимает клавишу на клавиатуре «Энигмы», чтобы ввести символ для шифрования.

2. Введенный символ проходит через набор переключателей (коммутационную панель или КП), которые могут менять соответствие между входными и выходными символами. Пользователь может настроить переключатели, чтобы изменить соответствие символов и добавить дополнительный уровень безопасности.

3. Затем символ проходит через несколько вращающихся дисков, называемых роторами, в штатном механизме их 3. Роторы имеют внутренние проводящие контакты, которые меняют символ, когда он проходит через них. После каждого ввода символа один или несколько роторов поворачиваются на одну позицию, изменяя соответствие между входными и выходными символами.

4. После прохождения через все роторы символ достигает отражателя (инвертора), который возвращает символ обратно через роторы, но по другому пути. Это обеспечивает симметричность шифрования и дешифрования в Энигме.

5. Далее сигнал проходит через роторы и КП еще раз в обратном направлении, меняя символ по описанному механизму еще раз.

6. Зашифрованный символ выводится на панели с лампочками, которая соответствует клавиатуре. Оператор записывает зашифрованный символ.

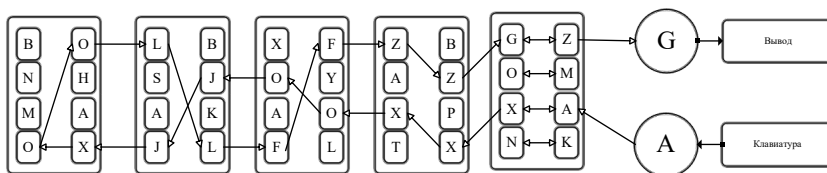


Рис. 1. Схема работы механизма

Алгоритм разработанного программного средства основан на принципе работы оригинального механизма. Ротор – один из основных элементов механизма, поэтому для него был создан класс Rotor:

```
class Rotor:
    def __init__(self, in_: list, out_: list) -> None:
        self.in_ = in_
        self.out_ = out_
    def rotate(self):
        mem = self.in_[0]
        self.in_.remove(mem)
        self.in_.append(mem)
        mem = self.out_[0]
        self.out_.remove(mem)
        self.out_.append(mem)
```

У объектов класса Rotor есть массивы входных и выходных символов (работают как контактные группы на оригинальном аппарате) и метод, позволяющий ротору вращаться.

Стоит отметить, что инвертор и КП тоже являются объектами класса Rotor, но в отличие от самих роторов (в которых массив входных символов случаен относительно выходных) в КП выходной массив зеркален входному, чтобы имитировать работу штатного механизма, а для инвертора было принято решение повторить реализацию КП, чтобы однозначно обеспечить зеркальную кодировку и исключить ситуации, невозможные на оригинальном аппарате.

Настройка кодирующего устройства в программном реализации происходит посредством чтения из файла строки, содержащей расположение роторов (rotor\_setup), а затем – строки, содержащей их начальное вращение (rotor\_start\_rotation):

```
# file reading preparations
file = open('Enigma_code/launch codes.txt', 'r')
file_line = 1
for line in file:
    if file_line == 1:
        rotor_setup = line
        file_line += 1
    else:
        rotor_start_rotation = line
file.close()
```

После того, как программа получила настройки, происходит определение всех переменных, с которыми будет происходить работа в зависимости от того, какими были настройки. Затем подключается секция ввода, в которую встроена проверка, чтобы пользователь не смог ввести недопустимую последовательность символов (так как алфавит ограничен). Над введенной строкой происходит процесс кодирования/декодирования, описанный выше (результаты приведены на рис. 2).

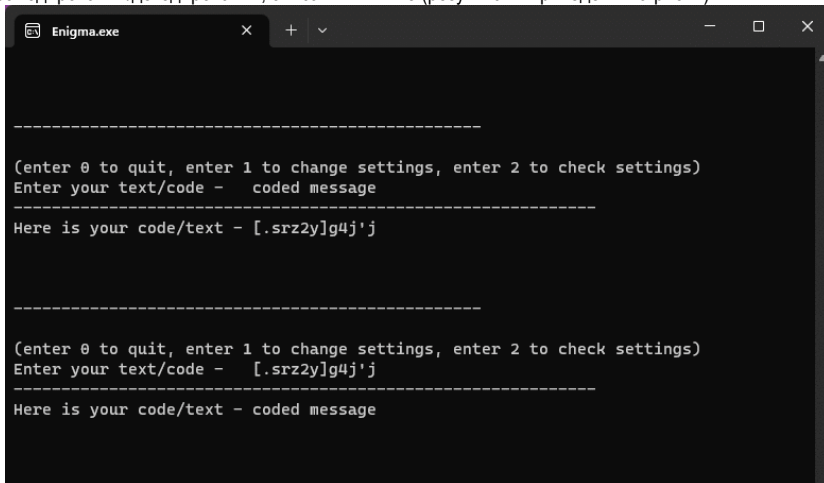
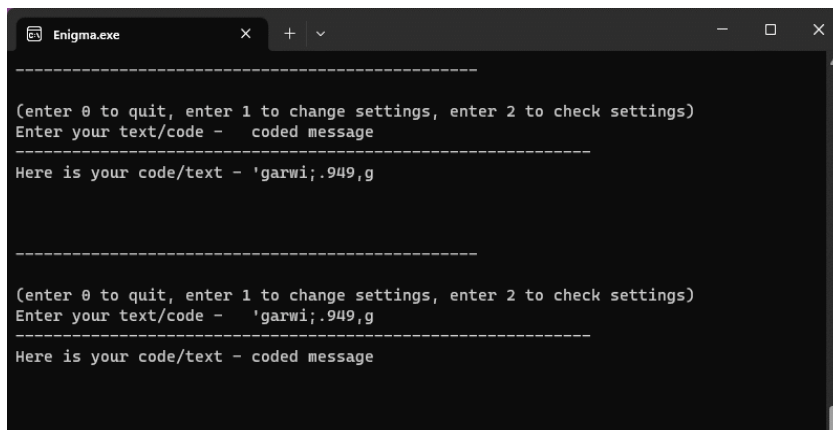


Рис. 2. Результаты кодирования и декодирования

Процесс работы программы идентичен оригинальному алгоритму за исключением моих модификаций: алфавит расширен с 26 до 48 символов (включены знаки пунктуации, скобки и некоторые другие небуквенные символы), количество роторов увеличено с трех до пяти. Помимо увеличения количества роторов в механизме, было также увеличено общее их количество (с 5 до 9), а также добавлена возможность использовать несколько одинаковых роторов для кодирования (что не предусмотрено в оригинальном механизме).







```
-----  
(enter 0 to quit, enter 1 to change settings, enter 2 to check settings)  
Enter your text/code - 'garwi;.949,g  
-----  
Here is your code/text - 'garwi;.949,g  
-----  
  
-----  
(enter 0 to quit, enter 1 to change settings, enter 2 to check settings)  
Enter your text/code - 'garwi;.949,g  
-----  
Here is your code/text - coded message
```

*Рис. 6. Результаты кодирования и декодирования при новых настройках машины*

Проведенная работа по моделированию алгоритма работы шифровальной машины «Энигма» и проверка функционирования программного средства показала адекватность созданной модели оригинальному механизму.

#### **Библиографический список**

1. Марк Лутц. Изучаем Python. СПб.: Символ-Плюс, 2009. 848 с.
2. Python 3.11.3 documentation. URL: <https://docs.python.org/3/> (дата обращения: 07.05.2023).
3. Майкл Керриган. Взламывая код "Энигмы". Как расшифрованный код помог выиграть Вторую мировую войну. АСТ, 2021. 240 с.
4. Алгоритм Энигмы. URL: <https://habr.com/ru/articles/217331/> (дата обращения: 07.05.2023).

УДК 004.42

**Д. В. Коптев**

студент кафедры прикладной информатики

**Н. Н. Григорьева** – старший преподаватель – научный руководитель

### СОЗДАНИЕ ВЕБ-ПРИЛОЖЕНИЯ «ПОМОЩНИК СТУДЕНТА»

Веб-приложение – клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети [1, 2]. Одним из преимуществ такого подхода является то, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными службами. В данной статье мы будем рассматривать создание приложения с подключенной базой данных.

Рассмотрим концепцию приложения. Его задача – формирование очереди студентов на сдачу учебных работ, эта задача актуальна в группах большой численности. Пользователь выбирает группу, тип сортировки и предмет. Далее появляется таблица студентов в очереди с количеством сданных ими работ. Таблица формируется по заданным критериям и поэтому итоговый продукт способен подойти различным пользователям с различными предпочтениями. В приложении предполагалось наличие функционала изменения таблицы из интерфейса приложения. Пользователь может изменить количество сданных работ, а также показатель “приоритет”, который служит вспомогательным параметром.

С точки зрения пользователя приложение состоит из нескольких разделов. Первый раздел называется “Объявления” (рис. 1). Он содержит ссылки на остальные разделы, приветственную информацию и секцию объявлений. В низу страницы располагаются контакты для обратной связи.

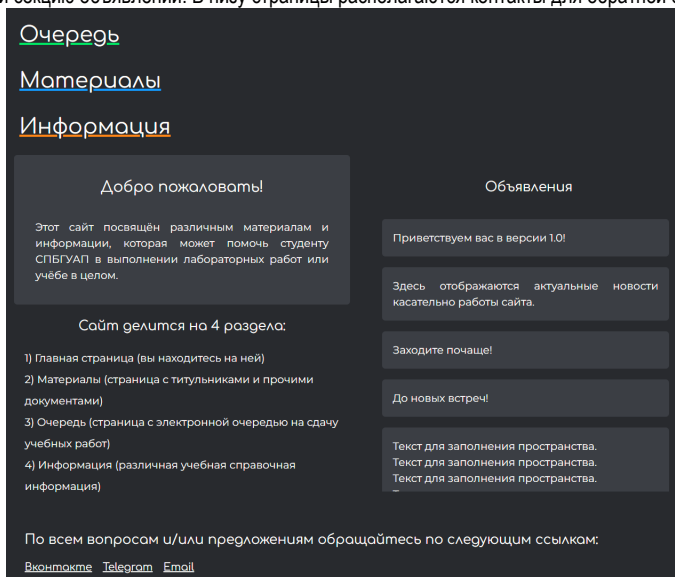


Рис. 1. Вид страницы “Объявления”

Раздел “Очередь” содержит форму выбора группы, типа сортировки и учебной дисциплины, дальше на странице располагается собственно таблица с кнопками экспорта и перехода в режим администратора (рис. 2 и 3).

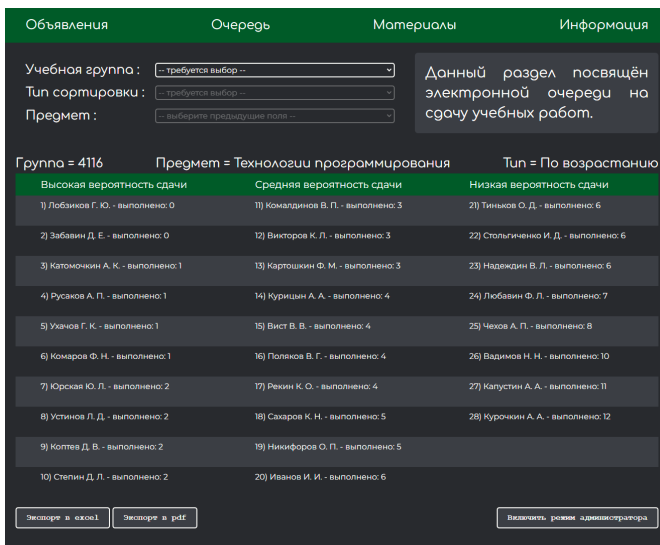


Рис. 2. Страница «Очередь» (в режиме просмотра)

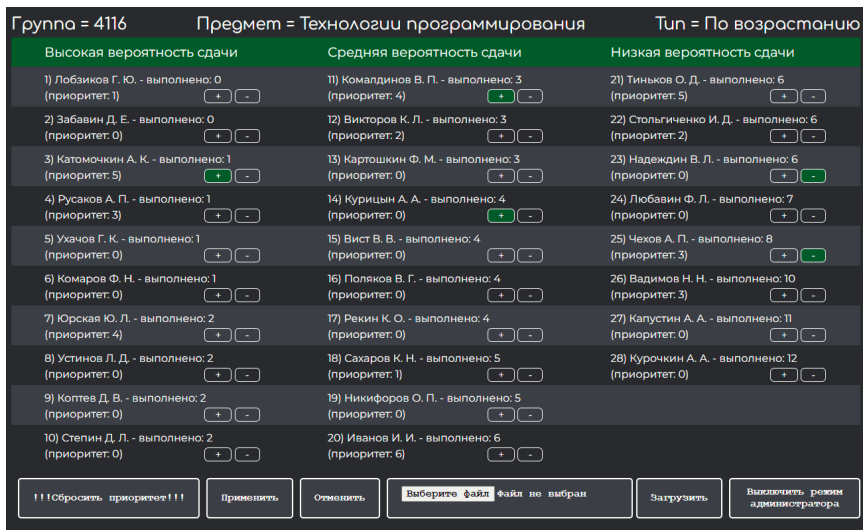


Рис. 3. Страница «Очередь» (в режиме редактирования)

Раздел “Материалы” содержит форму выбора семестра, дисциплины и преподавателя, а также данные, сформированные запросом, содержащие файлы титульных листов, различных учебных материалов и контакты для связи с преподавателем (рис. 4).

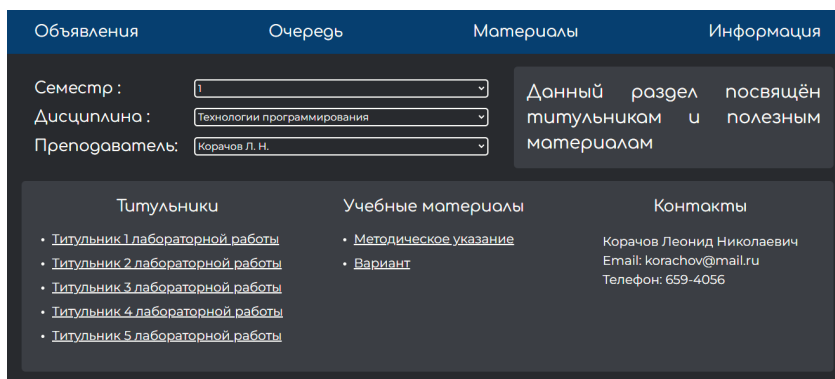


Рис. 4. Страница "Материалы"

Раздел приложения "Информация" содержит полезные справочные данные. В левой части находится секция с местоположениями мест для печати. В правой части находится раздел с наименованием институтов и кафедр (рис. 5).

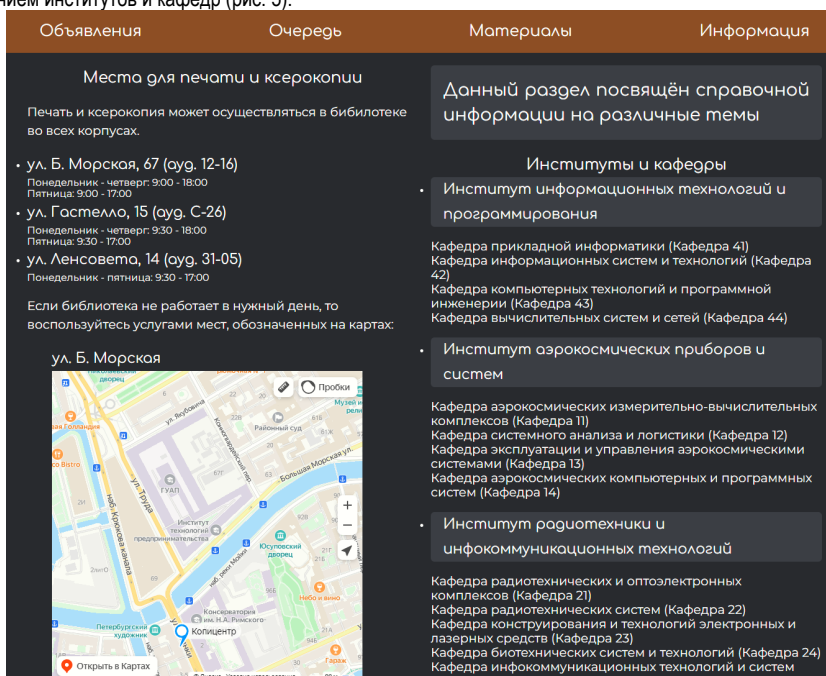


Рис. 5. Страница "Информация"

Принцип работы раздела "Очередь" показан на рис. 6.



Рис. 6. Схема взаимодействия пользователя, сервера и базы данных

Порядок формирования очереди рассмотрим на примере с типом сортировки «по возрасту». Первыми в такой очереди пойдут студенты с наименьшим количеством работ, если у студентов одинаковое количество работ, то выбирается тот, у кого наибольший приоритет (приоритет – это число, к которому прибавляется единица, каждый раз, когда другой студент успешно сдает работу). Если и приоритет одинаковый, то сортировка идет по индексу студента. Данные для базы данных заполняются администратором либо через интерфейс СУБД, либо через загрузку текстового файла с ФИО студентов через интерфейс раздела администратора (рис. 7).

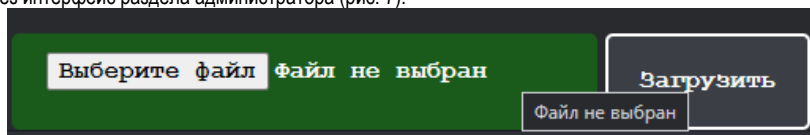


Рис. 7. Интерфейс импорта ФИО студентов через текстовый файл

В данной работе была сделана попытка решить проблему организации очереди при сдаче заданий на учебных занятиях. Конечно, пока разрабатываемое приложение далеко от идеала, его необходимо дорабатывать, но существующая базовая версия приложения уже способна выполнять требуемые функции.

Ссылка на репозиторий github с проектом: <https://github.com/DmitriiKoptev/flask>.

Ссылка на задеплойненную версию сайта: <https://dmitrii-koptev1.onrender.com>.

#### Библиографический список

1. Miguel Grinberg, Flask Web Development. O'Reilly Media, Inc., 2014. 274 с.
2. David Flanagan, JavaScript. The Definitive Guide, 7th Edition. O'Reilly Media, Inc., 2020. 1080 с.

УДК 004.032.26

**М. М. Кулешо**

студент кафедры прикладной информатики

**Е. Л. Турнецкая** – кандидат технических наук – научный руководитель**ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТЕЙ В DIGITAL-МАРКЕТИНГЕ**

По данным исследования в журнале *Proceedings of the National Academy of Science (PNAS)*, в ходе которого ученые опросили 4500 взрослых людей, выяснилось, что 28% опрошенных, совершавших покупки на продукты и услуги, позволяющие сэкономить время, были более довольны своей жизнью, независимо от уровня доходов. В числе их покупок были в том числе и покупки, связанные с новыми технологиями, такими как «умный дом»; покупки, связанные с электронной коммерцией и т.д.

Стремясь улучшить свою жизнь, люди обращают внимание не только на личные сферы, такие как дом, семья, отдых, но и профессиональную деятельность. В бизнесе новые технологии применяются скорее не с целью повышения эмоционального состояния сотрудников, а больше с целью повышения эффективности работы, экономия времени и, следовательно, понижения затрат компании. Так, например, в 2022 году стартовал проект по проверке сочинений школьников нейросетью. И в результате проверок выяснилось, что алгоритм фиксирует ошибки лучше, чем среднестатистический преподаватель-эксперт, а также программа находит грамматические, пунктуационные и смысловые ошибки. Ожидается, что нейросеть сэкономит учителям до 20% времени [1].

Нейросеть – это тип машинного обучения, при котором компьютерная программа имитирует работу человеческого мозга. Подобно тому, как нейроны в мозге передают сигналы друг другу, в нейросети информацией обмениваются вычислительные элементы [2].

Нейросети решают задачи распознавания образов и классификации, принятия решений и управления, кластеризации, прогнозирования, аппроксимации, сжатия данных и ассоциативной памяти, анализа данных и оптимизации. Если говорить более простым языком, то нейросети могут выполнять как простые действия, начиная от поиска информации в Интернете, до сложных, например, распознавания речи и звуков.

Технологию нейронных сетей применяют в упрощении задач многих профессий. Нейросеть уже используют в медицине: для того, чтобы облегчить рутинную работу врачей. Компьютер быстрее и точнее обнаруживает отклонения от нормы. Программы помогают находить на КТ (компьютерная томография) признаки рака легких, COVID-19, инсульта. Кстати, в поликлиниках Москвы в рамках ОМС (обязательное медицинское страхование) с помощью нейросети будут анализировать рентгеновские снимки [3].

Одно из востребованных мест применения нейросетей в последние годы – мир digital-маркетинга.

Digital-marketing – это направление маркетинга, которое подразумевает продвижение услуг и товаров с помощью цифровых технологий, применяемых на всех этапах взаимодействия с потребителями. От интернет-маркетинга он отличается тем, что использует не только Всемирную паутину, но и офлайн-инструменты (smart-гаджеты, POS-терминалы и др.) [4]. Например, когда зрителю в онлайн-кинотеатре предлагают перейти в соцсети, чтобы поучаствовать в конкурсе, или отсканировать QR-код и выиграть приз.

В digital-маркетинге нейросети применяют для подбора тем для статей, написания заголовков и рекламных текстов, генерации картинок и фотографий, создания анимации и даже ответов пользователям.

Одним из ярких примеров работы нейросети являются терминалы самообслуживания, которые стоят в некоторых сетевых точках фастфуда – таких как McDonald's и KFC. Когда пользователь делает заказ нейросеть предлагает ему дополнительно блюда, напитки и десерты, которые пользователь мог бы заказать дополнительно, исходя из того заказа, который он уже сделал. Например, пользователь приобретает кофе и нейросеть предлагает ему приобрести десерт [5].

И в 2023 году одними из наиболее популярных сервисов с применением технологий нейросетей стали ChatGPT, DALL-E 2 и Midjourney.

1. ChatGPT (<https://chat.openai.com/>). ChatGPT – это чат-бот с применением искусственного интеллекта. Он умеет работать в диалоге и поддерживает запросы на естественных языках [6]. Сервис способен отвечать на вопросы развернутыми ответами; искать информацию почти как любой поисковик; писать забавные диалоги и рассказы; составлять электронные письма и метатеги; разрабатывать простые приложения.

С помощью этого чат-бота работники сферы digital-маркетинга могут решать задачи по генерации ответов на комментарии пользователей, написанию лонгридов (статей) и постов для социальных сетей, созданию текстовых наполнений для сайтов, описаний товаров и многое другое. Благодаря данной технологии можно даже генерировать контент-планы, что во многом упрощает работу smm-специалистам.

Стоимость использования такого сервиса бесплатная, требуется только регистрация и авторизация, а также верификация по номеру мобильного телефона.

2. DALL·E 2 (<https://labs.openai.com/>). DALL·E 2 – это сервис, который используется в основном для генерации изображений по текстовому запросу или наброску, однако с помощью него также можно вносить невероятно реалистичные изменения в уже существующие фото и иллюстрации. Нейросеть принимает во внимание тени, отражения и текстуры.

Эта технология помогает создавать готовый контент для дизайнеров, креаторов и smm-социалистов, например, во избежание использования чужих изображений и иллюстраций.

Стоимость использования данного сервиса также бесплатная, но, как и в случае с чатом GPT, требуется регистрация и авторизация, а также верификация по номеру мобильного телефона.

3. Midjourney (<https://www.midjourney.com/home/>). Midjourney – как и в предыдущем случае, данная нейросеть предназначена для генерации изображений по текстовым запросам. Она создает четыре варианта изображений, после чего можно создать их дополнительные вариации или сделать масштабирование понравившегося варианта и сохранить себе итоговое изображение в хорошем качестве.

К сожалению, процесс работы с данной нейросетью немного затруднен, из-за сложного процесса регистрации и реализации запросов, но несмотря на это, она все равно пользуется большой популярностью среди дизайнеров и других специалистов сферы digital-маркетинга.

Помимо этих трех популярных сервисов, специалисты также пользуются многими другими полезными разработками. Вот несколько из них:

1. Балабоба (<https://yandex.ru/lab/yalm/>) – отечественный сервис от компании «Яндекс», который по запросу генерирует текст в выбранной стилистике. Сервис доступен для бесплатного использования без предварительной регистрации и авторизации. Благодаря этому сервису во многом упрощается работа копирайтеров – специалистов по написанию текстов: их задача теперь не написать текст полностью с нуля, а взять готовый текст, вдохнуть в него жизнь, добавить индивидуальной стилистики и тона компании.

2. RemoveBg (<https://www.removebg.dev/>) – нейросеть для удаления фона с фото, принцип взаимодействия с которой невероятно прост: нужно лишь загрузить фото на сайт и в течении нескольких секунд будет получен результат. Благодаря этому сервису упрощается работа дизайнеров, которые часами в фотошопе удаляли фон с фотографий клиентов – теперь эта задача решается в два клика.

3. Colorize (<https://colorize.cc/>) – сервис для раскрашивания и реставрации старых черно-белых фотографий. Его недостатком является только то, что он полностью платный, однако тестировать, как нейросеть справится с фото перед покупкой можно сколько угодно. Сервис можно использовать в различных целях, но один из основных – раскрашивание фото с целью передачи атмосферы эпохи, например, на праздники, связанные с военными событиями 20 и ранее веков.

4. Movavi (<https://www.movavi.ru/photo-editor/>) – фоторедактор, работающий на основе нейронных сетей. Функционал содержит все необходимые базовые инструменты для обработки фотографий. Искусственный интеллект в приложении включен в задачи ретуши, восстановления старых фото, удаления фона или объектов, а также в функцию интеллектуального автоулучшения, которая позволяет автоматизировать рутинные задачи по цветокоррекции и значительно улучшает качество изображения.

5. Riffusion (<https://www.riffusion.com/>) – нейросеть для создания музыки, с помощью которой можно сделать что-то фантастическое, а можно получить произведение в более реалистичном любии-



мом жанре. Работает прекрасно и выдает интересные музыкальные решения, но вот скачать получится лишь пятисекундный фрагмент. Подойдет для создателей аудио- и видеоконтента.

И это только малая часть сервисов, которыми пользуются специалисты digital-маркетинга и не только. Популярность нейросетей растет ежедневно. По данным аналитиков компании «Tele2», на фоне популярности Midjourney и ChatGPT с августа 2022-го по февраль 2023 г. количество пользователей нейросетей выросло в пять раз, а продолжительность взаимодействия с искусственным интеллектом (ИИ) – в три раза [7].

#### Библиографический список

1. Аксентов В. А. Использование искусственного интеллекта в образовании. URL: <https://cyberleninka.ru/article/n/ispolzovanie-iskusstvennogo-intellekta-v-obrazovanii/viewer> (дата обращения: 16.04.2023).
2. Нейросеть: что это, как работает и как применяется. URL: <https://trends.rbc.ru/trends/industry/641157be9a7947d3401fa3e8> (дата обращения: 14.04.2023).
3. Вкалывают роботы, а не человек. Что такое нейросети и зачем они нужны? URL: <https://dzen.ru/a/Y93N7VeUuQp1EM5b> (дата обращения: 14.04.2023).
4. Digital-маркетинг: что это такое и как работает диджитал маркетинг в Интернете. URL: <https://blog.ingate.ru/seo-wikipedia/digital-marketing/> (дата обращения: 14.04.2023).
5. Беккулиева Б. М., Разакова Д. И. Особенности применения нейротехнологий в маркетинге. URL: <https://cyberleninka.ru/article/n/osobennosti-primeneniya-neyrotehnologiy-v-marketinge/viewer> (дата обращения: 16.04.2023).
6. Обзор чат-бота ChatGPT: что это, возможности и примеры использования. URL: <https://habr.com/ru/companies/mtt/articles/711052/> (дата обращения: 14.04.2023).
7. Число пользователей нейросетей в России выросло в пять раз за полгода. URL: <https://www.vedomosti.ru/technology/articles/2023/03/01/964683-chislo-polzovatelei-neirosetei-v-rossii-viroslo> (дата обращения: 14.04.2023).

УДК 004

**И. Ю. Летунов**

студент кафедры прикладной информатики

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

## ОРГАНИЗАЦИЯ КЛИЕНТ-СЕРВЕРНОГО ВЗАИМОДЕЙСТВИЯ В АВТОМАТИЧЕСКОЙ СИСТЕМЕ ТЕХНИЧЕСКОЙ ДИАГНОСТИКИ МОСТИКОВЫХ СИСТЕМ СУДНА

Автоматизированная система технической диагностики (АСТД) является составной частью программного обеспечения корабля и предназначена для контроля состояния оборудования и систем корабля [1].

АСТД обеспечивает автоматический мониторинг состояния входящих в Интегрированные Мостиковые Системы (ИМС) технических средств, отображение актуального состояния технических средств на многоуровневых мнемосхемах с детализацией до конкретного устройства и его расположения, предоставление информации о способах устранения конкретной неисправности и необходимом для этого оборудовании, ведение архива неисправностей и хода их устранения, включая добавление комментариев исполнителей о ходе устранения, работу с архивом устраненных неисправностей и списком текущих неисправностей [2].

### Взаимодействие клиента с сервером

Информационное взаимодействие клиента АСТД с сервером должно происходить путем обмена сообщениями установленного формата через систему сообщений MB. Формат сообщения определяется его типом [2, 3].

Обмен сообщений происходит посредством записи и чтения их в очереди. Для приема и/или отправки каждого типа сообщений в Message Broker (MB) предназначены свои поименованные очереди и обменники (exchanges), подключение к которым происходит подпиской через вызовы API mbTransport. Для подписки на получение сообщений из определенной очереди или отправку сообщений в очередь и обменник (exchanges) необходимо знать и указать имя очереди обменника (через API mbTransport).

Каждое сообщение состоит из стандартного заголовка сообщения и тела сообщения (табл. 1).

Таблица 1

### Структура сообщения

Название	Тип	Описание
identity	string	Уникальный на время сессии ID отправителя сообщений (uuid -> string)
sourceID	string	ID источника сообщения
targetID	string	ID адресата сообщения (опционально)
msgType	uint16	Тип сообщения, т.е. фактически тип (формат) структуры данных, находящейся в последующем блоке данных msgData в формате «serialized JSON»
msgSubtype	uint16	Подтип сообщения, более точно специфицирующий сообщение и формат его данных (опционально)
timestamp	uint32_t	Дата/время отправки сообщения (UTC, сек)
msgData	string	Блок данных (содержание сообщения) в формате «serialized JSON»

Заголовок сообщения содержит несколько числовых и текстовых полей, в том числе ID источника (sourceID) и адресата (targetID) сообщения, тип (msgType) и подтип (msgSubtype) сообщения. Тело

сообщения (msgData) всегда представляет собой строку в формате JSON, структура которой однозначно зависит от значений полей заголовка сообщения.

В таблице два приведен перечень типов сообщений при взаимодействии клиента с сервером.

Таблица 2

**Типы сообщений**

Название	Код	Описание	Имя очереди по умолчанию
Command	1	Команда клиентом о квитировании или добавлении комментария к неисправности	std-requests
AnswerToCommand	13	Ответ сервера о результате выполнения команды.	std-data
DataRequest	2	Запрос клиентом каких-либо данных от сервера	std-requests
AnswerToRequest	3	Ответ сервера на ранее сделанный запрос	std-data
Faults	5	Сообщение о выработанных/обновленных неисправностях	std-data

Команда отправляется клиентом серверу. Для этого используется сообщение типа Command (msgType = Command), поле targetID должно содержать идентификатор сервера (см п.1), поле sourceID должно содержать идентификатор клиента (см п. 1), поле msgSubtype не используется, формат поля msgData приведен в табл. 3.

Таблица 3

**Параметры команды от клиента**

Название	Тип	Описание
MsgID	string	Уникальный ID команды (uuid -> string). Используется для ответа на данный запрос, чтобы организовать пару «команда-ответ о результате выполнения».
Code	uint32	Тип команды
CmdData	json-объект	Параметры команды, JSON. Формат параметров будет отличаться для разных типов команд

Пример команды:

```

"identity": "",
"sourceID": "STD.Client.ArmK",
"targetID": "STD",
"timestamp": 1579774766,
"msgType": 1,
"msgData": {
  "MsgID": "23a327fc-c570-4077-bcfc-3fe3715a814a",
  "Code": 1,
  "CmdData": {
    "sourceID": "DEVICE00005",
    "eventDbLogID": "f144e650-e634-4aa0-8d90-d45cb8066056"
  }
}
    
```

Ответ на команду вырабатывается сервером и отправляется клиенту. Сообщения ответа имеют тип AnswerToCommand (msgType = AnswerToCommand), поле msgSubtype не используется, формат поля msgData приведен в табл. 4.

## Параметры ответа на команду

Название	Тип	Описание
MsgID	string	Уникальный ID команды, на которую сформирован данный ответ
Code	uint32	Тип команды
ErrorCode	uint32	Код ошибки, возникшей на сервере во время выполнения команды.

Пример ответа:

```
"identity": "",
"sourceID": "STD",
"targetID": "STD.Client.ArmK",
"timestamp": 1579774766,
"msgType": 13,
"msgData": {
  "MsgID": "23a327fc-c570-4077-bcfc-3fe3715a814a",
  "Code": 1,
  "ErrorCode": 0
}
```

Запрос отправляется клиентом серверу. Для запроса данных используется сообщение типа DataRequest (msgType = DataRequest), поле targetID должно содержать идентификатор сервера (см п.1), поле sourceID должно содержать идентификатор клиента (см п.1), поле msgSubtype не используется, формат поля msgData приведен в табл. 5.

## Параметры запроса данных клиентом

Название	Тип	Описание
RequestID	string	Уникальный ID запроса (uuid -> string). Используется для ответа на данный запрос, чтобы организовать пару «запрос-ответ»
RequestType	uint32	Тип запрашиваемой информации
RequestData	json-объект	Параметры запроса, JSON. Формат параметров запроса будет отличаться для разных типов запросов

Пример запроса:

```
"identity": "",
"sourceID": "STD.Client.ArmK",
"targetID": "STD",
"timestamp": 1579774766,
"msgType": 2,
"msgData": {
  "RequestID": "23a327fc-c570-4077-bcfc-3fe3715a814a",
  "RequestType": 1,
  "RequestData": {
    "eventDbLogID": "f144e650-e634-4aa0-8d90-d45cb8066056"
  }
}
```

Ответ на запрос данных вырабатывается сервером и отправляется клиенту в ответ на его запрос. Сообщения ответа на запрос данных имеют тип AnswerToRequest (msgType = AnswerToRequest), поле msgSubtype не используется, формат поля msgData приведен в табл. 6.

Параметры ответа на запрос данных

Название	Тип	Описание
RequestID	string	Уникальный ID запроса, на который сформирован данный ответ
RequestType	uint32	Тип запроса (см. таблицу 2.2)
ErrorCode	uint32	Код ошибки, возникшей на сервере во время выполнения команды. См. таблицу 4.1
AnswerData	json-объект	Ответ на запрос, JSON. Формат полностью зависит от типа запроса. Если подсистема не может выдать данные по команде, в ответе поле AnswerData отсутствует.

Пример ответа:

```
"identity": "",
"sourceID": "STD",
"targetID": "STD.Client.ArmK",
"timestamp": 1579774766,
"msgType": 3,
"msgData": {
  "RequestID": "23a327fc-c570-4077-bcfc-3fe3715a814a",
  "RequestType": 1,
  "ErrorCode": 0,
  "AnswerData": {
    "alarmDesc": {
      "msgID": "aa977b97-beef-4bc8-b508-eedc868ff2d4",
      "eventID": "id.arm1.yellow2",
      "subsystemID": "device ID",
      "eventSourceID": "elm_key",
      "state": 0,
      "eventDbLogID": "f144e650-e634-4aa0-8d90-d45c",
      "alertType": 2,
      "closeTimestamp": 1577442745,
      "msgTimestamp": 1577360834,
      "eventTimestamp": 1577442745
    },
    "alarmChanges": [
      {
        "timestamp": 1577442745,
        "text": "Значение устарело",
        "sourceID": "STD"
      },
      {
        "timestamp": 1577360834,
        "text": "Случилась неисправность",
        "sourceID": "DEVICE00056"
      }
    ]
  }
}
```

Сообщения о выработанных/обновленных неисправностях (msgType = Faults) вырабатываются:

- сервером при возникновении соответствующих событий;
- сервером в качестве ответа на запрос клиента.

Может отправляться как сообщение об одной неисправности, так и сообщение с несколькими неисправностями.

Поле msgSubtype не используется, формат поля msgData приведен в табл. 7.

Таблица 7

**Параметры сообщения о неисправности**

Название	Тип	Описание
msgID	string	Уникальный ID данного сообщения о событии (uuid -> string), опц.
eventID	string	Глобальный идентификатор типа события (тип – пожар, дым, проникновение в помещение под охраной и т.д., отказ судового оборудования)
latitude	double	Широта (град) фиксации события.
longitude	double	Долгота (град.) фиксации события.
subsystemID	string	Уникальный ID типа бортовой системы, породившей неисправности. Например, пожарная или охранная сигнализация.
eventSourceID	string	Уникальный ID конкретного источника, породившего неисправность. Например, ID датчика, модуля, узла.
ackSourceID	string	Источник квитирования события, указывается сервером для квитированных неисправностей.
lastComment	string	Текст последнего комментария.
state	uint16	Состояние события: Неактуальная не квитированная (0) Актуальная не квитированная (1) Неактуальная квитированная (2) Актуальная квитированная (3) Неизвестное состояние(65535)
eventDbLogID	string	Идентификатор записи в журнале событий неисправностей в БД
alertType	uint16	Тип сообщения о событии
msgTimestamp	uint32	Временная метка создания исходного сообщения о событии (сек).
closeTimestamp	uint32	Временная метка закрытия события (сек).
eventTimestamp	uint32	Временная метка записи о событии в БД (создания/обновления неисправности/комментария), UTC.
ackTimestamp	uint32	Дата/время квитирования неисправности, UTC. Задается клиентом, при отправке статуса квитирования, в качестве источника времени используются системные часы.

### **Заключение**

Таким образом, интеграция между клиентской и серверной части АСТД формируется на основе передачи данных в формате JSON. При осуществлении взаимодействия требуется понимание структуры передаваемых данных. В статье показаны основные типы команд и параметров передаваемых сообщений

### **Библиографический список**

1. Интегрированная мостиковая система. URL: <https://npfmeridian.ru/> (дата обращения: 21.05.2023).
2. Лепский А. Г., Цветков В. В., Щеглов А. А. Основные направления развития систем комплексной автоматизации на флоте // Вестник МГТУ. 2004. № 3. URL: <https://cyberleninka.ru/article/n/osnovnyie-napravleniya-razvitiya-sistem-kompleksnoy-avtomatizatsii-na-flote> (дата обращения: 21.05.2023).
3. Сорокин В. Диагностирование судовых технических средств // Морские вести России. 2018. URL: <https://www.morvesti.ru/themes/1699/73980/> (дата обращения: 21.05.2023).

УДК 004

**И. Ю. Летунов**

студент кафедры прикладной информатики

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

### КОМПОНЕНТЫ АВТОМАТИЧЕСКОЙ СИСТЕМЫ ТЕХНИЧЕСКОЙ ДИАГНОСТИКИ МОСТИКОВЫХ СИСТЕМ СУДНА

Автоматизированная система технической диагностики (АСТД) является составной частью программного обеспечения корабля и предназначена для контроля состояния оборудования и систем корабля [1].

АСТД обеспечивает автоматический мониторинг состояния входящих в Интегрированные мостиковые системы (ИМС) технических средств, отображение актуального состояния технических средств на многоуровневых мнемосхемах с детализацией до конкретного устройства и его расположения, предоставление информации о способах устранения конкретной неисправности и необходимом для этого оборудовании, ведение архива неисправностей и хода их устранения, включая добавление комментариев исполнителей о ходе устранения, работу с архивом устраненных неисправностей и списком текущих неисправностей [2].

Для покрытия всех потребностей выбрана клиент-серверная архитектура из-за лучшей масштабируемости и имеет большую безопасность. В данном типе архитектуры можно добавить новых клиентов или серверы в систему, что позволит легко масштабировать приложение под изменяющиеся потребности. Также клиент-серверная архитектура обеспечивает более высокий уровень безопасности, поскольку клиент не имеет прямого доступа к базе данных и другим ресурсам сервера, что может быть защищено различными механизмами аутентификации и авторизации.

Весь требуемый функционал АСТД распределяется между подсистемой АСТД серверного программного обеспечения (ПО) ИМС, базой данных, клиентскими приложениями АСТД [2, 3].

Серверная подсистема АСТД структурно состоит из ядра подсистемы, постоянно присутствующего в составе серверного ПО и активизируемого при наличии соответствующей части конфигурации, и набора динамически подключаемых плагинов, провайдеров параметров АСТД. Провайдеры параметров АСТД обеспечивают выработку и передачу в ядро подсистемы первичных параметров АСТД, непосредственно взаимодействуя с контролируемыми техническими средствами или принимая сообщения с параметрами АСТД от Message Broker (МВ), и, опционально, обеспечивают прием в провайдер из ядра выработанных вторичных параметров АСТД.

Вторичными параметрами являются параметры, чьи значения были вычислены на основе значений других параметров. Первичные параметры – значения сигналов, поступающих непосредственно от оборудования или определенных программно, например, команды ping.

Рассмотрим подробнее компоненты серверной и клиентской частей АСТД.

#### Серверная часть АСТД

В сервере АСТД реализован интерфейс обмена через RabbitMQ в формате Json. В основном режиме работы сервер АСТД выполняет следующие функции:

- подключается к БД с заданными в конфигурации сетевыми параметрами подключения и авторизации;
  - в случае недоступности БД повторяет попытки подключения к ней;
  - считывает конфигурацию из БД;
  - определяет и принимает состояние мастер или резерв;
  - только сервер мастер:
    - выполняет мониторинг сетевой доступности устройств;
    - опрашивает контроллеры ввода вывода и выполняет требуемые записи в регистры контроллеров в соответствии с конфигурацией;
    - проводит расчет значений параметров в процессоре правил;



- открывает порт для подключения клиентов АСТД;
  - обеспечивает обмен данными с клиентами;
  - генерирует и записывает сообщения о неисправностях в БД;
- В режиме конфигурирования сервер АСТД выполняет функции:
- подключаться к БД;
  - подключать только один клиент-редактор для формирования конфигурации АСТД;
  - считывать и сохранять параметры конфигурации в БД;
  - подключаться к источникам данных.
- Схема связей таблиц БД ASTDBase представлена на рис. 1

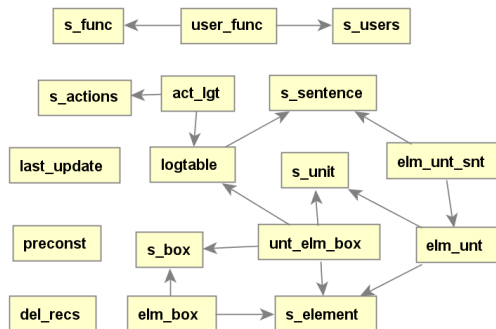


Рис. 1. Модель данных БД ASTDBase

PRECONST – Постоянно действующие настройки.

LAST\_UPDATE – дата-время последнего изменения записей в таблицах.

S\_USERS – пользователи.

S\_FUNC – справочник функций доступа.

USER\_FUNC – функции пользователей.

S\_SENTENCE – сообщения о неисправностях.

S\_UNIT – приборы.

S\_ELEMENT – типовые элементы замены.

S\_BOX – ящики ЗИП.

S\_ACTIONS – этапы действий по устранению неисправности.

ELM\_UNT – ТЭЗы в приборах.

ELM\_UNT\_SNT – ссылки на ТЭЗы в приборах из предложений о неисправностях.

ELM\_BOX – ТЭЗы в ящиках ЗИПа.

UNT\_ELM\_BOX – ссылки на прибор, ТЭЗ и ЗИП-ящик из журнала неисправностей (для сохранения истории).

LOGTABLE – журнал регистрации неисправностей.

ACT\_LGT – этапы действий по устранению конкретной неисправности.

### Клиентская часть АСТД

Клиент АСТД после запуска или потери связи с сервером АСТД, должен автоматически устанавливать связь с сервером по заданным в конфигурации IP адресам (hbc/2).

Клиент АСТД отображает:

- Мнемосхемы и элементы мнемосхем, обозначаемых прямоугольником с текстовым названием.
- Линии, символьные объекты, текст.
- Символы коммуникационных интерфейсов устройств, приборов.

- Элементы группировки – прямоугольная рамка с названием группы, объединяющая в себя элементы.

В клиенте состояние мнемосхем отображается различным цветом:

- нормальное состояние функционирования (элемент подключен по всем штатным связям – питание, информационные сопряжения (доступен по всем каналам), информация (если применимо) с данного элемента достоверна)– зеленый фон;

- желтый фон:
  - питание по одной из линий отсутствует (основное, резервное или аварийное);
  - один из информационных каналов неисправен или по нему передается недостоверная информация;

- комплект ЗИП для данного элемента исчерпан;
- неисправен – красный фон;
- состояние не определено или выключен – серый фон.

Фон неисправных элементов должен синхронно мигать на мнемосхеме с настраиваемой частотой и временем отображения яркого интервала. Пока неисправность не будет подтверждена (подтверждение – клик по элементу).

В клиенте АСТД возможно переходить по элементам мнемосхемы и подтверждать тревоги. Общая схема данных программно-аппаратной системы основана на взаимодействия между блоками системы по управляющим сетевым протоколам на уровне TCP/IP:

- прикладного уровня Simple Network Management Protocol (SNMP),
- уровня IP-соединения протокол одноадресной рассылки UNICAST для связи между двумя конечными узлами,
- уровня IP-соединения протокол многоадресной рассылки MULTICAST для передачи информации определенной группе узлов канальной среды (рис. 2).

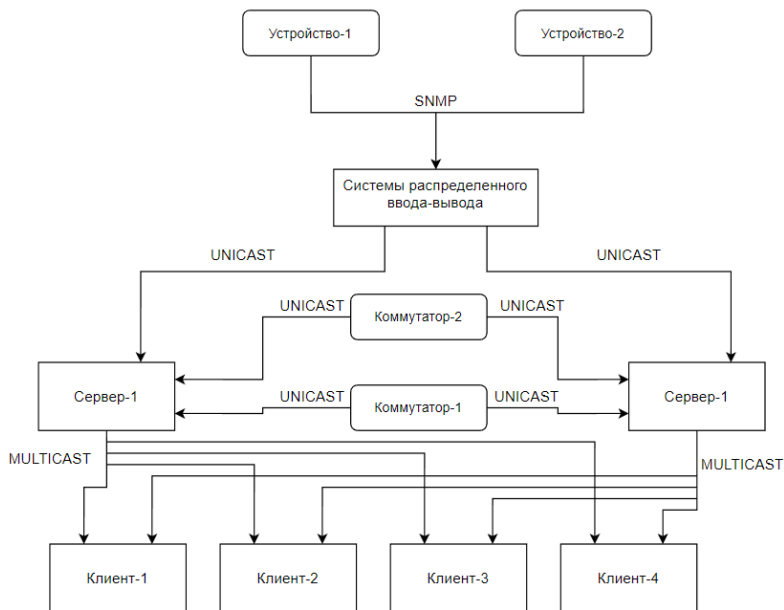


Рис. 2. Блок-схема программно-аппаратной системы

### **Заключение**

В статье показаны основные компоненты клиентской и серверной частей АСТД. Понимание структуры каждой компоненты, связей между блоками системы, протоколов взаимодействия и структуры БД необходимо при проведении пусконаладочных работ, во время эксплуатации и ремонтных работ.

### **Библиографический список**

1. Интегрированная мостиковая система. URL: <https://npfmeridian.ru/> (дата обращения: 21.05.2023).
2. Лепский А. Г., Цветков В. В., Щеглов А. А. Основные направления развития систем комплексной автоматизации на флоте // Вестник МГТУ. 2004. № 3. URL: <https://cyberleninka.ru/article/n/osnovnyie-napravleniya-razvitiya-sistem-kompleksnoy-avtomatizatsii-na-flote> (дата обращения: 21.05.2023).
3. Сорокин В. Диагностирование судовых технических средств // Морские вести России. 2018. URL: <https://www.morvesti.ru/themes/1699/73980/> (дата обращения: 21.05.2023).

УДК 004

**М. С. Ляхович**

студент кафедры прикладной информатики

**И. Ю. Немилостив**

студент кафедры прикладной информатики

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

## ФОРМИРОВАНИЕ ДИНАМИЧЕСКИ КОНФИГУРИРУЕМОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ НА ПРИНЦИПАХ ОРГАНИЗАЦИИ VPS/VDS

Введение и актуальность. Виртуальный выделенный сервер (VPS/VDS) стал очень популярным решением для хранения и обработки данных. Он позволяет пользователям получить доступ к выделенным вычислительным мощностям, без необходимости покупки и обслуживания физического сервера. В этой статье мы рассмотрим пример формирования динамически конфигурируемой информационной инфраструктуры на принципах организации VPS/VDS.

Облачная модель IaaS (Infrastructure as a Service) – инфраструктура как услуга позволяет за один и тот же бюджет получить больше ресурсов системы, либо уменьшить затраты для той или иной задачи в сравнении с содержанием или арендой выделенных серверов.

Примером такой модели являются виртуальные серверы (VPS/VDS). В отличие от PaaS (Platform as a Service), платформа как услуга, и SaaS (Software as a Service), программное обеспечение как услуга, у пользователя есть полный административный доступ к серверу, что позволяет устанавливать любую поддерживаемую гипервизором операционную систему, программное обеспечение и приложения, а внутри самой информационной инфраструктуры настраивать связи между серверами как при помощи виртуальной локальной сети, так и внешних связей. Даже крупные корпоративные заказчики стали чаще рассматривать возможности переноса своей информационной инфраструктуры в облака и VPS/VDS. Этому поспособствовало то, что за последние несколько лет облачные технологии значительно продвинулись как в технической, так и в программной частях.

Современные руководители все меньше предпочитают поддерживать работу собственных серверов, расположенных в офисе, так как это считается устаревшей практикой. Вместо этого, предприятия заинтересованы в функционирующем сервисе, который не требует обслуживания серверного оборудования и наличием соответствующих специалистов в штате, а все передается под обеспечение и сопровождение хостинг компании.

Преимущества использования VPS/VDS для размещения информационной инфраструктуры [1]:

1. Гибкость и масштабируемость: VPS/VDS позволяет пользователям гибко настраивать свои вычислительные мощности, добавлять и удалять ресурсы в зависимости от потребностей бизнеса.
2. Экономически выгодно: VPS/VDS значительно дешевле, чем физический сервер. Это особенно важно для малого и среднего бизнеса, которые могут сэкономить деньги на покупке и обслуживании собственного сервера.
3. Высокая надежность: VPS/VDS обеспечивает высокий уровень надежности, так как в случае сбоя на одном сервере, данные могут быть быстро перемещены на другой сервер.
4. Быстрое развертывание: VPS/VDS можно легко и быстро развернуть, что делает его идеальным выбором для тех, кто нуждается в мгновенном доступе к вычислительным ресурсам.
5. Круглосуточная техническая поддержка: запрос на исправление аварийной ситуации может отправить в сторону хостинга любой сотрудник организации.
6. Наличие резервного копирования: как правило данный функционал уже встроен в предоставление услуг VPS/VDS со стороны хостинга, но также есть возможность настройки собственного резервного копирования как внутри всей системы, так и с выгрузкой на внешние устройства по SSH/FTP/SFTP/S3 и другие [2].

7. Гарантия высокого показателя аптайма (uptime): хостинг постоянно отслеживает состояние своих дата-центров и обязуется в максимально короткие сроки ликвидировать аварийные ситуации без участия клиента.

Цель работы состоит в описании практического примера переноса информационной инфраструктуры предприятия на VPS/VDS сервер, чтобы в дальнейшем иметь возможность легко управлять вычислительными мощностями, моментально распределять доступ к ресурсам и проектам, надежно хранить данные и платить только за потребляемые ресурсы. Для миграции используется VPS/VDS и в целом IaaS.

Задача заключается в том, чтобы организовать информационную инфраструктуру предприятия на VPS/VDS для обеспечения непрерывной работы и возможности доступа к ПО и базам данных из любого места, тем самым обеспечить удаленный доступ для всех сотрудников предприятия.

Для решения поставленной задачи необходимо пройти все основные этапы переноса информационной системы с одного оборудования на другое. Это можно осуществить даже без наличия в штате предприятия сотрудника с соответствующими навыками, а при помощи специалистов хостинга.

Перед проведением миграции в облако необходимо выполнить ряд подготовительных этапов, включающих:

1. Анализ существующих сетей клиента.

Необходимо составить подробную топологию и схему взаимодействия, чтобы реализовать аналогичные сети в облаке и VPS/VDS без остановки бизнес-процессов.

2. Инвентаризацию физического оборудования или вычислительных мощностей.

Специалисты проводят инвентаризацию оборудования и виртуальных вычислительных ресурсов для определения объема услуг из облака, а также учет текущей загрузки и пиковых потребностей, чтобы разместить в облаке виртуальные машины необходимой мощности.

3. Определение состава инфраструктур и сервисов.

Составление перечня баз данных, систем и приложений, которые работают в ЦОДе, позволяет спланировать и организовать нужные сервисы, а также построить аналогичную систему в облаке с оптимизацией и перераспределением нагрузок.

4. Организацию VPN-каналов между внутренними сетями и облаком.

Необходимо обеспечить непрерывную работу предприятия при переходе в облако с помощью надежных каналов связи, пропускная способность которых оценивается исходя из потребностей текущего трафика, а также переноса систем и данных.

5. Выбор инструментов переноса физических серверов в облако.

В большинстве случаев с задачей миграции справляются стандартные конвертеры физических серверов в виртуальные машины. Также можно создать локальный образ сервера и перенести его на VPS/VDS с последующей настройкой портов и доступов.

После выполнения этих этапов локальная инфраструктура предприятия начинает постепенно безболезненно перемещаться в облако, что позволяет продолжать работу предприятия в штатном режиме.

Перенос сайта происходит еще проще, для этого почти не требуется подготовительных этапов, а весь процесс состоит из следующих частей [3]:

1. Подготовка и запуск VPS/VDS;

2. Нахождение путей файлов сайта на передающем сервере;

3. Экспорт дампа базы данных сайта;

4. Перенос файлов сайта с передающего на принимающий сервер;

5. Импортowanie дампа базы данных;

6. Изменение данных о базе данных в конфигурационных файлах сайта;

7. Проверка работы сайта на принимающем сервере;

8. Делегирование домена на IP-адрес нового сервера.

Результатом решения поставленной задачи будет полная миграция в облако существующей инфраструктуры предприятия.

IaaS является оптимальной площадкой для формирования динамически конфигурируемой информационной инфраструктуры предприятия, на которой также можно проводить тестирование разнообразных задач перед их воплощением. При правильной настройке каналов доступа, работа в облаке может быть незаметной и даже более эффективной, так как все данные передаются по сети Интернет из любой точки мира

В статье рассмотрены примеры методики переноса сложной локальной инфраструктуры предприятия и способы переноса инфраструктуры web-приложения.

#### **Библиографический список**

1. The advantages of VDS – server for website hosting // RealHosters. URL: <https://www.realhosters.com/en/buyvds.php> (дата обращения: 19.04.2023).
2. Бэкапы на VDS хостинге – все, что нужно знать о резервном копировании // Олег Adminvps. URL: <https://vc.ru/u/383134-oleg-adminvps/376968-bekapy-na-vds-hostinge-vse-chto-nuzhno-znat-o-rezervnom-kopirovanii> (дата обращения: 19.04.2023).
3. Перенос сайта на сервер // Timeweb Cloud. URL: <https://timeweb.cloud/docs/cloud-servers/site-transfer> (дата обращения: 19.04.2023).

УДК 004

**И. Ю. Немилостив**

студент кафедры прикладной информатики

**М. С. Ляхович**

студент кафедры прикладной информатики

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

## МЕТОДЫ ИНТЕГРАЦИИ ИНФОРМАЦИОННЫХ СИСТЕМ

В настоящее время, информационные системы уже стали неотъемлемой частью любой коммерческой или общественной организации. Каждая организация имеет свои специфические потребности, и чтобы эффективно решать свои задачи, они нуждаются в надежных и эффективных информационных системах. Однако, часто организации сталкиваются с проблемой совместимости различных информационных систем внутри компании. Для того чтобы решить эту проблему, необходимо использовать различные методы интеграции информационных систем.

Интеграция информационных систем – это процесс объединения нескольких информационных систем в единую с целью обеспечения надежности и эффективности работы организации. Но какую же информационную систему выбрать и какие методы использовать для интеграции систем?

Существует несколько методов интеграции информационных систем, которые можно использовать при решении проблем совместимости информационных систем:

– Интеграция на уровне данных.

Данный вид интеграции данных подразумевает объединение данных, хранимых в различных информационных системах, для того чтобы они могли использоваться вместе и обеспечивать согласованный обмен информацией между приложениями. Цель интеграции – конечные пользователи работают с едиными данными, а обрабатывают их разными инструментами [1].

– Интеграция на уровне сервисов.

Основным принципом интеграции на уровне сервисов является фиксация интерфейсов, использование открытых стандартов, общих форматов данных и протоколов связи между системами. Это позволяет разработчикам легче интегрировать сервисы в существующие системы и делать их более гибкими и масштабируемыми. Преимуществом является организация стремительной отработки межкорпоративной бизнес-логики [2].

– Интеграция на уровне брокеров.

Брокер – в информационных технологиях дополнительный программный модуль, выполняющий роль посредника, обращающийся в другие системы различными способами. Преимуществом данного метода интеграции является его универсальность. Практически в любом случае есть возможность реализовать дополнительный программный модуль, однако у данного способа есть также и существенный недостаток – реализация такого брокера очень сложный процесс, который обычно требует больших как финансовых, так и временных затрат [3].

– Интеграция на уровне пользователя.

Данный вид относится к неавтоматизированной интеграции. Самый простой подход интеграции, когда не происходит связывание программных компонентов системы, а пользователи напрямую взаимодействуют со всеми соответствующими информационными системами и вручную интегрируют выбранные данные. То есть пользователям приходится иметь дело с различными пользовательскими интерфейсами и языками запросов. Является наиболее простым видом/подходом, и часто применяется в тот момент, когда происходит подготовка внедрения программных систем, а деятельность компании не может прерываться [4].

– Интеграция при помощи Web-сервисов.

Данный вид интеграции является передовым и стремительно развивающимся подходом к интеграции приложений. Он базируется на предоставлении стандартного для Web-служб интерфейса доступа к приложениям и их данным [5]. Они позволяют организациям отдавать и принимать данные через Интернет без необходимости установки каких-либо дополнительных программ или оборудования.

Веб-сервисы используют стандартные протоколы, такие как HTTP, SOAP и XML. Это означает, что любая ОС или язык программирования может использовать этот метод.

– Интеграция на уровне интерфейсов (физических, программных и пользовательских).

Совмещение интерфейсов нескольких информационных систем в одном окне, для облегчения работы и повышения эффективности пользователя. Наиболее распространенный вариант использования – HTML-scraping, при котором специальный инструмент, идентифицирует компоненты HTML-документа, полученного в результате работы веб-приложения, и предоставляет эти компоненты для повторного использования и интеграции [5].

– Интеграция на функционально-прикладном и организационном уровнях.

Данный вид интеграции построен на объединении нескольких однотипных или похожих функций в макрофункции, в которых перераспределяются ресурсы, потоки данных, управление и механизмы исполнения. Как следствие, это влечет за собой реорганизацию информационных структур, бизнес-процессов и, соответственно, перестройку схем их информационного и документационного обеспечения [2].

– Интеграция на уровне корпоративных программных приложений.

Представляет собой совместное использование исполняемого кода (ядра), не ограничиваясь обменом информацией. Программы делятся на компоненты, которые впоследствии связываются между собой с помощью стандартизованных программных интерфейсов API (Application Programming Interface) и специализированного связующего ПО. Преимуществом такой интеграции можно назвать создание единого информационного пространства, которое значительно облегчает дальнейшую поддержку и расширение информационной системы. Однако проведение данного подхода может осложниться на этапе анализа интегрируемых приложений – для оценки возможности интеграции каждого отдельного приложения необходимо наличие исходных кодов, которые не всегда предоставляются разработчиками для свободного доступа [3].

Самый популярный метод интеграции информационных систем в настоящее время является интеграция с помощью API.

API-интеграция является предпочтительным методом для многих организаций, поскольку позволяет реализовать быстрое и безопасное взаимодействие между системами, а также упрощает процесс внедрения и сокращает время настройки. Кроме того, использование API-интеграции позволяет сократить ручной труд и избежать ошибок, связанных с механической обработкой данных.

Предположим, что в компании используется система управления контактами, в которой хранятся данные о клиентах. Хранимые данные компания использует для рассылки разного рода информации. Для удобства управления продажами в компании используют CRM-систему. Чтобы упростить существующие процессы и объединить информацию о клиентах, компании требуется произвести интеграцию между двумя системами.

Взаимодействие между системами выглядит следующим образом:

1. Получаем ключ доступа от системы управления контактами и регистрируем его в настройках CRM-системы.
2. Менеджер получает информацию о клиенте из CRM-системы и обращается к системе управления контактами для получения более подробной информации о контакте.
3. Запрос на получение более подробной информации отправляется через API системы управления контактами.
4. Система управления контактами отвечает на запрос и возвращает запрошенную информацию.
5. Менеджер использует эту информацию для улучшения процесса продаж и записывает ее в CRM-систему.
6. Обе системы могут использовать API для обмена информацией о контактах клиентов.

В заключение необходимо отметить то, что, методы интеграции информационных систем являются важным компонентом любой организации, которая хочет работать эффективно и надежно. Они позволяют объединять несколько информационных систем вместе и упрощать обмен данными между ними. Выбор метода интеграции информационных систем зависит от конкретных потребностей организации и ее целей.



**Библиографический список**

1. Шведенко В. Н. Критерии оценки и модели информационных систем, обладающих свойством поведения / В. Н. Шведенко, О. В. Щекочихин, П. В. Шведенко // Научно-технический вестник информационных технологий, механики и оптики. 2016. Т. 16. № 4. С. 649–654. DOI 10.17586/2226-1494-2016-16-4-649-654.
2. Думченков И. А. Обзор методов интеграции информационных систем, их преимуществ и недостатков // Молодой ученый. 2018. № 23 (209). С. 176–177.
3. Шевченко Г. С. Интеграция информационных систем. Основные виды и методы интеграции / Г. С. Шевченко, А. С. Кальченко, О. Н. Торгованова // Вестник образовательного консорциума Среднерусский университет. Информационные технологии. 2020. № 2 (16). С. 69–74.
4. Лебедев А. С. Проблемы интеграции корпоративных информационных систем – методы и технологии // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2020. № 8. С. 73–78. DOI 10.37882/2223-2966.2020.08.19.
5. Винокуров И. В. Методы интеграции данных в корпоративных информационных системах // Творчество молодых: дизайн, реклама, информационные технологии: Сборник трудов XV Международной научно-практической конференции студентов и аспирантов, Омск, 21–22 апреля 2016 года / Научный редактор Л. М. Дмитриева. Омск: Омский государственный технический университет, 2016. С. 96–99.

УДК 004

**В. В. Пономаренко**

студент кафедры прикладной информатики

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

## МЕТОДЫ ПОЛУЧЕНИЯ ИНФОРМАЦИИ С ВЕБ-РЕСУРСОВ

Многие сведения расположены в Интернете в свободном доступе. Они могут быть извлечены с веб-ресурсов и обработаны для получения значимых результатов. Сведения, полученные такими методами, являются ценными. Ведь, как говорил Натан Ротшильд: «Кто владеет информацией – тот владеет миром».

Извлечение данных с веб-ресурсов используется в различных областях и отраслях, таких как бизнес, исследования, электронная коммерция, журналистика, финансы. Поскольку полученная информация может помочь организациям принимать более обоснованные решения и получать конкурентные преимущества [1].

### 1. Методы извлечения данных с веб-ресурсов

Сбор информации вручную занимает очень много времени, поэтому не подходит для больших веб-приложений и задач, в которых необходимо многократно получать данные с ресурса на протяжении длительного срока. В таких случаях данные с ресурса извлекаются через API или с помощью веб-скрапинга (рис. 1).

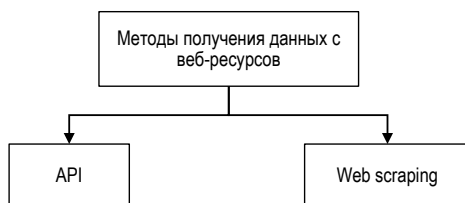


Рис. 1. Методы извлечения данных с веб-ресурсов

### 2. Получение данных через API

API – программный интерфейс веб-приложения, представляющий набор функций для облегчения взаимодействия между клиентом и сервером [2].

Зачастую большие веб-ресурсы имеют открытые API, благодаря которым данные быстро извлекаются напрямую с сервера в структурированном виде (JSON, XML).

Для получения данных с ресурса клиент отправляет GET-запрос к серверу, ответом на который и является необходимая информация (рис. 2). Такой метод прост и лаконичен при выгрузке большого объема данных с веб-ресурса, обладающего хорошо задокументированный API.

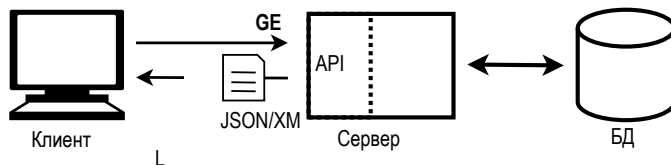


Рис. 2. Принцип получения данных с сервера через API

### 3. Получение данных через Web Scraping

Во многих случаях необходимо собрать наборы данных с большого количества сайтов, у которых нет единого API, или извлечь информацию с небольших сайтов, где API закрыт, ограничен по числу запросов, не предоставляет нужную информацию, плохо задокументирован или вовсе не разработан [3, 4]. В таких случаях для извлечения информации используют веб-скрапинг. Райан Митчелл [3] дает следующее толкование понятию: «веб-скрапинг (web scraping) – это сбор данных с использованием любых средств, за исключением программ, взаимодействующих с API».

Для работы веб-скрапера необходимо знать адрес страницы, с которой нужно извлечь данные. Первым делом скрапер отправляет запрос к серверу веб-ресурса на получение HTML-разметки нужной страницы. Затем выбирает нужные данные по тегам и классам HTML-разметки. Полученные данные при желании могут быть преобразованы к структурированному виду. После чего они экспортируются в файл или базу данных [3].

Если скрапер извлекает данные с нескольких страниц, то он повторяет процесс по кругу, используя новые адреса страниц. Наглядный принцип работы веб-скрапера изображен на рис. 3.

### 4. Программные средства для создания скрапера

Веб-скрапер может быть реализован на многих языках программирования: Java Script, Python, R, Java, Ruby, C++ и прочие [5]. В зависимости от языка разработки, программа может иметь разный объем, читаемость, простоту написания и скорость работы.

Самым популярным языком для создания веб-скраперов является Python, благодаря простоте написания кода, большому сообществу и перечню библиотек, а также дешевизне разработки, которая нивелирует невысокую скорость работы программы.

Для написания веб-скрапера на Python используются следующие библиотеки: Requests, BeautifulSoup, Scrapy, Lxml и Selenium.

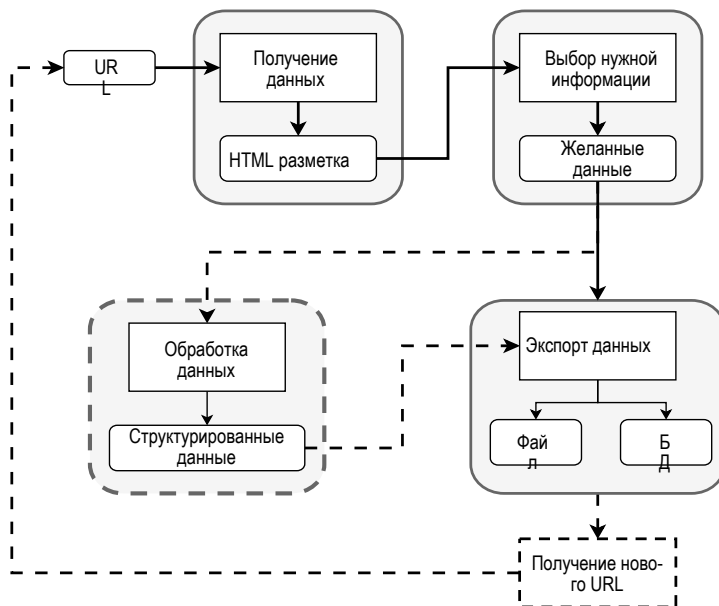


Рис. 3. Принцип работы веб-скрапера

## 5. Инструменты для скрапинга

Чтобы упростить процесс веб-скрапинга и сделать его доступным для людей с разным уровнем компьютерной грамотности, существует несколько видов инструментов (рис. 4) для веб-скрапинга: десктопные приложения, браузерные расширения и специальные API для скрапинга [6].

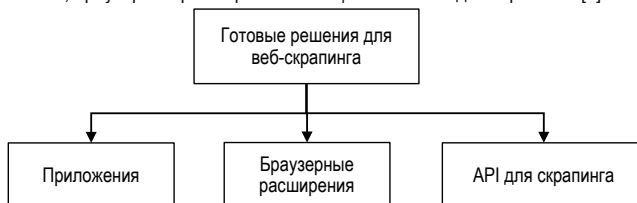


Рис. 4. Виды готовых решений для веб-скрапинга

### 5.1. Приложения для скрапинга

Самыми популярными готовыми решениями для самостоятельного скрапинга являются приложения для компьютера. Они позволяют пользователям без навыков программирования получить данные с желаемого веб-ресурса.

Поскольку у каждого приложения своя специфика работы, то для сбора данных рядовому клиенту придется посмотреть документацию или видеоинструкцию конкретного продукта, которая может быть сложна для освоения неопытными пользователями компьютера.

Существует большой перечень программного обеспечения для получения данных с сайтов. Наиболее популярными представителями являются: OctoParse, Datacol, ScrapeStorm и WebHarvy [6].

### 5.2. Браузерные расширения для скрапинга

Помимо приложений для компьютера существуют также браузерные расширения для извлечения данных с веб-страниц. Они имеют меньше возможностей и низкий порог вхождения, благодаря чему подходят для знакомства с веб-скрапингом и самостоятельного извлечения небольшого набора данных неопытными пользователями компьютера. Самыми популярными браузерными расширениями для скрапинга являются: Instant Data Scraper, Data Miner и Web Scraper [6].

### 5.3. API для скрапинга

Также существуют прикладные интерфейсы (API) для скрапинга веб-сайтов. Они используются программистами для упрощения разработки веб-скраперов сайтов с серьезной защитой от парсинга. Сервисы, предоставляющие API для скрапинга доступны на разных языках программирования. Для использования таких интерфейсов необходимо скачать специальную библиотеку и иметь аккаунт.

Также подобные сервисы имеют ряд готовых веб-скраперов для некоторых сайтов, которые могут быть доступны клиентами с продленной подпиской. Самыми популярными API для скрапинга являются: ScraperAPI, ScrapingBee и ScrapingAnt [6].

По результатам поделанной работы был произведен обзор основных методов извлечения информации с веб-ресурсов разных размеров. Также были описаны программные решения для веб-скрапинга, подходящие пользователям с разным уровнем компьютерной грамотности.

## Библиографический список

1. Patel J. Getting Structured Data from the Internet: Running Web Crawlers/Scrapers on a Big Data Production Scale. Apress, 2020. 420 с.
2. Masse M. REST API Design Rulebook. O'Reilly Media, 2012. 95 с.
3. Митчелл Р. Современный скрапинг веб-сайтов с помощью Python. 2-е изд. СПб.: Питер, 2021. 498 с.
4. Broucke S., Baesens B. Practical Web Scraping for Data Science: Best Practices and Examples with Python. Apress, 2018. 322 с.

5. Skakun V. The Best Programming Languages for Web Scraping. URL: <https://scrape-it.cloud/blog/web-scraping-languages> (дата обращения: 17.03.2023).

6. Кульгин М. Лучшие инструменты для самостоятельного парсинга веб-сайтов. URL: <https://vc.ru/services/249048-luchshie-instrumenty-dlya-samostoyatel'nogo-parsinga-veb-saytov-ischerpyvayushchiy-spisok> (дата обращения: 12.01.2023).

УДК 004.08

**Д. В. Сапунов**

студент кафедры прикладной информатики

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

## ИНТЕГРАЦИЯ ТЕХНОЛОГИЙ БЛОКЧЕЙН И ОДНОСТРАНИЧНЫХ ВЕБ-ПРИЛОЖЕНИЙ

### Введение

Для создания мощных децентрализованных приложений, способных обеспечить пользователям взаимодействие с данными блокчейна и смарт-контрактами в режиме реального времени, используют специализированные программные библиотеки и фреймворки, например? React и Web3.js [1 – 4]. При интеграции таких программных средств возникают сложности, поэтому в статье показана методика настройки их соединения между собой.

React – это популярная библиотека JavaScript для создания одностраничных приложений (англ. single page application, SPA), которая предоставляет гибкий и эффективный способ создания пользовательских интерфейсов [1]. Metamask – это кошелек-расширение для браузера, позволяющий пользователям взаимодействовать с блокчейном Ethereum непосредственно из своего веб-браузера [2, 3].

### Краткий обзор технологий

Технология блокчейн – это технология распределенного реестра, которая обеспечивает безопасное и прозрачное ведение учета транзакций. Технология имеет широкий спектр применений, от криптовалюты до управления цепочками поставок, и обладает потенциалом для преобразования различных отраслей [5].

Ethereum – это децентрализованная платформа, позволяющая разработчикам создавать децентрализованные приложения (dApps) поверх своего блокчейна [3]. Платформа предназначена для запуска смарт-контрактов – небольших самоисполняющихся интерфейсов для взаимодействия покупатель-продавец, условия взаимодействия которых описаны открытым исходным кодом. Это позволяет автоматизировать различные процессы, такие как финансовые транзакции, проверка личности и многое другое.

Metamask – это кошелек-расширение для браузера, который позволяет пользователям взаимодействовать с блокчейном Ethereum непосредственно из своего веб-браузера. Кошелек действует как мост между веб-браузером пользователя и сетью Ethereum, позволяя им управлять своими учетными записями Ethereum, отправлять и получать Ether и другие токены, а также взаимодействовать со смарт-контрактами [2].

Используя Metamask, пользователи могут безопасно управлять своими закрытыми ключами, не полагаясь на централизованную третью сторону. Это связано с тем, что Metamask шифрует закрытый ключ пользователя и хранит его локально на его компьютере, что означает, что пользователь полностью контролирует свои средства и транзакции. Более того, Metamask предоставляет удобный интерфейс, который упрощает процесс взаимодействия с сетью Ethereum для нетехнических пользователей.

### Настройка React SPA для взаимодействия с блокчейном

Для создания React SPA, взаимодействующего с блокчейном, необходимо выполнить следующие шаги:

1) создать новый проект React с помощью инструмента create-react-app. Этот инструмент предоставляет базовый шаблон для проекта React и упрощает процесс настройки [1]. В терминале использовать следующую команду:

```
prx create-react-app my-app
```

2) установить web3.js: для взаимодействия с блокчейном Ethereum необходимо использовать библиотеку JavaScript под названием web3.js. Эта библиотека предоставляет простой API для взаимодействия с узлами Ethereum и смарт-контрактами [3]. Установка web3.js происходит с помощью prn следующей командой:

```
npm install web3
```

3) инициализировать web3.js: после установки web3.js, необходимо инициализировать его с помощью правильного провайдера. Провайдер – это шлюзом между вашим приложением и сетью Ethereum [3]. В случае Metamask провайдер будет добавлен на веб-страницу автоматически при установке расширения. Для инициализации web3.js с провайдером Metamask следует использовать код:

```
import Web3 from 'web3';
const web3 = new Web3(window.ethereum);
```

4) после того, как среда разработки подготовлена, можно приступать к написанию приложения. Был создан компонент, который взаимодействует с кошельком и отправляет средства на другой кошелек при нажатии на кнопку (см. листинг).

Листинг. Программный компонент интеграции

```
function App() {
  const [web3, setWeb3] = useState(null); // объект web3 для взаимодействия с блокчейном
  const [account, setAccount] = useState(null); // адрес пользователя

  const sendTransaction = async () => {
    const to = '0xe2DEF3f5929D2f41C59196c069f79de6442a348a'; // адрес получателя
    const value = web3.utils.toWei('0.01', 'ether'); // отправляемая сумма
    const tx = { from: account, to, value }; // объект транзакции
    await web3.eth.sendTransaction(tx); // вызов транзакции из объекта
  };

  useEffect(() => {
    const newWeb3 = new Web3(window.ethereum); // инициализация web3 с помощью встроенного в браузер кошелек
    if (newWeb3) setWeb3(newWeb3);
  }, []);

  useEffect(() => {
    const enableMetamask = async () => {
      if (!web3) return
      const accounts = await web3.eth.getAccounts(); // запрос кошелька из Metamask
      setAccount(accounts[0]); установка адреса пользователя
    };

    if (window.ethereum) {
      enableMetamask(); // если кошелек обнаружен, то происходит вызов функции, которая получает адрес кошелька
    }
  }, [web3]);

  return (
    <div className="App">
      Account is {account} // отображаем адрес кошелька пользователя
      <button onClick={sendTransaction}>Send</button> // кнопка вызова транзакции
    </div>
  );
```

Хук useState использован для создания двух состояний – web3 и account. Первый объект содержит экземпляр web3, второй – текущий адрес кошелька пользователя.

Далее определена функция sendTransaction. Эта функция использует экземпляр web3 для отправки транзакции в размере 0,01 эфира на указанный адрес, был создан объект транзакции – tx со свойствами from, to и value. Затем этот объект будет использован как параметр в функцию отправки

транзакции `sendTransaction` у объекта `eth` в `web3` (функция является асинхронной и ожидает результата транзакции).

Два хука `useEffect` вызваны для инициализации `web3` и подключения `Metamask`. Первый хук `useEffect` инициализирует `web3` путем создания нового экземпляра класса `Web3` с помощью `window.ethereum` объекта. Если `newWeb3` успешно инициализирован, он устанавливает его в переменную `web3` с помощью функции `setWeb3`.

Второй хук `useEffect` имеет встроенный метод `enableMetamask`, который запрашивает подключения к сайту аккаунты `Metamask`. Если `window.ethereum` существует (т. е. у пользователя установлен `Metamask`), тогда будет вызван метод `enableMetamask`.

Затем компонент приложения возвращает элемент `div`, отображающий подключенный кошелек пользователя, и кнопку `button`, которая вызывает функцию `sendTransaction`.

### Заключение

В данной статье были показаны основы использования `React SPA` с `Metamask` для взаимодействия с блокчейном `Ethereum`. Кратко рассмотрены технология блокчейн и функциональность смарт-контрактов `Ethereum`, а также роль `Metamask` в качестве кошелька-расширения браузера для `Ethereum`. Описан алгоритм настройки `React SPA` для взаимодействия с блокчейном и внедрения `Metamask`.

В настоящее время существует значительный потенциал для дальнейшего развития взаимодействия между `React SPA` и технологией блокчейн. Например, рост децентрализованных автономных организаций (DAO) может привести к развитию `React SPA` со встроенными структурами управления, основанными на смарт-контрактах.

Интеграция технологии блокчейн с `React SPA` с использованием `Metamask` и `web3.js` может предоставить мощный набор инструментов для создания децентрализованных приложений.

### Библиографический список

1. Официальная документация `React`. URL: <https://reactjs.org/> (дата обращения: 18.04.2023).
2. Официальная документация `Metamask`. URL: <https://metamask.io/> (дата обращения: 18.04.2023).
3. Официальный сайт `Ethereum`. URL: <https://ethereum.org/en/> (дата обращения: 18.04.2023).
4. Официальная документация `web3.js`. URL: <https://web3js.readthedocs.io/en/> (дата обращения: 18.04.2023).
5. Tutorial: Building a web3 frontend with `React` URL: <https://medium.com/scrappy-squirrels/tutorial-building-a-web3-frontend-with-react-e0a87ea3bad> (дата обращения: 18.04.2023).



УДК 004.92

**Я. И. Сенина**

студент кафедры прикладной информатики

**А. В. Лацко**

студент кафедры прикладной информатики

**Н. Н. Григорьева** – старший преподаватель – научный руководитель**ПРОГРАММНЫЙ РЕНДЕРИНГ МЕТОДОМ ТРАССИРОВКИ ЛУЧЕЙ**

Трассировка лучей – это один из методов геометрической оптики, а именно исследование оптических систем путем отслеживания взаимодействия отдельных лучей с поверхностями. В данной статье мы будем рассматривать трассировку лучей в узком смысле, как технологию построения изображений трехмерных моделей в компьютерных программах.

Как же это работает? Рассматриваемый метод рендеринга позволяет создавать изображения с высокой степенью реализма благодаря имитации пересечения пути лучей света и объекта. Интересно, что на самом деле этот процесс обратный реальному освещению. В реальности свет выходит из источника и рассеивается во всех направлениях, часть лучей отражается от объекта, и лишь некоторая часть из всех изначальных лучей попадает в камеру. Трассировка же отслеживает лучи, выходящие из камеры, а не из источника света, что обеспечивает сокращение времени вычислений.

Работа алгоритма трассировки лучей построена на базовой векторной геометрии [1]. Каждый пиксель мы вычисляем с помощью единичного вектора (луча), который выходит из камеры. Вектор можно задать уравнением:

$$\text{ray}(t) = \text{camera} + \frac{\text{pixel} - \text{camera}}{|\text{pixel} - \text{camera}|}t,$$

где  $t$  – параметр, с помощью которого мы описываем местонахождение отдельного пикселя относительно камеры. При  $t = 0$  мы находимся в положении камеры, при увеличении параметра  $t$  мы будем удаляться от нее в направлении определенного пикселя. Далее определяем уравнение сферы

$$|x - O| = R,$$

где  $O$  – центр окружности,  $R$  – радиус, а  $x$  – произвольная точка, лежащая на сфере. Для нахождения точки пересечения луча, выходящего из камеры со сферой, решаем уравнение

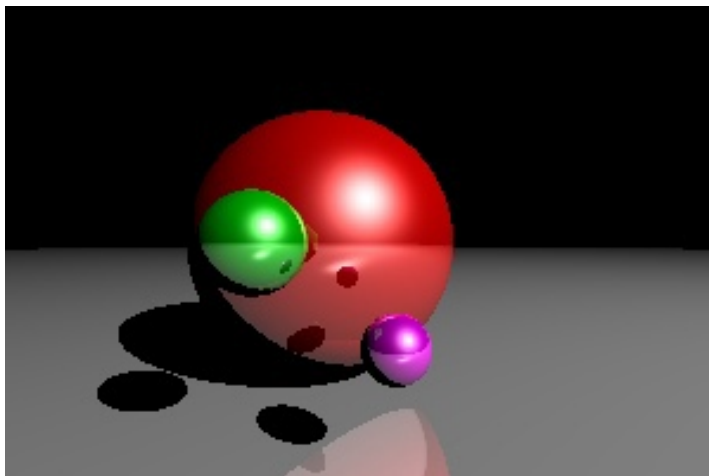
$$|\text{ray}(t) - O|^2 = r^2.$$

Для разработки программного средства трассировки были установлены основные объекты сцены, которые участвуют в визуализации: трехмерное пространство, объекты (в нашем случае сферы), источник света (точка, излучающая свет во всех направлениях), камера, экран. Изначально рендеринг изображения производился в очень низком качестве (200×300 пикселей), чтобы настроить объекты в сцене, свет и тени (рис. 1).

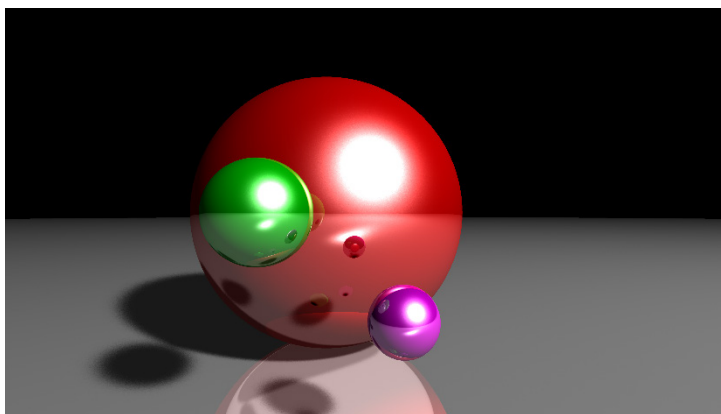
По рис. 1 хорошо видно, что в первоначальном варианте кода был прописан направленный источник света, что давало жесткие тени. Для того чтобы исправить это, в программе была реализована функция `soft_shadow`, которая позволяет сделать мягкие, настраиваемые тени. Принцип работы функции состоит в том, что из заданной позиции источника света генерируется  $N$  случайных точек на поверхности источника. Вычисляется расстояние между точкой источника и заданной точкой пересечения `intersection`.

Для реализации этой функции использовали методы Монте-Карло и Фонга, пробовали сначала реализовать эту функцию на одном объекте (рис. 2). Затем меняли настройки теней, для получения разных вариантов изображения.

Трассировка лучей как метод рендеринга имеет как свои преимущества, так и существенные недостатки.



*Рис. 1. Рендер сцены (размер 200×300 пикселей)*



*Рис. 2. Рендер в качестве 4к с мягкими тенями*

Главным преимуществом данной технологии, конечно же, является реалистичность отображения объектов и их взаимодействия со светом. Поэтому рейтрейсинг – одна из самых успешных техник 3D-рендеринга.

Основной проблемой трассировки лучей является требовательность к вычислительным ресурсам. В реальном мире фотоны двигаются сами по себе, в мире компьютеров этот процесс контролирует видеокарта.

При расчете всего одного кадра в разрешении Full HD компьютеру необходимо одновременно просчитать более двух миллионов виртуальных лучей, каждый из которых, прежде чем сформирует один пиксель на экране, будет взаимодействовать не только с объектом освещения, но и с другими своими «сородичами». Еще стоит учесть, что на практике обработка лучей ведется не в статичной сцене, а прямо во время игры, где кадры сменяются с частотой 50–60 FPS. В таком случае миллионы лучей превращаются в десятки или даже сотни миллионов задач на обработку.

В своей работе мы попытались воссоздать простую реализацию визуализации методом трассировки лучей на языке программирования Python. Конечно же, наш код является далеким от идеала, но на его основе хорошо можно понять основные принципы рендеринга.

Ссылка на облако с кодом разработанного программного средства и полученными изображениями: [https://drive.google.com/drive/folders/1xOXhUDuMv-TjBUaMUpTiDpXB64DBRFza?usp=share\\_link](https://drive.google.com/drive/folders/1xOXhUDuMv-TjBUaMUpTiDpXB64DBRFza?usp=share_link).

#### **Библиографический список**

1. Шикин Е. В., Боресков А. В. Компьютерная графика. Динамика, реалистические изображения. М.: АО "ДИАЛОГМИФИ", 1997. 288 с.

УДК 004

**Д. Н. Сигачева**

студент кафедры прикладной информатики

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

## ИССЛЕДОВАНИЕ ВЛИЯНИЯ ПАРАМЕТРОВ СТРИМИНГОВЫХ СЕРВИСОВ НА ВОСПРИЯТИЕ ЧЕЛОВЕКОМ МУЗЫКАЛЬНЫХ ПРОИЗВЕДЕНИЙ

### Введение

В современном мире музыка стала неотъемлемой частью нашей жизни, и все больше людей используют стриминговые сервисы для ее прослушивания. Однако, вопрос качества звучания на таких платформах остается открытым. Целью данного исследования является анализ параметров музыкальных сервисов и их влияние на восприятие аудиоконтента слушателем.

Первая задача исследования заключается в проведении патентного поиска для выявления уже существующих исследований, посвященных анализу параметров музыкальных сервисов и их влияния на качество звучания. Этот поиск позволит получить обзор текущего состояния исследований в данной области и определить пробелы, которые можно заполнить с помощью данного исследования.

Вторая задача исследования заключается в разработке методики, которая будет использована для проведения анализа параметров музыкальных сервисов и их влияния на восприятие аудиоконтента. Методика должна включать выбор треков, источников и форматов аудиофайлов для анализа, а также конкретные шаги и инструменты, которые будут использованы для проведения анализа. Это позволит обеспечить объективность и репрезентативность исследования.

### Методика исследования

С помощью программного пакета «Adobe Audition 2020» было проведено исследование аудиофайлов различных форматов и музыкальных жанров с разных источников, включая «VK музыку», «Звук. Музыка и подкасты», «YouTube» и «Яндекс Музыка». Для каждого файла был выполнен частотный анализ всего трека, после чего были выбраны 4 различных частотных резонанса [1]. Для каждой частоты были рассчитаны гармоники 1-го, 2-го и 3-го порядка, находящиеся на графике в районе среднего уровня громкости. При выборе частотных резонансов исключались пересекающиеся гармоники разных частот [2]. Далее проводилась оценка того, входят ли все гармоники в воспроизводимый диапазон частот. Анализ аудиофайлов осуществлялся в программном пакете «Adobe Audition 2020» без использования дополнительных программ. Ограничение воспроизводимого частотного диапазона для анализа было индивидуальным для каждого формата файлов.

Используемое оборудование и программное обеспечение

Adobe Audition (бывший Cool Edit Pro) – аудиоредактор производства компании Adobe. Поддерживает практически все звуковые форматы файлов. В данном редакторе используется функция «Frequency Analysis», которая предназначена для выявления проблематичных полос частот для последующего исправления ситуации. В этой панели горизонтальная ось составляет собой частоту (измеряется в Гц), а вертикально ось представляет собой амплитуду соответствующей частоты измеряется в децибелах.

Критериями оценки качества аудиофайлов будет являться величина потока данных с диска (кбит/с), оценка частных резонансов и влияния их гармоник на слуховые ощущения при прослушивании файла.

Для оценки были выбраны следующие треки: Imanbek & BYOR – Belly dancer, Пицца – Лифт, Harry Connick Jr. – Mind if I make love to you. Для проведения анализа качества выбраны стриминговые платформы «VK музыка», «YouTube», «Яндекс. Музыка» и «Звук. Музыка и подкасты».

### Анализ музыкального произведения «Imanbek & BYOR –Belly dancer»

Аудиофайлы платформы «Яндекс. Музыка» имеют ослабление верхних частот, начиная с 13500 Гц, с 18600 Гц и выше частоты не воспроизводятся совсем.

Аудиофайлы платформы «VK Музыка» имеет ослабление верхних частот начиная с 15000 Гц, с 19700 Гц и выше частоты не воспроизводятся совсем.

Аудиофайлы платформы «YouTube» имеют резкий срез верхних частот начиная с 15500 Гц, а начиная с 16600 Гц и выше частоты не воспроизводятся совсем.

Аудиофайлы платформы «Звук. Музыка и подкасты» имеют диапазон воспроизводимых частот в среднем от 50 до 21000 Гц, не имеют резких срезов верхних частот.

### Результаты исследования

Формат .flac отсутствует на платформах «VK музыка», «YouTube» и «Яндекс. Музыка», но присутствует на сервисе «Звук. Музыка и подкасты».

Формат .flac использует собственный алгоритм сжатия, который позволяет избежать неточностей не целочисленной арифметики и гарантирует сжатие без потерь.

Файлы в формате .flac имеют не сжатый диапазон воспроизводимых частот от 1 до 21500 Гц, что отличает их от файлов в формате MP3, где высокие частоты ослаблены или не воспроизводятся совсем. Аудиоконтент в формате .flac на сервисе «Звук. Музыка и подкасты» является лучшим по качеству звучания по сравнению с аналогичным контентом в формате .mp3 на других сервисах.

### Выводы

В результате анализа файлов с различных музыкальных платформ были обнаружены различия в ограничениях воспроизводимого частотного диапазона. Форматы такие как .wav не используются на платформах, скорее всего по причине большого размера файлов.

Формат .flac, который использует собственный алгоритм сжатия без потерь, обеспечивает более широкий диапазон воспроизводимых частот по сравнению с форматом .mp3. Наличие формата .flac на одной из платформ свидетельствует о возможности предоставления более высокого качества звучания на этой платформе.

Результаты исследования могут быть использованы для улучшения качества звучания музыки на стриминговых сервисах и повышения удовлетворенности пользователей. Возможные меры включают предоставление более широкого диапазона воспроизводимых частот, использование форматов аудиофайлов с меньшей потерей качества, и оптимизацию алгоритмов сжатия.

### Библиографический список

1. Отчет об исследовательской работе "Анализ аудиофайлов и техническая оценка их качества" / ООО "Октава ДМ". URL: [https://oktavatula.com/upload/lssledovatel'skaya\\_rabota\\_Analiz\\_audiofajlov\\_i\\_tekhnicheskaya\\_ocenka.pdf](https://oktavatula.com/upload/lssledovatel'skaya_rabota_Analiz_audiofajlov_i_tekhnicheskaya_ocenka.pdf) (дата обращения: 15.04.2023).

2. Alex Wilson Bruno Fazenda Perception of Audio Quality in Productions of Popular Music // Journal of the Audio Engineering Society. 2016. № 1/2. С. 23–34.

УДК 004.421+796.063.4

С. С. Чепурнова

студент кафедры прикладной информатики

Е. А. Яковлева – кандидат физико-математических наук, доцент – научный руководитель

## РЕАЛИЗАЦИЯ АЛГОРИТМА ПОДБОРА СПАРРИНГ-ПАРТНЕРА В СПОРТИВНЫХ ЕДИНОБОРСТВАХ

**Актуальность.** В мире спорта невозможно существовать вне конкуренции. Сама его суть заключается в соревновании, сопоставлении определенных способностей и навыков людей между собой. Спортивные единоборства – это состязание двух участников, которые соревнуются по правилам соответствующих боевых искусств. Единоборствами называются соревнования, цель которых – выявление лучшего боевого искусства, а также определение чемпионов в конкретных видах состязаний.

Организация соревновательного и подготовительного (тренировочного) периодов – трудоемкий процесс, требующий внимания ко всем своим сферам и нюансам. Одним из наиболее важных аспектов является грамотное распределение спарринг-партнеров. Зачастую оно проводится вручную, что довольно трудоемко и времязатратно. Если неправильно оценить способности и физические данные спортсменом, то можно не только организовать неравноценный поединок, но и создать угрозу здоровью или жизни человека. Для оптимизации процесса организации спаррингов и минимизации рисков необходимо его автоматизировать. Именно поэтому выбранная тема актуальна на сегодняшний день.

**Существующие методы реализации.** На сегодняшний день подбор партнера осуществляется по различным критериям вручную. Обычно либо тренер, либо жюри проводит анализ и оценку состояния спортсмена, а затем подбирает ему подходящую пару, либо же, в некоторых случаях, спортсмены самостоятельно выбирают себе соперника.

Приблизженными по структуре сервисами являются веб-сайты и мобильные приложения, предназначенные для теннисного спорта. В них поиск партнера представлен в формате социальных сетей или сайтов знакомств. Пользователь предоставляет свои спортивные характеристики и ищет второго игрока по необходимым критериям [1, 2]

**Преимущества.** Автоматизированная реализация алгоритма позволит ускорить процесс подбора пары, минимизировать риски нанесения вреда здоровью в следствие невнимательного выбора, оптимизировать процесс организации тренировочной/соревновательной деятельности в целом. На сегодняшний день появилась потребность в создании такой программы, ее концепция была разработана с участием заслуженного тренера России по французскому боксу Буланова Сергея Олеговича и, соответственно, уже имеет практическое назначение.

**Алгоритм.** Построены 2 блок-схемы алгоритма программы, представляющие последовательность распределения спортсменов в пары для тренировочного процесса и для соревновательного. Различие алгоритмов в количестве таблиц для базы данных и параметрах отбора. Для соревновательного периода добавляется «отбор по городам».

В статье представлен алгоритм подбора для тренировочного этапа (рис. 1)

**Среда разработки.** Для реализации проекта требуются базы данных. Создание и работа с ними будут осуществлены в сервисе MySQL [3].

Конкретный сервис был выбран по следующим причинам:

- удобный и простой интерфейс;
- масштабируемость (в системе одновременно может работать много пользователей);
- безопасность (программа работает с персональными данными);
- популярность среди крупных брендов (говорит о качестве и надежности сервиса)
- хорошо интегрируется с Python.

Важным фактором выбора является практический опыт работы со средой и языком программирования.

Программный код реализуется на языке Python в сервисе-интерпретаторе «PyCarm».

В табл. 1 приведено сравнение с другими языками по основополагающим критериям для наглядного объяснения выбора конкретного языка [4, 5].

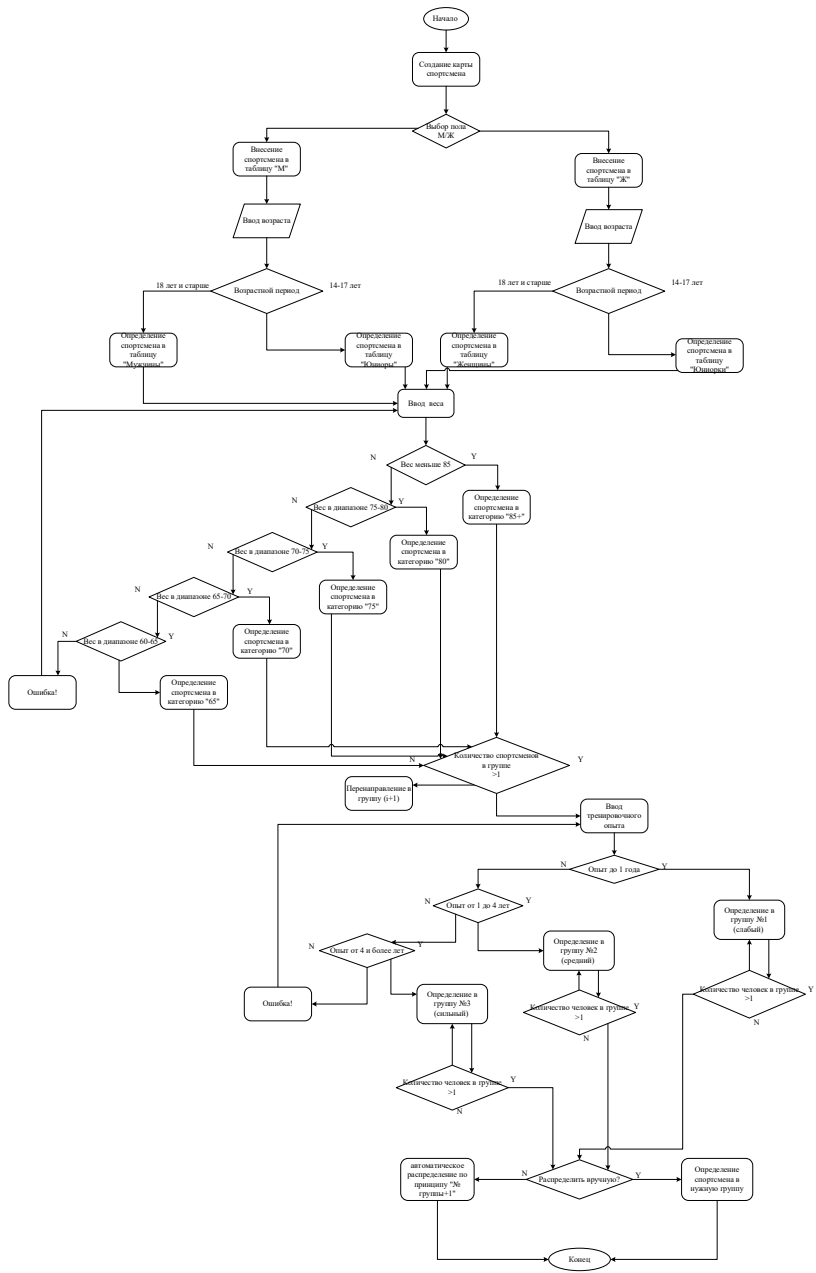


Рис. 1. Алгоритм подбора спарринг-партнера для тренировочного этапа

**Сравнение языков программирования по ключевым критериям**

Параметр	Python	C++	Java
Интерпретируемость	+	–	+
Простота в использовании	+	–	–
Простота поддержки кода	+	–	–
Сфера использования	Работа с данными, машинное обучение	Программирование встроенных систем, разработка игр	Встроенные системы, приложения, веб-сервисы
Производительность	–	+	+

Интерпретируемость играет важную роль, так как предоставляет возможность кроссплатформенности. Простота в использовании и поддержки из-за несложного синтаксиса очень важны для меня, как для начинающего разработчика.

Суть программы – работа с большим объемом данных, соответственно подходящая сфера использования у Python.

В сравнении с двумя другими представленными языками, у Python производительность ниже, но это не критично для первых ступеней реализации продукта. В дальнейшем, возможна его оптимизация. Преимущества выбранного языка соответствуют основным критериям разработчика, а недостатки не критичны для реализации продукта.

**Разработка интерфейса.** Создание пользовательского интерфейса такой же важный компонент реализации программы, как и написание кода. На рис. 2 представлен макет страницы авторизации.

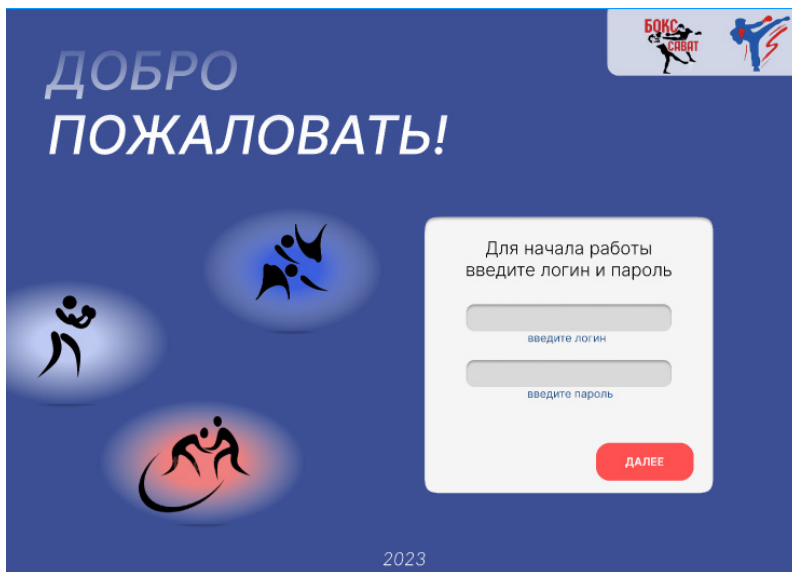


Рис. 2. Макет страницы авторизации



### **Заключение**

Проект разрабатывается для дальнейшего внедрения в организацию тренировочного и соревновательных процессов спортивного клуба «Созвездие». Сейчас работа на стадии разработки алгоритма программы, программный код написан для отдельных блоков и представлены промежуточные результаты.

### **Библиографический список**

1. SETKA. URL: <https://setka.tech/> (дата обращения: 18.04.2023).
2. Tennisgroups. URL: <https://tennisgroups.ru/> (дата обращения: 18.04.2023).
3. SQL простыми словами: основы и применение. URL: <https://boodet.online/blog/sql-prostyimi-slovami-osnovy-i-primenenie> (дата обращения: 18.04.2023).
4. Топ 16 различий между Python и C++. URL: <https://pythonru.com/baza-znaniy/python-ili-c> (дата обращения: 15.04.2023).
5. Python или Java: что лучше для изучения? URL: <https://ru.bitdegree.org/rukovodstvo/python-ili-java> (дата обращения: 15.04.2023).

## УДК 004.4

**Н. И. Чулочникова**

студент кафедры прикладной информатики

**Е. Л. Турнецкая** – кандидат технических наук – научный руководитель**ПРОГРАММНЫЕ СРЕДСТВА СОЗДАНИЯ ВИДЕОКОНТЕНТА В СФЕРЕ EDUTECH**

В мире технологий человеку проще и понятнее посмотреть несколько минут видео, чем прочитать страницу текста. Так, образование на основе видео становится все более популярным с каждым годом. Интерес к использованию видео в обучении растет как среди студентов, так и среди преподавателей. При этом повышаются требования к его качеству. Для современных студентов, выросших в цифровой среде, видео становится предпочтительным средством обучения. С этим согласны почти все респонденты (97%), 94% уверены, что использование видео приводит к росту успеваемости и в целом повышает удовлетворенность слушателей [1].

Использование видеозаписей в учебном процессе – не очередная модная новация, а научно обоснованный путь обновления, средство непрерывного образования. Видео – принципиально новое явление искусства и культуры, резко меняющее все системы учебной информации, создающее альтернативу традиционным моделям обучения.

Видеолекции дают создающему их преподавателю широкий набор средств управления познавательной деятельностью обучаемых и богатые возможности реализации содержания обучения. Особенно полезно использование видеолекций для активизации самостоятельной работы студентов в очном обучении. Ценность видеоконтента заключается в том, что он является более гибким и интересным [2].

С одной стороны, при создании обучающего контента аудио чаще всего работает вместе с видео, и к нему предъявляются технические требования: отсутствие шумов, подходящий уровень громкости, сбалансированные частоты. Эту задачу поможет решить интерактивная видеостудия ГУАП, где уже подготовлено все для записи видеоконтента: шумоизоляция, качественная запись звука и видео.

В видеолекции есть не только говорящая голова, но и визуальное представление текста – презентация. Сегодня презентация – это не просто набор слайдов с представленной текстовой и визуальной информацией, а полноценное средство вовлечения и управления вниманием зрителей. Правильная последовательность слайдов позволяет раскрыть тему и дополнить ее всевозможными визуальными средствами для облегчения восприятия.

В интерактивной видеостудии ГУАП вы взаимодействуете со своей презентацией через экран и пульт управления. Возможности студии позволяют записать видео таким образом, что после окончания записи, файл сразу готов к публикации. Но не у всех получается идеально рассказать текст, кто-то может запнуться или просто оговориться, для этого можно воспользоваться программами для монтажа. Какие же программы выбрать?

Существует множество инструментов для создания и обработки видео-контента, которые помогают сделать обучающие материалы более наглядными, эмоционально привлекательными и интерактивными. Рассмотрим сервисы, предоставляющие решение таких типичных задач, как цветокоррекция, эффекты склейки, добавления плагинов для перехода и чистки звука. На просторах Интернета есть множество статей про программы для монтажа, а какие самые популярные в России проанализировал РБК Life [5].

**1. Adobe Premiere Pro.**

Одна из самых популярных программ для монтажа на сегодняшний день. Premiere Pro является идеальным инструментом для любого процесса пост-обработки или монтажа снятого видео. Существенным преимуществом данной программы является ее свободная интеграция с другими продуктами Adobe. Часто при обработке видео используют связку: Premiere Pro + After Effects + Photoshop. Такая связка программ решает практически все задачи, с которыми может столкнуться профессионал или простой любитель при обработке видеоматериалов. Программа позволяет добавлять всторонние переходы, отчистить звук от шума, выполнить минимальную цветокоррекцию. Но для использования про-

граммы необходима подписка, которую сейчас не так легко приобрести и стоит подписка на полный пакет Adobe не очень дешево.

### 2. Sony Vegas Pro.

Аналогом может стать программа Sony Vegas Pro. Около десяти лет назад Vegas Pro считался главным конкурентом Premiere Pro. Сегодня же о ней говорят редко: программа обновляется не очень быстро и не поспевает за актуальными трендами. Ее основное преимущество перед Premiere Pro — возможность разовой покупки. Немаловажный факт, что разработчиками Vegas Pro является компания Sony – общепризнанный лидер и «законодатель мод» в мире цифрового видео [3]. Главным достоинством программы является ее удобство использования и освоения. Благодаря максимально удобному и интуитивному интерфейсу, этот редактор понравится как профессионалу, так и начинающему. У Vegas Pro есть пара уникальных эффектов. Самый интересный — повышение разрешения видеоматериала при помощи ИИ. Программу можно приобрести один раз и навсегда.

### 3. Movavi Video Editor.

В Movavi Video Editor можно разрезать кадры, работать с несколькими видео- и аудиодорожками, использовать заранее заготовленные переходы и эффекты, добавлять текст. Этого достаточно, чтобы сделать, например, видео для личного блога. Интерфейс программы очень простой: рабочее пространство состоит только из таймлайна, окна предпросмотра и панели с эффектами. Никаких дополнительных окон с маркерами, метаданными и прочими элементами, которые могут запутать новичка. Если интерфейс покажется слишком сложным, при первом запуске программа выводит контекстные подсказки, которые помогут освоиться с функциями. Из минусов в работе практически отсутствует гибкость. Допустим, в Movavi Video Editor можно добавить к видео популярные переходы, но более детально настроить их не получится. В этом и разница между ним и более продвинутыми программами. Программу можно скачать бесплатно, но некоторые функции будут недоступны, приобретение программы возможно по довольно доступной цене.

### 4. DaVinci Resolve.

DaVinci Resolve – «новичок» в индустрии программ для монтажа, но уже на равных конкурирует с лидерами рынка — во многом благодаря нескольким беспрецедентным особенностям. Во-первых, в отличие от Premiere Pro, DaVinci можно купить один раз и навсегда. К тому же есть бесплатная версия – она отличается количеством эффектов и рядом ограничений. DaVinci Resolve – не просто программа, а группа инструментов с классическим монтажным «столом», программой для обработки аудио, работы с эффектами и цветокоррекцией.

Таблица 1

### Сравнение программных средств создания видеоконтента

Программа	Уровень сложности	Возможность цветокоррекции	Редактирование эффектов	Цена
Adobe Premiere Pro	средний	минимальный уровень	+	Adobe приостановила все новые продажи товаров и услуг [4]
Sony Vegas Pro	продвинутый	есть возможность	+	к покупке недоступна, но есть бесплатная версия
Movavi Video Editor	начинающий	нет возможности	–	от 1190 до 4 490 руб. в год в зависимости от функционала
DaVinci Resolve	продвинутый	предназначен для цветокоррекции	+	бесплатно или \$295.00

Проект можно создать с нуля, не прибегая к помощи сторонних решений как у Adobe, где для работы со звуком придется включать Audition, а создание сложной графики не обойдется без After

Effects. Цветокоррекция в DaVinci Resolve считается самой продвинутой в индустрии. Это неудивительно: программа изначально и предназначалась только для цветокоррекции и лишь затем обросла функциями для полноценного монтажа.

На основе таблицы можно сказать, что на данный момент в России доступна для профессионального использования программа DaVinci Resolve и менее сложная к изучению Movavi Video Editor. В итоге, выбор программы для монтажа видео зависит от потребностей и опыта работы с монтажными программами. Важно выбрать подходящую программу, которая позволит решать задачи эффективно и без лишних проблем.

В интерактивной студии ГУАП монтаж происходит в программе Adobe Premier Pro, так как ВУЗ закупил лицензию и срок использования был продлен. Важно, при работе использовать лицензированное программное обеспечение. Это не только законно, но и гарантирует безопасность работы и отсутствие проблем с авторскими правами. Кроме того, лицензированное ПО обеспечивает более высокую производительность и функциональность, что является особенно важным при работе с большими объемами видеоматериалов.

#### **Библиографический список**

1. Каким будет EduTech в 2022 году. URL: [https://sberuniversity.ru/upload/edutech/digest/Digest\\_20.pdf](https://sberuniversity.ru/upload/edutech/digest/Digest_20.pdf) (дата обращения: 06.03.2023).
2. Скобелева И. Е. Создание учебных видеолекций как важная составляющая современного занятия // Вопросы педагогики. 2020. № 10-1. С. 199–203.
3. Холл А., Прокди Р. SONY VEGAS PRO 11. Профессиональный видеомонтаж. М.: Наука и Техника, 2013. 368 с
4. Adobe Creative Cloud. URL: <https://www.adobe.com/ru/creativecloud.html#> (дата обращения: 28.03.2023).
5. 6 лучших программ для монтажа видео. Плюсы, минусы и баги популярных любительских и профессиональных программ. URL: <https://www.rbc.ru/life/news/63037f3e9a7947b023d6399b> (дата обращения: 03.04.2023).

УДК 004.725.5

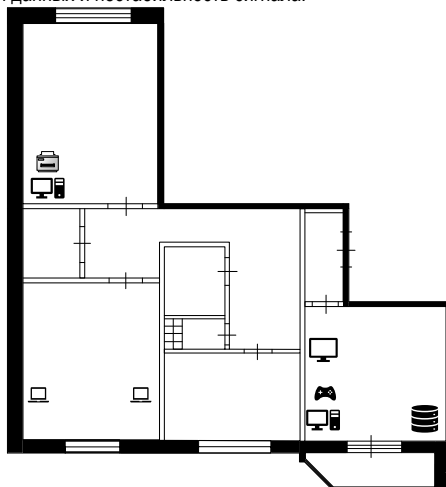
**А. С. Швайко**

студент кафедры прикладной информатики

**Б. К. Акопян** – ассистент – научный руководитель**СОЗДАНИЕ ОДНОРАНГОВОЙ ЛОКАЛЬНОЙ СЕТИ С СЕТЕВЫМ ХРАНИЛИЩЕМ****Введение**

Оптимизация локальной сети назначается для улучшения ее производительности и обеспечения безопасности. Главной задачей данного исследования стала оптимизация локальной сети путем разработки схемы оптимального размещения роутера и кабельного Интернета: необходимо предоставить компьютерам проводное подключение к сети, а Wi-Fi использовать только на мобильных устройствах.

На рис. 1 представлен схематический план квартиры, в которой будут проводиться работы, вместе с размещенными устройствами по комнатам. На начало проведения работ в наличии имеются 4 персональных компьютера, принтер, роутер (маршрутизатор), игровая консоль, умный телевизор, 6 смартфонов, 2 планшета. После произведенной сборки нового компьютера в удаленной комнате от роутера были замечены такие недостатки соединения с сетью, как периодические разрывы соединения, низкая скорость передачи данных и нестабильность сигнала.



*Рис. 1. Размещение устройств в квартире*

Кроме перечисленного списка, в наличии имеется локальное сетевое хранилище (NAS), которое не подключено к интернет-соединению. Его установка в сети в качестве мини-сервера представляется целесообразной, поскольку список задач, выполняемых данным хранилищем огромен: начиная от простого облачного хранилища и заканчивая собственным VPN-сервером. Для этого необходимо восстановить сетевое хранилище и подключить его к разработанной локальной сети.

**Оценка показателей качества текущего расположения локальной сети**

Предваряя реализацию поставленных задач, необходимо проанализировать текущее расположение локальной сети для точной оценки показателей качества. На начало работ локальная сеть со-

стоит из одного роутера для беспроводного подключения и четырех проводов типа витая пара для проводного подключения.

В беспроводной технологии Wi-Fi, в качестве среды распространения сигнала используются радиочастоты, как правило, либо 2,4 ГГц, либо 5,2 ГГц. На открытом пространстве сигнал практически не подвергается негативному воздействию, но проходя через различные препятствия, сильно ослабевает или отражается. Существует таблица потерь эффективности сигнала Wi-Fi при прохождении через различные среды [1].

Для определения показателей качества текущего расположения локальной сети при помощи программы Wi-Fi Scanner [3] были проведены замеры уровня сигнала в разных комнатах. На рис. 2 представлена схема текущего расположения локальной сети относительно всей квартиры, а также точки, из которых проводились замеры.

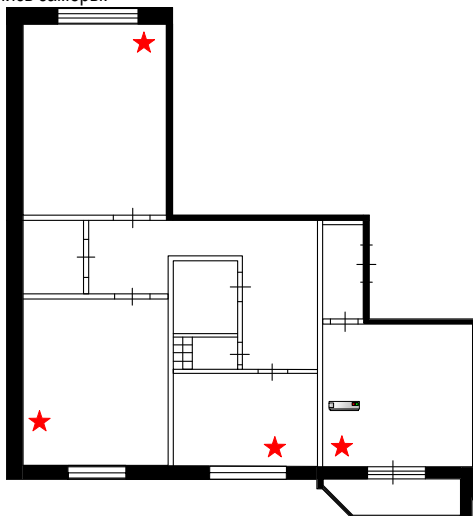


Рис. 2. Текущее расположение сети

Основные характеристики исследуемого роутера [3]:

- Беспроводной интерфейс: 802.11a/b/g/n/ac (по стандарту используется 802.11n).
- Интерфейс LAN: 10/100/1000 BASE-TX Ethernet (4 порта).
- WAN: 1 порт 10/100/1000 BASE-TX Ethernet для подключения кабельного.
- Коэффициент усиления антенны: 2dB (2,4 ГГц) / 2dB (5 ГГц).

Поскольку в квартире реализовано 2 типа связи: на частоте 2,4 ГГц, и на частоте 5,2 ГГц, то необходимо провести ряд измерений как для одной частоты, так и для другой. Результаты измерений приведены в табл. 1–2.

Таблица 1

### Результаты измерений для частоты 2,4 ГГц

Расположение	Уровень мощности сигнала	Качество соединения
Близость роутера	–29 dB	99%
Одна межкомнатная стена	–51 dB	87%
Две межкомнатные стены	–65 dB	80%
Комбинация несущих и межкомнатных	–74 dB	57%

Результаты измерений для частоты 5,2 ГГц

Расположение	Уровень мощности сигнала	Качество соединения
Близость роутера	-43 dB	92%
Одна межкомнатная стена	-57 dB	84%
Две межкомнатные стены	-79 dB	46%
Комбинация несущих и межкомнатных	-85 dB	29%

По данным табл. 1–2 можно заметить, что Wi-Fi сигнал с частотой 5,2 ГГц гораздо более чувствителен к количеству объектов между источником и потребителем. Также видно, что самая дальняя от роутера комната очень слабо покрывается Wi-Fi сигналом, что ожидаемо. Это показывает, что данное расположение роутера не является оптимальным, поскольку сеть Wi-Fi не покрывает всей площади квартиры.

### Восстановление NAS

NAS-сервер, или Network Attached Storage – это система хранения данных, которая предоставляет к ним доступ пользователям и другим устройствам по принятым сетевым протоколам. Среди основных задач таких систем можно выделить [4]:

- Хранение данных, объединение дисков в массивы.
- Предоставление доступа как к общим, так и к отдельным файлам, разделам и папкам для разных групп пользователей.
- Бэкап: объединение жестких дисков в RAID-массивы с возможностью восстановить данные после сбоев и организация дисков в массивы с избыточностью, таким образом защищаясь от удаления данных.
- Предоставление клиентам возможности автономного доступа к документам и файлам сети по различным протоколам. Вследствие этого в NAS-системах есть функции синхронизации данных с устройств, которые к нему подключены, централизованное резервирование, в том числе по расписанию.

Также можно выделить ряд дополнительных функций таких систем: объединение в кластеры, стриминг файлов, организация метаоблака (возможность одновременной работы нескольких пользователей над одним документом в браузере), почтового сервера и веб-сервера.

Процедуру восстановления и введения в эксплуатацию NAS можно разделить на несколько этапов:

#### 1. Выбор операционной системы.

Для сетевого хранилища можно использовать ОС на базе Linux (Ubuntu, Debian) или дистрибутивы – специализированные урезанные операционные системы с минимумом функциональности. Для решения имеющейся задачи были выбраны дистрибутивы, так как они не очень требовательны к аппаратным характеристикам, у них нет собственной графической среды, но есть веб-интерфейс администратора, чего достаточно. В процессе реализации поставленной цели возникла проблема с подбором дистрибутива для локального хранилища, поскольку аппаратная часть хранилища не поддерживает множества функций, созданных в новых дистрибутивах. Путь решения состоял в том, чтобы опытным путем найти такой дистрибутив, который подошел бы к серверу.

К дистрибутиву, помимо вышеперечисленных, также предъявляются следующие требования: простота установки (как на жесткий диск, так и на USB-накопитель, с которого будет загружаться ОС в NAS), малый объем занимаемого места в памяти, предоставление доступа к файлам по большому списку протоколов и функция защиты данных.

Существует несколько популярных операционных систем для NAS: FreeNAS, XigmaNAS, EasyNAS, OpenMediaVault и DiskStation Manager (DSM). Среди всего этого списка операционных систем была выбрана DSM, поскольку она не только соответствует предъявляемым требованиям к дистрибутиву, но и имеет утилиту Synology Assistant, благодаря которой можно производить поиск хранилища в локальной сети.

2. Установка ОС и подключение NAS к сети.

Для подготовки к процедуре установки ОС на NAS система вначале устанавливается на USB-накопитель объемом не менее 8 Гб. В процессе установки требуется изменить параметры номера разработчика устройства VID и номера устройства флешки PID. Эти два параметра можно найти через «Диспетчер устройств», как показано на рис. 3.

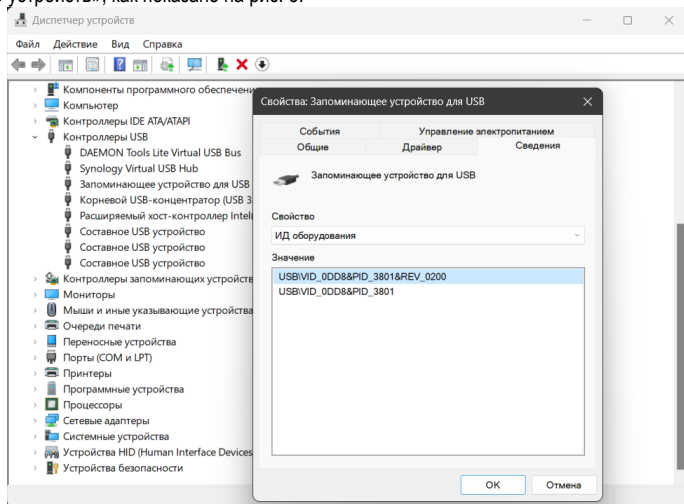


Рис. 3. Параметры флешки

Следующим этапом необходимо установить виртуальный образ диска с необходимым дистрибутивом, в котором требуется изменить параметры VID и PID в конфигурационном файле. Для этого образ диска был демонтирован и снова смонтирован после изменения значений параметров при помощи программы OSF Mount. После завершения работы над образом диска USB-накопитель преобразован в загрузочную флешку при помощи программы Rufus. Далее загрузочная флешка вставляется в сетевое хранилище, после чего на экране появится выбор типа загрузки, как это показано на рис. 4.

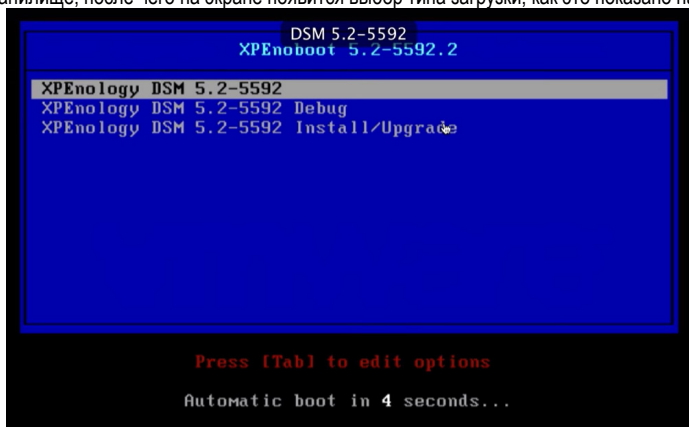


Рис. 4. Выбор типа загрузки



В данном окне необходимо выбрать последнюю строку для установки дистрибутива. После выбора данного способа начнется процесс установки дистрибутива, после завершения которого в приложении Synology Assistant впервые появится информация о сетевом хранилище. Из окна этого приложения необходимо подключиться к серверу и действовать по подсказкам далее.

### 3. Подключение к хранилищу.

После завершения установки дистрибутива на локальное хранилище к нему необходимо произвести подключение. Для этого в адресной строке любого браузера вводится ip-адрес NAS, который можно узнать при помощи Synology Assistant. И после авторизации открывается рабочий стол хранилища (рис. 5).

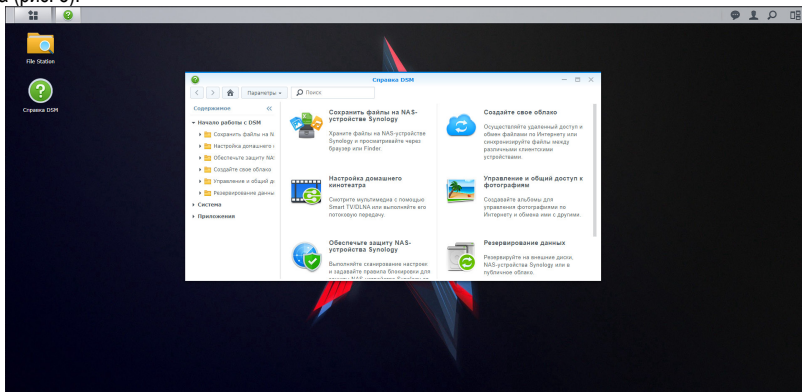


Рис. 5. Вход в хранилище

При первом входе на рабочий стол NAS на нем появляется информационная справка по настройке работы хранилища, в частности, по добавлению папки сетевого расположения в Проводник. На рис. 6 показано добавление сетевого адреса некоторой папки, находящейся на хранилище.

#### Укажите расположение вашего веб-сайта

Введите адрес веб-сайта, FTP-сайта или сетевой адрес общего ресурса, который можно будет открыть с помощью этого ярлыка.

Сетевой адрес или адрес в Интернете:

[Показать примеры](#)

Рис. 6. Добавление сетевого расположения

Если папка существует и к ней есть доступ, она добавится в Проводник и, если компьютер подключен к сети, подключение к этой папке будет автоматическим.

### Составление оптимальной схемы размещения

Для реализации данной задачи стоит определиться с классификацией по уровню однородности сети. Поскольку по масштабу создаваемая сеть будет домашней сетью и все компьютеры в сети занимают одинаковое положение, то по уровню однородности данная сеть будет одноранговой.

Далее необходимо определиться с ее топологией. Сетевая топология – это конфигурация графа, вершинами которого являются конечные узлы сети (компьютеры или коммуникационное оборудование), а ребрами – физические или информационные связи между вершинами [5]. Все множество топологий делится на: полностью связную, когда каждый компьютер связан друг с другом, неполностью связную, когда связь между компьютерами осуществляется через дополнительные узлы, и смешанную. Поскольку с экономической точки зрения полностью связная топология не оптимальна, а неполностью связная не покрывает всех потребностей, то выбор пал на топологию смешанного типа. В таких сетях можно выделить подсети, имеющие типовую топологию. При этом подсети между собой соединены посредством дополнительных устройств. Исходя из этого была разработана схема оптимальной сети для данной задачи, которая представлена на рис.7.

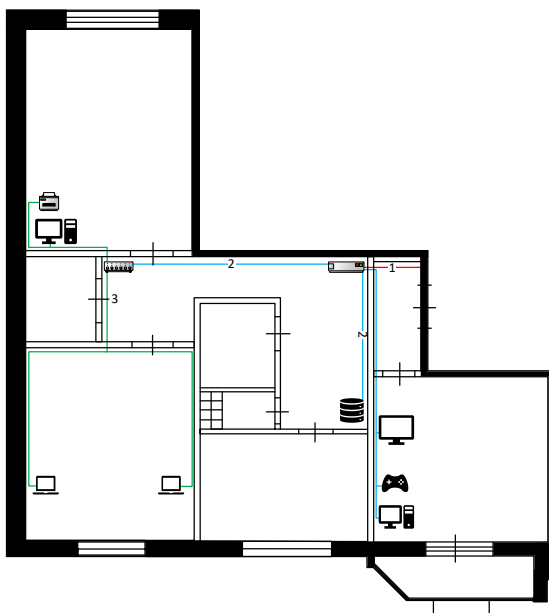


Рис. 7. Оптимальная сеть (провод 1 – входной; провод 2– основная сеть; провод 3 – подсеть)

Как видно из данного рисунка, в топологии данной сети можно выделить одну подсеть типа «звезда» [5], поскольку каждый компьютер подключен кабелем к коммутатору, который, в свою очередь, подключен к остальной сети. Данные от передающей станции (маршрутизатора) передаются через коммутатор на потребителей (синий провод). Таким образом информация поступает и принимается только теми абонентами, которые этого требуют.

Соединение основной сети (роутера) с подсетью (ноутбуками и компьютером) осуществляется через соединительное устройство – коммутатор. И для того, чтобы подходить к условиям созданной сети, он должен обладать следующими характеристиками:

- Количество LAN-портов: не менее 5.
- Базовая скорость передачи данных: 1 Гбит/с.

- Тип управления: неуправляемый (поскольку управление им не будет осуществляться).
- Сетевые стандарты: подходит под стандарты IEEE.

В качестве кабелей подключения устройств к сети будут выступать кабели типа «витая пара», поскольку разъем RJ-45, который является входным портом для данного типа кабелей, на данный момент более универсален, в отличие от оптоволоконного кабеля.

### **Заключение**

Подводя итог проделанной работы, можно сказать, что изначальная проблема с ненадежным интернет-соединением компьютера была решена после составления оптимальной схемы размещения. Оптимальность схемы заключается в том, чтобы ни один из компьютеров не был подключен по Wi-Fi соединению, а оставить Wi-Fi только на мобильные устройства.

Кроме этого, в созданную локальную сеть подключено локальное сетевое хранилище. Оно будет применяться не только как облачное хранилище файлов, но и как мультимедийный сервер, например, для просмотра фильмов, фотографий.

Реализация поставленной цели позволила получить базовые знания по созданию локальных сетей, которые можно применять в качестве практической основы при осуществлении масштабных проектов.

### **Библиографический список**

1. Коэффициенты затухания сигнала Wi-Fi при прохождении через различные среды. URL: <https://click.ru/HCАvu> (дата обращения: 26.02.2023).
2. Wi-Fi Scanner // LizardSystems.com. URL: <https://lizardsystems.com/wi-fi-scanner/> (дата обращения 26.02.2023).
3. DIR-850L. URL: <https://www.dlink.ru/products/2/1846.html> (дата обращения: 14.05.2023).
4. Дистрибутивы для NAS-сервера. URL: <https://selectel.ru/blog/nas-distributions/#EasyNAS> (дата обращения: 28.02.2022).
5. Сетевая топология. URL: [https://network.xsp.ru/top\\_net.php](https://network.xsp.ru/top_net.php) (дата обращения: 08.04.2023).

УДК 004.942

**О. А. Буслова**

студент кафедры информационных систем и технологий

**А. П. Шенета** – доктор технических наук, профессор – научный руководитель**МОДЕЛИРОВАНИЕ ЛОГАРИФИЧЕСКИ НОРМАЛЬНОГО АНИЗОТРОПНОГО ПОЛЯ****Введение**

Требования к разрешающей способности радиолокационных комплексов по координатам становятся все более строгими с каждым годом. Это связано с расширением возможностей применения высокоточных бортовых РЛС в различных областях, таких как мониторинг экологической обстановки, геологоразведка, поиск объектов в зонах бедствий и т. д. Поэтому очень важно совершенствовать методы, которые повышают разрешающую способность, включая использование сложных сигналов и методов синтеза апертуры антенны для создания высокоточного радиолокационного изображения [1]. Это является актуальной задачей.

Особенности синтеза нелинейных многоканальных дискретных формирующих фильтров

Рассматриваемые методы [2, 3] можно естественным образом обобщить на моделирование анизотропных гауссовских и негауссовских случайных полей в пространственных координатах. Единственным ограничением для прямого использования предложенного метода синтеза является факторизация корреляционных функций по пространственным координатам для порождающего негауссовского поля, соответствующего гауссовскому полю (в анизотропном случае). Однако это ограничение не является существенным с практической точки зрения, так как результаты экспериментальных исследований полей часто представлены сечениями их пространственно-корреляционной функции в каждой координате и по времени. Таким образом, введенное ограничение всегда позволяет построить статистическую модель поля, соответствующую экспериментальным данным. Следует отметить, что предположение факторизации пространственно-временной корреляционной функции поля относится только к порождающему гауссовскому полю. Моделируемое негауссовское поле может иметь нефакторизируемую пространственно-временную корреляционную функцию. Более того, порождающее гауссовское поле в практических приложениях часто может быть разложено на ортогональные составляющие. Факторизация является естественной и не ограничивает задачу моделирования порождающего поля.

При синтезе НМДФФ [3] сделано предположение о равенстве нормированных временных статистических характеристик в канальных формирующих фильтрах. Это предположение существенно повышает скорость работы алгоритмов моделирования и не является слишком ограничительным для практического использования синтезированных алгоритмов, поскольку оно часто выполняется во многих практических случаях без явного указания. Поэтому, при синтезе алгоритмов моделирования полей будет считаться это предположение выполненным. В то же время, следует отметить, что средние значения и дисперсии процессов могут изменяться, и это учитывается в практических ситуациях. По аналогии, предполагается одинаковость нормированных статистических характеристик по каждой пространственной координате. Поставлено ограничение явным описанием алгоритма моделирования для двумерных полей.

**Описание алгоритма**

Для описания введены следующие параметры:  $\xi$  – нормальные совместно независимые случайные (псевдослучайные) величины с нулевыми значениями средними и единичными дисперсиями;  $V$  – элементы негауссовского поля;  $U$  – элементы порождающего поля;  $r^{(X)}, r^{(Y)}, r^{(Z)}$  – корреляционные функции;  $f(\cdot)$  – функциональное преобразование [4].

Алгоритм моделирования трехмерного поля может быть представлен в следующем виде:

$$\left\{ \begin{array}{l} X_{i,j,t} = -\sum_{l=1}^{i-1} (D_{l,i}^{(X)} / D_{l,j}^{(X)}) \cdot X_{l,j,t} + \sqrt{D_i^{(X)} / D_{i-1}^{(X)}} \cdot \xi_{i,j,t}, \quad i=1,2,\dots,M^{(X)}, \\ Y_{i,j,t} = -\sum_{p=1}^{j-1} (D_{p,j}^{(Y)} / D_{p,p}^{(Y)}) \cdot Y_{i,p,t} + \sqrt{D_j^{(Y)} / D_{j-1}^{(Y)}} \cdot X_{i,j,t}, \quad j=1,2,\dots,M^{(Y)}, \\ U_{i,j,t} = \sum_{s=1}^N a_s \cdot U_{i,j,t-s} + \sum_{d=0}^{N-1} b_d \cdot Y_{i,j,t-d}, \\ \tilde{V}_{i,j,t} = f(\bar{U}_{i,j,t} + \sigma_{i,j,t} \cdot U_{i,j,t}) = \bar{U}_{i,j,t} \cdot \exp(\sigma_{i,j,t} \cdot U_{i,j,t}), \end{array} \right. \quad (1)$$

Алгоритм (1) удобно интерпретировать в следующем виде:

1. Верхнее выражение преобразует вектор  $\xi_{M^{(X)},t} = (\xi_{1,t}, \xi_{2,t}, \dots, \xi_{M^{(X)},t})$  в вектор  $\mathbf{X}_{M^{(X)},t} = (X_{1,t}, X_{2,t}, \dots, X_{M^{(X)},t})$ , но с нормированной корреляционной матрицей, независимой от значений  $t$ .  $t = \dots, -3, -2, -1, 0, 1, 2, 3, 4, \dots$ ,  $D_1^{(X)} = D_0^{(X)} = 1$ ,  $D_l^{(X)} = D_{l-1}^{(X)}$ ,  $l = 1, 2, \dots, M^{(X)}$ . В свою очередь  $D_{M^{(X)}}^{(X)} = \left\| D_{M^{(X)}}^{(X)} \right\|$  и является нормированной корреляционной матрицей каждого из векторов  $\mathbf{X}_{M^{(X)},t}$

$$D_l^{(X)} = \begin{pmatrix} 1 & r_1^{(X)} & r_2^{(X)} & \dots & r_{l-1}^{(X)} \\ r_1^{(X)} & 1 & r_1^{(X)} & \dots & r_{l-2}^{(X)} \\ r_2^{(X)} & r_1^{(X)} & 1 & \dots & r_{l-3}^{(X)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{l-1}^{(X)} & r_{l-2}^{(X)} & r_{l-3}^{(X)} & \dots & 1 \end{pmatrix}. \quad (2)$$

Данное преобразование представлено следующим образом:

$$\xi_{M^{(X)},M^{(Y)}}^{D^{(X)}} \Rightarrow \mathbf{X}_{M^{(X)},M^{(Y)},t}. \quad (3)$$

Блок-схема выражения представлена на рис. 1.

2. Второе выражение в (1) при каждом фиксированном  $t$  преобразует матрицу  $\mathbf{X}_{M^{(X)},t} = (X_{1,t}, X_{2,t}, \dots, X_{M^{(X)},t})$  в стохастическую матрицу  $\mathbf{Y}_{M^{(Y)},t} = (Y_{1,t}, Y_{2,t}, \dots, Y_{M^{(Y)},t})$ . Математические ожидания и дисперсии элементов матрицы равны нулю и единице соответственно, все элементы распределены по нормальному закону.

Данное преобразование представлено следующим образом:

$$\mathbf{X}_{M^{(X)},M^{(Y)},t} \xRightarrow{D_{M^{(Y)}}^{(Y)}} \mathbf{Y}_{M^{(X)},M^{(Y)},t}. \quad (4)$$

Блок-схема выражения представлена на рис. 2.

3. Следующее выражение в (1) преобразует каждый из элементов  $\mathbf{X}_{M^{(X)},t} = (X_{1,t}, X_{2,t}, \dots, X_{M^{(X)},t})$  в  $\mathbf{U}_{i,t} = (\dots, -2, -1, 0, 1, 2, 3, \dots)$  с помощью линейного дискретного формирующего фильтра [5-7] с коэффициентами  $\mathbf{a}_N, \mathbf{b}_N$ , которые определяют нормированную временную функцию корреляции последовательности

$$r^{(B)}(t_1, t_2) = r^{(B)}(|t_1 - t_2|) = r^{(B)}(\tau)$$

для каждого из элементов матрицы. Полученные значения представляют собой поля порождающих нормальных процессов [8], которые используются для создания эхо-сигналов, полученных в результате отражения.

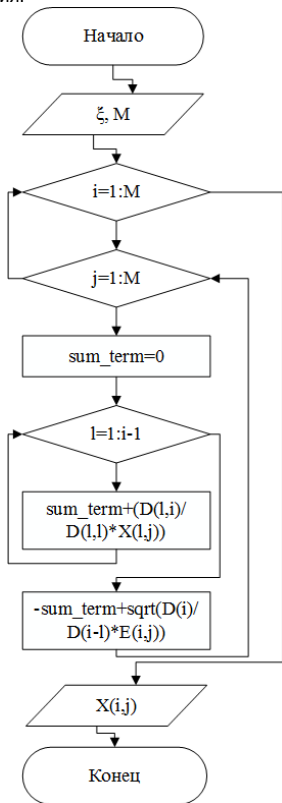


Рис. 1. Блок-схема расчета  $X_{M^{(x)}, M^{(y)}} t$

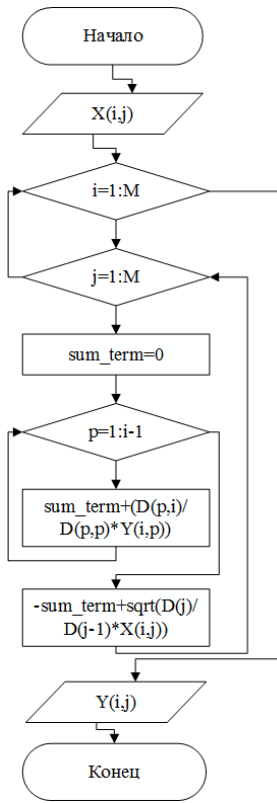


Рис. 2. Блок-схема расчета  $Y_{M^{(x)}, M^{(y)}} t$

Данное преобразование представлено следующим образом:

$$Y_{M^{(x)}, M^{(y)}} t \stackrel{(a_N, b_N)}{\Rightarrow} U_{M^{(x)}, M^{(y)}} t \quad (5)$$

Блок-схема выражения представлена на рис. 3.

4. Выражение в последней строчке формулы (1) определяет негауссовский векторный процесс

$$\tilde{V}_{M^{(x)}} t,$$

где  $(\bar{U}_{M^{(x)}} t, \alpha_{M^{(x)}} t) = (\bar{U}_{1,t}, \bar{U}_{2,t}, \dots, \bar{U}_{M^{(x)},t}; \sigma_{1,t}, \sigma_{2,t}, \dots, \sigma_{M^{(x)},t})$ .

Данное преобразование представлено следующим образом

$$\mathbf{U}_{M^{(X)}, M^{(Y)}, t} \stackrel{f(\cdot, \bar{\mathbf{U}}_{M^{(X)}, M^{(Y)}, i}, \boldsymbol{\alpha}_{M^{(X)}, M^{(Y)}, i})}{\Rightarrow} \tilde{\mathbf{V}}_{M^{(X)}, M^{(Y)}, t}, \quad (6)$$

Блок-схема выражения представлена на рис. 4.

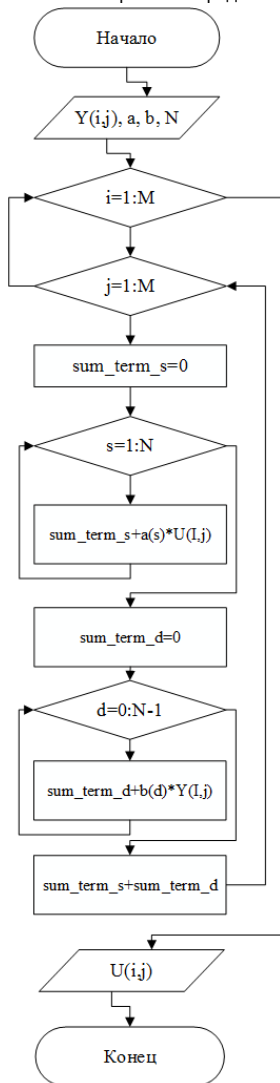


Рис. 3. Блок-схема расчета  $\mathbf{U}_{M^{(X)}, M^{(Y)}, t}$

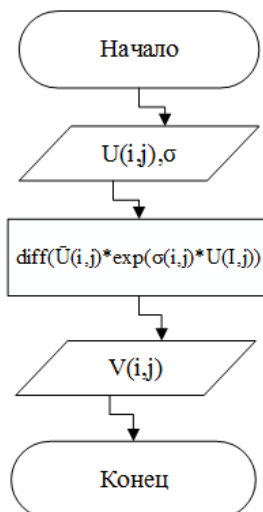


Рис. 4. Блок-схема расчета  $\tilde{\mathbf{V}}_{M^{(X)}, M^{(Y)}, t}$

Из вышесказанного следует, что алгоритм (1) можно представить в виде

$$\xi_{M^{(X)}, M^{(Y)}, t} \begin{matrix} \mathbf{D}_{M^{(X)}}^{(X)} \\ \Rightarrow \mathbf{X}_{M^{(X)}, M^{(Y)}, t} \\ \Rightarrow \mathbf{U}_{M^{(X)}, M^{(Y)}, t} \end{matrix} \begin{matrix} \mathbf{D}_{M^{(Y)}}^{(Y)} \\ \Rightarrow \mathbf{Y}_{M^{(X)}, M^{(Y)}, t} \\ \Rightarrow \tilde{\mathbf{V}}_{M^{(X)}, M^{(Y)}, t} \end{matrix} \Rightarrow \begin{matrix} (\mathbf{a}_N, \mathbf{b}_N) \\ f(\cdot, \bar{\mathbf{U}}_{M^{(X)}, M^{(Y)}, i} \diamond \mathbf{M}_{M^{(X)}, M^{(Y)}, i}^{(X)}) \end{matrix} \quad (7)$$

Важно отметить, что приведенный алгоритм дает возможность моделировать поле с анизотропными свойствами, которые изменяются в зависимости от пространственных координат, что соответствует физическим явлениям, рассматриваемым в данном контексте. При этом к пространственным корреляционным матрицам предъявляется одно естественное требование – положительная определенность.

### Практическое применение

Флуктуации эхо-сигнала от поверхности земли или моря могут быть представлены в виде двухмерного случайного поля, в то время как отражения от гидрометеоров образуют трехмерное поле. При наблюдении этих сигналов в стробе приемного устройства, все типы флуктуаций могут быть представлены как одномерное поле или вектор на выходе приемника.

Разработанная блок-схема позволяет визуально представить последовательность шагов и процессов, осуществляемых в рамках алгоритма, что упрощает его реализацию и анализ. Это полезный инструмент для исследователей, разработчиков и практиков, работающих с данным алгоритмом, и может способствовать его более эффективному использованию в различных областях применения.

### Библиографический список

1. Isakov V., Shepeta D. Simulation of location signals when determining a coastal edge // Wave Electronics and its Application in Information and Telecommunication Systems (WECONF). 2021.
2. Ненашев В. А., Сенцов А. А., Куюмчев Г. В. Моделирование процесса формирования радиолокационного изображения высокого разрешения в бортовых РЛС // Вопросы радиоэлектроники. 2013. Т. 2. № 3. С. 48–56.
3. Изранцев В. В., Шепета Д. А. Моделирование внешних сигналов бортовых приборных комплексов летательных аппаратов пятого поколения // Изв. Вузов. Приборостроение. 2000. № 2. С. 76–83.
4. Isakov V. I., Shepeta D. A. Simulation of location signals when determining a coastal edge // Wave Electronics and its Application in Information and Telecommunication Systems (WECONF). 2021. P. 1–5.
5. Shepeta A. P., Nenashev V. A. Modeling algorithm for SAR image based on fluctuations of echo signal of the Earth's surface // Proc. of SPIE Remote Sensing, Vol. 9642. Toulouse, France. 2015. P. 96420X-1 – 96420X-8.
6. Ваттимена Г. М. Математические модели локационных сигналов, отраженных от подстилающих поверхностей земли и моря. Моделирование PCA изображений для задач мониторинга экосистем // Современные исследования социальных проблем (электронный научный журнал). 2017. Т. 8. № 4-2. С. 49–58.
7. Kevin Poliano, Marianne Clausel, Valérie Perrier. Texture modeling by Gaussian fields with prescribed local orientation // IEEE International Conference on Image Processing (ICIP). Oct. 2014.
8. Буре В. М., Парилина Е. М. Теория вероятностей и математическая статистика. СПб.: Лань, 2013. С. 416.



УДК 004.05; 004.8

**Ю. В. Ветрова**

студент кафедры информационных систем и технологий

**В. В. Фомин** – профессор, доктор технических наук – научный руководитель

## ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ПРОГРАММНЫХ РЕШЕНИЙ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ В РАМКАХ СВОБОДНО РАСПРОСТРАНЯЕМЫХ АНАЛИТИЧЕСКИХ ПЛАТФОРМ

### Введение

Анализ данных и машинное обучение тесно связаны между собой и являются важными инструментами для работы с большими объемами информации [1]. Примером использования машинного обучения в анализе данных может быть задачи классификации объектов, прогнозирования процессов, распознавания изображений и т. д. [2 – 3]. Существует большое количество программных решений для анализа данных, включающих в себя различные методы машинного обучения.

В существующем многообразии инструментария анализа данных становится актуальными технологии оценки качества и эффективности программных решений методов машинного обучения.

Целью работы является анализ и оценка эффективности программных решений алгоритмов машинного обучения в рамках свободно распространяемых аналитических платформ.

### Постановка задачи

В рамках исследования рассмотрена одна из основных задач машинного обучения – задача классификации. Исследование эффективности решения выбранной задачи машинного обучения выполнялось с использованием алгоритма k-ближайших соседей (KNN), метода опорных векторов (SVM) и нейронной сети (NN) [4 – 6].

Для оценки эффективности методов классификации предлагается интегральный подход, в основе которого лежат три модели оценки качества:

1. Классическая оценка.

Определение временных затрат на решение поставленной задачи. Время классическая метрика ресурсоемкости алгоритма

2. Математическая оценка.

Выявление особенностей реализации алгоритмов со стороны математического аппарата используя метрики: ошибка классификации, precision, F-мера, recall, ROC, AUC [7].

3. Эргономика программных продуктов.

Данный вид оценки определяет эффективность внедрения, обучения и скорости манипуляции пользователей. Оценка эргономики программных решений осуществляется на основе метода GOMS и закона Хика [8].

В качестве программных решений были выбраны популярные аналитические платформы: KNIME, RapidMiner и Orange [9 – 11].

Для полной оценки каждого алгоритма классификации в табл. 1 представлена краткая характеристика выбранных наборов данных.

Таблица 1

### Характеристика наборов данных

Набор данных	Количество атрибутов	Количество строк (векторов)	Объем информации	Тип файла
HeartDisease	12	918	35 Кб	CSV
Water quality	21	7999	812 Кб	CSV
NASA	10	9998	1033 Кб	CSV

### Результаты

Для оценки эффективности алгоритмов классификации на аналитические платформы был проведен ряд экспериментов. В каждом эксперименте было проведено сравнение эффективности, на основе полученных метрик и характеристик, одного из трех алгоритмов классификации с одним набором данных на трех аналитических платформах.

В ходе проведения экспериментов была выполнена оценка методов машинного обучения. При сравнительном анализе алгоритмов машинного обучения были замечены следующие тенденции.

На рис. 1 – 2 представлены диаграмма оценки и временных затрат алгоритма  $k$  – ближайших соседей.

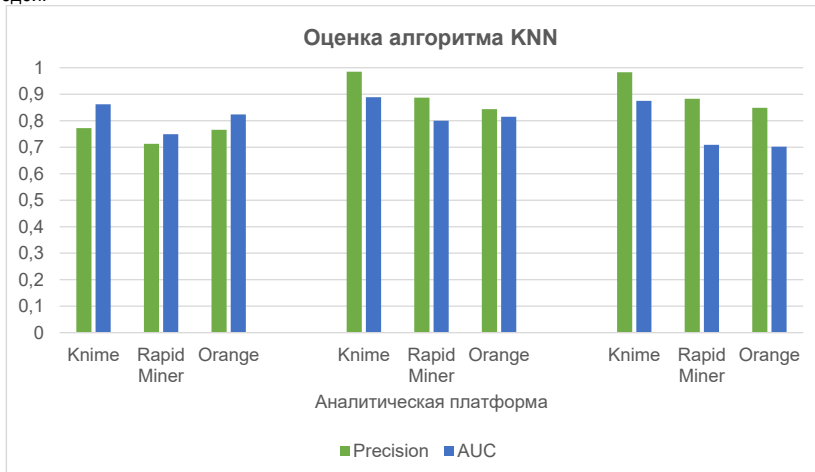


Рис. 1. Диаграмма оценки алгоритма KNN

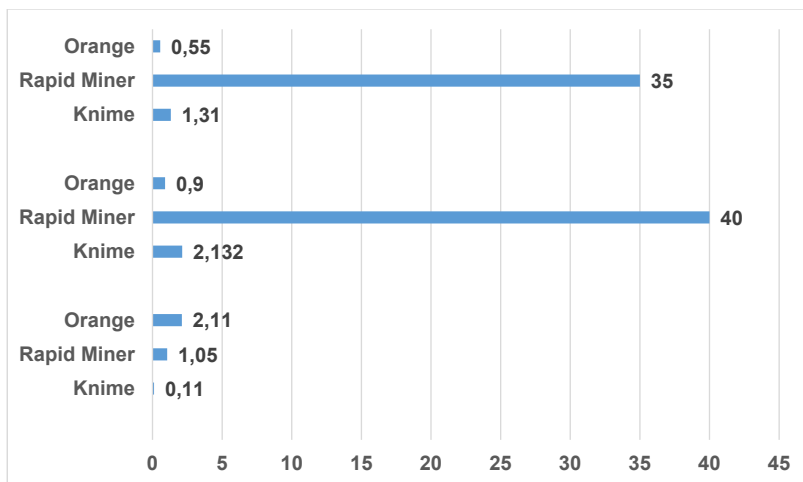


Рис. 2. Время работы алгоритма KNN, сек

Алгоритм KNN для трех наборов данных более качественно выполняет задачу классификации на платформе Knime. Важно отметить, что точность и производительность алгоритма возрастает с объемом набора данных.

Немного по-другому ведет себя алгоритм SVM. Такие параметры как точность и AUC показывают наивысший результат для трех наборов данных на разных платформах. На рис. 3 – 4 представлена диаграмма оценки и временных затрат метода опорных векторов, соответственно.

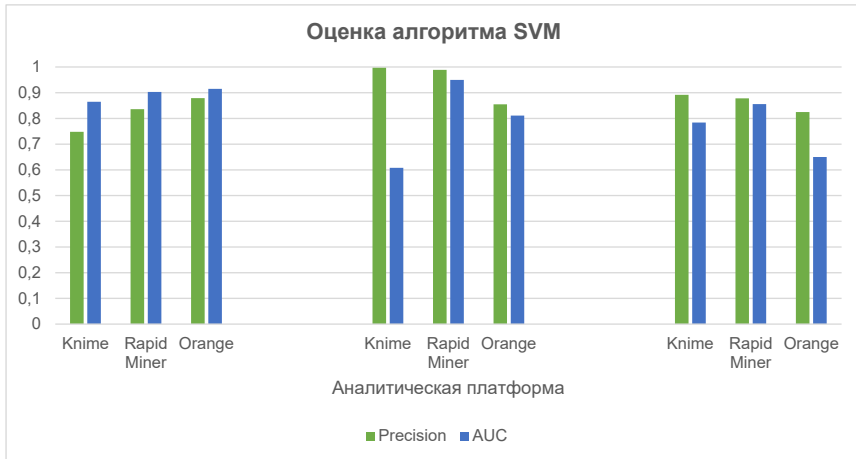


Рис. 3. Диаграмма оценки алгоритма SVM

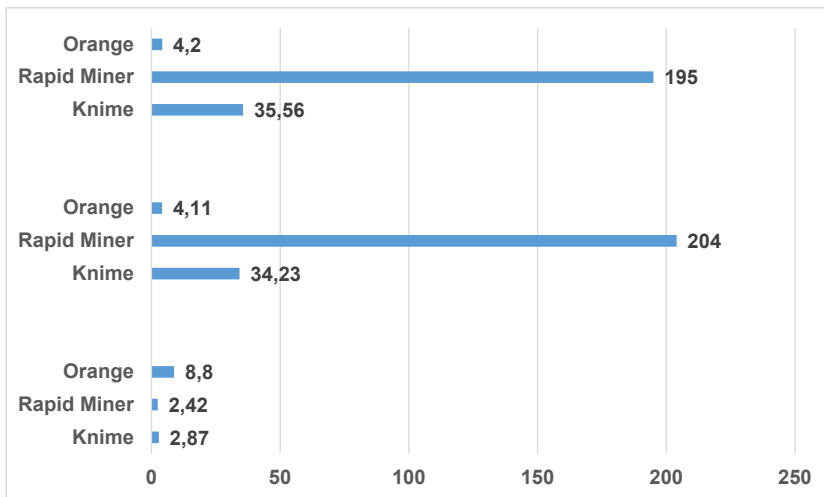


Рис. 4. Время работы алгоритма SVM, сек

Стоит отметить то, что классификация методом SVM дает высокую оценку всех наборов данных на платформе RapidMiner.

На рис. 5 – 6 представлена диаграмма оценки классификации методом NN.

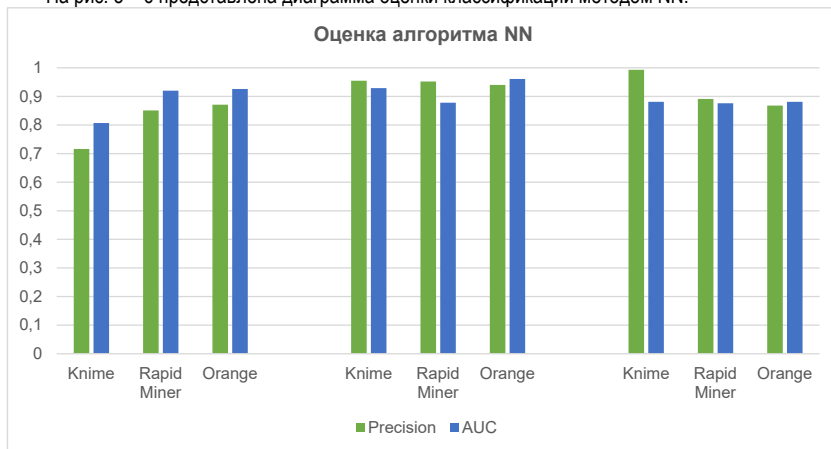


Рис. 5. Диаграмма оценки алгоритма NN

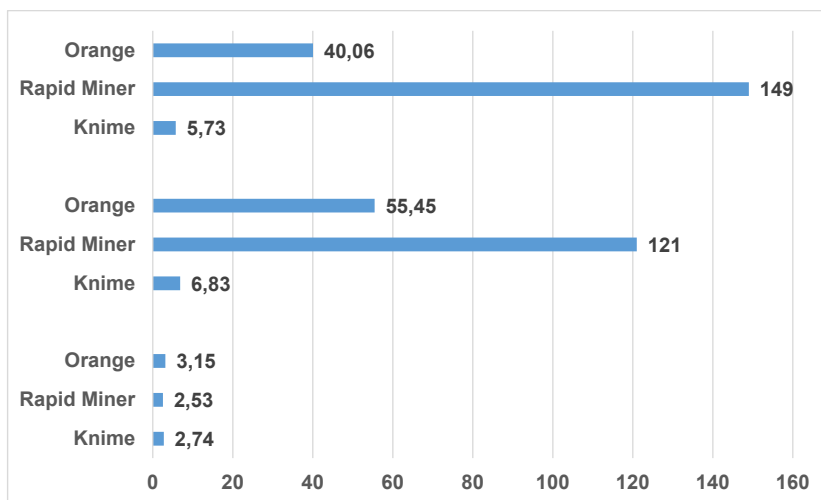


Рис. 6. Время работы алгоритма NN, сек

Выше представленные диаграммы показывают, что на каждой платформе получены достаточно высокие показатели качества классификации.

#### Оценка эргономики интерфейса

Согласно методике GOMS и закону Хика, была произведена численная оценка качества интерфейсов аналитических платформ Orange, Knime и RapidMiner. Табл. 2 содержит информацию о общем времени выполнения задач на каждой аналитической платформе, сгруппированных по методам оценки интерфейсов. График сравнения результатов представлен на рис. 7.

## Общее время выполнения задач

Оценка интерфейса	Orange	Knime	RapidMiner
Метод GOMS	69,3 сек.	141,55 сек.	139,7 сек.
Закон Хика	1,16 сек.	1,66 сек.	2,19 сек.

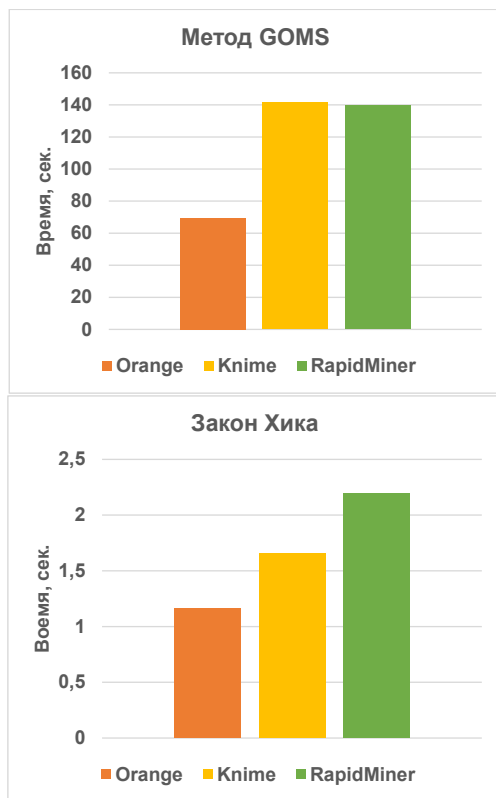


Рис.7. Диаграмма затраченного времени

По диаграммам видно, что для выполнения двух задач опытный пользователь значительно меньше тратит время на платформе Orange, чем на двух других платформах. Так для построения модели классификации методом SVM он затратил 69,3 сек., что в два раза меньше, чем на платформе Knime (141,55 сек.).

По закону Хика разница между временем принятия решения незначительна среди рассмотренных платформ. Наилучшую оценку интерфейса по полученным результатам также получает платформа Orange (1,16 сек.).

Стоит отметить, что платформы Knime и RapidMiner имеют схожий интерфейс. Разницу во времени принятия решений и построения модели, по сравнению с платформой Orange, можно объяснить

тем, что обе платформы имеют больше объектов для моделирования, так как нацелены на более полное, глубокое и профессиональное решение задач машинного обучения.

### Выводы

С учетом затраченного времени на выполнения алгоритма, для классификации маленького объема данных лучше всего использовать платформу RapidMiner. Для наборов данных среднего и большого объема безусловным лидером по качеству классификации и минимальному затраченному времени является платформа Kнime.

Так как исходные данные и настройки моделей для каждого алгоритма машинного обучения были одинаковы, то отличие в показателях можно объяснить различием заложенного математического аппарата для каждого метода МО внутри аналитических платформ.

Согласно проведенной количественной оценке качества интерфейса, для работы и решения задач, связанных с классификацией данных, удобным, понятным и более нацелен на неопытного пользователя со стороны эргономики интерфейса, является аналитическая платформа Orange.

### Библиографический список

1. Поручиков М. А. Анализ данных: учеб. пособие. Самара. С. 88–2016.
2. Сравнительный анализ методов машинного обучения для решения задачи классификации документов научно-образовательного учреждения / М. Н. Краснятский, А. Д. Обухов, Е. М. Соломятина, А. А. Воякина // Вестник ВГУ. Тамбов. 2018. С. 173–182.
3. О разработке инструментария интеллектуального анализа данных / В. В. Лаптев, А. В. Флегонтов, В. В. Фомин // Информатизация образования и науки. СПб. 2022. С. 121–138.
4. Бизнес аналитика от данных к знаниям: учеб. пособие / Н. Палкин, В. Орешков. 2013.
5. Курочкин С. В. Нейронные сети: просто о сложном. Теория вероятностей и ее применения. URL: <http://www.tvp.ru/> (дата обращения: 17.04.2023).
6. Соколов Е. А. Семинары по выбору моделей. М., 2015.
7. Сравнительный анализ алгоритмов машинного обучения для проведения классификации сетевого зашифрованного трафика / Д. В. Костин, О. И. Шелухин. 2016.
8. Джеф Раскин. Интерфейс. Новые направления в проектировании компьютерных систем. М.: Символ, 2005. 272 с.
9. Rapid Miner. URL: <https://rapidminer.com/> (дата обращения: 25.09.2022).
10. KNIME. URL: <https://www.knime.com/> (дата обращения: 25.09.2022).
11. Orange. URL: <https://orangedatamining.com/> (дата обращения: 25.09.2022).

УДК 378.1

**Е. Д. Григорьева**

студентка кафедры информационных систем и технологий

**В. А. Ушаков** – старший преподаватель – научный руководитель

## ПРИМЕНЕНИЕ ЦИФРОВЫХ ТЕХНОЛОГИЙ В СФЕРЕ ОБРАЗОВАНИЯ

### Введение

В современном мире технологии развиваются с «космической» скоростью, а вместе с ними меняется и сознание человека. Цифровые технологии окружают нас везде. Они стали неотъемлемой частью жизни каждого из нас, и их значимость и влияние на человека стремительно растет с каждым годом. Молодое поколение, знакомое с цифровыми технологиями с детства, воспринимает информацию уже совершенно иначе. Привычные нам учебники, пособия и книги стремительно переходят в цифровой формат, и это облегчает восприятие информации и заинтересовывает современных учеников. А эффективная мотивация школьника или студента к обучению — один из наиболее действенных способов повысить качество и уровень образования. Поэтому применение цифровых технологий в разных сегментах сферы образования является одним из перспективных и быстроразвивающихся направлений в современных реалиях. Однако важно помнить, что цифровизация значительно повышает объем информации, получаемый слушателями, поэтому особую актуальность приобретает не просто предоставление информации в цифровой форме, а ее грамотная передача от преподавателя к ученику.

### Цифровые технологии в образовании

Цифровые технологии (ЦТ) в образовании – это способ организации учебного процесса, основанный на использовании электронных систем, повышающий качество и эффективность учебного процесса. На сегодняшний день в сфере образования наибольшим спросом пользуются нейротехнологии и искусственный интеллект (ИИ), технологии беспроводной связи, технологии виртуальной и дополненной реальности. «Перспективными областями применения являются индивидуальные образовательные траектории, иммерсивный образовательный опыт, дистанционное обучение, индивидуальные образовательные помощники, интерактивные образовательные комплексы и т. д.» [1].

Внедрение ЦТ (в частности, сквозных) в сферу образования уже идет «полным ходом». В частности, в России активно идет создание и воплощение в жизнь различных образовательных программ, использующих данные технологии. С 2018 года запущен целый ряд крупных образовательных проектов, например, «Образование-2024», «Цифровая школа», «Современная цифровая образовательная среда», «Цифровая экономика Российской Федерации», и многие другие.

В нашем вузе, например, реализуется проект «Цифровые кафедры» (ЦК) [2], который является частью федерального проекта «Развитие кадрового потенциала ИТ-отрасли» национальной программы «Цифровая экономика Российской Федерации». В рамках данного проекта студенты могут изучить дополнительные ИТ-компетенции, необходимые для выполнения нового вида профессиональной деятельности. В ГУАП для реализации в 2022/23 учебном году были разработаны программы профессиональной переподготовки (ПП): «Введение в разработку корпоративных приложений на Java», «Инженер по тестированию», «Основы Frontend-разработки», «Основы тестирования ПО», «Программирование на языке Python», «Проектирование на FPGA» и «Основы аналитики данных по отраслям».

### Чат-бот для взаимодействия со слушателями

В рамках программы ПП «Введение в разработку корпоративных приложений на Java» [3] для облегчения взаимодействия со слушателями, а также для их помощи им в поиске необходимой для обучения информации был разработан чат-бот «Помощник по ЦК» для мессенджера телеграмм (Telegram). Чат-бот был написан на языке программирования высокого уровня Python [4] с использованием асинхронной библиотеки (фреймворка) Aiogram [5]. В качестве базы данных (БД) была выбрана SQLite [6]. В БД хранятся списки групп слушателей программы ПП и данные о самих слушателях.

Разработанный чат-бот обладает следующими функциональными возможностями: регистрация пользователя, вывод расписания занятий группы слушателей программы ПП «Введение в разработку корпоративных приложений на Java», вывод варианта и задания по выбранной лабораторной работе, а также связь с куратором программы ПП. На рис. 1 представлен пример взаимодействия с чат-ботом (пример вывода расписания группы слушателя, получения задания к лабораторной работе и получения информации о кураторе программы ПП).

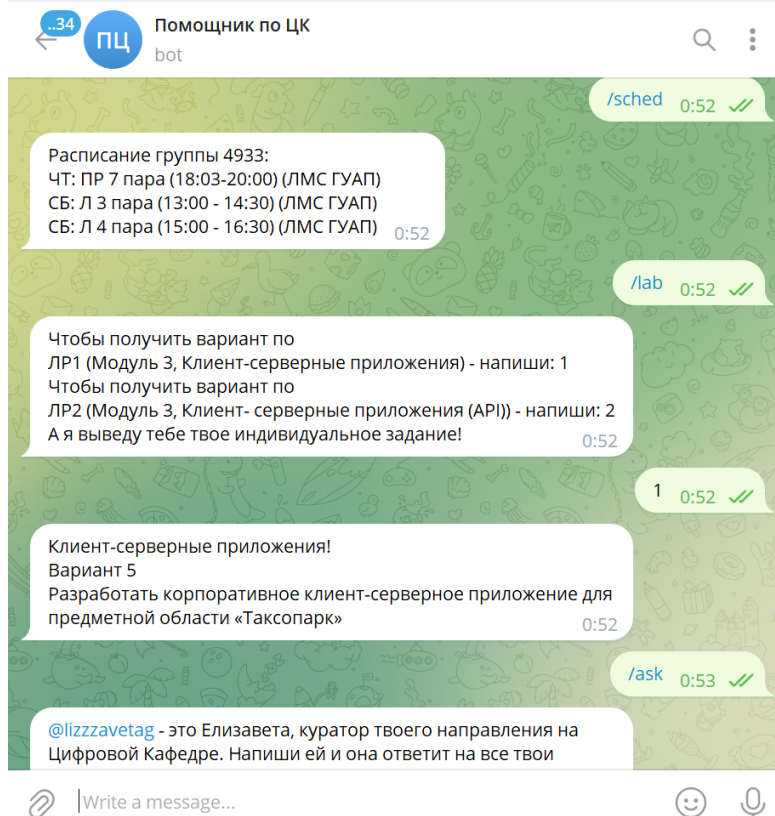


Рис. 1. Пример взаимодействия с чат-ботом «Помощник по ЦК»

Чат-бот «Помощник по ЦК» позволил оптимизировать процесс обучения по программе ПП и подтвердил необходимость применения ЦТ в сферу образования. С его помощью слушатели могут получать всю необходимую информацию в одном, привычном для современного поколения, месте, а также при необходимости быстро связаться с куратором программы ПП через мессенджер.

### Применение чат-бота для цифровизации образовательного процесса

Обучение по программе ПП «Введение в разработку корпоративных приложений на Java» проходит более 250 слушателей. При такой численности слушателей могут возникать сложности в коммуникации с преподавателями, и как следствие это приводит к повышению нагрузки на всех участников образовательного процесса. Разработанный чат-бот помогает снизить нагрузку со слушателей и пре-



подавателей и самое главное – позволяет повысить качество образования по программе ПП. В ходе обучения проводились тестирования, называемые в рамках проекта «Цифровые кафедры» ассесментами. Слушатели проходили ассесмент трижды – в начале, в середине и в конце обучения на платформе университета «Иннополис» для независимого контроля уровня сформированности компетенций.

Для оценки качества подготовки были взяты результаты студентов одного из направлений подготовки (88 слушателей), обучающихся на программе ПП «Введение в разработку корпоративных приложений на Java». На рис. 2 представлены круговые диаграммы, отражающие процентное соотношение результатов входного, промежуточного и итогового ассесмента среди выбранных слушателей. Из рисунка видна положительная динамика уровня, получаемых слушателями знаний. Процент слушателей, имеющих базовый уровень знаний, уменьшился с 23% до 6%, а процент слушателей с экспертным уровнем, вырос с 7% до 13%.

Из полученных результатов можно сделать вывод о целесообразности применения в процессе обучения разработанного чат-бота, что в свою очередь позволяет снизить нагрузку на преподавателей.

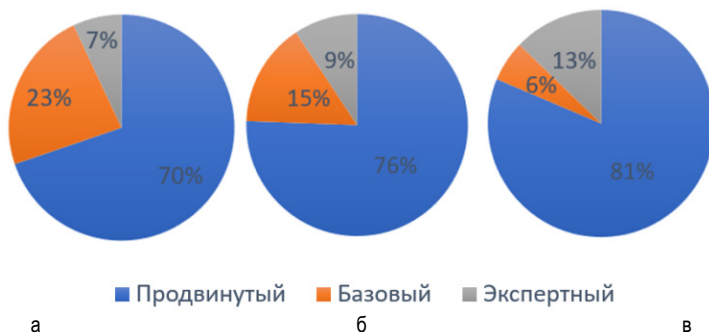


Рис. 2. Обобщенная информация о прохождении слушателями этапов ассесмента: а – входной, б – промежуточный, в – итоговый

### Заключение

Применение ЦТ в сфере образования является актуальной задачей, особенно учитывая происходящую в настоящее время эволюцию цифровой экономики [7 – 8]. В статье рассмотрено использование чат-ботов для решения указанной задачи и продемонстрирован положительный эффект от их внедрения.

### Библиографический список

1. Цифровые технологии в российской экономике / К. О. Вишневский, Л. М. Гохберг, В. В. Дементьев [и др.]; под ред. Л. М. Гохберга. М.: НИУ ВШЭ, 2021. 116 с. DOI: 10.17323/978-5-7598-2199-1.
2. Цифровые кафедры. URL: <https://guar.ru/m/digital> (дата обращения: 17.04.2023).
3. Ушаков В. А. Разработка корпоративных приложений на Java // Обработка, передача и защита информации в компьютерных системах. Третья Международная научная конференция: сборник докладов. СПб. 10–17 апреля 2023. ГУАП. С. 130–133. DOI: 10.31799/978-5-8088-1824-8-2023-3-130-133.
4. Python. URL: <https://www.python.org/> (дата обращения: 17.04.2023).
5. Aiogram. URL: <https://aiogram.dev/> (дата обращения: 17.04.2023).
6. About SQLite. URL: <https://www.sqlite.org/about.html> (дата обращения: 17.04.2023).
7. Цифровая экономика: 2022: краткий статистический сборник / Г. И. Абдрахманова, С. А. Васильковский, К. О. Вишневский [и др.]. М.: НИУ ВШЭ, 2022. 124 с. DOI: 10.17323/978-5-7598-2599-9.
8. Лапидус Л. В. Эволюция цифровой экономики // Ломоносовские чтения-2018. Секция экономических наук. Цифровая экономика: человек, технологии, институты: сборник тезисов выступления. М.: Экономический факультет МГУ имени М. В. Ломоносова, 2018. С. 153–158.

УДК 004.056.53

**К. А. Закиева**

студент кафедры информационных систем и технологий

**Н. В. Богословская** – доцент, кандидат технических наук – научный руководитель

### ЗАЩИТА ВЕБ-ПРИЛОЖЕНИЙ ОТ ХАКЕРСКИХ АТАК И УГРОЗ СЕТЕВОЙ БЕЗОПАСНОСТИ

Защита WEB-приложений является актуальной задачей большинства современных компаний, вне зависимости от их сферы деятельности. Связано это с тем, что для увеличения продаж, привлечения новых клиентов, повышения продуктивности и быстрого развития все больше организаций используют WEB-технологии. Они стали неотъемлемой частью жизнедеятельности многих видов бизнеса.

Различные интернет-сервисы предоставляют своим пользователям множество преимуществ, но вместе с тем они также несут и скрытые угрозы. Основная опасность связана с кибератаками. Опыт показывает, что с каждым годом число киберпреступлений только увеличивается [1].

Согласно исследованию Forrester, 39% всех атак разрабатывались для использования уязвимостей в веб-приложениях. Данный вид кибератак остается самым частым типом атак. На рис. 1 представлены основные способы осуществления внешних атак по отчету Forrester за 2021 год.



Рис. 1. Способы осуществления внешних атак

В таких условиях возникает вопрос: что предпринять для защиты веб-приложений? Защитные меры можно внедрять на двух этапах жизни приложения – разработки и эксплуатации [2]. На этапе разработки используются различные инструменты тестирования безопасности: статический, динамический и интерактивный анализ. Если говорить о безопасности уже эксплуатируемого приложения, то здесь можно применить наложенные средства защиты – это системы предотвращения вторжений, межсетевые экраны следующего поколения (NGFW – Next Generation Firewall), а также средства фильтрации трафика прикладного уровня, специально ориентированные на веб-приложения (WAF – Web Application Firewall). Применение Web Application Firewall традиционно считается наиболее эффективным подходом к защите веб-ресурсов.

Web Application Firewall – защитный экран уровня приложений, предназначенный для выявления и блокирования современных атак на веб-приложения, в том числе и с использованием уязвимостей нулевого дня:

- SQL Injection – sql инъекции;
- Remote Code Execution (RCE) – удаленное выполнение кода;
- Cross Site Scripting (XSS) – межсайтовый скриптинг;
- Cross Site Request Forgery (CSRF) – межсайтовая подделка запросов;
- Remote File Inclusion (RFI) – удаленный инклюд;
- Local File Inclusion (LFI) – локальный инклюд;
- Auth Bypass – обход авторизации;
- Insecure Direct Object Reference – небезопасные прямые ссылки на объекты;
- Bruteforce – подбор паролей [3].

#### **SQL-инъекции**

При атаке злоумышленник изменяет SQL-операторы для манипуляции пользовательскими данными. Злоумышленник использует код приложения для доступа или повреждения содержимого базы данных. В случае успеха злоумышленник сможет:

- читать конфиденциальные данные из баз данных (имена, пароли пользователей);
- создавать, изменять или удалять данные, хранящиеся в серверной базе данных;
- внедрять вредоносный контент в уязвимые поля.

#### **Межсайтовый скриптинг (XSS)**

Уязвимости XSS нацелены на скрипты, выполняющиеся у клиента, а не на сервере. Злоумышленник выполняет вредоносный сценарий в веб-браузере жертвы. Фактическая атака происходит, когда жертва посещает ресурс, выполняющий вредоносный код. Как правило, используется на форумах, сайтах объявлений и веб-страницах, на которых можно оставлять комментарии.

#### **Отказ в обслуживании (DoS)**

При DoS-атаках компьютеры в целевой сети поражают поддельные пакеты информации. Злоумышленники используют неправильно настроенные сетевые устройства. Цель – помешать веб-сайту обслуживать пользователей в обычном режиме.

#### **Подделка межсайтовых запросов (CSRF)**

При CSRF-атаке пользователя обманом заставляют выполнить действие. Сторонний веб-сайт отправляет запрос веб-приложению, в котором пользователь аутентифицировался (например, в их банке). После злоумышленник получает доступ к функциям через браузер жертвы. Целями становятся социальные сети, почтовые клиенты, онлайн-банкинг и веб-интерфейсы для сетевых устройств.

#### **Другие виды угроз**

Злоумышленники также создают вредоносные пакеты и маскируют их под легитимные программы. Такие пакеты разрабатываются с целью скрыться от обнаружения некоторых систем, избежать проверки и загрузиться на устройства жертв.

Нарушение в аутентификации и управлении сеансом связаны с проблемами сохранения личности пользователя. Если учетные данные для аутентификации и идентификаторы сеанса не защищены, злоумышленник может захватить активный сеанс и присвоить себе личность пользователя [2].

Также существуют угрозы сетевой безопасности, которыми может подвергаться пользователь.

**Сетевая безопасность** – список требований, рекомендаций и политик, которые используются в сетевой инфраструктуре для повышения ее уровня защиты и отказоустойчивости.

Выделяют 4 основных принципа проектирования сетевой безопасности на объекте информатизации:

- Оборудование должно быть отказоустойчивым и предусматривать возможность быстрого восстановления. Подразумевается наличие дублирующих компонентов в критически важных узлах.
- Систематический мониторинг всей инфраструктуры компании для обнаружения уязвимых точек.
- Постоянный мониторинг пропускной способности сетевого канала. Это гарантирует своевременную блокировку нежелательного трафика.

– Критически важные узлы инфраструктуры организации должны обеспечивать высокую доступность при любой угрозе либо атаке на компанию. Это достигается за счет создания второй независимой площадки, которая реплицирует данные с первой в синхронном режиме.

### **Средства обеспечения сетевой безопасности**

Различают 2 вида сетевых атак по статусу действия: активные и пассивные. Они также могут быть внутренними или внешними [4].

Активная атака проводится с непосредственным воздействием на сеть, которые может предусматривать ограничение ее работоспособности, модификацию настроек. Воздействие такого рода обязательно оставляет следы, поэтому при его планировании изначально предусматривается обнаружение.

Пассивная атака проводится без непосредственного влияния на работу сети. Однако в ее результате нарушается сетевая безопасность. Обнаружить пассивную атаку намного сложнее именно из-за отсутствия прямого воздействия. Примером таких угроз можно назвать постановку наблюдения или прослушки.

В качестве примеров наиболее распространенных сетевых атак можно привести следующие виды воздействия:

– Применение нестандартных протоколов. Тип протокола пакета данных определяется по содержащемуся в нем специальному полю. При изменении злоумышленниками значения в этом поле осуществляется передача данных, которую система не может определить.

– Ping Flooding. Такая атака может быть реализована только при условии доступа к высокоскоростному интернету. В результате создается избыточная нагрузка на сеть, что приводит к нарушениям в ее работе.

– Фрагментация данных. При передаче по IP пакет данных делится на части, а на стороне получателя – собирается. В случае атаки выполняется отправка значительного числа подобных фрагментов с засорением буфера обмена и нарушениям работы сети [5].

При любом раскладе рекомендуется использовать следующие меры защиты:

- прокси-серверы;
- межсетевые экраны;
- системы сетевого мониторинга;
- VPN.

### **Прокси-сервер**

Прокси-сервер – это компьютер-посредник между пользователем и ресурсом в Интернете. Если нужно открыть веб-страницу, прокси может сам сделать соответствующий запрос к нужному сайту, взять оттуда данные, проверить и отправить в браузер [6].

### **Межсетевые экраны**

Это программное обеспечение или аппаратно-программные продукты, предназначенные для блокировки нежелательного трафика.

Разрешение или запрет доступа межсетевым экраном осуществляется на основе заданных администратором параметров. В том числе могут использоваться следующие параметры и их комбинации [7].

### **Системы сетевого мониторинга**

Системы мониторинга сети – это программное обеспечение, позволяющее отслеживать состояние сетевых устройств, их работоспособность, исправность и другие характеристики. Системы мониторинга сети можно разделить на те, что отслеживают производительность сети и сигнализируют при перегрузке каналов, а также на те, что производят мониторинг сети с целью поиска сбоев и других проблем, связанных с работоспособностью серверного оборудования и других систем [8].

## VPN

VPN – это безопасное зашифрованное подключение пользователя к сети, с которым он может обходить локальные ограничения и сохранять конфиденциальность [9].

В современном мире существует большое количество атак. Несмотря на это разработали, а также будут разрабатываться защитные меры по их устранению и блокировки для обеспечения защищенности сайта.

### Библиографический список

1. Защита web-приложений. URL: <https://kazteleport.kz/news/zachemnuzhna-usluga-zashchity-web-prilozheniy/> (дата обращения: 24.04.23).
2. Атаки на web-приложения. URL: <https://cloudnetworks.ru/application-protection/> (дата обращения: 24.04.23).
3. Web Application Firewall. URL: <https://habr.com/ru/post/60590/> (дата обращения: 24.04.23).
4. Сетевая безопасность. URL: <https://itglobal.com/ru-ru/company/glossary/setevaya-bezopasnost-bezopasnost-seti/> (дата обращения: 25.04.23.)
5. Виды атак. URL: [https://www.smart-soft.ru/blog/typy\\_setevyh\\_atak\\_i\\_sposoby\\_borjby\\_s\\_nimi/](https://www.smart-soft.ru/blog/typy_setevyh_atak_i_sposoby_borjby_s_nimi/) (дата обращения: 25.04.23).
6. Прокси-сервер. URL: <https://www.unisender.com/ru/blog/sovety/chto-takoe-proksi-i-zachemnuzhen-kak-polzovatsya/> (дата обращения: 25.04.23).
7. Межсетевые экраны. URL: <https://www.smart-soft.ru/blog/mezhsetevye-ekrany-vidy/> (дата обращения: 25.04.23).
8. Системы сетевого мониторинга. URL: <https://www.anti-malware.ru/security/network-monitoring> (дата обращения: 25.04.23).
9. VPN. URL: <https://trends.rbc.ru/trends/industry/604f0a309a79477d332569e3> (дата обращения: 25.04.23).

УДК 004.51

**И. С. Зацепин**

студент кафедры информационных систем и технологий

**А. В. Бржезовский** – доцент, кандидат технических наук – научный руководитель


## ТЕХНОЛОГИИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ПРИ ПРОЕКТИРОВАНИИ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ

С каждым годом растет популярность использования искусственного интеллекта для решения задач во всех сферах человеческой деятельности. Искусственный интеллект находит место в разных отраслях. ИИ окружает нас уже почти во всех аспектах человеческой жизни: от спам-фильтров, музыкальных подборок в онлайн-плеерах до автопилотов в самолетах.

Проектирование пользовательских интерфейсов, включающее в себя UX/UX-дизайн, не стало исключением. Инструменты, содержащие в себе методы искусственного интеллекта, а также созданные с их помощью продукты породили целую концепцию в сфере дизайна [1].

ИИ уже способен повысить продуктивность работы дизайнера. Пользовательский опыт и интерфейсы, основанные на искусственном интеллекте, – будущее, которое стучится в двери уже сегодня. Многие дизайнеры экспериментируют, обращаясь к чат-ботам с искусственным интеллектом, таким как Sber GigaChat, OpenAI ChatGPT, Microsoft Bing Chat, Google Bard – за советами для принятия решений. Приведенные чат-боты могут пригодиться в таких задачах как: подбор цветовой палитры; заполнение полей условными текстовыми данными; поиск и вставка изображений, в том числе созданных методами ИИ; помощь при составлении портрета пользователя и проведении UX-исследования [2].

В качестве примера чат-боту ChatGPT был отправлен запрос на составление портрета пользователя приложения для студентов. Полученный текстовый ответ был разделен на несколько частей, отредактирован и приведен в удобную форму представления портрета пользователя, который приведен на рис. 1.

 <p><b>Пользователь студенческого приложения</b></p> <p>Возраст: 18-25 лет Пол: мужской/женский</p>	<p><b>Общее описание</b></p> <p>Пользователь студенческого приложения – молодой человек или девушка в возрасте от 18 до 25 лет. Он или она являются студентами вузов или колледжей, активно учатся и заинтересованы в получении новых знаний и навыков. Они часто используют различные гаджеты для учебы и коммуникации, такие как смартфоны и ноутбуки. Также у них высокая социальная активность: они участвуют в различных студенческих мероприятиях, клубах и обществах. Студенты, использующие приложения для обучения, часто ориентированы на успех и готовы много работать, чтобы достичь своих целей. Они постоянно ищут новые возможности для роста и самосовершенствования, и приложения для учебы помогают им в этом.</p>	<p><b>Мотивы</b></p> <ul style="list-style-type: none"> <li>Желание оптимизировать учебный процесс</li> <li>Иметь быстрый доступ к требуемой информации</li> <li>Иметь возможность связаться с преподавателем/сокурсником</li> </ul> <p><b>Потребности и ожидания</b></p> <p>Пользователь заинтересован в быстром доступе к информации и удобной навигации внутри приложения. Он оценивает возможность заработка дополнительных баллов, получения сертификатов и удобства взаимодействия с другими пользователями.</p>
--	--	--

*Рис. 1. Портрет пользователя студенческого приложения, предложенный искусственным интеллектом*

Важной частью дизайна интерфейсов является персонализация, – изучение и последующее использование информации о поведении пользователя с целью предоставления ему контента и функциональности, соответствующих его индивидуальным предпочтениям [3]. Она способствует привлечению и удержанию пользователей продукта.

Персонализация требует тщательной сегментации пользователей и получения от них обратной связи по функциям продукта, которые предполагаются полезными для этой группы. В этом случае искусственный интеллект можно использовать для ускорения анализа пользовательских данных. В результате чего разработчики смогут быстрее выделять полезные знания, тем самым будет расти уровень понимания потребностей пользователей. В итоге, основываясь на полученных выводах, удастся создать высоко персонализированный пользовательский интерфейс.

Персонализация, в отличие от кастомизации, не требует участия пользователя и контролируется системой. Кастомизация – явный инструмент выявления предпочтений пользователей. В данном случае пользователям предоставляется возможность адаптировать опыт под свои потребности и интересы.

Алгоритмы искусственного интеллекта могут получать доступ к огромным объемам данных о пользователях в режиме реального времени. Это поможет повысить качество поддержки приложений. Также система может точно предвидеть опыт взаимодействия пользователя и соответствующим образом кастомизировать его.

Методики изучения и анализа поведения пользователей направлены на создание удобного пользовательского интерфейса путем устранения препятствий, которые мешают получить положительный опыт пользования. Искусственный интеллект можно использовать для повышения продуктивности при выполнении рутинных задач, путем автоматизации.

Для примера можно привести редактирование изображений с помощью графических инструментов, и такие распространенные задачи, как: ретушь, цветокоррекция, изменение формата и объединения изображений. Так, американская компания Adobe смогла автоматизировать эти функции в своем популярном графическом редакторе Adobe Photoshop. Пользователи смогли повысить свою продуктивность при работе с этим инструментом.

Концепция сбора и интерпретации информации о поведении пользователей не является новой. Дизайнеры, менеджеры по продукту и маркетологи всегда собирают самые разнообразные данные. Главная задача состоит в расшифровке и анализе полученной информации.

Обучение искусственного интеллекта распознаванию образов может в разы ускорить сбор, анализ и интерпретацию пользовательских данных. Многие популярные приложения, ориентированные на решение этих задач, такие как Google Analytics, уже содержат в себе функционал, использующий машинное обучение [4]. В пример можно Automated Insights, созданные искусственным интеллектом, в инструменте аналитики посещений сайта Search Console Insights. На рис. 2 можно увидеть окно Insights в Google Analytics.

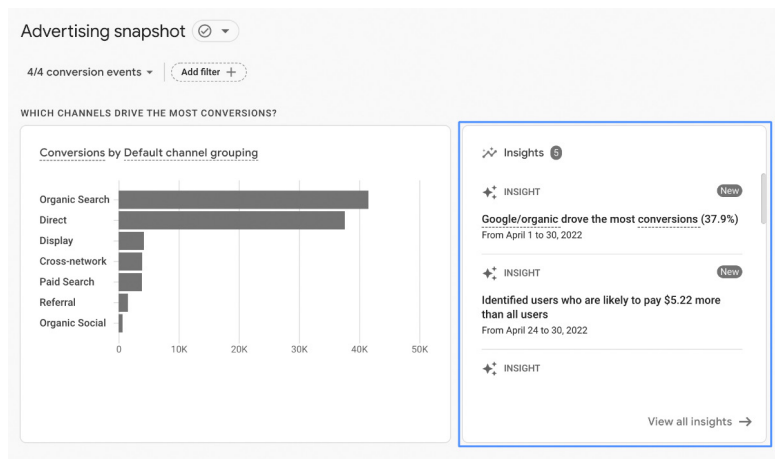


Рис. 2. Окно Automated Insights в Google Analytics

Другими словами, искусственный интеллект способен оптимизировать анализ огромных объемов данных для поиска закономерностей. В UX-исследованиях это помогает получить ценную информацию в более короткие сроки.

Существуют проекты, где искусственный интеллект может являться не просто удобным подспорьем при разработке, а формирует целое ценностное предложение. Укрепление доверия пользователей – ключевой момент при работе над проектами такого типа. Единственное решение – это объяснение пользователям того, как функционирует искусственный интеллект, а также демонстрация результатов его использования.

Следующей проблемой является этика искусственного интеллекта [5]. Разработчики несут этическую ответственность за свой продукт. Не все что может быть создано с помощью искусственного интеллекта – должно быть реализовано. На работу искусственного интеллекта стоит накладывать ограничения, во избежание ситуаций, когда результаты, полученные искусственным интеллектом, переступают за этические нормы. Так, например, чат-бот с искусственным интеллектом ChatGPT от OpenAI блокирует выдачу данных, в которых может содержаться информация о создании оружия или способная вызвать межнациональную и религиозную ненависть.

Использование искусственного интеллекта открывает новые возможности при проектировании пользовательского интерфейса продукта. Крупные иностранные и отечественные компании заинтересованы в разработке и использовании методов искусственного интеллекта. Рациональное использование возможностей искусственного интеллекта при проектировании программных продуктов положительно влияет на весь процесс разработки.

### Библиографический список

1. How Artificial Intelligence is shaping UI/UX design. URL: <https://fireart.studio/blog/ai-driven-design-how-artificial-intelligence-is-shaping-ui-ux-design> (дата обращения: 13.04.2023).
2. Методы UX-исследования на этапах проектирования продукта. URL: <https://vc.ru/ui/943786-andrey-smagin/310965-metody-ux-issledovaniya-na-etapah-proektirovaniya-produkta> (дата обращения: 14.04.2023).
3. Персонализация в UX URL: <https://www.uprock.ru/articles/personalizaciya-v-ux> (дата обращения: 16.04.2023).
4. Google Аналитика: информация для разработчиков. URL: <https://developers.google.com/analytics?hl=ru> (дата обращения: 16.04.2023).
5. Кодекс этики в сфере ИИ. URL: <https://ethics.a-ai.ru/> (дата обращения 17.04.2023).



УДК 004.432.2

**Д. Д. Знаменская**

студент кафедры информационных систем и технологий

**Н. В. Богословская** – кандидат технических наук, доцент – научный руководитель

## СРАВНЕНИЕ СРЕДСТВ FLUTTER И XAMARIN ДЛЯ СОЗДАНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА В РАЗРАБОТКЕ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

### Введение

В настоящее время разработка мобильных приложений — одна из самых быстроразвивающихся сфер информационных и коммуникационных технологий, как в бизнесе, так и в государственном секторе. Согласно глобальной статистике компании Statcounter [1], представленной на рис. 1, за последние десять лет доля мобильных устройств по сравнению с персональными компьютерами увеличилась с 13,56% до 58,52%. Вследствие массового распространения мобильных устройств значительно возросла и потребность в создании широкого спектра приложений для данной категории платформ.

Desktop vs Mobile vs Tablet Market Share Worldwide

Jan 2013 - Jan 2023

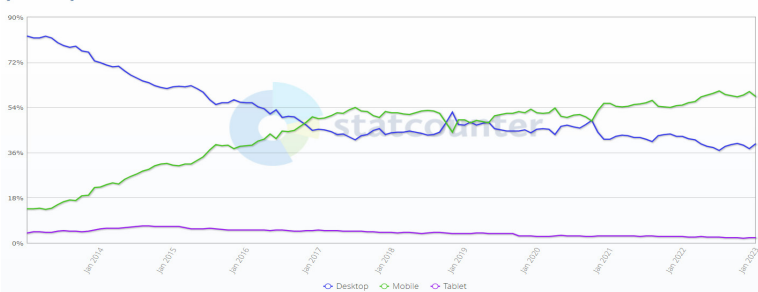


Рис. 1. Изменение доли персональных компьютеров, мобильных [1] и планшетных устройств на рынке за последние 10 лет [1]

С учетом стремительного роста количества пользователей мобильных устройств и высоких темпов цифровизации таких сфер жизни, как обслуживание, образование, культура, производство, медицина, при разработке приложения стало необходимым учитывать множество особенностей перечисленных областей. Кроме того, приложение должно функционировать на разных мобильных платформах; графический интерфейс должен адаптироваться под любой экран, при этом соответствовать принципам юзабилити и выглядеть привлекательно; приложение должно демонстрировать хорошую производительность, чтобы обеспечить комфортное взаимодействие пользователя с ним. При выборе платформы для разработки мобильного приложения учитываются и перечисленные, и многие другие факторы.

В статье выполнено сравнение возможностей Flutter и Xamarin для создания пользовательского интерфейса мобильных приложений для устройств на базе операционных систем Android и iOS.

### Описание технологий

Flutter – фреймворк с открытым исходным кодом от компании Google для создания мобильных приложений под Android и iOS, веб-приложений, а также настольных приложений под Windows, macOS и Linux. Flutter использует объектно-ориентированный язык программирования высокого уровня Dart и с его помощью реализует разнообразный набор виджетов, анимаций и жестов управления, а также навигацию, обработку HTTP-запросов и прочее. Важная особенность языка Dart – поддержка двух типов

компиляции, JIT и AOT. JIT-компиляция (англ. just-in-time, динамическая) [2] – компиляция, осуществляющаяся во время работы программы, когда среда выполнения анализирует исполняемый код, оптимизируя часто вызываемые участки (что, тем не менее, замедляет работу программы). Благодаря JIT-компиляции Flutter-разработчик может использовать “горячую перезагрузку” – функцию, позволяющую вносить изменения в программный код без необходимости останавливать работу приложения. AOT-компиляция (англ. ahead-of-time, статическая) [2] – процесс полной компиляции кода приложения в машинный код при создании исполняемого файла; для того чтобы увидеть результат внесения изменений, необходимо целиком перекомпилировать программу, однако такая программа будет работать предсказуемо и без пауз для анализа. За отрисовку пользовательского интерфейса Flutter-приложения отвечает графический движок Skia.

Xamarin – фреймворк с открытым исходным кодом от компании Microsoft для кроссплатформенной разработки мобильных приложений (iOS, Android, Windows Phone). В основе Xamarin лежит платформа .NET с объектно-ориентированным языком программирования высокого уровня C#. Платформа Xamarin предоставляет собой уровень абстракции, который обеспечивает управление взаимодействием между общим кодом и кодом базовой платформы [3]. Xamarin выполняется в управляемой среде, которая реализует такие возможности, как выделение памяти и сборка мусора. Платформа Xamarin полагается на ряд субплатформ, а именно: Xamarin.Android (библиотеки для создания Android-приложений) и Xamarin.iOS (библиотеки для создания iOS-приложений). Приложения Xamarin.Android компилируются из языка C# в промежуточный язык, который при запуске приложения претерпевает JIT-компиляцию в машинную сборку. Приложения Xamarin.iOS проходят полную AOT-компиляцию из языка C# в собственный код сборки [3].

#### **Критерии сравнения фреймворков в контексте разработки пользовательского интерфейса для мобильных приложений**

Ознакомившись с различными источниками [4, 5], описывающими характеристики средств фреймворков для создания пользовательского интерфейса, автор статьи предлагает следующие критерии для сравнения таких фреймворков:

1. Единая кодовая база – следование принципу “Написал раз – запустил везде” (англ. “Write once – run everywhere”). Применение одного и того же кода для разных платформ, в данном случае: Android и iOS.
2. Возможность реализации адаптивных макетов – фреймворк позволяет разработать структурно сложный пользовательский интерфейс и обеспечить его адаптацию под любое мобильное устройство.
3. Гибкость верстки интерфейса – предоставление разработчику удобного и понятного инструментария для компоновки и настройки пользовательского интерфейса.
4. Возможность создания единого дизайна для любой платформы – доступность реализации не платформоспецифичного дизайна.
5. Кастомизация компонентов интерфейса – наличие у фреймворка широкого набора средств для серьезного изменения внешнего вида стандартных компонентов пользовательского интерфейса или создания совершенно новых.
6. Наличие эффективных инструментов отладки – фреймворк ускоряет процесс разработки за счет возможности просматривать и тестировать вносимые в код изменения без необходимости останавливать отладку.
7. Возможность создания сложных анимаций – фреймворк может реализовать сложные графические анимации без потерь производительности.

#### **Разработка графического пользовательского интерфейса с помощью фреймворка Flutter**

Flutter – фреймворк с единой кодовой базой [6]: любое приложение, разрабатываемое на Flutter, будет в полной мере поддерживаться и на Android-, и на iOS-устройствах (критерий 1).

Пользовательский интерфейс Flutter-приложения строится из виджетов. Виджет – визуальный элемент. Все, что относится к графическому слою приложения – это виджет. Виджеты описывают, как должен выглядеть их вид с учетом текущей конфигурации и положения (критерий 3). Когда состояние

виджета меняется, виджет перестраивает его описание, которое фреймворк отличает от предыдущего описания для того, чтобы определить минимальные изменения, необходимые в дереве рендеринга для перехода из одного состояния в другое. Каждый виджет может являться дочерним классом либо класса `StatelessWidget` (не имеющие состояния), либо класса `StatefulWidget` (имеющие состояние) [7]. В процессе работы приложения виджеты, которые не имеют состояния, могут изменяться лишь посредством внешних событий, которые возникают на родительских виджетах. Виджеты, имеющие состояние, в процессе работы приложения могут изменять свои свойства динамически, а состояние виджета хранится в виде объекта класса `State`. `StatefulWidget` позволяют создавать анимации любой сложности (критерий 7), которые могут постоянно воспроизводиться со скоростью 60 кадров в секунду [7]. В составе каждого виджета находятся узкоспециализированные блоки, которые в совокупности могут произвести сложную композицию. Разработчик может кастомизировать абсолютно все виджеты и контролировать буквально каждый пиксель [6], реализуя любое дизайн решение (критерий 5).

Для помощи в создании приложений с адаптивным стилем визуальных элементов команда Flutter разработала две библиотеки виджетов: `Material` дизайн для `Android`-устройств и `Cupertino` дизайн для `iOS`-устройств. Оба пакета имитируют дизайн нативных компонентов данных мобильных платформ. Однако не обязательно привязываться к внешнему виду элементов управления конкретной платформы (критерий 4): Flutter предоставляет широкие возможности быстро и просто разработать единый дизайн графического интерфейса для любого устройства.

Адаптация (критерий 2) интерфейса под разные экраны осуществляется благодаря свойствам класса `MediaQuery`, который дает доступ к множеству свойств, описывающих текущий пользовательский контекст, например, свойству `size`, содержащему информацию о ширине и высоте экрана.

Вдобавок, как упоминалось выше, Flutter позволяет отлаживать приложения с помощью “горячей перезагрузки”. Вносимые в код (и верстка, и написание логики приложения осуществляются через код) изменения сразу будут отражены в работающем приложении без повторной компиляции (критерии 3 и 6).

### **Разработка графического пользовательского интерфейса с помощью фреймворка Xamarin**

В состав `Xamarin` входит фреймворк `Xamarin.Forms`, который позволяет создать единую верстку интерфейса сразу и под `Android`, и под `iOS` платформу. Верстка производится один раз на базе `Xamarin`-компонентов, а на целевой платформе для каждого из компонентов вызывается код-рендерер, отрисовывающий нативные компоненты. Реализация интерфейса платформоспецифична (критерий 4). Сама верстка доступна как с помощью кода на языке `C#`, так и с помощью языка разметки `XAML` [3]. Внешний вид библиотечных или пользовательских компонентов настраивается благодаря `XAML` стилям (критерий 3) или каскадным таблицам стилей (`CSS`). `Xamarin.Forms` содержит ограниченный стандартный набор элементов управления. При необходимости создать новый, собственный элемент управления приходится разрабатывать рендереры [8] для каждой целевой платформы, т. е. отдельно – для `Android`, и отдельно – для `iOS`, что становится камнем преткновения в идее кроссплатформенности фреймворка и использовании единой кодовой базы (критерий 1). Также возникают трудности с кастомизацией графического интерфейса, если имеется желание использовать сложные анимации, градиенты, тени и т. д.: заметно понижается производительность приложения (критерии 5 и 7). Дизайн тяжело привести к единому стилю, `Xamarin` не содержит средств автоматизации разработки для этих целей.

Адаптивность пользовательского интерфейса (критерий 2) реализована за счет визуальных состояний. С их помощью можно существенно изменить макет на основе размера окна или внести другие правки. Разработчик имеет возможность определять различные визуальные состояния пользовательского интерфейса и применять их, когда ширина или высота окна превысит заданное пороговое значение.

Для ускорения разработки `Xamarin` предоставляет инструмент (критерий 6) “горячая перезагрузка”, позволяющий увидеть внесенные в `XAML`-файлы изменения без необходимости развертывать приложение заново. Однако данный инструмент имеет ограничения: он не может перезагрузить `C#` код, включая обработчики событий, пользовательские элементы управления, код страницы программной части и дополнительные классы [9].

### Сравнение фреймворков Flutter и Xamarin

В табл. 1 представлено краткое сравнение фреймворков Flutter и Xamarin по различным критериям, описанным в пункте 2. Использовались следующие оценки: “+” (полностью реализует), “+/-” (частично реализует) и “-” (не реализует или имеет проблемы с реализацией).

Таблица 1

#### Сравнение Flutter и Xamarin

Параметры сравнения / фреймворк	Flutter	Xamarin
Единая кодовая база	+	+/-
Возможность реализации адаптивных макетов	+	+
Гибкость верстки интерфейса	+	+
Возможность создания единого дизайна для любой платформы	+	-
Кастомизация компонентов интерфейса	+	+/-
Наличие эффективных инструментов отладки	+	+/-
Возможность создания сложных анимаций	+	-

Как видно из табл. 1, у фреймворка Flutter оказалось значительно больше преимуществ, чем у фреймворка Xamarin.

#### Выводы

Оба фреймворка предоставляют сопоставимые по эффективности возможности для разработки как простых, так и сложных кроссплатформенных мобильных приложений, однако, исходя из анализа средств Flutter и Xamarin для создания пользовательского интерфейса мобильного приложения, следует, что чаша весов перевешивает в пользу Flutter. Flutter хорошо справляется с тяжелой графикой благодаря мощному, постоянно совершенствуемому компанией Google движку, предоставляет широкий выбор инструментов для кастомизации визуальных компонентов и позволяет реализовать удобный интерфейс с ярким, красивым и привлекательным дизайном.

#### Библиографический список

1. Глобальная статистика по использованию устройств // Сервис веб-аналитики StatCounter. URL: <https://gs.statcounter.com/> (дата посещения: 29.03.2023).
2. Windmill, E. Flutter in Action // Shelter Island: Manning, 2020. 368 p.
3. Что такое Xamarin // Документация Microsoft. URL: <https://learn.microsoft.com/ru-ru/xamarin/get-started/what-is-xamarin> (дата посещения: 02.04.2023).
4. Как правильно выбрать frontend-framework // Medium. URL: <https://medium.com/fafnur/как-правильно-выбрать-frontend-framework-e995d2765682> (дата посещения: 02.04.2023).
5. Бастрикина В. В. Сравнительный анализ адаптивных css фреймворков // Актуальные проблемы авиации и космонавтики. 2016. Т. 1. С. 595–596.
6. Flutter-multiplatform // Сайт Flutter. URL: <https://flutter.dev/multi-platform> (дата посещения: 04.04.2023).
7. Flutter // Документация Flutter. URL: <https://docs.flutter.dev/> (дата посещения: 05.04.2023).
8. Пользовательские отрисовщики // Документация Microsoft. URL: <https://learn.microsoft.com/ru-ru/xamarin/xamarin-forms/app-fundamentals/custom-renderer/introduction> (дата посещения: 07.04.2023).
9. Горячая перезагрузка // Документация Microsoft. URL: <https://learn.microsoft.com/ru-ru/xamarin/xamarin-forms/xaml/hot-reload> (дата посещения: 10.04.2023).

УДК 004.021

**А. А. Князева**

студент кафедры информационных систем и технологий

**Н. В. Богословская** – доцент, кандидат технических наук – научный руководитель

### СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ КЛАСТЕРИЗАЦИИ КАТЕГОРИАЛЬНЫХ ДАННЫХ

На данный момент мир переживает «информационный стресс», который связан с непрерывно растущими потоками информации. Ежегодно объем генерируемых в мире данных увеличивается на 30,7%, о чем свидетельствуют исследования, проводимые крупнейшими международными исследовательскими компаниями IDC (International Data Corporation) [1] и Statista [2].

По данным немецкой компании Statista в 2022 году объемы всех данных, накопленных человечеством, составили примерно 97 зеттабайт, но уже к 2025 году это число возрастет примерно в два раза, до 180 зеттабайт.

При этом согласно исследованиям компании IDC, большая часть генерируемых данных временно создается или реплицируется для использования, так к 2021 году было сохранено менее 2% новых данных.

В связи с колоссальным ростом информации, возрастает спрос на методы и средства ее хранения и обработки. По данным IDC, в 2021 году спрос на технологии, позволяющие обрабатывать и анализировать большие данные вырос на 10,1% по сравнению с 2020 годом. В число лидеров, интерес к которым постоянно растет, входят такие технологии, как интеллектуальный анализ данных, Data Mining, а вместе с тем и методы кластерного анализа.

Кластеризация – это объединение схожих объектов данных или наблюдений в непересекающиеся группы, называемые кластерами, основанное на близости значений признаков этих объектов, с целью выявления скрытых структур [3].

На рис. 1 представлена схема кластерного анализа данных.

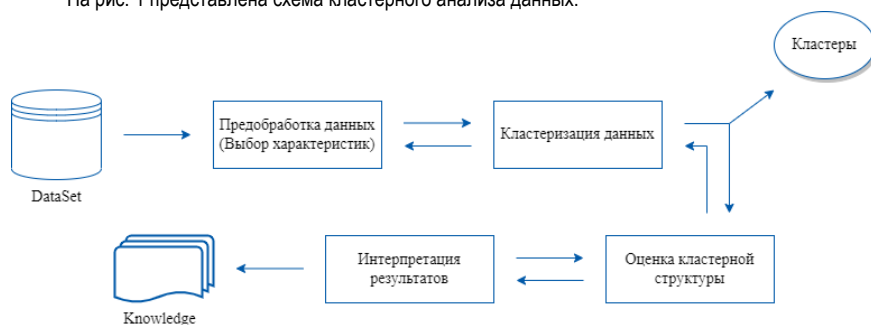


Рис. 1. Процесс кластерного анализа

Кластерный анализ применяется повсеместно, в различных сферах нашей жизни. Список прикладных областей велик: сегментация изображений, новостные потоки, маркетинг, службы поддержки клиентов, борьба с мошенничеством, прогнозирование, анализ текстов.

Самым известным на данный момент решением является метод k-средних или k-means. Данный алгоритм широко применим в сфере интеллектуального анализа данных, но имеет серьезное ограничение – возможность обрабатывать только числовые данные, что не позволяет использовать его для кластеризации данных реального мира.

В отличие от данных векторного пространства, практически во всех областях, в педагогических науках, социологии, исследованиях рынка, биологии и биоинформатике данные включают категориаль-

ные атрибуты, а поскольку категории не имеют естественного порядка, методы кластеризации, разработанные для данных метрического пространства, не могут быть применены.

Методы кластеризации категориальных данных сильно отличаются от методов кластеризации числовых данных с точки зрения определения меры сходства. Большинство числовых методов кластеризации используют функции расстояния, например, евклидово расстояние, для определения меры подобия, в то время как между категориальными значениями нет внутреннего значения расстояния.

На данный момент разработано немало алгоритмов кластеризации и их модификаций, которые позволяют обрабатывать категориальные данные. В данной статье производится сравнительный анализ следующих четырех алгоритмов: k-modes, k-medoids, ACE и ROCK.

Сравнение указанных алгоритмов производилось на 10 наборах разной размерности смешанных или категориальных данных, полученных из архива машинного обучения UCI [4] и интернет-платформы для изучения Data Science Kaggle [5]. Характеристики наборов представлены в табл. 1.

Таблица 1

**Характеристики наборов данных**

	Наименование набора данных	Количество векторов	Количество атрибутов	Количество категорий	Тип данных
1	Набор данных о звездах	240	7	910	Смешанный
2	Набор данных о пациентах	918	12	529	Смешанный
3	Набор данных о грибах	1 504	22	116	Номинальный
4	Набор данных о пассажирах авиакомпаний	1 500	23	1 313	Смешанный
5	Набор данных о переписи населения	1 979	15	2 260	Номинальный
6	Набор данных о грибах	8 124	22	117	Номинальный
7	Набор данных о клиентах банка	10 127	20	23 243	Смешанный
8	Набор данных о пассажирах авиакомпаний	20 168	23	3 851	Смешанный
9	Набор данных о переписи населения	32 535	15	22 052	Номинальный

Перед кластерным анализом данные были предварительно обработаны и оптимизированы, исключены дублирующие записи и незначимые характеристики, такие как идентификаторы и признаки с одной категорией.

Для того, чтобы оценить качество кластерных структур – результата кластеризации данных, используют различные метрики качества, которые можно разделить на три группы: внешние, внутренние и относительные.

Внешние индексы основаны на некоторой заранее заданной структуре, которая является отражением априорной информации о данных, и используются в качестве стандарта для проверки решений кластеризации.

Внутренние индексы не зависят от внешней информации, они исследуют структуру кластеров, опираясь на сведения, которые можно получить непосредственно из исходных данных – метки классов.

К сожалению, в большинстве случаев проведения кластерного анализа у специалистов нет сведений о структуре исследуемых данных. В таких случаях применяются относительные метрики, при использовании которых акцент делается на сравнении различных кластерных структур с целью определения, какая из них лучше всего раскрывает характеристики исследуемых объектов [3].

Разные алгоритмы кластеризации могут приводить к разным кластерным структурам, именно поэтому перед проведением сравнительного анализа были определены оптимальные количества кластеров для каждого тестового набора данных, полученные с помощью алгоритмов кластеризации.

Для оценки качества полученных кластерных структур были рассчитаны следующие относительные метрики: индекс Дэвида – Булдина, индекс оценки силуэта и индекс S\_Dbw, – а также время и объем оперативной памяти, которые были затрачены на обработку данных. На рис. 2 представлены диаграммы значений индекса силуэта, полученных для каждой кластерной структуры.

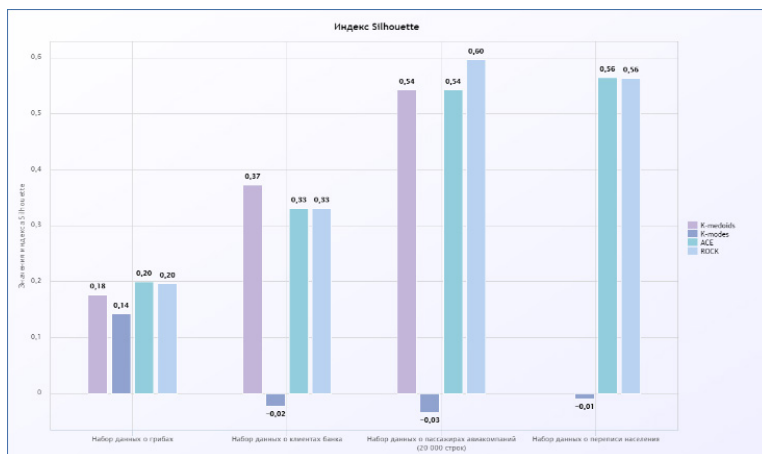


Рис. 2. Диаграммы значений индекса силуэта

Силуэт – это метрика, которая определяет меру непохожести отдельного элемента с элементами ближайшего кластера. Максимальное значение данного индекса свидетельствует о лучшем разбиении данных и построении более оптимальной кластерной структуры, а это значит, что расстояние между элементами внутри кластера достаточно мало, а расстояние между кластерами наоборот велико [6].

По полученным значениям индекса силуэта можно сделать вывод, что наилучшие результаты обработки категориальных наборов данных показывают алгоритмы иерархической кластеризации ACE и ROCK. Алгоритм *k-medoids* наравне с алгоритмами ROCK и ACE показывает хорошие результаты при кластеризации смешанных данных, в то время как алгоритм *k-modes* с обработкой числовых признаков не справляется и применим для кластеризации данных исключительно номинального типа, наличие нумеративных признаков в значительной степени ухудшает результаты работы алгоритма.

Полученные значения индексов Дэвиса – Булдина (рис. 3) и *S\_Dbw* (рис. 4) подтвердили полученные результаты.



Рис. 3. Диаграммы значения индекса Дэвиса – Булдуна





Рис. 4. Диаграммы значений индекса S\_Db

Помимо вычисления значений метрик для сравнения работы алгоритмов кластеризации также были произведены измерения времени и объема оперативной памяти, необходимых алгоритмам для построения кластерных структур. Полученные значения представлены в виде диаграмм на рис. 5 и 6.

Все рассматриваемые алгоритмы кластеризации на обработку относительно небольшого набора данных затрачивают примерно одинаковое количество оперативной памяти (1,4 Gb). Но с увеличением объемов данных количество гигабайтов ОЗУ, необходимое алгоритмам k-medoids, ROCK и ACE для обработки, значительно возрастает, в то время как алгоритму k-modes на кластеризацию 32 000 строк требуется всего 1,5 GB.



Рис. 5. Диаграммы значений времени

Алгоритмы алгомеративной кластеризации, а также алгоритм k-medoids обрабатывают данные в разы быстрее алгоритма k-modes. С увеличением объемов данных время, требуемое алгоритму на построение кластерной структуры, пропорционально возрастает.

Также были проведены эксперименты по оценке масштабируемости алгоритмов, которые проводились в интеллектуальной среде выполнения с выделенной оперативной памятью 12,7 GB.

В табл. 2 представлены полученные результаты. В некоторых ячейках указано значение n/a, которое свидетельствует о том, что алгоритму не хватило предоставленных ресурсов для обработки набора данных.



Рис. 6. Диаграммы значений объема оперативной памяти

Таблица 2

Оценка масштабируемости

Набор данных	Алгоритмы			
	K-medoids	ROCK	ACE	K-modes
Менее 1 000 записей	+	+	+	+
2 000 записей	+	+	+	+
5 000 записей	+	+	+	+
8 000 записей	+	+	+	+
10 000 записей	+	+	+	+

Набор данных	Алгоритмы			
	K-medoids	ROCK	ACE	K-modes
20 000 записей	+	+	+	+
25 000 записей	+	+	+	+
28 000 записей	+	+	+	+
29 000 записей	+	+	+	+
30 000 записей	+	+	+	+
31 000 записей	+	+	+	+
32 000 записей	n/a	+	+	+
35 000 записей	n/a	+	+	+
37 000 записей	n/a	+	+	+
38 000 записей	n/a	+	+	+
39 000 записей	n/a	n/a	n/a	+
40 000 записей	n/a	n/a	n/a	+
100 000 записей	n/a	n/a	n/a	+

По итогам эксперимента видно, что наиболее масштабируемым алгоритмом кластеризации является алгоритм k-modes, которому для обработки больших массивов данных требуется меньше объемов выделенной оперативной памяти, чем другим алгоритмам.

Несмотря на это, кластерные структуры, полученные в результате работы алгоритма k-modes, по качеству значительно уступают структурам, являющимися результатом работы других рассматриваемых алгоритмов, что не может компенсироваться более экономным расходованием ресурсов.

В результате проведения сравнительного анализа можно сделать вывод, что иерархические алгоритмы являются более универсальным решением кластерного анализа, так как позволяют обрабатывать данные разных типов с небольшими затратами времени и объема оперативной памяти и достижением наилучшего результата.

#### Библиографический список

1. International Dance Center (IDC). URL: <https://www.idc.com/> (дата обращения: 20.04.2023).
2. Statista. URL: <https://www.statista.com/statistics/1185888/worldwide-global-datasphere-unique-replicated-data/> (дата обращения: 20.04.2023).
3. Survey of clustering algorithms / R. Xu, D. Wunsch // IEEE Transactions on Neural Networks. 16 (3). 2005. С. 645–678.
4. UCI. URL: <https://archive-beta.ics.uci.edu/> (дата обращения: 25.04.2023).
5. Data Science Kaggle. URL: <https://www.kaggle.com/datasets> (дата обращения: 25.04.2023).
6. Finding Groups in Data. An Introduction to Cluster Analysis / L. Kaufman, P. J. Rousseeuw // Wiley, 2005. С. 342.

УДК 004.584

**И. М. Комаров**

студент кафедры информационных систем и технологий

**Н. В. Богословская** – доцент, кандидат технических наук – научный руководитель**ОСОБЕННОСТИ РАЗРАБОТКИ ПРОГРАММНОГО БОТА ДЛЯ ПОМОЩИ  
СОТРУДНИКАМ КОМПАНИИ, ИСПОЛЬЗУЮЩЕГО REST-API СОЦИАЛЬНЫХ СЕТЕЙ**

В наше время, в век цифровизации и огромных технических возможностей большую роль играет автоматизация рутинных процессов, ведь человеческий труд стоит намного дороже, чем даже работающий круглосуточно сервер. К тому же, ручной труд отнимает много времени и часто подвержен ошибкам.

Современному этапу развитой корпоративной информатизации хорошо соответствуют технологии роботизации бизнес-процессов (RPA) программные роботы, способные работать круглосуточно и без человеческих ошибок, связанных с невнимательностью, усталостью и т. д., берут на себя часто повторяющиеся рутинные рабочие операции.

**TADVISER**

Рис. 1. Автоматизация в различных отраслях [1]

Чаты-боты являются частным случаем автоматизации бизнес-процессов. На сегодняшний день это инструмент, необходимый для всевозможных мессенджеров, социальных сетей и сайтов. Чат-бот – это многовариантный алгоритм, способный воспринимать информацию в самой простой и доступной для человека форме – диалог [1].

Разрабатывая программного бота, можно столкнуться с некоторыми особенностями. Например, существует несколько вариантов для проверки наличия сообщений от пользователя: long polling и webhooks.

Long polling – первый и наиболее простой вариант, он заключается в периодическом опросе REST-API серверов социальных сетей на предмет наличия новой информации, т.е. открывается соединение с серверами на непродолжительное время и все обновления тут же прилетают боту [2].

Webhooks же работают несколько иначе, ибо здесь сервера социальной сети уже сами отправляют обновления на сервер с чат-ботом, что значительно снижает используемый трафик. При этом от-

падает необходимость периодически самому опрашивать серверы, однако за это приходится платить необходимостью установки полноценного веб-сервера на ту машину, на которой планируется запускать программного бота. На рис. 2 представлена схематично работа программного бота и взаимодействие его с конечным пользователем. Здесь сервера соц. сетей являются прослойкой между пользователем и сервером обработки действий пользователя.

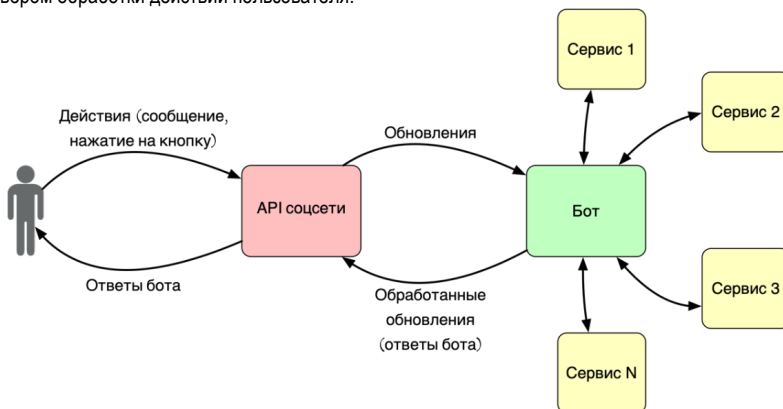


Рис. 2. Схематичное представление работы бота

В зависимости от сложности и функций бота, можно сделать его монолитным или с помощью микросервисной архитектуры. Microservice architecture – принципиальная организация распределенной системы на основе микросервисов и их взаимодействия друг с другом и со средой по сети, а также принципов, направляющих проектирование архитектуры, ее создание и эволюцию [3].

В микросервисной архитектуре приложение разбивается на ряд независимо развертываемых сервисов, которые взаимодействуют с помощью API-интерфейсов. Благодаря такому подходу каждый отдельный сервис можно развертывать и масштабировать независимо от других. В результате программисты могут быстрее и чаще поставлять объемные и сложные приложения.

В отличие от монолитного приложения, с микросервисной архитектурой можно быстрее внедрять новые возможности и вносить изменения, при этом им не приходится переписывать большие фрагменты существующего кода.

Пример микросервисной архитектуры программного бота представлен на рис. 3 (бот изображен посередине).

В микросервисной архитектуре хорошим тоном считается использование контейнеризации микросервисов.

Docker – это платформа контейнеризации с открытым исходным кодом, с помощью которой можно автоматизировать создание приложений, их доставку и управление. Платформа позволяет быстрее тестировать и выкладывать приложения, запускать на одной машине требуемое количество контейнеров [4].

Особенности контейнеров:

- 1) сравнительно короткий жизненный цикл. Любой контейнер можно остановить, перезапустить или уничтожить, если это необходимо;
- 2) контейнеризация обеспечивает надежную изоляцию процессов и повышает уровень безопасности систем. Приложения, которые работают внутри контейнера, не имеют доступа к основной ОС и не могут на нее влиять;
- 3) благодаря контейнерам можно автоматизировать развертывание приложений;
- 4) использование контейнеров позволяет перейти с монолита на микросервисную архитектуру.

По своему функционалу и назначению контейнеры похожи на VM, которые действуют на основе аппаратной виртуализации. Однако, не следует напрямую отождествлять понятия "контейнеризация" и

“аппаратная виртуализация”. Это отдельные технологии, которые решают схожие задачи разными способами.

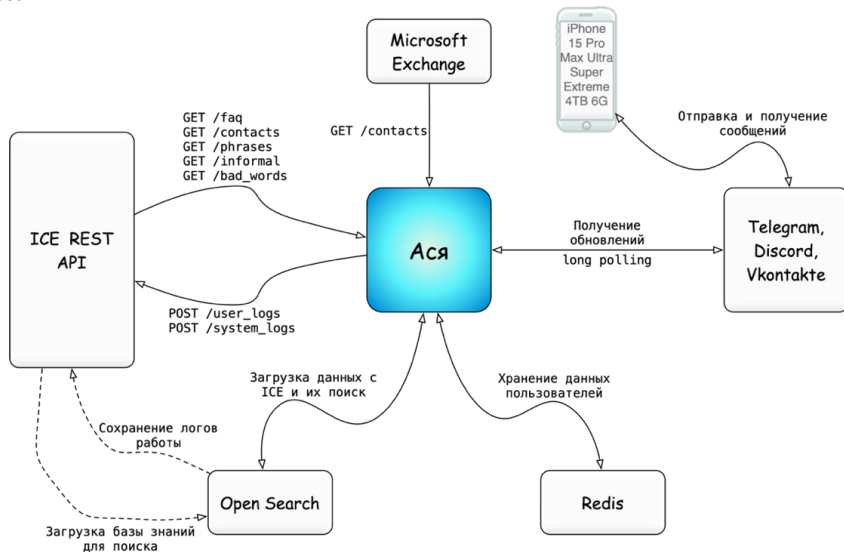


Рис. 3. Пример микросервисной архитектуры бота

В отличие от аппаратной виртуализации, контейнеризация обеспечивает разделение ресурсов не на аппаратном уровне, а на базе ядра операционной системы. Контейнеры более легковесны, менее требовательны и полностью зависят от “материнской” ОС, чем VM. На рис. 4 наглядно показаны различия виртуализации и контейнеризации [5].

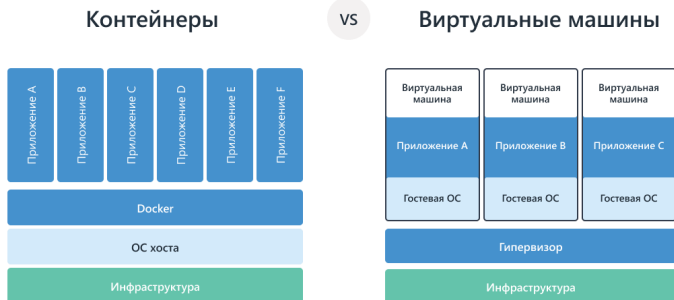


Рис. 4. Отличие контейнеризации от виртуализации (источник: eternalhost.net)

Виртуальная машина — это полноценная операционная система внутри другой ОС, с собственным ядром и другими изолированными ресурсами, а контейнер лишь изолированный механизм для запуска одного приложения.

Таким образом, чтобы сделать качественного, легко масштабируемого для разработчиков, что немаловажно, и надежного программного бота, нужно хорошо продумать системную архитектуру, знать принципы взаимодействия бота и пользователя через сервера социальных сетей и иметь знания о контейнеризации для более быстрого и легкого развертывания своего ПО.

**Библиографический список**

1. Чат-боты, виртуальные собеседники // Tadviser. URL: [https://www.tadviser.ru/index.php/Статья:Чат-боты\\_\(Chat-bot\)\\_Виртуальные\\_собеседники](https://www.tadviser.ru/index.php/Статья:Чат-боты_(Chat-bot)_Виртуальные_собеседники) (дата обращения: 15.05.2023).
2. Длинные опросы // Современный учебник Javascript. URL: <https://learn.javascript.ru/long-polling> (дата обращения: 16.05.2023).
3. Просто о микросервисах // Habr. URL: <https://habr.com/ru/company/raiffeisenbank/blog/346380/> (дата обращения: 15.05.2023).
4. Что такое Docker // Yandex Cloud. URL: <https://cloud.yandex.ru/blog/posts/2022/03/docker-containers> (дата обращения: 17.05.2023).
5. Docker и Kubernetes – чем отличаются технологии контейнеризации // EternalHost. URL: <https://eternalhost.net/blog/razrabotka/docker-kubernetes> (дата обращения: 18.05.2023).



## УДК 004.4

**И. М. Кудрин**

студент кафедры информационных систем и технологий

**В. А. Ушаков** – старший преподаватель – научный руководитель**РАЗРАБОТКА ИНТЕРФЕЙСА ПРИБОРНОЙ ПАНЕЛИ ЭЛЕКТРОМОБИЛЯ****Введение**

В настоящее время, электромобили являются актуальной и быстроразвивающейся отраслью цифровой экономики [1]. Электромобили – это автомобили, которые работают на электрической энергии, получаемой от аккумулятора, который заряжается от внешнего источника электроэнергии. С 2016 по 2022 год произошел значительный рост продаж электромобилей, что говорит о заинтересованности потребителей в данной отрасли (рис. 1). Прирост продаж в 2022 году относительно 2016 года составил около 760%.

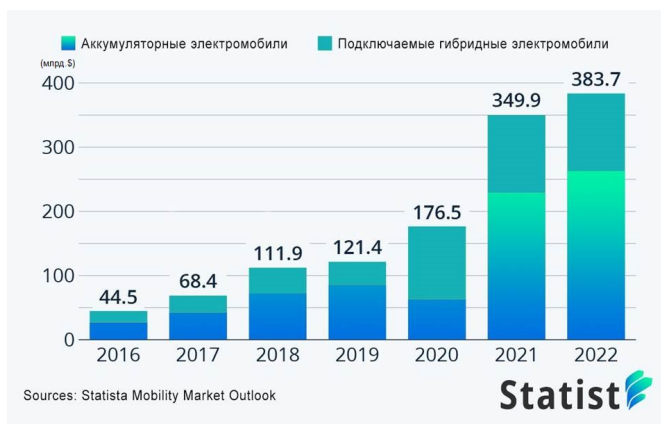


Рис. 1. Статистика продаж аккумуляторных и подключаемых гибридов электромобилей (с 2016 по 2022 гг.) [2]

Для возможности конкуренции России на данном рынке предполагается разработка и полное производство электромобилей на территории РФ [3]. Однако в рамках данной работы, рассматривается только решение задачи разработки интерфейса приборной панели электромобиля.

**Прототип приборной панели электромобиля**

Для электромобилей не подходит стандартная приборная панель от автомобилей, потому что некоторые важные параметры и показатели, возможно, демонстрировать только при помощи ЖК экрана такие как график расхода и рекуперации электроэнергии. Так же такой вариант приборной панели [4] дает возможность производителю предоставить дополнительную информацию водителю.

В ходе решения поставленной задачи был разработан дизайн интерфейса прототипа приборной панели электромобиля (рис. 2). В центральной части приборной панели располагается спидометр, справа и слева от него находятся информационные зоны, куда может выводиться различная текстовая и графическая информация, например, график потребления энергии, сведения о положении дверей электромобиля, навигатор или информация о входящих звонках. Сверху и снизу в ряд расположены основные пиктограммы, отвечающие за быстрое и понятное предоставление информации пользователю.



Рис. 2. Разработанный интерфейс приборной панели

Для простоты восприятия информации о скорости электромобиля на приборной панели используется как стрелочный спидометр, так и дублирующее его значение цифровая индикация. В правом сегменте расположена стрелка расхода энергии, показывающая информацию в кВт. Под цифровым значением скорости находится зеленый графический индикатор зарядки аккумулятора, цифровое табло запаса хода и селектор режима работы электродвигателей.

### Программная реализация прототипа

Для разработки дизайна интерфейса приборной панели были выбраны специальные инструментальные средства – графический редактор, дающий разработчику набор инструментов необходимых для создания и настройки внешнего вида и среда разработки для написания и отладки программного кода. Из ряда популярных графических редакторов, таких как: Adobe Illustrator, CorelDRAW, Inkscapе, Figma был выбран последний. Несмотря на то, что Adobe Illustrator и CorelDRAW являются профессиональными программами для решения подобных задач, основной минус этих приложений – это обязательная платная подписка. В то время как Inkscapе и Figma являются полностью бесплатными редакторами. Однако Inkscapе так же имеет ряд минусов, например, частые ошибки с сохранением файлов и относительно медленную работу приложения. Поэтому Figma [5] является оптимальным решением для поставленной задачи, которое имеет большое количество инструментов, функций и библиотек. Для написания программного кода приложения и отладки интерфейса приборной панели была выбрана среда разработки QT Creator [6]. QT Creator является бесплатной кроссплатформенной интегрированной средой разработки приложений, включающей в себя отладчик и визуальные средства разработки интерфейса. Данная среда поддерживает языки программирования C++ и QML [7], которые часто используются для разработки графических интерфейсов.

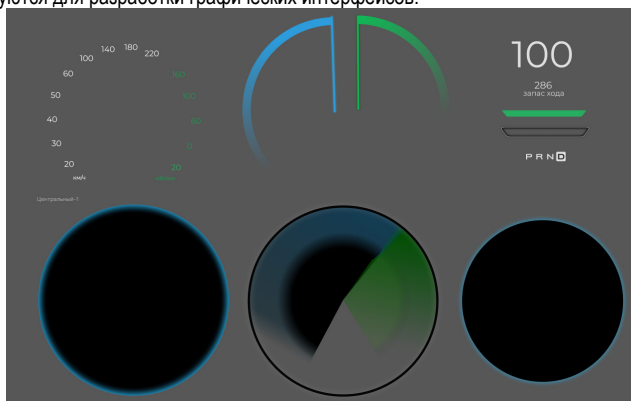


Рис. 3. Составляющие части интерфейса

В Figma был создан общий вид интерфейса и позже по нему нарисованы все составные элементы дизайна, такие как: задний фон, пиктограммы значков, задний фон спидометра и его стрелки, индикатор заряда аккумулятора и так далее (рис. 3). Все элементы были экспортированы в формат PNG для удобства дальнейшей работы.

### Тестирование и отладка прототипа

В кроссплатформенной интегрированной среде разработки QT Creator при помощи декларативного языка программирования QML были описаны все элементы приборной панели и их параметры. Далее при помощи языка C++ был написан программный код, вызывающий все описанные ранее файлы .qml. После этого была выполнена отладка приложения для выявления и исправления ошибок различного типа (рис. 4 и 5). Например, была исправлена ошибка отображения, при которой изображения пиктограмм ближнего и дальнего света фар накладывались друг на друга (рис. 6).

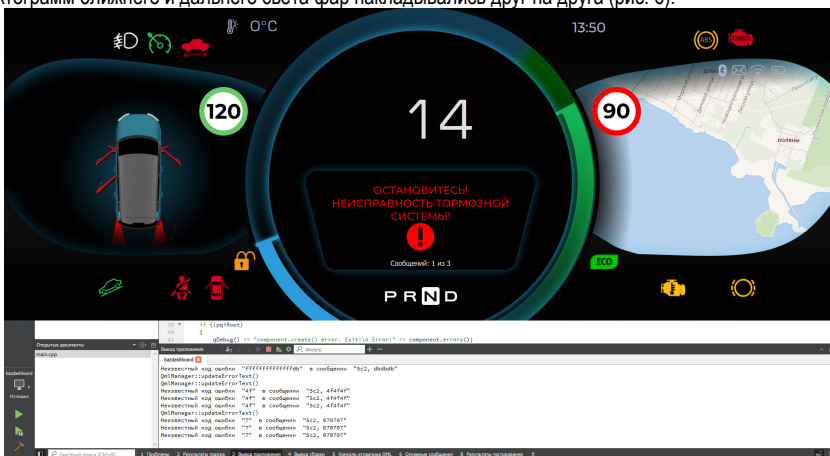


Рис. 4. Отладка прототипа приборной панели (сообщение об ошибке)

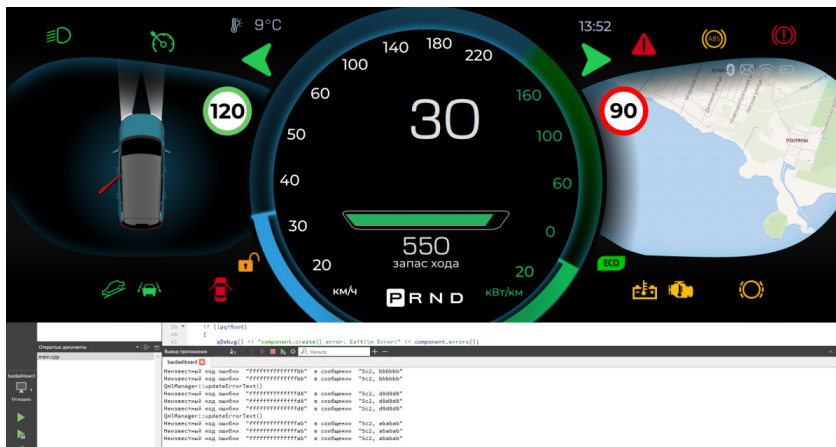


Рис. 5. Отладка прототипа приборной панели (спидометр)

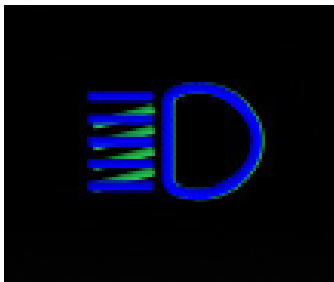


Рис. 6. Пример ошибки отображения пиктограмм

После завершения тестирования и отладки прототипа был собран рабочий экземпляр интерфейса приборной панели электромобиля.

### Заключение

В ходе работы был разработан новый оригинальный дизайн приборной панели для электромобиля российского производства. В дальнейшем планируется интегрировать поддержку других языков, а также написать документацию.

### Библиографический список:

1. Липидус Л. В. Эволюция цифровой экономики // Ломоносовские чтения-2018. Секция экономических наук. Цифровая экономика: человек, технологии, институты: сборник тезисов выступлений. М.: Экономический факультет МГУ имени М. В. Ломоносова, 2018. С. 153–158.
2. Электромобили (мировой рынок). URL: [https://www.tadviser.ru/index.php/Статья:Электромобили\\_\(мировой\\_рынок\)](https://www.tadviser.ru/index.php/Статья:Электромобили_(мировой_рынок)) (дата обращения: 15.05.2023).
3. Распоряжение Правительства РФ от 23.08.2021 № 2290-р // Собрание законодательства РФ. 2021. № 35. Ст. 6327.
4. Приборные панели: Методы оптимизации расположения органов управления / Д. В. Предтеченский, А. А. Волосюк, А. А. Балканский // Международная научная конференция по проблемам управления в технических системах. 2021. Т. 1. С. 161–165.
5. Figma. URL: <https://www.figma.com/> (дата обращения: 15.05.2023).
6. Qt Creator. URL: <https://www.qt.io/product/development-tools> (дата обращения: 15.05.2023).
7. Qt Documentation. URL: <https://doc.qt.io/qt-6/qtqml-index.html> (дата обращения: 15.05.2023).

УДК 629.3.018

**А. П. Куст**

студент кафедры информационных систем и технологий

**А. Д. Жуков** – кандидат технических наук, доцент – научный руководитель

### МОДЕЛЬ АНТИБЛОКИРОВОЧНОЙ СИСТЕМЫ АВТОМОБИЛЯ

Антиблокировочная тормозная система (ABS) – это электрогидравлическая система активной безопасности, позволяющая сохранить управляемость и устойчивость автомобиля при торможении за счет предотвращения блокировки колес. ABS особенно эффективна на дорожных покрытиях с невысоким коэффициентом сцепления, а также при плохой погоде (снег, гололед, дождь) [1].

При торможении фрикционные накладки тормозной колодки прижимаются к поверхности тормозного диска или тормозного барабана. При этом возникают силы трения, величина которых зависит от силы нажатия водителем на тормозную педаль. Произведение силы трения на плечо приложения этой силы в итоге дают тормозной момент  $M_B$ . Эффективный момент, возникающий в процессе торможения, действует в пятне контакта колеса с дорогой.

Тормозная сила  $F_B$  возникает между шиной и поверхностью дороги в результате приложения к колесу тормозного момента. Величину приложенной к поверхности дороги тормозной силы, обозначают как силу трения  $F_{тр}$ .

Во время торможения на колеса одновременно происходят два воздействия:

- 1) создается тормозной момент  $M_T$ , который препятствует вращению колеса;
- 2) в результате взаимодействия покрышки с дорогой возникает момент сцепления  $M_{сц}$  стремящийся поддержать вращение колеса.

Чрезмерное усилие, прилагаемое к тормозной педали при высокой скорости движения и соответствующих дорожных условиях, может привести к блокировке колес, если сила трения между шиной и проезжей части, обусловленная коэффициентом сцепления, окажется меньше тормозной силы. Если колесо автомобиля заблокировано, то оно оказывается не способным передавать никакие боковые силы, поэтому транспортное средство становится неуправляемым [2].

Для моделирования системы ABS использована среда имитационного блочного моделирования Simulink. За основу исследования взята модель антиблокировочной системы, предложенная разработчиками MATLAB. На рис. 1 представлена модель «Anti-lock Braking System».

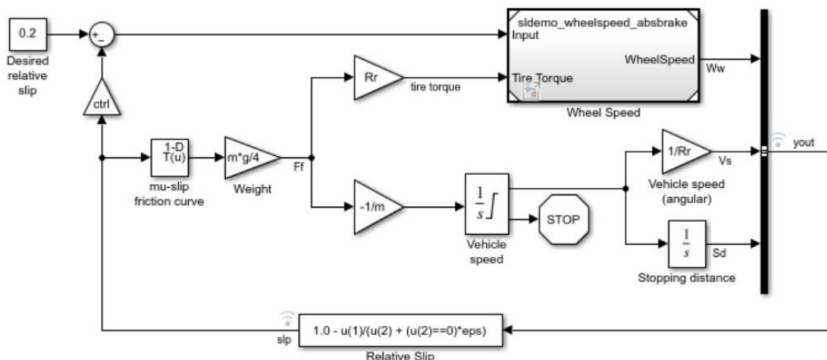


Рис. 1. Модель ABS

Данный пример представляет простую модель для антиблокировочной тормозной системы. Иллюстрирует поведение автомобиля при условиях резкого торможения. Эта модель рассчитана на

одно колесо, которое может быть дублировано, чтобы смоделировать многоколесное транспортное средство. Скорость колеса вычисляется в отдельном блоке «*slidemo wheelspeed absbrake*».

До нажатия на педаль тормоза, колесо имеет начальную угловую скоростью равную скорости автомобиля. Для вычисления угловой скорости колеса и скорости автомобиля используются отдельные интеграторы. При расчете скольжения (1) используем две скорости.

Скорость транспортного средства выражаем в виде угловой скорости:

$$\omega_v = \frac{V}{R}; \omega_w = \frac{V_w}{R_r}, \quad (1)$$

$$slip = 1 - \frac{\omega_w}{\omega_v}, \quad (2)$$

где  $\omega_v$  – угловая скорость автомобиля,  $V$  – линейная скорость автомобиля,  $R_r$  – радиус колеса,  $\omega_w$  – угловая скорость колеса

В модели применен идеальный антиблокировочный тормозной контроллер, который использует релейное управление, основанное на разнице между фактическим и желаемым проскальзыванием. Желаемое скольжение устанавливаем такое же, как и скольжение, при котором кривая *mu-slip* достигает пикового значения.

Модель (рис. 1) умножает коэффициент трения *mu* на вес колеса, чтобы получить силу трения, действующую на окружность шины. Сила трения делится на массу транспортного средства для получения замедления транспортного средства. Далее значение интегрируется для получения скорости.

Для контроля скорости изменения тормозного давления, модель вычитает фактическое скольжение из желаемого и подает сигнал в регулятор *bang-bang* (+1/-1, в зависимости от знака разности). Эта скорость включения/ выключения проходит через аperiodическое звено первого порядка, которая представляет собой задержку, связанную с гидравлическими линиями тормозной системы. Затем модель интегрирует расход, чтобы получить фактическое тормозное давление. Результирующий сигнал, умноженный на площадь поршня и радиус относительно колеса, представляет собой тормозной момент, приложенный к колесу.

Модель умножает силу трения на колесе на радиус колеса, чтобы дать момент сцепления дорожного покрытия на колесе. Тормозной момент вычитается, чтобы получить чистый крутящий момент на колесе. Деление чистого крутящего момента на момент инерции колеса дает ускорение колеса, которое затем интегрируется для обеспечения скорости колеса. Для поддержания положительной скорости колеса и скорости автомобиля в этой модели используются ограниченные интеграторы [3].

За основу моделирования возьмем базовую модель АБС (рис. 1), принцип работы которой был описан выше. Первым этапом модификации модели будет добавление дополнительного колеса. Для каждого из колес будет задаваться свой коэффициент сцепления. Это позволит смоделировать ситуацию, когда одна пара колес (переднее и заднее) будет наезжать на поверхность с отличным коэффициентом от второй пары колес. Таким образом, из одноканальной системы АБС будет получена двухканальная с разными сцеплениями правых и левых колес.

Вторым этапом модификации модели является внедрение *fuzzy-логики*. Система нечеткой логики создается и настраивается в приложении *Fuzzy Logic Designer*.

В блок *Fuzzy* подается два сигнала: скольжение и скорость. Это те переменные, которые подаются как входные данные в системе нечеткой логики. Для передачи данных создано 2 глобальных переменных, значения которых вычисляются в рабочей области модели.

На рис. 2 показана полученная АБС модель. В верхней части модели находится мануальный переключатель между релейным управлением и системой нечеткой логики. С его помощью мы сможем сравнивать работу АБС в обоих случаях. Система содержит в себе 2 пары подсистем. Одна пара отвечает за расчет момента сцепления колес с поверхностью для каждого из колес (рис. 3), а вторая вычисляет скорость колес (рис. 4).

Произведем тестирование модели при различных состояниях дорожного полотна. С помощью коэффициента сцепления будем имитировать сухое, мокрое и обледеневшее дорожное полотно. Те-

стирование будем проводить для обоих вариантов модели (с релейным управлением и с fuzzy-логикой), сравним результаты и проведем их анализ.

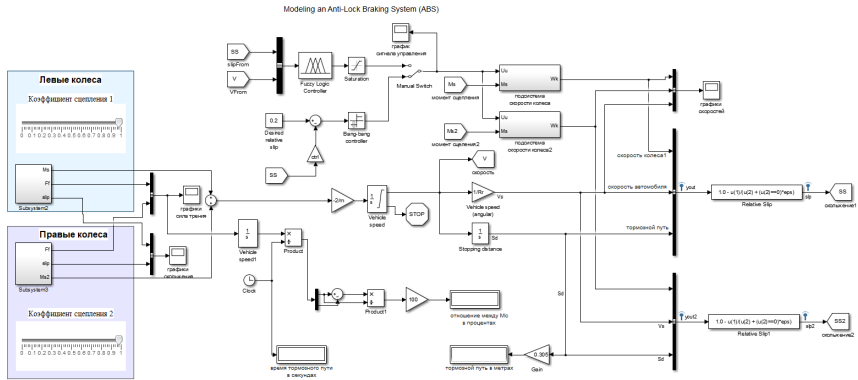


Рис. 2. Модифицированная АБС модель

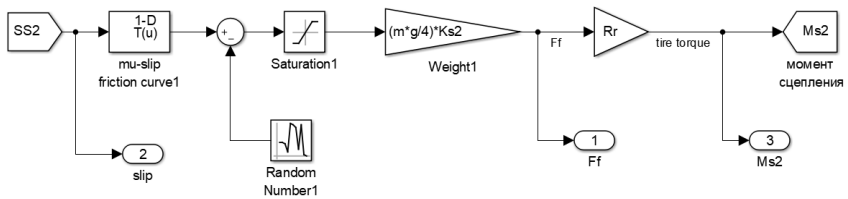


Рис. 3. Подсистема момента сцепления

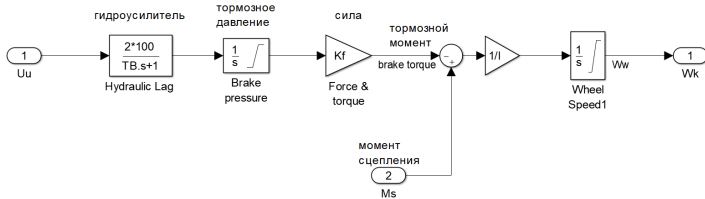


Рис. 4. Подсистема скорости колеса

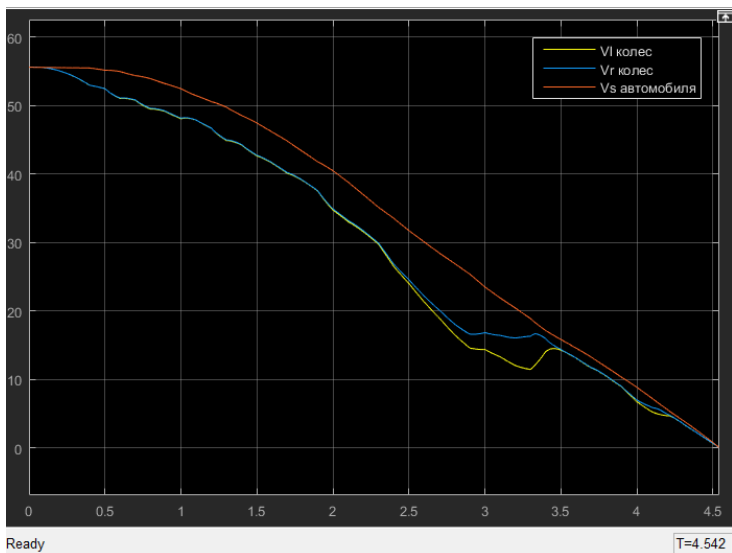
Тестирование модели будет осуществлено на асфальтированной дороге при начальной скорости 160км/ч (или 100 миль/час) в следующих ситуациях:

1. Все колеса находятся на сухом дорожном полотне.
2. Все колеса находятся на мокром дорожном полотне.
3. Левые колеса находятся на сухом, а правые на обледеневшем дорожном полотне.

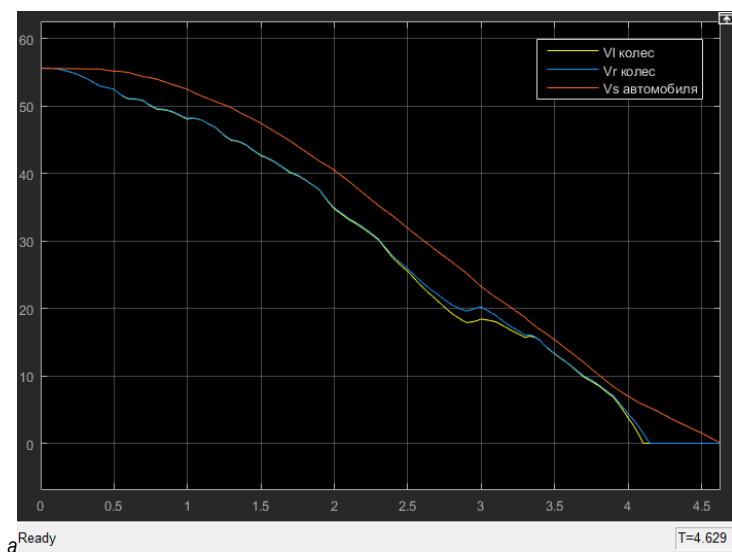
### Испытание 1

На рис. 5 показаны графики скоростей: скорость автомобиля и скорости левых и правых колес. Слева график модели с релейным управлением, справа – с системой нечеткой логики. В случае с fuzzy-

логикой, колеса были заблокированы примерно с 4 секунды и до полной остановки машины. При релейном управлении блокировки не произошло.



а)

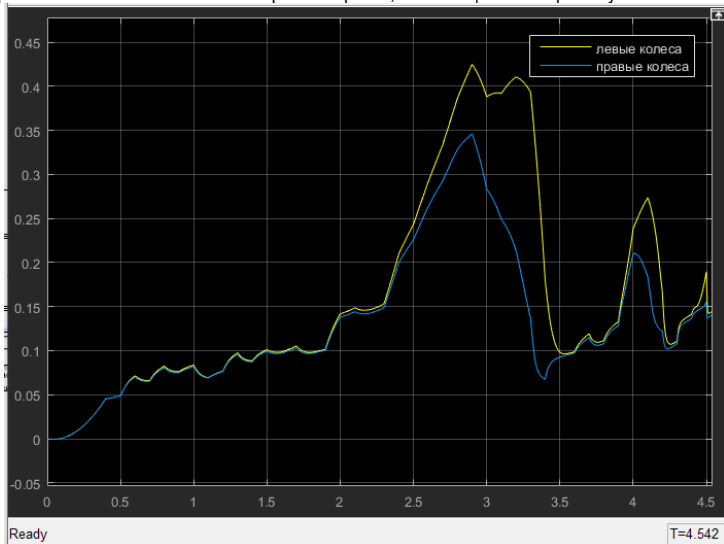


б)

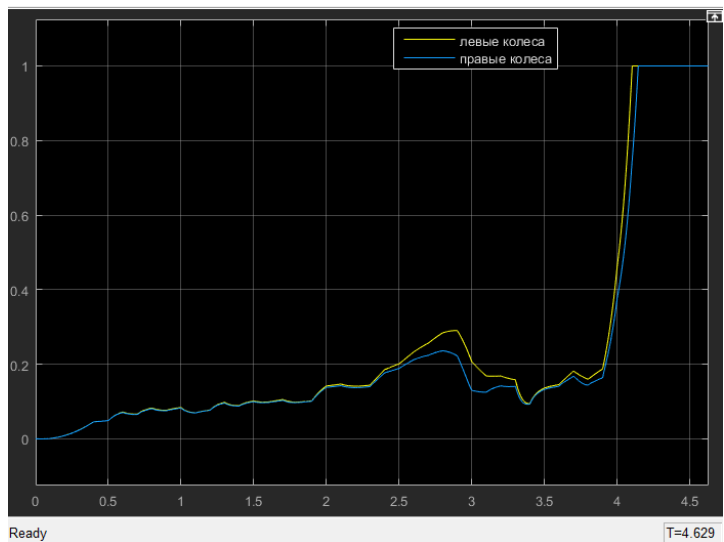
Рис. 5. Графики скоростей: а – релейное упр.; б – справа – fuzzy



На рис. 6 показаны графики скольжения. При релейном управлении в середине тормозного пути произошел скачок значения скольжения в 2 раза, но колеса в юз не ушли. При управлении с fuzzy-регулятором значение скольжения было равномерным, но в конце колеса резко ушли в юз.



а)

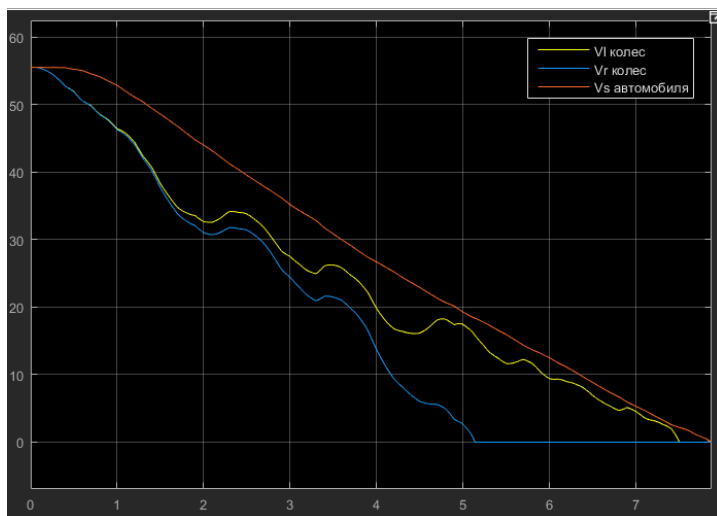


б)

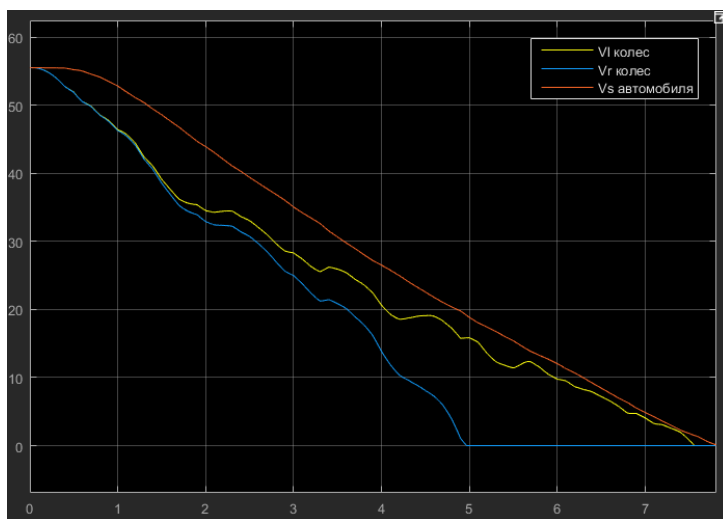
Рис. 6. Графики скольжения: а – релейное упр., б – fuzzy

Испытание 2

На рис. 7 показаны графики скоростей. В данных условиях тестирования графики почти идентичны, в обоих случаях правые колеса оказались заблокированы ~ с 5 секунды и до конца движения автомобиля, а левые колеса заблокировались в самом конце пути с ~ с 7,5 секунды.



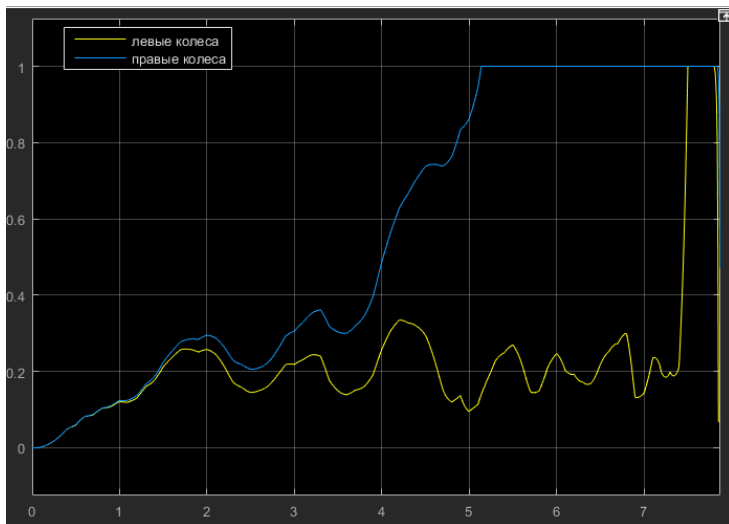
а)



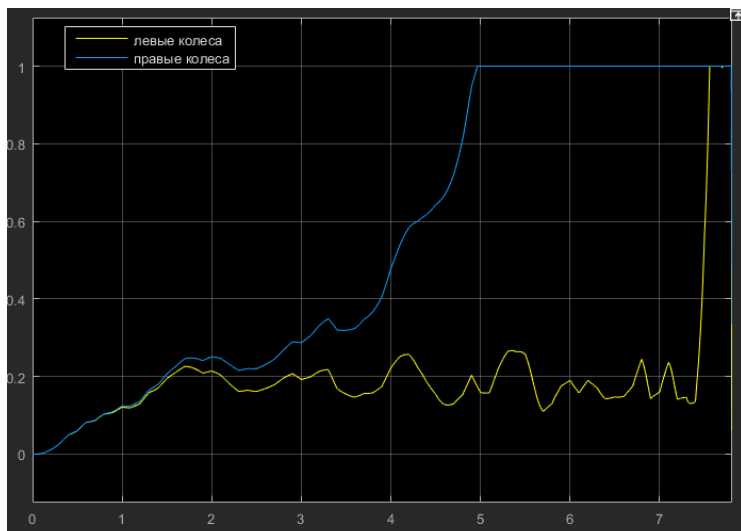
б)

Рис. 7. Графики скоростей: а – релейное упр., б – fuzzy

На рис. 8 показаны графики скольжения. В обоих случаях правые колеса ушли в юз с 5 секунды, а левые с 7,5 секунды. При этом при управлении с fuzzy-логикой амплитуда колебаний значения скольжения левых колес чуть меньше, нежели при релейном управлении.



а)

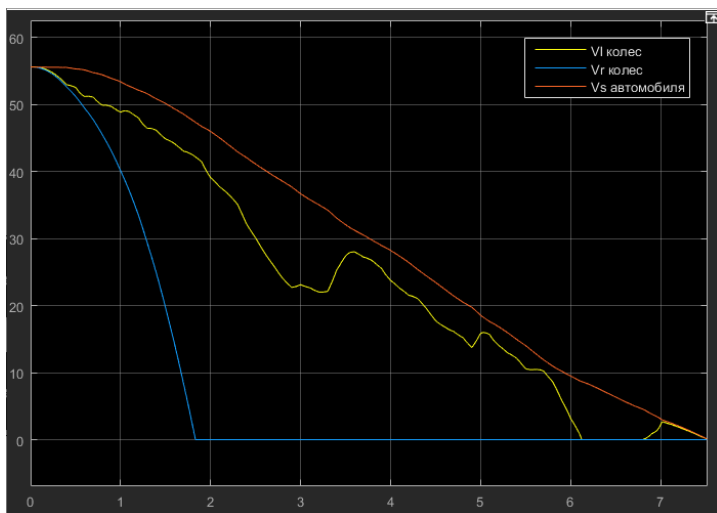


б)

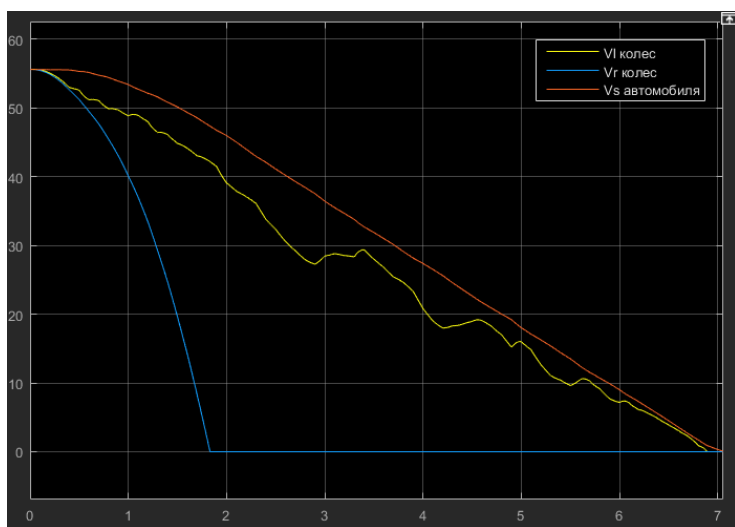
Рис. 8. Графики скольжения: а – релейное упр., б – fuzzy

## Испытание 3

На рис. 9 показаны графики скоростей для обоих вариантов модели: с релейным управлением и с системой нечеткой логики. Как видно, они почти идентичны, отличается только скорость левых колес. При релейном управлении левые колеса оказались заблокированы ~ с 6 по 7 секунду.



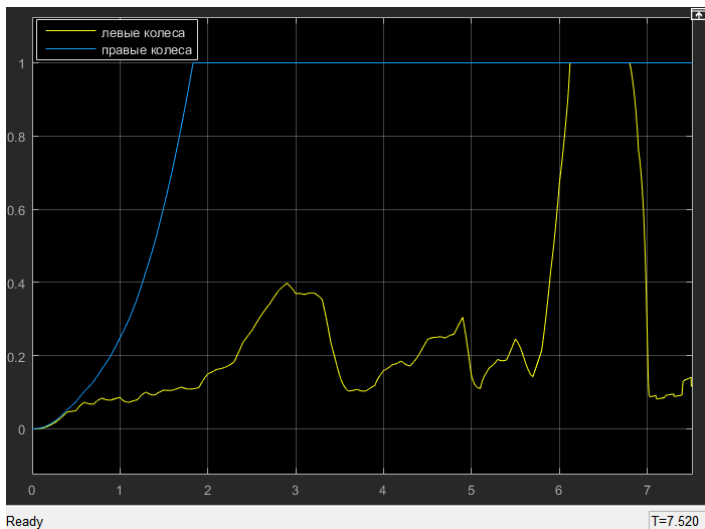
а)



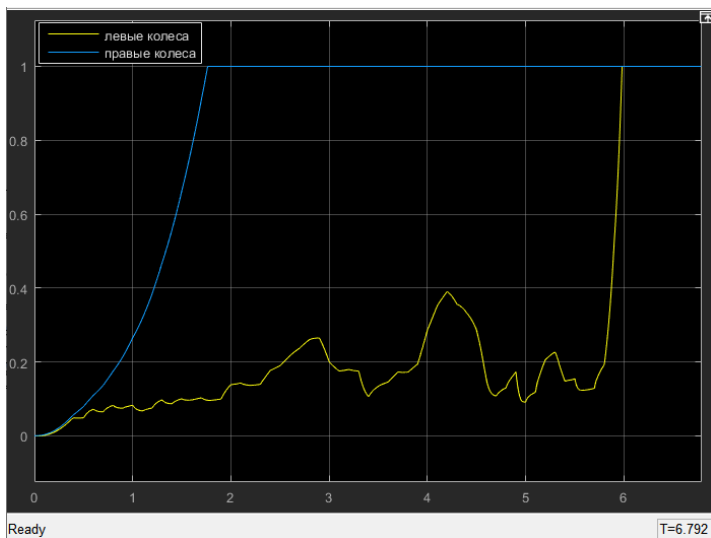
б)

Рис. 9. Графики скоростей: а – релейное упр., б – fuzzy

На рис. 10 показаны графики скольжения. При релейном управлении правые колеса ушли в юз примерно с 6 по 7 секунды, а в fuzzy юза не произошло. В обоих случаях правые колеса ушли в юз примерно на 1,8 секунде.



а)



б)

Рис. 10. Графики скольжения: а – релейное упр., б – fuzzy

Ниже показана сравнительная табл. 1 с конечными результатами.

Таблица 1

**Сравнительная таблица**

Испытание			Релейное управление			fuzzy-логика		
№	$k_1$	$k_2$	$S$ , м	$t$ , с	Юз колес	$S$ , м	$t$ , с	Юз колес
1	0,8	0,8	83,5	4,5	нет	82,8	4,63	да
2	0,4	0,4	123	7,9	да	121	7,8	да
3	0,8	0,1	122,4	7,52	да	120	7,56	нет

В результате тестирования видно, что в обоих моделях в 2 из 3 случаев происходила блокировка колес. Поэтому критерием оценивания становится значение тормозного пути, в котором лучшие результаты показала модель с fuzzy-управлением, сократив путь примерно на 2 метра во всех испытаниях.

**Библиографический список**

1. Устройство и принцип работы системы ABS. URL: <https://techautoport.ru/hodovaya-chast/tormoznaya-sistema/sistema-abs.html> (дата обращения: 15.03.2023).
2. Антиблокировочная система (АБС, ABS): торможение без юза. URL: <https://www.autoopt.ru/articles/products/4513916> (дата обращения: 15.03.2023).
3. Документация Simulink. Моделирование антиблокировочной тормозной системы. URL: <https://docs.exponenta.ru/simulink/slref/modeling-an-anti-lock-braking-system.html> (дата обращения: 16.03.2023).

УДК 004.773

**И. И. Микрюков**

студент кафедры информационных систем и технологий

**О. И. Красильникова** – кандидат технических наук, доцент – научный руководитель

## СОЗДАНИЕ ВЕБ-ЧАТА С ПОМОЩЬЮ WEBSOCKET НА ЯЗЫКЕ GOLANG

### Введение

Данная работа посвящена вопросам разработки веб-чата, который можно использовать для прямой трансляции или в других системах, где необходимо обмениваться информацией в реальном времени с минимальной задержкой между большим количеством пользователей. На основе сравнительного анализа различных технологий передачи данных между сервером и клиентом выбран наиболее подходящий протокол для разработки такого веб-чата и описана работа как со стороны сервера, так и со стороны клиента.

### Сравнение протоколов обмена данными в сети Интернет

Существуют различные способы обмена данными в сети Интернет. Рассмотрим и сравним самые популярные из них и на этой основе сделаем выбор. При сравнении протоколов использовалась информация с сайтов [1] и [2]. Самый используемый протокол передачи данных между клиентом и сервером, это HTTP (HyperText Transfer Protocol — «протокол передачи гипертекста»). HTTP – является однонаправленным протоколом передачи данных, хорошо подходит, если необходимо единожды сделать запрос информации с сервера, после чего соединение будет автоматически разорвано. На рис. 1 схематично представлена передача данных по HTTP протоколу. Для передачи большого объема информации в реальном времени значительному количеству пользователей такая передача не подходит, так как имеет следующие недостатки:

- 1) нет постоянного двунаправленного соединения;
- 2) необходимо часто отправлять запросы на сервер, чтобы проверить, есть ли новая информация;
- 3) каждый раз отправляется заголовок, где указывается, какой тип запроса хотим сделать;
- 4) медленная скорость работы и лишняя нагрузка на сеть;
- 5) перезагрузка страницы после выполнения запроса.

Есть некоторое усовершенствование HTTP запросов, это асинхронные запросы с использованием JavaScript и XML (eXtensible Markup Language – «расширяемый язык разметки»), т. е. AJAX (Asynchronous Javascript and XML – «асинхронный JavaScript и XML») запросы. Они решают проблему того, что автоматически получают и отправляют данные без перезагрузки всей веб-страницы, а только некоторой части. Так как это асинхронные запросы, то пользователь во время выполнения запроса может совершать другие действия на веб-странице. Ответ от сервера можно получать не только в формате XML, но также в виде обычного текста и в формате JSON (JavaScript Object Notation – «текстовый формат обмена данными, основанный на JavaScript»). За счет асинхронных запросов последнюю проблему можно считать решенной, но предыдущие недостатки все-таки заставляют сделать вывод о нецелесообразности использования протокола HTTP при создании веб-чата.

Еще один протокол передачи данных, заслуживающий внимания для решения поставленной задачи, это протокол WebSocket. Проанализировав информацию с сайта [3], отметим, что в данном протоколе проблемы, присущие протоколу HTTP, решены. Во-первых, протокол WebSocket является двунаправленным, т. е. можно одновременно отправлять и получать данные. На рис. 2 схематично представлена передача данных по протоколу WebSocket. Кроме того, установленное соединение можно использовать повторно, что повышает скорость работы и уменьшает нагрузку на сеть. Пользователь отправляет на сервер запрос, при котором происходит как бы «рукопожатие» с сервером и «согласование» того, что будет использоваться WebSocket протокол. Также данный протокол поддерживает возможность сжатия данных. В протоколе реализован механизм, проверяющий подключен еще пользователь или нет. Он работает следующим образом: сервер периодически посылает запрос пользователю,

чтобы тот сделал "ответный" запрос на сервер, если ответ в течение некоторого времени не приходит, то происходит разрыв соединения. С его помощью можно реализовать не только веб-чат, но и другие системы, где передается большое количество информации между пользователями и сервером, например: биржи, push-уведомления, игровые приложения.

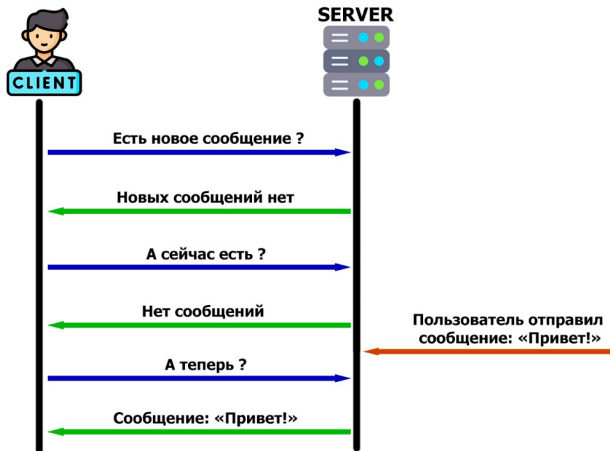


Рис. 1. Передача сообщения по протоколу HTTP

Рассмотрение особенностей протокола WebSocket показывает, что его использование в веб-чате более целесообразно, чем HTTP.

После выбора протокола перейдем к описанию технической части реализации веб-чата. Рассмотрим вопросы, связанные с проектированием серверной и клиентской сторон веб-чата.

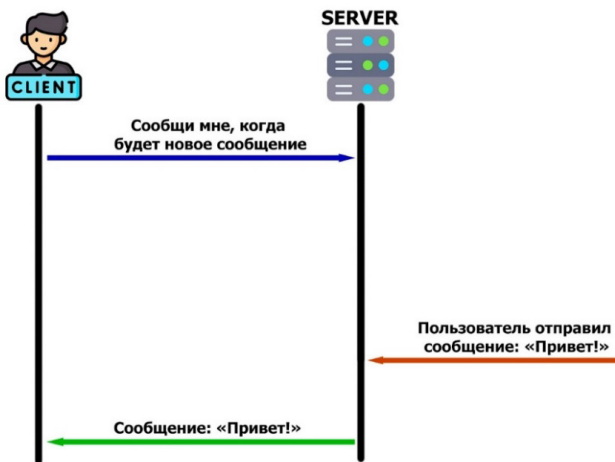


Рис. 2. Передача сообщения по протоколу WebSocket



Программный модуль сервера, обрабатывающего сообщения от пользователей, было решено написать на языке программирования Golang, так как его производительность в некоторых случаях выше, чем у языков PHP, JavaScript, Python и близка к C++. Golang – это компилируемый многопоточный язык, в котором есть такое понятие как goroutines (горутины). Горутины — это легкие потоки, которые реализуют конкурентное программирование. С их помощью можно создать большое количество потоков и при этом потратить меньше памяти, чем если бы это реализовывалось на других языках программирования. Если рассматривать это в контексте веб-чата, то для масштабирования данной системы, горутины отлично подойдут.

При написании сервера использовалась информация, представленная на сайте [4]. Для реализации соединений с сервером по WebSocket была выбрана библиотека Gorilla WebSocket. Более подробная информация для работы с данной библиотекой имеется на сайте [5]. Библиотека имеет хорошую документацию и в настоящее время пользуется большой популярностью.

Рассмотрим детально, как выполняются этапы работы с созданным сервером.

- На сервере есть функция, которая принимает HTTP запросы и обновляет их до WebSocket, т. е. совершает “рукопожатие”.

- Далее после успешного установления соединения, клиент записывается в массив подключенных в данный момент к серверу пользователей.

- Также есть функция, которая принимает сообщения от пользователя и передает их в канал данных.

- Следующая функция читает данные из канала и отправляет их всем пользователям, которые находятся в массиве активных пользователей.

- Имеется также функция, которая проверяет соединение с пользователями и разрывает его, если они неактивны.

- Сервер при получении нового сообщения от клиентов сдвигает значения ячеек массива с последними 10 сообщениями на одну позицию вверх пока не дойдет последнего элемента, а в последнюю ячейку массива будет записываться только что полученное новое сообщение от клиента.

- При подключении к чату новый клиент будет получать последние 10 сообщений.

Теперь перейдем к рассмотрению клиентской части веб-чата. Она написана на JavaScript и работает в браузере. Изначально создается переменная типа WebSocket. Так как данные с сервера на клиента отправляются в JSON формате, то в функции получения сообщений, которая запускается при совершении события “получения сообщения” в вебсокет, необходимо распарсить полученные с сервера данные (записать данные в определенную структуру), после чего записать в элементы интерфейса.

Кроме того, имеется функция для отправки сообщения от клиента на сервер. В ней данные, полученные из пользовательского интерфейса, записываются в структуру, которая переводится в JSON формат и передается на сервер. На рис. 3 показан веб-интерфейс разработанного чата.

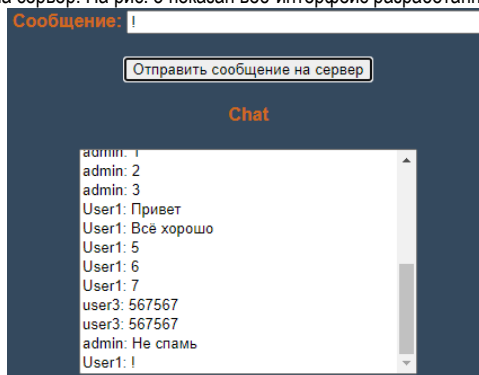


Рис. 3. Работа веб-чата в браузере

Подводя итог проделанной работе, можно сказать, что обоснованный выбор протокола WebSocket, а также проектирование серверной и клиентской сторон веб-чата, подробное описание которых приведено в данной статье, позволили создать полностью рабочий веб-чат, обеспечивающий обмен информацией между большим количеством пользователей в реальном времени с минимальной задержкой.

#### **Библиографический список**

1. Почему протокол WebSocket – лучший выбор: сравнение с HTTP. URL: <https://webformymself.com/websockets-vs-http/> (дата обращения: 05.05.2023).
2. WebSocket (WSS): что это и как работают сокет, асинхронный сервер. URL: <https://blog.skillfactory.ru/glossary/websocket/> (дата обращения: 05.05.2023).
3. WebSocket. URL: <https://learn.javascript.ru/websocket> (дата обращения: 06.05.2023).
4. Простейший сервер на Gorilla WebSocket // Хабр. URL: <https://habr.com/ru/articles/599737/> (дата обращения: 06.05.2023).
5. Websocket package – github.com/gorilla/websocket – Go Packages. URL: <https://pkg.go.dev/github.com/gorilla/websocket> (дата обращения: 07.05.2023).

УДК 004.773.5

**И. И. Микрюков**

студент кафедры информационных систем и технологий

**О. И. Красильникова** – кандидат технических наук, доцент – научный руководитель**СОЗДАНИЕ ПРЯМОЙ ТРАНСЛЯЦИИ С ИСПОЛЬЗОВАНИЕМ NGINX-RTMP-MODULE****Введение**

Передача видео и звука по сети в реальном времени используется во многих приложениях и сайтах, которые работают как на стационарных компьютерах, так и на телефоне.

Существуют различные типы прямых трансляций, например, peer to peer подключения для общения между двумя людьми в мессенджере. В данном случае часто используется технология WebRTC (real-time communications – коммуникации в реальном времени). Такой тип подключения хорошо подходит для небольшой группы людей и не требует отдельно сервера для обработки видео. Также есть прямые трансляции, где один вещающий пользователь и много слушателей, подобные трансляции использует YouTube, Twitch.

Для создания сервиса с прямой трансляцией для очень большого количества людей уже необходимо использовать сервер, который будет обрабатывать получаемое видео и отправлять всем пользователям. В таком случае используются дополнительные сервисы, которые могут это реализовать за определенную плату или пишутся свои решения, что довольно затратно.

В данной статье рассмотрена задача создания прямой трансляции, а также выбор технологий и инструментов, которые позволят это реализовать за короткий промежуток времени и будут бесплатными. После необходимо будет сделать тесты задержки видео на трансляции и подобрать подходящие настройки, чтобы как можно сильнее уменьшить задержку.

Рассмотрим прямую трансляцию, когда ведущий отправляет видео на сервер, сервер обрабатывает его и рассылает слушателям в браузер. В данном случае будет один источник видео и много слушателей, т. е. это пример обычной прямой трансляции для большого количества людей. Дополнительную информацию можно найти на сайте [1]. Подробную информацию об установке сервера можно найти на сайте [2].

Перед тем как видео дойдет до зрителя, ему необходимо пройти несколько этапов, (рис. 1):

– с помощью энкодера (в данном случае используется программа OBS (Open Broadcaster Software)) видео записывается, сжимается, преобразуется в заданный формат;

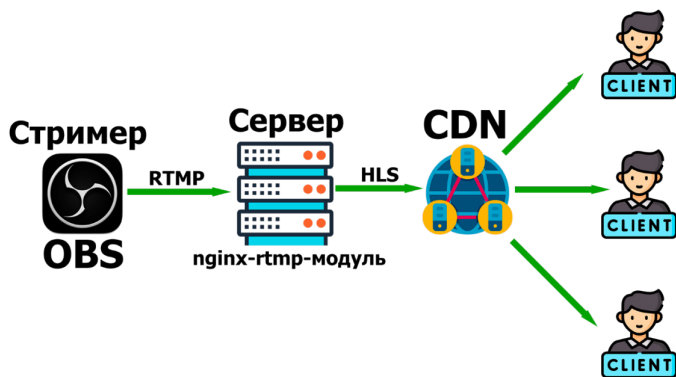


Рис. 1. Этапы доставки видео

- данные отправляются на сервер для дальнейшей обработки по протоколу RTMP (Real Time Messaging Protocol);
- сервер, получив данные, может обработать их с целью, например, изменения. После сервер транскодирует видео в определенные форматы, которые могут воспроизводить зрители;
- транскодированное видео передается по определенным протоколам (в данном случае используется протокол HLS (HTTP Live Streaming)) передачи видео зрителю. Часто используются CDN (Content Delivery Network), что представляет собой систему серверов, расположенных в разных физических местах.

### Протоколы передачи видео в сети Интернет

Сначала рассмотрим протокол RTMP. Это протокол обмена информацией в реальном времени. Поддерживается на различных платформах и имеет совместимость с разными устройствами записи и серверами обработки видео.

Данные передаются с задержкой около 5 секунд, создается постоянный поток данных пользователей. Но этот протокол в данный момент уже мало где поддерживается в браузерах. Данные необходимо преобразовывать, чтобы их могло воспроизвести как можно большее число устройств, и передавать уже по другим протоколам.

Для решения этой задачи используются следующие протоколы: DASH (Dynamic Adaptive Streaming over HTTP) или HLS. В данном случае используется протокол HLS так как его было проще настроить. Этот протокол хорошо подходит для трансляции очень большому количеству людей. Его поддерживают много устройств и браузеров. Он поддерживает передачу видео с адаптивным битрейтом, т. е. он будет изменяться в зависимости от скорости интернет-соединения.

Изначально задержка доставки видео могла достигать десятков секунд, что явно не подходит для прямых трансляций. Но в новой версии, которую поддерживают все браузеры и мобильные устройства, можно добиться времени задержки, не превышающего двух секунд.

В данном протоколе видео делится на фрагменты заданной длины. Также создается индексный файл, в котором хранится порядок фрагментов, он имеет заданную длину. Клиент скачивает с сервера индексный файл и доступные на данный момент фрагменты видео. Индексный файл содержит ссылки на несколько последних фрагментов. При повторном обращении к индексному файлу ссылки будут в нем обновляться на новые доступные фрагменты. Зная последовательность фрагментов из индексного файла и имея несколько последних, браузер (плеер) добавляет эти фрагменты в буфер воспроизведения в необходимом порядке. Аналогично работает доставка видео по протоколу DASH.

### Измерение задержки на прямой трансляции

В данном случае используется Ubuntu server, на который установлен веб-сервер NGINX. Для обработки видеопотока используется NGINX-RTMP-модуль, который позволяет принимать по RTMP протоколу видео, обрабатывать его, и отправлять пользователям в браузер по протоколу HLS или DASH с различными настройками. Больше информации о возможных настройках и параметров можно узнать на сайтах [3] и [4]. Ниже показаны настройки сервера, а на рис. 3 приведен программный код для работы плеера в браузере.

При настройках, указанных на рис. 2, задержка составляет 5,15 секунд, и трансляция работает стабильно. Настройки прямой трансляции вещающего в программе OBS:

- разрешение 1280x720,
- 30 кадров в секунду,
- битрейт 3000 Мбит/сек,
- ключевой кадр каждые 2 сек.

При настройках (hls\_fragment 1, hls\_playlist\_length 2), задержка составляет 3,54 секунд, но возможны появления "подгрузок" видео, которые скорее всего связаны с малой мощностью используемого сервера, в данном случае сервер был установлен на виртуальной машине. Настройки прямой трансляции вещающего в программе OBS:

- разрешение 1280x720,
- 30 кадров в секунду,

- битрейт 3000 Мбит/сек,
- ключевой кадр каждую 1 сек.

```
GNU nano 6.2 nginx.conf
worker_processes auto;
events {
    worker_connections 1024;
}

# RTMP configuration
rtmp {
    server {
        listen 1935; # Listen on standard RTMP port
        chunk_size 4096;

        application show {
            live on;
            # Turn on HLS
            hls on;
            hls_path /nginx/hls/;
            hls_fragment 2;
            hls_playlist_length 4;
        }
    }
}
```

Рис. 2. Настройки nginx-rtmp-модуля

```
function vvv() {
    if (Hls.isSupported()) {
        var video = document.getElementById('video');
        var hls = new Hls();
        hls.lowLatencyMode = true;
        hls.loadSource('http://192.168.0.171:8080/hls/test.m3u8');
        hls.attachMedia(video);
        hls.on(Hls.Events.MANIFEST_PARSED, function () { video.play(); });
    }
}
```

Рис. 3. Настройка плеера в браузере

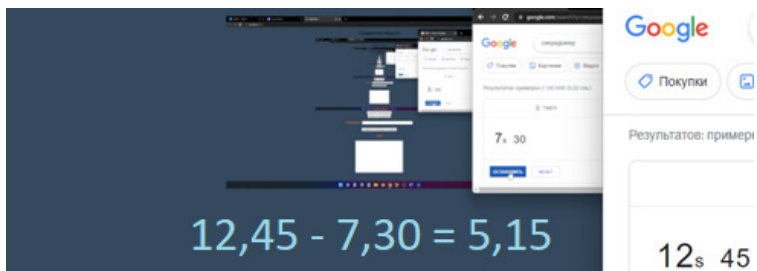


Рис. 4. Задержка 5,15 сек

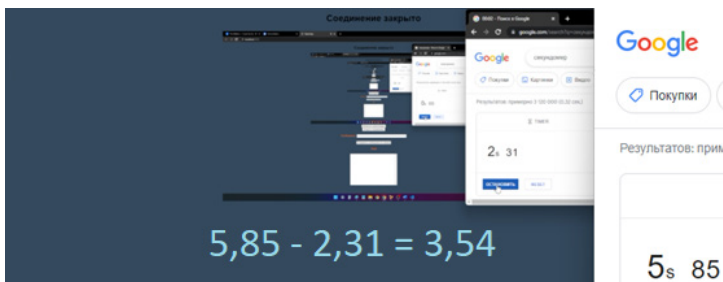


Рис. 5. Задержка 3,54 сек

С помощью каких команд можно обработать видео на сервере можно узнать на этом сайте [5]. В данном случае изменения видео на сервере не происходит и получаемое видео без изменений его параметров, например, битрейта будет отправляться зрителю, чтобы снизить время задержки на трансляции.

### Заключение

В итоге мы смогли создать прямую трансляцию используя бесплатные инструменты. Провести тесты времени задержки и подобрать значения параметров на стороне сервера и вещающего для уменьшения задержки на трансляции.

### Библиографический список

1. Как делать прямые трансляции? Протоколы, кодирование, CDN. URL: <https://stream-park.ru/blog/kak-delat-priamye-translyatsii-terminy-protokoly-kodirovanie-cdn/> (дата обращения: 02.05.2023).
2. NGINX + rtmp. Трансляция видео с веб-сервера. Установка и настройка nginx-rtmp-module. URL: <https://www.dmosk.ru/instruktsions.php?object=nginx-rtmp> (дата обращения: 02.05.2023).
3. arut/nginx-rtmp-module: NGINX-based Media Streaming Server. URL: <https://github.com/arut/nginx-rtmp-module> (дата обращения: 03.05.2023).
4. Directives arut/nginx-rtmp-module Wiki. URL: <https://github.com/arut/nginx-rtmp-module/wiki/Directives> (дата обращения: 03.05.2023).
5. Онлайн вещание через Nginx-RTMP: несколько готовых рецептов // Хабр. URL: <https://habr.com/ru/articles/174089/> (дата обращения: 04.05.2023).

УДК 004.02

**Д. А. Остапченко**

студент кафедры информационных систем и технологий

**Н. В. Богословская** – кандидат технических наук, доцент – научный руководитель**МЕТОДЫ ВЕКТОРИЗАЦИИ ТЕКСТА НА ЕСТЕСТВЕННОМ ЯЗЫКЕ****Введение**

Текстовое представление данных является наиболее широко используемой формой общения и выражения. Эти данные растут с каждым днем, и потребность в поиске инновационных механизмов и методов обработки, хранения, понимания и извлечения из них скрытой информации возрастает.

Для применения различных методов машинного обучения и интеллектуального анализа данных, документы необходимо преобразовать в формат, понятный машине. Первым шагом к тому, чтобы сделать текстовые документы машиночитаемыми, является векторизация, которая определяется как преобразование текстового документа в цифровой вектор, и представляет собой процесс извлечения признаков из текста для выполнения любых задач интеллектуального анализа текста и математического решения проблем.

В статье выполнен обзор основных методов векторизации текста для последующего выбора наиболее подходящего при реализации виртуального ассистента.

**Прямое кодирование**

Самым простым и примитивным подходом к векторизации слов является one-hot encoding или прямое кодирование. Он предполагает, что формируется словарь, содержащий все токены используемых документов. Для каждого токена определяется его номер – позиция в словаре. Токену ставится в соответствие вектор, в котором все компоненты нулевые за исключением одной, соответствующей номеру токена [1].

**Мешок слов**

Для создания вектора, который кодировал бы всю фразу, достаточно объединить one-hot-векторы каждого слова, входящего в состав фразы. Соответственно, предложение, закодированное таким же образом, будет суммировать все векторы слов, входящих в него [2].

Данный метод носит название bag of words или мешок слов, потому что в нем не используется никакая информация о порядке слов или их структуре. Для создания признаков данная модель использует только информацию о нахождении того или иного слова в тексте.

**TF-IDF**

TF-IDF – статистическая мера, которая используется для оценки важности слова в контексте документа, являющегося частью набора документов.

Для того, чтобы решить проблему зависимости важности слов от объема текста вводится понятие TF (англ. term frequency – частота слова).

TF определяется по формуле:

$$TF = \frac{n_t}{\sum_{i=1}^k n_i},$$

где  $n_t$  – количество отдельных слов в документе,  $\sum_{i=1}^k n_i$  – общее количество всех слов в документе.

Однако TF показывает только то, на сколько слово важно для конкретного документа, но для решения задачи используется большое количество документов. Необходимо также учитывать то, на сколько конкретное слово часто встречается и в остальных документах, ведь ценность слова будет

настолько важна для конкретного текста, насколько оно будет реже встречаться в остальных текстах. Исходя из этого вводится еще одно понятие – IDF (англ. inverse document frequency – обратная частота документа).

Обратная частота документа определяется по формуле:

$$IDF = \log \left( \frac{n_c}{\sum_{j=1}^m n_j} \right),$$

где  $n_c$  – количество документов всего,  $\sum_{j=1}^m n_j$  – количество документов, в которых содержится интересное слово [3].

Таким образом, TF-IDF – это произведение TF на IDF:

$$TF - IDF = TF \times IDF$$

### Векторное представление слов

Методы прямого кодирования, мешка слов и TF-IDF не учитывают семантику слова, разные слова с близкими значениями будут далеко друг от друга в векторном пространстве. Еще одна проблема заключается в том, что данные методы не учитывают порядок слов во фразе.

Эти проблемы решает word embedding или векторное представление слов – метод обучения, при котором каждое слово или фраза из словаря отображается в вектор размерности N действительных чисел. Подобные векторы можно складывать, вычитать и сравнивать. Например, если из слова «Король» вычесть слово «Мужчина» и прибавить слово «Женщина», то получится «Королева».

На рис. 1 представлен пример векторного представления слов, отражающий связь между словами «Мужчина» и «Женщина», «Король» и «Королева».

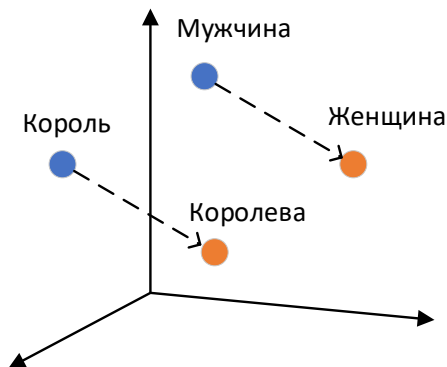


Рис. 1. Пример векторного представления слов

На базе векторного представления слов существует множество моделей, из них сейчас наиболее популярны Word2Vec, GloVe, FastText, а также GPT и BERT. Все они используют нейронные сети или глубокое обучение с учетом контекста [4].



### **Заключение**

В статье рассмотрены основные методы векторизации текста. Очевидно, что метод прямого кодирования является наихудшим из представленных из-за минимальной информативности и большого размера занимаемой памяти. Метод векторного представления слов предоставляет более осмысленные результаты, однако для обучения моделей, основанных на этом методе, нужно много данных, что в случае с узконаправленным виртуальным ассистентом не всегда возможно предоставить. Методы мешка слов и TF-IDF в свою очередь занимают больше памяти и предоставляют менее информативные результаты, чем метод векторного представления слов, однако для виртуального помощника этого может быть достаточно.

Определить наиболее подходящий метод векторизации текста можно будет на следующем этапе разработки при подборе классификатора машинного обучения.

### **Библиографический список**

1. Раскатова М. В., Челышев Э. А. Векторизация текстов в задачах обработки естественного языка: история и развитие // IV Международная научно-практическая конференция «Современное программирование». 2022. С. 284–288.
2. Частикова В. А., Козачек К. В., Гуляй В. Г. Методы обработки естественного языка в решении задач обнаружения атак социальной инженерии // Вестник АГУ. 2021. № 4. С. 95–108.
3. Григорян Э. Г., Паршин М. Н. Методы NLP для предобработки текстовых данных и выделения признаков // Бизнес и общество. 2021. № 3. С. 1–8.
4. Prokhorov S., Safronov V. // International Conference on Artificial Intelligence: Applications and Innovations. 2019. P. 76–81.

УДК 004.058

**А. С. Половников**

студент кафедры информационных систем и технологий

**О. И. Красильникова** – кандидат технологических наук, доцент – научный руководитель**ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ ДЕЯТЕЛЬНОСТИ В ФИРМЕ ПО СБОРКЕ ПК**

Подбор комплектующих ПК (персонального компьютера) может оказаться непростой задачей, в данной теме есть множество нюансов: не все комплектующие будут работать вместе, нужно знать, какая материнская плата будет подходить для процессора, сможет ли видеокарта показывать заявленную производительность, какой мощности необходимо брать блок питания, какого объема нужна оперативная память. Даже при выборе жестких дисков есть свои тонкости, при этом для разных задач могут быть необходимыми совершенно разные комплектующие, поэтому предоставление услуги по подбору комплектующих и их сборке является востребованной не только для человека далекого от компьютерных технологий, но и для людей, тесно связанных с ними, такими как видеомонтажер, разработчики игр и приложений, 3D-дизайнер. Они нуждаются в компьютерах, имеющих акцент на различный набор характеристик.

Немаловажным ограничением при подборе ПК является количество средств – правильное распределение денежных ресурсов на комплектующие обеспечивает комфортную производительность в целевых задачах.

Из вышесказанного следует, что для пользователей, которые плохо знакомы с рынком комплектующих, обычный конфигуратор ПК будет непонятен. Вариантом решения проблемы по подбору конфигурации будет использование в виде входных данных критериев, которых пользователь может легко определить без знаний о внутреннем строении: цена и цель использования.

**Входные данные**

Критерии для подбора:

- цена;
- цель использования.

Цель использования разбита на следующие категории:

- работа с офисными приложениями;
- видеомонтаж;
- 2D- и 3D-моделирование;
- игры.

Если указано более одной цели, нужно назначить приоритет для каждой выбранной категории.

**Исходные данные**

1. Имеется таблица со всеми возможными вариантами сборок комплектующих, имеющих ограничения помимо совместимости.

- Жесткий диск является обязательным.
- Оперативная память может быть в 8 Гб × 1 шт, 8 Гб × 2 шт, 8 Гб × 4 шт, 16 Гб × 2 шт.

Минимальная мощность блока питания, который может быть установлен, определяется из суммы потребления всех комплектующих в пике +25% [1], верхнюю границу можно также указать. Подразумевается, что все рассматриваемые блоки питания имеют сертификат, соответствующий их мощности.

Видеокарта может быть не установлена, если есть iGPU (интегрированная видеокарта).

Твердотельный накопитель не обязателен.

Для процессоров, которые могут работать с разными типами оперативной памяти (на разных материнских платах), берется более современный тип.

2. Помимо данных о комплектующих таблица содержит информацию о производительности отдельных компонентов (видеокарты, процессора), соответствующей цели использования на основе бенчмарков (теста производительности) ПО и игр:

– для монтажа могут использоваться результаты обработки файла в видеоредакторах (DaVinci, Corona [2]);

– для моделирования – Blender [3], Компас;

– для игр – 3DMARK, бенчмарки игр [4].

3. Все данные производительности нормализованы методом MinMax, чтобы значения в равной мере влияли на результат.

4. У каждой записи берется среднее значение результатов бенчмарков, относящихся к одной цели использования.

5. Для каждой записи рассчитана суммарная стоимость.

6. Для каждой цели использования установлены минимальные комплектующие (видеокарта, процессор, объем оперативной памяти).

#### Работа алгоритма

1. Ограничения на минимальную производительность вместе с ограничением на цену применяются к таблице: исключаются записи, у которых цена больше указанной и результаты бенчмарков ниже, чем минимальные.

2. Проверяем наличие записей, если их нет, завершаем алгоритм, передаем информацию об отсутствии подходящих конфигураций.

3. Далее, если выбраны две и более цели использования, в соответствии с их приоритетом у каждой записи рассчитывается итоговое значение производительности по бенчмаркам:

$$v = \sum_{i=1}^n w_i x_i,$$

где  $n$  – это количество целей использования,  $x_i$  – значение производительности  $i$ -й цели использования,  $w_i$  – приоритет, причем

$$\sum_{i=1}^n w_i = 1.$$

4. Группируем и сортируем данные по полученной производительности (по убыванию) и цене (по возрастанию).

5. Возвращаем набор комплектующих и их суммарную стоимость из первой записи. Полученная сборка будет наиболее производительной в выбранных задачах.

Приложение реализовано на языке программирования JavaScript с использованием системы управления базами данных PostgreSQL. Для создания интерфейса была выбрана библиотека React, разработка сервера велась с помощью Node.js [5].

#### Заключение

В работе были приведены критерии, на основе которых можно совершить подбор комплектующих. Также был разобран алгоритм для получения готовой конфигурации ПК на основе данных критериев, тестирование дало положительный результат подбора. Данная работа является актуальной, так как существующие аналоги позволяют подбирать конфигурации лишь самостоятельно, когда разработанный алгоритм позволяет получить результат без знаний пользователя о комплектующих компьютера.

#### Библиографический список

1. PC Buyer's Guide: выбор блока питания. URL: <https://habr.com/ru/companies/ulmart/articles/365243/> (дата обращения: 04.05.2023).
2. Corona Benchmark. URL: <https://corona-renderer.com/benchmark> (дата обращения: 05.05.2023).

3. Benchmark. URL: <https://opendata.blender.org> (дата обращения: 05.05.2023).

4. Benchmark results from 3DMark, PCMark 10, and VRMark. URL: [https://www.3dmark.com/search?\\_ga=2.17023610.292534561.1684521842-636960552.1684521842](https://www.3dmark.com/search?_ga=2.17023610.292534561.1684521842-636960552.1684521842) (дата обращения: 05.05.2023).

5. Node.js Express & PostgreSQL: CRUD Rest APIs example with Sequelize. URL: <https://www.bezkoder.com/node-express-sequelize-postgresql/> (дата обращения: 06.05.2023).

УДК 004.514

**А. С. Радаева**

студент кафедры информационных систем и технологий

**О. И. Красильникова** – доцент, кандидат технических наук – научный руководитель**ПРИЕМЫ НЕЙРОДИЗАЙНА И ИХ ВЛИЯНИЕ НА УСПЕШНОСТЬ БИЗНЕСА**

Дизайн сайта – это первое, что видят посетители ресурса. По данным исследований около 50% пользователей доверяют компании сразу при первом посещении сайта, где понятно оформлена необходимая информация [1]. По данным других исследований, проведенных Top Design Firms в 2021 году, половина посетителей сайта покидают его из-за неудобного интерфейса или плохой функциональности [2].

Оформление сайта играет важную роль: оно может повысить узнаваемость бренда среди потенциальных клиентов, передать основные ценности компании и ее миссию. Правильная подача информации превращает простых посетителей в клиентов, нередко даже постоянных. Это значит, что дизайн стоит рассматривать как стратегический усилитель бизнеса, а не только как визуальную эстетику.

В оформлении наполнения сайта важна глубина проработки, которая включает в себя следующие аспекты:

- четкая структуризация компонентов,
- наличие особой концепции дизайна, уникальность иллюстраций,
- позиционирование бренда,
- гармоничное расположение элементов,
- общее удобство использования и взаимодействия с контентом.

Исследования показали, что учет этих факторов существенно повысит рейтинг страницы в поисковой выдаче [3]. На это ориентирована интенсивно развивающаяся в настоящий момент область дизайна – нейродизайн.

Нейродизайн объясняет, как наш мозг устроен и как создавать простые, вызывающие приятные эмоции и побуждающие к действию проекты. Поскольку люди в основном воспринимают все визуально, именно визуальные стимулы должны быть особенно интересны дизайнерам пользовательского опыта.

Основоположниками теории нейродизайна являются исследователи взаимодействия человека и компьютера – американские ученые Джейкоб Нильсен, Джилл Гердхарт-Пауэлс и Бен Шнайдермен.

В своих работах они сформировали пять основных положений.

1. Визуальное доминирование.

Среди всех типов сигналов многие люди выделяют визуальные стимулы как наиболее действенные. Иллюстрацией данного положения является широко известный эффект Гарри МакГурка [4].

2. Быстрая и медленная обработка.

Согласно исследованиям в области когнитивной психологии, проведенным Карлтонским университетом, человеку требуется около 50 миллисекунд для субъективной оценки сайта [5]. Причина такой скорости кроется в устройстве нашей лимбической системы. Идея нейродизайна в том, чтобы максимально качественно обратиться к ней.

3. Ощущение против восприятия.

Возможности мозга, в том числе его способность к восприятию информации, ограничены. Стоит учитывать этот факт и не добавлять слишком много визуальных стимулов.

4. Внимание.

Создание эффективного интерфейса заключается в использовании таких элементов, которые будут привлекать и удерживать внимание. Учет этого фактора привел к включению программно-анимируемых элементов в пользовательские интерфейсы.

5. Мотивация.

Необходимо побуждать людей к действию с помощью дизайна. В частности, это очень важно учитывать при разработке дизайна сайтов типа «лендинг пейдж», целью которых является сделать из простого посетителя сайта потенциального потребителя предлагаемой услуги.

Учет этих положений и результаты проведенных экспериментов [6] привели к появлению так называемых эвристик нейродизайна, которые состоят в следующем.

- Избавление от лишнего шума. Необходимо наполнять сайт только той информацией, которая необходима и относится к задаче.
- Использование лиц для привлечения внимания и эмоций. При просмотре какого-либо изображения человек обращает внимание в первую очередь на лица, а затем на остальные детали. Причем следующий объект внимания будет тот, на который устремлен взгляд на изображении. На рис.1 представлена тепловая карта, где красным цветом показаны объекты, на которые обращено больше всего внимания, а зеленым – меньше всего.



Рис. 1. Тепловая карта внимания

- Яркость как инструмент для передачи значимости. Яркие элементы всегда бросаются в глаза и забирают внимание. Желтый цвет привлечет больше внимания, чем бордовый. Важна именно яркость, а не оттенок цвета.
- Группировка. Необходимо группировать связанные по содержанию или функциональности данные, чтобы установить отношения между видимыми элементами. Различные виды группировки объектов представлены на рис. 2.



**Простота**  
Сгруппированные объекты воспринимаются как одна фигура. Склонность упрощать.



**Близость**  
Объекты воспринимаются как группа, если они расположены близко друг к другу.



**Сходство**  
Объекты воспринимаются как группа, если они похожи.



**Непрерывность**  
Когда несколько объектов пересекаются, люди склонны воспринимать их как единый цельный объект.



**Завершение**  
Люди воспринимают формы как завершенные, даже если некоторые их части отсутствуют.



**Симметрия**  
Люди склонны воспринимать объекты как симметричные формы, образованные вокруг условных центров.

Рис. 2. Способы группировки объектов

– Движение. Движение всегда привлекает внимание пользователей прежде всего, но его чрезмерное использование может приглушить этот эффект.

– Обратная связь. Необходимо обеспечить краткую и своевременную обратную связь, чтобы у пользователя была мотивация продолжать работу

Научным исследователям компании McKinsey удалось выявить закономерность между качеством дизайна сайта и успешностью бизнеса. Корреляцию нельзя считать прямой, но компании с отзвучившим дизайном статистически чаще имеют хорошие финансовые показатели [7].

Эксперты из McKinsey вычислили индекс MDI (McKinsey Design Index), отражающий качество дизайна и его связь с бизнес-показателями. Авторы исследования пришли к выводу, что между высоким MDI и отличными финансовыми результатами есть заметная корреляция. Учеными были изучены несколько отраслей: медицинские технологии, потребительские товары, банковские услуги, а также сфера информационных технологий. Компании с индексом выше среднего увеличивают выручку на 15% быстрее. Кроме того, такие предприятия увеличивают совокупный доход своих акционеров на 27% быстрее конкурентов отрасли.

Полученные результаты позволяют сделать вывод о том, что данный индекс достоверен для любых компаний, независимо от вида работы: офлайн или онлайн.

В заключение подчеркнем важность учета того, что дизайн является одним из наиболее важных факторов успеха бизнеса. Он может как привлечь внимание посетителей и повысить коммерческий успех, так и препятствовать ему, отталкивая потенциальных клиентов. Это справедливо для всех сфер: услуги, реальные товары или digital-продукты.

В настоящее время понятие нейродизайна мало известно специалистам в области разработки пользовательских интерфейсов, однако, его положения чрезвычайно важны для каждого frontend-разработчика, так как все, что касается взаимодействия с пользователем, сводится к нейродизайну.

#### Библиографический список

1. SiteElite: Как дизайн контента сайта влияет на развитие бизнеса. URL: <https://st-It.ru/blog/kak-dizajn-kontenta-sajta-vliyaet-na-razvitie-biznesa.html> (дата обращения: 15.03.2023).
2. MartechSeries: Half of consumers consider a Company's Website Design crucial to their opinion of that brand. URL: <https://martechseries.com/video/brand-safety/half-of-consumers-consider-a-companys-website-design-crucial-to-their-opinion-of-that-brand/> (дата обращения: 12.03.2023).
3. VENNGAGE: 16 Visual Content Marketing Statistics to Know for 2022. URL: <https://venngage.com/blog/visual-content-marketing-statistics/> (дата обращения: 05.03.2023).
4. Hearing lips and seeing voices / Harry McGurk, John MacDonald // Nature. 1976. Vol. 264. № 5588. P. 746–748.
5. Uxdesign: Usability heuristic frameworks: which one is right for you. URL: <https://uxdesign.cc/usability-heuristic-frameworks-which-one-is-right-for-you-1962387b7cc> (дата обращения: 18.03.2023).
6. Nielsen Norman Group: Weekly User Testing. URL: <https://www.nngroup.com/articles/weekly-user-testing-tivo/> (дата обращения: 20.03.2023).
7. Sostav: McKinsey: как дизайн влияет на бизнес-показатели. URL: <https://www.sostav.ru/publication/mckinsey-kak-dizajn-vliyaet-na-biznes-pokazateli-34471.html> (дата обращения: 18.03.2023).

УДК 519.654

**М. А. Савельев**

студент кафедры информационных систем и технологий

**А. П. Шелета** – доктор технических наук, профессор – научный руководитель**ИСПОЛЬЗОВАНИЕ МЕТОДОВ НАИМЕНЬШИХ КВАДРАТОВ И НАИМЕНЬШИХ МОДУЛЕЙ  
В ЭКОНОМИКЕ**

Метод наименьших квадратов (МНК) широко применяется в экономике для оценки параметров линейных моделей, которые описывают зависимости между экономическими переменными. Например, МНК может использоваться для оценки коэффициентов регрессии, которые описывают связь между объемом производства и затратами на производство, или для оценки эластичности спроса на товары в зависимости от цены. МНК также может использоваться для оценки параметров в более сложных моделях, таких как множественная регрессия или временные ряды. МНК является одним из основных инструментов для проведения эмпирических и статистических исследований, а также для прогнозирования экономических показателей [1]. Однако, важно помнить, что МНК не является универсальным методом и может быть менее эффективным в случае нарушения предпосылок модели или наличия выбросов в данных. В таких случаях может быть более подходящим использование альтернативных методов, как метод наименьших модулей (МНМ) или робастные методы оценки.

Метод наименьших модулей (МНМ) также широко применяется в экономике для оценки параметров моделей. В отличие от МНК, который предполагает нормальное распределение ошибок, метод МНМ не делает никаких предположений о распределении ошибок и позволяет более гибко моделировать зависимости между переменными. МНМ может использоваться для оценки параметров в различных моделях, таких как линейная и нелинейная регрессия, модели временных рядов и другие. Он также может быть полезен в случаях, когда данные содержат выбросы или нарушения предпосылок модели.

В экономике МНМ используется для проведения эмпирических и статистических исследований и для прогнозирования экономических показателей. Например, МНМ может быть использован для оценки параметров моделей спроса и предложения на рынке, моделей инфляции и безработицы, а также для анализа влияния различных экономических факторов на развитие отраслей и регионов. Одним из преимуществ МНМ является его способность обрабатывать данные с различными распределениями ошибок, что делает его более универсальным методом в сравнении с МНК. Однако, как и любой другой метод оценки параметров моделей, МНМ также имеет свои ограничения и требует правильного выбора модели и предварительной обработки данных [2].

В основе обоих методов лежит построение линии тренда. В методах МНК и МНМ нельзя линию тренда проводить через точки зависимости  $X$  и  $Y$ , но для большей корреляции она должна быть максимально приближена к этим точкам. Эффективность методов (МНК и МНМ) определяется алгоритмом минимализации функционалов [3].

Так для МНК будет справедлива формула:

$$\sum_{i=1}^N (y_i - \bar{y})^2 = \min;$$

для МНМ:

$$\sum_{i=1}^N |y_i - \bar{y}| = \min,$$

где  $y_i$  – фактические ординаты поля;  $\bar{y}$  – это среднее значение ординаты с абсциссой.



На основе указанных методов были проведены исследования статистических данных, взятых с официального сайта Управления Федеральной службы государственной статистики по г. Санкт-Петербургу и Ленинградской области [5].

Линии тренда МНК и МНМ были построены с помощью пакета прикладных программ MATLAB [4]. Данные для исследования представлены в табл. 1: год (X) и сумма прибыли, млрд руб. (Y).

Таблица 1

**Финансовое состояние организаций Ленинградской области  
(без субъектов малого предпринимательства) [5]**

Статистический показатель	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
Сумма прибыли, млрд. руб.	76,4	83,9	103,9	131,7	123,9	168,3	228,1	187,8	215,8	205,7	225,9	351,3

По заданным данным были построены линии тренда МНК (рис. 1), МНМ (рис. 2) и совмещение двух линий (рис. 3).

Построение производилось по формуле:

$$y = ax^2 + bx + c.$$

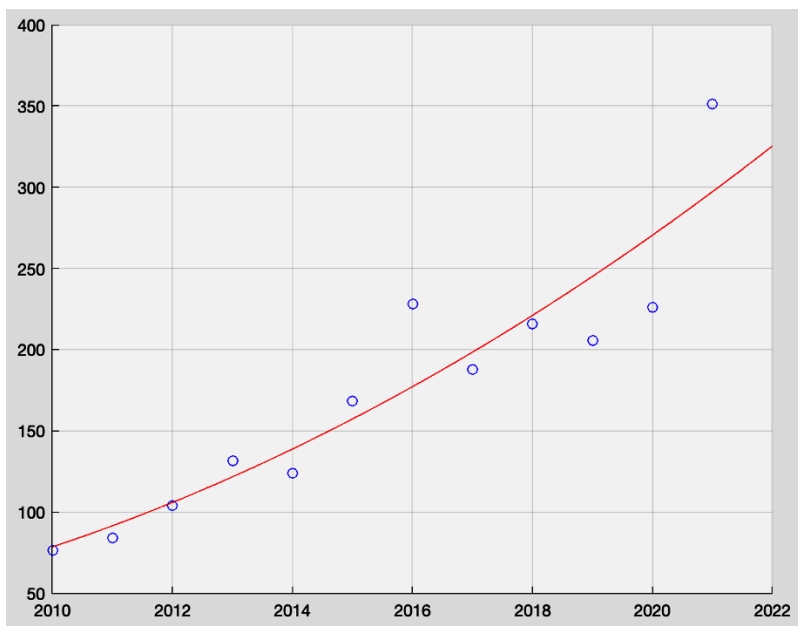


Рис. 1. Линия тренда финансового состояния организаций Ленинградской области за 2010–2021 гг. методом МНК

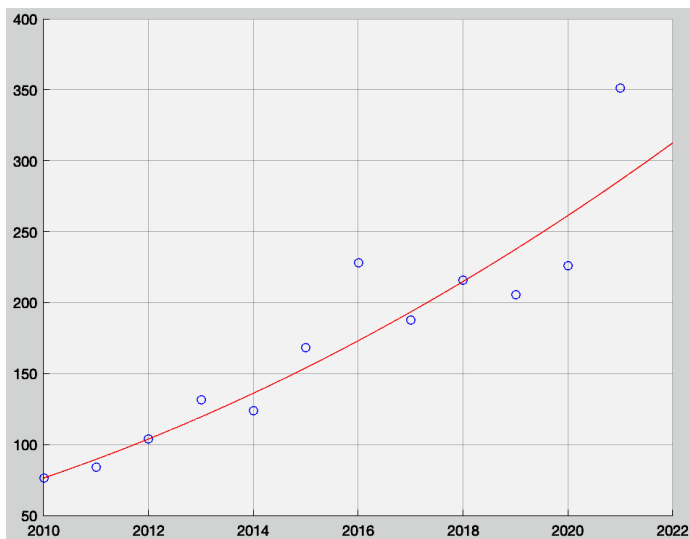


Рис. 2. Линия тренда финансового состояния организаций Ленинградской области за 2010–2021 гг. методом МНМ

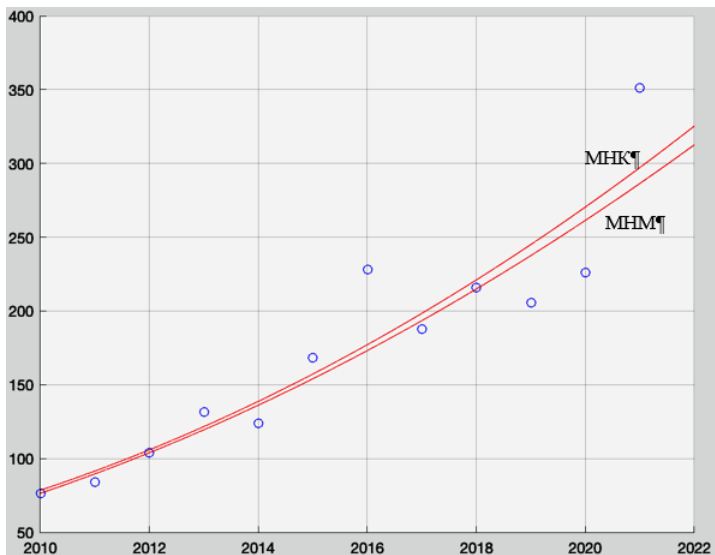


Рис. 3. Совмещение линий тренда методом МНК и МНМ

Как видно из рисунков построенные линии тренда МНК и МНМ действительно очень похожи. Однако линия тренда, построенная методом МНК, в отличие от метода МНМ сильнее реагирует на выбросы.

### Вывод

Рассмотренные методы, примененные к исследованию статистических данных, показали как преимущества, так и недостатки, а именно:

К достоинствам метода МНК можно отнести:

– метод является широко используемым и изученным, используется для оценки параметров линейной регрессии;

– имеет аналитическое решение, что делает его быстрым и эффективным;

– обладает хорошими статистическими свойствами, такими как эффективность и состоятельность.

К недостаткам метода МНК относятся:

– чувствительность к выбросам в данных, что может привести к неправильным оценкам параметров;

– МНК может быть неэффективным, если данные не являются линейно зависимыми;

– данный метод не учитывает возможные ошибки измерений, которые могут привести к неточным оценкам параметров.

Преимуществами метода МММ являются:

– меньшая чувствительность к выбросам в данных, по сравнению с методом МНК, что делает его более устойчивым к неправильным оценкам параметров;

– МММ может быть более эффективным, если данные не являются линейно зависимыми;

– метод учитывает возможные ошибки измерений, что делает его более точным методом оценки параметров;

– МММ может быть более устойчивым к нарушению нормальности распределения ошибок, чем МНК.

К недостаткам метода МММ можно отнести:

– метод может быть менее эффективным, если данные линейно зависимы;

– МММ может привести к более широким доверительным интервалам, чем МНК, что может уменьшить точность оценок параметров;

– метод МММ может быть менее удобен для вычислений, чем МНК, особенно для больших наборов данных;

– МММ может быть менее распространенным, чем МНК, что может затруднить поиск соответствующей литературы и программного обеспечения для его применения.

### Библиографический список

1. Кремер Н. Ш., Путко Б. А. Эконометрика. М.: Юнити, 2002. 311 с.
2. Тырсин А. Н., Максимов К. Е. Оценивание линейных регрессионных уравнений с помощью метода наименьших модулей // Заводская лаборатория. Диагностика материалов. 2012. № V. С. 55–71.
3. Волков Н. Г., Кондрашов В. С., Мороз З. Д., Сравнение алгоритмов минимализации при реализации метода наименьших модулей. URL: [https://inis.iaea.org/collection/NCLCollectionStore/\\_Public/19/073/19073549.pdf](https://inis.iaea.org/collection/NCLCollectionStore/_Public/19/073/19073549.pdf).
4. MATLAB – MathWorks. URL: <https://www.mathworks.com/products/matlab.html> (дата обращения: 07.05.2023).
5. Управления Федеральной службы государственной статистики по г. Санкт-Петербургу и Ленинградской области. URL: <https://78.rosstat.gov.ru/> (дата обращения: 15.05.2023).

УДК 004.932.72'1

**И. Д. Свежин**

студент кафедры информационных систем и технологий

**И. А. Воронаев** – ассистент – научный руководитель

## ИССЛЕДОВАНИЕ АЛГОРИТМОВ СЛЕЖЕНИЯ ЗА ОБЪЕКТОМ НА ОСНОВЕ АДАПТИВНЫХ КОРРЕЛЯЦИОННЫХ ФИЛЬТРОВ

### Введение

Слежение за объектами на видео является одной из важнейших подзадач анализа видео, который используется во множестве различных сфер. Существует множество подходов в реализации так называемого «трекинга», и самый лучший из них на текущий момент – использование нейронных сетей. Современные технологии позволяют проектировать модели с огромным числом промежуточных слоев, которые могут выдавать очень точный результат даже на самых сложных примерах, однако у таких сетей есть ряд существенных недостатков.

Во-первых, даже на современном оборудовании сложно добиться быстрого отклика от обученной нейросети, что полностью исключает возможность их использования в системах, работающих в режиме реального времени.

Во-вторых, такой подход просто невозможно использовать в случаях, когда нет доступа к дорогим и мощным вычислительным системам, поэтому были разработаны специальные алгоритмы, позволяющие решать те же задачи, ограничиваясь лишь линейной алгеброй и преобразованиями Фурье.

### Адаптивные корреляционные фильтры

В основе работы корреляционных алгоритмов лежит операция свертки матриц. Данный прием используется и во многих других фильтрах, напр. для размытия изображения или увеличения резкости [1], однако, в данном случае свертка нужна для определения степени корреляции фильтра и соответствующей ему области изображения. Согласно Теореме Свертки, в целях сокращения количества вычислений для расчета отклика корреляционного фильтра можно использовать алгоритм Быстрых Преобразований Фурье (далее – БПФ). Имея исходное изображение  $F$  и матрицу частот фильтра  $W$ , полученную из выделенной области, содержащей интересующий объект, можно вычислить кросс-корреляцию следующим образом:

$$\begin{aligned}\hat{W}(\omega, \nu) &= \mathcal{F}(W(x, y)), \\ \hat{F}(\omega, \nu) &= \mathcal{F}(F(x, y)), \\ \widehat{G_{conv}}(\omega, \nu) &= \mathcal{F}(G_{conv}(x, y)), \\ G_{conv} &= F \otimes W \rightarrow \widehat{G_{conv}} = \hat{F} \circledast \hat{W}^*.\end{aligned}$$

Из итоговой формулы выразим матрицу частот фильтра:

$$\hat{W}^* = \frac{\widehat{G_{conv}}}{\hat{F}}.$$

Далее, применив к полученной матрице обратное преобразование Фурье, получим значения фильтра в пространственной области. Самые затратные операции при таком подходе – расчет прямого и обратного БПФ, при этом сложность алгоритма будет равна  $O(P \log P)$ , где  $P=m*n$  (размеры выделенного «окна»).

### Average of Synthetic Exact Filters (ASEF)

Исследуемая видеозапись представляет собой последовательность кадров, соответственно, для работы нативного алгоритма необходимо отдельно рассчитывать значения фильтра для каждого

изображения. Данное условие существенно увеличивает время обработки видео, поэтому, был разработан алгоритм Average of Synthetic Exact Filters (ASEF) [2], идея которого – усреднение набора «идеальных» фильтров в один. «Идеальным» фильтром считается тот, который дает максимальный отклик (нулевую ошибку) для конкретного изображения. Имея одно обучающее изображение  $F_k$  и одно выходное изображение  $G_k$ , получить для них идеальный фильтр можно, решив следующее уравнение:

$$\widehat{W}_k^* = \frac{\widehat{G_{conv_k}}}{\widehat{F}_k} = \frac{\widehat{G_{conv_k}} \times \widehat{F}_k^*}{\widehat{F}_k \times \widehat{F}_k^*}.$$

Такой фильтр, полученный с помощью лишь одного исходного изображения, становится (выражаясь терминами машинного обучения) переобученным, в следствие чего на новом изображении он не выдаст никакого результата. Поэтому прибегают к идее «усреднения» весов множества идеальных фильтров, «обученных» на большем наборе изображений. Формула для усредненного фильтра выглядит следующим образом:

$$\widehat{W}^* = \frac{1}{N} \sum_{k=1}^N \widehat{W}_k^* = \frac{1}{N} \sum_{k=1}^N \frac{\widehat{G}_k}{\widehat{F}_k} = \frac{1}{N} \sum_{k=1}^N \frac{\widehat{G_{conv_k}} \times \widehat{F}_k^*}{\widehat{F}_k \times \widehat{F}_k^*}.$$

### Minimum Output Sum of Squared Error (MOSSE)

Фильтры типа MOSSE [3] являются модификацией фильтров ASEF, поэтому они так же основаны на БПФ и операциях свертки, однако, позволяют вычислить более точные значения фильтра на основе меньшего количества обучающих изображений. Инициализация фильтра на первом изображении, как и в случае с ASEF, вычисляется при помощи поэлементного деления корреляционной матрицы и матрицы самого изображения, полученных после применения БПФ. Идея фильтра MOSSE заключается в том, чтобы найти «средний» фильтр, удовлетворяющий всем изображениям с определенной погрешностью.

$$\widehat{W}^* : \min_{\widehat{W}} \sum_{k=1}^N \|\widehat{F}_k \circ \widehat{W}^* - \widehat{G}_k\|_2.$$

Для нахождения минимума необходимо взять производную от этого выражения, перейдя к непосредственно самим элементам фильтра (где  $\omega$  и  $\nu$  теперь индексы элементов матриц, а  $i$  – сам номер изображения), и приравнять все к нулю:

$$0 = \frac{\delta}{\delta \widehat{W}_{\omega\nu}^*} \sum_i \|\widehat{F}_{i\omega\nu} \widehat{W}_{\omega\nu}^* - \widehat{G}_{i\omega\nu}\|_2,$$

$$\widehat{W}_{\omega\nu}^* = \frac{\sum_i \widehat{F}_{i\omega\nu} \widehat{G}_{i\omega\nu}}{\sum_i \widehat{F}_{i\omega\nu} \widehat{F}_{i\omega\nu}^*}.$$

Далее необходимо обратно перейти к индексированию через номер кадра:

$$\widehat{W}^* = \frac{\sum_{k=1}^N \widehat{F}_k \circ \widehat{G}_k}{\sum_{k=1}^N \widehat{F}_k \circ \widehat{F}_k^*},$$

$$\widehat{W}^* = \frac{\sum_{k=1}^N \widehat{F}_k^* \circ \widehat{G}_k}{\sum_{k=1}^N \widehat{F}_k \circ \widehat{F}_k^*}.$$

Получив один усредненный фильтр, для вычисления координат объекта на новом изображении необходимо найти его текущий отклик в частотной области, посредством поэлементного умножения фильтра на саму матрицу изображения:

$$\widehat{G}_{conv} = \widehat{F} \odot \widehat{W}^*$$

На финальном этапе остается перейти от комплексных чисел к реальным при помощи обратного преобразования Фурье, и, найдя максимум в получившейся матрице  $G_{conv}$ , получим новое положение объекта.

### Оптимизация

В описанных выше подходах формулы отвечают за расчет фильтра при его инициализации на исходном кадре, при этом для остальных кадров он не изменяется. «Обучение» фильтра происходит за счет разбиения первого в последовательности кадра на множество окон исходного размера, и в зависимости от значений при разных положениях окна подстраиваются значения фильтра. Тем не менее, когда объект претерпевает серьезные изменения в форме или размере, первоначальный фильтр перестает работать, поэтому необходимо его подстраивать с каждым следующим кадром.

Для фильтра ASEF обновление будет рассчитываться следующим образом (в обоих вариантах коэффициент  $\eta$  означает скорость обучения, и подбирается эмпирически):

$$\widehat{W}_i^* = \eta \frac{\widehat{G}_{conv_i} \times \widehat{F}_i^*}{\widehat{F}_i \times \widehat{F}_i^*} + (1 - \eta) \widehat{W}_{i-1}^*$$

В случае с MOSSE необходимо производить отдельный расчет числителя и знаменателя, поэтому заменим их на условные  $A_i$  и  $B_i$ :

$$\widehat{W}_i^* = \frac{A_i}{B_i}, \text{ где } A_i = \eta \widehat{G}_{conv_i} \odot \widehat{F}_i^* + (1 - \eta) A_{i-1}, B_i = \eta \widehat{F}_i \odot \widehat{F}_i^* + (1 - \eta) B_{i-1}.$$

При использовании такой схемы фильтр очень быстро адаптируется к различным деформациям объекта, так как в большей степени зависит от значений на предыдущем кадре, нежели чем от обучающего набора изображений.

### Оценка качества детекции

Для оценки результатов детекции создатели MOSSE предлагают использовать специальную метрику, именуемую PSR – Peak to Sidelobe Ratio [3]. Вычисляется она следующим образом:

Пусть  $g_{max}$  – максимальное значение выходной матрицы  $G_{conv}$ , а  $sl$  – все остальные значения матрицы, за исключением квадрата размером  $11 \times 11$  вокруг  $g_{max}$ . Вычислив среднее линейное ( $\mu_{sl}$ ) и среднеквадратичное ( $\sigma_{sl}$ ) отклонения значений от  $g_{max}$  по текущей области, можно вычислить PSR:

$$PSR = \frac{g_{max} - \mu_{sl}}{\sigma_{sl}}$$

Принятие решения о том, была ли детекция удачной, основывается на величине данного показателя. Обычно значения лежат в пределах от 20 до 60, если полученный PSR получился выше определенного порога – значит детекцию можно считать удачной.

### Полученные результаты тестирования

Описанный выше алгоритм MOSSE имеет множество реализаций на разных языках программирования, в данной статье результаты тестирования представлены для утилиты собственной разработки, написанной на языке Python. Утилита представляет собой фильтр для фреймворка GStreamer, работающий по алгоритму MOSSE. В качестве исходных данных было взято видео в формате 4K

(здесь и далее – стандарт видео с разрешением 3840x2160 пикселей [4]), с частотой кадров в 30 кадр./сек. и длительностью ровно 30, что соответствует ровно 900 кадрам. Для оценки скорости работы используется метрика Frames Per Second (количество кадров в секунду, далее – FPS), вычисляемая следующим образом:

$$FPS = \frac{1}{t_f},$$

где  $t_f$  – время одного кадра.

С учетом подхода к реализации алгоритма, в коде программы можно измерить время обработки одного кадра (вычисления предполагаемой области нахождения объекта), однако данный показатель измеряется в микросекундах, что неудобно при отображении данных на графиках. Используемый показатель FPS – это средняя мера, показывающая, сколько бы кадров отобразилось за одну секунду, если бы каждый кадр выводился за время  $t_f$ .

Обработывая одно и то же видео, можно сравнить скорость обработки кадров для областей определения объекта (далее – ROI, Region Of Interest) разного размера. Для 4K видео область в 500x500px. (здесь и далее – “px.” – сокращение от англ. “pixels”, т. е. «пикселей») соответствует объекту среднего размера, в исследовании также были взяты области вдвое больше и вдвое меньше указанного. В целях более наглядного представления результатов данные были подвергнуты несильному сглаживанию. График измерений FPS представлен на рис. 1.

Отслеживание объекта на видео 4K, 30FPS, длительность 30 сек

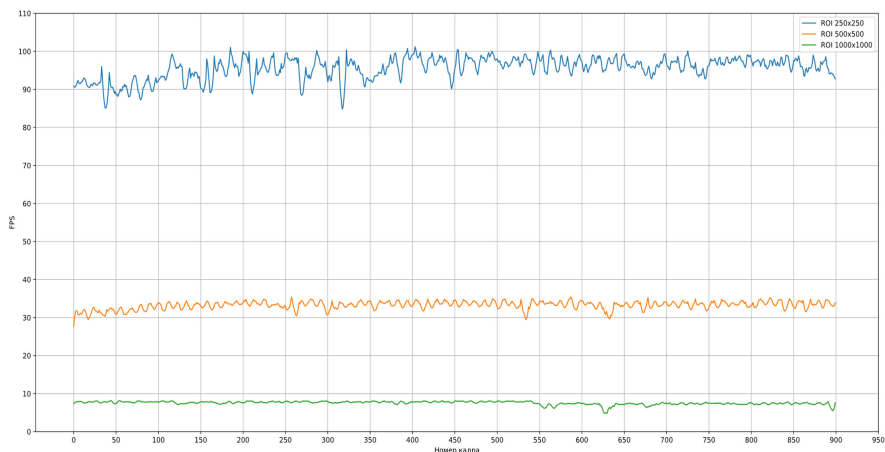


Рис. 1. График зависимости FPS от размера области определения объекта

Качество детекции объекта можно определить по метрике PSR, описанной выше. Необходимо учитывать, что с увеличением ROI увеличивается и значение PSR, однако главное в этом показателе – минимальный порог. График измерений PSR представлен на рис. 2.

Исходя из полученных данных, алгоритм может уверенно обрабатывать потоковое 4K видео со скоростью до 30 кадр./сек. с учетом ROI размером в 500x500px. Качество детекции при таком режиме можно охарактеризовать как хорошее. Так как на графике PSR хоть и заметны очень большие пики и впадины, однако, уровень не опускается ниже 40, когда пороговый уровень определения объекта – 8.

Другой подход к измерению скорости работы алгоритма – снижение разрешения исходного видео.

Отслеживание объекта на видео 4K, 30FPS, длительность 30 сек

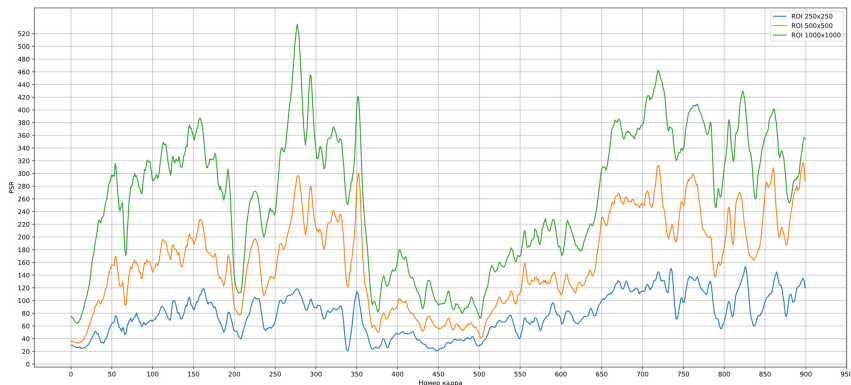


Рис. 2. График зависимости PSR от размера области определения объекта

Для этого видеозапись при помощи программы Adobe Premiere Pro была подвергнута процедуре «даунскейлинга» – снижение размеров кадра с формата 4K до Full HD (иначе – FHD, стандарт разрешения видео, равный 1920×1080 пикселям [4]) и HD (здесь и далее – стандарт разрешения видео, равный 1280×720 пикселям [4]), с сохранением частоты кадров и битрейта. При этом, целевая область определения объекта тоже должна измениться, причем пропорционально:

– 4K: формат видео – 3840×2160px, размер ROI – 500×500px

– Full HD: формат видео – 1920×1080px, размер ROI – 250×250px, так как стороны изображений 4K и FHD соотносятся как 1 к 2.

– HD: формат видео – 1280×720px, размер ROI – 167×167px, так как стороны изображений 4K и HD соотносятся как 1 к 3.

График зависимости изменения частоты кадров при снижении разрешения представлен на рис. 3.

Отслеживание объекта на видео, 30FPS, длительность 30 сек

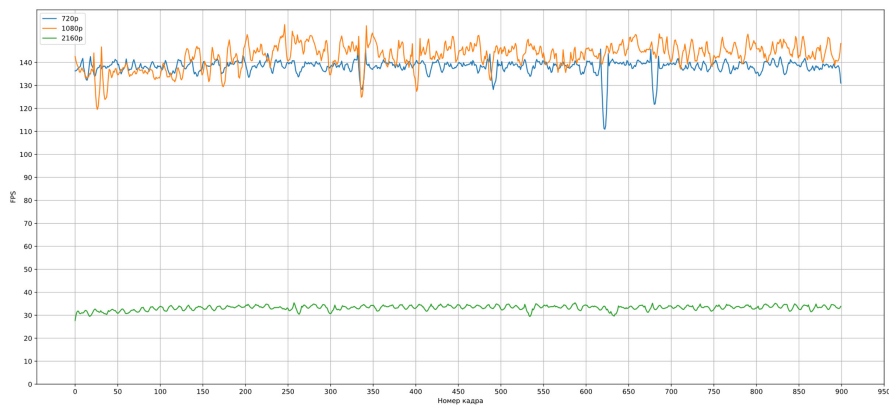


Рис. 3. График зависимости FPS от разрешения



На графике заметна сильная разница в более чем 100 кадр/сек. между обработкой 4K и FHD, однако, стоит учитывать, что средняя частота кадров для FHD и HD видео в 140 кадр./сек избыточна почти для всех стандартов передачи и воспроизведения видео. Таким образом, алгоритм MOSSE позволяет обрабатывать видео с разрешением вплоть до крайне требовательного стандарта 4K с сохранением высокого уровня детекции объектов.

### **Заключение**

В данной статье были рассмотрены два алгоритма слежения за объектом, основанные на адаптивных корреляционных фильтрах. Заложенные в них подходы к анализу изображений позволяют довольно точно определить расположение интересующего объекта без необходимости использовать мощные вычислительные системы. Скорость работы и точность подобных алгоритмов в некоторых ситуациях может быть до 20 раз больше в сравнении с современными многослойными нейронными сетями, которые наиболее часто используются в сфере анализа изображений и видео.

### **Библиографический список**

1. Матричные фильтры обработки изображений. URL: <https://habr.com/ru/articles/142818/> (дата обращения: 16.04.2023).
2. David S. Bolme, Bruce A. Draper, J. Ross Beveridge. Average of Synthetic Exact Filters // IEEE Conference on Computer Vision and Pattern Recognition, 2009. С. 2–4.
3. David S. Bolme, J. Ross Beveridge, Bruce A. Draper, Yui Man Lui. Visual Object Tracking using Adaptive Correlation Filters // IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010. С. 3–5.
4. Karen Goulekas. Visual Effects in a Digital World: A Comprehensive Glossary of over 7000 Visual Effects Terms. Elsevier, 2001. 600 p.

УДК 004.652

**В. Н. Трифонов**

студент кафедры информационных систем и технологий

**Н. В. Богословская** – доцент, кандидат технических наук – научный руководитель**СТРУКТУРА ПРОСОПОГРАФИЧЕСКИХ БАЗ ДАННЫХ И ОБЛАСТИ ИХ ПРИМЕНЕНИЯ**

С появлением необходимых вычислительных мощностей и технологий обработки больших объемов данных исследования в области изучения биографических данных исторически значимых персон и событий получили новые возможности.

Целью данной статьи является анализ современных научных работ в этой области [1, 2, 3], и представляется интересным обсудить следующие вопросы: сущность просопографии, сферы ее прикладного применения, а также особенности проектирования просопографических баз данных.

Термин «просопография» состоит из двух древнегреческих корней «prosopon» и «graphia». «Prosopon» переводиться как «лицо, личность, внешние черты», «graphia» – «пишу, описываю». Отчего просопографию можно воспринимать, как науку, описывающую наиболее характерные черты и атрибуты группы персон.

Разинков С. Л. (доцент кафедры документоведения РГПУ г. Екатеринбург) под термином «просопография» подразумевает «жанр исследований, предполагающий изучение массовых источников с целью создания на основе статистического анализа их информации динамических «коллективных биографий» определенных социальных группы, страт» [1].

Общедоступными примерами прикладного приложения просопографических баз данных являются сайты генеалогических деревьев. Подобные проекты содержат информацию о предках и потомках, а также о родственных связях между различными членами семьи. Пользователи могут использовать эту информацию для построения своего родословного дерева и изучения своей генеалогии.

В процессе исследования были выявлены наиболее популярные проекты в этой области:

– MyHeritage ([myheritage.com](http://myheritage.com)) – это сайт, предоставляющий доступ к более чем 12 миллиону генеалогических записей. Он включает в себя информацию о родственных связях, иммиграции, рождении, браке, смерти и многое другое. Сайт также предоставляет возможность строить родословные деревья, использовать функцию сопоставления ДНК и связываться с другими исследователями.

– FamilySearch ([familysearch.org](http://familysearch.org)) – это бесплатный сайт, предоставляющий доступ к миллиону генеалогических записей.

– Geni ([geni.com](http://geni.com)) – это сайт, предоставляющий возможность строить родословные деревья и связываться с другими исследователями. Сайт также позволяет добавлять фотографии и документы, связанные с каждым членом семьи. Сайт доступен как в бесплатной, так и в платной версии.

– WikiTree ([wikitree.com](http://wikitree.com)) – помимо базового функционала по редактированию родословного древа сайт также позволяет добавлять фотографии и документы, связанные с каждым членом семьи. Одним из преимуществ сайта является то, что он предоставляет бесплатный доступ к своим данным и не требует подписки на платную версию.

Приведем примеры общедоступных зарубежных и отечественных просопографических баз данных: «Информацией о жителях англосаксонской Британии» (Prosopography of Anglo-Saxon England, PASE) [4], «Византийская просопография» (Prosopography of the Byzantine World, PBW) [5], «Уральское монашество синодального периода» (по мужским и женским монастырям Среднего Урала) [6], «Офицеры флота в выборных органах революционной власти в 1917–1918 гг.» [7], Личные дела рабочих нефтепромышленной фирмы «Товарищество братьев Нобель» (изучения рынка рабочей силы в нефтяной промышленности Баку) [8].

Таким образом, на представленных примерах можно увидеть разнообразие задач прикладного применения цифровой просопографии. Однако промежуточным этапом любого просопографического исследования является проектирование баз данных с учетом особенностей их структур и описания концептуальной модели.

Согласно классификации Дж. Аока (профессора кафедры информатики парижской высшей инженерной школы CNAM), для ведения каталога просопографических данных различают реляционные, полуструктурированные и сетевые модели [3].

Так, на реляционной модели основывались первые просопографические базы данных COEL [3], PASE [3], а недавняя работа профессора кафедры китайской истории Гарвардского университета Бола П. К. предлагает использовать географические информационные системы, основанные на реляционных базах данных, для обнаружения, например, пространственных закономерностей [9]. Таким образом, реляционная модель позволяет выполнять эффективные поисковые запросы по ограниченному количеству таблиц. На рис. 1 представлен пример концептуальной модели реляционной базы данных.

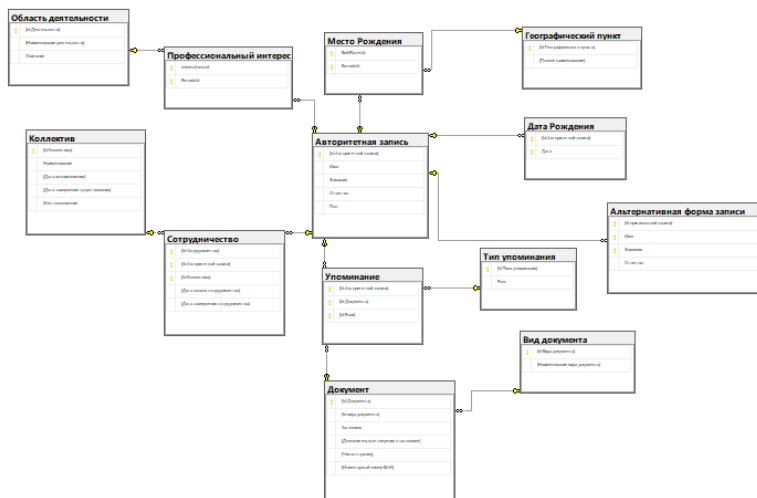


Рис. 1. Пример концептуальной модели реляционной базы данных

Отличительной особенностью полуструктурированной модели является возможность ограничить операции соединения за счет использования древовидной структуры. Таким образом, модель допускает многозначные атрибуты и интеграцию (полу)структурированных объектов в реляционную схему. Отчего данный вид моделей адаптирован к просопографическим базам данных, где элемент «человек» может состоять из элементов «место работы», «образование» и т. д., которые сами являются структурированными элементами. Примерами просопографических баз, основанных на таком представлении данных, являются проекты STUDIUM PARISIENSE [3] и PROCO [3]. Полуструктурированная модель позволяет отразить связи между людьми/объектами/местами/фактами, но запрос более сложных связей становится затруднительным. На рис. 2 представлен пример концептуальной модели полуструктурированной базы данных.

По этой причине широкое применение получила сетевая модель. Например, тип представления «социальные сети» встречается в работах [10, 11, 12]. Такой подход позволяет при поиске данных обнаруживать связи между людьми/объектами/местами/фактами или повторяющиеся закономерности. Недавние разработки также полагаются на сетевое представление данных. Например, проект, руководителем которого выступает профессор кафедры компьютерных наук университета Аалто Йоуни Туоминен, представляет собой событийную, личностно-ролевую модель для представления деятельности, в которой человек участвовал в течение своей жизни [3]. Пример концептуальной сетевой модели представлен на рис. 3.

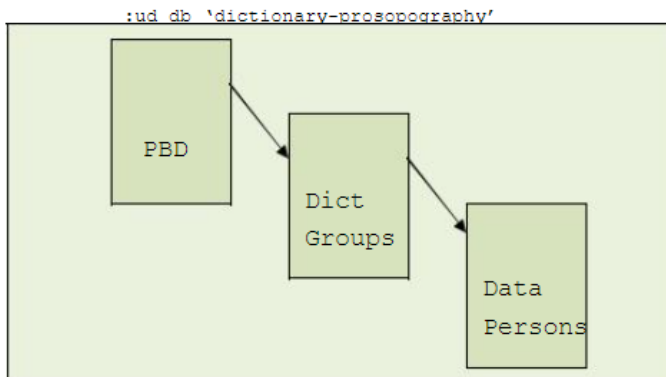


Рис. 2. Пример концептуальной модели полуструктурированной базы данных

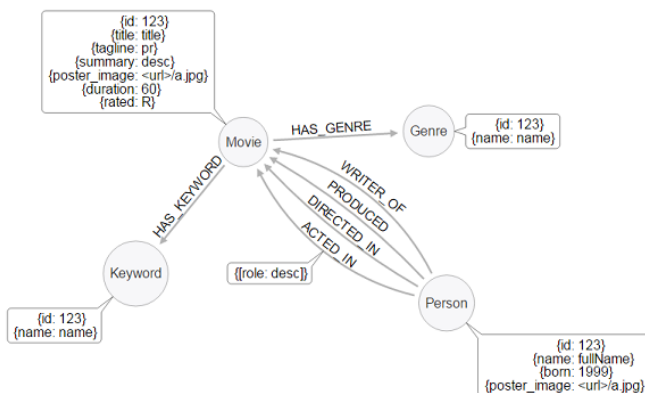


Рис. 3. Пример концептуальной модели сетевой базы данных

Центральными в любом просопографическом проекте является концепция времени – атрибут времени в базах данных составляет источник неопределенности, поэтому представление временных данных является также одной из задач, возникающих на этапе даталогического проектирования.

Профессор кафедры компьютерных наук Ольборгского университета Кр. Дженсен выделяет два типа представления временных данных: «мгновенные» и «интервальные» [3].

К системам, основанным на временных интервалах, можно отнести следующие проекты: GIT-Схема [3], ACM [3].

Проект GENTECH [3] поддерживает создание конфликтующих временных отношений, выражающих разные точки зрения, что относит ее к базам данных с «мгновенным» типом представления времени.

В то же время модель времени в проекте AROM-ST [3] предлагает использование несколько типов атрибута времени, включая мгновенные, интервальные, мультимгновенные и мультиинтервальные типы, так атрибут времени в данной модели объединяет дискретное время (дата), непрерывное время (интервал) и их состав (несколько потенциальных дат, или несколько возможных интервалов,

или несколько кумулятивных интервалов, например, «он присутствовал с 1492 по 1500 год, затем с 1503 по 1508 год»).

Также в своем проекте заведующая лабораторией кафедры прикладных наук о жизни Киотского университета Джун Огава ввела новую модель описания времени, согласно которой человек описывается не как отдельная сущность, а как совокупность контекстуальных сущностей, каждая из которых соответствует временному аспекту личности [3].

В данной статье было рассмотрено понятие цифровой просопографии, круг решаемых ею задач и областей применения, особенности проектирования просопографических даталогических моделей.

Просопография позволяет систематизировать информацию о людях и их характеристиках, каждое просопографическое исследование уникально и имеет свою специфику, требования которой обязательно должны быть отражены при даталогическом проектировании и выборе концептуальной модели базы данных. Выбор модели опирается на критерий полноты имеющейся информации о предметной области, а также на представление концепции времени и методы устранения неопределенности данных.

Таким образом, цифровизация просопографии позволила исследователям задействовать большие вычислительные мощности, хранить данные каталогов при помощи удобных форматов хранения сложноорганизованной информации, а также обеспечить доступность данных для других исследователей.

#### Библиографический список

1. Поршнева О. С., Разинков С. Л. Женщины на высших дипломатических должностях в государствах бывшего СССР: опыт создания базы данных // Вестник РГГУ. Серия «Политология. История. Международные отношения». 2023. № 1. С. 57–75. DOI: 10.28995/2073- 6339-2023-1-57-75.
2. Просопографическое исследование Преподавательского состава Тульского Механического Института: Главная страница проекта. URL: <http://www.teachers-1930.narod.ru/prosopography.html> (дата обращения: 10.04.2023).
3. Conceptual Modeling of prosopographical Databases Integrating Quality Dimensions / A. Jacky, Isabelle Comyn-Wattiau, Stéphane Lamassé, Cédric du Mouza, 2021.
4. PASE.: главная страница проекта. URL: <https://pase.ac.uk/index.html> (дата обращения: 10.04.2023).
5. PBW 2016: главная страница проекта. URL: <https://pbw2016.kdl.kcl.ac.uk/> (дата обращения: 10.04.2023).
6. Портал Архивы России: главная страница проекта. URL: <https://rusarchives.ru/> (дата обращения: 10.04.2023).
7. Павленко А. П. Просопографическая база данных «Офицеры флота в выборных органах революционной власти в 1917–1918 гг.» // Документ. Архив. История. Современность: материалы VII Всероссийской научно-практической конференции с международным участием. Екатеринбург: Изд-во Уральского университета, 2018. С. 494–500.
8. От просопографии к статистике: методика анализа баз данных по источникам, содержащим динамическую информацию. URL: <http://www.hist.msu.ru/Labour/Article/barnaul.htm> (дата обращения: 10.04.2023).
9. Bol P.K. GIS. Prosopography and History. *Annals of GIS*. 2012. 18 (1). P. 3–5.
10. Graham S., Ruffini G. Network Analysis and Greco-Roman Prosopography // *Prosopography Approaches and Applications. A Handbook*, 325–336. K.S.B. Keats-Rohan (ed.), 2007. P. 579–601.
11. Verbruggen C. Combining Social Network Analysis and Prosopography // *Prosopography Approaches and Applications. A Handbook*. Linacre College, 2007.
12. Tuominen J. Emlo prosopographical data model: Towards a biographical conceptual reference model // Technical report, Cost Action IS1310, Reassembling the Republic of Letters, Aalto University, 2016.

УДК 004.912

**М. В. Угольников**

студентка кафедры информационных систем и технологий

**А. В. Бржезовский** – кандидат технических наук, доцент – научный руководитель

**ПОДСИСТЕМА ДЛЯ СОПОСТАВЛЕНИЯ НОМЕНКЛАТУРЫ ПОСРЕДСТВОМ ТОКЕНИЗАЦИИ ТЕКСТА**

На различных производствах, в процессе обработки поступивших заказов, часто требуется решать проблему сопоставления поступившего запроса от клиента с существующей номенклатурой на предприятии. В реальной жизни менеджеры могут принимать заказы из разных источников, таких как почта, мессенджеры, телефонные звонки, заявки с сайта и т. д. Например, в номенклатурной базе существует позиция: «Переходник с накидной гайкой STOUT 16xG 1/2F». Заказчики, оформляя заявку на закупку этой позиции, могут называть ее по-разному. Примеры названий деталей представлены на рис. 1.

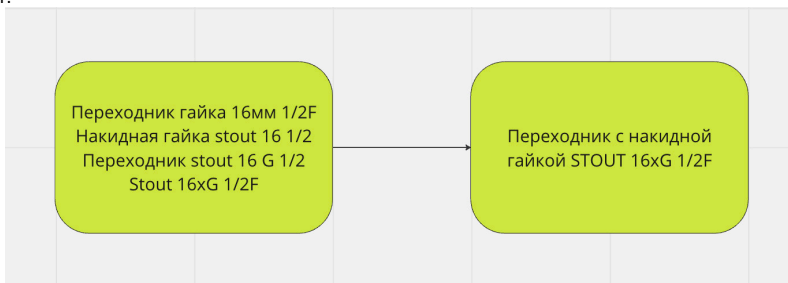


Рис. 1. Пример сопоставления

Далее эти данные собираются в единую систему, откуда их начинают обрабатывать. На рис. 2 представлена схема бизнес-процесса обработки поступивших заказов.

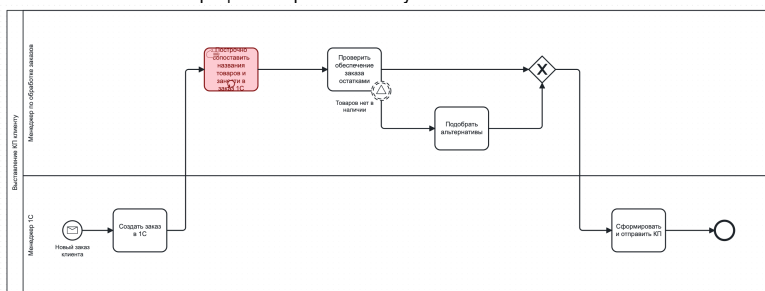


Рис. 2. BPMN обработки поступивших заказов

Если процесс не автоматизирован, то каждое сообщение от заказчика требуется проверить вручную. Скорость обработки заявки зависит от человеческого фактора, что вносит определенную нестабильность в процесс. На реальном примере было зафиксировано, что 23 менеджера сопоставляют заказы от 3 до 5 дней.

С ростом производства будет увеличиваться поток заказов, расширяться номенклатурная база, будут появляться новые каналы для входящих запросов. Соответственно, будет увеличиваться штат сотрудников для обработки заказов. А это увеличит временные и денежные затраты предприятия.

Из всего вышесказанного можно сделать вывод, что процесс сопоставления заказов и номенклатуры производства требует автоматизированного решения для увеличения скорости обработки поступивших запросов, уменьшения расходов на выплату заработных плат и, в итоге, кратного увеличения прибыли предприятия.

Сопоставление данных часто ассоциируется с применением решений на базе обработки данных, хранящихся в базах данных. Например, когда требуется объединить в одну таблицу информацию из разных источников [1, 2]. Принцип сопоставления в данном случае реализован либо на абсолютно точном совпадении, либо на универсальном совпадении – совпадении с неким ключом [3]. В данной ситуации использование подобных решений сможет обработать лишь малую часть информации, остальное просто не пройдет проверку, и придется все также обрабатывать данные вручную.

Более подходящим вариантом реализации автоматизации процесса сопоставления видится разработка информационной системы, основанной на технологиях полнотекстового поиска и аналитики. Подобные поисковые системы позволяют в режиме реального времени обрабатывать, хранить и анализировать поступающую информацию, что немаловажно для оперативной обработки заказов.

Одной из самых популярных систем поиска является Elasticsearch (ES) [4]. Высокие рейтинги ES обусловлены стабильной производительностью на большом объеме данных как для индексов, так и для всех типов запросов [5]. Принцип работы с текстом основывается на внутреннем анализаторе системы. Анализатор представляет собой последовательность из обработчиков текста.

Первый обработчик анализатора перебирает полученный текст через символьный фильтр. Принцип работы заключается в модификации символов к определенному виду, удалении ненужных символов или дополнении. Например, перевод текста в нижний регистр, удаление html-тегов, преобразование к нужной раскладке клавиатуры.

Далее обработанный текст передается токенизатору. Этот обработчик использует метод токенизации слов – процесс разбиения текстового потока по правилам на отдельные слова. Классическим символом для разбиения является пробел. После сегментации формируется набор слов или корней слов. На рис. 3 представлен пример сегментации поступивших заказов. В данном случае символом-разделителем является запятая.

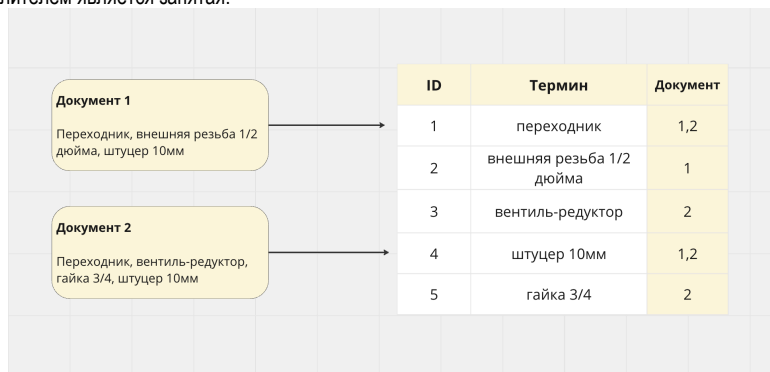


Рис. 3. Токенизация документов

Полученный набор попадает на этап фильтрации токенов. Токены могут добавляться, удаляться или изменяться. Например, фильтры могут удалять часто используемые служебные символы, слова. Заказчик может перечислить характеристики товаров через точку с запятой. Этот символ не является полезным при поиске, он нужен для лучшей читабельности человеком. После всех этапов анализатора на выходе имеется набор токенов, который сохраняется в индексе Elasticsearch. Процесс преобразования текста в анализаторе позволяет сохранять максимум смысла при минимуме объема знаков.

Elasticsearch осуществляет поиск слов из запроса по индексу. Способ поиска можно настроить: выбрать предварительный анализ или нет, сделать поиск по точному совпадению или нечеткому.

Для того, чтобы поиск заказов осуществлялся точнее и быстрее, требуется обучить систему на поступающих данных. Для этого потребуется помощь операторов, которые будут распределять результаты поиска на белые и черные списки. Операторы проверяют корректность сопоставления терминологии из документов заказчика с набором товаров предприятия в базе данных. Если связь «Запрос заказчика» – «Номенклатура в БД» верна, то такая запись попадает в белый список. Это сопоставление сохраняется и в дальнейшем будет использоваться без Elasticsearch. В противном случае связь попадает в черный список. Возможно, запрос был некорректно составлен или такая позиция товара в номенклатуре отсутствует. В любом случае, такое сопоставление использоваться больше не будет.

Таким образом подобная система реализует 2 основных функциональных модуля:

1. Гибкий и масштабируемый поиск и сопоставление данных с помощью утилиты Elasticsearch.
2. Фиксирование и повторное использование удачных результатов сопоставления в автоматическом режиме.

На рис. 4 представлена схема бизнес-процесса сопоставления номенклатуры с учетом разработанной подсистемы сопоставления.

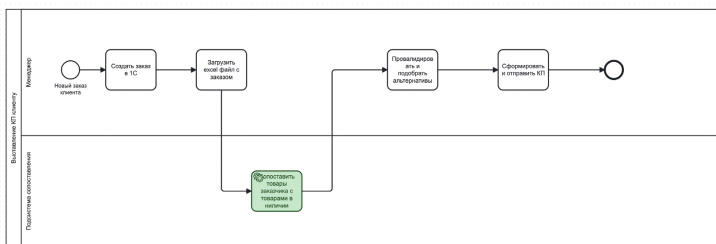


Рис. 4. BPMN обработки заказов с подсистемой

Подпрограмма для сопоставления заказов с номенклатурой производства состоит из нескольких компонентов, каждый из которых выполняет определенную функцию. Основными компонентами подпрограммы являются система сопоставления (СС), Ядро приложения для обеспечения взаимодействия между узлами, CRM система и система учета заказчика (ERP), например 1С. Схема взаимодействия компонентов подпрограммы представлена на рис. 5.

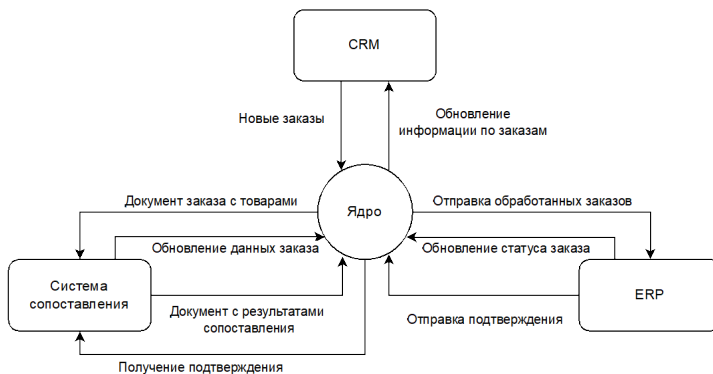


Рис. 5. Схема взаимодействия



Алгоритм работы подпрограммы начинается с того, что Ядро приложения делает запрос в CRM-систему о количестве созданных за текущий день заказов. Если такие заказы отсутствуют в базе данных ядра, они добавляются в базу.

При обращении в Ядро СС или ERP с данными об обновлении заказа, Ядро запускает бизнес-процесс в CRM, который записывает данные в карточку соответствующего заказа. СС опрашивает Ядро на предмет появления новых заказов в CRM за текущий день и, если такие заказы отсутствуют в базе данных СС, они записываются и сохраняются в БД. Если поставщик заказа отсутствует в СС, система автоматически отправляет запрос в Ядро для создания нового заказа в CRM-системе и записи его в базу данных.

При получении данных от ERP об успешном создании заказа, СС отправляет информацию в Ядро для обновления соответствующего заказа в CRM. При изменении статуса заказа в ERP, она отправляет измененные данные в Ядро, которое обновляет их в CRM. Когда ERP запишет все заказы, список заказов, которые корректно записались, отправляются обратно в СС. СС меняет статус заказа на "доставлен в ERP " и отправляет изменения в Ядро для последующего обновления в CRM.

Таким образом, алгоритм работы подпрограммы обеспечивает своевременное и точное сопоставление заказов с номенклатурой производства, а также эффективный обмен данными между компонентами системы.

Разработка и внедрение таких сервисов возможно в любой сфере деятельности, где требуется автоматизировать процесс обработки входных данных, не имеющих определенную структуру, а только набор ключевых понятий в разнородном виде и имеющиеся блоки хранимой информации, носящей справочный характер. Это поможет значительно сэкономить время и трудозатраты, выделяемые на обработку такой задачи в ручном режиме. Клиент сможет повысить поток поступающих запросов, что незамедлительно приведет к увеличению потока денежных средств внутри производства и, как следствие, прирост чистой прибыли компании.

### Библиографический список

1. Механизм сопоставления данных при обмене через универсальный формат. URL: <https://its.1c.ru/db/metod8dev/content/5870/hdoc> (дата обращения: 01.03.2023).
2. Сопоставление данных. URL: <https://learn.microsoft.com/ru-ru/sql/data-quality-services/data-matching?view=sql-server-ver16> (дата обращения: 01.03.2023).
3. Гусятников В. Н., Палькинч Е. А. Эффективность алгоритмов сопоставления персональных данных // Программные продукты и системы. 2021. № 4.
4. Поиск по вашему сайту, как в Яндексe или Google: зачем компаниям нужен Elasticsearch. URL: <https://vc.ru/services/260157-poisk-po-vashemu-saytu-kak-v-yandekse-ili-google-zachem-kompaniyam-nuzhen-elasticsearch> (дата обращения: 05.03.2023).
5. Сравнение эффективности поиска: Elasticsearch и конкуренты. URL: <https://habr.com/ru/articles/581394> (дата обращения: 05.03.2023).

УДК 004.65

**Ю. А. Ускова**

студент кафедры информационных систем и технологий

**А. В. Бржезовский** – кандидат технических наук, доцент – научный руководитель

### КОМБИНАЦИЯ ШАБЛОНОВ ПРИ СОЗДАНИИ КОРПОРАТИВНОГО ПРИЛОЖЕНИЯ С МИКРОСЕРВИСНОЙ АРХИТЕКТУРОЙ

В настоящее время все больше компаний создают свои приложения с микросервисной архитектурой, а уже существующие предпочитают отходить от монолита к микросервисам. При создании и переходах к микросервисам часто используются паттерны проектирования.

Корпоративные приложения помогают оптимизировать работу компании. Это заключается в упрощении коммуникации сотрудников, ускорении обмена информацией внутри фирмы и создании общего доступа ко всем данным организации. Корпоративное приложение может использоваться в интернет-магазине. В таком приложении будет упрощена коммуникация между продавцами и покупателями, оно поможет вести учет товаров и финансов, а также возможность собирать ошибки.

Монолиты подходят для приложений на начальных этапах проекта. Монолиты являются одним файлом, в которых содержится все данные, из-за чего обеспечивают легкое развертывание. Это позволяет сразу выпускать все, что есть в приложении. Так как файл один, то и при изменении его содержания придется снова грузить весь файл, что не очень удобно. По мере добавления в монолитный проект новых возможностей рано или поздно возникают сложности при работе нескольких разработчиков с единой базой кода. Учащаются конфликты в коде и увеличивается риск того, что при обновлении одной возможности появятся ошибки в другой, не связанной возможности. Для избежания подобных ошибок следует перейти на микросервисную архитектуру, которая позволит распределить нагрузку на сервер, разделяя ее между собой. Каждый микросервер имеет собственную базу и отвечает только за свои действия, что обеспечивает надежность системы и упрощает работу с системой. [1, 2].

Шаблоны (паттерны) проектирования – типовые решения, которые представляют простые решения, часто повторяющихся задач проектирования. Задача паттерна является предложение решения определенной задачи в конкретном контексте. Для решения определенной задачи выбирается конкретный шаблон и способ его использования.

В разрабатываемом приложении паттерн посредник облегчит общение между элементами. Паттерн обеспечит оформление заказа. По нажатию на кнопку должно всплывать окно, сообщающее о принятии заказа. На рис. 1, а каждый объект зависит от другого объекта. Прежде чем передать значение в диалог кнопка должна проверить значения в текстовых файлах и чек-боксах, которые тоже зависят друг от друга. На рис. 1, б у каждого объекта своя роль, и единственное действие кнопки – это передать значение в диалог, а не проверять все файлы.

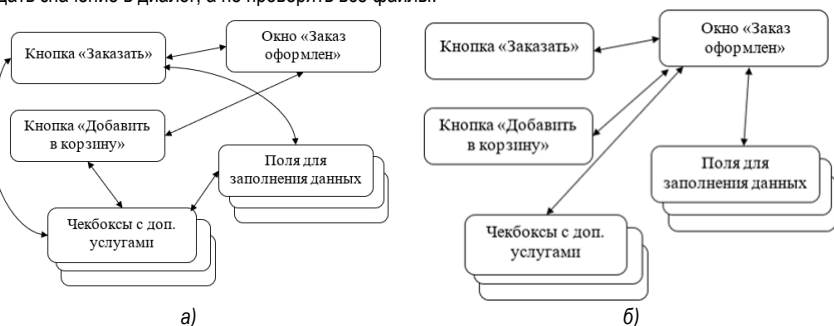


Рис. 1. Оформление заказа:

а – до использования шаблона Посредник; б – после использования шаблона

Паттерн Душитель обеспечит миграцию монолитного приложения на микросервисную архитектуру путем постепенного переноса существующих функций в микросервисы. Так как изначально будет обеспечиваться доставка в два региона, то начальный этап приложения будет сделан как монолит и в последствии перенесен на новый микросервис.

Паттерн Api Getaway подходит к паттерну Душитель как шлюз. После заполнения и сбора всех данных отправляет их и получает обратно.

У интернет-магазина есть десктопная версия и мобильное приложение. Без шлюза каждая из версий обрабатывает несколько микросервисов одновременно, что замедляет работу приложения и может привести к ошибке (рис. 2, а). Шлюз Api помогает избавиться от этой проблемы, и сократить обработку до одного сервиса (рис. 2, б).

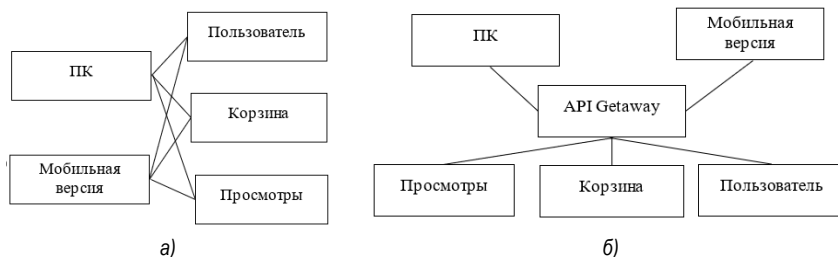


Рис. 2. Синхронизация устройств: а – до использования шлюза; б – после использования шлюза

Шаблон «База данных на сервис» (Database Per Service). Каждой БД предоставляется свой сервис. Повышает автономность микросервисов и уменьшает связь между командами, разрабатывающими отдельные сервисы. Микросервисы не могут использовать базы данных других сервисов.

При оформлении заказа сервис должен убедиться, что товар есть в наличии, а также иметь возможность объединить несколько заказов из одного региона в один

Шаблон Поиск событий. По сути, паттерн автоматически обновляет базу данных и отправляет данные. Также к нему подходит шлюз API, который будет являться промежуточным связующим звеном между сервисами.

Применение шаблона «Event sourcing», «Database per service» влечет за собой сложность извлечения данных из нескольких сервисов. Для решения этой проблемы был применен паттерн «CQRS».

Шаблон «Разделение команд и запросов» (Command Query Responsibility Segregation, CQRS). Шаблон, который отделяет чтение и изменение данных друг от друга. Шаблон обеспечивает согласованность данных и изменяет только те данные, которые находятся в зоне ответственности [3, 4].

Эти три шаблона чаще всего используются вместе. То есть после отправки команды, какой-то микросервис ее выполнит. Event sourcing сохранит это действие и отправит на сервер.

Рассмотренные выше шаблоны хорошо сочетаются вместе. Есть паттерны которые облегчат переход от монолита к микросервисам и паттерны, которые облегчат работу с микросервисами.

На основании изученного материала планируется разработать микросервисы, подходящее для закононаправленного интернет-магазина.

Как тестовый вариант предлагается создать монолит с двумя регионами, которые в последствии с помощью паттерны душитель будут перенесены в микросервисы, где в последствии добавятся другие города.

Предполагается, что доставка будет осуществляться во все города, товар может находится как в городе покупателя, так и в другом городе. Верхний слой будет разделен по регионам, который будет иметь одинаковые базы. Здесь удобней использовать паттерн база данных на сервер, чтобы отделить данные от других микросервисов (рис. 3).

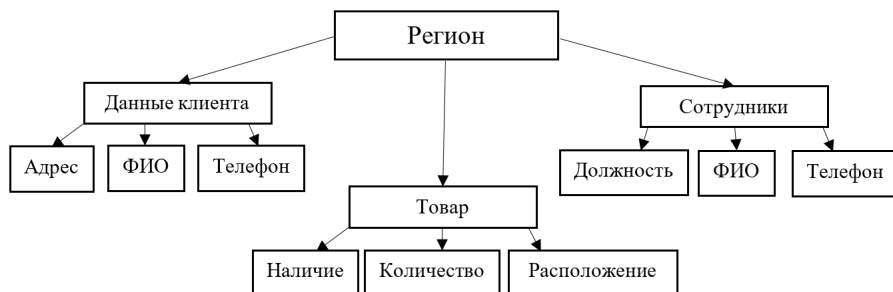


Рис. 3. Разделение данных

В каждом регионе будет информация о покупателе, сотруднике, и товаре. Так как некоторые поля будут одинаковые и взаимодействовать друг с другом, лучше использовать паттерн посредник, для уменьшения нагрузки на сервер. У магазина будет мобильное приложение и версия для ПК, поэтому для синхронизации данных будет использован шлюз API Getaway.

#### Библиографический список

1. Рогозов Ю. И., Свиридов А. С., Кучеров С. А. Архитектура информационных систем: учеб. пособие. Ростов-на-Дону: Изд-во ЮФУ, 2014. 117 с.
2. Ньюмен С. Создание микросервисов. СПб.: Питер, 2016. 304 с.
3. Паттерны проектирования: какие бывают и как выбрать нужный. URL: <https://gb.ru/blog/patterny-proektirovaniya/> (дата обращения: 10.04.2023).
4. 26 основных паттернов микросервисной разработки. URL: <https://mcs.mail.ru/blog/26-osnovnyh-patternov-mikroservisnoj-razrabotki> (дата обращения: 10.04.2023).

УДК 612.843.72

**В. Д. Хвойновский**

студент кафедры информационных систем и технологий

**О. И. Красильникова** – доцент, кандидат технических наук, доцент – научный руководитель

### ОБЗОР МЕТОДОВ СОЗДАНИЯ КАРТ ГЛУБИНЫ

Карта глубины представляет собой изображение, яркость пикселей которого определяется расстоянием от камеры до данной точки сцены [1]. В графическом виде она представляется как изображение в градациях серого, где черный цвет обозначает наиболее удаленную точку на изображении, а белый – самую близкую к наблюдателю.

Сегодня карты глубины находят применение во многих сферах человеческой деятельности, например, для преобразования изображения и видео в стереоскопический формат, в системах, использующих технологии виртуальной и смешанной реальности, для создания беспилотных автомобилей или обработки фотографий.

Для создания карт глубины разработано множество различных по своей сложности методов. В целом их можно разделить на ручные и автоматические.

При использовании ручного метода оператор должен самостоятельно сегментировать изображение на отдельные фрагменты и закрасить их определенным цветом или градиентом, соответствующим удаленности объектов от камеры. Сегментация обычно выполняется путем ретоскопирования – последовательного обрисовывания каждого объекта. Ручной способ позволяет создавать достаточно качественные и детализированные карты глубины, при этом его главным недостатком является высокая трудозатратность, так как все работы оператор осуществляет самостоятельно [2].

С целью оптимизировать затраты и ускорить создание карты глубины создаются полуавтоматические способы. Именно такие методы описаны в статье [2], среди них можно выделить следующие:

- аппроксимация изображения набором плоскостей;
- формирование карты глубины из исходного изображения;
- формирование карты глубины из контуров исходного изображения;
- использование фактора движения при формировании карты глубины.

Первый из методов подразумевает создание упрощенной карты глубины, содержащей ограниченный набор аппроксимирующих плоскостей. При этом такая карта глубины может быть успешно использована при конвертировании 2D-изображения в стереоскопическое, так как в таком случае допустимы достаточно сильные погрешности в карте глубины [2, 3]. Второй и третий методы подразумевают специальную обработку изображения, которая в полуавтоматическом режиме позволяет отделить объекты переднего плана от фона.

Автоматические методы можно разделить на активные и пассивные. Пассивные методы обычно используют пару или серию снимков для извлечения информации о глубине. К таким методам относятся:

- метод построения глубины из стерео;
- метод получения карты глубины по анализу движения объектов в кадре;
- получение сведений о глубине объекта в кадре путем анализа фокусировки, дефокусировки объектов на снимках;
- пленоптические камеры (Light Field Camera).

Активные методы, в свою очередь, подразумевают использование в своем составе излучателя. К таким методам можно отнести: времяпролетные камеры (Time of Flight, ToF) и структурированные световые камеры (Structured Light Camera).

В отдельную группу можно выделить методы на основе искусственных нейронных сетей (ИНС). Основным преимуществом нейросетей является возможность получить карту глубины из одиночного изображения, снятого на обычную камеру, вместо использования специализированных активных камер или групп фотографий.

Развитию ИНС сегодня существенно способствует рост вычислительной способности устройств, а также интеграция специальных нейронных модулей для работы машинного обучения во многие современные устройства [4].

В источниках [1, 5, 6] представлено большое количество исследований нейронных сетей для генерации карты глубины по исходному 2D-снимку, что показывает интерес к данной области. При этом большая часть из представленных для данной задачи нейронных сетей включают в себя сверточные слои, т. е. являются сверточными нейронными сетями. Причиной такой популярности данного вида ИНС является их возможность учитывать двумерную топологию изображения, а также наличие частичной устойчивости к искажениям: изменениям масштаба, смещениям, поворотам, сменам ракурса. Также одним из главных ее достоинств является большая скорость работы по сравнению с классическим перцептроном [7].

Рассмотрим создание карты глубины с использованием ИНС более подробно на примере сверточной нейронной сети «MiDaS», представленной «Embodied Ai Foundation» и выложенной в открытый доступ на платформе «GitHub». Преимуществом данной нейронной сети является использование большого количества разнородных данных для обучения, что должно давать хорошие результаты для широкого класса входных изображений [8].

Для тестирования подобран набор изображений, содержащих разнообразные типы сцен: крупный план отдельных объектов, кадры с наличием выраженной перспективы (например, уходящая вдаль улица), кадры с наличием выраженной линии горизонта, плоские поверхности.

В результате анализа результатов на выбранном наборе входных изображений выделены основные ошибки, присущие данному виду конвертации:

- ошибки определения глубины объектов на плоской поверхности, где ИНС ошибочно пытается добавить к рассматриваемой сцене перспективу, что приводит к ошибкам (рис. 1, слева);
- ошибки определения глубины объектов без явной точки опоры, например, парящие в воздухе воздушные шары, птицы и т. п (рис. 1, справа).

Также высока вероятность ошибки в обработке изображений с плавным удалением объектов к линии горизонта. Такие объекты на карте глубины должны быть показаны градиентной заливкой, однако, ИНС не всегда корректно справляется с этим (рис. 2).

Подборка изображений, с которыми ИНС справилась успешно приведена на рис. 3.

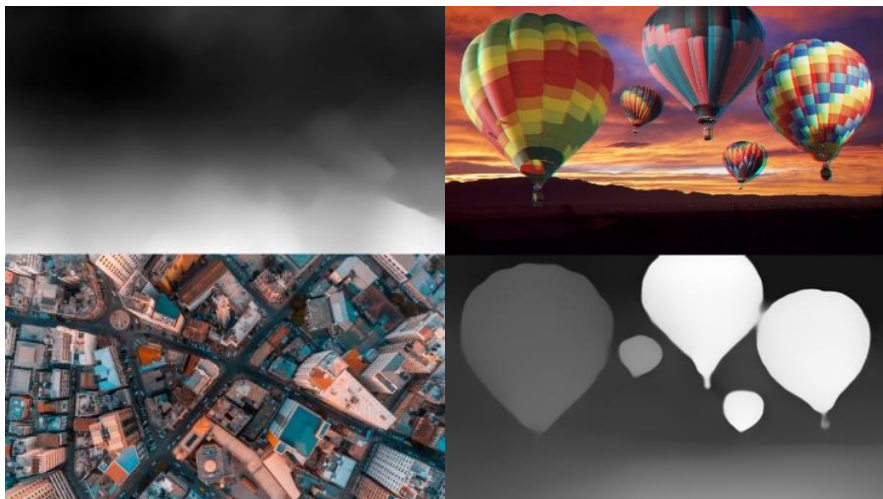


Рис. 1. Примеры ошибок при создании карты глубины ИНС



Рис. 2. Примеры отсутствия градиента в созданной ИНС карте глубины



Рис. 3. Примеры качественных карты глубины, сгенерированных ИНС

Кроме перечисленных выше недостатков, которые проявляются при использовании ИНС для создания карты глубины, можно также назвать проблему изменения формируемой карты глубины даже при небольших модификациях исходного изображения. Эти изменения выражаются в небольших колебаниях положения границ объектов или в слегка изменяющейся глубине одних и тех же объектов на сравниваемых кадрах.

Данные недостатки можно заметить при покадровом формировании карты глубины по видео. При этом, при использовании полученной карты глубины для конвертации видео в стереоскопический формат, второй недостаток остается незаметным зрителю, а первый замечен преимущественно только в малоподвижных сценах.

На сегодняшний день существует множество различных способов создания карт глубины от ручных методов до специализированных камер и нейронных сетей, позволяющих автоматизировать рутинные операции.

Применение ИНС является одним из самых новых и перспективных направлений и позволяет создавать карты глубины из одиночного снимка, что позволяет существенно упростить задачу получения карты глубины. При этом ИНС нередко допускают ошибки при построении карты глубины. Кроме того, следует помнить, что ИНС не имеют реальных данных по глубине сцены, а лишь ее предсказывают.

Таким образом, нейронные сети могут быть успешно применены для задач, не требующих максимальной точности и надежности результата, например, при конвертации видео в стереоскопический формат. Однако следует иметь в виду возможные ошибки и, при необходимости, дорабатывать и корректировать проблемные моменты вручную. В таких сферах как робототехника или беспилотные автомобили максимальную точность и надежность предсказания могут обеспечить только специализированные камеры.

### Библиографический список

1. ИТМО Вики. Карта глубины. URL: [https://neerc.ifmo.ru/wiki/index.php?title=Карта\\_глубины](https://neerc.ifmo.ru/wiki/index.php?title=Карта_глубины) (дата обращения: 15.05.2023).
2. Красильников Н. Н., Красильникова О. И. Методы конвертации 2D-изображений и видео в стереоскопический формат // Информационно-управляющие системы. 2015. № 5 (78). С. 18–25.
3. Ватолин Д. Камеры глубины – тихая революция (когда роботы будут видеть). URL: <https://habr.com/ru/post/457524/> (дата обращения: 15.05.2023).
4. Droider.Ru. Зачем нам нужны нейронные процессоры? URL: <https://habr.com/ru/company/droider/blog/508290/> (дата обращения: 16.05.2023).
5. Селихов В. А. Построение карты глубины с использованием глубокой сверточной нейронной сети // Известия ТулГУ. Технические науки. 2018. № 7. С. 262–277.
6. Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, Yasutaka Furukawa. PlaneNet: Piece-wise Planar Reconstruction from a Single RGB Image. URL: [https://openaccess.thecvf.com/content\\_cvpr\\_2018/papers/Liu\\_PlaneNet\\_Piece-Wise\\_Planar\\_CVPR\\_2018\\_paper.pdf](https://openaccess.thecvf.com/content_cvpr_2018/papers/Liu_PlaneNet_Piece-Wise_Planar_CVPR_2018_paper.pdf) (дата обращения: 11.05.2023).
7. Бредихин А. И. Алгоритмы обучения сверточных нейронных сетей // Вестник ЮГУ. 2019. № 1 (52).
8. Rene Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, Vladlen Koltun. Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer // IEEE Transactions on pattern analysis and machine intelligence. URL: <https://arxiv.org/abs/1907.01341> (дата обращения: 17.05.2023).



УДК 004.921

**А. П. Шабунин**

студент кафедры информационных систем и технологий

**А. В. Аграновский** – доцент, кандидат технических наук – научный руководитель

### ПРИМЕНЕНИЕ МЕТОДОВ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ В МОБИЛЬНОМ ПРИЛОЖЕНИИ ДЛЯ ВЫБОРА ПРЕДМЕТОВ МЕБЕЛИ

В настоящее время сфера информационных технологий достигла значительного развития, и многие процессы жизни людей переведены в цифровую среду.

Многие люди ввиду своей занятости предпочитают делать покупки онлайн, что значительно экономит личное время, однако становится актуальной проблема выбора товара.

Мир маркетинга не стоит на месте, постоянно совершенствуется и предлагает новые способы представления товаров. Господствует экономика впечатлений – концепция, отражающая создание ценности и универсального впечатления для конкретного потребителя, и дальнейшее его вовлечение в процесс различными способами.

Одним из перспективных направлений маркетинга на сегодняшний день является использование технологии дополненной реальности (AR – Augmented Reality).

Дополненная реальность — это интеграция в реальное пространство текста, графики или аудио с помощью носимого устройства отображения или проецируемых графических наложений в режиме реального времени [1].

Компания «Lumus Vision» опубликовала факты о дополненной реальности, где говорится о том, что 71% покупателей предпочли выбрать те магазины розничной торговли, где предлагают опыт дополненной реальности [2].

Если рассматривать современную тенденцию к переходу к повсеместному использованию мобильных устройств в повседневной жизни, то применение информационных технологий имеет большое преимущество для привлечения и заинтересованности конечного потребителя. Нынешние устройства обладают большой оперативной и постоянной памятью, большей процессорной мощностью, отличной камерой, удобными форматами экрана, что позволяет использовать дополненную реальность почти на любом современном мобильном устройстве.

Дополненная реальность может оказать огромную помощь в развитии электронной торговли компаниям, которые ориентируются на непосредственное взаимодействие с потребителями, используя технологии AR в своих маркетинговых коммуникациях.

Например, технологии дополненной реальности быстро развивались и распространялись на рынке мебельной продукции и товаров для дома. Согласно статистике International Data Corporation, на момент начала 2020 года, на фоне пандемии продажи упали на 60% из-за введенных мер, при этом онлайн-продажи за данный срок выросли почти в два раза в годовом выражении [3]. Усиление конкуренции среди мебельных ритейлеров привело к активному внедрению AR-приложений, которые упростили процесс выбора предметов интерьера. Среди таких приложений можно выделить Hoff дизайн, Roomle 3D & AR планировщик и Столплит.

На основе результатов анализа существующих приложений, которые используют визуализацию средствами дополненной реальности, можно выделить явные недостатки в аспектах реалистичности отображения объектов, функциональности и работы приложения в целом. Предлагаемое автором статьи программное решение обеспечивает улучшение качества визуализации виртуальных объектов, а также оптимизацию для работы на мобильных устройствах без торможения и аварийного закрытия приложения из-за нехватки ресурсов.

Для реализации технологии дополненной реальности необходимы два основных программных компонента – трекинг и визуализация. На настоящий момент исследователями сформирована обширная теоретическая и алгоритмическая база для их реализации как в виде различных отдельных компонентов, так и в виде интегрированных программ и наборов средств разработки. К отдельным компонентам относятся библиотеки и фреймворки компьютерного зрения, трехмерной графики и другие решения.

Трекинг – это сложный процесс, связанный с отслеживанием положения наблюдателя относительно окружающей обстановки. Для использования в разработанном приложении был выбран SLAM – метод на основе алгоритмов, которые функционируют по технологии одновременной навигации и составления карты, что позволяет работать в реальном времени, не используя маркер, а строя в процессе трекинга трехмерную модель [3].

На самом простом уровне карта SLAM представляет собой разреженное трехмерное облако точек, в котором каждая точка соответствует координатам оптического объекта в сцене (например, точке на поверхности стола). Обычно они также содержат значительное количество дополнительных метаданных, например, «надежность» точки, измеряемая тем, на скольких кадрах эта особенность была обнаружена в тех же координатах за последнее время (например, белое пятно на кошке не может быть помечено как надежное, потому что кошка перемещается) [4].

В качестве движка для трекинга был использован набор средств разработки ARFoundation (кроссплатформенный фреймворк, сочетающий функционал ARCore и ARKit). Данное средство было специально создано для мобильных устройств.

Для реализации части технологии дополненной реальности, связанной с трехмерной визуализацией, может быть применено множество существующих решений. В ходе поиска и анализа была выбрана среда разработки Unity. В ней присутствует графический движок, оптимизированный для работы на мобильных устройствах.

Для совершенствования визуализации средствами дополненной реальности на массовых мобильных устройствах были рассмотрены следующие вопросы:

- Реалистичность отображения объектов.
- Возможность изменения параметров объектов при визуализации.
- Формат и структура хранения объектов.
- Манипуляция объектами в пространстве.
- Особенности проектирования графического интерфейса пользователя для ДР.

Методики, разработанные с целью улучшения визуализации и оптимизации работы информационной системы с дополненной реальностью на массовых мобильных устройствах, показаны на рис. 1.

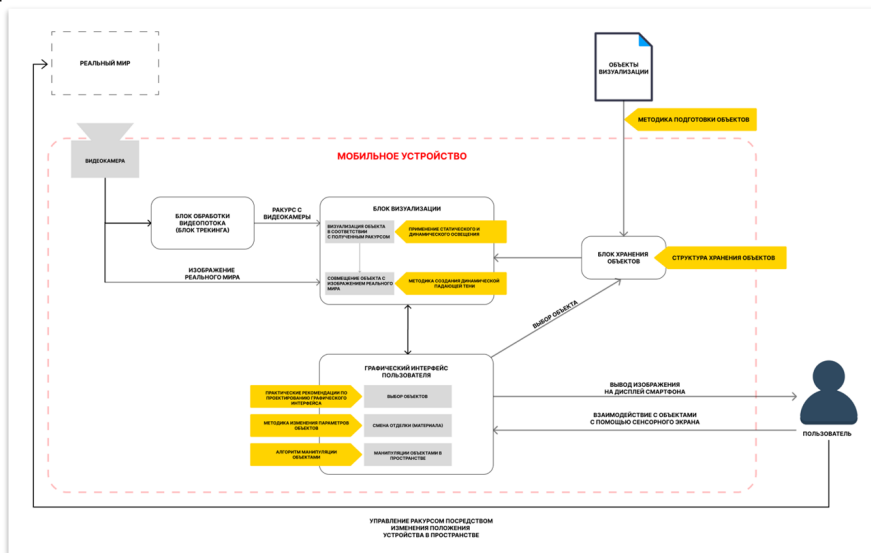


Рис. 1. Общая схема структуры разработанной информационной системы

В первую очередь были рассмотрены вопросы реалистичности визуализируемых объектов. Синтезированное компьютерное изображение может соответствовать оригиналу при визуальном наблюдении только на психофизическом уровне (в голове человека).

Максимальное приближение к оригиналу при этом можно обеспечить если математическая модель и обрабатывающая программа точно передают условия освещения, геометрическую форму объектов, их взаимное положение, размер, положение теней и другие особенности реальной сцены.

Выбранная среда разработки поддерживает все стандартные источники освещения, а также различные элементы, используемые для тесселяции и отражений. Их комбинация может быть применена для создания схемы освещения объекта, образованной несколькими точечными источниками и рассеянным светом. Такая структура освещения вполне может служить примером схемы динамического освещения, которая просчитывается в каждом кадре. При обычной визуализации используются сложные ресурсозатратные алгоритмы для расчета освещения, которые позволяют достичь реалистичного изображения. Для визуализации в режиме реального времени требуются иные подходы.

На сегодняшний день современные потребительские смартфоны не могут при работе с дополненной реальностью в режиме реального времени производить сложные расчеты с применением алгоритмов визуализации для непрямого освещения, а также трассировки лучей. Это связано с тем, что их вычислительные ресурсы не обладают необходимым уровнем производительности и оперативной памяти для просчета в каждом кадре трассировки лучей, визуализации освещения объекта, бликов и других аспектов.

Представленная на рис. 2 разработанная методика предполагает использование источников направленного и рассеянного света, интенсивность которых управляется при помощи написанного скрипта, использующего оценку освещения библиотеки ARFoundation. ARKit и ARCore используют различные аппаратные датчики смартфона для расчета этих данных об освещении.

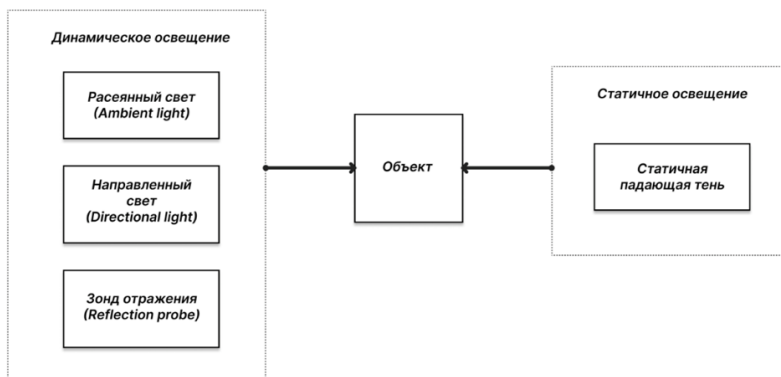


Рис. 2. Методика, совмещающая динамическое и статичное освещение

Также разработана методика по созданию статичной падающей тени, отбрасываемой на плоскую поверхность, на которой располагаются виртуальные объекты. Методика продемонстрирована на рис. 3.

Так же, как и текстура затенения, падающая тень просчитывается заранее в высоком качестве и сохраняется в виде текстуры. Далее текстура применяется к плоскости, расположенной под объектом. В целях оптимизации шейдер материала плоскости не использует информацию об источниках освещения.

Падающая полупрозрачная тень вносит существенный вклад в реалистичность встраивания виртуального объекта в реальное окружение. Отличительной особенностью представленного подхода является возможность его успешного применения в задачах реального времени.



Рис. 3. Методика статичной падающей тени

Имплементация предложенных методик по обеспечению реалистичности при смене материалов изделий потребовала разработки определенного процесса предварительной подготовки моделей (рис. 4), а также разработки особой структуры хранения объектов (рис. 5).



Рис. 4. Методика подготовки трехмерной модели

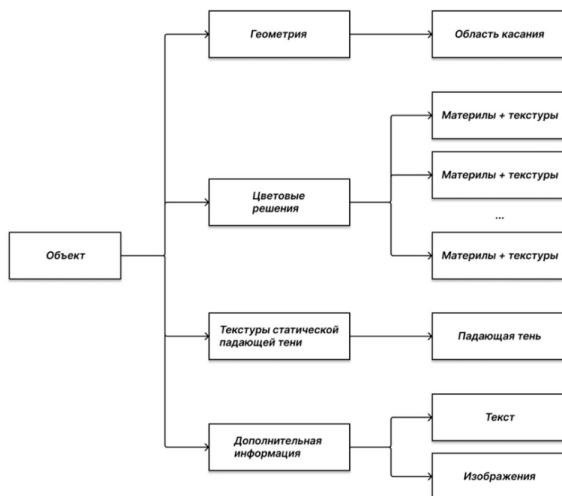


Рис. 5. Структура хранения объектов

С целью улучшения пользовательского взаимодействия был разработан механизм манипуляции трехмерными объектами с использованием жестовых методов ввода, которые продемонстрированы на рис. 6.

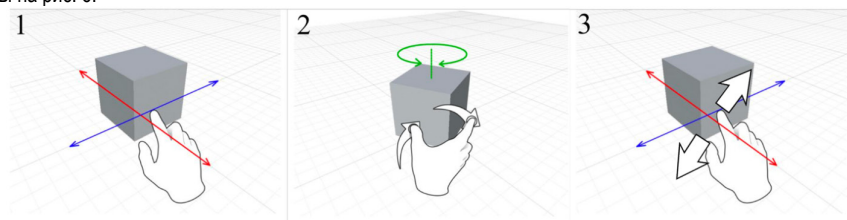


Рис. 6. Использование жестов для вращения (2), перемещения (1) и масштабирования объекта (3)

Задача перемещения трехмерных объектов зачастую может быть упрощена до двухмерной. В данном случае перемещение осуществляется в пределах плоскости по осям  $X$  и  $Z$ , а вращение вокруг своей оси. В таком случае координата  $Y$  остается неизменной, а координаты  $X$  и  $Z$  рассчитываются согласно формуле:

$$\begin{bmatrix} Z_1 \\ X_1 \end{bmatrix} = \begin{bmatrix} \cos & -\sin \\ -\sin & \cos \end{bmatrix} \begin{bmatrix} Z_0 \\ X_0 \end{bmatrix} + \begin{bmatrix} r_3 \\ r_1 \end{bmatrix},$$

где  $\varphi$  – угол поворота объекта вокруг оси  $Y$ ;  $X_0, Y_0, Z_0$  – начальные координаты точки объекта,  $X_1, Y_1, Z_1$  – конечные координаты точки объекта,  $\vec{r} = (r_1, r_2, r_3)$  – вектор перемещения объекта.

Для манипуляции объектами на сцене был разработан алгоритм (рис. 7), обрабатывающий ввод с сенсорного экрана. При обнаружении касания экрана производится проверка, произошло ли попадание в область виртуального объекта. Для этого применяется метод проброса лучей. Виртуальный луч пробрасывается от точки касания пользователем сенсорного экрана, находящейся на плоскости

экрана и найденной при помощи виртуальной камеры Unity. В случае пересечения лучом объекта возвращается положительный результат. При наличии на сцене нескольких виртуальных объектов необходимо определить, с какими из них производится взаимодействие. При обнаружении более одного касания на одном объекте распознается режим вращения.

Угол, на который нужно повернуть объект, определяется путем измерения угла вращения прямой, проведенной через две точки касания, по сравнению с предыдущим кадром. При обнаружении более двух касаний на объекте, обрабатываются только два первых.

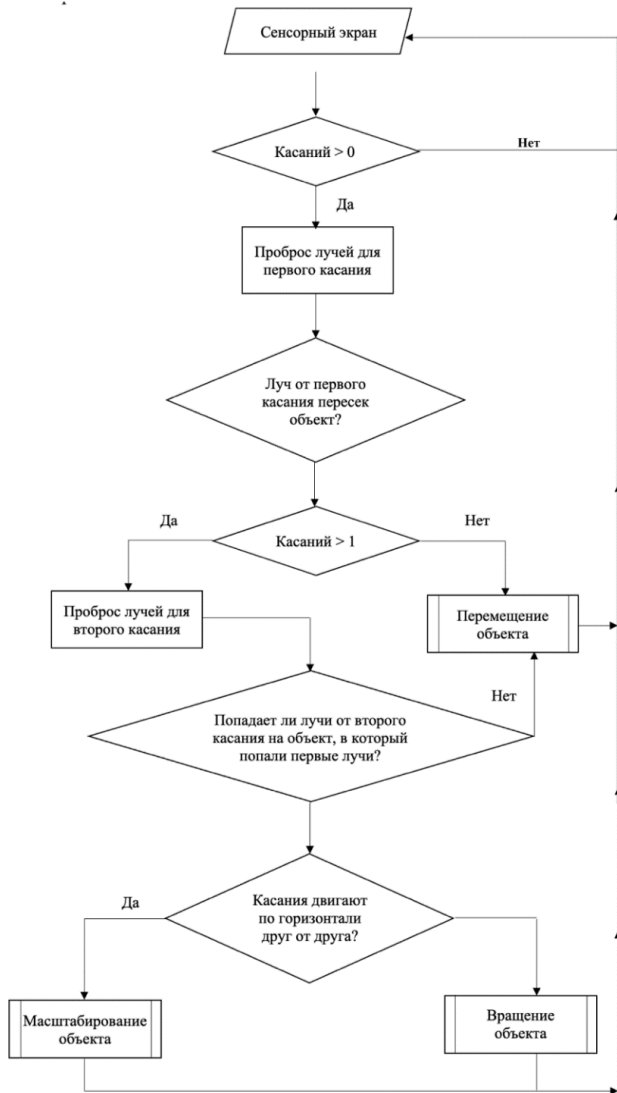


Рис. 7. Алгоритм манипуляции жестовыми методами

Были предложены изменения параметров объекта при визуализации дополненной реальности на примере смены материалов. Они включают в себя смену параметров в реальном времени и упрощенный интерфейс пользователя, доступный для неспециалиста.

Например, смена материалов отдельных полигональных сеток модели была упрощена до выбора пользователем одного из «цветовых решений» – заданных наборов материалов. Алгоритм смены материалов представлен на рис. 8.



Рис. 8. Алгоритм смены материалов

С целью хранения информации о материалах был разработан класс Material Manager. Информация размещена в хранилище в виде массива ссылок на материалы. При выборе пользователем одного из цветовых решений из списка происходит применение соответствующих материалов к частям модели в режиме реального времени.

Разработка графического интерфейса для визуализации средствами дополненной реальности требует ответственного подхода, так как необходимо как можно больше площади экрана выделить под визуализацию и трансляцию изображения реального мира с камеры смартфона, но при этом организовать графический функционал.

Сравнение применяемых в различных приложениях способов организации меню для выбора предметов мебели показало, что главная проблема заключалась в вопросе работы потока, который обеспечивал работу трекинга и самого видеопотока. При рассмотрении данного вопроса было решено скрыть отрисовку видеопотока и перекрыть полноценным экраном с меню для меньшей нагрузки на смартфон, но сам трекинг и видеопоток продолжать обрабатывать в фоновом режиме для того, чтобы на выходе система была в том состоянии, в котором его оставил пользователь.

С целью достижения максимально простого и удобного способа смены материалов отделки объектов в верхней левой части экрана отображается кнопка «+», которая раскрывает меню смены материалов. Меню смены материалов представляет собой маленькие кнопки с размещенными на них фрагментами графических изображений существующих материалов.

Примеры графического интерфейса представлены на рис. 9.

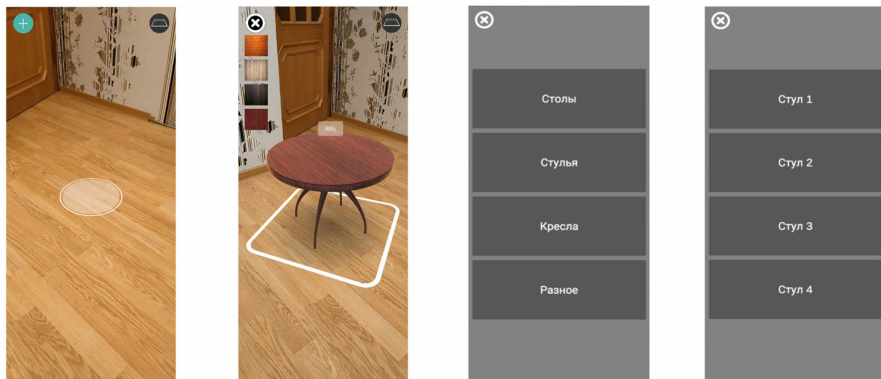


Рис. 9. Примеры графического интерфейса

Таким образом, рассмотренное в статье разработанное мобильное приложение, функционирующее в режиме реального времени, решает важную маркетинговую задачу, позволяя каждому посетителю онлайн-магазина с использованием бытового смартфона легко выбирать предметы мебели, наиболее подходящие к уже имеющемуся интерьеру.

#### Библиографический список

1. Raghav Sood Pro Android Augmented Reality. М.: Аpress. 2012. 1 с.
2. Статистика дополненной реальности. URL: <https://lefronic.com/blog/augmented-reality-statistics/> (дата обращения: 08.03.2023).
3. Руководство IDC по расходам прогнозирует сильный рост дополненной и виртуальной реальности. URL: <https://www.businesswire.com/news/home/20221130005781/en/IDC-Spending-Guide-Forecasts-Strong-Growth-for-Augmented-and-Virtual-Reality> (дата обращения: 12.05.2023).
4. SLAM (метод) // Wikipedia.org. URL: [https://ru.wikipedia.org/wiki/SLAM\\_\(метод\)](https://ru.wikipedia.org/wiki/SLAM_(метод)) (дата обращения: 12.10.2022).
5. Erin Pangilinan, Steve Lukas Creating Augmented & Virtual Realities // Mapping. 2019. 91 с.



УДК 621.3.051

**И. С. Шаров**

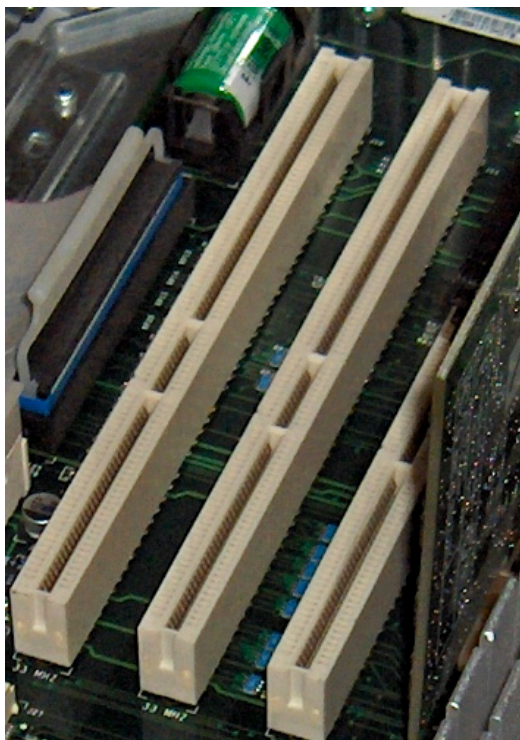
студент кафедры информационных систем и технологий

**А. П. Шенета** – доктор технических наук, профессор – научный руководитель**РАЗРАБОТКА ПЛАТЫ-НОСИТЕЛЯ МИКРОМЕЗОНИННЫХ ПЛАТ**

В настоящее время существуют платы расширения.

Плата расширения – печатная плата, которую устанавливают в слот расширения другой платы системы с целью добавления дополнительных функций. Платы расширения, необходимые для подключения внешних устройств, могут также называться адаптерами или контроллерами этих устройств [1].

Самым распространенным примером является материнская плата со слотами расширения шины PCI (рис. 1).



*Рис. 1. Слоты расширения на материнской плате*

В каждый слот можно вставлять только одну из плат расширения (рис. 2 – 3):

- Видеокарта.
- Звуковая карта.
- Сетевая карта.
- ТВ-тюнер.
- Контроллеры различных портов.

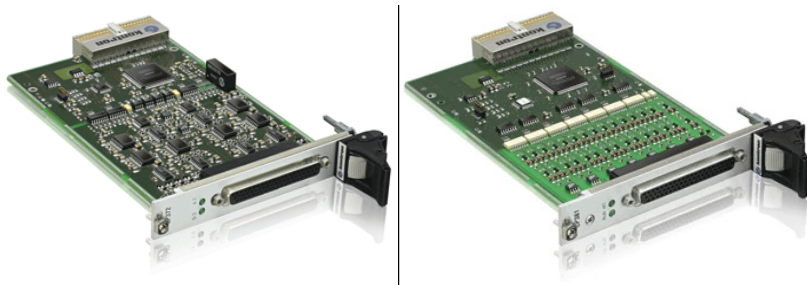


Рис. 2. Пример плат расширения

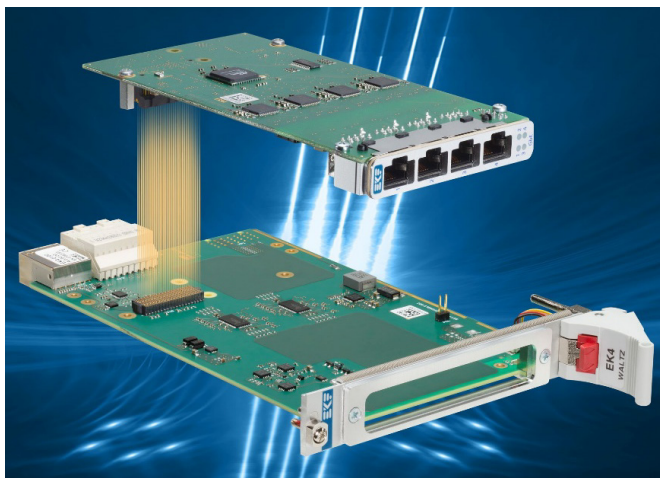


Рис. 3. Пример подключения платы расширения

На данный момент наиболее известной технологией использования плат расширения является наращивание массива ПР (рис. 4).

Проанализировав доступные и известные на данный момент платы расширения, платы-носители и технологии их использования можно выделить следующие положительные стороны:

- Возможность встраиваемости всей системы в конечные изделия.
- Универсальность слотов расширения – к каждому слоту можно подключать любую ПР.
- Возможность конфигурирования плат расширения.

Отрицательные стороны:

- Достаточно большие размеры одной платы расширения.
- Каждая плата расширения является обособленной единицей всей системы – управляется через плату-носитель, но с внешней стороной общается самостоятельно, что приводит к использованию отдельных интерфейсных разъемов на каждой такой ПР.

– Недостаточно гибкое конфигурирование плат-расширений. Как уже известно, есть две технологии использования ПР – подключение напрямую в слоты расширения и наращивание массива плат-расширений. При использовании первой технологии количество слотов расширения технически ограничено, и к каждому отдельному слоту можно подключить только одну ПР. При использовании второй

технологии наращивание ПР приводит к значительному увеличению габаритов всей системы в целом, что не допускает гибкую встраиваемость в конечные изделия пользователя.



*Рис. 4. Наращивание массива плат расширений*

Таким образом, на основании анализа предметной области была поставлена задача разработать систему формирования интерфейсов передачи данных, которая сохраняет или улучшает известные преимущества и не имеет описанные недостатки.

Система получила название «Модуль интерфейсный универсальный».

К МИУ выдвигались следующие тактико-технические требования [2]:

- Максимально гибкое конфигурирование набора интерфейсов.
- Универсальность.
- Встраиваемость модуля в конечные изделия пользователя.
- Определенный набор интерфейсов – GPIO, RS-232, RS-485/RS-422, CAN.
- Малые габариты конечной системы.

Целью выполнения работы является разработка универсального интерфейсного модуля, позволяющего формировать низкоскоростные интерфейсы таких как (GPIO, RS-232, RS-485/RS-422, CAN). И других, в которых может возникнуть потребность. Конструкция и принцип работы должны предусматривать возможность расширения функций [3, 4, 5, 6].

Интерфейсный модуль состоит из платы носителя, обеспечивающей гальваническую развязку и сопряжение с интерфейсом PCIe и микромезонинных плат, формирующих физический уровень, либо сочетание физического и логического уровня интерфейса.

Вся система представлена в виде печатных электронных плат (PCB), с распаянными на них определенными комплектующими. Один набор PCB представляет собой 4-слойные печатные платы, другой набор – 12-слойные печатные платы.

Система реализована на технологии мезонин. Мезонин – плата, вставляемая в карту расширения и располагающаяся параллельно плате-носителю. В качестве носителя как правило используются карты расширения стандартов ISA, PCI, VMEbus, CompactPCI, OpenVPX и др. Носитель может иметь несколько слотов для размещения мезонин-модулей и, следовательно, допускает гибкую функциональную конфигурацию (рис. 5 – 6).

CAN (Controller Area Network) – стандарт промышленной сети, ориентированный, прежде всего, на объединение в единую сеть различных исполнительных устройств и датчиков. Режим передачи – последовательный, широкополосный, пакетный.

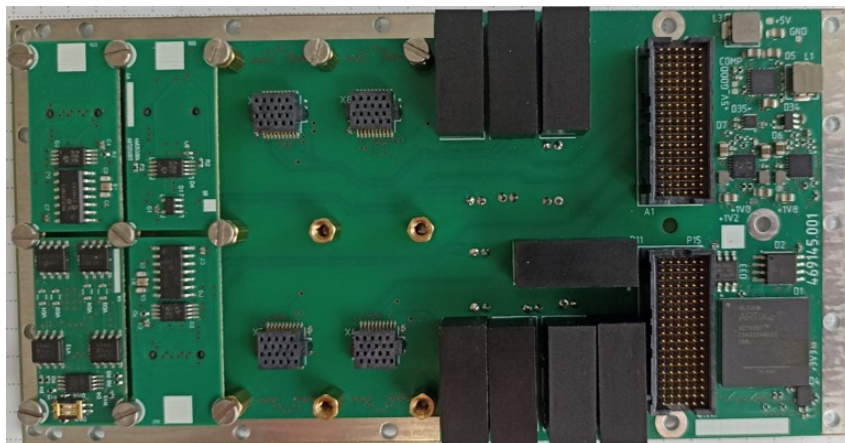


Рис. 5. Модуль интерфейсный универсальный (вид сверху)

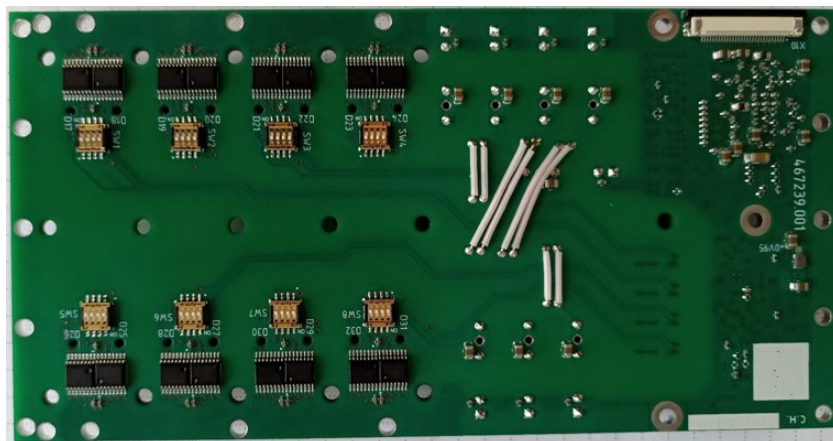


Рис. 6. Модуль интерфейсный универсальный (вид снизу)

Как пример, одной из микромезонинной плат является интерфейс CAN (рис. 7 – 8) [6].

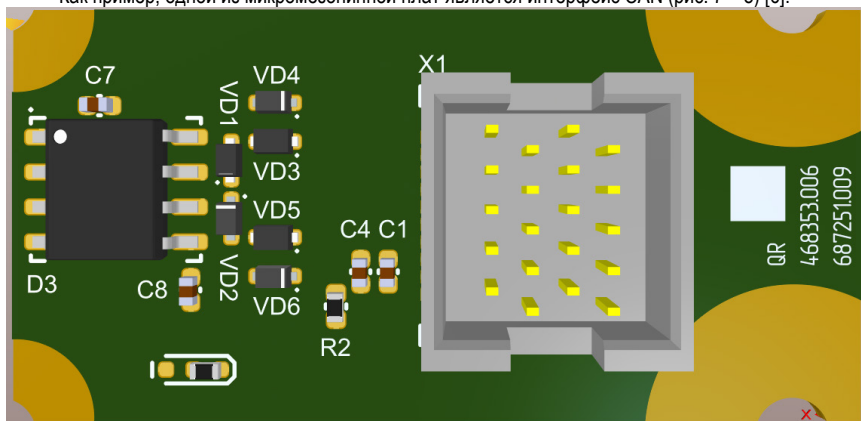


Рис. 7. 3D-вид микромезонинной платы CAN (вид сверху)

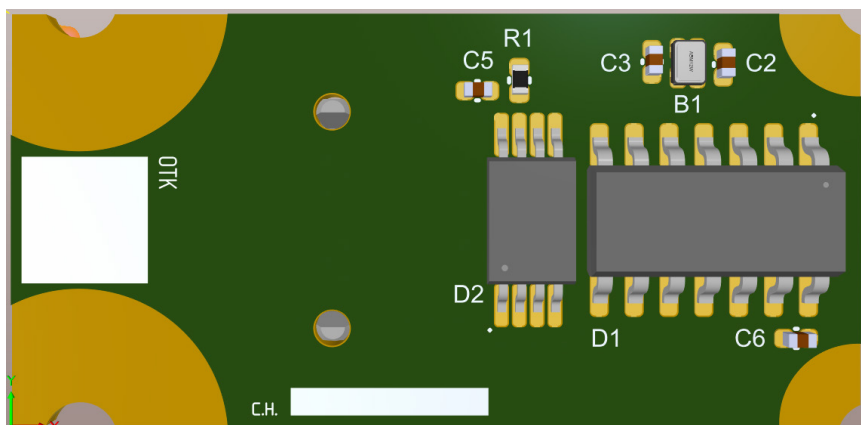


Рис. 8. 3D-вид микромезонинной платы CAN (вид снизу)

### Библиографический список

1. RS-232. URL: [https://www.kipis.ru/info/index.php?ELEMENT\\_ID=3078](https://www.kipis.ru/info/index.php?ELEMENT_ID=3078) (дата обращения: 17.05.2023).
2. CAN-шина. URL: <http://embeddedsystem.ru/index.php?page=can-bus-can-interface> (дата обращения: 18.05.2023).
3. Что такое RS-485. URL: <https://aveon.ru/services/006/> (дата обращения: 18.05.2023).
4. RS-422. URL: <https://vkmodule.com.ua/Description/Description2.html> (дата обращения: 18.05.2023).
5. GPIO для чайников. URL: <https://raspberrypi.ru/43-gpio-dlya-chaynikov-chast-1> (дата обращения: 18.05.2023).
6. Технологические возможности РЕЗОНИТ. URL: <https://www.rezonit.ru/directory/tehnologicheskie-osobennosti-proizvodstva/> (дата обращения: 18.05.2023).



УДК 635.3.035

**И. С. Шиманский**

студент кафедры информационно-сетевых технологий

**С. В. Мичурин** – заведующий кафедрой, доцент, кандидат технических наук – научный руководитель**АЛГОРИТМЫ АНАЛИЗА ФОТОДАННЫХ ДЛЯ ОПРЕДЕЛЕНИЯ РАСПОЛОЖЕНИЯ СКОТА**

Наиболее востребованные сейчас решения, связанные с отслеживанием животных во время выпаса на пастбищах и мониторинг и наблюдение за процессом отела у коров или других животных, где требуется непрерывное наблюдение, контроль и своевременное оповещение.

Решения на основе ИИ [1] определяют по камерам, какие именно животные, когда и как передвигаются, фиксируют все передвижения. Контролируют, чтобы коровы не ушли с пастбища. По фото с дрона могут быстро пересчитать животных, или найти оптимальное пастбище, где трава сочнее.

Современные овцеводческие хозяйства для удобства и облегчения рабочего процесса практикуют ведение номерного учета поголовья скота. Биркование, или маркировка, овец, как способ снабжения животных информационными знаками отличия, позволяет фермерам классифицировать стадо, оптимизировать процессы учета и убоя, своевременно проводить лечебно-профилактические меры. Процедуру по праву относят к главным организационным элементам скотоводческой сферы.

Зачастую при выпасе нескольких поголовий скота на одном пастбище их помечают, рисуя цифры или символы сбоку.



*Рис. 1. Нанесение метки на животное*

Но при использовании анализа фотоданных необходимо отделять своих овец от других. Для этого предлагается маркировать овец на спине, как на рис. 2, 3. Вариант на рис. 2 подходит больше для пастбищ, на которых пасутся несколько поголовий скота принадлежащих разным владельцам, или просто несколько разных поголовий. Разноцветные метки сверху, помогают определять к какому из поголовий они относятся и определять их движение.

На рис. 3 каждая овца имеет свой порядковый номер. Но в случае с использованием дрона для сбора фото- видеоматериала цифры могут плохо отображаться из-за качества съемки что в свою очередь может привести к ошибкам распознавания. Такие меры можно использовать для распознавания и разделения овец внутри помещений.



Рис. 2. Нанесение краски на стадо



Рис. 3. Идентификационные номера

Обучив нейросеть на фотографиях отмеченных овец, можно научить ее распознавать свое поголовье.

Наиболее перспективным направлением использования нейронных сетей в задачах распознавания скота является возможность использования небольшой компьютерной системы с функцией машинного зрения в составе беспилотного летательного аппарата [2]. Так как в таком случае качество фото/видео ограничено, из-за нестабильного положения дрона то особенно важна предварительная обработка изображений [3, 4, 5].

Есть возможность использовать полную сверточную сеть (FCN) для распознавания овец по изображениям с метками.

Для более обширных пастбищ есть вариант использования небольших самолетов.

Перспективным направлением исследований в этой отрасли является использование одно-платного компьютера. Но использование методов глубокого обучения на самом популярном одно-платном компьютере – Raspberry PI из-за ограниченной оперативной памяти и из-за низкой скорости процессора практически невозможно.

Решением может быть использование NVIDIA Jetson TX1 и TX2 – специальная платформа для вычислений нейронных сетей. Главный недостаток – высокая стоимость. Так же существует компания STMicroelectronics, которая сделала возможным использование методов глубокого обучения на микро-контроллерах. Они выпустили X-CUBE-AI пакет расширений для STM32CubeMX. Это расширение может работать с различными библиотеками глубокого обучения такими как Caffe, Keras, TensorFlow, ConvNetJS.

Для распознавания образов можно использовать несколько моделей.

### Single-Shot Detector (SSD)

В качестве входных данных модель принимает изображение, которое проходит через несколько сверточных слоев с различными размерами фильтров (10x10, 5x5 и 3x3). Карты объектов из сверточных слоев в разных положениях сети используются для прогнозирования ограничивающих рамок. Они обрабатываются специальными сверточными слоями с фильтрами 3x3, называемыми дополнительными слоями объектов, для создания набора ограничивающих рамок, подобных якорным рамкам Fast R-CNN.

Минусом данной модели является снижение степени точности при идентификации более мелких вещей.

### YOLO

Плюсами данного алгоритма являются:

- Скорость: Этот алгоритм повышает скорость обнаружения, поскольку он может прогнозировать объекты в режиме реального времени.
- Высокая точность: YOLO – это метод прогнозирования, который обеспечивает точные результаты с минимальными фоновыми ошибками.
- YOLO может обобщать изображение, не нагружая память обработки.

Минусы: YOLO страдает от значительно большего количества ошибок локализации и имеет проблемы с идентификацией ближайших предметов (рис. 4).

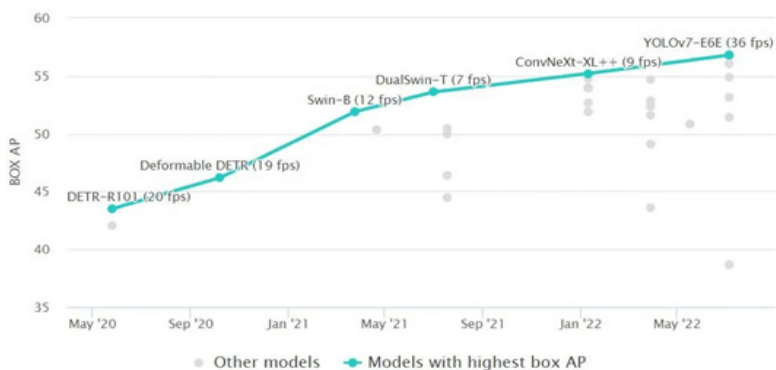


Рис. 4. Сравнительный анализ моделей

Так как в обнаружении объектов в реальном времени в тесте COCO (самом современном по средней точности) (рис. 4) лучшим алгоритмом считается YOLOv7 с результатом 3,5 мс на кадр, то лучшим выбором можно считать его.



**Библиографический список**

1. Распознавание образов с помощью искусственного интеллекта. URL: <https://habr.com/ru/articles/709432/> (дата обращения: 18.05.2023).
2. Russell J. A. A circumplex model of affect // Journal of Personality and Social Psychology. URL: [https://www.researchgate.net/publication/235361517\\_A\\_Circumplex\\_Model\\_of\\_Affect](https://www.researchgate.net/publication/235361517_A_Circumplex_Model_of_Affect) (дата обращения: 04.05.2023).
3. Программа Open CV. URL: <https://opencv.org/opencv-4-5-4> (дата обращения: 19.05.2023).
4. Программное обеспечение FaceReader. URL: <https://www.noldus.com/facereader> (дата обращения: 18.05.2023).
5. Object Detection. Распознавай и властвуй. Ч. 2. URL: <https://habr.com/ru/companies/jetinfosystems/articles/498652/> (дата обращения: 18.05.2023).

УДК 62-5

**В. А. Васильев**

студент кафедры компьютерных технологий и программной инженерии

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

**АПРОБАЦИЯ РЕЛЕЙНОГО УПРАВЛЕНИЯ РОБОТОМ-МАНИПУЛЯТОРОМ  
В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ**

**Введение**

Работа посвящена изучению математической модели релейного управления роботом-манипулятором. За основу взят манипулятор, состоящий из трех звеньев, имеющих соединения, допускающие вращательное движение. Третье звено может представлять собой ковш экскаватора, который может дополнительно включать в себя грунт неизвестной массы. В качестве входных параметров модели могут выступать длины и массы трех звеньев, а также координаты или углы между звеньями, описывающие его начальное положение.

Актуальность работы заключается в распространенности использования роботоманипуляторов в промышленности, вынужденных функционировать в условиях неполно определенных. Это могут быть сварочные работы, сборка различных деталей на заводе, погрузка и разгрузка транспортных средств на складе, железной дороге или в порту. Также робот-манипулятор может использоваться в ходе бурильных работ и т. д.

**Структура объекта и постановка задачи управления**

Схема моделируемого робота-манипулятора приведена на рис. 1, где OA, AB и BC – соответствующие звенья.

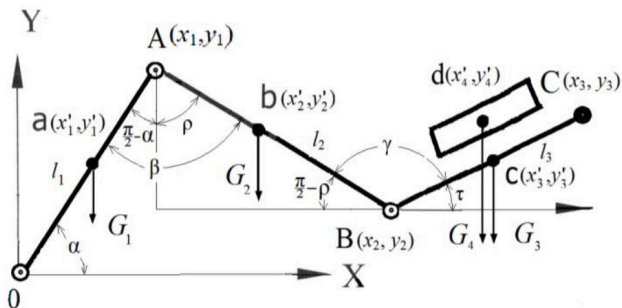


Рис. 1. Схема робота-манипулятора

Предполагается, что известны координаты вершин звеньев A, B, C или углы  $\alpha$ ,  $\beta$ ,  $\gamma$  между соответствующими звеньями. Между этими величинами имеет место взаимно однозначное соответствие:

$$\begin{aligned} x_1 &= l_1 \cos(\alpha), y_1 = l_1 \sin(\alpha), \\ x_2 &= x_1 + l_2 \sin(\rho), y_2 = y_1 - l_2 \cos(\rho), \\ x_3 &= x_2 + l_3 \cos(\tau), y_3 = y_2 + l_3 \sin(\tau), \end{aligned}$$

где

$$\rho = \alpha + \beta - \frac{\pi}{2}, \tau = -\gamma + \rho + \frac{\pi}{2} = \alpha + \beta - \gamma.$$

Если заданы координаты точек, то

$$\alpha = \arcsin\left(\frac{y_1}{l_1}\right), \beta = \frac{\pi}{2} - \alpha + \arccos\left(\frac{y_1 - y_2}{l_2}\right), \gamma = \alpha + \beta - \arcsin\left(\frac{y_3 - y_2}{l_3}\right).$$

Обозначим  $\varphi = [\alpha, \beta, \gamma]$ ,  $z = [x_1, x_2, x_3, y_1, y_2, y_3]$ . Тогда вращательное движение вокруг точек  $O$ ,  $A$  и  $B$  при отсутствии неизвестной массы описывается следующим уравнением

$$J(z)\ddot{\varphi} = M - R(z),$$

где  $J$  – вектор моментов инерции,  $M$  – вектор управляющих моментов,  $R$  – вектор весовых моментов.

Рассмотрены две задачи управления – полноопределенная и недоопределенная с неизвестной массой 3-го звена.

**Задача 1.** Определенная задача. На интервале времени  $(0, t_k)$  найти такие управляющие моменты  $M_i$ , при которых система

$$\ddot{\varphi}_i = \frac{M_i - R_i(z)}{J_i(z)}, \quad i=1,2,3, \quad (1)$$

из начального неподвижного состояния  $\varphi(0)=\varphi_0, \dot{\varphi}(0) = 0$ , переходит в другое неподвижное состояние  $\varphi(t_k)=\varphi^*, \dot{\varphi}(t_k) = 0$ .

**Задача 2.** Неопределенная задача. На интервале времени  $(0, t_k)$  найти такие управляющие моменты  $M_i$ , при которых система

$$\ddot{\varphi}_i = \frac{M_i - R_i(z) - q\Delta R_i(z)}{(J_i(z) + q\Delta J_i(z))}, \quad i=1,2,3. \quad (2)$$

из начального неподвижного состояния  $\varphi(0)=\varphi_0, \dot{\varphi}(0) = 0$ , переходит в другое неподвижное состояние  $\varphi(t_k)=\varphi^*, \dot{\varphi}(t_k) = 0$  или в его некоторую окрестность.

В (2) смысл константы  $q$  следующий. Если неизвестную массу 3-го звена (например, зачерпнутого ковшем грунта) обозначить как  $m_4$ , то ее можно представить в виде  $m_4 = q m_0$ , где  $m_0$  – максимальная масса грунта, которую может зачерпнуть ковш,  $q$  – неизвестная константа, лежащая в интервале  $0 \leq q \leq 1$ .

Замечание. Подробный вывод систем (1), (2) получен в работах [1, 2].

### Основные формулы управления

Пусть плотность звеньев постоянна, значит центры тяжести находятся в центрах звеньев – соответственно точки  $a, b$  и  $c$ . Для нахождения моментов инерции и весовых моментов можно воспользоваться следующими формулами (подробнее см. [1]):

$$J_1(z) = m_1(x_1'^2 + y_1'^2) + m_2(x_2'^2 + y_2'^2) + m_3(x_3'^2 + y_3'^2),$$

$$J_2(z) = m_2\left((x_2' - x_1')^2 + (y_2' - y_1')^2\right) + m_3\left((x_3' - x_1')^2 + (y_3' - y_1')^2\right),$$

$$J_3(z) = m_3\left((x_3' - x_2')^2 + (y_3' - y_2')^2\right),$$

$$R_1(z) = gm_1x_1' + gm_2x_2' + gm_3x_3',$$

$$R_2(z) = gm_2(x'_2 - x_1) + gm_3(x'_3 - x_1),$$

$$R_2(z) = gm_3(x'_3 - x_2).$$

Предполагается также, что известно время  $t_k$ , необходимое для перемещения из начального положения в конечное. Поскольку в конечном положении манипулятор стоит неподвижно, значит угловая скорость в момент  $t_k$  равна нулю. Значит первую часть пути звенья двигаются с ускорением, а затем замедляются. Допустим  $t_1 = t_k/2$ . В таком случае угловое ускорение можно найти как

$$\ddot{\varphi} = \frac{\varphi_k - \varphi_0}{t_1^2},$$

где  $\varphi_k$  – вектор углов, описывающих конечное положение, а  $\varphi_0$  – начальное.

Теперь можем найти управляющие моменты:

$$M = J(z)\ddot{\varphi} + R(z).$$

При добавлении неизвестной массы к третьему звену, предлагается использовать адаптивное управление, блок-схема которого приведена на рис. 2.

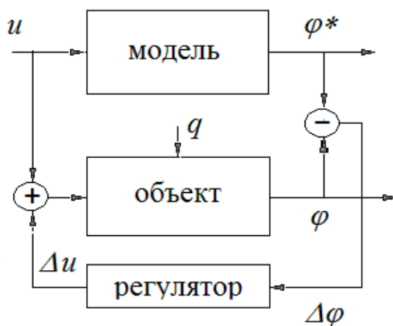


Рис. 2. Блок-схема адаптивного управления

Происходит это следующим образом: на вход объекту подается управление, которое приводит к отклонению соответствующего звена на угол  $\varphi$ . Ожидаемое отклонение модели равно  $\varphi^*$ . Отличие заключается в воздействии на объект неизвестной массы  $m_0$ . На выходе имеем  $\Delta\varphi$  – отклонение положения звена от ожидаемого, которое нужно учесть на следующем шаге. Таким образом получаем  $\Delta u$  – корректировку для управления. Далее цикл повторяется.

Соответственно, в системе меняется момент инерции и весовой момент. Теперь вращательное движение описывается следующим уравнением:

$$\ddot{\varphi} = \frac{M - R(z) - \Delta R(z)}{J(z) + \Delta J(z)},$$

где

$$\Delta J_1(z) = m_0(x_4'^2 + y_4'^2),$$

$$\Delta J_2(z) = m_0\left((x_4' - x_1)^2 + (y_4' - y_1)^2\right),$$

$$\Delta J_3(z) = m_0 \left( (x_4' - x_2)^2 + (y_4' - y_2)^2 \right),$$

$$\Delta R_1(z) = gm_0 x_4', \Delta R_2(z) = gm_0 (x_4' - x_1), \Delta R_3(z) = gm_0 (x_4' - x_2).$$

**Численное моделирование и обсуждение результатов**

Для моделирования и последующей визуализации будем изменять углы  $\alpha, \beta, \gamma$  на каждом шаге  $t_i$ . Сначала опишем движение робота-манипулятора следующей формулой (подробнее см. [2]):

$$\varphi(t) = \begin{cases} \varphi_0 + v \frac{t^2}{2} & \text{для } 0 < t < t_1, \\ \varphi_0 + v \left( 2t_1 t - \frac{t^2}{2} - t_1^2 \right) & \text{для } t_1 < t < t_k, \end{cases}$$

где  $v$  – угловое ускорение.

Перепишем формулу в рекурсивном виде:

$$\varphi(t) + f(t) = \varphi(t+1),$$

где  $f$  – искомая добавка к текущему значению угла на каждом шаге.

$$f(t) = \varphi(t+1) - \varphi(t),$$

$$f(t) = \begin{cases} \varphi_0 + v \frac{(t+1)^2}{2} - \varphi_0 - v \frac{t^2}{2} & \text{для } 0 < t < t_1, \\ \varphi_0 + v \left( 2t_1(t+1) - \frac{(t+1)^2}{2} - t_1^2 \right) - \varphi_0 - v \left( 2t_1 t - \frac{t^2}{2} - t_1^2 \right) & \text{для } t_1 < t < t_k, \end{cases}$$

$$f(t) = \begin{cases} v \left( t + \frac{1}{2} \right) & \text{для } 0 < t < t_1, \\ v \left( 2t_1 - t - \frac{1}{2} \right) & \text{для } t_1 < t < t_k. \end{cases}$$

Начальные положения манипуляторов с грузом и без него совпадают, затем из-за дополнительного веса груза с тем же управляющим моментом угловое ускорение изменится, будем его считать по следующей формуле:

$$\ddot{\varphi} = \frac{(\ddot{\varphi}_0 J(z) + R(z)) - R(z) - \Delta R(z)}{J(z) + \Delta J(z)},$$

$$\ddot{\varphi} = \frac{\ddot{\varphi}_0 J(z) - \Delta R(z)}{J(z) + \Delta J(z)},$$

где  $\ddot{\varphi}_0$  – угловые ускорения робота-манипулятора без груза.

Корректировкой к изменению углов на каждом шаге будет разность значений углов робота-манипулятора без груза и с ним на предыдущем шаге.

Визуализация описанного выше процесса показана на рис. 3 и 4.

Исходные данные: длины звеньев:  $l_1 = 1,2$  м,  $l_2 = 1,0$  м,  $l_3 = 0,8$  м; массы звеньев:  $m_1 = m_2 = m_3 = 100$  кг; масса груза:  $m_0 = 20$  кг. Начальные значения углов:  $\varphi(0) = [\pi, \pi, \pi]$ .

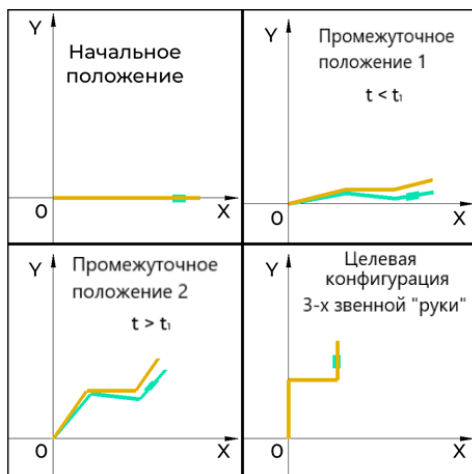


Рис. 3. Визуализация движения робота-манипулятора

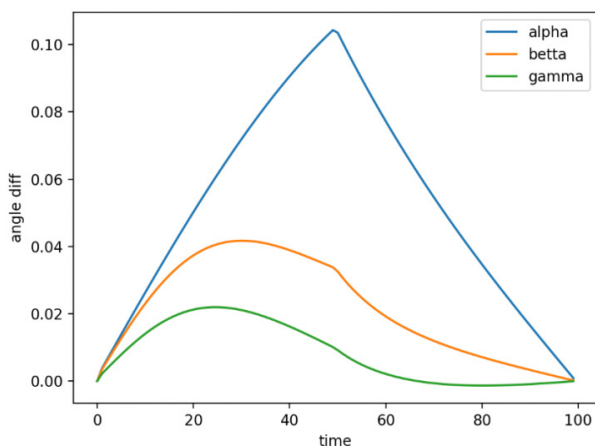


Рис. 4. Разность значений углов

Конечные значения углов:  $\varphi(t_k) = [\pi/2, \pi/2, \pi/2]$ .

Из рис. 3 следует, что дополнительная масса груза приводит к отклонению манипулятора (показан зеленым цветом) от ожидаемого положения (показано оранжевым цветом). Постепенно разрыв увеличивается, пока ожидаемое положение манипулятора не достигает соответствующего положения в момент времени  $t_1$ . Тогда движение ожидаемого положения манипулятора начинает замедляться и положение манипулятора с дополнительной массой постепенно уменьшает разрыв.

Это же мы видим и на рис. 4, который показывает зависимость отклонения положения манипулятора с дополнительной массой от ожидаемого в каждый момент времени  $t$ . Здесь  $t_1 = 50$ . Также можно отметить, что самое большое отклонение наблюдается по углу  $\alpha$ , так как на первое звено вес груза

оказывает большее влияние из-за большего плеча воздействия силы тяжести по сравнению с остальными звеньями.

### **Заключение**

Решена задача визуализации релейного управления переводом «руки» 3-х звенного манипулятора из одного неподвижного положения в другое неподвижное в условиях неопределенности. Незвестным параметром здесь является масса 3-го звена, прозрачно интерпретируемого как масса зачерпнутого ковшом грунта. Приведены результаты численного моделирования. Используемый метод к решению задач управления роботами-манипуляторами может быть распространен на другие виды неопределенностей: неточное измерение текущих координат звеньев, воздействие внешних случайных воздействий.

### **Библиографический список**

1. Параев Ю. И., Колесникова С. И., Цветницкая С. А. Управление роботом-манипулятором в условиях неопределенности // Вестник Томского государственного университета. 2021. № 57. С. 4–12.
2. Параев Ю. И., Колесникова С. И., Цветницкая С. А. Новый подход к решению задачи управления роботом-манипулятором // Вестник Томского государственного университета. 2021. № 55. С. 4–10.

УДК 004.942

**Е. М. Глушкова**

студент кафедры компьютерных технологий и программной инженерии

**М. Д. Поляк** – старший преподаватель – научный руководитель

## АНАЛИЗ ПОВЕДЕНИЯ БИОЛОГИЧЕСКОГО ОБЪЕКТА НА ОСНОВЕ МОДЕЛИ С ВНУТРИВИДОВОЙ КОНКУРЕНЦИЕЙ

### Введение

Несмотря на огромное разнообразие биологических систем, возможно выделить перечень важнейших качественных свойств, присущих им: рост, самоограничение роста, существование нескольких стационарных режимов и способность к переключению между ними, биоритмы (автоколебательные режимы), неоднородность в пространстве и квазистохастичность [1]. Сравнительно простые нелинейные динамические модели способны продемонстрировать все эти свойства [2].

Среди методов системной и математической биологии можно выделить моделирование биологических систем, позволяющее устанавливать более глубокие и сложные взаимосвязи между биологической теорией и опытом [3].

### Используемая математическая модель

В данной работе будет рассматриваться математическая модель Лотки-Вольтерра, а именно, модель популяционной динамики, описывающей взаимодействия между видами хищника и жертвы [4] в долгосрочной перспективе с учетом стабилизирующих факторов внутривидовой конкуренции каждой популяции [5]:

$$\begin{cases} \dot{x}_1 = \alpha_1 x_1 - \frac{x_1 x_2}{1 + \beta x_1} - \varepsilon x_1^2 \\ \dot{x}_2 = -\alpha_2 x_2 + \frac{x_1 x_2}{1 + \beta x_1} - \delta x_2^2 \end{cases} \quad (1)$$

где  $x_1$  – плотность популяции жертв,  $x_2$  – плотность популяции хищников,  $\alpha_1 > 0$  – скорость естественной смертности жертв,  $\alpha_2 > 0$  – скорость естественной смертности хищников,  $0 < \beta < 1$  – коэффициент, характеризующий насыщение хищников. Данная модель учитывает три фактора взаимодействия [6]:

$-\varepsilon x_1^2$  – внутривидовая конкуренция жертв,

$-\delta x_2^2$  – внутривидовая конкуренция хищников за отличные от жертв ресурсы,

$\frac{x_1 x_2}{1 + \beta x_1}$  – взаимодействие популяций, описываемое трофической функцией Холлинга второго

типа [7].

Важно помнить, что система будет иметь биологический смысл, только если  $x_1$  и  $x_2$  неотрицательные.

### Алгоритм непрерывной адаптации для неслучайного возмущения

Для дальнейшего изучения, введем алгоритм непрерывной адаптации для неслучайного возмущения (далее в тексте NAD) [8] для начальной системы:

$$\begin{cases} \frac{dx_1}{dt} = \alpha_1 x_1 - \frac{x_1 x_2}{1 + \beta x_1} - \varepsilon x_1^2 \\ \frac{dx_2}{dt} = -\alpha_2 x_2 + \frac{x_1 x_2}{1 + \beta x_1} - \delta x_2^2 \end{cases} \quad (2)$$



Рассмотрим систему с управлением по цели  $\psi_1 = x_1 - x_1^* \xrightarrow{t \rightarrow \infty} 0$ , где задачей управления является достижение видом  $x_i$  целевой численности  $x_i^*(t)$ , где  $x_1^*$  удобнее всего представить константой, не зависящей от времени:  $x_1^* = \text{const}$ .

После введения управления система примет вид:

$$\begin{cases} \dot{x}_1 = \alpha_1 x_1 - \frac{x_1 x_2}{1 + \beta x_1} - \varepsilon x_1^2 + u + \xi \\ \dot{x}_2 = -\alpha_2 x_2 + \frac{x_1 x_2}{1 + \beta x_1} - \delta x_2^2 \end{cases} \quad (3)$$

После синтеза системы управления:

$$\begin{cases} \dot{x}_1 = \alpha_1 x_1 - \frac{x_1 x_2}{1 + \beta x_1} - \varepsilon x_1^2 + u + z \\ \dot{x}_2 = -\alpha_2 x_2 + \frac{x_1 x_2}{1 + \beta x_1} - \delta x_2^2 \\ \dot{z} = \eta \psi_1 \end{cases} \quad (4)$$

где  $\dot{z} = \eta \psi_1 \Rightarrow z = \eta \int \psi_1 dt$  и является внутренней дополнительной фазовой переменной, на которую было переведено внешнее возмущение, а  $\eta$  – коэффициент пропорциональности. Начнем поиск  $z$ :

$$\begin{aligned} \psi_1 &= x_1 - x_1', \\ \psi_{end} &= \psi_1 + kz \rightarrow 0, \\ \psi_1 &= -kz, \\ \dot{z} &= -\eta kz, \\ \frac{dz}{dt} &= -\eta kz, \\ \frac{dz}{z} &= -\eta k dt, \\ \int \frac{dz}{z} &= -\eta k \int dt, \\ \ln|z| &= -\eta kt + C, \\ z &= e^{-\eta kt + C} \Rightarrow z = Ce^{-\eta kt}, \eta, k > 0. \end{aligned}$$

Решив уравнения Эйлера-Лагранжа относительно  $\psi_{end}$ , выведем управление для NAD-алгоритма:

$$\begin{aligned} T_1(\psi_1 + kz)' &= -\psi_{end}, \\ \psi_1' + kz' &= \frac{-\psi_{end}}{T_1}, \\ f_1 + u + z + kz &= \frac{-\psi_{end}}{T_1}, \\ T_1 \dot{\psi}_{end} + \psi_{end} &= 0, \\ u &= -f_1 - z - kz - \frac{\psi_{end}}{T_1}, \end{aligned}$$

$$u = -f_1 - z - k\eta\psi_1 - \frac{\psi_{end}}{T_1}.$$

Итоговая система:

$$\left\{ \begin{array}{l} \dot{x}_1 = \alpha_1 x_1 - \frac{x_1 x_2}{1 + \beta x_1} - \varepsilon x_1^2 + u + \xi \\ \dot{x}_2 = -\alpha_2 x_2 + \frac{x_1 x_2}{1 + \beta x_2} - \delta x_2^2 \\ \dot{z} = \eta \psi_1 \\ u = z \left( k^2 \eta - 1 \right) - f_1 - \frac{\psi_{end}}{T_1} \\ \psi_{end} = \psi_1 + kz \\ \psi_1 = x_1 - x_1^* \\ f_1 = \alpha_1 x_1 - \frac{x_1 x_2}{1 + \beta x_1} - \varepsilon x_1^2, \end{array} \right. \quad (5)$$

где  $\xi(t)$  – независимая псевдослучайная функция,  $T_1$  – параметр дискретизации.

С помощью этого же алгоритма получим систему с управлением по  $\psi_2 = x_1 - \rho x_2 \xrightarrow{t \rightarrow \infty} 0$ , задачей управления является поддержание постоянной пропорции между численностями видов  $x_1$  и  $x_2$ . Итоговая система будет выглядеть следующим образом:

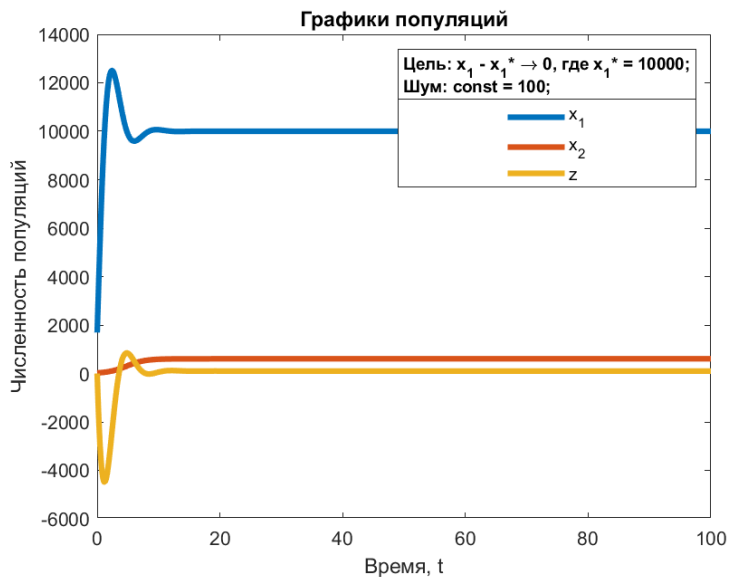
$$\left\{ \begin{array}{l} \dot{x}_1 = \alpha_1 x_1 - \frac{x_1 x_2}{1 + \beta x_1} - \varepsilon x_1^2 + u + \xi \\ \dot{x}_2 = -\alpha_2 x_2 + \frac{x_1 x_2}{1 + \beta x_2} - \delta x_2^2 \\ \dot{z} = \eta \psi_1 \\ u = z \left( k^2 \eta - 1 \right) - f_1 - \frac{\psi_{end}}{T_1} \\ \psi_{end} = \psi_1 + kz \\ \psi_1 = x_1 - x_1^* \\ f_1 = \alpha_1 x_1 - \frac{x_1 x_2}{1 + \beta x_1} - \varepsilon x_1^2, \end{array} \right. \quad (6)$$

где  $\rho$  – постоянная пропорция между популяциями.

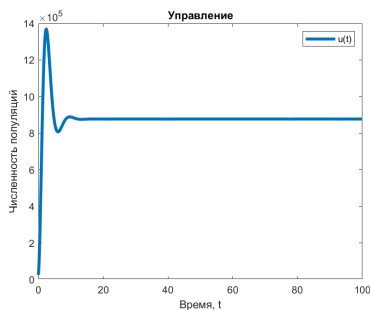
### Графическое моделирование

Основной целью работы является изучение устойчивости управления к разным видам неслучайного шума [9]. Для получения более прикладных результатов параметры модели будут выступать коэффициенты существующего исследования  $\alpha_1 = 32$ ,  $\beta = 0.21$ ,  $\varepsilon = 0.0087$ ,  $\alpha_2 = 60$ ,  $\delta = 0.0011$  и начальные значения  $x_1 = 1689$  и  $x_2 = 24$  [10]. Для системы с управлением по цели  $\psi_1$  константа  $x_1^* = 10000$ , а для управления по цели  $\psi_2$  постоянная  $\rho = 10$ .

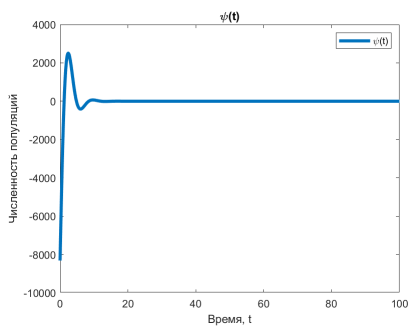
Для первой исследуемой функции шума, являющейся константой и равна  $\xi(t) = 100$ , рассмотрим вид системы (рис. 1) с управлением по первой цели  $\psi_1$ .



а)



б)



в)

Рис. 1. Графики: а – зависимости численности популяций от времени и фазовой переменной, на которую было переведено внешнее возмущение; б – введенного управления; в – макропеременной

Можно заметить, что цель достигается как на графике популяций, становясь равной 10000, так и на графике самой цели становясь равной нулю. Управление также стабилизируется – из чего следует вывод, что система устойчива.

Для той же исследуемой функции шума  $\xi(t) = 100$  рассмотрим вид системы (рис. 2) с управлением по второй цели  $\psi_2$ .

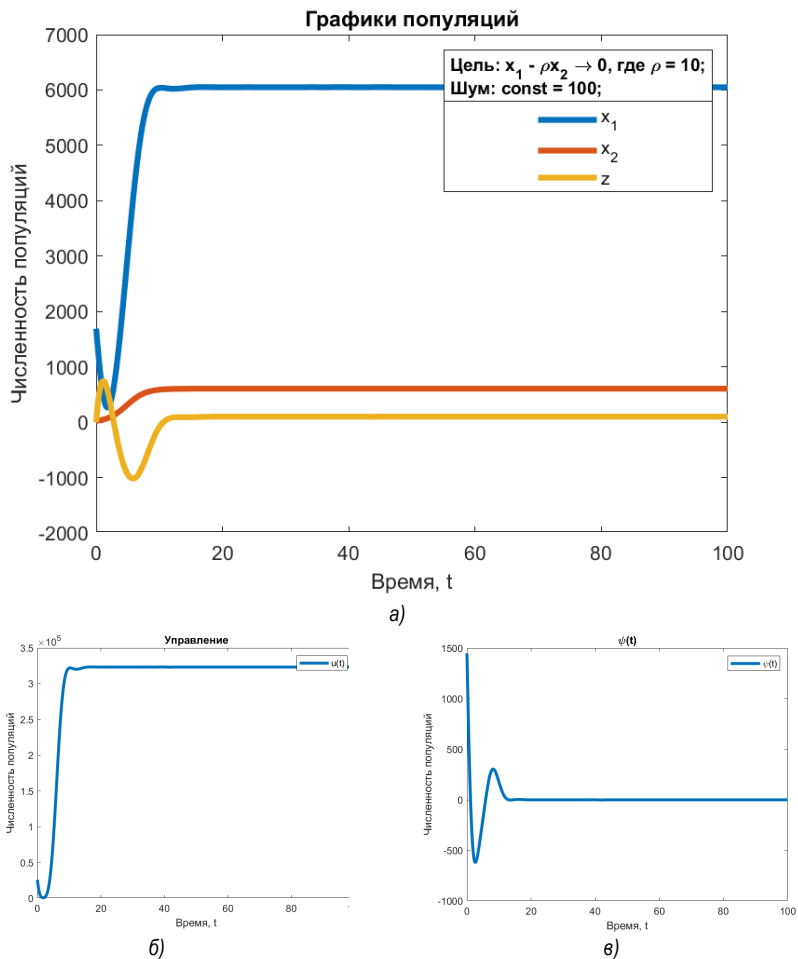


Рис. 2. Графики: а – зависимости численности популяций от времени и фазовой переменной, на которую было переведено внешнее возмущение; б – выведенного управления; в – макропеременной

В этом случае также можно заметить, что цель достигается как на графике популяций, так и на графике самой цели, как и управление – стабилизируется.

Следующий исследуемый внешний шум – гармонические колебания, имеющие уравнение  $\xi(t) = A(\omega t + \varphi)$ , где  $A$  – амплитуда колебаний,  $\varphi$  – начальная фаза колебаний, а  $\omega$  – циклическая частота. Начальная фаза колебаний не рассматривается в исследовании, так как влияет только положение графика на оси абсцисс. При  $\xi(t) = 10000\sin(t + \varphi)$  с управлением по цели  $\psi_1$  примет вид (рис. 3).

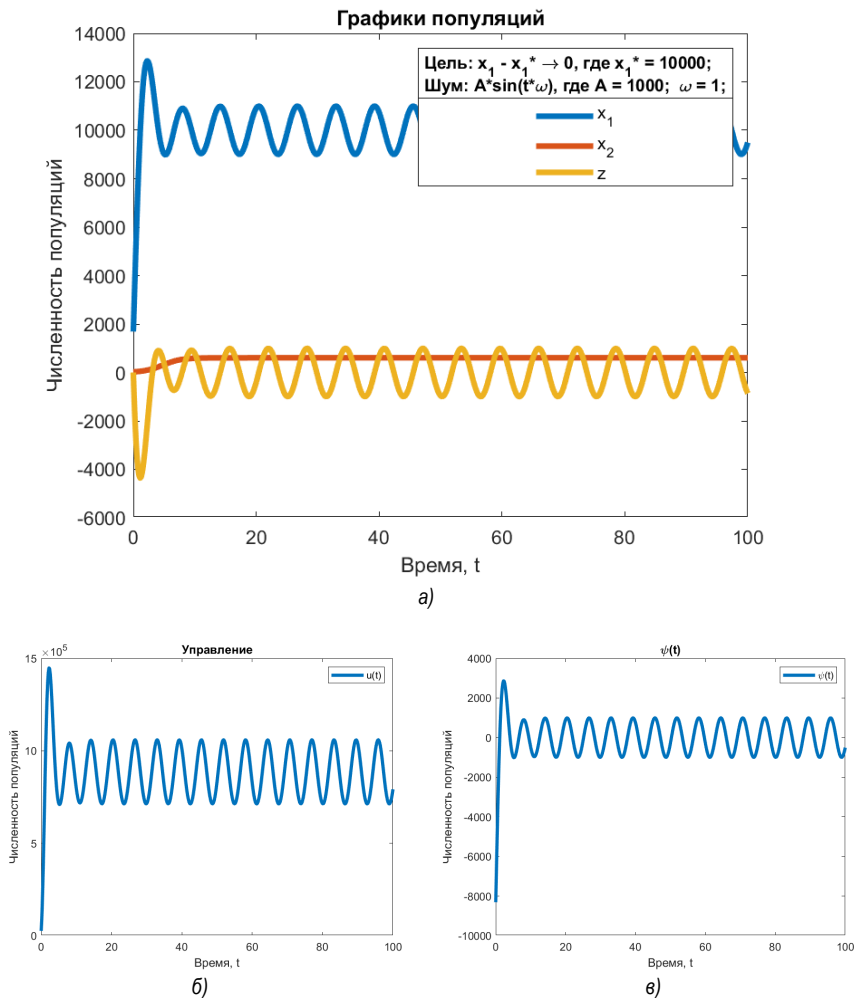


Рис. 3. Графики: а – зависимости численности популяций от времени и фазовой переменной, на которую было переведено внешнее возмущение; б – введенного управления; в – макропеременной

Несмотря на разброс макропеременной можно увидеть, что система приходит к равновесию, так как график популяции  $x_1$  колеблется вокруг целевого значения, как и графики управления и цели.

Также рассмотрим внешний шум, имеющий нормальное распределение, определяемое переменными  $\mu$  и  $\sigma$ , где  $\mu$  – математическое ожидание, которое будем считать равным 0, а  $\sigma$  – среднеквадратическое отклонение. При  $\sigma=1000$  для управления по  $\psi_2$  цели получим результаты (рис. 4).

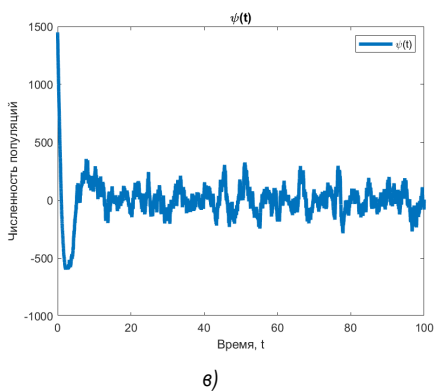
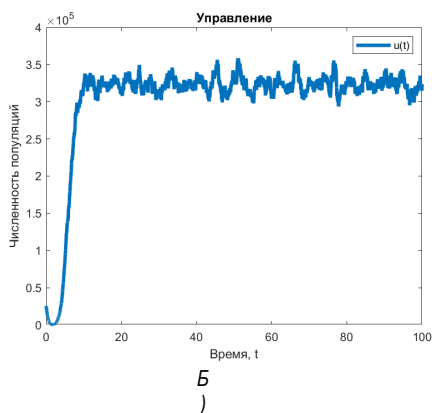
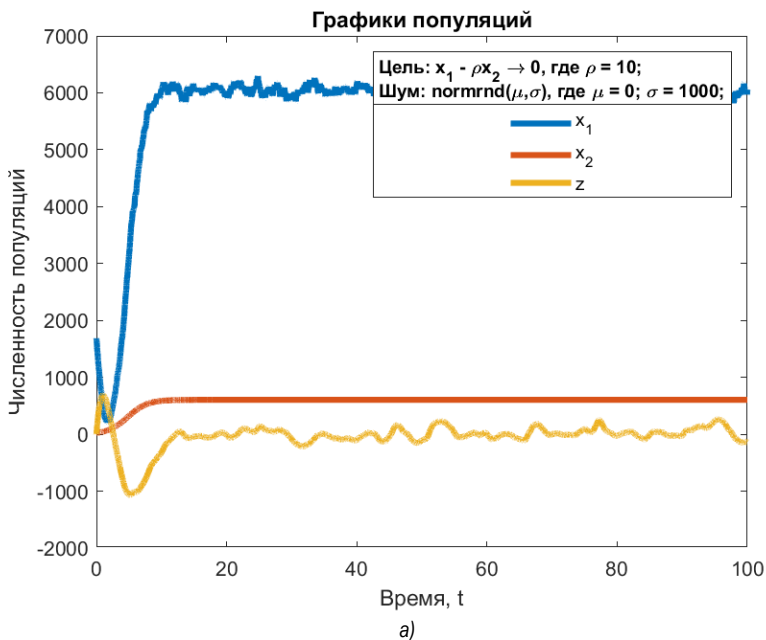


Рис. 4. Графики: а – зависимости численности популяций от времени и фазовой переменной, на которую было переведено внешнее возмущение; б – выведенного управления; в – макропеременной

Наибольшая отзывчивость колебания макропеременной пришлось на изменение  $A$  и  $\sigma$ . При увеличении амплитуды гармонических колебаний дисперсия экспоненциально растет при управлении по обоим целям (рис. 5).

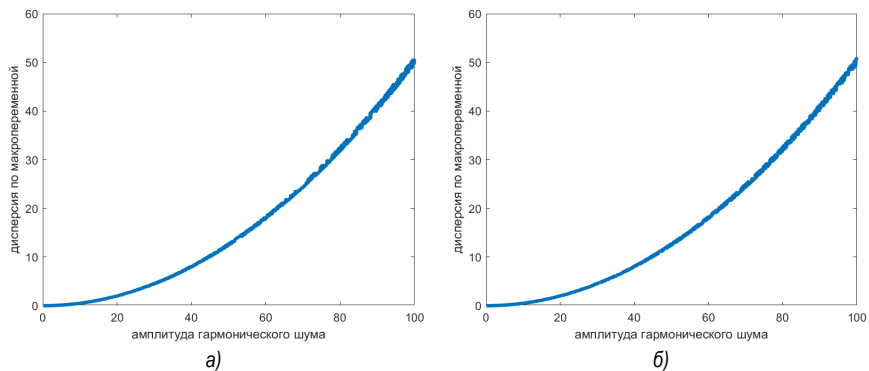


Рис. 5. Графики зависимости амплитуды гармонического шума от дисперсии по макропеременной при управлении: а – по первой цели; б – по второй цели

При увеличении среднеквадратического отклонения данные приняли вид (рис. 6).

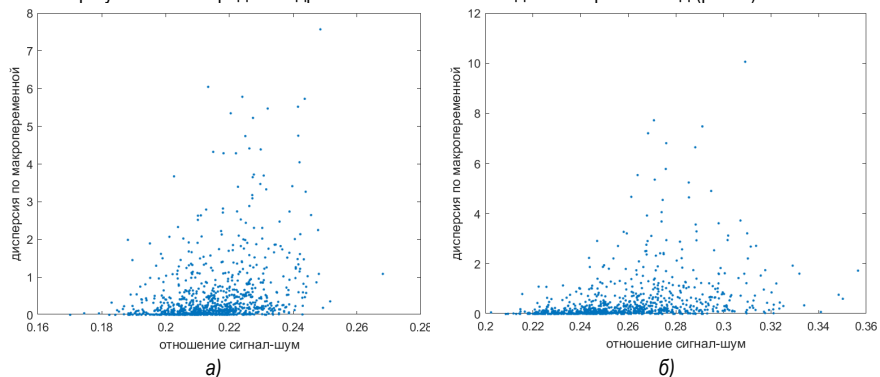


Рис. 6. Графики для отношения сигнал-шум при управлении: а – по первой цели; б – по второй цели

### Заключение

В ходе работы была исследована математическая модель Лотки-Вольтерра с учетом стабилизирующих факторов внутривидовой конкуренции каждой популяции и было выведено NAD-управление по двум целям, изучено его устойчивость к разным видам шума – постоянному, гармоническими колебаниям и псевдослучайному. Также было выполнено графическое моделирование для разных параметров полученной системы.

### Библиографический список

1. Ризниченко Г. Ю. Математические модели в биофизике и экологии. М.; Ижевск: Институт компьютерных исследований, 2003. 184 с.

2. Основания синергетики. / Е. Н. Князева, С. П. Курдюмов. СПб.: Алетейя, 2002. 414 с.
3. Агрофизика. / Е. В. Шейн, В. М. Гончаров; МГУ им. М. В. Ломоносова. М.: Феникс, 2006. 400 с.
4. Динамические режимы стохастической модели «хищник – жертва» с учетом конкуренции и насыщения / Е. П. Абрамова, Т. В. Рязанова // Компьютерные исследования и моделирование. 2019. Т. 11. № 3. С. 515–531.
5. Базыкин А. Д. Нелинейная динамика взаимодействующих популяций. Ижевск: Институт компьютерных исследований, 2003. 368 с.
6. Computer Research and Modeling. 2019. Vol. 11. No. 3. Pp. 515–531.
7. Тютюнов Ю. В., Сенина И. Н. Трофическая функция как результат активного пространственного поведения хищника // Проблемы проектирования и управления экономическими системами: инвестиционный аспект. Ростов-на-Дону: РГЭА, 1998. С. 132–135.
8. Noise in ecosystems: a short review / B. Spagnolo, D. Valenti, A Fiasconaro // Math Biosci Eng. 2004. No 1 (1). Pp. 185–211.
9. Lotka Volterra Model Simulation for Rice-field Rat and Tyto Alba Owls in Sumpiuh District, Banyumas Regency, Central Java / Reza Kusuma Nurrohman1, A. Ardiansyah, Bayu Dwi Apri Nugroho // Jurnal Agritech. No 39 (4). 2019. 323 p.



УДК 519.688

**А. А. Горбунов**

студент кафедры компьютерных технологий и программной инженерии

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

## FRIS-ФУНКЦИЯ В ЗАДАЧЕ КЛАСТЕРИЗАЦИИ

### Введение

На данный момент разработано большое количество разных методов в области анализа данных, но из-за специфичности моделей их нельзя применить к большому числу прикладных задач. Из-за этой особенности появилась задача создать такую модель, чтобы она, по возможности, охватывала весь спектр решаемых задач в области анализа данных. В качестве такой модели Н. Г. Загоруйко предложил FriS-функцию [1].

### Меры сходства

Все подходы в статистике основываются на способе измерения сходства между объектами [2]. Но вот в чем их проблема. Если обратиться к формулам (1) и (2), то можно заметить, что сходства между объектами  $a$  и  $b$  зависят только от этих двух объектов. FriS-функция позволяет принимать решение не по абсолютному значению расстояний, как бы оно не вычислялось, а по тому, в сравнении с чем происходит сравнение [3]:

$$FS_1(a,b) = 1 - \sqrt{\sum_{i=1}^n a_i (x_i^a - x_i^b)^2} \quad (1)$$

$$FS_1(a,b) = 1 - \sum_{i=1}^n a_i |x_i^a - x_i^b| \quad (2)$$

### FriS-функция

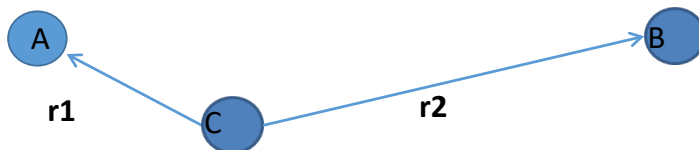


Рис. 2. Пример использования FriS – функции

В качестве примера возьмем три объекта А, В и С. Чтобы определить принадлежность класса объекта С необходимо измерить расстояние до объекта А и до объекта В, а затем применить формулу (3).

$$F = \frac{(r_2 - r_1)}{(r_2 + r_1)} \quad (3)$$

FriS-функция принимает значения от [-1; 1]. В данном случае, если значение функции  $F = 1$ , то объект С совпадает с объектом А, т. е.  $r_1 = 0$ . Если же  $F = -1$ , объект С совпадает с объектом В, т. е.

$r_2 = 0$ . В случае, когда  $F = 0,5$ , можно говорить о том, что объект  $C$  равнозначно можно отнести к объекту  $A$  или объекту  $B$ , или же не отнести.

### Сравнение с другими метриками

FRiS-функция является относительной мерой сходства. Такие меры применялись и раньше. Например, классическое правило Байеса, где контрольный объект относится к классу  $A$ , если вероятность принадлежности к  $A$  больше, чем к  $B$ . Метод  $kNN$ , где вычисляется среднее расстояние до объектов и принимается решение в пользу наименьшего. Также есть Relief и Ширина профиля, которые, как и FRiS-функция, являются количественными мерами. В табл. 1 приведены сравнения метрик на двух выборках.

Таблица 1

### Сравнение метрик

N	V1			V2		
	50	100	150	50	100	150
FRiS	0,930308	0,938042	0,939497	0,818071	0,866152	0,866458
SW	0,924218	0,937614	0,943244	0,655248	0,690778	0,702515
kNN	0,895255	0,894467	0,906625	0,735635	0,727285	0,80093
Fisher	0,935972	0,943948	0,942755	0,598515	0,60059	0,605303
RELIEF	0,899302	0,932148	0,930718	0,710267	0,754834	0,791442

### Заключение

В ходе сравнение различных критериев для выбора информативных признаков. FRiS-критерий сравнивался с такими критериями выбора признаков как, Silhouette Width (SW), ошибка скользящего контроля по правилу ближайшего соседа (NN), критерий Фишера (Fisher) и RELIEF. Сравнение осуществлялось на двух сериях выборок. В первом случае образы были представлены нормальными распределениями, искусственно «испорченными» добавлением зашумленных и случайных признаков (V1). Во втором случае образы изначально имели сложную полимодальную структуру, которая маскировалась добавлением зашумленных и случайных признаков (V2). Решалась задача выбора информативной подсистемы признаков. Для оценки эффективности того или иного критерия использовалась надежность распознавания тестовой выборки в выбранном с помощью критерия признаковом пространстве.

Анализ представленных результатов позволяет заключить, что если в простых случаях унимодальных образов с небольшой зоной пересечения все критерии работают примерно одинаково, то в случае сложных полимодальных структур FRiS-критерий значительно опережает другие критерии.

### Библиографический список

1. Конкурентное сходство как универсальный базовый инструмент когнитивного анализа данных / Н. Г. Загоруйко, И. А. Борисова, О. А. Кутенко, В. В. Дюбанов, Д. А. Леванов. URL: [http://ontology-of-designing.ru/article/2015\\_1%2815%292\\_Zagoruiko.pdf](http://ontology-of-designing.ru/article/2015_1%2815%292_Zagoruiko.pdf) (дата обращения: 25.05.2023).
2. Machinelearning: цифровой ресурс. URL: <http://www.machinelearning.ru/wiki/index.php?title=FRiS-функция> (дата обращения: 25.05.2023).
3. Обучение распознаванию без переобучения / Н. Г. Загоруйко, О. А. Кутенко, А. О. Зырянов, Д. А. Леванов. URL: <https://www.elibrary.ru/item.asp?id=32260625> (дата обращения: 25.05.2023).

УДК 004.89

**В. Ф. Губайдулин**

студент кафедры компьютерных технологий и программной инженерии

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

## ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ОПРЕДЕЛЕНИЯ ПАРАМЕТРОВ ОБЪЕКТОВ СПОРТИВНЫХ МЕРОПРИЯТИЙ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ ТРЕКИНГА

### Введение

В рамках предыдущей научно-исследовательской работы была реализована минимально жизнеспособная версия интеллектуальная система определения параметров объектов спортивного мероприятия. Свою основную задачу, определять параметры объектов спортивного мероприятия и перенести их на проекционную карту, система успешно выполняла [0, 0].

Несмотря на это был обнаружен ряд ограничений в работе системы:

- Каждый новый кадр обрабатывался системой по-новому, без учета предыдущих. Из-за этого стабильность определения объектов оставляла желать лучше, обнаруживаемые объекты то пропадали, то появлялись на итоговом видеоряде.

- Частота кадров в секунду редко переваливала за 15 единиц, из-за этого видео воспроизводилось недостаточно плавно.

- Возможность работы только с одной заранее известной видеозаписью.

- Отсутствие модуля расчета статистических данных обозреваемой игры.

Для решения данных ограничений было решено внедрить следующие инструменты, которые станут частью полноценной системы:

- DEEP SORT – simple online and realtime tracking.

- Пользовательский интерфейс.

- Сбор личных игровых метрик.

Была поставлена цель реализовать интеллектуальную систему определения параметров объектов спортивного мероприятия с использованием библиотеки трекинга. Для достижения озвученной цели были поставлены задачи:

- Использовать трекинг библиотеку DEEP-SORT для увеличения стабильности определения и трекинга объектов.

- Реализация сбора и демонстрации статистических метрик игры.

- Реализация интерфейса для настройки конфигурации работы системы.

### Описание структуры системы

Разработана структура разрабатываемого решения, представлена на рис. 1.

Желтым подсвечены элементы, которые будут добавлены при реализации усовершенствованного решения. Белые элементы уже включены в старую реализацию системы. [1, 2, 4].

Конфигурационный файл будет представлять из себя json файл в котором будут определены атрибуты, необходимые для работы сверточной нейронной сети, маппинга и трекинга в целом. [6] В табл. 1 представлены названия атрибутов, их описание и пример значения.

Заполнять данный файл можно будет с помощью прямого изменения json файла, но намного проще границы поля исходной видеозаписи и карты, а также диапазоны цветов указывать с помощью интерактивных инструментов, которые будут наглядно демонстрировать пользователю значения координат выбранных пикселей видеоизображения, либо же значение цвета, выбранного с помощью палитры. Было принято решение реализовать интерфейс настройки конфигурационного файла для обеспечения комплексности и полноценности разрабатываемого решения. Интерфейс будет реализован с помощью фреймворка Qt Designer. Qt Designer является фреймворком для реализации графических интерфейсов с использованием архитектурного паттерна MVC. Одним из главных плюсов данного фреймворка является поддержка HTML и CSS, что открывает безграничные возможности для визуализации графических интерфейсов.

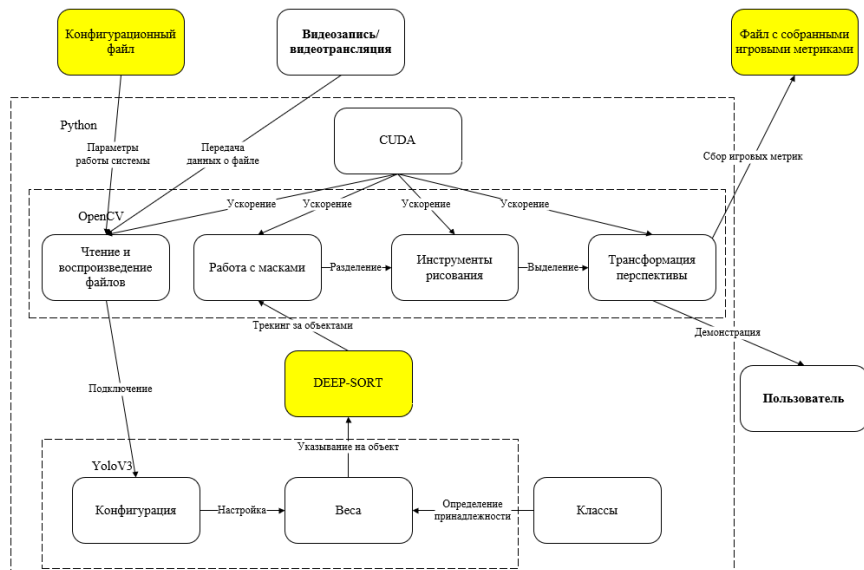


Рис. 1. Структура разрабатываемой системы

Таблица 1

Список атрибутов конфигурационного файла

Название атрибута	Тип	Описание атрибута	Пример значения
modelConfiguration	String	Путь к файлу, содержащему конфигурацию модели сверточной нейронной сети	'yolov3-320.cfg'
modelWeights	String	Веса, которые будут применяться к сверточной нейронной сети	'yolov3-320.weights'
initBoardPts	List<List<int>>	Значения координат исходного видеоизображения границ поля спортивного мероприятия	[[[0, 100], [1650, 104], [0, 1025], [1918, 1026]]]
mapBoardPts	List<List<int>>	Значения координат карты проекции границ поля спортивного мероприятия (должны соответствовать координатам из атрибута initBoardPts)	[[[0, 100], [1650, 104], [0, 1025], [1918, 1026]]]
colorList	List<string>	Названия цветов, на которые будет разбиваться определение объектов. Цвет зависит от цвета формы игроков.	['red', 'blue']
colorBoundaries	List<List<tuple>>	Диапазоны определения цветов (должны соответствовать названиям из атрибута colorList)	[[[(17, 15, 75), (50, 56, 200)], [(43, 31, 4), (250, 88, 50)]]]

DEEP-SORT с математической точки зрения основан на использовании: 1) расстояния Махаланобиса, 2) фильтра Калмана [3, 5].

Расстояние Махаланобиса – мера расстояния между векторами случайных величин, обобщающая понятие евклидова расстояния. Расстояние Махаланобиса необходимо, когда расчет простого евклидова расстояния от точки до точки не подходит, а также необходимо сделать поправку на корреляцию между значениями и дисперсию значений.

$$D_M(x) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)},$$

где  $D_M(x)$  – расстояние,  $x$  – многомерный вектор,  $\mu$  – среднее значение,  $S$  – матрица ковариации.

Для расчета используется ковариационная матрица. Матрица ковариации – матрица, из попарных ковариаций элементов одного или двух случайных векторов. Из формулы, чем выше корреляция между переменными, тем меньше дистанция, так как идет деление на меньшее число.

Фильтр Калмана – рекурсивный фильтр, оценивающий вектор состояния динамической системы, используя ряд неполных и зашумленных измерений:

1) усиление Калмана:

$$KG = \frac{E_{EST}}{E_{EST} + E_{MEA}},$$

где  $E_{EST}$  – ошибка значения, которое оценивается,  $E_{MEA}$  – погрешность прибора;

2) оценка значения с учетом посчитанного усиления:

$$EST_t = EST_{t-1} + KG(MEA - EST_{t-1}),$$

где  $MEA$  – новое показание прибора,  $EST_{t-1}$  – значение, оцененное на предыдущем этапе;

3) оценка новой ошибки:

$$E_{EST_t} = E_{EST_{t-1}}(1 - KG).$$

На рис. 2 представлена схема алгоритма DEEP-SORT.

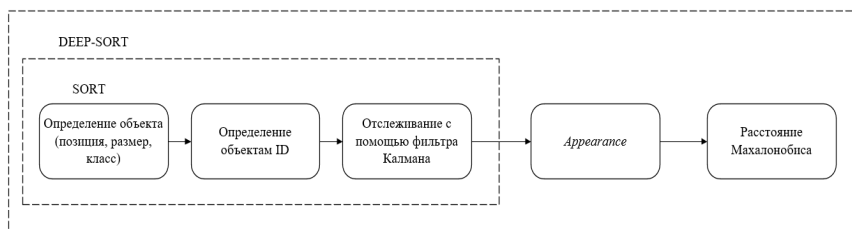


Рис. 2. Алгоритм DEEP-SORT

Было принято решение реализовать сбор игровых статистических данных с видеоизображения. В качестве сбора игровых статических данных с изображения будут сниматься: карта касаний мяча определенного игрока, карта ударов по воротам определенного игрока. Касание мяча игроком будет считаться, когда область обнаружения игрока будет «соприкасаться» с областью обнаружения мяча. На схему поля будет наносится точка-местоположения игрока в момент соприкосновения с мячом. Удар по мячу игроком будет считаться, когда после соприкосновения области обнаружения игрока и области обнаружения мяча, мяч оказывается в заранее определенной области на изображении у ворот. На схему поля будет наносится точка-местоположения игрока в момент соприкосновения с мячом.

### Заключение

Были определены ограничения предыдущей реализации и были обозначены методы их устранения. Для повышения стабильности работы системы, а также улучшения алгоритма необходимо имплементировать алгоритм DEEP-SORT. После имплементации алгоритма необходимо провести различные тесты: насколько повысилась стабильность, насколько оптимизированная система и прочее. Для повышения гибкости приложения необходимо реализовать конфигурационный файл, а также реализовать интерфейс изменения конфигурационного файла. Наконец, для повышения сложности и расширения области применения системы необходимо реализовать модуль подсчета статистических данных обозреваемого спортивного мероприятия. Особую благодарность автор выражает кандидату технических наук, доценту кафедры прикладной информатики Турнецкой Елене Леонидовне за помощь в формировании темы и задач научно-исследовательской работы.

### Библиографический список

1. Губайдулин В., Турнецкая Е. Интеллектуальная система определения параметров объектов спортивного мероприятия // Семьдесят пятая международная студенческая научная конференция. ГУАП. Ч. 2. Технические науки. 2022. С. 128–131.
2. D'Orazio T. [и др.]. A Semi-automatic System for Ground Truth Generation of Soccer Video Sequences // Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance. 2009. С. 559–564.
3. Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, Ben Uppcroft. Simple Online and Realtime Tracking // IEEE International Conference on Image Processing (ICIP). 2016. С. 3464–3468.
4. Takezawa Y., Hasegawa M., Tabbone S. Camera-captured document image perspective distortion correction using vanishing point detection based on Radon transform // 23rd International Conference on Pattern Recognition. 2016. С. 3957–3968
5. Nicolai Wojke, Alex Bewley, Dietrich Paulus. Simple Online and Realtime Tracking with a Deep Association Metric // IEEE International Conference on Image Processing (ICIP). 2017. С. 645–6649.
6. Николенко С., Кадулин А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей. СПб.: Питер, 2022. 480 с.
7. Redmon J. [и др.]. You Only Look Once: Unified, Real-Time Object Detection // IEEE Conference on Computer Vision and Pattern Recognition. 2016. С. 779–788.

УДК 004.912

**А. Е. Дулимов**

студент кафедры компьютерных технологий и программной инженерии

**А. А. Ключарев** – кандидат технических наук, доцент – научный руководитель

### ФОРМАЛИЗАЦИЯ ЗАДАЧИ ФОРМИРОВАНИЯ ЛЕКЦИОННЫХ ПОТОКОВ ПРИ ПЛАНИРОВАНИИ УЧЕБНОГО ПРОЦЕССА ВУЗА

Решение задач автоматизации планирования и сопровождения учебного процесса вуза актуально в связи со значительным количеством реализуемых образовательных программ (ОП), которые должны соответствовать множеству формальных критериев, и большим контингентом студентов, обучающихся по различным формам. При этом необходимо обеспечить высокое качество подготовки выпускников при имеющихся ресурсах аудиторного фонда, учебного оборудования и преподавательского состава. Для решения этих задач в СпбГУАП используется информационная система (АИС УМО), позволяющая автоматизировать основные процессы, реализуемые учебно-методическим отделом (УМО). В настоящее время эта система эффективно используется для решения следующих основных задач:

1. Разработка ОП в соответствии с требованиями образовательных стандартов, нормативных актов и документов системы менеджмента качества вуза.
2. Автоматизированное формирование комплекта документов, входящих в ОП – календарных графиков учебного процесса, учебных планов, рабочих программ дисциплин и т. п.
3. Планирование учебной нагрузки и ее распределение между кафедрами с учетом текущих изменений контингента студентов.
4. Подготовка исходных данных для составления расписания занятий и штатного расписания преподавательского состава кафедр.



Рис. 1. Этапы формирования и сопровождения образовательной программы

Здесь перечислены только основные функции АИС УМО, связанные с планированием и сопровождением учебного процесса (рис. 1). Характерным для используемой системы является включение функций автоматического контроля корректности данных на всех этапах формирования документов.

Важной, но пока еще не решенной в АИС УМО задачей, является формирование потоков для проведения лекционных занятий. Корректное решение этой задачи позволяет более эффективно использовать имеющийся потенциал преподавательского состава, обеспечивает рациональную загрузженность аудиторного фонда и, в конечном счете, приводит к повышению качества образования.

По сути, эта задача относится к области многокритериальной оптимизации, и для ее решения необходимо сформулировать и обосновать критерии выбора оптимальной структуры лекционных потоков и выполнить формализацию составляющих, влияющих на их состав.

#### Анализ задачи формирования лекционных потоков

##### Определения

Для построения формальной модели требуется определить, что является лекционным потоком и от чего зависит его формирование [1, 2]. Лекционный поток – это несколько учебных групп, слушаю-

ших вместе одну и ту же дисциплину у одного и того же преподавателя в одной и той же аудитории. Множество лекционных потоков – все сформированные потоки. Формально это можно обозначить следующим образом:

$$P = \{p_1, p_2, \dots, p_n\} \text{ – множество потоков.}$$

Поток

$$p_i = \{G_p, t_i, d_i, a_i\},$$

где  $G_p$  – множество групп потока;  $t_i$  – преподаватель;  $d_i$  – дисциплина (количество часов, отведенное на дисциплину);  $a_i$  – аудитория.

Из всех характеристик группы при составлении потоков нас интересуют ее численность и дисциплины, которые ей необходимо освоить. Для данной задачи множество групп будет описываться так:

$$G = \{g_1, g_2, \dots, g_m\} \text{ – множество групп.}$$

Группа

$$g_i = \{n_s, D_g\},$$

где  $n_s$  – число студентов;  $D_g = \{d_1, d_2, \dots, d_o\}$  – множество дисциплин, которые должны читаться группе.

Основными параметрами преподавателя, в свою очередь будут являться списки дисциплин, которые он потенциально может преподавать и дисциплин, которые он поставлен вести, а также предельная и текущая нагрузка (в часах). Предельная нагрузка исходит из нормативных актов, текущая же будет являться суммой часов дисциплин, которые он ведет. Формально это можно представить так:

$$T = \{t_1, t_2, \dots, t_j\} \text{ – множество преподавателей.}$$

Преподаватель

$$t_i = \{dr, dr_{max}, D_t\},$$

где  $D_t$  – множество дисциплин, которые может читать преподаватель;  $D_{w_i}$  – множество дисциплин, которые читает преподаватель;  $dr = \sum_{d_j \in D_{w_i}} d_j$  – нагрузка на преподавателя (часы);  $dr_{max}$  – допустимая нагрузка (часы).

Последнее, что необходимо определить – это аудитории. Два основных параметра, интересующих нас для данной задачи – это ее вместимость и предельная загруженность. Вместимость – число мест для студентов, предельная загруженность – максимальное число пар, которые возможно провести в данной аудитории за семестр. Формальное описание:

$$A = \{a_1, a_2, \dots, a_k\} \text{ – множество аудиторий.}$$

Аудитория

$$a_j = \{n_s, dr\},$$

где  $n_s$  – число студентов;  $dr$  – предельная нагрузка (часы).

### Целевая функция

В рамках данной работы основным критерием оптимизации было решено взять минимизацию нагрузки на преподавателей по двум основным причинам:

1. Ввиду последних требований министерства науки и высшего образования РФ [3, 4] к составу образовательных программ, необходимо читать большое количество общих для всех направлений подготовки дисциплин (таких как двухсеместровый курс истории России и основы военной подготовки), в



связи с чем резко повышается нагрузка на часть преподавательского состава, ответственную за данные дисциплины.

2. Снижение нагрузки на преподавательский состав влечет за собой снижение общего числа занятий, что, в свою очередь, приводит к снижению нагрузки и на аудиторный фонд вуза.

Соответственно, целевую функцию формальной модели можно записать следующим образом:

$$\sum_{t_i \in T} dr_{t_i} \rightarrow \min.$$

**Ограничения**

Список ограничений, которые необходимо соблюдать при формировании потоков и их формальных описаний:

1. Для каждого потока общая численность студентов в нем не должна превышать вместимость выбранной аудитории:

$$\sum_{g_j \in G_{p_i}} n_{sg_j} \leq n_{sa_i}, \forall p_i \in P.$$

2. Суммарное количество часов дисциплин, ведущихся преподавателем, не должно превышать его предельную нагрузку:

$$dr_{t_i} \leq dr_{\max t_i}, \forall t_i \in T.$$

3. Суммарное количество часов дисциплин, ведущихся в аудитории, не должно превышать ее предельную загруженность. Для формального описания дополнительно определим  $D_{a_i}$  как множество всех дисциплин, читаемых в аудитории  $a_i$ , тогда:

$$\sum_{d_j \in D_{a_i}} d_{j_{a_i}} \leq dr_{a_i}, \forall a_i \in A.$$

4. Желательно добавлять в поток студентов только одной укрупненной группе специальностей и направлений, либо выбирать группы другому критерию близости программ, заданному экспертно. При невозможности формирования потоков с соблюдением всех ограничений, данное может быть отброшено (либо должен быть изменен критерий близости программ). Для формального описания необходимо дополнительно определить следующую функцию:

$$sim(g_j) = \begin{cases} 1, & \text{if } g_j \in G_{sim} \\ 0, & \text{if } g_j \notin G_{sim} \end{cases}$$

Тогда данное условие формулируется следующим образом:

$$\sum_{g_j \in G_{p_i}} sim(g_j) = |G_{p_i}|, \forall p_i \in P.$$

**Итоговый вид модели:**

$$\sum_{t_i \in T} dr_{t_i} \rightarrow \min, \left\{ \begin{array}{l} \sum_{g_j \in G_{p_i}} n_{sg_j} \leq n_{sa_i}, \forall p_i \in P \\ dr_{t_i} \leq dr_{\max t_i}, \forall t_i \in T \\ \sum_{d_j \in D_{a_i}} d_{j_{a_i}} \leq dr_{a_i}, \forall a_i \in A \\ \sum_{g_j \in G_{p_i}} sim(g_j) = |G_{p_i}|, \forall p_i \in P \end{array} \right.$$

### **Заключение**

В результате выполнения данной работы были обозначены основные задачи автоматизации при планировании и сопровождении образовательной программы и проведен анализ задачи формирования лекционных потоков, результатом которого явилась ее формальная модель.

### **Библиографический список**

1. Шпак А. В. Математическая модель формирования учебных потоков с целью оптимизации использования лекционно-практического аудиторного фонда // Автоматика. Информатика. 2002. № 1-2(10-11). С. 51-54.
2. Сеньковская А. А. Математическое моделирование и методы оптимизации учебного процесса вуза (на примере задач формирования рабочих учебных планов): автореферат диссертации на соискание ученой степени кандидата технических наук. Омск, ФГБОУ ВО «Омский государственный университет им. Ф. М. Достоевского», 2021. 20 с.
3. Приказ Минобрнауки России от 06.04.2021 № 245 (ред. от 02.03.2023). Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры. URL: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_393023](https://www.consultant.ru/document/cons_doc_LAW_393023) (дата обращения: 12.04.2023).
4. Приказ Минобрнауки России от 22.12.2014 № 1601 (ред. от 13.05.2019). О продолжительности рабочего времени (нормах часов педагогической работы за ставку заработной платы) педагогических работников и о порядке определения учебной нагрузки педагогических работников, оговариваемой в трудовом договоре. URL: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_175797](https://www.consultant.ru/document/cons_doc_LAW_175797) (дата обращения: 12.04.2023).

УДК 004.934.8\*1

**Р. С. Зулкашев**

студент кафедры информационных технологий и программирования

**М. Д. Поляк** – старший преподаватель**АЛГОРИТМ СИНТЕЗА ДАТАСЕТА ДЛЯ ЗАДАЧИ ДИАРИЗАЦИИ****Введение**

Диаризация – это процесс разделения и идентификации говорящих на аудиозаписи. Эта задача имеет большое практическое значение в области распознавания речи, поскольку она позволяет улучшить качество распознавания и анализа речевых данных, а также автоматизировать ее сбор. К примеру, диаризация позволяет создавать транскрипцию интервью, анализировать телефонные разговоры и многое другое. Одна из основных проблем в задаче диаризации — это доступность размеченных данных. Это связано с тем, что каждый участок речи должен быть аннотирован с точностью до миллисекунд. Уже существует несколько достаточно объемных наборов аудиоданных, но все они состоят из иноязычных аудиозаписей [1, 2]. Один из возможных подходов для создания набора данных на любом языке — это его генерация. В данном случае наибольший интерес вызывает создание именно русскоязычного набора данных, который далее можно было бы использовать для обучения модели диаризации. Проводились полноценные исследования по созданию и анализу искусственных аудиоданных, которые продемонстрировали эффективность применения таких данных в процессе обучения моделей [3, 4, 5]. В данной статье будут обобщены существующие подходы и исследованы новые в способах генерации аудиоданных для задачи диаризации.

**Цель:** показать реализованные подходы к генерации аудио данных для задачи диаризации и оценить качество полученных данных

**Описание логики программы**

Описание алгоритма генерации датасета (набора данных) стоит начать с описания входных и выходных наборов данных. На вход программе поступают аудиофайлы, содержащие запись голоса только одного говорящего. Файлы сгруппированы по каждому чтецу. В качестве выхода программа будет генерировать датасет с временными метками для каждого участка с речью. Общую схему работы алгоритма можно описать в 5 основных этапах (рис. 1):

1. Выделение участков с речью из входного аудиофайла.
2. Случайный выбор двух чтецов и перемешивание их аудио фрагментов с речью.
3. Нормализация.
4. Добавление пауз и перебиваний.
5. Добавление шумов.

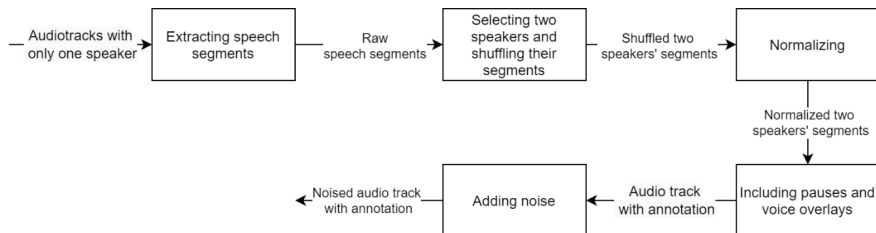


Рис. 1. Блок-схема логики программы

Первый этап создания датасета – это выделение фрагментов речи из аудиодорожки. Данную процедуру можно проделать с помощью алгоритма voice activity detection (VAD). В нашей реализации

для этих целей была использована модель из ruannote [6]. Таким образом, после обработки аудиофайла получается список аудио сегментов с речью.

Следующий этап состоит в том, чтобы объединить списки фрагментов речи нескольких говорящих и добавить метки класса для каждого элемента. Затем получившийся массив фрагментов аудио и меток одинаково перемешивается. Так в датасете достигается симуляция процесса общения между людьми.

На третьем этапе фрагменты речи нормализуются по громкости. Для этого вычисляется средняя громкость переданных аудио фрагментов и громкость каждого отрезка речи смещается к этому среднему. Результаты такой обработки показаны на рис. 2, 3. В данном случае графики выводят сырые значения аудиоданных для более лучшей визуализации различий.

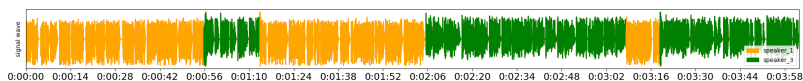


Рис. 2. Без нормализации по громкости

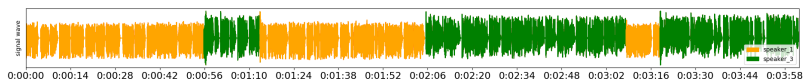


Рис. 3. После нормализации по громкости

Во время четвертого этапа, получившийся список аудио фрагментов соединяется в единую аудиодорожку. Процесс работает в цикле по каждому аудио фрагменту и добавляет его к уже объединенным. При этом сохраняются метки времени каждого добавленного фрагмента. Объединение происходит случайно одним из двух способов: соединение через паузу и через наложение дорожек друг на друга.

В первом случае генерируется пауза, и если она достаточно продолжительная, то вставляется между объединенными кусками речи. Иначе элементы объединяются без добавления паузы. Вдохновившись исследованиями естественных диалогов [3, 7, 8], для генерации длительности паузы использовалось распределение Рэлея с модой в 200 мс.

Во втором же варианте сначала проверяется условие возможности наложения двух аудио фрагментов, а после вычисляется ее максимально возможная продолжительность. Условие проверки примитивное – один и тот же говорящий не может перебивать сам себя. Время наложения ограничено:

- продолжительностью накладываемой дорожки;
- продолжительностью дорожки, на которую накладывают;
- не может накладываться на предыдущего говорящего;
- не может выходить за рамки максимума и минимума времени наложения.

Далее, если вычисленная продолжительность наложения больше минимального порога, то с заданной вероятностью может произойти наложение речи собеседников друг на друга. Причем реальное время наложения будет выбираться случайно из диапазона [пороговый минимум, вычисленный максимум]. Это основная логика в генерации датасета. Но есть еще один подход для улучшения синтетических данных.

На последнем шаге к выходному аудио файлу добавляется шум из датасета DEMAND [9], состоящий из звуков толпы, улиц, леса, метро и т. п. Необходимость в наложении шума исходит из того факта, что входные данные записаны в идеальных условиях, а диалоги без посторонних звуков в реальности редкое явление. Поэтому для приближения синтетического датасета к реальным данным необходимо добавлять шум.

В качестве примера один из сгенерированных аудио файлов программой показан на рис. 4, 5. Цвет указывает какому говорящему какой участок аудио принадлежит. График временных участков на рис. 4 служит для простой демонстрации взаимного наложения участков речи. Когда происходит нало-

жение, то накладываемый временной отрезок смещается на побочную (нижнюю) ось. Осциллограмма аудио дорожки на рис. 5 нужна для более полной информации об аудио записи.



Рис. 4. График временных участков

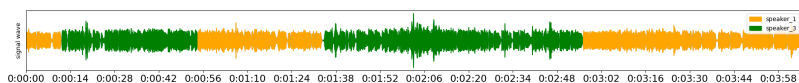


Рис. 5. Осциллограмма звуковой дорожки

### Оценка качества диаризации

Оценим качество сгенерированных аудиоданных, пропустив их через модель диаризации и посчитав метрику DER. Метрика считается по формуле:  $(False\ Alarm + Miss + Overlap + Confusion) / Reference\ Length$ . Т.е. чем меньше DER – тем лучше качество диаризации. Чтобы распределение речевых фрагментов не менялось при каждом запуске, используется фиксированный random state. Результаты представлены в табл. 1.

Таблица 1

### Метрика DER

Вариант аудио	DER
Без наложения	0,0105
С наложением	0,0147
Без наложения + шум	0,0296
С наложением + шум	0,0254
С наложением + шум + 4 говорящих	0,0767

Результат показывает, что наложения, шум и увеличение кол-ва говорящих усложняют процесс диаризации. И если судить только по метрикам, то сгенерированные данные приближаются к реальным. Но на сколько хорошо? Для сравнения, вычисленная метрика DER на реальной аудио дорожке на английском языке равна 0,111. Но в примере аудио содержало уже 9 говорящих, тем самым объясняя сильную разницу в качестве. Следовательно, для достижения более качественной симуляции нужно увеличить кол-во говорящих. Также еще стоит улучшить нормализацию по громкости, так как на данном этапе можно визуально отличить (рис. 2, 3), кто и когда начинает говорить, просто взглянув на осциллограмму.

### Заключение

Искусственные данные имеют свои преимущества и недостатки. Генерация набора данных гораздо проще и дешевле, чем сбор реальных данных. Однако следует отметить, что синтетические данные не могут полностью заменить реальные, потому что нельзя в полной мере воссоздать все возможные условия записи и особенности стиля общения людей. Кроме того, создание высококачественных синтетических данных требует определенных знаний в предметной области.

Результаты метрик показывают, что сгенерированные аудиоданные имеют еще недостаточно хорошее качество, но уже являются неплохим начальным приближением к реальному распределению. Для более качественной генерации диалогов должна быть реализована более качественная нормализация по громкости.

Таким образом, использование синтетических данных для задачи диаризации может быть эффективным решением при правильном подходе к их созданию, но также требует осторожности и дополнительной проверки качества модели на реальных данных.

Код данной работы доступен по ссылке:  
[https://github.com/Bumstem/synthetic\\_diarization\\_dataset\\_generation](https://github.com/Bumstem/synthetic_diarization_dataset_generation).

### Библиографический список

1. Neville Ryant, Prachi Singh, Venkat Krishnamohan, Rajat Varma, Kenneth Church, Christopher Cieri, Jun Du, Sriram Ganapathy, Mark Liberman. The Third DIHARD Diarization Challenge, 2020. URL: <https://arxiv.org/abs/2012.01477> (дата обращения 10.03.2023).
2. Awesome Speaker Diarization. Datasets – github.com. URL: <https://github.com/wq2012/awesome-diarization#Datasets> (дата обращения: 13.03.2023).
3. Edwards E. [et al.]. A Free Synthetic Corpus for Speaker Diarization Research. Speech and Computer // SPECOM 2018. Lecture Notes in Computer Science(). Vol. 11096. Springer, Cham. URL: [https://doi.org/10.1007/978-3-319-99579-3\\_13](https://doi.org/10.1007/978-3-319-99579-3_13) (дата обращения: 13.03.2023).
4. Federico Landini, Alicia Lozano-Diez, Mireia Diez, Lukáš Burget. From Simulated Mixtures to Simulated Conversations as Training Data for End-to-End Neural Diarization. URL: [https://www.researchgate.net/publication/359729621\\_From\\_Simulated\\_Mixtures\\_to\\_Simulated\\_Conversations\\_as\\_Training\\_Data\\_for\\_End-to-End\\_Neural\\_Diarization](https://www.researchgate.net/publication/359729621_From_Simulated_Mixtures_to_Simulated_Conversations_as_Training_Data_for_End-to-End_Neural_Diarization) (дата обращения 10.03.2023).
5. Popov A. S., Ivanov S. A. Neural Network Models for Russian Language Speaker Recognition // 2021 International Conference on Quality Management, Transport and Information Security, Information Technologies (IT&QM&IS). Yaroslavl, Russian Federation. 2021. Pp. 533–536. Doi: 10.1109/ITQMIS53292.2021.9642756.
6. Bredin Herve, Laurent Antoine. End-to-end speaker segmentation for overlap-aware resegmentation // Proc. Interspeech 2021. URL: <https://arxiv.org/pdf/2104.04045.pdf> (дата обращения 10.03.2023).
7. Mattias Heldner, Jens Edlund. Pauses, gaps and overlaps in conversations // Journal of Phonetics. Vol. 38. Issue 4. 2010. Pp. 555–568. ISSN 0095-4470.
8. Universals and cultural variation in turn-taking in conversation / Tanya Stivers, N. J. Enfield, Penelope Brown [et al.] // Proceedings of the National Academy of Sciences. 2009. 10587–10592. 106. 26. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.0903616106> (дата обращения: 13.03.2023).
9. Thiemann Joachim, Ito Nobutaka, & Vincent, Emmanuel. DEMAND: a collection of multi-channel recordings of acoustic noise in diverse environments (1.0). 21st International Congress on Acoustics. ICA 2013. Montreal, Canada. Zenodo. URL: <https://doi.org/10.5281/zenodo.1227121> (дата обращения: 13.03.2023).

УДК 004

**А. П. Калинин**

студент кафедры компьютерных технологий и программной инженерии

**П. А. Степанов** – кандидат технических наук, доцент – научный руководитель**РАЗРАБОТКА АЛГОРИТМОВ И ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ ПЛАНИРОВАНИЯ ПУТЕШЕСТВИЙ НА ОСНОВАНИИ ПЕРЕЧНЯ ОСНОВНЫХ ПАРАМЕТРОВ****Введение**

На сегодняшний день туристическая отрасль является важной частью экономики как страны в целом, так и ее отдельных регионов. Развитие туризма способствует динамичному созданию рабочих мест, росту предпринимательской активности, развитию сферы обслуживания, повышению уровня жизни населения и росту доходов [1].

Потенциал для развития туристической индустрии в нашей стране обеспечивают природное разнообразие и богатое культурное наследие. На территории России располагаются уникальные достопримечательности и объекты всемирного наследия ЮНЕСКО, а также множество курортов и лечебных зон. В нашей стране туризм переживает качественный скачок, становясь одной из ведущих отраслей национальной экономики. Появляются новые перспективные направления, возрастает спрос, открываются предприятия [2].

Полезным инструментом в развитии туристической индустрии служат мобильные технологии. Их бурный рост привел к распространенности мобильных приложений, в том числе и для сферы туризма. Как показывает практика российских турагентств, все больше клиентов предпочитают выбирать и бронировать туры с мобильных устройств. По оценке агентства OHM Solutions, доля путешествий, забронированных онлайн, в России составляет около 30%, при этом абсолютное большинство (75–80%) заказов приходится на мобильные устройства. При этом, на авиабилеты приходится более половины всех онлайн-покупок, следом идут ж/д билеты и оформление проживания в отелях.



Рис. 1. Доля путешествий, забронированных онлайн по странам, и сегменты онлайн-рынка туризма по оценке OHM Solutions

### Актуальность исследования

Следует отметить, что, несмотря на высокие темпы роста, национальный туризм не обладает достаточной степенью поддержки на государственном уровне. Вызывает беспокойство и тот факт, что вследствие современных геополитических ограничений, туристический поток из-за рубежа сильно сократился, что привело к нерезализованности рекреационного потенциала и снижению спроса на российские курорты.

Принимая во внимание указанные проблемы, возникает необходимость в дальнейшем развитии сферы отечественного туризма. Выбранная тема посредством разработки полноценного мобильного приложения вносит вклад в доступность и удобство туристических услуг.

Использование мобильных приложений в туристической отрасли имеет множество преимуществ:

- Помогает наладить связь между турфирмами и клиентами, позволяет пользователям оперативно получать информацию о новых туристических услугах. Это повышает качество и привлекательность многих внутренних туристических направлений.

- Приложения могут использоваться для привлечения новых пользователей, популяризации памятников культуры и рекламы туров.

- Мобильные приложения экономически выгодны и удобны в использовании. Благодаря им туристические продукты становятся более индивидуальными и гибкими.

- Цифровизация отрасли помогает упростить многие бизнес-процессы, повышает безопасность и минимизирует риски.

В целом, имеется достаточно много конкурирующих мобильных приложений в туристической сфере, однако все они нацелены на подбор и бронирование туров по всему миру. В пределах одной страны это может сократить количество предлагаемых вариантов размещения и достопримечательностей. Кроме того, часто в приложении бывает трудно сравнивать множество похожих поездок с незначительными различиями. Поэтому полезной функцией было бы введение рекомендаций на основе личных предпочтений пользователя, с помощью алгоритмов машинного обучения. Это стало бы шагом к тому, чтобы упростить пользователю решение его задач и облегчить взаимодействие с приложением.

### Цель и задачи

Основной целью работы является повышение привлекательности и доступности услуг внутренней туристической отрасли, помощь турфирмам в привлечении новой аудитории благодаря разработке мобильного приложения для рекомендации путешествий.

Для решения этих задач необходимо:

- Осуществить анализ известных отечественных продуктов, предлагающих услуги поиска и оформления туров.

- Изучить технологии создания и применения моделей машинного обучения для iOS-приложений (TensorFlow + CoreML) [3, 4, 5].

- На основе тренировочных данных разработать модель машинного обучения, реализующую алгоритм «фильтрации на основе содержания» для рекомендации путешествий.

- Встроить рекомендательную модель в мобильное приложение, наполнить его широким функционалом по поиску и рекомендации поездок.

- Предоставить пользователю возможность управлять своим аккаунтом и избранными путешествиями.

### Выбор технических средств и создание приложения

В качестве целевой операционной системы для создания мобильного приложения будет выбрана iOS от компании Apple. Данная система высоко автономна, удобна в использовании и практически не имеет проблем с совместимостью. Устройства iPhone, работающие под управлением iOS, несмотря на недавние геополитические ограничения, остаются одной из самых популярных категорий смартфонов в России. По оценке ТАСС, доля iPhone в сетях российских операторов в 2022 году выросла в среднем до 18–19%.



В качестве среды разработки будет выбрана среда XCode, позволяющая создавать приложения для iPhone. Приложение будет разработано с помощью языка программирования Swift.

В качестве СУБД и сервиса для работы с аутентификацией будет использована платформа Google Firebase.

Для построения системы рекомендаций будет использована технология CoreML от Apple, а также TensorFlow от Google [5, 6].

Разработанное ПО позволит пользователям производить следующие действия:

- Указать свои предпочтения в формате тестирования.
- Оценить/добавить в избранное пакетные туры, включающие город-курорт, отели, достопримечательности; ознакомиться с их описанием.
- Изменять избранные туры и удалять их.
- Возможность управлять своим аккаунтом и изменять информацию о себе.
- Осуществить поиск с фильтрами и увидеть доступные варианты бронирования отелей, покупки авиабилетов [7, 8].
- Получить рекомендации пакетных туров, с возможностью изменить их составляющие по своему усмотрению [9, 10].

### Схема работы рекомендательной модели

Рекомендательная модель будет реализовывать принцип машинного обучения «фильтрации на основе содержания». Она будет рекомендовать пользователю туры, похожие на те, которые понравились ему в прошлом. Сходство туров будет рассчитываться на основе характеристик, присущих каждому элементу: оценки от пользователей и город назначения. В конечном счете модель сконцентрируется на собственных предпочтениях каждого пользователя.

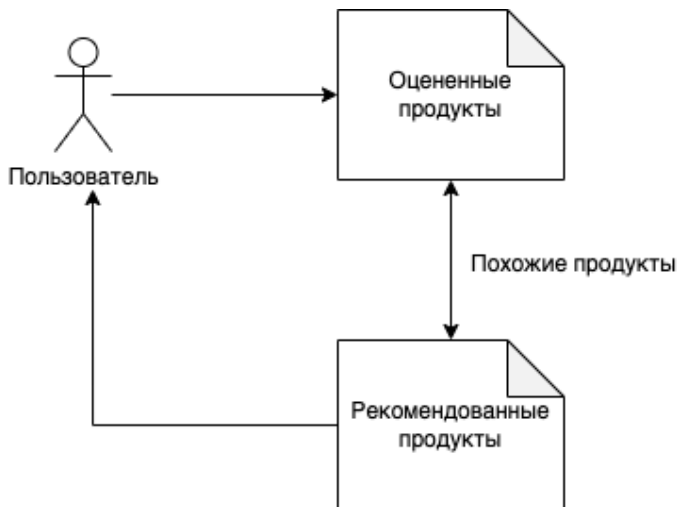
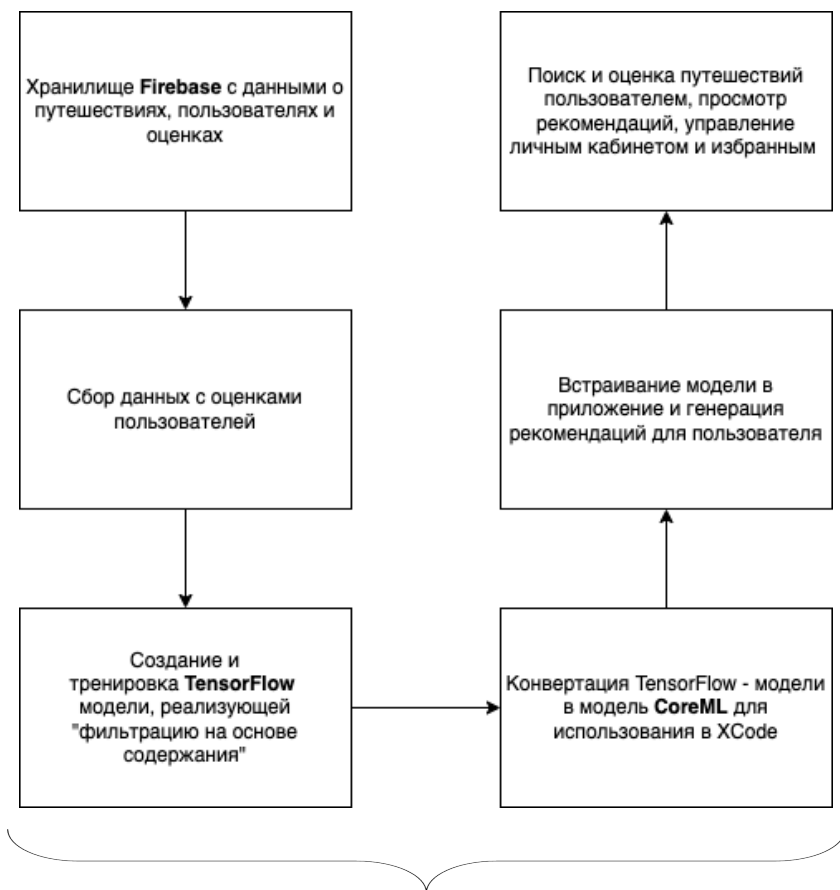


Рис. 2. Схема работы модели «фильтрации на основе содержания»

Модель будет создаваться, храниться и дообучаться при необходимости на удаленном сервере. Фреймворк TensorFlow от Google позволяет создать рекомендательную модель и точно настроить ее, а также может учитывать более одного параметра (оценки) рекомендуемых сущностей. После создания и тренировки с помощью TensorFlow, модель будет преобразована в модель CoreML, для возможности экспорта в среду XCode [5, 6].



*Данные действия производятся на сервере*

*Рис. 3. Общая схема создания рекомендательной модели*

Готовая модель будет встроена в приложение и использована для генерации рекомендаций пользователю, учитывая его собственные оценки и предпочтения. Пользователь на своей стороне может совершать все вышеуказанные действия с путешествиями и своим личным кабинетом.

### **Области предполагаемого применения результатов исследований**

Разработанный продукт будет предназначен для пользователей, желающих спланировать поездку по России. Благодаря дружественному интерфейсу и легкости приложения в эксплуатации, пользователи смогут максимально удобно получать всю необходимую информацию.

Помимо рядовых пользователей, программный продукт будет интересен различным туристическим операторам и фирмам, которые специализируются на внутреннем туризме. Встроив в приложение собственные туры и программы, они смогут более эффективно продвигать свои услуги, и привлекать новую аудиторию.

**Библиографический список**

1. Gallagher A., Hollemans M., Tam A., LaPollo C. Machine Learning by Tutorials. Razeware LLC, 2019. 586 с.
2. Thakkar M. Beginning Machine Learning in iOS: CoreML Framework. Apress, 2019. 171 с.
3. Nyzsitor K., Nyzsitor M. Machine Learning with Core ML 2 and Swift: A beginner-friendly guide to integrating machine learning into your apps. Amazon Digital Services LLC, 2018. 187 с.
4. The Official Google Documentation. TensorFlow Recommenders. URL: <https://www.tensorflow.org/recommenders?hl=ru> (дата обращения: 27.03.2023).
5. George Pipis. Content-Based Recommender Systems with TensorFlow Recommenders. URL: <https://predictivehacks.com/content-based-recommender-systems-with-tensorflow-recommender/> (дата обращения: 20.03.2023).
6. Маскри М. Swift 3: разработка приложений в среде Xcode для iPhone и iPad с использованием iOS SDK. М.: Диалектика, 2017. 896 с.
7. Бочарников В. Н., Лаврушина Е. Г., Блиновская Я. Ю. Информационные технологии в туризме. М.: Флинта, 2008. 144 с.
8. Traveldon.ru. Развитие туризма в России на современном этапе. URL: <https://traveldon.ru/stato-turizme-i-puteshestviyah/razvitie-turizma-v-rossii> (дата обращения: 02.04.2023).
9. Официальный сайт Агентства социальной Информации. Социально-ответственный туризм: путешествия по России. URL: <https://www.asi.org.ru/news/2016/12/02/sotsialnootvet> (дата обращения: 02.04.2023).
10. Вишневская М. Рейтинг лучших приложений для поиска туров на 2020 год для любителей организованного отдыха. URL: <https://zuzako.com/rejting-luchshih-prilozheniy-dlya-poiska-turov> (дата обращения: 25.03.2023).

УДК 681.5

**Д. Р. Котин**

студент кафедры компьютерных технологий и программной инженерии

**В. В. Мышко** – кандидат технических наук, доцент – научный руководитель

### МАТЕМАТИЧЕСКАЯ МОДЕЛЬ УПРАВЛЕНИЯ ТЕХНИЧЕСКИМ СОСТОЯНИЕМ СЛОЖНОГО ТЕХНИЧЕСКОГО ОБЪЕКТА

Современные технические системы сильно декомпозированы, состоят из многих подсистем, элементов, связанных между собой. Стоимость реализации таких систем крайне высока, как и стоимость непредвиденного отказа в процессе работы, что накладывает жесткие требования на их надежность. К подобным системам можно отнести многие современные технические средства, в особенности – космические.

Распределенная структура технического средства накладывает определенные требования на обеспечение надежности. Любой элемент может выйти из строя, тем самым нарушив работу системы.

В настоящее время активно используются системы наблюдения технических средств. Возможность наблюдения и диагностирования закладывается в структуру современных систем. Процедуры наблюдения и диагностирования необходимо проводить без нарушения функционирования технического средства.

Для наблюдения элемента необходим, как минимум, односторонний обмен информацией с системой наблюдения. По результатам наблюдения проводится процедура диагностирования, с целью идентификации состояния.

В данной работе рассматриваются существующие математические модели диагностирования и модели процесса анализа технического состояния (ТС) сложного технического объекта (СТО).

Целью работы является создание математической модели объекта управления и процесса управления ТС СТО, формулировка задачи синтеза программы управления.

Наблюдаемый СТО имеет набор диагностических признаков (ДП), которые могут представлены дискретными значениями или непрерывными интервалами значений. Если известны значения каждого ДП, то известно и ТС объекта. Но получение информации о каждом ДП избыточно, зачастую достаточно меньшего набора проверок для определения ТС.

Рассмотрим модели объекта анализа (ОА) для дискретных и интервальных значений ДП.

Модель ОА ТС СТО при дискретной форме представления ДП [1] имеет вид:

$$M_{OA} = \langle S, \Pi, \Gamma, P, \Phi \rangle,$$

где  $S$  – множество ТС, подлежащих распознаванию,  $S = \{S_i | i = 1, m\}$ ;  $\Pi$  – множество ДП, на котором все ТС  $S_i \in S$  попарно различимы,  $\Pi = \{\pi_j | j = 1, n\}$ ;  $\Gamma$  – множество модельных значений ДП  $\pi_j \in \Pi$  в

ТС  $S_i \in S, \Gamma = \{s_{ij} | i = 1, m; j = 1, n\}$ ;  $P$  – множество вероятностей ТС  $S_i \in S, P = \left\{ P(S_i) \mid \sum_{i=1}^m P(S_i) = 1 \right\}$ ;

$\Phi = S \times \Pi \rightarrow \Gamma$  – отображение, согласно которому задается значение  $s_{ij}$  каждого ДП  $\pi_j \in \Pi$  в каждом ТС  $S_i \in S$ , т. е.  $s_{ij} = \Phi(S_i, \pi_j)$ .

Модель анализа ТС СТО при непрерывной форме представления ДП [1] имеет вид:

$$M_{OA}^* = \langle S, \Pi, L, \Phi^* \rangle,$$

где  $S$  – множество ТС, подлежащих распознаванию  $S = \{S_i | i = 1, m\}$ ;  $\Pi$  – множество ДП, на котором все ТС  $S_i \in S$  попарно различимы,  $\Pi = \{\pi_j | j = 1, n\}$ ;  $L$  – множество интервалов на вещественной чис-

ловой оси, задающих разброс значений ДП  $\pi_j \in \Pi$  в ТС  $S_i \in S$ ,  $L = \{l_{ij} | i = 1, m; j = 1, n\}$ ;  $\Phi^* : S \times \Pi \rightarrow L$  – отображение, согласно которому задается интервал значений  $l_{ij}$  каждого ДП  $\pi_j \in \Pi$  в каждом ТС  $S_i \in S$ , т. е.  $l_{ij} = \Phi^*(S_i, \pi_j)$ .

Модели ОА удобно представить в виде следующих таблиц.

Таблица 1

**Табличное представление модели ОА при дискретной форме ДП**

	$\Pi_1$	...	$\Pi_n$
$S_1$	$S_{11}$	...	$S_{1n}$
...	...	...	...
$S_m$	$S_{m1}$	...	$S_{mn}$

Таблица 2

**Табличное представление модели ОА при непрерывной форме ДП**

	$\Pi_1$	...	$\Pi_n$
$S_1$	$(l_{11}; r_{11})$	...	$(l_{1n}; r_{1n})$
...	...	...	...
$S_m$	$(l_{m1}; r_{m1})$	...	$(l_{mn}; r_{mn})$

В работе предполагается, что объект управления (ОУ) и объект анализа физически являются одним СТО. Сформулируем критерии совместимости с моделью ОА для модели ОУ:

1. ТС ОУ должны совпадать с ТС ОА.
2. ДП ОУ должны совпадать с ДП ОА.
3. Результат управления изменяет значений ДП.

Выполнение критериев совместимости гарантирует, что просиходит именно управление ТС СТО.

Модель объекта управления ТС СТО имеет вид:

$$M_{Oy} = \langle S, \Pi, F, V, U \rangle,$$

где  $S$  – множество ТС, подлежащих распознаванию  $S = \{S_i | i = 1, m\}$ ;  $\Pi$  – множество ДП, на котором все ТС  $S_i \in S$  попарно различимы,  $\Pi = \{\pi_j | j = 1, n\}$ ;  $F$  – множество функций управления признаками

$$F = \{f_k | k = 1, l; D(f_k) \in Y^{(n)}, E(f_k) \in Y^{(n)}\};$$

$V : Y^{(n)} \rightarrow S$  – отображение, задающее ТС вектору значений признаков ОУ;  $U : S \rightarrow R \times F$  – отображение, согласно которому задается множество возможных функций управления и стоимость управления в каждом ТС.

Отображение  $V$  обязано быть сюръекцией, иначе модель имеет ненаблюдаемые ТС.

В моделях ОА и ОУ ТС полностью определяется по значениям ДП. Объект реагирует на сигналы управления  $u \in U$ , реакция на управляющий сигнал отражается в изменениях значений ДП.

Для каждого ТС могут быть заданы разные допустимые управляющие сигналы, так как ТС отражает функциональное состояние СТО [2].

Модель ОУ, также, как и модели ОА, можно представить в виде таблиц управляющих сигналов в каждом ТС.

Таблица 3

**Модель ОУ**

$S_i$	$\pi_1$	...	$\pi_n$	$C(u_i)$
$u_1$	$f_{11}(y_1)$	...	$f_{1n}(y_n)$	$c_1$
...	...	...	...	...
$u_m$	$f_{m1}(y_1)$	...	$f_{mn}(y_n)$	$c_m$

Модель процесса анализа [1] представляет собой кортеж:

$$M_{ПА} = \{Y, \Omega, P, \hat{\Pi}, R\},$$

где  $Y$  – множество зарегистрированных значений  $y_i$  ДП, значения  $y_j \in R, \forall j = 1, n$  являются вещественными числами, равномерно распределенных на заданных интервалах  $I_{ij}, Y = \{Y | Y = (y_1, y_2, \dots, y_n)^T\}$ ;  $\Omega$  – множество информационных состояний (ИС) моделируемого процесса анализа,  $\Omega = \{R | R \subseteq S\}$ ;  $P$  – вероятностная мера на множестве  $\Omega, P = \{P(R) | R \in \Omega\}$ ;  $\hat{\Pi}$  – множество проверок ДП,  $\hat{\Pi} = \{\hat{\pi}_j | j = 1, n\}$ ;  $R$  – множество решений о ТС объекта,  $R = \{R_i^* | i = 1, m\}$ .

ИС отражает возможные ТС на этапе процесса анализа. Одноэлементное ИС является решением о ТС СТО. В начале процесса анализа ИС состоит из всех возможных ТС СТО, т. е. ТС полностью неопределено.

Основным инструментом получения измерительной информации является проверка. Под проверкой  $\hat{\pi}_j \in \hat{\Pi}$  понимается совокупность операций, связанных с измерением текущего значения  $y_j$  признака  $\pi_j \in \Pi$  и сопоставлением измеренного значения  $y_j$  с модельными значениями данного признака. Исходом проверки  $\hat{\pi}_j$  называется событие совпадения  $y_j$  с модельным значением  $s_{ij}$  признака  $\pi_j$  или событие попадания  $y_j$  в интервал  $I_{ij}$ .

По исходам проверок принимается решение  $R_j^* \in R$  в пользу распознаваемого ТС  $S_j \in S$ .

Отдельная проверка  $\hat{\pi}_j$  при исходе  $(v = 1, \omega_{kj})$  переводит процесс анализа из некоторого ИС  $R_k \subseteq S$  в состояние  $R_{kj}^v \subset R_k$ , реализую отображение

$$\hat{\pi}_k : R_k \rightarrow R_{kj}^v, y_i = s_{ij}^v (v = 1, \omega_{kj}).$$

Процесс анализа происходит в соответствие с, обычно оптимальной в некотором смысле, синтезированной гибкой программой анализа (ГПА) ТС СТО. Оптимизируемые метрики и алгоритмы генерации ГПА ТС СТО описаны в [1].

Модель процесса управления представляет собой кортеж:

$$M_{ПУ} = \{Y^{(n)}, Y_0, Y_k, \hat{U}, C\},$$

где  $Y^{(n)}$  – пространство значений признаков;  $Y_0$  – множество зарегистрированных значений  $y_j$  ДП, в которых инициирован процесс управления;  $Y_k$  – множество целевых значений  $y_j$  ДП, которых требуется достигнуть в процессе управления;  $\hat{U}$  – множество управляющих сигналов

$$\hat{U} = \{ \hat{u}_k | k = 1, l; D(u_k) \in Y^{(n)}; E(u_k) \in Y^{(n)} \};$$

$C : \hat{U} \rightarrow R$  – функция стоимости управления.

Для решения задачи синтеза программы управления (ПУ) необходимо определить множество недопустимых и допустимых ТС. Когда в процессе анализа ТС определяется как недопустимое, инициируется управление с целью перевода ОУ в допустимое ТС.

Задача синтеза ПУ ТС СТО – найти вектор (программу) управления  $U_{ПУ} = \{ \hat{u} \}$  такой, что

$$\hat{u}_n(\hat{u}_{n-1}(\dots \hat{u}_0(Y_0))) = Y_k,$$

т. е. перевести объект управления из начального состояния в целевое.

Искомый вектор управления должен быть оптимален в смысле минимума:

$$C(U_0) = \sum_{\hat{u}_i} C(\hat{u}_i) \rightarrow \min,$$

ПУ ТС СТО решает задачу поиска оптимальной последовательности управляющих сигналов, а не вывода функции управления.

Приведем диаграмму взаимодействия субъекта анализа/управления (СА/СУ) с ОА/ОУ (рис. 1).

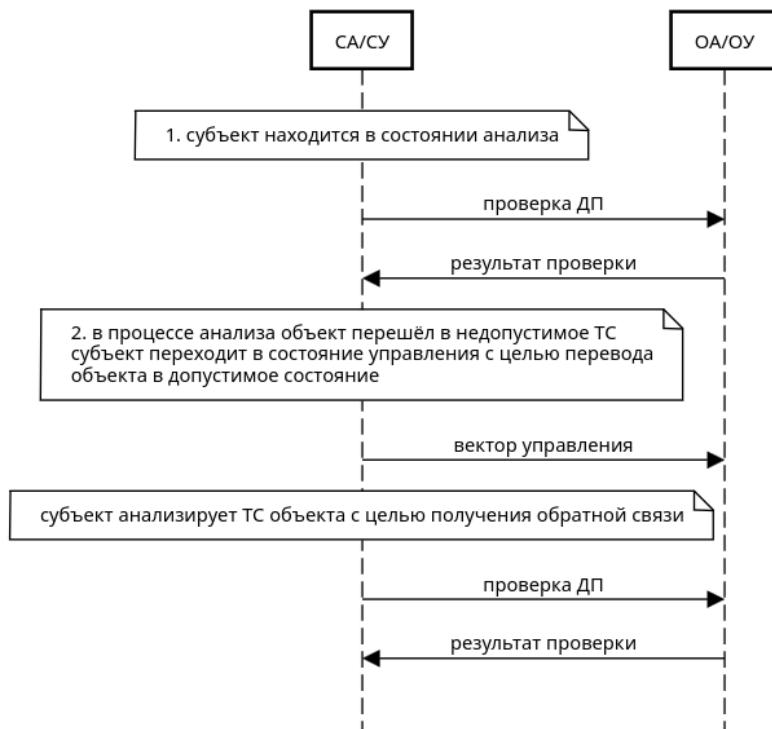


Рис. 1. Диаграмма взаимодействия с ОА/ОУ

Композиция моделей анализа и управления дает модель управления с обратной связью. Наличие обратной связи позволяет сформулировать задачу адаптации ПУ ТС СТО:

$U_{ПУ} = \{\hat{U}\}$  – синтезированная ПУ ТС СТО;

$Y_0$  – начальное состояние, в котором применена ПУ ТС СТО;

$Y_k$  – ожидаемое конечное состояние;

$Y_k^r$  – реальное конечное состояние.

Для адаптации ПУ ТС СТО необходимо обновить модельные (ожидаемые) результаты управляющих воздействий в соответствии с реальным конечным состоянием, т.е. реализовать отображение:

$$A: U \times Y^{(n)} \times Y^{(n)} \times Y^{(n)} \rightarrow U^*,$$

такое, что  $A(U_{ПУ}, Y_0, Y_k, Y_k^r) = U_{ПУ}^*: U_{ПУ}^*(Y_0) = Y_k^r$ , где  $U^*$  – обновленное (адаптированное) множество управляющих сигналов.

В данной статье разработаны модели объекта управления и процесса управления, совместимые с существующими моделями анализа. Сформулирована задача синтеза оптимальной программы управления, адаптации синтезированной программы управления.

#### Библиографический список

1. Анализ технического состояния космических средств: учебник / В. А. Чикуров, В. В. Шмелев, Е. В. Копкин, В. В. Мышко, В. В. Алейник. СПб.: ВКА имени А.Ф. Можайского, 2019. 399 с.
2. ГОСТ 27.002-2015. Надежность в технике. Термины и определения. М.: Изд-во стандартов, 2015. С. 28.



УДК 004.942

**Е. В. Кустова**

студент кафедры компьютерных технологий и программной инженерии

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

## АЛГОРИТМИЧЕСКИЕ И МАТЕМАТИЧЕСКИЕ ПОДХОДЫ ПРИ ИССЛЕДОВАНИИ ИММУНОЛОГИЧЕСКИХ МОДЕЛЕЙ

### Введение

Будут рассмотрены основные алгоритмы на примере базовой инфекционной модели, которые используются при работе с моделями инфекционных заболеваний, а также основные принципы метода синергетической теории управления – метода аналитического конструирования агрегированных регуляторов (АКАР).

Целью работы является разработка математического аппарата системы поддержки принятия врачебных решений при лечении инфекционных заболеваний на примере базовой инфекционной модели и модели гепатита В.

### Базовая инфекционная модель

Модели инфекционных заболеваний – продолжение идеи модели «хищник-жертва». Одной из множества моделей является обобщенная модель инфекционного заболевания, составленная Г. И. Марчуком [1, 2]. Общая безразмерная математическая модель сформулирована в виде задачи Коши для системы дифференциальных уравнений с запаздывающим аргументом, описывающей изменение четырех переменных состояния [3, 4]:

$$\begin{aligned} \dot{V} &= \underbrace{(a_1 - a_2 F)}_{\text{жертвы}}, \\ \dot{C} &= a_3 \xi(m) F(t - \tau) V(t - \tau) - a_5 (C - 1) \\ \dot{F} &= \underbrace{a_4 (C - F) - a_8}_{\text{хищники}}, \\ \dot{m} &= a_6 V - a_7 m. \end{aligned} \quad (1)$$

где  $a_i, i = 1, \dots, 8$  – параметры базовой модели;  $\tau$  – время, требуемое для формирования плазматических клеток;  $V$  – концентрация антигенов;  $F$  – концентрация антител;  $C$  – концентрация плазматических клеток;  $m$  – относительная характеристика пораженного органа;  $\xi(m)$  – функция, которая отвечает за производительность выработки антител при поражении жизненно важных органов, в которой  $m^*$  – граница работоспособности органа.

### Алгоритмы численного моделирования. Метод шагов

Все дифференциальные уравнения решаются с помощью методов численного приближения, но поскольку в модель входит запаздывание  $\tau$ , необходимо использовать специальные методы. Одним из таких методов является **метод шагов** (метод последовательного интегрирования), который позволяет решить дифференциальное уравнение как обычное дифференциальное уравнение (ДУ) без запаздывания. Его идея заключается в следующем [5]:

Пусть дано ДУ первого порядка:

$$\dot{x} = F(t, x, x(t - \tau)), \quad x(t) = \varphi(t), t \leq 0 \quad (2)$$

Для решения ДУ с запаздывающим аргументом можно применить метод шагов, используя численную схему Рунге-Кутты к объекту (2):

$$k_i^1 = \begin{cases} hf(t_i + h, x_i, \varphi(t_i - \tau)), & i < m \\ hf(t_i + h, x_i, x_{i-m}), & i \geq m \end{cases},$$

$$k_i^2 = \begin{cases} hf(t_i + h, x_i + k_i^1, \varphi(t_i - \tau + h)), & i < m \\ hf(t_i + h, x_i + k_i^1, x_{i-m} + k_{i-m}^1), & i \geq m \end{cases},$$

$$x_{i+1} = x_i + \frac{1}{2}(k_1 + k_2),$$

где  $\tau = hm$ ,  $h$  – шаг дискретизации,  $m$  – некоторое целое положительное число.

### Моделирование вмешательства в модель, синергетическая теория управления

**Управление** – вмешательство в динамику математической модели, воздействие, целью которого является стабилизация количества антигенов  $V$  в окрестностях некоторого значения  $V^*$ , например  $V^* = 0$  (или некоторой бесконечно малой константе):

$$V(t) - V^* \rightarrow 0, t \rightarrow \infty, \tag{3}$$

причем управление  $u$  должно удовлетворять следующему свойству:

$$0 \leq u(t) \leq b \in [0, T],$$

где  $b$  – физиологически допустимые дозы применения лекарственных препаратов;  $T$  – конец ввода препаратов по времени.

В синергетической теории управления предложен новый подход в понимании сущности процессов управления, предлагая осуществить переход к самоорганизации системы: все переменные системы осуществляют целевое направленное перемещение в фазовом пространстве вдоль желаемых инвариантных многообразий [6].

### Метод АКАР

Задачей метода АКАР является синтез управления  $u(t)$ , удовлетворяющего следующему функционалу качества для непрерывного случая [7]:

$$J_1 = \int_0^{\infty} (T^2 \dot{\psi}^2 + \psi^2) dt \rightarrow \min, \tag{4}$$

где  $\psi$  – притягивающее многообразие.

Для уменьшения повреждения организма при острых формах заболеваний используется иммунотерапия [4]. Рассмотрим описание объекта (1), применительно к задаче управления, где соответствующая функция  $f$  содержит в себе правую часть уравнения:

$$\begin{cases} \dot{x}_1 = f_1(x_1, x_3) \\ \dot{x}_2 = f_2(x_1, x_2, x_3, x_4) \\ \dot{x}_3 = f_3(x_1, x_2, x_3) + u \\ \dot{x}_4 = f_4(x_1, x_4) \\ x_1 = V, x_2 = C, x_3 = F, x_4 = m \end{cases}. \tag{5}$$

Цель управления заключается в обеспечении асимптотической стабилизации функции  $x_1$  в окрестности заданного значения  $x_1^*$ :

$$\psi = x_1 - x_1^*. \quad (6)$$

Приведем алгоритм синтеза управления для объекта (3.3), приводящий к цели (6) с помощью метода АКАР:

**Шаг 1.** Передадим управление от переменной  $x_3$  к вспомогательной функции  $\phi$ , называемой в методе АКАР внутренним управлением, которая находится на конечном этапе синтеза:

$$\psi_1 = x_3 - \phi. \quad (7)$$

Получим общую структуру управления. Структура управления (согласно АКАР) определяется на основе функционального уравнения  $T_1 \dot{\psi}_1 + \psi_1 = 0$ . Из данного уравнения следует:

$$u = -\psi_1 T_1^{-1} + \dot{\phi} - f_3(x_1, x_2, x_3).$$

**Шаг 2.** Проведем декомпозицию системы на достигнутом многообразии  $\psi_1 = 0, t \rightarrow \infty$ :

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, \phi), \\ \dot{x}_3 &= \phi. \end{aligned}$$

Определим структуру внутреннего управления  $\phi$  из решения уравнения для первоначальной цели  $T_2 \dot{\psi} + \psi = 0$ , где  $\psi = x_1 - x_1^*$ :

$$\phi = \frac{-\psi T_2^{-1} - a_1 x_1}{-a_2 x_1}.$$

Найдем полную производную  $\dot{\phi}$ :

$$\dot{\phi} = \frac{d\phi}{dx_1} \frac{dx_1}{dt} = \frac{x_1^*}{-a_2 T_2 x_1^2} \dot{x}_1.$$

Итоговая система:

$$\begin{aligned} \psi &= x_1 - x_1^*, \psi_1 = x_3 - \phi, \\ \phi &= \frac{-\psi T_2^{-1} - a_1 x_1}{-a_2 x_1}, \dot{\phi} = \frac{x_1^*}{-a_2 T_2 x_1^2} \dot{x}_1, \\ u &= -\psi_1 T_1^{-1} + \dot{\phi} - f_3(x_1, x_2, x_3), \\ \dot{x}_1 &= f_1(x_1, x_3), \dot{x}_2 = f_2(x_1, x_2, x_3, x_4), \\ \dot{x}_3 &= f_3(x_1, x_2, x_3) + u, \dot{x}_4 = f_4(x_1, x_4). \end{aligned} \quad (8)$$

### Результаты численного моделирования

Значения коэффициентов системы, использованных при моделировании:

$$a_1 = 2, a_2 = 0.8, a_3 = 10000, a_4 = 0.17, a_5 = 0.5, a_6 = 10, a_7 = 0.12, a_8 = 8$$

$$m^* = 0.5, \tau = 0.5, h = 0.01, T_1 = 1, T_2 = 1, x_1^* = 0$$

$$x_1(0) = 1e^{-6}, x_2(0) = 1, x_3(0) = 1, x_4(0) = 0$$

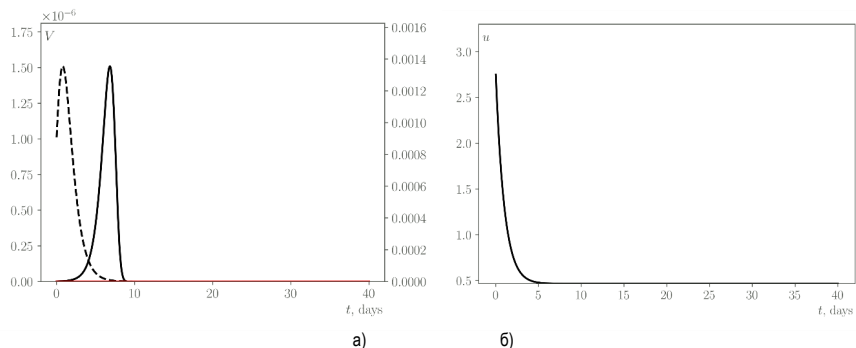


Рис. 1. Концентрация антигенов при острой стадии по времени (сутки), где черная сплошная линия – система без управления, черная штрихованная – система с управлением, красная сплошная – целевое значение (а); управление (б)

Значения коэффициентов системы, использованных при моделировании:

$$a_1 = 1.54, a_2 = 0.77, a_3 = 880, a_4 = 0.15, a_5 = 0.5, a_6 = 12, a_7 = 0.12, a_8 = 8;$$

$$m^* = 0.5, \tau = 2.5, h = 0.01, T_1 = 1, T_2 = 1, x_1^* = 0;$$

$$x_1(0) = 1e^{-6}, x_2(0) = 1, x_3(0) = 1, x_4(0) = 0.$$

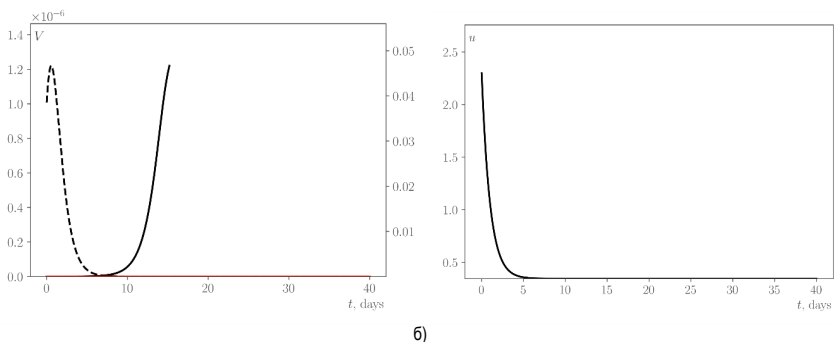


Рис. 2. Концентрация антигенов при летальной стадии по времени (сутки), где черная сплошная линия – система без управления, черная штрихованная – система с управлением, красная сплошная – целевое значение (а); управление (б)

### Заключение

Была синтезирована система для управления непрерывной базовой инфекционной моделью. Представлены результаты моделирования острой и летальной стадий заболевания, при которых управление успешно приводит концентрацию антигенов в организме к выбранной константе  $x_1^*$  ( $x_1^* = 1e^{-15}$ ) за конечное число дней и предотвращает летальный исход во втором случае. По достижении цели управление стабилизируется в окрестности константы.

Работа выполнена, как часть гранта РФФИ №23-29-00336.

**Библиографический список**

1. Марчук Г. И. Математическое моделирование в иммунологии и медицине / Отв. ред. Г. А. Бочаров // Избранные Труды. М.: РАН, 2018. 650 с.
2. Марчук Г. И. Математическое моделирование противовирусного иммунного ответа при вирусном гепатите В / Г. И. Марчук, А. А. Романюха, Г. А. Бочаров; под ред. С. В. Яблонского // Математические вопросы кибернетики. Вып. 2. М.: Наука, 1989. С. 5–70.
3. Чирков М. В. Методика моделирования управления процессом иммунного ответа в условиях неопределенности: дис. ... канд. ф-м. наук: 05.13.18 / Пермский государственный национальный исследовательский университет; Чирков Михаил Владимирович. Пермь, 2017. 133 с.
4. Болодурина И. П. Управление динамикой взаимодействия иммунной системы человека с инфекционными заболеваниями / И. П. Болодурина, Ю. П. Луговскова, Д. А. Закиров // Вестник ТвГУ. 2009. С. 101–117.
5. Хайрер Э., Нерсетт С., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи. М.: Мир, 1990. 512 с.
6. Колесников А. А. Синергетическая теория управления: концепции, методы, тенденции развития // Известия Южного федерального университета. 2001. С. 7–27.
7. Колесников А. А., Колесников Ал. А., Кузьменко А. А. Методы АКАР и АКОР в задачах синтеза нелинейных систем управления // Мехатроника, автоматизация, управление. 2016. С. 657–669.

УДК 004.94

**Н. А. Малинов**

студент кафедры компьютерных технологий и программной инженерии

**В. Н. Коромысличенко** – кандидат технических наук, доцент – научный руководитель

## РАЗРАБОТКА ВИДЕОИГР С ПОМОЩЬЮ ОНТОЛОГИЙ

### Введение

В современном мире разработка приложений, программ и устройств часто сталкивается с проблемой понимания требований заказчика. Большинство бизнесменов не обладает глубокими знаниями программирования, что приводит к непониманию, задержкам и постоянным изменениям в процессе разработки. Чтобы улучшить коммуникацию между разработчиком и заказчиком и снизить издержки, можно использовать онтологические модели. В данном случае рассматривается проект разработки видеоигры. Видеоигры стали неотъемлемой частью нашей культуры. От простой игры Tennis for Two, где играли в теннис с помощью осциллографа, мы продвинулись до VR-игр, где люди полностью погружаются в игровой мир с помощью шлемов виртуальной реальности и даже специальных беговых установок.

За последние 60 лет произошел огромный скачок в развитии видеоигр. Видеоигры популярны во всех возрастных группах: от обычных людей, решающих головоломки в автобусе, до профессиональных киберспортсменов, для которых игры стали основным занятием и возможностью достичь огромного заработка и популярности. Инвестиции на миллионы долларов, множество рекламных предложений и бизнесы, ориентированные на аренду игровых компьютеров, приносят огромные доходы.

Разработка игр связана со множеством проблем и сложностью подходов к ним. В данной статье будет рассмотрена разработка видеоигр с использованием онтологий.

### Известные подходы в области разработки видеоигр

Существует несколько методов разработки видеоигр [1]:

**Каскадная модель.** Эта модель основана на последовательном выполнении этапов разработки, начиная с определения требований, проектирования, реализации, тестирования и завершения игры. Каждый этап завершается до начала следующего, и изменения вносятся с трудом после завершения этапа.

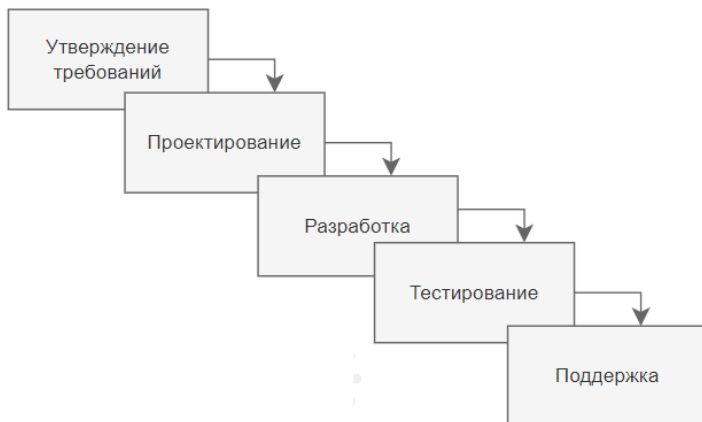


Рис. 1. Схема каскадной модели разработки видеоигры

*Плюсы каскадной модели разработки:*

Структурированность. Каскадная модель предлагает последовательный подход к разработке, где каждый этап завершается перед переходом к следующему.

Предсказуемость и планирование. Каскадная модель позволяет лучше планировать проект, так как требования и этапы разработки определяются заранее.

Документация. Каскадная модель ставит акцент на создание документации на каждом этапе разработки. Это позволяет сохранить записи и описания процесса разработки, требований и решений, что полезно для поддержки и обслуживания игры в будущем.

Легкая отслеживаемость. Поскольку каждый этап завершается до перехода к следующему, прогресс разработки легко отслеживать и контролировать.

*Минусы каскадной модели разработки:*

Ограниченная гибкость. Каскадная модель не предусматривает гибкого реагирования на изменения требований или обратную связь. Поскольку каждый этап завершается до перехода к следующему, внесение изменений после завершения этапа может быть сложным и затратным.

Недостаточное вовлечение заказчика. Каскадная модель ориентирована на однонаправленное движение от требований заказчика к завершенной игре. Это может ограничить вовлечение заказчика в процесс разработки и привести к непониманию или несоответствию между ожиданиями заказчика и финальным продуктом.

Ограниченная обратная связь. В каскадной модели обратная связь возможна в основном на начальных этапах разработки, что может ограничить возможности улучшения и оптимизации проекта в процессе разработки.

Риски задержек. Если на одном из этапов возникнут проблемы или задержки, они могут сказаться на всем процессе разработки, так как последующие этапы зависят от успешного завершения предыдущих.

Каскадная модель разработки имеет свои преимущества в структурированности и планировании процесса разработки, однако ограниченная гибкость и возможность внесения изменений могут быть ее недостатками в динамичной среде разработки видеоигр.

**Итеративная модель.** В этом методе разработка происходит через серию итераций. На каждой итерации разработчики выполняют определенные этапы, такие как проектирование, разработка, тестирование и оценка, и затем возвращаются к предыдущим этапам для внесения изменений и улучшений.

Итеративная модель разработки видеоигр также имеет свои плюсы и минусы.

*Плюсы итеративной модели разработки:*

Гибкость и возможность внесения изменений. Итеративная модель позволяет разработчикам быстро реагировать на изменения требований и обратную связь. Каждая итерация предоставляет возможность внести корректировки и улучшения в игру, что способствует более гибкому и адаптивному процессу разработки.

Постепенное совершенствование. Итеративная модель позволяет постепенно улучшать функциональность и качество игры с каждой итерацией. Команда разработчиков может сосредоточиться на наиболее значимых и сложных аспектах игры, чтобы достичь лучших результатов.

Более высокая вероятность успеха. Поскольку каждая итерация проходит через процесс разработки, тестирования и оценки, это уменьшает риск возникновения крупных проблем или несоответствия ожиданиям заказчика. Более раннее выявление проблем и их регулярное исправление повышают вероятность успешной разработки игры.

Улучшенное вовлечение заказчика. Итеративная модель позволяет более активное вовлечение заказчика в процесс разработки. Заказчик имеет возможность видеть промежуточные результаты и давать обратную связь на ранних этапах, что способствует лучшему пониманию требований и удовлетворению их потребностей.

*Минусы итеративной модели разработки:*

Возможность потери фокуса. Постоянные изменения и итерации могут привести к потере фокуса и неопределенности в процессе разработки. Если не установлены четкие приоритеты и контроль

изменений, это может привести к затягиванию процесса или неэффективному использованию ресурсов.

Необходимость управления временем и бюджетом. Частые итерации и внесение изменений могут повлиять на управление временем и бюджетом проекта. Необходимо тщательно планировать и контролировать время и ресурсы, чтобы избежать перерасхода и несоответствия расписанию.

Необходимость тщательного тестирования. Поскольку каждая итерация вносит изменения, необходимо проводить тщательное тестирование и проверку каждого обновления. Это может потребовать дополнительных усилий и времени для обеспечения качества игры.

Не подходит для всех проектов. Итеративная модель может быть более сложной и неэффективной для небольших и простых проектов, где требования стабильны и просты. Она лучше применима в случаях, когда требования меняются или сложны и требуют дополнительного исследования и экспериментов.

В целом, итеративная модель разработки видеоигр предоставляет гибкость и возможность вносить изменения, но требует хорошего управления и внимательного планирования, чтобы извлечь максимальные выгоды и справиться с ее ограничениями.

**Гибкая модель разработки (Agile) [5].** Agile-методологии, такие как Scrum и Kanban, предлагают итеративный и гибкий подход к разработке, где команды работают в краткосрочных циклах разработки, известных как спринты. Каждый спринт включает планирование, разработку, тестирование и оценку, и команда гибко реагирует на изменения и обратную связь.

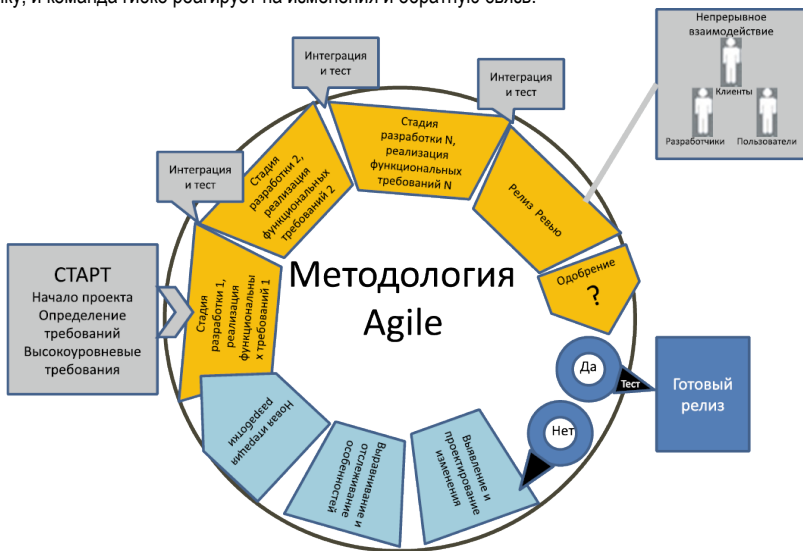


Рис. 2. Схема гибкой модели разработки

Гибкая модель разработки, имеет свои плюсы и минусы.

**Плюсы гибкой модели разработки:**

Гибкость и адаптивность. Гибкая модель позволяет быстро реагировать на изменения требований и обратную связь. Команда разработчиков может легко адаптироваться к новым условиям и решать поставленные задачи по мере их возникновения.

Вовлечение заказчика. Гибкая модель активно вовлекает заказчика в процесс разработки. Заказчик имеет возможность видеть промежуточные результаты, вносить изменения и давать обратную связь на каждом этапе разработки.



Быстрые результаты. Гибкая модель разработки предлагает короткие итерации и регулярные выпуски программ. Это позволяет получать быстрые результаты и получать обратную связь от пользователей или заказчика раньше, что помогает лучше удовлетворить их потребности и предпочтения.

Снижение рисков. Благодаря постоянному тестированию, обратной связи и корректировке, гибкая модель помогает своевременно выявлять и устранять проблемы и риски.

*Минусы гибкой модели разработки:*

Необходимость активного участия и коммуникации. Гибкая модель требует активного участия всех участников команды и постоянной коммуникации. Если команда неэффективно работает или несостоятельна в коммуникации, это может привести к затягиванию процесса или неправильному пониманию требований.

Высокая зависимость от команды разработчиков. Гибкая модель требует команды разработчиков, которая обладает высоким уровнем навыков и способностей. Если команда не имеет достаточной компетентности или опыта, это может негативно сказаться на процессе разработки и качестве конечного продукта.

Неопределенность в планировании. Гибкая модель может создать неопределенность в планировании проекта. Постоянные изменения и приоритеты могут затруднить определение конечных сроков и бюджета. Это требует более гибкого и динамического подхода к планированию и управлению проектом.

Не подходит для всех типов проектов. Гибкая модель разработки может быть менее эффективной для проектов с жесткими требованиями, строгими сроками или высокими рисками без возможности частых корректировок. В таких случаях более прогнозируемые и структурированные модели могут быть предпочтительными.

В целом, гибкая модель разработки предоставляет гибкость и адаптивность, но требует активного участия команды и хорошего управления проектом для успешного применения.

#### **Метод онтологий в разработке видеоигр**

Онтологический подход в разработке видеоигр основан на использовании формализованных онтологических моделей. Онтология представляет собой явное описание понятий, связей и правил в игровом мире, которое помогает лучше понять структуру игры и взаимосвязи между ее компонентами.

Разработчики создают онтологическую модель игрового мира, в которой описываются объекты, персонажи, события, правила взаимодействия и другие элементы игры. Онтология может быть представлена с помощью языков описания знаний, таких как OWL (Web Ontology Language) [2], RDF (Resource Description Framework), а также можно использовать реляционные базы данных [3].

Следует понимать, какими *преимуществами* обладает онтологическая модель разработки видеоигры, в сравнении с другими моделями:

Улучшенное понимание требований. Онтологическая модель позволяет более точно определить понятия, отношения и связи в предметной области разработки видеоигр. Это помогает лучше понять требования заказчика и создать более точное представление о конечном продукте.

Улучшенное взаимодействие. Онтологическая модель способствует улучшению коммуникации и сотрудничеству между разработчиками, заказчиками и другими участниками проекта. Общая модель понимания, определенная в онтологии, помогает снизить непонимания, уточнить требования и улучшить обмен информацией.

Повышенное управление знаниями. Онтологическая модель позволяет формализовать и структурировать знания и экспертизу в области разработки видеоигр. Она предоставляет способ организации и хранения знаний, а также их повторного использования в различных проектах.

Высокая гибкость и адаптивность. Онтологическая модель может быть легко расширена и адаптирована для новых требований и изменений. Ее модульная структура позволяет добавлять и изменять понятия и отношения без необходимости изменения всей модели разработки.

Качество и надежность. Благодаря более четкому пониманию требований и улучшенной коммуникации, онтологическая модель помогает повысить качество и надежность разрабатываемого продукта. Это снижает риск ошибок и несоответствий, а также улучшает удовлетворенность заказчика.

Обладая своими сильными преимуществами, *онтологический метод [5] так же имеет некоторые недостатки:*

Сложность и время разработки онтологии. Разработка и создание онтологии требует значительного времени, усилий и экспертизы. Определение понятий, отношений и иерархий может быть сложным и требовать глубокого понимания предметной области. Это может затянуть процесс начала разработки проекта.

Требуется экспертиза в онтологических технологиях. Использование онтологической модели требует знаний и навыков в области онтологических технологий. Разработчики должны быть знакомы с концепциями и инструментами онтологической инженерии, чтобы эффективно создавать и использовать онтологию.

Необходимость поддержки и согласования сторон. Успешное использование онтологической модели требует согласования и поддержки всех заинтересованных сторон. Заказчики и разработчики должны быть готовы участвовать в процессе разработки онтологии и сотрудничать для обеспечения ее эффективного использования.

Ограничения формализации: в некоторых случаях сложно формализовать все аспекты разработки видеоигр с помощью онтологической модели. Некоторые концепции или ситуации могут быть трудными для формализации, что может привести к ограничениям в применении онтологии в разработке.

### **Заключение**

Онтологическая модель разработки по-своему сложна, но в то же время она способна вывести на высочайший уровень понимание разработчиков и заказчиков. Объединив все в формализованную структуру, недопонимание и неясности между командой художников, программистов, гейм-дизайнеров будет стремиться к нулю, а полученная онтология может помочь создать следующие продукты выше качеством, развить искусственный интеллект на базе полученных знаний, что в свою очередь способно подарить много новых ощущений простым пользователям.

### **Библиографический список**

1. Бабкин Э. А., Князькин В. П., Шиткова М. С. Сравнительный анализ языковых средств, применяемых в методологиях бизнес-моделирования // Бизнесинформатика. 2011. № 2 (16). С. 31–42.
2. Язык OWL. URL: <https://www.w3.org/TR/owl2-syntax> (дата обращения: 20.05.2023).
3. Кучуганов М. В. Синтез схем баз данных на основе онтологии // Онтология проектирования. 2016. № 4 (22). С. 475–484.
4. Методология Agile. URL: <https://scrumtrek.ru/blog/agile-scrum/4029/metodologiya-agile/> (дата обращения: 20.05.2023).
5. Онтологии и тезаурусы: модели, инструменты, приложения: учеб. пособие / Б. В. Добров [и др.]. М.: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. 172 с.

УДК 519.688

**А. А. Павлов**

студент кафедры компьютерных технологий и программной инженерии

**М. Ю. Охтилев** – доктор технических наук, профессор – научный руководитель**МОДЕЛИ И АЛГОРИТМЫ ОЦЕНИВАНИЯ ТЕХНИЧЕСКОГО СОСТОЯНИЯ СЛОЖНЫХ ТЕХНИЧЕСКИХ ОБЪЕКТОВ НА ОСНОВЕ СТРУКТУРНО-СТОХАСТИЧЕСКИХ МОДЕЛЕЙ****Введение**

Надежность и безопасность работы сложных технических объектов (СТО) во многом связаны с постоянным контролем надежности эксплуатационных свойств и параметров машины или ее отдельных составляющих. Проводимый при этом контроль необходимо осуществлять без демонтажа оборудования или выведения его из строя, поэтому задачи такого типа требуют скрупулезного подхода и тщательной проработки всех возможных путей решения.

Предпринимаются попытки внедрения концептуально новых технологий для осуществления таких задач. Также активно применяются способы комбинирования и усовершенствования уже существующих подходов и методов в целях оптимизации и замещения недостатков одних положительными сторонами других. Мотивацией таких решений всегда является получение наиболее точного и во всех смыслах эффективного результата. Одной из таких классических технологий анализа состояния сложных технических объектов является моделирование состояния технического объекта с использованием структурно-стохастических моделей, сочетающее в себе преимущества как структурных, так и статистических (стохастических) моделей [1, 2].

При существенном возрастании потоков измерительной информации, при усложнении задач распознавания становится очевидным громоздкость и неприемлемость традиционных дискриминантных методов. Альтернативным становится подход, использующий в процессе распознавания структурные свойства измерительной информации, т. е. отношения на множестве признаков, иерархичность и регулярность. Методы распознавания, базирующиеся на этом подходе, получили название структурных (синтаксических, лингвистических) методов.

Основная задача анализа информации, состоит в получении значений параметров в соответствии с целью анализа. При сборе измерительной информации вследствие разного рода неучтенных и непредсказуемых факторов, разрушающих информацию, возможны различные по природе искажения значений измеряемых параметров технического состояния объекта анализа. Это, в свою очередь, может повлиять на определение значений вычисляемых параметров ТС и, как следствие, к неправильному оцениванию параметров цели анализа. Случайный характер искажений в системах анализа информации приводит к необходимости использовать модели стохастического типа.

При использовании структурных (лингвистических) моделей функционирования объекта анализа такими моделями являются стохастические грамматики и стохастические языки, являющиеся структурно-стохастическими моделями функционирования объекта анализа.

**Теория распознавания образов**

Теория распознавания образов является наиболее распространенной теоретической практикой локальных задач оценки технического состояния (ТС). В рамках этой теории ядром используемых методов распознавания является отнесение распознаваемого объекта (технического состояния) к одному из классов. Множество классов может быть не только частью всего множества объектов распознавания, но и их объединением. В конечном счете, однако, решение о том, принадлежит ли объект к определенному классу, принимается с использованием некоего решающего правила, которое включает в себя разделение всего набора объектов.

Методология распознавания ТС ОА за длительную историю своего существования прошла масштабный путь развития и совершенствования. И все существующие на сегодняшний день методы использовали и улучшали имеющийся аппарат теории распознавания образов [3].

Все такие методы можно разделить на следующие группы:

- детерминированные;
- логические;
- статистические (стохастические);
- структурные.

В данной работе основное внимание уделяется комбинации последних двух методов из данного перечисления – структурно-стохастическим моделям.

### Структурно-стохастические модели

Стохастический вид структурных вычислительных моделей является их необходимым качеством в условиях наличия неопределенности, упомянутых выше. Стохастические грамматики, как один из случаев, подходят для формального описания данного тезиса.

Стохастической грамматикой (СГ) называется кортеж:

$$G_S = T, N, R, S, \Pi_S, \pi_S, \Pi_R, \pi_R,$$

где  $T$  – конечное множество терминальных символов;  $N$  – конечное множество нетерминальных символов;  $R$  – конечное множество правил вывода;  $S$  – конечное множество аксиом;  $\Pi_S$  – вероятностная мера:  $p \in \Pi_S \in [0;1]$ ;  $\pi_S$  – отображение, сопоставляющее каждому элементу  $A_j \in S$  его вероятностную меру  $p \in \Pi_S \in [0;1]$ ;  $\pi_S : S \rightarrow \Pi_S$ ;  $\Pi_R$  – вероятностная мера:  $p \in \Pi_R \in [0;1]$ ;  $\pi_R$  – отображение, сопоставляющее каждому правилу подстановки  $r \in R$  его вероятностную меру  $p \in \Pi_R \in [0;1]$ ;  $\pi_R : R \rightarrow \Pi_R$ .

Любая СГ порождает стохастический язык (СЯ)  $L(G_S) = L^S$ . Отличительной чертой такого языка является тот факт, что каждое его слово  $\alpha^S \in L^S$  имеет свою вероятностную меру, сумма вероятностей всех слов языка равна единице, а СГ, задающая этот язык, называется согласованной [4]:

$$\sum_{\alpha^S \in L^S} P(\alpha^S) = 1.$$

Из всего вышесказанного следует, что вероятность вывода любой цепочки языка  $\alpha^S \in L^S$  со словом вывода  $R_{\alpha^S}$  в предположении независимости шагов вывода определяется следующим образом:

$$\prod_{r_{\alpha^S} \in R_{\alpha^S}} r_{\alpha^S} \rightarrow \pi_r.$$

### Обучение структурно-стохастической модели

Пусть:

- $t_j$  – некоторый вычисляемый параметр ТС,  $t_j \in T$ ;
- $D_{t_j}$  – множество дискретных значений, которые принимает параметр  $t_j \in T$ ;
- $|D_{t_j}|$  – мощность множества значений параметра  $t_j$ .

Для восстановления стохастической грамматики  $G_S$  используется отношение обучения такое, что языку  $L^S$ , который она задает, соответствует множество пар:

$$\eta = \{ \alpha_j, m_j \mid \alpha_j \in L^S \},$$

где  $m_j$  – кратность слова  $\alpha_j$  в языке  $L^S$ .

Предполагается, что задача восстановления разрешима.

В простейшем случае, когда мощность  $|D_{t_j}| = 1$ , то отношение обучения задается единственным обучающим языком:

$$L^S = \{ \alpha_j, m_j \mid \alpha_j \in L^S \}.$$

Введем в рассмотрение следующие величины:

$\hat{n}_{ij}(\alpha_l)$  – случайную величину, равную числу вхождений правила  $A_i \rightarrow B_j$  в вывод цепочки  $\alpha_l$

для  $G_S$ :

$S \rightarrow \alpha_j$ ;

$\tilde{n}_{ij}(\alpha_l)$  – оценку математического ожидания случайной величины  $\hat{n}_{ij}(\alpha_l)$  – для всего обучающего языка  $L^S$ .

Тогда можно дать величине  $\tilde{n}_{ij}(\alpha_l)$  следующее определение [5]:

$$\tilde{n}_{ij} = \tilde{M}[\hat{n}_{ij}] = \frac{\sum_{\alpha_l \in L^S} m_l \hat{n}_{ij}(\alpha_l)}{\sum_{\alpha_l \in L^S} m_l}.$$

В условиях согласованности СГ  $G_S$  оценка вероятностей правил подстановок будет вычисляться по формуле:

$$\tilde{p}_{ij} = \frac{\tilde{n}_{ij}}{\sum_{i,j \in R^i} n_{ij}} = \frac{\sum_{\alpha_l \in L^S} m_l \tilde{n}_{ij}(\alpha_l)}{\sum_{i,j \in R^i} \sum_{\alpha_l \in L^S} m_l \tilde{n}_{ij}(\alpha_l)},$$

где  $I_{(R^i)}$  – множество индексов правил подстановок в  $R_S$  с левыми частями  $A_i \in N$ .

### Анализ технического состояния с использованием структурно-стохастической модели

Так как  $|D_{t_j}| = 1$ , имеется единственная СГ  $G_S$ , поэтому анализ входной цепочки  $\alpha^*$  сводится к определению ее вероятности в СЯ  $L(G_S)$ .

Вероятность цепочки  $\alpha^*$  можно определить следующим образом:

$$p(\alpha^*) = p(\alpha_V^*),$$

где  $\alpha_V^*$  – слово вывода цепочки  $\alpha^*$  в СГ  $G_S$ .

Слово вывода  $\alpha_V^*$  цепочки  $\alpha^*$  в ФГ  $G$  определяется как слово в алфавите

$$I_R = \{r_1, r_2, \dots\},$$

где  $r_i \in R$  – номер правила вывода в схеме ФГ  $G$ .

При этом

$$\alpha_V^* = r_{j_1} r_{j_2} \dots r_{j_k}$$

такое, что каждый элемент  $\alpha_V^*$  – есть номер того правила в  $R$ , которое применяется в терминальном выводе  $S \rightarrow \alpha^*$ , а порядок применения этих правил задается порядком следования символов  $r_i$  в слове вывода  $\alpha_V^*$  [4].



**Библиографический список**

1. Гросс А., Лантен А. Теория формальных грамматик / Пер. с франц. И. А. Мельчука; под ред. А. В. Гладкого. М.: Мир, 1971. 294 с.
2. Горелик А. Л., Гуревич И. Б., Скрипкин В. А. Современное состояние проблемы распознавания. Некоторые аспекты / Под ред. ДТН проф. А. Л. Горелика. М.: Радио и связь, 1985. 160 с.
3. Охтилев М. Ю. Системы искусственного интеллекта и их применение в автоматизированных системах мониторинга состояния сложных организационно-технических объектов. СПб.: ГУАП, 2018. 261 с.
4. Рышков Ю. П., Охтилев М. Ю., Богомолов С. Е., Шевченко А. Ю., Длужневский В. В., Петроченков В. В. Актуальные вопросы автоматизированной обработки и анализа информационных процессов. МО РФ, 1992. 140 с.
5. Фу К. Структурные методы в распознавании образов / Пер. с англ. Н. В. Завалишина, С. В. Петрова, Р. Л. Шейнина; под ред. М. А. Айзермана. М.: Мир, 1977. 320 с.

УДК 004.89

**П. Ю. Парфишов**

студент кафедры компьютерных технологий и программной инженерии

**Ю. А. Скобцов** – кандидат технических наук, профессор – научный руководитель

## НЕЙРОСЕТЕВОЙ МЕТОД РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ НА ПРИМЕРЕ ОБНАРУЖЕНИЯ ОРУЖИЯ

### Введение

С быстрым развитием машинного обучения и компьютерного зрения, нейросетевые методы распознавания изображений становятся все более популярными. Одно из возможных применений – обнаружение оружия на изображениях с помощью нейронных сетей.

Аэропорты, правительственные здания, школы и другие учреждения по всему миру стремятся обеспечить безопасность своих посетителей и персонала. В связи с этим, появление и развитие технологий, позволяющих автоматически обнаруживать оружие в реальном времени, стало чрезвычайно важным вопросом. Большинство современных систем обнаружения оружия основаны на использовании машинного обучения и нейросетевых методов [1]. Предлагаемый в данной статье метод основан на нейросетевой архитектуре Yolov7 и реализован с использованием Python и библиотеки PyTorch [2]. Данный метод демонстрирует высокую точность обнаружения оружия на изображениях и может быть применен в различных общественных местах для обеспечения безопасности.

### Текущие проблемы

Большинство существующих методов обнаружения оружия на изображениях основаны на классических моделях компьютерного зрения, таких как каскады Хаара и методы, использующие глубокое обучение [3]. Несмотря на успешное применение этих методов, они имеют несколько ограничений и проблем, которые необходимо преодолеть:

1. Нисходящий этап обнаружения объектов может приводить к снижению скорости и точности обнаружения, особенно при использовании сложных моделей.
2. Большая часть существующих методов требует значительного времени на обучение, особенно для сложных моделей.
3. Некоторые методы могут быть нестабильными при работе с различными типами изображений и условиями освещения.
4. Часто необходимы высокие вычислительные мощности для решения обнаружения оружия на изображениях, что может затруднять их применение в реальных сценариях.

Все эти ограничения делают необходимым введение новых методов и подходов, основанных на последних достижениях в области машинного обучения и компьютерного зрения.

### Подход к решению проблемы обнаружения оружия

Подход к обнаружению оружия на изображениях основан на нейронной сети Yolov7 [4]. Она эффективно распознает объекты на изображениях. Yolov7 использует инновационную архитектуру, которая одновременно анализирует все изображение и предсказывает класс и координаты объектов.

Архитектура Yolov7 основана на Darknet-53 и содержит 294 слоя. На вход нейросети поступают изображения, обрабатываемые для обнаружения объектов. Размер входных изображений может быть произвольным, но рекомендуется использовать разрешение, кратное 32. Нейросеть обучается на размеченных данных с помощью метода обучения с учителем. Для обучения Yolov7 используются различные методы оптимизации, такие как Adam, SGD, Adagrad, а также методы улучшения обучения, такие как OneCycleLR. Использование Yolov7 для обнаружения оружия на изображениях эффективно и позволяет достичь высокой точности и скорости работы.



## Процесс обучения модели компьютерного зрения

### Подготовка набора изображений

Для решения задачи обнаружения оружия на изображениях необходимо иметь набор изображений, на которых присутствует оружие. Для сбора такого набора можно использовать различные источники, например, открытые базы данных или собственные снимки. Важно, чтобы на изображениях были различные типы оружия, различные углы и освещение, чтобы модель могла обучаться на разнообразных примерах.

### Аннотирование изображений

Для обучения модели необходимо отметить каждое изображение тегом, указывающим на наличие или отсутствие оружия на изображении [5]. Для этого можно использовать специальное программное обеспечение, например, LabelImg. С его помощью можно создать файлы аннотаций в форматах XML или YOLO, которые будут использоваться для обучения модели.

### Процесс обучения нейронной сети

Для обучения модели на собранном наборе изображений и аннотациях необходимо выбрать архитектуру нейронной сети, например, YOLOv7. Затем необходимо настроить гипер-параметры модели, такие как learning rate, batch size, epoch number и т. д. После этого происходит обучение модели на выбранном наборе данных.

### Оценка результатов обучения

После обучения модели необходимо ее оценить. Для этого можно использовать такие метрики, как точность, полнота, F1-score. Можно также провести валидацию модели на тестовом наборе данных и оценить ее работу на реальных изображениях. Если результаты не удовлетворяют требованиям, необходимо повторить обучение с изменением гипер-параметров или выбором другой архитектуры нейронной сети.

## Проведение эксперимента

Для проведения эксперимента был создан набор для обучения, изображенный на рис. 1, и набор для проверки, изображенный на рис. 2. Изображения были взяты из открытых источников сети Интернет.

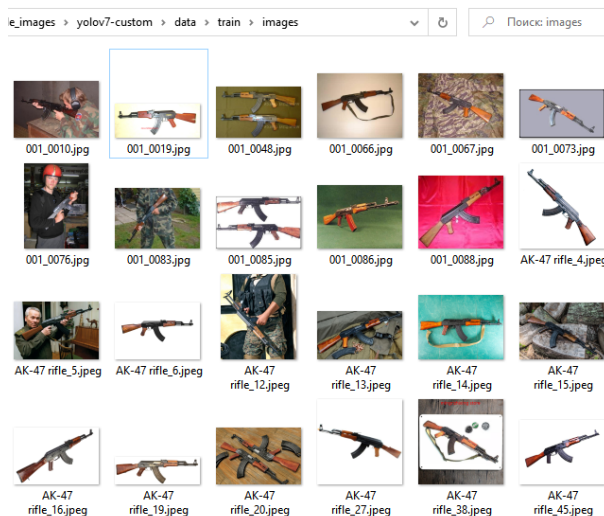


Рис. 1. Набор изображений для обучения (train)

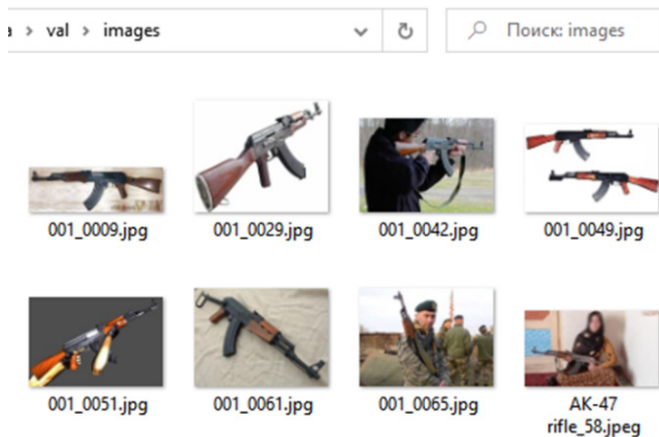


Рис. 2. Набор изображений для проверки (validation)

Затем была осуществлена аннотация получившихся наборов с помощью программы LabelImg. После этого на аннотированных изображениях обучалась нейронная сеть на архитектуре Yolov7 в течение 100 эпох.

### Результаты эксперимента

После обучения были получены изображения из набора для проверки с предсказанными рамками (рис. 3). Можно заметить, что на некоторых изображениях рамки предсказаны не были. Скорее всего это связано с недостаточным размером наборов изображений для обучения.

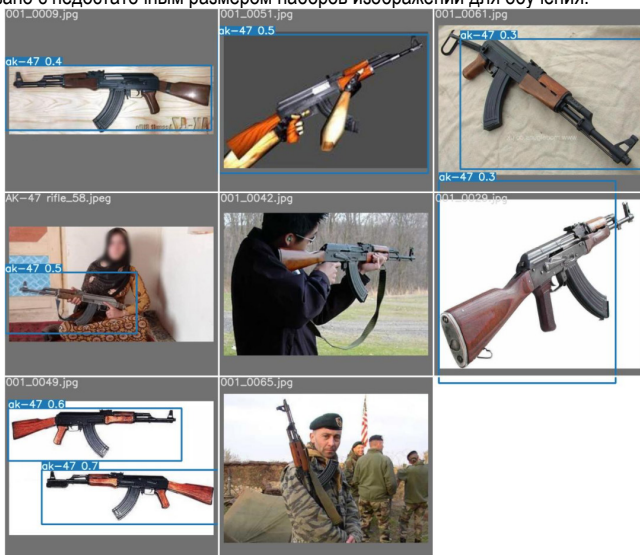


Рис. 3. Предсказанные рамки объектов из набора изображений для проверки

Построен график кривой точность-полнота (рис. 4), который используется для оценки качества выходных данных. Большая площадь под кривой свидетельствует о высокой полноте и точности модели.

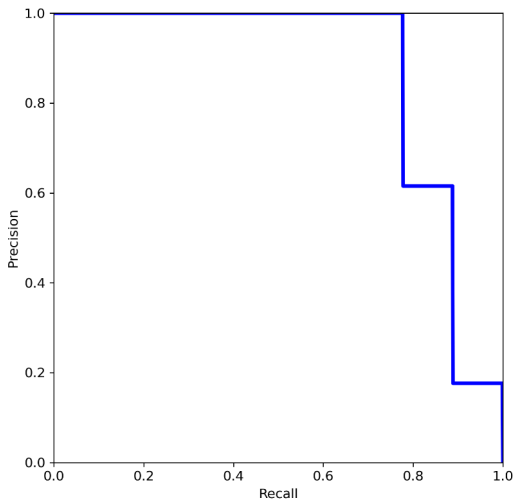


Рис. 4. Кривая точность-полнота

Обученная модель дополнительно была протестирована на видеозаписях. Модели удалось обнаружить автоматы на изображениях, однако вероятность принадлежности к классу “ak-47” оказалась довольно мала (рис. 5), что говорит о том, что требуется добавить в исходные наборы изображений для обучения больше вариаций автоматов в различных положениях.



Рис. 5. Кадры из видеозаписей с предсказанными рамками

Также был построен график mAP (mean Average Precision) для IoU (Intersection over Union) = 0,5, изображенный на рис. 6, который наглядно показывает, как увеличивалась средняя точность прогнозов модели в течение 100 эпох обучения.

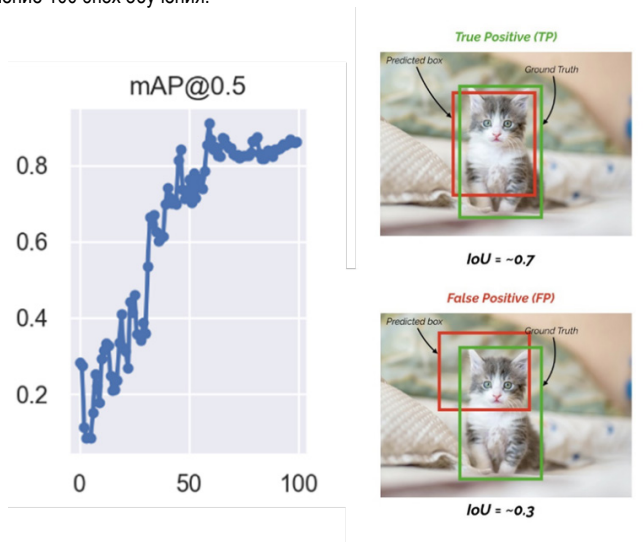


Рис. 6. mAP для эпох обучения при IoU = 0,5

### Выводы

В статье рассмотрен нейросетевой метод обнаружения оружия на изображениях с использованием архитектуры Yolov7, и представлен эксперимент, в котором использовалась обученная модель для распознавания оружия на новых изображениях. Результаты эксперимента показали высокую точность детектирования, но требуется доработка модели для решения реальных задач. Использование нейросетевых методов для распознавания изображений является перспективным направлением в различных областях, включая безопасность и охрану. Однако проблема распознавания объектов на изображениях требует дальнейших исследований и поиска надежных решений с высокой точностью и быстродействием.

### Библиографический список

1. Клетте Р. Компьютерное зрение. Теория и алгоритмы. М.: ДМК Пресс, 2019. 506 с.
2. Солям Я. Э. Программирование компьютерного зрения на Python. М.: ДМК Пресс, 2016. 312 с.
3. Селянкин В. В., Компьютерное зрение. Анализ и обработка изображений. СПб.: Лань, 2019. 152 с.
4. YOLO: Real-time object detection. URL: <https://pjreddie.com/darknet/yolo/> (дата обращения: 23.03.2023).
5. Image Processing, Analysis, and Machine Vision / M. Sonka, V. Hlavac, R. Boyle. UK: CL Engineering, 2007. 872 с.

УДК 519.248

**В. В. Поляков**

студент кафедры компьютерных технологий и программной инженерии

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

## ПРОГРАММА ДЛЯ ФАКТОРНОГО АНАЛИЗА ПРИЧИН ДТП НА ПЕРЕКРЕСТКАХ САНКТ-ПЕТЕРБУРГА

### Введение

Из года в год обстановка на дорогах городов характеризуется возрастанием уровня дорожно-транспортных происшествий (ДТП), обусловленным, в том числе, увеличением количества транспортных средств. Самым опасным средством передвижения, как правило, считается автомобиль.

Исходя из данных аналитического агентства «Автостат» [1] всего за 3 года с начала 2019 по начало 2022 количество легковых автомобилей на дорогах России увеличилось с 43,5 млн до 45,5 млн штук. При этом Санкт-Петербург занимает 4 место в списке регионов с самым большим количеством авто. По последним данным, на 2021 год в Санкт-Петербурге насчитывалось более 2,3 млн машин.

По данным ВОЗ ДТП входят в ТОП-10 главных причин гибели людей. Причины аварии можно условно разделить на следующие классы:

- нарушения в дорожных условиях (дефекты дорожного покрытия, неисправное освещение, несоответствие дорожных ограждений предъявляемым требованиям и т. д.);
- нарушения ПДД водителями и пешеходами;
- недостатки или поломки транспортных средств.

Для решения задачи повышения безопасности на дорогах необходимо уметь выявлять значимые факторы, влияющие на риски возникновения ДТП и обстановку в целом.

Факторы ДТП – это некоторые события, называемые причинами ДТП, определяющие характер ДТП и частоту его появления, или, другими словами, события, появление которых повышает вероятность возникновения ДТП. Определяя наличие и влияние таких факторов, можно спрогнозировать и предупредить возникновение опасных ситуаций на дорогах, путем воздействия на данные факторы.

Целью настоящего исследования является разработка программного обеспечения (ПО) для ранжирования факторов (причин) ДТП по значимости, которое может использоваться как рекомендации для принятия мер к устранению наиболее значимых факторов.

### Известные подходы в области факторного анализа ДТП

Среди существующих исследований касательно не только СПб практически отсутствует количественный анализ аварийных факторов. В основном они носят описательный характер и приводят различную собирательную статистику.

Одно из них – исследование компании «Яндекс» от 2017 г. [2], в котором были выявлены наиболее аварийные и наиболее вероятные относительно ДТП участки дорог по Санкт-Петербургу.

Второе исследование – статья, описывающая методику анализа ДТП с пострадавшими на примере Санкт-Петербурга [3]. В рамках рассматриваемой работы на основе результатов корреляционного анализа были составлены модели линейной регрессии зависимости общего количества ДТП от факторов для некоторых выборок перекрестков.

### Факторный анализ и метод анализа иерархий

Факторный анализ – это метод комплексного и систематического исследования влияния того или иного фактора на величину результирующего показателя.

Данный вид анализа предоставляет возможность аналитику кратко и в то же время разносторонне описать исследуемый объект. Во время проведения анализа выясняется внутренняя взаимосвязь между исследуемыми величинами. Возможна ситуация, когда по результатам исследования выяснилось, что несколько показателей имеют близкие значения оценок, что позволяет сделать вывод об их зависимости от какого-либо общего признака. Этот признак называется фактором.

Логика этапов проведения факторного анализа может быть следующая.

1. Постановка задачи проведения анализа.
2. Выбор влияющих факторов на конкретный вид ДТП.
3. Типизация и систематизация факторов.
4. Определение модели зависимости (в том числе регрессии) результирующего показателя от факторов.
5. Построение множественной (линейной/нелинейной) регрессии.
6. Оценка и ранжирование степеней влияния факторов на результирующий показатель.
7. Верификация полученной факторной модели на практике и ее уточнение.

Остановимся на одном из основных этапов факторного анализа – ранжирование факторов с точки зрения их степени влияния на отдельный вид ДТП. Одним из подходов к решению данной проблемы является метод анализа иерархий (МАИ), разработанный Томасом Саати в 80-х годах прошлого века [4].

Данный метод относится к классу критериальных методов. Одно из главных его достоинств – то, что он не предоставляет единственно «правильное» решение, а приводит к варианту, который наиболее четко согласуется с пониманием цели задачи и ее условиями.

Применение МАИ связано с задачами выбора, для которых формулируются:

- цель исследования;
- альтернативы (варианты достижения цели);
- критерии для оценивания альтернатив.

Эти элементы представляются в виде иерархической структуры, которая изображается в виде дерева от цели (корень) через критерии к альтернативам (ветви). В иерархической структуре может быть 3 и более уровней, в зависимости от решаемой задачи.

Реализация метода анализа иерархий осуществляется поэтапным построением матриц парных сравнений критериев по цели и альтернатив по критериям с последующей скаляризацией векторных оценок весовых коэффициентов альтернатив пропорционально вкладу важности критерия относительно анализируемой цели.

Особенностью данного исследования является применение корректной модификации классического МАИ [5, 6], свободной от основного его недостатка, а именно – нарушение достигнутых предпочтений при добавлении дополнительных альтернатив в множество сравниваемых [7].

### **Разработка программного обеспечения ранжирование факторов с точки зрения их степени влияния на отдельный вид ДТП**

При разработке ПО ставились две следующие задачи.

Задача А. Ранжировать причины ~ факторы для данного типа аварии на определенном участке дороги;

Задача Б. Ранжировать типы аварий по убыванию влияния отдельно взятой причины.

Согласно показателям состояния безопасности дорожного движения [8], ДТП разделяются на определенные типы (например, столкновение, наезд на препятствие, падение пассажира, съезд с дороги, наезд на стоящее ТС и др.).

Участок дороги определен до начала алгоритма ранжирования, поэтому данные, над которыми проводится анализ, предоставляются уже с учетом этого.

Критериями задачи А являются:

- тип ДТП (общее число аварий данного типа);
- число аварий из рассматриваемых (данного типа), для которых характерно нарушение режима движения;
- число физически пострадавших;
- число погибших.

Альтернативами выступают факторы (причины) аварии.

Критериями в задаче Б были выбраны 4 показателя:

- фактор ДТП (общее число аварий с присутствием данного фактора);

- число аварий из рассматриваемых (с присутствием данного фактора), для которых характерно нарушение режима движения;
- число физически пострадавших;
- число погибших.

Альтернативами в задаче Б выступают типы аварии.

Исходными данными является документ открытых информационных данных по дорожно-транспортным происшествиям по городу Санкт-Петербургу за 2020 год в формате xls, который доступен на сайте ГИБДД [8]. Из всего набора данных, предоставляемых в данном документе, используются только следующие данные:

- местоположение аварии (район, улица, объекты УДС (перекрестки и т. д.);
- факторы аварии, относящиеся к выделенным ранее классам причин аварий;
- тип ДТП;
- последствия ДТП (нарушение движения, количество погибших и пострадавших).

Была проведена практическая апробация программного продукта для столкновений в Кировском районе Санкт-Петербурга с использованием описанных выше исходных данных за 2020 год.

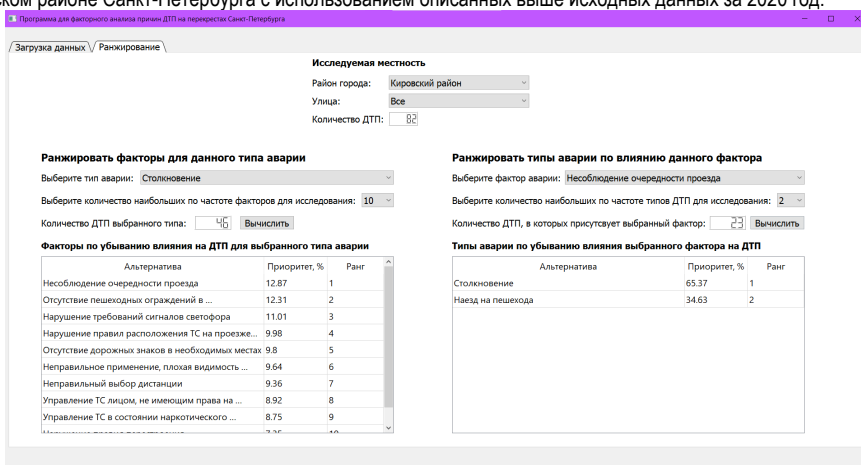


Рис. 1. Практическая апробация программного продукта

В результате проведенного тестирования можно сделать вывод, что наиболее значимым фактором ДТП при столкновениях в Кировском районе за 2020 год является «Несоблюдение очередности проезда», что является нарушением ПДД и относится к человеческим факторам ДТП. При этом чуть менее важным считается фактор «Отсутствие пешеходных ограждений в необходимых местах», что является нарушением дорожных условий и может быть исправлено уполномоченными органами.

Также для наиболее значимого фактора ДТП при столкновениях в Кировском районе за 2020 год было произведено исследование его влияния на разные типы ДТП. В результате данного исследования выяснилось, что, кроме столкновений, рассматриваемый фактор также влияет на такой тип ДТП, как наезд на пешехода. При этом степень влияния на столкновения почти в 2 раза больше, чем степень влияния на наезд на пешехода.

### Заключение

Разработанное ПО реализует метод и соответствующий алгоритм ранжирования альтернатив, на основе которого стало возможным частичное решение проблемы оценивания относительной важности степени влияния различных факторов на возникновение ДТП.

Говоря о получаемых результатах в ходе работы ПО, хотелось бы отметить, что метод анализа иерархий, применяемый при ранжировании в данной программе, имеет математическое обоснование в виде положений линейной алгебры. Это позволяет вынести утверждение, что результаты ранжирования по множеству критериев не могут быть улучшены с применением других методов оценки

#### **Библиографический список**

1. Агентство «Автостат». URL: <https://www.autostat.ru> (дата обращения: 20.05.2023).
2. Исследования Яндекса – Автомобильные аварии в Санкт-Петербурге. URL: <https://yandex.ru/company/researches/2017/spb/accidents> (дата обращения: 20.05.2023).
3. Атаев П. Г., Геллер Р. М., Липаткин Д. В. Методика анализа дорожно-транспортных происшествий с пострадавшими на примере Санкт-Петербурга // Транспорт Российской Федерации. 2019. № 5 (84). С. 50–53.
4. Саати Т. Л. Принятие решений. Метод анализа иерархий. М.: Радио и связь, 1993. 278 с.
5. Самохвалов Ю. Я. Групповой учет относительного превосходства альтернатив в задачах принятия решений // Кибернетика и системный анализ. 2003. № 6. С. 141–145.
6. Колесникова С. И. Особенности применения линейной свертки критериев в методе парных сравнений // Информационные технологии. 2011. № 1. С. 24–29.
7. Колесникова С. И. Модификация метода анализа иерархий для динамических набор альтернатив // Прикладная дискретная математика. 2009. № 4. С. 102–109.
8. Показатели состояния безопасности дорожного движения. URL: <http://stat.gibdd.ru/> (дата обращения: 20.05.2023).



УДК 004.492.3

**Д. Д. Савельева**

студент кафедры компьютерных технологий и программной инженерии

**Т. М. Татарникова** – доктор технических наук, профессор – научный руководитель

## ОБЗОР КИБЕРУГРОЗ ЗА 2022 ГОД

### Введение

За 2022 год произошли геополитические события, которые повлияли на ситуацию в сфере информационной информации, в том числе и интернета вещей. Стало увеличиваться количество инцидентов, связанных с киберпреступностью.

### Цели и средства киберугроз

В 2022 году каждая десятая успешная атака пришлась на промышленное предприятие. Чаще всего были атакованы государственные учреждения, реже медицинские учреждения, научные и образовательные организации и промышленность.

Среди последствий атак можно отметить утечку конфиденциальной информации, нарушение основной деятельности, прямые финансовые потери и даже ущерб интересам государства (например, после атаки группировки Conti на государственные учреждения в стране было объявлено чрезвычайное положение).

Из успешных атак большая часть была направлена на компьютеры, серверы и сетевое оборудование — основные мишени преступников. Также злоумышленники проводили атаки на персонал промышленных организаций с помощью вредоносных рассылок по электронной почте и фишинговых сайтов.

В большинстве случаев успешных атак в качестве инструментов используются вредоносное ПО и социальная инженерия. Из вредоносного ПО лидируют программы-шифровальщики, которые шифруют данные пользователя и требуют денежный перевод за расшифровку. В 2022 году популярной также стала программа, удаляющая данные с жесткого диска. Оба вида вредоносного ПО представляют собой опасность для любых предприятий и требуют к себе пристального внимания.

Также в некоторых случаях было использовано шпионское ПО и ПО для удаленного управления. В первом случае возможна потеря конфиденциальных данных, нарушение основной деятельности или установка дополнительных нежелательных программ. Во втором случае злоумышленники могут получить такую информацию как нажатие клавиш, снимки экрана и даже сетевой трафик.

Чаще всего такое вредоносное ПО распространяется через компрометацию компьютеров, серверов или сетевого оборудования и через электронную почту. Последний путь стал очень популярным благодаря социальной инженерии. Злоумышленники маскируют вредоносное ПО под различные документы, рассылают письма с вредоносными ссылками или создают фишинговые сайты.

В даркнете (скрытом сегменте Интернета, связанный с продажей запрещенных товаров и услуг) также продаются доступы к промышленным компаниям. Причем в 2022 году такая продажа составляла 75% от всех объявлений, касающихся промышленности. Также популярными объявлениями считаются продажа данных и организация ddos-атак.

### Кибергруппировки

Профессионалами в атаках на промышленные предприятия являются различные преступные группировки, которые можно разделить на три класса в зависимости от целей, которые они преследуют.

– Шифровальщики. Такие злоумышленники используют вредоносное ПО для шифрования и эксквизитации данных с целью вымогательства. Наиболее известны такие группировки как LockBit, BlackCat, Cl0p и Conti.

– Хактивисты – злоумышленники, чьи кибератаки направлены на привлечение внимания общественности к различным социальным и политическим проблемам, а не на получение прибыли. Для до-

стижения своих целей киберактивисты проводили массированные DDoS-атаки, взламывали сайты промышленных компаний, похищали деловую переписку компаний с почтовых серверов. Самые известные группировки в этом сегменте: Anonymous, Guacamaya, NB65 и Gonjeshke Darande.

– АPT-группировки – организованные группы злоумышленников с высоким уровнем квалификации и технической оснащённостью. Их основная цель – шпионаж. Среди таких группировок выделяются такие как Space Pirates, APT31 и ChamelGang. Наиболее активная из них – Space Pirates, использующая фишинг и внешние уязвимости продуктов атакуемых компаний.

В 2022 году активизировались политически настроенные хакеры, которые чаще всего атакуют государственные учреждения и СМИ с целью проведения DDoS-атаки и кражи данных.

### Рост числа DDoS-атак

За 2022 год число DDoS-атак только на российский организации увеличилось на 700% по сравнению с 2021 годом. Почти половина этих атак приходится на банки и платежные системы. Всего было зарегистрировано около 1,2 миллионов инцидентов.

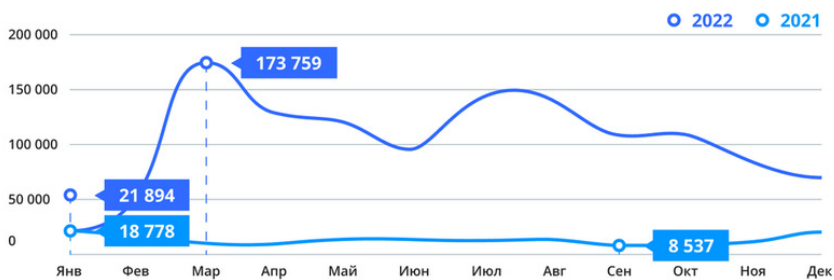


Рис. 1. Распределение DDoS-атак по месяцам

Длительность атак сократилась, но частота возросла в 4 раза. Такие изменения произошли потому, что использовать мощные цепочки устройств слишком дорого. Поэтому злоумышленники объединяются в сообщества для участия в атаках. Таким образом можно атаки становятся массовыми, но непродолжительными [1, 2].

### Киберугрозы для интернета вещей

С развитием интернета вещей взгляды злоумышленников все больше обращаются к этой области как к инструменту достижения своих целей, так как сферы, в которых используется интернет вещей, обширны и делятся на две группы: Clot – Consumer Internet of Things и IIoT – Industrial Internet of Things. Первая группа ориентирована на конечного пользователя и больше связана с умными бытовыми приборами и носимыми гаджетами. Вторая группа ориентирована на более крупные проекты для экономики, такие как умные города, электросети, автоматизация производства и многое другое [3 – 5]. Как и для любого другого устройства подверженного атакам, одними из самых популярных киберугроз для устройств интернета вещей являются – это DDoS-атаки и кража личных данных. В первом случае, атаки становятся возможными благодаря цепочке подключенных к Интернету устройств и заражении каждого из них вредоносной программой, которая позволяет удерживать полный контроль над устройством без согласия владельца. Размер таких цепочек может составлять от десятков до сотен тысяч устройств. Во втором случае, речь идет о данных, получаемых с различных датчиков и гаджетов пользователя, которые в случае кражи могут дать похитителю информацию личного характера. Например, датчик движения в доме может быть использован для получения информации о том, когда в доме есть активность, а когда дом пустует. Другими словами, во втором случае угроза состоит в потенциальной опасности попадания информации к злоумышленникам [6].

Большинство гаджетов не способны противостоять DDoS-атакам и краже данных по ряду причин:

- Круглосуточная работа устройств позволяет даже в случаях частичного внешнего контроля обойти систему защиты.

- Полный контроль со стороны пользователя отсутствует.

- Сами владельцы редко замечают, что их устройства подвержены атаке или краже данных.

Таким образом, все причины связаны с тем, что внешний контроль пользователя отсутствует или очень слаб.

### Прогнозы

Исходя из приведенных данных можно сделать вывод, что в будущем проблема информационной безопасности станет только более актуальной. В частности, из-за ухода многих иностранных инструментов защиты с российского рынка потребуются разработка качественного отечественного ПО, что означает привлечение новых кадров и ресурсов в отрасль информационной безопасности. Также ожидается увеличение количества инцидентов, связанных с кибершпионажем и кражей данных, появление новых группировок и вредоносных программ.

### Библиографический список

1. Купреев О., Гутников А., Шмелев Я. DDoS-атаки в третьем квартале 2022 года. URL: <https://securelist.ru/ddos-report-q3-2022/106012/> (дата обращения: 15.04.2023).

2. Волошин Е. Фишинг, DDoS, дефейс: с какими еще киберугрозами столкнулись российские компании в 2022 году. URL: <https://sber.pro/digital/publication/fishing-d-do-s-defeys-s-kakimi-eshhe-kiberugrozami-stolknulis-rossijskie-kompanii-v-2022-godu> (дата обращения: 15.04.2023).

3. Сахаров Д. В., Козлов Д. С. Обнаружение аномального поведения устройства IoT в сети на основе модели трафика. 2019. Т. 7. № 3. С. 50–55.

4. Ромасевич Е. П. Исследование агрегированного трафика беспроводных IoT устройств // Современные информационные технологии и ИТ-образование. 2017. Т. 3. № 3. С. 122–128.

5. Татарникова Т. М., Богданов П. Ю., Краева Е. В. Предложения по обеспечению безопасности системы умного дома, основанные на оценке потребляемых ресурсов // Проблемы информационной безопасности. Компьютерные системы. 2020. № 4. С. 88–94.

6. Татарникова Т. М., Бимбетов Ф., Богданов П. Ю. Выявление аномалий сетевого трафика методом глубокого обучения // Известия СПбГЭТУ ЛЭТИ. 2021. № 4. С. 36–41.

7. Золотова Т. Positive Technologies: больше всего в 2022 году в России хакеров интересовали государственный, медицинский и промышленный секторы. URL: <https://servernews.ru/1080271> (дата обращения: 15.01.2023).

8. Новиков А. Кто и как атакует российские организации. URL: <https://www.ptsecurity.com/ru-ru/research/analytics/ogo-kakaya-ib/#id6> (дата обращения: 15.04.2023).

УДК 681.3

**А. Н. Секунов**

студент кафедры компьютерных технологий и программной инженерии

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

**ПРИМЕНЕНИЕ АЛГОРИТМА ОТРИЦАТЕЛЬНОГО ОТБОРА  
ДЛЯ АНАЛИЗА ПОВЕДЕНИЯ СЕТЕВОГО ТРАФИКА НА ОСНОВЕ К-D ДЕРЕВА**

**Введение**

На сегодняшний день многие сферы деятельности связаны с сетевыми технологиями, например: транспорт, связь, здравоохранение, финансы, военный комплекс, различного рода промышленность и многие другие. Все они работают с информацией, доступ к которой можно получить из сети Интернет, и этим могут воспользоваться злоумышленники: в частности, помимо кражи и изменения конфиденциальной информации, возможно нарушение работы систем, что может привести к более страшным последствиям.

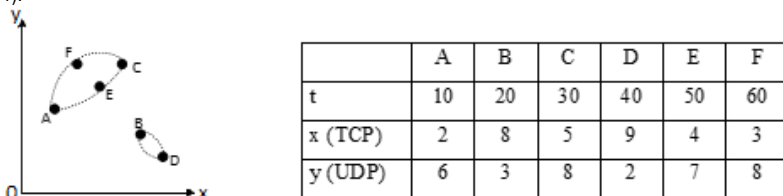
Поэтому всегда актуальны программные средства, нацеленные на обнаружение и предотвращение таких угроз. Например, существуют системы по обнаружению сетевых вторжений (NIDS, Network Intrusion Detection System), и одним из возможных вариантов их реализации могут служить иммунные системы, так как в них заложена одна очень важная особенность: предотвращение даже тех угроз, с которыми они раньше не имели дела.

Алгоритмы на основе иммунитета активно исследуются на протяжении последних трех десятилетий. Проводились отдельные конференции, посвященные иммунным системам (ICARIS, International Conference on Artificial Immune Systems), на котором предлагались различные способы их применения в сферах компьютерной безопасности, машинного обучения, обнаружения неисправностей и других. Для данной статьи особенно интересны результаты Д. Дасгупты и Ф. А. Гонсалеса в 2003 г. [1]. Они предложили вариант алгоритма отрицательного отбора (АОО), работающего на основе вещественного кодирования.

Цель данной статьи заключается в рассмотрении данного метода и предложении своего варианта модификации на основе структуры данных k-d дерева для анализа поведения пакетов сетевого трафика.

**Способ кодирования поведения сетевого трафика**

Пусть,  $x$  – количество пришедших пакетов TCP, а  $y$  – количество пакетов UDP. Каждый вектор фиксирует значения в пределах 10 секунд. Тогда можно расположить эти точки в 2-d пространстве (рис. 1).



*Рис. 1. Размещение статистики по двум параметрам в пространстве*

Таким образом можно представить множество независимых друг от друга признаков как одну точку в многомерной пространстве и анализировать все характеристики в совокупности.

В данной работе тесты проводились над следующим вектором статистики:

- 1) общее количество пакетов TCP;
- 2) общее количество пакетов UDP;

- 3) общее количество пакетов ICMP;
- 4) общее количество пакетов других протоколов;
- 5) количество полукоткрытых соединений TCP (пакеты с флагом SYN);
- 6) количество открытых соединений TCP (пакеты с флагом ASK, отправленные как ответ на SYN+ASK получателя);
- 7) количество закрытых соединений TCP (пакеты с флагом FIN);
- 8) количество сброшенных соединений TCP (пакеты с флагом RST);
- 9) количество обращений к разрешенным портам TCP;
- 10) количество обращений к неразрешенным портам TCP;
- 11) количество обращений к разрешенным портам UDP;
- 12) количество обращений к неразрешенным портам UDP.

Данные признаки были взяты из принципа, что их легче всего было отследить, в другой работе [2] для анализа использовали вектор, состоящий из 33 признаков. Все представленные параметры имеют целочисленное значение от 0 до 65535. В качестве разрешенных портов для TCP считались значения: 20, 21, 80, 445, 1234, 1236, а для UDP: 53, 1235, 1237.

### Фитнесс-функция на основе вычисления площади k-мерного гипер-прямоугольника

После расположения статистик в многомерном пространстве, стоит задача генерации детекторов, которые определяют границы аномальной области. Дасгупта и Гонсалес, в своей работе, использовали прямоугольные области (рис. 2), как правила обнаружения аномалий [1].

Тогда правило «Rule 1» будет звучать следующим образом:

ЕСЛИ S1 находится в пределах [0, 1]

И S2 находится в пределах [0.8, 1],

ТО ненормально.

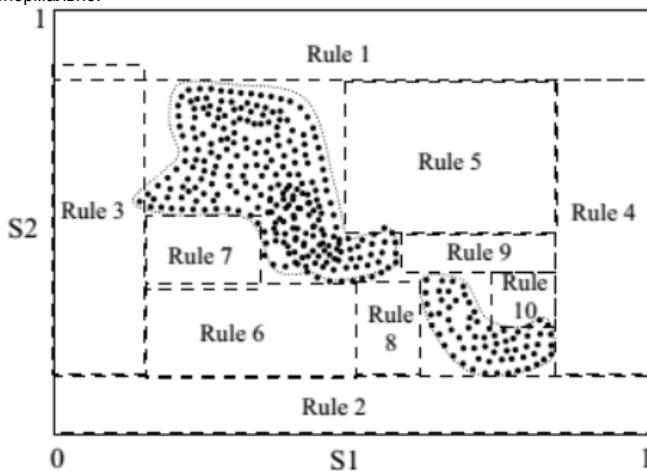


Рис. 2. Пример обнаружения вторжений на основе АОО

Как видно из рис. 2, данные правила не всегда могут охватить нужные аномальные области. Помимо этого, присутствуют случаи захвата области нормального поведения или же осуществляется покрытие уже контролируемой области.

Поэтому Дасгупта и Гонсалес привели процесс генерации детекторов к проблеме оптимизации [1]. Было поставлено несколько целей:

- наличие меньшего набора правил;
- малое количество пересечений областей;

– уменьшение числа покрытий нормальных образцов.

Для решения первой цели используется функция вычисления площади прямоугольника  $k$ -мерного пространства (в данном примере  $k = 2$ ):

$$volume(R) = \prod_{i=1}^k (high_i - low_i), \quad (1)$$

где  $R$  – хромосома правила, например для «Rule 1» это значение имеет вид (0; 1; 0.8; 1),  $high$  и  $low$  задают длину ребра гипер-прямоугольника в  $i$ -м измерении. Чем больше площадь, тем меньше правил требуется для обхвата аномальных областей.

Для третьей цели приводится следующее выражение:

$$error\_number(R) = \{x_i \in S \mid x_i \in R\}, \quad (2)$$

где  $x_i$  –  $i$ -я нормальная выборка,  $S$  – нормальная область,  $R$  – правило. Данная функция подсчитывает количество нормальных образцов, которое охватывает правило. При этом выполняется принцип АОО, т. е. правила, которые охватывают больше нормальных данных, будут отбрасываться.

Тогда фитнес-функция будет иметь вид:

$$raw\_fitness(R) = volume(R) - C \cdot error\_number(R), \quad (3)$$

где значение  $C$  определяется пользователем и задает оценку штрафных эффектов покрытия нормальных образцов. Чем больше значение функции  $raw\_fitness$ , тем лучше будут результаты.

Чтобы избежать повторного нахождения одного и того же правила, требуется изменять целевую функцию после каждого запуска алгоритма:

$$fitness(R) = raw\_fitness(R) - \sum_{R_j \in rule\_set} volume(R \cap R_j) \quad (4)$$

где  $R_j$  – ранее найденное правило,  $R$  – текущее правило.

Однако применение данной формулы приводит к большому объему вычислений, особенно когда пространство  $k$  достаточно большое (как было сказано ранее, в статье рассматривалось только 12 признаков, но существует пример, где применялось 33 признака). Поэтому в качестве модификации было предложено использовать структуру данных  $k$ -d дерева.

### Модификация с использованием $k$ -d дерева

$k$ -d дерево используется для уменьшения диапазона поиска в  $k$ -мерное пространство, выполняя его деление  $k$ -мерными плоскостями. Структура данных находит применение в задачах поиска ближайшего соседа, проектирование точек, трассировки лучей и т. д.

Для построения дерева вычисляется внешний  $k$ -мерный прямоугольник, который охватывает все нужные точки (далее, для упрощения изложения, будет использовать термин «куб», хотя под ним имеется ввиду любой многомерный прямоугольник), затем производится его деление пополам.

При этом, обычно, рассматривают четыре варианта разделения [3]:

1. По наибольшей стороне «куба».
2. Путем чередования, сначала ось  $OX$  пополам, затем  $OY$  и т. д.
3. По медиане, для этого точки сортируются по одной из сторон, а затем выбирается средний элемент списка.
4. Используя специальную функцию оценки.

Из перечисленного был выбран 2 способ, как самый простой и быстрый, так как в работе не стоит задача поиска ближайших друг к другу точек, важно определить факт попадания аномальной точки в пустое пространство. Так, например, на рис. 3 показана логика построения дерева для 3-мерного пространства. В квадратных скобках указаны диапазоны значений. Рассматриваются только целочисленные координаты, при расчете средних значений происходит округление в меньшую сторону. Параметр в узле дерева входит в множество левого поддерева.

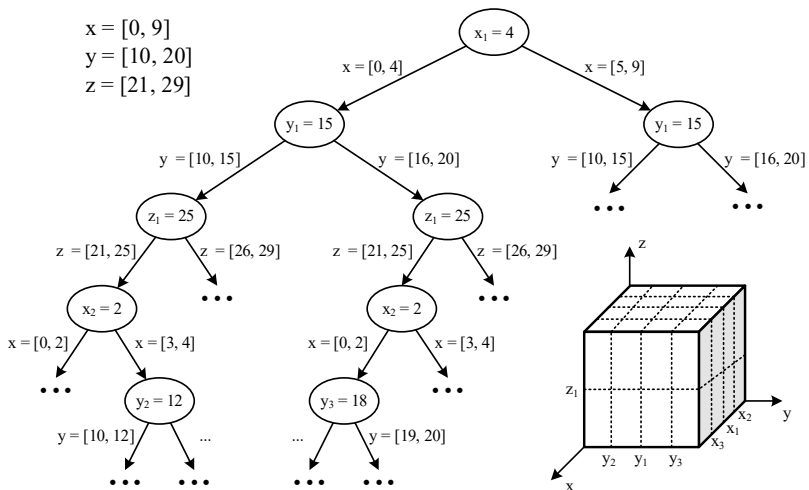


Рис. 3. Логика построения k-d дерева

Следующим шагом является заполнение дерева векторами статистик, чтобы определить какие области пустые (аномальные), а какие содержат шаблоны нормального поведения. Чтобы не хранить большое количество точек в одном листе, предлагается хранить информацию в виде двух векторов минимума и максимума значений в данном листе, аналогично главному «кубу». Первый вектор задает начальное положение единичного пространства, а последующие корректируют его форму (рис. 4).

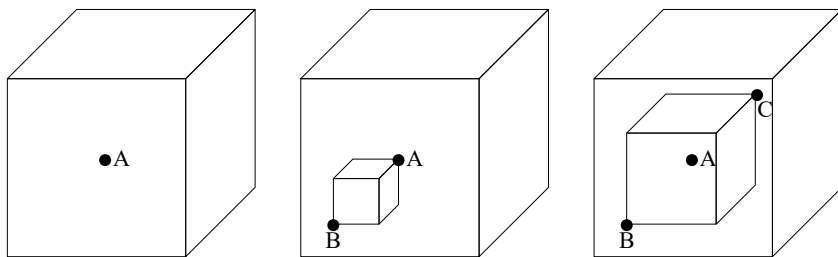


Рис. 4. Процесс определения внутреннего пространства листа

Если в процессе обучения попадает вектор, которые выходит за границы основного «куба» уже построенного дерева, то строится новое дерево. В него также добавляются, по отдельности, вектора максимума и минимума каждого листа старого дерева, чтобы восстановить ранее полученные данные.

С применением такого метода, в роли детекторов выступают:

- пространство вне основного «куба»;
- пустые листья дерева;
- расстояние между содержимым листа и краями листового «куба».

В итоге если новый вектор поведения сети, в режиме мониторинга, стремится в пустую область, то генерируется событие наличия аномалии.

### Сравнительный анализ результатов работы двух алгоритмов

Для анализа было решено взять дампы прослушивания сетевого трафика из набора данных DARPA 1999 [4]. В нем имеются файлы «\*.tcpdump» для внешней и внутренней сети, сгруппированных по 5 неделям, в каждом по 5 дней (с понедельника по пятницу) с информацией о прослушивании сетевого трафика в течении 22 часов. Первые 3 недели используются для обучения, 1 и 3 неделя с безопасными данными, а 2 с сетевыми атаками.

Конкретно в этой работе, были использованы данные только внешнего сетевого трафика за 2 недели. При этом, с учетом того, что используется упрощенный вектор статистики, то для ускорения, обрабатывались пакеты с установленной версией заголовка IPv4. При переборе данных, ввелась статистика по ранее указанному вектору с интервалом в одну секунду.

Так при рассмотрении набора данных, состоящих из примерно 13 млн пакетов за 2 недели было сформировано около 400 тыс. статистик ( $k=12$ ,  $t=1$  с.), в среднем первый метод обрабатывал данные за 60 мс используя 5 Кбайт, а второй – чуть менее 1 мс используя 200 Кбайт оперативной памяти.

### Заключение

В данной работе рассмотрены две реализации алгоритма отрицательного отбора для обнаружения сетевых атак: на основе вычисления площади  $k$ -мерного гипер-прямоугольника и структуры данных  $k$ - $d$  дерева. Измерены параметры: времени выполнения и размера, требуемого для хранения диапазонов (детекторов). Как и предполагалось, что структура  $k$ - $d$  дерева позволяет быстрее обрабатывать данные, но из-за этого требует большого количества памяти. С учетом того, что объемы памяти становится все доступнее, а поток запросов многократно расчет (как и потребность в скорости их обработки), то данная модификация с  $k$ - $d$  деревом может быть актуальна в реализации систем защиты от сетевых вторжений.

### Библиографический список

1. Скобцов Ю. А. Введение в искусственные иммунные системы: учеб. пособие. СПб: ГУАП, 2022. 212 с.
2. Бурлаков М. Е., Ивкин А. Н. Система обнаружения вторжения на основе искусственной иммунной системы / Самарский национальный исследовательский университет им. академика С. П. Королева, 2019. 17 с.
3. Камынин И. Анатомия KD-Деревьев. URL: <https://habr.com/ru/post/312882/> (дата обращения: 30.05.2020).
4. 1999 DARPA Intrusion Detection Evaluation Data Set. URL: <https://archive.ll.mit.edu/ideval/data/1999data.html> (дата обращения: 30.11.2020).



УДК 004.623

**Я. О. Сениченкова**

студент кафедры компьютерных технологий и программной инженерии

**М. Д. Поляк** – старший преподаватель – научный руководитель

## СОЗДАНИЕ КОНВЕЙЕРА ОБРАБОТКИ ДАННЫХ В APACHE NIFI И AIRFLOW ДЛЯ РЕШЕНИЯ ETL-ЗАДАЧ

### Введение

ETL (Extract-Transform-Load) – это процесс извлечения, преобразования и загрузки данных из разных источников в систему хранения данных, такую как база данных или хранилище данных. ETL-задачи являются неотъемлемой частью большинства бизнес-процессов, которые требуют обработки и использования данных, например, в области банковского дела или ритейла. Важно отметить, что ETL-задачи связаны не только с самим процессом обработки данных, но и с архитектурой системы хранения данных, анализом результатов и предоставлением пользовательского интерфейса для доступа к данным. В связи с этим, рекомендуется использовать такие инструменты, как Apache NiFi, Talend ETL и Microsoft SQL Server Integration Services, чтобы легко настраивать ETL-задачи и упростить процесс работы с данными [1].

### Постановка задачи

Рассмотрим задачу извлечения и дальнейшей обработки метеоданных. В качестве источника данных о погоде будем рассматривать сайт [r5.ru](http://r5.ru), содержащий как исторические данные многолетних наблюдений, так и текущие актуальные показания датчиков метеостанций, а также краткосрочные прогнозы. Порядок действий при построении ETL-пайплайна может быть следующим. Сначала необходимо извлечь данные с сайта [r5.ru](http://r5.ru) (распарсить сайт), затем данные необходимо трансформировать в формат CSV, чтобы упростить их дальнейшую обработку. После этого записи можно загрузить в шину данных Apache Kafka [2]. В завершение, данные могут быть перемещены в Apache Hadoop [3], где они будут храниться в распределенной файловой системе и могут быть использованы для более сложных аналитических задач. Схема обработки данных представлена на рис. 1.



Рис. 1. Схема обработки данных

На успешную реализацию ETL-процессов существенно влияет выбор инструментов для их настройки. На текущий момент популярными являются такие инструменты, как Apache NiFi [4], Apache Airflow [5].

### Извлечение данных с помощью Apache NiFi

Apache NiFi – это платформа обработки и передачи данных, которая позволяет перемещать данные между любыми источниками и местами назначения. Главная задача NiFi – это перенос данных, и это делается быстро, надежно и с большими объемами. Согласно работе [6], NiFi имеет следующие преимущества:

1. Наличие удобного веб-интерфейса и большого числа интегрированных инструментов NiFi значительно упрощает работу с данными и повышает уровень продуктивности пользователя. Также это

позволяет производить настройку процесса в реальном времени благодаря просмотру сообщений об ошибках, статистики и состояния потока данных.

2. Система гарантирует обработку всех записей вне зависимости от объема потока данных. Это является основным принципом NiFi.

В данной работе NiFi-процесс используется для извлечения данных с сайта gr5.ru и перемещения их в Apache Kafka.

Схема NiFi-процесса представлена на рис. 2.



Рис. 2. Схема NiFi-процесса извлечения данных с сайта gr5.ru

В качестве примера на рис. 3 приведена реализация блока «Обработка данных» из рис. 2 в NiFi.

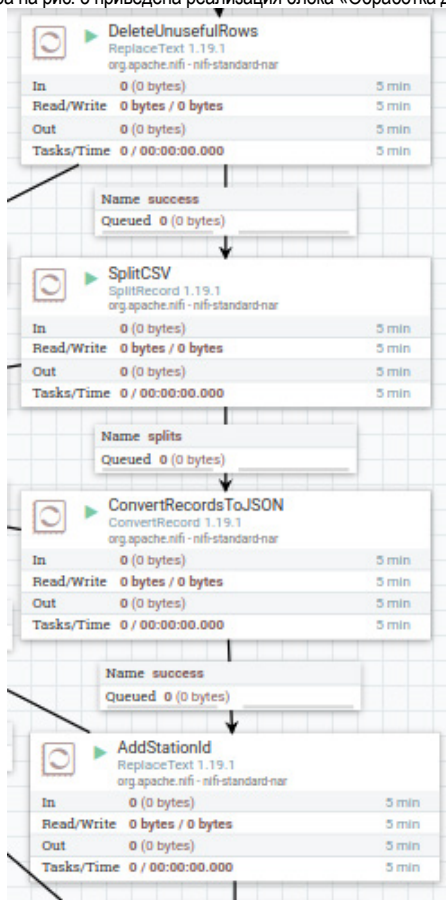


Рис. 3. Реализация блока «Обработка данных» в NiFi

Фрагмент машинного описания ETL-процесса, визуальное представление которого приведено на рис. 2, приведено в листинге 1.

Листинг 1. Фрагмент машинного описания ETL-процесса

```
<processGroups>
<id>d770e0b2-ae17-3bed-0000-000000000000</id>
<parentGroupld>b25c2b8b-dfd1-3178-0000-000000000000</parentGroupld>
<position>
  <x>0.0</x>
  <y>0.0</y>
</position>
<versionedComponentId>d770e0b2-ae17-3bed-baeb-6851930da767</versionedComponentId>
<comments></comments>
<contents>
  <connections>
    <id>0b30b098-34d0-3971-0000-000000000000</id>
    <parentGroupld>d770e0b2-ae17-3bed-0000-000000000000</parentGroupld>
    <versionedComponentId>80c4e464-c8aa-386e-8ca0-5fc09837284</versionedComponentId>
    <backPressureDataSizeThreshold>1 GB</backPressureDataSizeThreshold>
    <backPressureObjectThreshold>10000</backPressureObjectThreshold>
    <bends>
      <x>926.2738037109375</x>
      <y>1222.20849609375</y>
    </bends>
  </connections>
</contents>
```

### Перемещение данных в Hadoop с помощью Apache Spark

После перемещения данных в Kafka, необходимо выгрузить данные в Apache Hadoop. Для выполнения данной задачи необходимо написать DAG (Directed Acyclic Graph, направленный ациклический граф) в Apache Airflow. Этот инструмент был выбран для данного этапа благодаря преимуществам, выделенным в работе [7]:

1. Пайплайны пишутся на языке Python, что обеспечивает гибкость данному процессу.
2. Планирование работы DAG'ов. Планировщик в Airflow позволяет задавать интервал, по прошествии которого пайплайн будет выполнен. Таким образом, можно обеспечить автоматический запуск DAG'a раз в час, раз в сутки и тд.
3. Веб-интерфейс, позволяющий легко мониторить результаты выполнения пайплайна, что упрощает отладку DAG'a.

Пример создания DAG'a в airflow представлен в листинге 2 [9]

Листинг 2. Пример создания DAG'a в airflow

```
import datetime

from airflow import DAG
from airflow.operators.empty import EmptyOperator

with DAG(
    dag_id="my_dag_name",
    start_date=datetime.datetime(2021, 1, 1),
    schedule="@daily",
):
    EmptyOperator(task_id="task")
```

Схема DAG'a в Airflow для задачи перемещения данных из Kafka представлена на рис. 3. Данный вычислительный граф состоит из единственной вершины и не имеет ребер в силу простоты по-

ставленной задачи. В общем случае DAG позволяет разбить выполняемую задачу на подзадачи (вершины графа), задать порядок их выполнения и зависимости между ними (ребра графа).



Рис. 4. Схема простейшего вычислительного графа в Airflow, отвечающего за перемещение данных из Kafka в Hadoop

Внутри данного пайплайна находится Spark job, осуществляющая перенос данных. Spark job – это задание, выполняемое на кластере Apache Spark [8], которое включает в себя серию этапов, необходимых для обработки данных. Оно может содержать множество этапов вычислений, включая чтение и запись данных, агрегацию, фильтрацию и многие другие операции, которые могут выполняться одновременно. Каждый этап выполняется на отдельном узле кластера и в результате комбинируется для производительности высшего уровня.

Схема Spark job для перемещения данных из Kafka в Hadoop представлена на рис. 4.

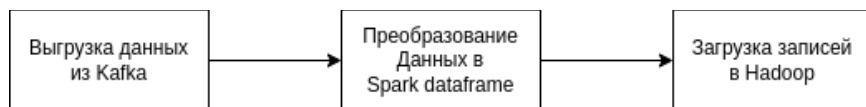


Рис. 5. Схема spark job, перемещающей данные из Kafka в Hadoop

### Заключение

В заключении можно отметить, что использование технологий NiFi, Spark и Airflow позволяет реализовать высокопроизводительный и гибкий ETL-процесс для обработки больших объемов данных в режиме реального времени. NiFi позволяет управлять перемещением данных, Spark обеспечивает быстрое выполнение вычислений, а Airflow позволяет оркестрировать все эти процессы, оберывая Spark job в DAG. Таким образом, такие технологии могут стать новым поколением ETL-процессов в IT-индустрии, обеспечивая высокую скорость и эффективность обработки данных.

### Библиографический список

1. Kimball R. The Data Warehouse ETL Toolkit // John Wiley & Sons. NY. 2004. 528 с.
2. Apache Kafka. URL: <https://kafka.apache.org/> (дата обращения: 15.04.2023).
3. Apache Hadoop. URL: <https://hadoop.apache.org/> (дата обращения: 15.04.2023).
4. Apache NiFi. URL: <https://nifi.apache.org/> (дата обращения: 15.04.2023).
5. Apache Airflow. URL: <https://airflow.apache.org/> (дата обращения: 15.04.2023).
6. Chatti S. Using spark, kafka and NIFI for future generation of ETL in IT industry // J Innov Inf Technol. 2019. № 3 (2). С. 11–14.
7. Harenslak B. P., de Ruiter J. Data Pipelines with Apache Airflow. NY. Simon and Schuster, 2021. 480 с.
8. Apache Spark. URL: <https://spark.apache.org/> (дата обращения: 15.04.2023).
9. Declaring a DAG. URL: <https://airflow.apache.org/docs/apache-airflow/stable/core-concepts/dags.html#declaring-a-dag> (дата обращения: 15.04.2023).

УДК 519.688

**Д. А. Фомин**

студент кафедры компьютерных технологий и программной инженерии

**С. И. Колесникова** – доктор технических наук, профессор – научный руководитель**АЛГОРИТМ АНАЛИЗА МУЗЫКАЛЬНОГО ВРЕМЕННОГО РЯДА ДЛЯ ФОРМИРОВАНИЯ ПРОГНОЗА****Введение**

Автоматизация генерации музыкальных композиций может быть сведена к предсказанию последовательностей нот, звуков (волн). Несмотря на существование четкой формализации основ составления музыкальных композиций, правила эти нередко нарушаются самими композиторами, поэтому воссоздать процесс создания мелодии человеком («поверить алгеброй гармонию») крайне проблематично.

Если представить музыкальную композицию как временной ряд  $Y(t) = \{y_t\}_{t=1}^T$ , (поступательно-го изменения амплитуды сигнала), являющегося суммой амплитуд звуковых волн, возможна постановка альтернативной задачи – о генерации «продолжения» исходного фрагмента методами прогнозирования нестационарных временных рядов [1, 2, 3].

Нестационарный временной ряд обладает меняющейся со временем функцией распределения, и в простейшем случае может быть представлен в виде двух составляющих: детерминированной и случайной компонент [3]:

$$Y(t) = f(t) + \varepsilon(t), \quad (1)$$

где  $f(t)$  – детерминированная составляющая,  $\varepsilon(t)$  – шумовая составляющая.

Музыкальная композиция представляет собой сумму гармонических колебаний звуковых волн с разными параметрами (частота, амплитуда) и некоторой шумовой составляющей, воспроизводимых как одновременно, так и в разные промежутки времени. Рассмотрение звуковых временных рядов как совокупности отдельных компонент, может привести к увеличению точности прогнозирования (на обучении), что, в свою очередь, позволит достичь лучшего результата звучания и общей целостности полученной композиции.

Цель настоящего исследования: во-первых, представить некоторый обзор достижений в области применения технических методов к прогнозированию музыкального ряда, описание которого включает взаимовлияющие процессы; во-вторых, изложить новый подход к решению задачи «продления» фрагмента музыкальной композиции.

Под «продлением» понимается генерация продолжения композиции на основе заданного шаблона (фрагмента), части которого выступают в качестве «музыкальных паттернов», под которыми понимается не привязанная к конкретным параметрам звучания (частота, амплитуда) последовательность нот, повторяющаяся в композиции.

**Обзор существующих подходов**

1. Композиция как последовательность нот, образующая цепь Маркова [4, 5].

К положительным характеристикам отнесем возможность генерации учебных (академических) с хорошо структурированной системой правил композиций, и, как следствие, формализованного критерия оценки результата. Отметим, что в популярных жанрах отступление от правил далеко не редкость. В результатах генерации присутствует нарушение целостности структуры композиции.

2. Генетический алгоритм [5].

Положительные характеристики совпадают с таковыми для цепи Маркова. Применение генетических алгоритмов для задач генерации композиций популярных жанров рассматривает различные подходы к построению функции приспособленности особой популяции: применение критериев оценки на основе структурированной системы правил (что далеко не всегда является возможным); привлече-

ние экспертов (людей), которым требуется оценить результаты многих поколений популяций множества особей (что также проблематично).

3. Алгоритм на основе гармонического поиска [6, 7, 8].

Основная сложность применения данного алгоритма заключается в постановки задачи генерации композиции как задачи многокритериальной оптимизации и построении функции оценки результатов генерации (проблема идентична формированию функции приспособленности при применении генетических алгоритмов).

4. Методы на основе машинного обучения [5, 7].

Дают возможность генерации разножанровых музыкальных композиций на основе соответствующих обучающих выборок. К отрицательным чертам можно отнести: а) отсутствие формализованного критерия оценки результата; б) ограничение возможных вариантов генерации обучающей выборкой.

### Постановка задачи

Рассмотрим композицию как нестационарный временной ряд поступательного изменения амплитуды сигнала, моделируемого в виде (1).

Задача прогнозирования временного ряда заключается в определении будущих значений  $\hat{y}(t)$  на заданном интервале, называемом горизонтом прогноза ( $h$ ) с заданным уровнем доверительной вероятности [1, 2, 3].

Критерием качества прогноза выберем меру близости к исходной мелодии, заданной как train-выборка (кроме субъективной оценки с позиции инициатора генерации фрагмента мелодии).

### Решение задачи

Сложность анализа временных рядов, порождаемых музыкальным инструментом, заключается не только в его многомерности (частота, качество звука, громкость сопоставленные высоте звука, тембру и амплитуде, соответственно), но и в различии этих характеристик для разных инструментов. Однако существует метод эмпирической модовой декомпозиции сигнала (Empirical Mode Decomposition, EMD [9]), в основе которого лежит принцип: каждый реальный процесс (линейный/нелинейный, стационарный/нестационарный) есть колебание, в определенной степени «симметричное» относительно локального среднего значения, что делает его привлекательным для решения поставленной задачи.

Суть метода EMD заключается в разложении произвольного временного ряда на семейство функций внутренних мод (intrinsic mode function, IMF), каждая из которых есть модель определенного колебательного режима реального процесса, а их сумма есть исходный анализируемый ряд.

Выполним алгоритм EMD (приложение А) для следующего фрагмента музыкальной композиции, длительностью 0,1 с, частота дискретизации 44,1 кГц, размер выборки составляет 4410 значений.

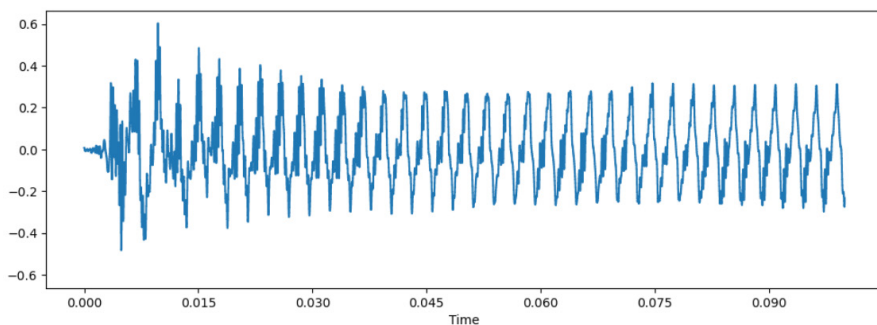


Рис. 1. Представление исходного временного ряда, на горизонтальной оси отмечено время (мс), на вертикальной оси – амплитуда звука

**Основные положения алгоритма обработки фрагмента временного ряда для формирования прогноза**

1. Декомпозиция временного ряда на основе метода EMD (рис. 1).

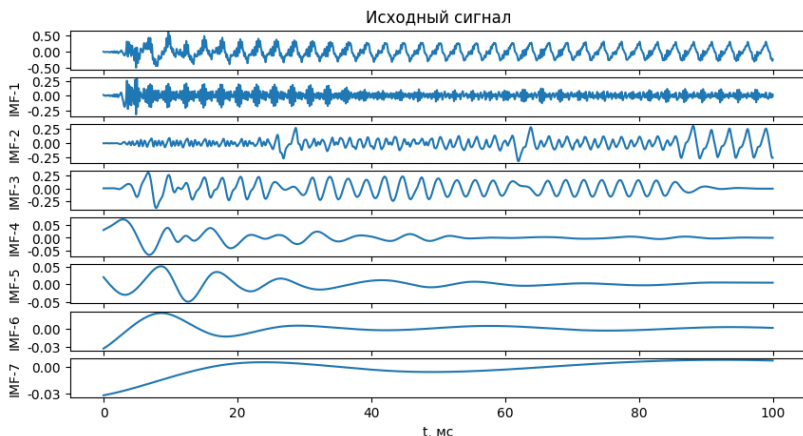


Рис. 2. Результат применения EMD

2. Вычисление степени вклада внутренней моды IMF в исходный сигнал. С этой целью приведем далее используемые показатели качества операций, применяемых к временному ряду (табл. 1).

Таблица 1

**Некоторые показатели качества результатов декомпозиции, анализа и прогнозирования временного ряда**

Показатель качества (отбора IMF, прогноза)	Комментарий
$SNR = \frac{P_{\text{сигнала}}}{P_{\text{шума}}} = \left( \frac{A_{\text{сигнала}}}{A_{\text{шума}}} \right)^2,$ $SNR' = 10 \log_{10}(SNR),$ $A = \sqrt{\frac{1}{T} \sum_{i=1}^T a_i^2}.$	Отношение мощности полезного сигнала к мощности шума (ОСШ, SNR); SNR' – альтернативная форма показателя; P – средняя оценка мощности сигнала и шума, A – среднеквадратическая оценка амплитуды [10]
$RMSE = \sqrt{\frac{1}{T} \sum_{i=1}^T (\hat{y}_i - y_i)^2}.$	$y_i, \hat{y}_i$ – исходное и восстановленное значения сигнала в $i$ -й промежуток времени, соответственно.
$K = \lim_{\varepsilon \rightarrow 0} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{n=0}^{T-1} (K_{n+1} - K_n) =$ $= - \lim_{\varepsilon \rightarrow 0} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{i_0 \dots i_T} P_{i_0 \dots i_N} \ln P_{i_0 \dots i_T}.$	Энтропия Колмогорова-Синая (КС-энтропия) [11, 12] характеризует среднюю скорость потери информации по времени; $K_{n+1} - K_n$ – потери информации о системе на промежутке $\{n, n+1\}$ ; $\{P_{i_0 \dots i_N}\}$ – конечномерное распределение вероятности состояния системы

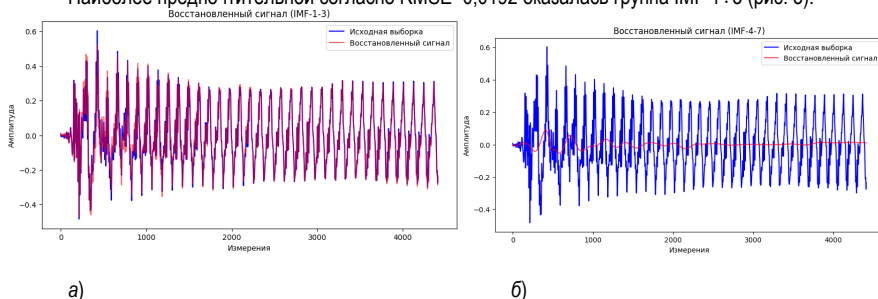
Показатель качества (отбора IMF, прогноза)	Комментарий
$\hat{K} = \frac{1}{t} \ln \frac{C_m(\varepsilon)}{C_{m+1}(\varepsilon)}$	$\hat{K}$ – оценка $K$ , применяемая на практике, $C_m$ – корреляционный интеграл, $t$ – шаг (временная задержка), $m$ – размерность вложения (минимальное число динамических переменных, однозначно описываемых наблюдаемый процесс); $\varepsilon$ – относительная погрешность данных о состоянии системы, породившей временной ряд [11, 12]
$T_m \sim \frac{1}{K} \ln \frac{1}{\varepsilon}, T'_m \sim K^{-1}$	Оценка горизонта прогноза на основе энтропии; $K$ – энтропия, показывающая насколько система хаотична, которую можно определить как информационную энтропию или КС-энтропию; $T'_m$ – упрощенный вид в силу малости влияния $\varepsilon$ [11]

Сформулируем правило: степень вклада внутренней моды IMF в исходный сигнал обратно пропорционально отношению SNR.

Значения показателя SNR для IMF 1÷7 соответственно составляют: 9,1284; 5,3178; 2,8825; 19,2458; 21,0243; 27,8234; 26,9553.

3. Отбор мод из полученного множества как основу построения прогноза согласно SNR (табл. 1) с применением алгоритма кластеризации (здесь использовано расстояние Евклида). Согласно показателю SNR рассмотрены 2 группы: IMF 1÷3 (максимальный вклад в сигнал) и IMF 4÷7 (минимальный вклад в сигнал).

Наиболее предпочтительной согласно RMSE=0,0192 оказалась группа IMF 1÷3 (рис. 3).



а) б)  
 Рис. 3. Результат суммирования мод а) IMF 1÷3 (красное начертание) и б) – IMF 4÷7, соответственно, в сравнении с исходным (синее начертание)

Оценка горизонта прогноза составляет:

Исходный фрагмент:  $K_2^{-1} \approx 2,2161, m=7$ .

IMF 1÷3:  $K_2^{-1} \approx 3,1412, m=3$ .

4. Выделение «музыкальных паттернов» может быть выполнено на основе:

- 1) сезонной составляющей временного ряда (применима модель SARIMA [3, 14]);
- 2) «гармоничности» композиции (исходный временной ряд разделяется на фрагменты исходя из соотношения «золотого сечения»);
- 3) «ритмичности» композиции (фрагментация временного ряда выполняется на основе значения показателя BPM (beats per minute – число ударов в минуту)).

### Заключение

Под «автоматизацией искусства» понимают роботизацию некоторых простых процессов через совершенствование и разработку методов, алгоритмов для целевой аудитории определенной направ-



ленности. В рамках данной работы были сформулированы основные положения алгоритма обработки фрагмента музыкального временного ряда для дальнейшего формирования прогноза.

### Библиографический список

1. Hyndman R. J. Forecasting: principles and practice. Athanasopoulos. Australia: Monash University, 2018. 504 с.
2. Лоскутов А. Ю. Анализ временных рядов: курс лекций / Физический факультет МГУ. URL: [https://chaos.phys.msu.ru/loskutov/PDF/Lectures\\_time\\_series\\_analysis.pdf](https://chaos.phys.msu.ru/loskutov/PDF/Lectures_time_series_analysis.pdf). (дата обращения: 09.05.2023).
3. Колесникова С. И. Математические модели в исследовании систем: учеб. пособие. СПб.: ГУАП, 2020. 150 с.
4. Sharpio I., Huber M. Markov chains for computer music generation // Journal of humanistic mathematics. 2021. Vol. 11 (2). P. 167-195. DOI:10.5642/jhummath.202102.08.
5. Towsey M., Brown A., Wright S., Diederich J. Towards Melodic Extension Using Genetic Algorithms. 2001. Vol. 4 (2). URL: [https://www.researchgate.net/publication/27462953\\_Towards\\_melodic\\_extension\\_using\\_genetic\\_algorithms](https://www.researchgate.net/publication/27462953_Towards_melodic_extension_using_genetic_algorithms) (дата обращения: 09.05.2023).
6. Music-Inspired Harmony Search Algorithm. Theory and Applications // Studies in Computational Intelligence. URL: <https://doc.lagout.org/5D.pdf>. (дата обращения: 09.05.2023).
7. Geem Z. W., Choi J. Y. Music Composition Using Harmony Search Algorithm // Giaco-bini. M. (eds) Applications of Evolutionary Computing. EvoWorkshops. Lecture Notes in Computer Science. Springer, Berlin. 2007. URL: <https://doi.org/71805-5> (дата обращения: 09.05.2023).
8. Xing B. Innovative computational intelligence. A rough guide to 134 clever algorithms. Switzerland: Springer International Publishing, 2014. 469 с.
9. Huang N. E., Shen Z., Long S. R., Wu M. C., Shih H. H., Zheng Q., Yen N.-C., Tung C. C., Liu H. H. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proceedings of R. Soc. London, Ser. A, 1998. 454 с.
10. Welvaert M., Yves R. On the definition of signal-to-noise ratio and contrast-to-noise ratio for fMRI data. 2013. URL: <https://journals.plos.org/10.1371> (дата обращения: 09.05.2023).
11. Лоскутов А. Ю., Козлов А. А., Хаханов Ю. М. Энтропия и прогноз временных рядов в теории динамических систем // Изв. вузов «ПНД». 2009. Т. 17. № 4. С. 98–113.
12. Grassberger P., Procaccia I. Estimation of the Kolmogorov entropy from a chaotic signal // The American Physical Society. 1983. Vol. 28 (4). P. 28–30. URL: <https://courses.physics.ucsd.edu/Procaccia-Signal.pdf> (дата обращения: 09.05.2023).
13. Лайонс Р. Цифровая обработка сигналов. М.: Бином, 2006. 656 с.
14. Нильсон Э. Практический анализ временных рядов: прогнозирование со статистикой и машинное обучение. СПб.: Диалектика, 2021. 544 с.

УДК 528.854

**С. А. Шквиро**

студент кафедры компьютерных технологий и программной инженерии

**М. Д. Поляк** – старший преподаватель– научный руководитель

## ОПРЕДЕЛЕНИЕ РЯДОВ ВИНОГРАДА ПО ДАННЫМ ДЗЗ

### Введение

В последнее время наблюдается растущий интерес к применению дронов и систем дистанционного зондирования земли (ДЗЗ) для сбора различных данных, включая данные о сельскохозяйственных культурах. Виноградарство является одной из наиболее важных и развивающихся отраслей сельского хозяйства, и его эффективное управление требует точных и надежных методов определения урожайности и качества урожая. В данной статье рассматривается метод определения рядов винограда по данным ДЗЗ с применением алгоритмов обработки изображений. Представленные результаты исследования могут быть полезными для улучшения управления виноградарством и повышения его эффективности.

### Постановка задачи

Цель данной научной статьи заключается в разработке и определении рядов винограда по данным, полученным с дронов при использовании дистанционного зондирования земли. Анализ и обработка этих данных будут проводиться с использованием алгоритмов обработки изображения. Ожидается, что конечные результаты данной работы обеспечат более точную и эффективную оценку кустарников в сельском хозяйстве, что повысит эффективность управления урожаями и повысит производительность виноградарских хозяйств.

### Исходные данные

Исходными данными для данного задания является изображение формата TIFF размером 10000 на 10000 пикселей. TIFF стоит за Tagged Image File Format и является распространенным форматом для хранения цифровых изображений. В отличие от некоторых других форматов, таких как JPEG, он не использует сжатие с потерей, что означает, что изображение сохраняется в высоком качестве.



Рис. 1. Исходное изображение

Для определения рядов винограда используется преобразование изображения путем вычисления индекса разности для каждого пикселя [1]:

$$2G\_RBi = 2G - (R + B);$$

где  $2G\_RBi$  – индекс разности,  $G$  – значение яркости канала зеленого цвета,  $R$  – значение яркости канала красного цвета,  $B$  – значение яркости канала синего цвета.

В результате данного преобразования получается новая матрица, которую можно представить в виде черно-белого изображения. Изображение было повторно обработано и было присвоено значение 0 или 255 в зависимости от того, на какой стороне порога находилось значение на предыдущем этапе обработки. Порог был вычислен эмпирически и равняется 60.

Также для определения рядов винограда использовался метод вычисления процента зеленого для каждого пикселя [1]:

$$\%G = \frac{G}{G + R + B};$$

где  $\%G$  – процент зеленого,  $G$  – значение яркости канала зеленого цвета,  $R$  – значение яркости канала красного цвета,  $B$  – значение яркости канала синего цвета.

Как и в предыдущем методе проводится дополнительная обработка с порогом 0.405, вычисленным эмпирически.



Рис. 2. Индекс разности

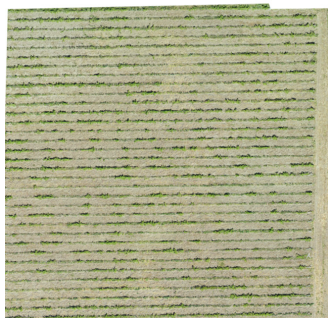


Рис. 3. Процент зеленого

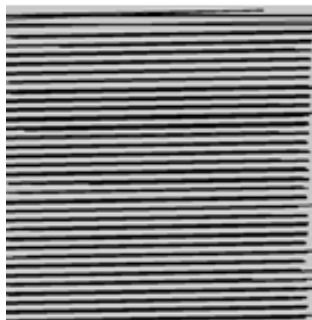
### Нахождение рядов

Для дальнейшей обработки была выбрана метрика индекса разности, как наиболее верно отражающей ситуацию. На изображениях ряды представлены не как абсолютно прямые линии, в связи с этим был выбран алгоритм преобразования Хафа. Алгоритм Хафа может обнаружить прямые даже в том случае, если они искажены, размыты, повернуты. Помимо этого, у данного алгоритма настраиваются параметры поиска линий, что делает его гибким инструментом для обработки изображения [2].

Алгоритм показывает большую точность при обработке изображения меньшего размера, в связи с чем был взят сегмент виноградного поля.



*Рис. 4. Исходный сегмент изображения*



*Рис. 5. Вычисленные ряды, нарисованные с помощью алгоритма преобразования Хафа*

На рис. 5. изображены только линии, полученные с помощью алгоритма преобразования Хафа.

### **Заключение**

Использование дронов для сбора данных, метрика разности, а также преобразование Хафа для вычисления рядов позволило получить точные результаты и определить расположение рядов винограда на участке. Такой подход может быть полезен для сельскохозяйственных предприятий и фермеров, которые могут использовать эти данные для повышения эффективности сбора урожая и более точного планирования обработки полей.

### **Библиографический список**

1. Carlos Poblete-Echeverría. Detection and Segmentation of Vine Canopy. URL: <https://www.mdpi.com/2072-4292/9/3/268> (дата обращения: 27.03.2023).
2. Документация OpenCV. URL: <https://docs.opencv.org> (дата обращения 27.03.2023).

УДК 004.942

**А. А. Щеголева**

студент кафедры компьютерных технологий и программной инженерии

**С. И. Колесникова** – доктор технических наук, профессор – научный руководитель**М. Д. Поляк** – старший преподаватель – научный консультант

### ПРИМЕНЕНИЕ МЕТОДА АНАЛИТИЧЕСКОГО КОНСТРУИРОВАНИЯ АГРЕГИРОВАННЫХ РЕГУЛЯТОРОВ К ТРЕХКОМПОНЕНТНЫМ МОДЕЛЯМ ВЗАИМОДЕЙСТВИЯ ПИТАНИЯ, ФИТОПЛАНКТОНА И ЗООПЛАНКТОНА

Согласно многочисленным отчетов и научных публикаций [1, 2] цветение водорослей может привести к нехватке кислорода в воде, что ведет к экологической деградации водоема. В последние десятилетия это отмечается в Балтийском море. На рис. 1, а представлено среднее пространственное распределение биомассы фитопланктона. Как видно начиная с 50-х гг. прошлого века до настоящего времени концентрация биомассы фитопланктона неуклонно растет, что повышает вероятность возникновения цветения. На рис. 1, б представлена карта с изображением частоты появления цветения разных видов планктона и других морских существ. Основным источником цветения является цианобактерия, цветение которой как видно из рисунка стало частым явлением в Балтийском море.

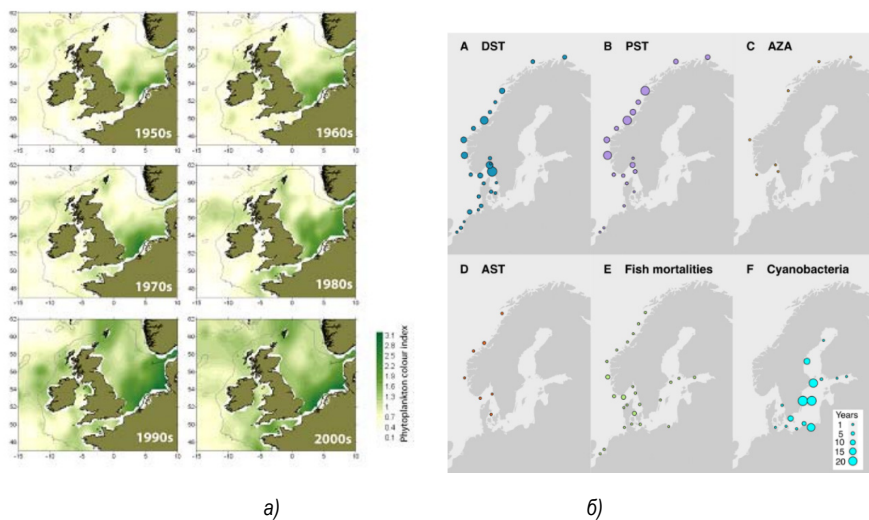


Рис. 1. Среднее пространственное распределение запаса фитопланктона за каждое десятилетие с 1950-х годов по настоящее время (а); распределение вредных явлений, связанных с водорослями в период с 1987 по 2019 год (б)

#### Модель «питание-фитопланктон-зоопланктон»

Целью работы является создание новой модели цветения водорослей, основанной на трехкомпонентной модели питания-фитопланктон-зоопланктон (NPZ) [3, 4] и теории синергетического управления [5, 6]. Применение методики синергетического управления дает возможность создавать собственную регуляцию систем по заданным параметрам. В качестве базовой модели используется NPZ-

модель, где  $x_1$  – плотность питания,  $x_2$  – плотность фитопланктона,  $x_3$  – плотность зоопланктона. NPZ-модель является наиболее общей моделью, отражающей взаимодействие фитопланктона, зоопланктона и питательных элементов. Она основана на общей модели Лотка-Вольтерра (хищник-жертва)

$$\begin{cases} \frac{dx_1}{dt} = ax_2 + bx_3 - cx_1x_2 \\ \frac{dx_2}{dt} = cx_1x_2 - dx_2x_3 - ax_2 \\ \frac{dx_3}{dt} = dx_2x_3 - bx_3 \end{cases} \quad (1)$$

### Модель с управлением

Первой целью управления является достижение фитопланктоном заданного значения  $\psi = x_2 - x_2^*$ .

$$\begin{cases} \frac{dx_1}{dt} = ax_2 + bx_3 - cx_1x_2 + u \\ \frac{dx_2}{dt} = cx_1x_2 - dx_2x_3 - ax_2 \\ \frac{dx_3}{dt} = dx_2x_3 - bx_3 \\ u = -\frac{\psi^{(l)}}{T_1} - ax_2 - bx_3 + cx_1x_2 + \frac{d\varphi}{dt} \\ \psi(t) = x_2(t) - x_2^* \\ \psi^{(l)}(t) = x_1(t) - \varphi(t) \\ \varphi(t) = \frac{-\frac{\psi(t)}{T_2} + dx_2(t)x_3(t) + ax_2(t)}{cx_2(t)} \\ \frac{d\varphi(t)}{dt} = \frac{\left(-\frac{1}{T_2} + dx_3 + a\right)x_2 - \left(-\frac{\psi}{T_2} + dx_2x_3 + ax_2\right)}{cx_2^2} f_2 + \frac{d}{c} f_3 \end{cases} \quad (2)$$

Такая цель позволит построить простейшую модель для прогнозирования нарастания или спада цветения. Управление здесь можно интерпретировать как питание фитопланктона биогенными элементами, т. е. изменение плотности биогенных элементов влияет на развитие цветения.

Стационарные точки системы:

1.  $E_0(x_1^*, x_2^*, 0)$ , где  $x_1^* = \frac{a}{c}, x_2^*$  – целевое значение популяции фитопланктона.
2.  $E_1(x_1^*, x_2^*, x_3^*)$ , где  $x_1^* = \frac{dx_3 + a}{c}, x_2^*$  – целевое значение популяции фитопланктона.

На рис. 2 представлено численное моделирование системы с управлением (2) по методу АКАР, где наблюдается нарастание цветения. Как видно из рисунка со временем популяция фитопланктона

увеличивается до пикового значения. Количество биогенных элементов, необходимых для цветения, здесь регулируется введенным управлением.

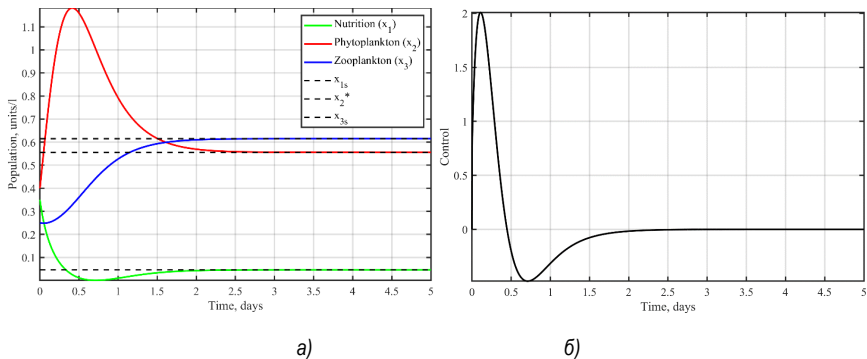


Рис. 2. Численное решение системы (а); изменение функции управления (б)

Второй целью управления является достижение популяцией фитопланктона и количеством питания пропорциональных ( $\rho$  – коэффициент пропорциональности) друг другу значений при заданной емкости  $q$  среды  $\psi = x_2 + \rho x_1 - q$

$$\begin{cases} \frac{dx_1}{dt} = ax_2 + bx_3 - cx_1x_2 + u \\ \frac{dx_2}{dt} = cx_1x_2 - dx_2x_3 - ax_2 \\ \frac{dx_3}{dt} = dx_2x_3 - bx_3 \\ u = -\frac{\psi}{\rho T_1} - \frac{f_2}{\rho} - ax_2 - bx_3 + cx_1x_2 \\ \psi(t) = x_2(t) + \rho x_1(t) - q \end{cases} \quad (3)$$

Стационарные точки системы:

1.  $E_0(x_1^*, 0, 0)$ , где  $x_1^* = \frac{q}{\rho}$ .
2.  $E_1(x_1^*, x_2^*, 0)$ , где  $x_1^* = \frac{a}{c}$ ,  $x_2^* = \frac{cq - a\rho}{c}$ .
3.  $E_2(x_1^*, x_2^*, x_3^*)$ , где  $x_1^* = \frac{dq - b}{d\rho}$ ,  $x_2^* = \frac{b}{d}$ ,  $x_3^* = \frac{cdq - bc - ad\rho}{d^2\rho}$ .

На рис. 3 представлено численное моделирование системы с управлением (3). Как видно из рисунка со временем популяция фитопланктона увеличивается пикового значения. Количество биогенных элементов, необходимых для цветения, здесь регулируется введенным управлением.

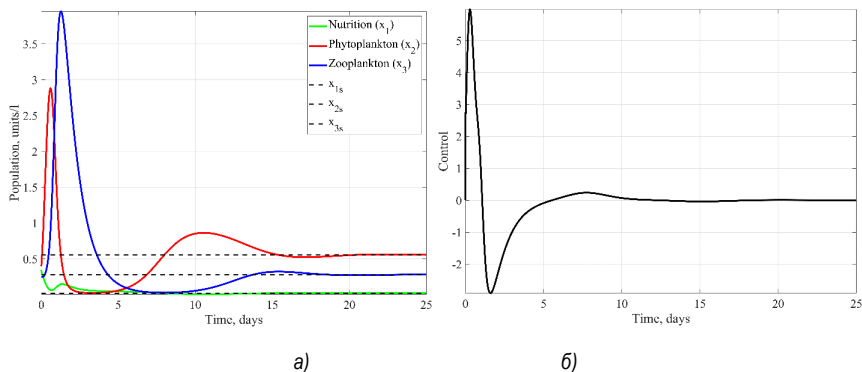


Рис. 3. Численное решение системы (а); изменение функции управления (б)

Третьей целью управления является достижение фитопланктоном и зоопланктоном пропорциональных друг другу значений при заданной емкости среды

$$\begin{cases}
 \frac{dx_1}{dt} = ax_2 + bx_3 - cx_1x_2 + u, \\
 \frac{dx_2}{dt} = cx_1x_2 - dx_2x_3 - ax_2, \\
 \frac{dx_3}{dt} = dx_2x_3 - bx_3, \\
 u = -\frac{\Psi(l)}{T_1} - ax_2 - bx_3 + cx_1x_2 + \frac{d\varphi(t)}{dt}, \\
 \frac{d\varphi(t)}{dt} = \frac{\left(-\frac{1}{T_2} + dx_3 + a - \rho dx_3\right)x_2 - \left(-\frac{\Psi}{T_2} + dx_2x_3 + ax_2 - \rho(dx_2x_3 - bx_3)\right)}{cx_2^2} f_2 + \\
 + \frac{-\frac{\rho}{T_2} + dx_2 - \rho(dx_2 - b)}{cx_2} f_3, \\
 \varphi(t) = \frac{-\frac{\Psi}{T_2} + dx_2x_3 + ax_2 - \rho(dx_2x_3 - bx_3)}{cx_2}, \\
 \Psi(t) = x_2(t) + \rho x_3(t) - q, \\
 \Psi(l)(t) = x_1(t) - \varphi(t).
 \end{cases} \quad (4)$$

Стационарные точки системы

1.  $E_0(x_1^*, x_2^*, 0)$ , где  $x_1^* = \frac{a}{c}$ ,  $x_2^* = q$ .
2.  $E_1(x_1^*, x_2^*, x_3^*)$ , где  $x_1^* = \frac{a\rho - b + dq}{c\rho}$ ,  $x_2^* = \frac{b}{d}$ ,  $x_3^* = \frac{dq - b}{d\rho}$ .



На рис. 4 представлено численное моделирование системы (4). Как видно из рисунка со временем популяция фитопланктона увеличивается до устоявшегося значения согласно стационарной точке.

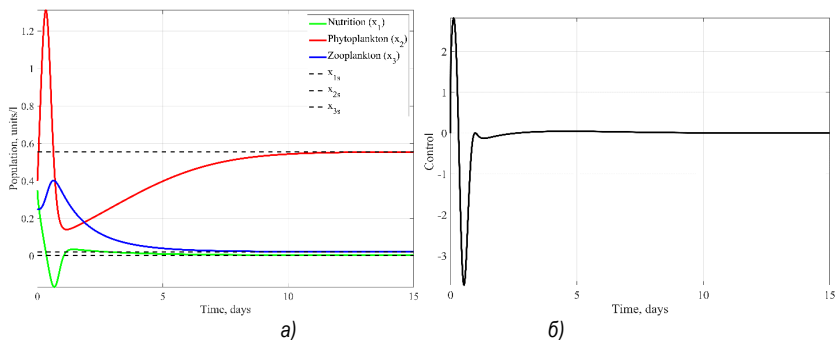


Рис. 4. Численное решение системы (а); изменение функции управления (б)

### Заключение

При выполнении работы была сформулирована и доказана следующая гипотеза: математическое моделирование с применением последних достижений синергетической теории управления дает возможность строить управляемые объекты, численные решения, которых являются устойчивыми, отвечают принципам оптимального управления и имеют биологическую интерпретацию. С помощью полученных моделей возможно по-разному управлять численностью популяций, в частности фитопланктона и зоопланктона, в зависимости от поставленной задачи.

*Работа выполнена, как часть гранта РФФ №23-29-00336.*

### Библиографический список

1. Licandro Priscilla, et al. Overview of trends in plankton communities. ICES. URL: <https://oar.marine.ie/bitstream/handle/10793/1269/Licandro%20et%20al.%20Overview%202011.pdf?sequence=1&isAllowed=y> (дата обращения: 24.05.2023).
2. Karlson B., Andersen P., Arneborg L., Cembella A., Eikrem W., John U., Suikkanen S. Harmful algal blooms and their effects in coastal seas of Northern Europe // *Harmful Algae*. 2021. № 102. С. 101989.
3. Franks P.J.S. NPZ Models of Plankton Dynamics: Their Construction, Coupling to Physics, and Application // *Journal of Oceanography*. 2002. № 58 (2). С. 379–387.
4. Asif M. Numerical modeling of NPZ and SIR models with and without diffusion // *Results in Physics*. 2020. № 19. С. 103512.
5. Колесников А. А. Синергетика и проблемы теории управления. М: ФИЗМАЛИТ, 2004. 504 с.
6. Kolesnikova S. I. Synthesis of the Control System for a Second Order Non-Linear Object with an Incomplete Description // *Autom Remote Control*. 2018. № 79. С. 1558–1568.

УДК 656

**А. М. Яппаров**

студент кафедры информационных технологий и программирования

**А. В. Фомин** – кандидат технических наук, доцент – научный руководитель

### ТЕХНОЛОГИЯ ИНТЕРНЕТА ВЕЩЕЙ В ЛОГИСТИКЕ И ТРАНСПОРТНОЙ СИСТЕМЕ

Эффективное управление логистикой зависит от контроля за множеством видов деятельности, таких как поставки, хранение и транспортировка товаров. Но в этом процессе возникает множество факторов, которые могут привести к задержкам и негативным последствиям. Чтобы повысить качество обслуживания и оптимизировать процессы, используются различные инновационные технологии. Одной из наиболее популярных является интернет вещей (Internet of Things, IoT), которая позволяет контролировать работу всех элементов цепочки поставок и своевременно реагировать на изменения в процессе.

**Модель Интернета вещей.** Отдел стандартов электросвязи Международного союза электросвязи (МСЭ-T) в Рекомендации Y.2060 дает следующее объяснение понятия «Интернет вещей» – это глобальная инфраструктура информационного общества, обеспечивающая передовые услуги с помощью организации связи между вещами (физическими или виртуальными) на основе существующих и развивающихся совместимых информационных и коммуникационных технологий [1]. Так же приводит эталонную модель архитектуры системы «Интернет вещей». Согласно модели, есть 6 уровней (рис. 1).



Рис. 1. Эталонная модель IoT

IoT-приложения используются в разных отраслях. Сеть IoT поддерживается несколькими уровнями, такими как приложения, сервисы, сетевой и уровень устройств.

Уровень приложений обрабатывает данные и реализует IoT-приложения для разных отраслей.

Уровень поддержки сервисов и приложений включает возможности по обработке и хранению данных как для всех объектов IoT, так и для конкретных подмножеств приложений.

Сетевой уровень объединяет функции управления ресурсами и транспортной сети, а уровень устройств предоставляет возможности для обмена данными между устройством и сетью связи через шлюз или беспроводную сеть. Шлюзы также обеспечивают поддержку интерфейсов для устройств и сетевых соединений, а также конверсию протоколов. Протоколы связи классифицируются по радиусу действия – малый, средний и дальний.

Уровень управления включает управление устройствами, локальными сетями и трафиком, а также обеспечивает безопасность и конфиденциальность данных.

Уровень безопасности зависит от горизонтального уровня и включает функции для авторизации, аутентификации, защиты информации и контроля доступа. Уровни управления и безопасности охватывают все четыре горизонтальных уровня.

Одна из особенностей IoT – получение ценных данных. Эти данные не структурированы, в связи с этим приоритетной задачей является правильная фильтрация и адекватная интерпретация. Правильная обработка и анализ позволяют значительно повысить производительность и эффективность бизнес-процессов в сфере логистики и транспорта, вследствие чего оптимизируется вся система, включая складские операции, перевозку, доставку.

Сфера логистики идеально подходит для технологических достижений больших данных. Логистические и транспортные компании управляют непрерывным потоком товаров, тем самым создают большие объемы данных. Аналитика этих данных используется для качественного принятия решений, оптимизации сервисов.

IoT-платформы являются неотъемлемой частью инфраструктуры, предназначенной для обмена данными между конечными устройствами и облачным хранилищем. Они позволяют поддерживать различные типы коммуникаций и успешно используются в производстве, промышленности, системах наблюдения и приложениях автоматизации [2].

В IoT широко применяются облачные технологии. Данные от различных периферийных устройств интегрируются, систематизируются и анализируются на облачном сервере с использованием методов искусственного интеллекта, таких как машинное обучение. Результаты интеллектуального анализа данных визуализируются в виде графиков, диаграмм, таблиц и т.д., что позволяет пользователям IoT-платформы получать актуальную информацию о состоянии системы (рис. 2).



Рис. 2. Передача данных с устройства в облако и обратно

Логистические процессы довольно сложны, поэтому для эффективного функционирования компаниям крайне важно получать всю информацию «здесь и сейчас». Сам процесс обычно включает в себя несколько этапов, которые требуют пристального внимания и постоянного совершенствования. Некоторые части процесса, которые могут извлечь выгоду после внедрения интернета вещей [3]:

- контроль транспорта;
- хранение и складирование;
- планирование;
- управление запасами.

Возможность наблюдения за продукцией по всей цепи поставок посредством современных решений повышает безопасность транспортировки. Камеры, GPS, радиочастотные метки (RFID), различные датчики позволяют собирать данные о местоположении, скорости, температуре, давлении и т. д [4].

Интернет вещей объединяет транспорт, склады и товары в одну единую сеть. Современные складские комплексы оснащены системами управления складом (Warehouse Management System, WMS). Такая система, к примеру, получает данные от штрих-кодов и RFID-меток, которые размещены на товарах. Система контроля склада (Warehouse Control System, WCS) – система на уровень выше WMS, так как датчиками и сенсорами оснащается и складское оборудование, например, полки, контейнеры. Для контроля и управления условиями среды внутри склада используется система автоматизации зданий (Building Automation Systems, BAS). С ее помощью производится контроль освещением, вентиляцией воздуха, управление камерами, а также обеспечивается безопасность в случае нестандартных ситуаций.

Все эти системы используются в первую очередь для получения информации о товарно-материальных запасах, оценке их состояния. Так же реализуются условия безопасного хранения и предотвращения потерь.

Наиболее распространенные варианты использования IoT в сфере логистики:

*Отслеживание местоположения.* Использование специальных датчиков на транспорте и грузах позволяет отслеживать их местоположение в режиме реального времени, а также регистрировать любые изменения в процессе доставки. Для отслеживания данных используются технологии радиочастотной идентификации (RFID), GPS, GPRS и географической информационной системы (ГИС). Эти технологии позволяют получать точную информацию о местоположении транспорта и груза, статусе доставки, что повышает прозрачность процесса и позволяет управлять доставкой более эффективно. Заказчик может получить актуальную информацию о статусе доставки и уведомления о любых изменениях, что позволяет ему планировать свое время и принимать необходимые меры в случае непредвиденных обстоятельств.

*Оптимизация маршрута.* Система способна получать данные в режиме реального времени от устройств, расположенных на дорогах, анализировать их и делать прогнозы. Основываясь на постоянном потоке информации от камер, детекторов дорожного движения, спутников, происходит оценка маршрутов и их корректировка, тем самым оптимизируется дальнейшее движение. Так же позволяет снизить затраты на топливо, снизить риски дорожно-транспортных происшествий, простоев, износа оборудования, что положительно сказывается на экономии средств.

*Отслеживание эксплуатационных характеристик и параметров и их корректировка.* Беспроводные сенсорные сети могут контролировать условия окружающей среды издалека, благодаря чему возможно удаленное управление этими условиями в режиме реального времени. Например, датчики, подключенные к Интернету вещей, могут постоянно собирать данные о перевозке грузов, такие как температура, давление, освещенность и влажность. Эти данные передаются на простейший компьютер – микроконтроллер, который считывает сигналы от датчиков, после чего информация отправляется на сервер для дальнейшей обработки.

*Прогнозная аналитика.* Аналитика – это особая часть сферы логистики. Прогнозы позволяют вводить более эффективные стратегии, оптимизировать процесс принятия решений, генерировать эффективные бизнес-идеи, управлять рисками. Использование аналитических решений на базе Интернета вещей дает более точные прогнозы и повышает производительность. Системы анализируют огромные объемы данных из различных источников внутри организации и за ее пределами [5]. Системы прогнозной аналитики повышают производительность складов и сокращают затраты на доставку.

*Обслуживание.* Устройства IoT используются для мониторинга производительности и состояния транспортных средств и оборудования, такого как двигатели, тормоза, шины, для прогнозирования и предотвращения отказов. Датчики анализируют вибрацию, звуковые частоты, температуру и т. д., и на основе этих данных отправляют запрос на обслуживание.

Интеллектуальными объектами, включенными в Интернет вещей, может быть и транспорт. В таком случае они образуют Internet of Vehicles (Connected Cars) – Интернет транспорта. Он объединяет аппаратные средства и различные сети, которые позволяют автомобилям, пешеходам и различным объектам на дороге обмениваться информацией в режиме реального времени, и для этого существует несколько типов подключений [6]:

V2V (транспорт-транспорт) – для обмена данными между транспортными средствами (ТС);

V2I (транспорт-инфраструктура) – обмен данными между ТС и дорожной инфраструктурой (камеры, светофоры и т.д.);

V2H (транспорт-человек) – для «общения» ТС и пешеходов или, например, велосипедистов;

V2N (транспорт-сеть) – реализует взаимодействие бортовых систем с внешним миром.

Предполагается, что Интернет транспорта будет служить важной платформой для сбора и обработки данных для интеллектуальных транспортных систем. Данные, полученные с улиц, отчетов пользователей транспорта, камер наружного наблюдения, детекторов дорожного движения и других источников используются для управления трафиком в городе. Используя эти данные, система может определить оптимальное расписание работы пассажирского транспорта и принимать другие меры для снижения трафика на дорогах. Это позволяет повысить качество предоставляемых услуг, уменьшить время ожидания и сократить задержки в пути для пассажиров. Более эффективное использование ресурсов также позволяет сократить затраты на обслуживание и эксплуатацию транспортной инфраструктуры, что в конечном итоге может привести к экономическим выгодам для города [7].

*Примеры использования.* SkyCell, технологическая компания, разработала рефрижераторные контейнеры, которые позволяют отслеживать даже самые маленькие изменения параметров груза. Датчики IoT в каждом контейнере SkyCell подключаются к облачной платформе, что позволяет удаленно контролировать и следить за состоянием каждого контейнера, гарантируя высокое качество перевозки независимо от его местоположения. Собранные данные используются для анализа и моделирования маршрутов перевозок, что устраняет области с наибольшим риском сбоев в цепи поставок [8].

Другим примером может служить зарубежная железнодорожная компания Union Pacific – использование IoT технологий помогает своевременно реагировать на обслуживание оборудования. Акустические, визуальные датчики, термометры, датчики, измеряющие вибрацию, устанавливаются на железнодорожные пути и на вагоны и контролируют вибрацию, целостность колес, температуру. Аналитика параметров, снимаемых датчиками в процессе эксплуатации, помогает предсказывать критические ситуации задолго до момента их наступления [9].

### Заключение

Развитие IoT в сфере логистики показывает наличие масштабной цифровой трансформации отрасли. С ростом влияния технологий увеличивается прогнозируемость, прозрачность процессов работы. Доступность детализированных данных в режиме реального времени прокладывает путь к более умной и эффективной цепочке поставок, позволяет направлять транспортный поток, планировать парковочные места и прочее. Аналогичная система контроля транспорта улучшает транспортную инфраструктуру и создает предпосылки для ее развития.

### Библиографический список

1. Обзор интернета вещей. Рекомендация МСЭ-Т Y.2060. URL: [https://www.itu.int/rec/dologin\\_pub.asp?lang=e&id=T-REC-Y.2060-201206-!!!PDF-R&type=items](https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-Y.2060-201206-!!!PDF-R&type=items) (дата обращения 15.05.2023).
2. Ядровская М. В., Поркшеян М. В., Синельников А. А. Перспективы технологии интернета вещей // *Advanced Engineering Research*. 2021. Т. 21. № 2. С. 207–217.
3. Дирко С. В. Технологии Интернета вещей в логистике: зарубежный и отечественный опыт // *Интеллектуальные ресурсы – региональному развитию*. 2021. № 1. С. 421–425.
4. Николаева Е. М., Шпилькина Т. А. Роль интернет-вещей в логистике // *Потенциал российской экономики и инновационные пути его реализации*. 2020. Т. 1. С. 187–191.
5. Шатилов А. С., Шатилов С. В. Перспективы использования технологии Интернета вещей в логистике и на транспорте // *Современные технологии управления транспортным комплексом России*. 2019. С. 255–259.
6. Свиридова Ю. Л. Технология V2X в контексте интеллектуальной транспортной системы мегаполиса // *Альманах научных работ молодых ученых Университета ИТМО*. 2022. Т. 3. Ч. 2. С. 146–151.
7. Павлов А. О. Интернет вещей в логистике // *Естественные и технические науки: проблемы трансдисциплинарного синтеза: сборник научных трудов по материалам Международной научно-практической конференции. 25 декабря 2020 года. Белгород: Общество с ограниченной ответственностью "Агентство перспективных научных исследований", 2020. С. 53–56.*

8. Примеры технологии блокчейн в логистике и ее внедрение. URL: <https://merehead.com/ru/blog/top-benefits-blockchain-logistics-use-cases/> (дата обращения: 11.04.2023).
9. Железнодорожная компания Union Pacific оказалась в десятке лучших после внедрения "Больших данных". URL: <https://www.computerra.ru/209517/union-pacific-railroad/> (дата обращения: 11.04.2023).

УДК 004.932.2

**А. С. Агеев**

студент кафедры вычислительных систем и сетей

**А. А. Деятов**

студент кафедры вычислительных систем и сетей

**К. Ф. Сафина**

студент кафедры вычислительных систем и сетей

**Е. К. Григорьев** – ассистент – научный руководитель

## О МЕТРИКАХ КАЧЕСТВА МАСКИРОВАНИЯ ИЗОБРАЖЕНИЯ

### Введение

Матричное маскирование изображений – это простой и эффективный матричный метод защиты от несанкционированного доступа к цифровым изображениям с малым периодом актуальности, при котором исходное изображение разрушается до вида близкого к шуму [1].

Анализ литературы [1 – 4] показывает, что метрик качества маскирования, характеризующих близость результирующего изображения к шуму, не существует. В работе [4] для этих целей были использованы широко известные метрики SSIM, MSE, PSNR. Тем не менее, данные метрики не позволяют объективно судить о «качестве» разрушения изображения, например, метрики SSIM и MSSIM, коэффициент корреляции, среднеквадратическая ошибка (MSE), а также пиковое отношение сигнал-шум (PSNR) для тех изображений, где отчетливо видны контуры исходного изображения и для изображения, где структура исходного изображения уже разрушена выглядят так как показано на рис. 1. Как видно, метрики SSIM, MS-SSIM и коэффициент корреляции для всех изображений близки к нулю, метрики MSE и PSNR хоть и показывают динамику, однако не позволяют сформировать явный порог, после которого контуры изображения будут полностью разрушены.

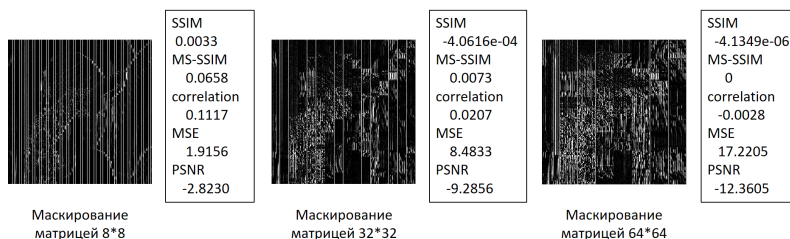


Рис. 1. Сравнение изображений разной степени маскирования

В связи с этим, актуальной является задача поиска или разработки метрики, адаптированной под специфику матричного маскирования изображений, при помощи которой можно будет понять, в какой момент контур на изображении будет разрушен, и можно будет остановить увеличение размеров матрицы маскирования.

Сформулируем цель работы – проанализировать метрики качества изображений, дать рекомендации по выбору конкретной метрики.

### Описание эксперимента

Для исследования были взяты три класса изображений: аэрофотоснимки, текстуры, и изображения с ярко выраженным объектом. Было разработано ПО для MATLAB, которое осуществляет одно-

стороннее матричное маскирование изображений различными матрицами, однако в рамках данной работы рассматривались только матрицы Адамара разного размера. Все маскированные изображения и их метрики были внесены в таблицы. После чего был проведен анализ собранных данных.

Известно такое понятие, как энтропия динамической системы, в теории динамических систем – это мера хаотичности в поведении траекторий системы, а в теории информации есть введенная Шэнноном информационная энтропия [5] – мера неопределенности источника сообщений, определяемая вероятностями появления тех или иных символов при их передаче.

Из анализа данных двух определений, было решено провести исследование по анализу зависимости величины энтропии от размера матрицы маскирования. На графиках, изображенных на рис. 2, представлены зависимость энтропии изображения от размера матрицы маскирования — сначала наблюдался рост энтропии до какого-то максимального ее значения, а затем происходил спад. Обнаружив данную закономерность в результатах, было решено проверить гипотезу о том, совпадает ли момент достижения максимума энтропии с разрушением контура изображения.

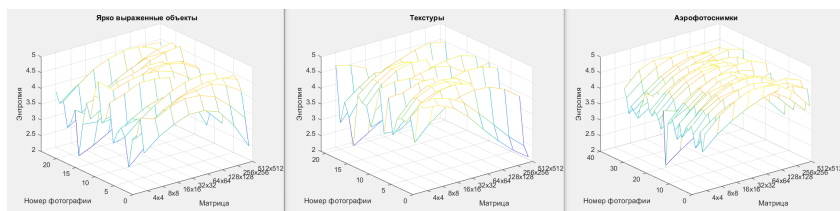


Рис. 2. Графики зависимости энтропии от размера матрицы маскирования

Для проверки гипотезы, данные об энтропиях вносились в таблицы (пример – табл. 1), где у них выделялись максимумы и размерность матрицы маскирования, при которой структура исходного изображения была разрушена. Для дополнительной проверки того, что структура изображения была разрушена, было написано специализированное программное обеспечение на MATLAB для поиска контуров на изображении. Результат представлен на рис. 3. Видно, что разрушенное изображение не различимо человеческим глазом, а также методы поиска контуров на изображении не способны их выделить.

Таблица 1

**Метрики изображения**

Изображение	Параметры			
	Энтропия	SSIM	MS-SSIM	Корреляция
Входное	7.1441	–	–	–
С шумом	4.0397	0.0500	0.4019	0.5041
4×4	3.1733	0.0054	0.2734	0.1464
8×8	3.6742	0.0021	0.0411	0.0897
16×16	4.1389	5.2117e-04	0.0135	0.0577
32×32	4.4526	-3.8127e-04	0.0076	0.0328
64×64	4.6012	2.0259e-05	0.0104	0.0199
128×128	4.5305	2.3081e-04	0.0124	0.0368
256×256	4.2324	9.4357e-05	0.0015	0.0143
512×512	3.7733	-5.7321e-05	0.0025	0.0186



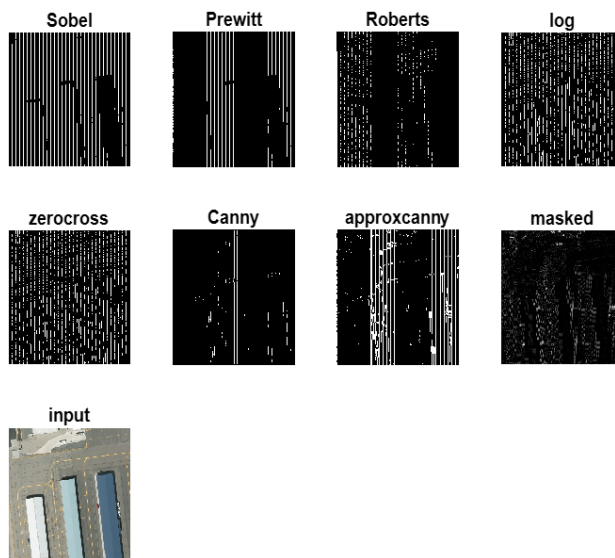


Рис. 3. Выделенные контуры изображений

### Заключение

Для аэрофотоснимков характерно разрушение изображения до точки максимума энтропии или одновременно с ней. Средняя матрица для разрушения размером  $64 \times 64$ .

Для текстур нет определенной характеристики, так как в равном количестве изображения разрушаются до точки максимума энтропии и после. На 90% изображений, достаточно матриц размером  $8 \times 8$  для разрушения контура.

Для изображений с ярко выраженным объектом тоже нет точной характеристики, так как изображения разрушаются в равной степени до и после экстремумов. В среднем для разрушения контура достаточно матрицы размером  $64 \times 64$ .

Таким образом, энтропию можно использовать в качестве некоторого маркера, однако использовать в качестве объективной метрики для оценки качества маскирования нельзя. Целесообразна дальнейшая работа в данной области.

### Библиографический список

1. Востриков А. А. Маскирование цифровой визуальной информации: термин и основные определения / А. А. Востриков, М. Б. Сергеев, М. Ю. Литвинов // Информационно-управляющие системы. 2015. № 5 (78). С. 116–123.
2. Ерш И. Л., Сергеев М. Б. Скоростное шифрование разнородных сообщений // Вопросы передачи и защиты информации: сб. статей. СПб.: ГУАП, 2006. С. 133–155.
3. Литвинов М. Ю. Алгоритмы маскирующих преобразований видеоинформации: автореф. дис. канд. техн. наук. СПб.: ГУАП, 2009. 23 с.
4. Чернышев С. А. Разработка и исследование метода матричного маскирования видеоинформации в глобально распределенных системах: автореф. дис. канд. техн. наук. СПб.: ГУАП, 2018. 19 с.
5. Shannon C. E. A mathematical theory of communication // The Bell System Technical Journal. Vol. 27. No. 3. Pp. 379–423. July 1948. Doi: 10.1002/j.1538-7305.1948.tb01338.x.

УДК 004.946

**М. М. Азаренок**

студентка кафедры вычислительных систем и сетей

**А. В. Никитин** – кандидат технических наук, доцент – научный руководитель

## МЕХАНИКА ПОДСКАЗОК В ИГРАХ ГОЛОВОЛОМКАХ

### Введение

В статье рассматривается разработка игры в жанре головоломка. В первой части проводится анализ аналогов и разработка требований для прототипа.

Доля игр в жанре головоломка составляет около 8% от общего числа игр, доступных на платформе Steam. Это означает, что в Steam доступно более 4 тыс. игр в жанре головоломка из общего числа более 53 тыс. игр.

Популярность игр данного жанра имеет несколько причин [1]:

1. Вызов – ощущение успеха, после решения сложной задачи.
2. Развлечение – относительно простые и однообразные задачи, помогающие скоротать время.
3. Персонаж и окружение – интересные персонажи и захватывающие истории.
4. Духовное развитие – поиск решения, ощущение озарения.

Для более подробного изучения механик головоломок необходимо решить следующие, поставленные задачи:

1. Исследовать механики подсказок.
2. Провести анализ существующих решений и их недостатков.
3. Выбрать средства разработки.

Рассмотрим более подробно данный жанр. Существует 4 жанра игр головоломок [1]:

1. Активные действия – требует от игрока помимо решения головоломки еще и активных действий.
2. Рассказ – сюжетные головоломки вроде MYST.
3. Конструирование – предлагает игроку создавать объекты из деталей.
4. Стратегия – зачастую встречаются в многопользовательских играх, таких как пасьянс или шахматы.

Проектируемый прототип будет представлять из себя сочетание жанров рассказа и конструирования. Также рассмотрим, какие виды подсказок существуют:

1. Ненативные – заставляют исследовать игровое пространство, использовать знакомые предметы по-новому, что может привести игрока к решению.
2. Подсказки при помощи аудио, видеофрагментов.
3. Прямые или косвенные текстовые подсказки.
4. Подсказки при помощи диалога с NPC.

### Разбор аналогов

Для выявления недостатков был приведен следующий анализ существующих аналогов.

Анализ показал отсутствие использования в играх 2 видов головоломок: пространственных и на поиск предметов, а также подсказок. Эту проблему планируется решить в разрабатываемом прототипе.

В качестве средств разработки предполагаются:

1. Blender для создания 3d-моделей.
2. Visual Studio для написания сценариев на C#.
3. Игровой движок Unity для создания самого прототипа.

Игра предназначена для ПК, игрок будет управлять персонажем при помощи клавиатуры и мыши.

Рассмотрим более подробно этапы реализации прототипа:

1. Разработка сцены. Сцена представляет из себя комнату с 2 выходами, от которых отходят лестницы, ведущие в другие комнаты. Внутри комнаты находятся игровые предметы: кубы, пирамиды.

Игрок может поднимать эти предметы, к примеру куб (нажатие на E), а затем поворачивать. После этого предмет меняет перспективу и заменяется на пирамиду. После того как игрок поймет, какой предмет куб или пирамиду нужно использовать в этой комнате, дверь открывается и пропускает игрока в другие локации.

Таблица 1

**Анализ аналогов**

Antichamber	Monument Valley	Dicsoriented	Hyper Rogue
Использование пространственных объектов, имп-артов	+	+	-
Смена перспективы, для решения головоломки	+	+	+
Использование неевклидовой геометрии	+	+	+

Для создания статической сцены в качестве референса была использована литография голландского художника Мариуца Эшера [2], представленная на рис. 1.

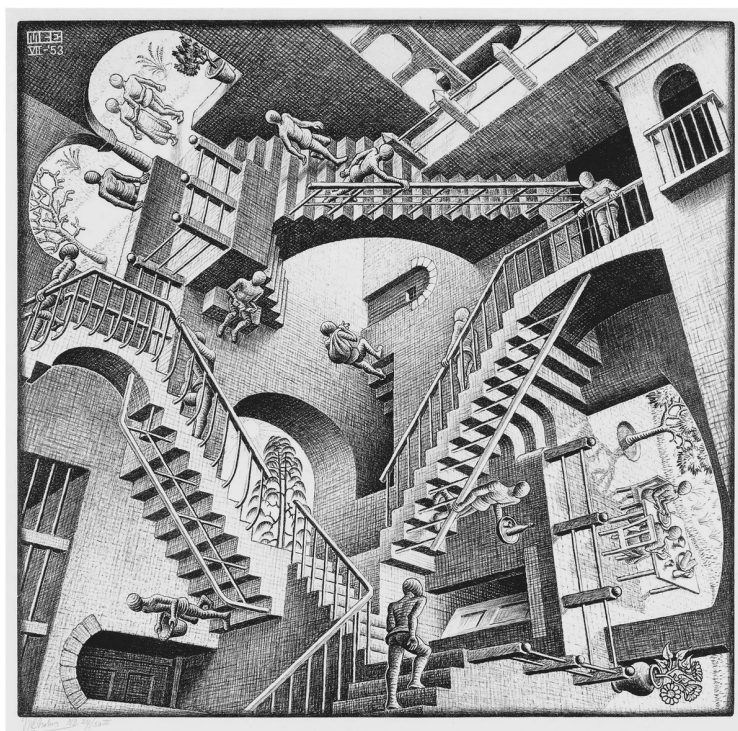
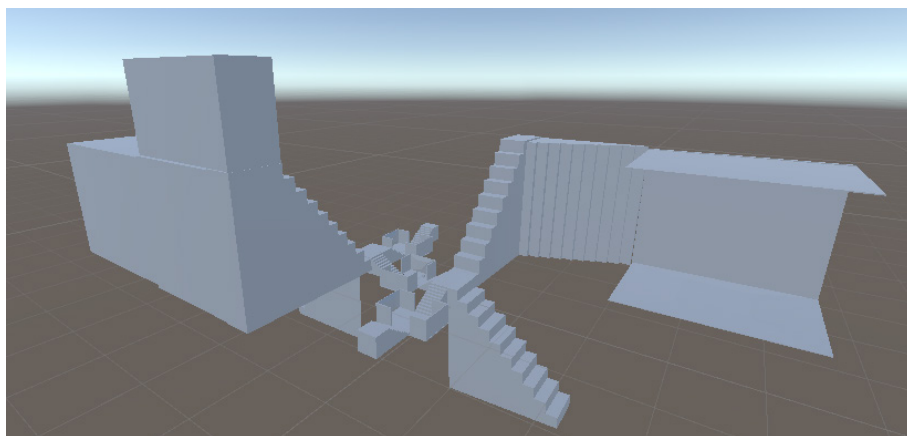


Рис. 1. Относительность



*Рис. 2. Внешний вид игровой сцены*

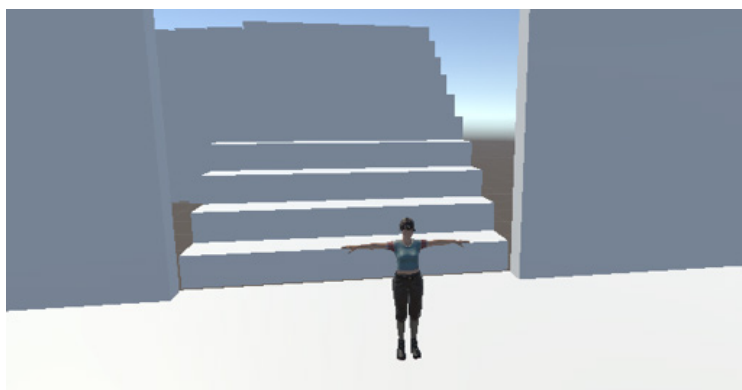
2. Создание персонажа. При помощи Fuse создается внешний вид персонажа. В данном случае женщина. Затем при помощи mixamo добавляем кости и анимации для выбранного аватара. Персонажу были добавлены анимации (ходьба, бег, прыжок).

Игровой персонаж представляет пользователя от первого лица (возможно переключение на вид от третьего лица).

Игрок управляет персонажем при помощи клавиатуры:

- *w*, *a*, *s*, *d* – вперед, влево, назад, вправо;
- *space* – прыжок;
- *e* – подбор предмета.

Управление направлением взгляда персонажа при помощи мыши.



*Рис. 3. Внешний вид персонажа в сцене*

3. Создание 3D игровых предметов (кубы, пирамиды, сферы).

Игрок входит в триггерную зону игрового предмета, после чего он может поднять его при помощи клавиши *E*, затем предмет можно повернуть вдоль одной из осей *x*, *y*, *z*. Игровые предметы имеют 6 степеней свободы.

В прототипе также планируется сделать подсказки. Одним из видов подсказок будут интерактивные изображения на стенах комнат, помогающие игроку определить, какой из игровых предметов он должен использовать, для выхода из комнаты.

### **Заключение**

В результате анализа существующих игр головоломок были выявлены следующие недостатки: отсутствие использования в играх 2 видов головоломок (пространственных и на поиск предметов), а также подсказок. На основании этого были разработаны требования к прототипу и была начата его разработка.

### **Библиографический список**

1. Бонд Джереми Гибсон. Unity и C#. Геймдев от идеи до реализации. СПб., Питер 2019. 928 с.
2. Относительность: литография. URL: [https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BD%D0%BE%D1%81%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C\\_\(%D0%BB%D0%B8%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F\)](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BD%D0%BE%D1%81%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C_(%D0%BB%D0%B8%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F)) (дата обращения: 26.05.2023).

УДК 004.738.5

**А. А. Андреев**

студент кафедры вычислительных систем и сетей

**Н. А. Балонин** – доктор технических наук, профессор – научный руководитель

### СОХРАНЕНИЕ НАУЧНЫХ САЙТОВ ОТ ИСЧЕЗНОВЕНИЯ

Сегодня в Интернете находится бесчисленные объемы разнообразной научной информации, от отдельных научных статей до полноценных сайтов разных ученых, загружающих таким образом результаты своих трудов в открытый доступ. Но существует проблема: многие сайты, платформы, не могут поддерживаться их создателями вечно, а, следовательно, сохранность этих ресурсов со всей содержащейся на них информацией гарантировать нельзя.

Как пишет журнал ScienceMag, 92 научных журнала по социальным и гуманитарным наукам и 84 по естественным пропали из Интернета за последние два десятилетия, поскольку издатели перестали их поддерживать. Потенциально это могло привести к утере полезной информации об исследованиях, проведенных публиковавшими в этих журналах авторами.

Под угрозой исчезновения находятся еще около 900 онлайн-журналов, говорится в совместном исследовании специалистов из университетов Берлина и Готтингена в Германии и Школы экономики Ханкена в Хельсинки. В его рамках были изучены научные публикации в журналах с 2009 по 2019 годы: оказалось, что их количество утроилось за этот период. При этом уже исчезнувшие журналы в среднем находились в сети в открытом доступе в течение 10 лет, по мнению авторов исследования, это означает, что из сети может пропасть намного больше ценных публикаций [1].

Такую информацию необходимо заведомо сохранять в более надежных хранилищах. Следует отметить, что это очень древняя практика. Людям приходилось переписывать и сохранять информацию еще со времен появления письменности. Для этого требовались сотни людей, владевших на тот момент редким навыком грамоты, сотни часов работы над каждым трудом и огромные помещения, в которых собирались, систематизировались и хранились сотни тысяч книг, за которыми все так же требуется периодический уход и поддержание благоприятных условий.

В эпоху цифровых технологий данный процесс значительно упростился: в цифровом виде информация занимает значительно меньше физического объема, чем в письменном, а цифровое копирование занимает совершенно незначительное время в сравнении с рукописным копированием. С другой стороны, и самой информации стало куда больше и, хотя человечеству стали доступны новые инструменты сохранения информации, все еще требуется приложить усилия для сохранения огромных объемов информации, находящейся в Интернете.

Одним из таких инструментов является создание **зеркального сайта** [2] – точной копии оригинального сайта, размещенной на другом сервере. В таком случае при потере доступа к оригинальному сайту его зеркальная копия в полном объеме (на момент копирования) будет доступна для использования.

Но стоит отметить, что при таком способе сохранения информации, зеркала сайтов все еще нужно обеспечивать серверами, которые так же требуют постоянной поддержки и обеспечения, как пришлось бы поддерживать хостинги оригинальных сайтов, что не только дорого, но и не дает долгосрочных гарантий, как и в случае с поддержанием сайта его создателем.

**Архивирование веб-сайтов** является значительно менее ресурсозатратным процессом, в результате которого копии данных загружаются с веб-сервера в виде файлов, которые впоследствии можно преобразовать в другие форматы. В целом это позволяет хранить огромные объемы данных в сжатом виде без предоставления постоянного хостинга для каждого элемента.

Предпосылками для развития веб-архивирования стали проблемы вымирания ссылок и потенциальное наступление цифрового темного века. В 1996 году был создан «Архив Интернета» – первая некоммерческая организация, поставившая перед собой цель создать «снимки» всех страниц в Интернете.

Так «Архив» запустил сервис по архивированию сайтов Wayback Machine – бесплатный онлайн-архив некоммерческой библиотеки «Архив Интернета». С помощью поисковых роботов Wayback Machine архивирует и делает общедоступной большую часть «открытого» Интернета. Сервис был запущен в 1996 году, однако стал доступен для общественности только в 2001. За первые 20 лет существования Wayback Machine каталогизировал и сохранил коллекцию из более чем 286 млрд веб-страниц. Благодаря сохраненным в Wayback Machine документам пользователи могут отслеживать происходящие на сайтах изменения и сравнивать разные версии правок. По состоянию на январь 2023 года Wayback Machine предоставляет доступ к более чем 767 млрд сохраненных веб-страниц [3].

Главными минусами такого способа является достаточно медленный доступ к сохраненной информации, а также сложности архивирования динамических сайтов, поскольку контент, скрипты и шаблоны хранятся на веб-сервере как отдельные ресурсы.

Но эти способы можно назвать в некотором роде «слепыми». В том плане, что они достаточно эффективны в сохранении огромных объемов информации, но ни о какой систематизации знаний здесь говорить не приходится. Некоторые публикации ученых могут быть так и не выделены для их сохранения, а многие затеряются уже после.

Если мы говорим именно об ученых и научных статьях, следует учитывать специфику их работы. Чтобы не приходилось копаться в бесконечных архивах Интернета, сохраняющих информацию хаотично, нужно собирать их работы еще при их публикации в одном месте. Если мы хотим, чтобы ученый воспользовался не своим сайтом, который впоследствии возможно придется спасать, а разместил свои работы на стороннем централизованном ресурсе то, во-первых, ресурс должен предоставлять все инструменты, необходимые ученому при публикации его исследований, а во-вторых он должен быть уверен, что этот ресурс продержится долго или, чтобы этот ресурс мог отделить его труд и превратить в самостоятельный сайт, который он может употребить по своему усмотрению. Разместить на своем более дешевом сервере. Или рассылать, как рассылают PDF-книги. Если задуматься, отделимость сайтов создает условия для их инъекции в окружающий нас мир с целью сохранения.

Для того, чтобы облегчить написание алгоритмических и мультимедийных книг нужно, чтобы некоторые наиболее продуктивные авторы могли делиться библиотеками программ. Эти алгоритмы можно брать прямо из их книг или из справочников, размещаемых в файлообменниках, сервисах наподобие GitHub, или на борту издательских систем, выпускающих сайты-книги.

Роль издательских систем в прошлом для авторов играли последовательно Лексикон, короткое время Чарайтер, наконец, Винфорд (потеснивший LaTeX) в сочетании с конверсией в отделимый формат (PDF). Времена изменились, и сильно. Современная «бумага» – это сайт, что закономерно ведет к появлению систем подготовки, учитывающих возросшие возможности.

Сервер для ученых исследует альтернативную Хранилищу технологию "книг нового поколения" (КНП). Математические HTML-JS книги легко скачиваются с серверов, и столь же легко возвращаются на них, ведь им нужно доменное имя. Государство, способное создать такую систему хранения ВЕБ-книг (сайтов), образовав национальную электронную библиотеку HTML-JS книг и учебников, сможет выиграть в своем развитии.

Математические HTML-JS книги превосходят PDF формат возможностью вести вычисления в алгебре матриц и теории конечных полей и групп, а также повышенными мультимедийными способностями, включая графики, диаграммы и 3D-моделирование. Для верстки таких книг полезная среда, включающая вставки книжного текста в формате HTML или FB2+ (гипотетического расширения, позволяющего вставлять он лайн алгоритмы) и пакеты готовых функций по разным областям математики (матричная алгебра, теория полей и групп, дифференциальные уравнения и прочее, прочее, все можно поместить в сеть) [4].

В итоге проблема сохранения информации находится в некотором зависшем состоянии: огромное количество научных статей, журналов и сайтов ученых требуют их сохранения, а технические возможности вполне позволяют реализовать это, но чтобы сохранить их требуются значительные ресурсы. Государственные и некоммерческие организации во многом помогают с сохранением трудов ученых, но наиболее действенной системой могло бы стать повсеместное распро-

странение и использование КНП, которые при достаточном финансировании и обеспечении ресурсами позволили бы ученым в универсальном формате эффективно сохранять и распространять свои научные работы.

#### **Библиографический список**

1. Десятки научных журналов исчезли из интернета за последние 20 лет, и никто их не сохранил. URL: <https://habr.com/ru/news/t/518594/> (дата обращения: 23.03.2023).
2. Что такое зеркальный сайт? URL: [https://bulgar-promo.ru/Zerkalniy\\_sayt\\_znachenije](https://bulgar-promo.ru/Zerkalniy_sayt_znachenije) (дата обращения: 23.03.2023).
3. Архивирование веб-сайтов. URL: [https://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D0%B2%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\_%D0%B2%D0%B5%D0%B1-%D1%81%D0%B0%D0%B9%D1%82%D0%BE%D0%B2](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D0%B2%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%B2%D0%B5%D0%B1-%D1%81%D0%B0%D0%B9%D1%82%D0%BE%D0%B2) (дата обращения: 23.03.2023).
4. Балонин Н. А. Перспективы хостинга для ученых в России. URL: <http://artspb.com/> (дата обращения: 23.03.2023).



УДК 007.65.01

**А. А. Антонов**

студент кафедры вычислительных систем и сетей

**Н. А. Балонин** – доктор технических наук, профессор – научный руководитель

### СОЗДАНИЕ СЕРВЕРА ПОД ХРАНЕНИЕ ДОСТИЖЕНИЙ РОССИЙСКОЙ НАУКИ

**Сервер** – выделенный или специализированный компьютер для выполнения сервисного программного обеспечения без непосредственного участия человека. Серверы размещаются в специально оборудованных помещениях, называемых дата-центром. Серверное оборудование зачастую предназначено для обеспечения работы сервисов в режиме 24/7, поэтому часто комплектуется дублирующими элементами, позволяющими обеспечить «пять девяток» (99,999%; время недоступности сервера или простой системы составляет менее 6 минут в год) [1].

Для чего нужен хостинг и домен? Домен и хостинг – это базовые услуги, без которых невозможно создать ни один сайт в Интернете. Чем отличается домен от хостинга? Домен – онлайн-адрес сайта, а хостинг для сайта – это пространство, на котором он «живет» [2].

Любой сайт – это по факту набор файлов (тексты, изображения, видео, программные скрипты, база данных и т. д.). Но для чего нужен хостинг для сайта? Чтобы пользователи Интернета вообще смогли увидеть его.

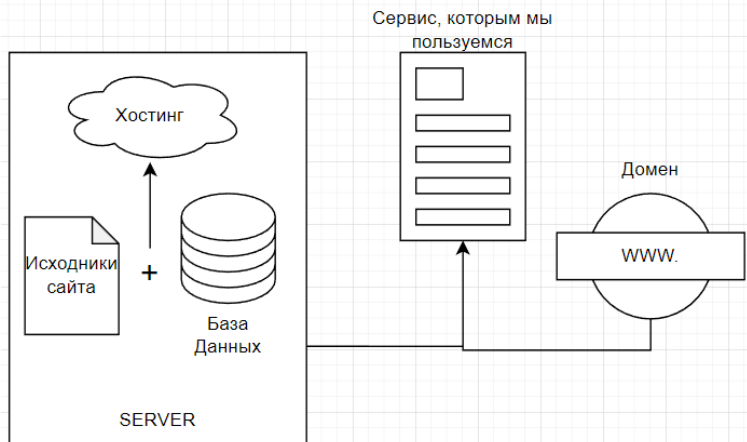


Рис. 1. Что проходит сайт, чтобы мы начали им пользоваться [1, 4]

«Чем вышесказанное важно?» – в современном мире нужно понимать, что кроется за ресурсами в сети Интернет, которыми мы пользуемся каждый день. Для того чтобы ресурсы были общедоступными, необходимо оплачивать этот процесс.

А теперь ответьте себе на вопрос «Чем закончился второй том «Мертвых душ»?». А ответа нет, как и второго тома. Николай Гоголь сжег его 24 февраля 1852 года [3]. Да, поговаривают, что черновые рукописи четырех глав второго тома были найдены у графа А. Толстого и даже по сей день находятся где-то в целости и сохранности. А видело ли это общество? Конечно, нет.

К чему все это? Вопросов мы задаем много. А в современном мире информационных технологий вопросы стоит решать современным способом. Теперь представьте, что Гоголь поделился своими трудами с обществом, буквально одним касанием, таким образом, что каждый в любое время может взять и изучить их. И никому не пришлось бы гадать, где эти труды, что они из себя представляют.

Знаете, почему Гоголь не мог этого сделать? Технологии не были развиты до такого уровня как сейчас.

Технологии всегда идут вперед, и к этому причастны все. Но есть люди, которых называют учеными, они являются основной силой прогресса. Это значит, что их труды будут полезны всем. Многие это понимают и создают свои сайты, на которых размещают полезные материалы из разных областей науки. Математики, физики, химики, биологи, все они делятся своими знаниями на созданных ими сайтах. За открытость сайта приходится платить, учитывая что владельцами сайтов являются сами ученые, из их кармана и оплачивается сайт.

Все мы не вечные и когда-нибудь уйдем из этого мира. Но что будет с сайтом, когда платить будет некому? Что сделать, чтобы не произошло ситуации как с Николаем Гоголем? Гениальным решением является создание Единого портала трудов отечественных ученых.

Представьте, вы – человек, обладающий какой-либо научной степенью, – получаете от государства право размещать свои труды на бесплатной основе. При этом вы можете обсуждать достижения науки с другими такими же учеными, как и вы. Плоды ваших умов отсортированы так, что они ни за что не перемешаются с другими материалами.

Или наоборот, вы – молодой студент или начинающий ученый, жаждущий изучения вершин науки. Вы можете наблюдать за обсуждениями ученых. Изучать материалы больших и малых сроков давности. Применять теорию и результаты практики в личных целях, на коммерческой и некоммерческой основах.

Все это будет двигать прогресс вперед. Учитывая ситуацию в мире, и конкуренцию ведущих держав, было бы целесообразно ограничить такой ресурс границей нашей родины. Это не только создаст огромную базу данных, но и обеспечит им сохранность.

Создание русскоязычного ресурса повысит уровень образования, ведь изучать материалы станет проще. Еще проще будет, когда данные будут отсортированы.

Вечно обеспечивать сохранность не получится – логично будет хранить информацию определенное время, пока данные актуальны, так как прогресс не стоит на месте. На место устаревших данных должны приходиться актуальные.

*Статья написана при поддержке доктора технических наук, профессора Николая Алексеевича Балонина в рамках организационных вопросов науки и культуры.*

### Библиографический список

1. Проблема сохранения сайтов. URL: <http://artspb.com/> (дата обращения: 20.04.2023).
2. Интернет-курсы Санкт-Петербургского государственного университета аэрокосмического приборостроения. URL: <http://livelab.spb.ru/labs/index.php> (дата обращения: 20.04.2023).
3. Мертвые души. Википедия. URL: [https://ru.wikipedia.org/wiki/Мертвые\\_души#:~:text=В%20феврале%201852%20г.%2C%20чувствуя,писателя%2C%20опечатанных%20после%20его%20смерти](https://ru.wikipedia.org/wiki/Мертвые_души#:~:text=В%20феврале%201852%20г.%2C%20чувствуя,писателя%2C%20опечатанных%20после%20его%20смерти) (дата обращения: 20.04.2023).
4. Что такое хостинг простыми словами. URL: <https://help.reg.ru/support/hosting/zakaz-hosting-arbota-s-uslugoy/chto-takoye-hosting-prostyimi-slovami> (дата обращения: 20.04.2023).

УДК 004.946

**Е. В. Еременок**

студентка кафедры вычислительных систем и сетей

**А. В. Никитин** – кандидат технических наук, доцент – научный руководитель

## ПРИЛОЖЕНИЯ ДЛЯ ТУРИЗМА И ПУТЕШЕСТВИЙ НА ОСНОВЕ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

Виртуальная реальность (VR) стала перспективным инструментом во многих отраслях, включая туризм и путешествия. VR-приложения для туризма могут обеспечивать пользователей возможностью исследовать достопримечательности, города и курорты, не покидая своего дома. В данной статье мы рассмотрим создание VR-приложений для туризма и путешествий на языке программирования Swift.

### I. Swift и виртуальная реальность в туризме: актуальность и перспективы

#### *Потребность в инновациях в туризме*

С развитием технологий и изменением потребительских предпочтений туристическая индустрия нуждается в новых подходах и инструментах для привлечения клиентов. VR-приложения, разработанные на Swift, могут предложить пользователям неповторимый опыт виртуальных путешествий и помочь туристическим компаниям привлечь новых клиентов [1].

#### *Недавние исследования и разработки*

Исследователи и разработчики активно изучают возможности использования Swift в области VR для туризма. В рамках таких исследований осуществляется работа над созданием библиотек и инструментов, которые упрощают разработку VR-приложений для туризма и путешествий.

### II. Особенности создания VR-приложений для туризма на Swift

#### *Реалистичная визуализация и анимация*

Одним из ключевых аспектов создания VR-приложений для туризма является реалистичная визуализация и анимация объектов и сцен. Swift в сочетании с фреймворками, такими как SceneKit и ARKit, позволяет разработчикам создавать высококачественную графику и анимацию, которая погружает пользователя в виртуальное пространство [2, 3].

#### *Интерактивность и пользовательский опыт*

Для того чтобы VR-приложение для туризма было успешным, оно должно быть интерактивным и обеспечивать пользователю комфортный и интересный опыт. Swift позволяет разработчикам создавать интуитивные и адаптивные пользовательские интерфейсы, а также реализовывать интерактивные элементы, такие как жесты, виртуальные туры и аудиогиды.

#### *Геолокация и карты*

Интеграция с геолокационными сервисами и картами является важным аспектом для создания VR-приложений в туризме. Swift позволяет легко использовать сервисы геолокации и карты Apple для предоставления пользователю точной информации о местоположении объектов и планирования маршрутов в виртуальных путешествиях.

#### *Интеграция с платформами бронирования и социальными сетями*

Для привлечения большего количества пользователей и увеличения продаж туристических услуг, VR-приложения могут быть интегрированы с платформами бронирования и социальными сетями. Swift облегчает интеграцию с такими платформами, позволяя пользователям делиться своим опытом виртуальных путешествий и осуществлять бронирование прямо из приложения.

Swift представляет собой мощный и перспективный инструмент для создания VR-приложений в области туризма и путешествий. Благодаря своей производительности, гибкости и возможностям интеграции с различными сервисами и фреймворками, разработчики могут создавать инновационные и функциональные приложения, которые обеспечивают пользователям неповторимый опыт виртуальных путешествий и способствуют развитию туристической индустрии.

### III. Swift и виртуальная реальность в туризме: цифры и факты

#### *Рынок VR-туризма*

Согласно исследованию MarketsandMarkets, глобальный рынок VR-туризма оценивается в \$1,56 млрд в 2021 году и ожидается, что он вырастет до \$7,64 млрд к 2026 году, со среднегодовым темпом роста в 38,7%. Разработка VR-приложений на Swift может помочь туристическим компаниям занять свою долю на этом быстрорастущем рынке.

#### *Популярность Swift среди разработчиков*

По данным Stack Overflow Developer Survey 2021, Swift занимает 14 место среди наиболее популярных языков программирования, используемых разработчиками по всему миру. Благодаря своей простоте, скорости и гибкости, Swift становится все более популярным выбором для создания VR-приложений в туризме.

### IV. Особенности создания VR-приложений для туризма на Swift: количественные данные

#### *Реалистичная визуализация и анимация*

С использованием SceneKit и ARKit, разработчики Swift могут достичь впечатляющего уровня детализации при создании VR-приложений для туризма. Например, VR-приложение может воспроизводить 3D-модели объектов с точностью до 1 см и обеспечивать анимацию с частотой кадров до 90 кадров в секунду для обеспечения плавности и реалистичности.

#### *Время разработки и стоимость проектов*

С использованием Swift, разработчики могут сократить время разработки VR-приложений для туризма на 20–30% по сравнению с традиционными языками программирования благодаря высокоуровневым абстракциям, упрощенному синтаксису и обширному набору библиотек. Это также может привести к снижению затрат на разработку на 15–25%, делая инвестиции в VR-приложения для туризма более доступными для компаний различного масштаба [4 – 7].

#### *Уровень пользовательского удовлетворения*

Оценка пользовательского удовлетворения (User Satisfaction Score) для VR-приложений туризма, разработанных на Swift, может достигать 85–95% благодаря высокой производительности, интерактивности и стабильности приложений. Это свидетельствует о высоком качестве пользовательского опыта и способствует увеличению лояльности клиентов к туристическим компаниям, использующим VR-технологии.

Разработчики Swift продолжают изучать новые подходы и возможности в области виртуальной реальности для туризма, стремясь создать приложения, которые будут востребованы на рынке и обеспечат высокое качество пользовательского опыта.

### Библиографический список

1. The Swift Programming Language / Apple Inc. URL: [swift.org](https://swift.org) (дата обращения: 04.03.2023).
2. SceneKit Documentation / Apple Developer. URL: [developer.apple.com](https://developer.apple.com/documentation/scenekit) (дата обращения: 04.03.2023).
3. ARKit Documentation / Apple Developer. URL: [developer.apple.com](https://developer.apple.com/documentation/arkit). (дата обращения: 04.03.2023).
4. Tussyadih I., Zach F. J. Virtual reality for tourism destinations: A perspective article on prospects and challenges // Journal of Travel Research. 2017. Т. 56. № 7. С. 809–819.
5. Kim D. J. et al. The role of presence in the experience of virtual reality tourism: A study of tourist destinations in the Mediterranean // Journal of Travel Research. 2018. Т. 57. № 8. С. 1061–1079.
6. Jung T. H., Lee J. S. Virtual reality, presence, and attitude change: Empirical study on VR tourism marketing // Journal of Travel Research. 2020. Т. 59, № 1. С. 15–30.
7. Virtual Reality (VR) in Tourism – Thematic Research // GlobalData. <https://www.globaldata.com/store/report/vr-in-travel-and-tourism-theme-analysis/> (дата обращения: 24.03.2023).

УДК 004.921

**Д. А. Кибардин**

студент кафедры вычислительных систем и сетей

**И. Н. Соя**

студент кафедры вычислительных систем и сетей

**А. В. Никитин** – кандидат технических наук, доцент – научный руководитель

## ВЛИЯНИЕ ВЗАИМОДЕЙСТВИЯ ИГРОВЫХ МЕХАНИК НА ОПЫТ ИГРОКА

### Введение

Сегодня игры играют огромную роль в жизни людей. Игроки могут погружаться в виртуальные миры и переживать различные приключения, испытывать разнообразные эмоции и наслаждаться игровым процессом во время досуга. В свою очередь, разработчики игр продолжают совершенствовать технологии и игровые механики, делая игры все более реалистичными и увлекательными.

### Опыт игрока

Опыт пользователя (user experience) – восприятие и ответные действия пользователя, возникающие в результате использования и/или предстоящего использования продукции, системы или услуги.

То, что одни игроки находят привлекательным, совсем не интересует других, а разные игроки могут получать удовольствие от одной и той же игры по разным причинам [1]. Такого рода впечатления всегда соотносятся с конкретными игроками и их мотивацией. Чтобы понять, что делает игры привлекательными, нужно для начала исследовать игроков и их мотивы (рис. 1).



Рис. 1. Диаграмма типов мотивации игроков

Знание мотивации игрока помогает понять, как он будет реагировать на разрабатываемый игровой опыт, благодаря чему можно постараться предвосхитить его ожидания [1]. Мотивация делится категории, но при этом мотивы игрока индивидуальны, и у любого могут наблюдаться различные сочетания.

Кластеры показывают (рис. 2), что в мотивации игроков могут наблюдаться определенные закономерности. Например, близость стратегии и вызова (или сюжета и дизайна) предполагает, что игроков, которым нравятся оба этих аспекта, будет больше, чем тех, которым, нравятся одновременно вызов и дизайн. А открытие и влияние находятся примерно посередине между кластерами и служат «связующими звеньями», привлекательными для широкой аудитории.



### 3 кластера

1. Экшен-социализация.
2. Мастерство-достижения.
3. Погружение-творчество.

### Связующие звенья

- Открытие — связующее звено между погружением-творчеством и мастерством-достижениями.
- Влияние — связующее звено между экшен-социализацией и мастерством-достижениями.

Рис. 2. Диаграмма расстояний между типами мотивации игроков

### Игровые механики

Игровые механики – это правила и системы, которые определяют поведение игрока и взаимодействие с игровым миром. Они могут создавать разные виды опыта для игрока.

Например, механика выбора может создавать опыт выбора, когда игрок должен принимать решения, которые влияют на развитие игры. Это может быть важно для игроков, которые желают иметь больше контроля над игровым миром.

Механика боя может предоставлять опыт экстрима и возбуждения, если бой динамичен и заполнен действием. В то же время, если бой ужасно балансируется и может привести к повышенным уровням стресса, то она может создавать чувство разочарования.

Механика прогресса может поддерживать чувство удовлетворения от того, что игрок достиг какой-то меры успеха в игре. Эта механика обычно включает систему рангов, уровней, опыта, которые позволяют игроку отслеживать их прогресс и прогрессировать дальше.

Кроме того, игровые механики могут создавать опыт коллективной игры, если игроки работают вместе над достижением общих целей, а не играют отдельно.

Это лишь некоторые примеры того, как игровые механики могут влиять на опыт игрока. В конечном итоге, качество игры зависит от ее особенностей и того, что ищет игрок в игровом процессе.

### Метрики опыта игрока

На опыте игроков можно принять решение о создании игровых механик:

- создание предметов снаряжения (крафт);
- боевая система с элементарными взаимодействиями;
- открытый мир.

Для установления зависимостей между игровыми механиками и опытом игрока вводятся метрики:

- число побежденных врагов;
- число созданного снаряжения;
- число побежденных врагов, с помощью созданного снаряжения.

Требования к прототипу:

1. Платформа: прототип должен быть разработан для операционных систем Windows.
2. Графика: прототип должен иметь 3D графику с высокой детализацией и реалистичным световым оформлением.

3. Боевая система: в прототипе должна быть боевая система, которая позволяет игрокам использовать различные типы оружия, заклинаний и умений в бою.

4. Создание предметов (крафт): система создания предметов должна быть простой в использовании и понимании. Игроки должны иметь возможность создавать новые предметы, используя ресурсы, которые они находят в игре.

5. Открытый мир: прототип должен иметь большой открытый мир, полный разнообразных локаций и событий. Мир должен быть интересным для исследования и полным секретов и загадок.

6. Управление персонажем: игроки должны иметь возможность управлять своим персонажем, его характеристиками. Персонажи должны иметь уникальные способности и навыки, которые можно развивать в процессе игры.

7. Оптимизация: прототип должен быть хорошо оптимизирован и работать на средних и высоких конфигурациях компьютеров без проблем с производительностью.

### Процесс разработки

Процесс разработки в общем виде для проверки взаимодействия игровых механик и опыта игрока можно подразделить на следующие этапы (рис. 3). В прототипе планируется к разработке две глобальные механики: боевая и создание предметов (крафт). Когда боевая система и система крафта определены, следует первоначальная реализация каждой из них. Затем разработчики должны протестировать каждую систему отдельно, чтобы убедиться, что она работает в соответствии с замыслом.



Рис. 3. Процесс разработки

Далее следует рассмотреть взаимодействие между этими двумя системами. В зависимости от конечных целей проекта, взаимодействие между боевой системой и системой крафта может быть достаточно простым, например, использование созданных предметов в бою. Однако, для повышения глубины игрового процесса желательно синергия между этими двумя системами.

Например, игрок может создавать новое оружие или броню, которые дадут ему преимущество в бою. Или игрок может использовать ресурсы из своего инвентаря для создания ловушек и других предметов, которые помогут ему преодолеть сложные участки игры. Важно помнить, что взаимодействие между этими двумя системами должно быть понятным и доступным для игрока. Если игрок не понимает, как эти системы взаимодействуют, игровой процесс может стать неприятным и утомительным.

### **Заключение**

Взаимодействие игровых механик имеет огромное влияние на опыт игрока. Игровые механики— это механизмы, которые управляют тем, как игрок взаимодействует с игрой и как игра взаимодействует с игроком.

Недостаточно только создать хорошие игровые механики, также необходимо, чтобы они были связаны друг с другом и образовывали единое целое. Взаимодействие между игровыми механиками может быть сложным и не всегда очевидным для игрока, но если оно выполнено правильно, то игрок может получить удовольствие от процесса разгадывания этих механик.

Кроме того, знание игровых механик может дать игроку преимущество в игре. Если игрок понимает, как работают механики, то он может лучше использовать их в своих интересах и достичь больших результатов.

### **Библиографический список**

1. Роберт Зубек. Элементы гейм-дизайна. Как создавать игры, от которых невозможно оторваться / Пер. с англ. О. И. Перфильева. М.: Эксмо, 2022. 272 с. (Мировой компьютерный бестселлер. Гейм-дизайн). ISBN 978-5-04-1233200-9.



УДК 004.946

**М. В. Климов**

студент кафедры вычислительных систем и сетей

**А. В. Никитин** – кандидат технических наук, доцент – научный руководитель

## VR-СИМУЛЯТОР ПОВЕДЕНИЯ ЧЕЛОВЕКА ПРИ ПОЖАРЕ

### Введение

В современном мире существует множество приложений для отработки процессов поведения при пожаре. Однако, большинство подобных по тематике приложений разработано со следующим (или похожим) сценарием: Пользователь появляется в комнате, в которой рядом с ним что-то горит, и ему необходимо нажать на кнопку взять огнетушитель, который лежит рядом с ним, после чего зажатием кнопки и направлением контроллера в нужную сторону, избавиться от возгорания. Данный сценарий не позволяет пользователю погрузиться в ситуацию и запомнить, какие действия необходимо делать своими руками, чтобы потушить возгорание вручную или, тем более эвакуироваться из пожара при помощи различного пожарного инвентаря. Также стоит уточнить, что все помещения, генерируемые в данных сервисах весьма ситуативные и не позволяют также легко сориентироваться в других.

Недостатком существующих симуляторов, созданных для обучения и отработки действий при пожаре является то, что они нацелены лишь на использование огнетушителя в небольшом ограниченном пространстве, либо, в некоторых случаях, на выход из здания, но лишь при помощи примитивного использования кнопок манипулирования. Обучение в таком стиле не помогает людям понять, что им нужно делать при необходимости эвакуации из здания в случае пожара или подобной чрезвычайной ситуации, с использованием плана эвакуации и имеющегося оборудования.

### Разбор аналогов

Для выделения основных преимуществ и недостатков были взяты наиболее приближенные к проблеме аналоги (табл. 1).

Таблица 1

### Недостатки и преимущества аналогов

Аналоги	VR Fire Extinguishing Experience Simulator [1]	VR-сценарий действий при пожаре в школе [2]	Интерактивная визуализация инструкций действий человека при пожаре с использованием дополненной реальности [3]	Fully immersive VR learning solutions for training in hazardous and emergency situations [4]
Особенности				
Реалистичность взаимодействий*	+	–	–	+
Погружение в виртуальную реальность**	+	+	–	+
Различный инвентарь, который потребуются в случае пожара***	–	+	+	–
На сцене находится полное здание, а не только одна комната	–	+	–	–

\*Пользователь взаимодействует с окружением при помощи правильных движений руе, а не примитивного нажатия кнопок манипуляторов.

\*\*Полное аудиовизуальное погружение.

\*\*\*Огнетушитель, план эвакуации, блок самоспаса, пожарный конус и так далее.

При рассмотрении аналогов были выделены следующие основные недостатки:

1. Инвентарь для эвакуации и пожаротушения сильно ограничен и зачастую заключается лишь в огнетушителе.
2. Взаимодействия с объектами чаще всего сводятся к простым нажатиям на кнопки контроллера, т. е. без прямого взаимодействия руками с объектом.
3. План здания почти всегда ограничен лишь одной комнатой, что никак не поможет в общем случае, поскольку нет перемещения по коридорам и лестницам.
4. А также все необходимые объекты всегда лежат либо рядом с игроком, либо рядом с источником огня, а не там, где им следует лежать.

### Предлагаемое решение

Предлагаемое решение данной проблемы заключается в том, чтобы разработать VR- приложение с возможностью совместной работы (использование несколькими пользователями одновременно), в которое необходимо поместить здание общежития и весь имеющийся там пожарный инвентарь (огнетушитель, план эвакуации, блок самоспаса, пожарный конус и так далее). Пользователю должна быть дана возможность аудиовизуального погружения и взаимодействовать со своим окружением при помощи киберперчаток.

При решении проблемы данным образом, преимущества будут заключаться в эргономическая адекватность должна быть выше, чем навыки, получаемые навыки должны быть приближены к тем, которые можно получить при попадании в реальную ситуацию, за счет аудиовизуального погружения в виртуальную реальность и взаимодействия с ним буквально «своими руками». Взаимодействие будет возможно с различным пожарным инвентарем, а не только с огнетушителем. Также сцена будет заключаться не только в одной небольшой комнате, а в целом многоэтажном здании.

Стоит уточнить, что использование совместной работы поможет, при отработке одним пользователем действий, другому наблюдать и направлять эти самые действия.

### Разработка требований

На данный момент в этой работе выделена проблема, проведено сравнение аналогов, выделены их недостатки. Также была подготовлена среда разработки, составлены концепт- документ и дизайн-документ [5] и разработаны требования к прототипу:

- Использование на ПК (ОС Windows) с клавиатурой/мышью.
- Сцена должна содержать здание (2 этажа), в котором будут комнаты, коридоры, двери (анимированные, открываются по триггеру), окна и огнетушитель, который можно взять и поставить на пол.
- На сцене должен быть персонаж с аудиовизуальным погружением на сцену с управлением от 1 лица, с 3 степенями свободы перемещения/вращения.

Данную задачу можно формализовать следующим образом: определяются кортежи из объектов и их свойств, на основе которых разрабатывается граф состояний системы в зависимости от изменений.

Все множество объектов поведения человеком при пожаре можно описать как кортеж <Помещение, Двери, Аватар, Огонь, Дым, Огнетушитель> . .

Для каждого элемента кортежа можно ввести состояния:

- Помещение. Находится или не находится там аватар. Опасное или не опасное помещение.
- Двери. Открыты или закрыты.
- Аватар. Двигается/взаимодействует или бездействует.
- Огонь. Усиливается или ослабевает.
- Дым. Есть или нет.
- Огнетушитель. Активен или неактивен.

Взаимосвязи объектов: дым связан с огнем; огонь и двери связаны с помещением; огнетушитель связан с аватаром.

Взаимосвязи состояний. Огнетушитель неактивен, если аватар с ним не взаимодействует. Огонь усиливается если открыты окна и двери в помещении и ослабевает, если огнетушитель активен

и направлен на него. Дым есть только если огонь уже усилился; его нет, если двери открыты и огонь не усилятся.

В настоящей статье проведен анализ и разработаны требования к прототипу, который будет получен позже.

Польза данного решения заключается в том, что данное приложение позволит обучать студентов из общежития необходимым действиям во время эвакуации. Также, благодаря этому приложению, пользователи, не живущие в данном общежитии, смогут понять, как пользоваться каждым из добавленных видов пожарного инвентаря, которые добавлены в приложение, включая пользование планом эвакуации.

#### **Библиографический список**

1. KVRBE – Новости виртуальной и дополненной реальности. URL: [https://vrbe.ru/vr\\_news/1178-kak-vesti-sebya-vo-vremya-pozhara-rasskazhet-virtualnaya-realnost.html?ysclid=laiaausjtj319204081](https://vrbe.ru/vr_news/1178-kak-vesti-sebya-vo-vremya-pozhara-rasskazhet-virtualnaya-realnost.html?ysclid=laiaausjtj319204081) (дата обращения: 15.10.2022).
2. Комплекс для проведения инструктажей и подготовки персонала. URL: <http://nntpb.ru/%d0%ba%d0%be%d0%bc%d0%bf%d0%bb%d0%b5%d0%ba%d1%81-2/> (дата обращения: 15.10.2022).
3. Никитин А. В., Потехин Д. С. Интерактивная визуализация инструкции действий человека при пожаре в общежитии с использованием дополненной реальности // Семьдесят вторая международная студенческая научная конференция ГУАП: сб. докладов. СПб ГУАП, 2019.
4. Safer. Smarter. Deeper learning. URL: <https://flaimesystems.com/> (дата обращения: 15.05.2023).
5. Научно-Исследовательская работа. URL: <https://pro.guap.ru/inside/student/reports/3427066> (дата обращения: 26.05.2023).

УДК 004.946

**М. А. Мурашко**

студентка кафедры вычислительных систем и сетей

**А. В. Никитин** – кандидат технических наук, доцент – научный руководитель

## МЕХАНИКИ СТРАХА В ИГРАХ ТИПА ХОРРОР

### Введение

Игры выходят ежегодно, во всех существующих жанрах. Так, например, игры жанра хоррор развились и теперь обладают поджанрами. Рассмотрим характерные черты каждого из них:

– Survival horror: ограниченные ресурсы, бои, потеря контроля над происходящим, помехи в виде камеры с неудобного вида, антураж.

– Хоррор-шутеры: больше ресурсов, удобное расположение камеры, много вражеских юнитов.

– Хоррор-аркады: неясный сюжет, вид сбоку, обилие головоломок и доходчивая демонстрация смерти персонажа [10].

Интерактивное кино: блуждание по локациям и выполнение динамичных сцен в игре, в которой игроку предлагается нажать комбинацию клавиш в определенном порядке или с определенной частотой, чтобы завершить ее, выбор реплик.

За последние несколько лет хотелось бы отметить компанию Supermassive Games выпускающую серию игр «Dark Pictures» с 2019 года, последняя часть вышла 2022 году. Игры в жанре интерактивного кино с элементами survival horror быстро завоевали популярность публики и каждый новый релиз ожидается с нетерпением. Ранее компания выпускала игры лишь на консоли, но вскоре расширила платформы, завоевав еще большую аудиторию.

Для устрашения она активно использует различные резкие звуки и анимации (скримеры), сами локации – опустевшие, мрачные – не дают игроку расслабиться. Запутанная история, которая порой может свести с толку, красочные пугающие сцены гибели персонажей, все это помогает сохранять напряжение от начала и до конца.

Игры жанра платформер сильно поменялись с первого их появления. Такому жанру характерно передвижение персонажа влево/вправо, прыжок и атака. Так в самой первой игре ««Space Panic» герой мог перемещаться по платформам только с помощью лестниц. В следующей за ней игре «Donkey Kong» появился прыжок, но нужен он был исключительно для преодоления препятствий в виде бочек на пути главного героя» [5].

Недавно вышедшая (в 2021 г.) «It takes two» покорила всех – это платформер, приключенческий боевик повествует сюжет, который многим будет близок (потеря близости и понимания в отношениях). Яркая картинка, красивые локации, передвижения по локациям и возможность забираться на некоторые предметы – это все создает интересный геймплей.

Почти ежегодно игры жанров хоррор и платформер завоевывают интерес публики. Разработчики научились удивлять и совершенствовать графику.

Ежегодно количество игр в том или ином жанре растет равномерно. Собрав статистику с сайта SteamSpy (рис. 1, 2) мы можем заметить, что среди таких жанров как: хоррор; платформер; экшн; стратегия; ритм-игры; платформер и хоррор примерно на одном уровне по количеству выпускаемых компьютерных игр на платформе Steam за последние 5 лет.

Ведущими поджанрами хоррора на платформе Steam являются экшн и выживание.

Причины:

1. Адреналин – быстрый способ получения сильных эмоций (страх, тревожность, напряжение) во время игры.

2. Развлечение – занимает досуг.

3. Личный интерес – хоррор играм присущи темные/запретные темы (смерть, монстры, божество).

Исходя из выше написанного можно сделать вывод о том, что жанр хоррор игр активно развивается и имеет много поклонников. На вопрос, почему люди любят хорроры, уже давно ответила

наука – потому что эффект от нервного напряжения во время игры или фильма и последующего расслабления сравним по количеству выработанных эндорфинов со смехом. Иными словами, если у вас нет каких-то предубеждений или особых медицинских обстоятельств, пугаться очень и очень весело [9].

**Цель:** изучение особенностей механики страха в играх типа хоррор.

**Основная часть**

*Методы и инструменты исследования*

Как опознать страх? Чтобы понять страшно человеку или нет самое первое что необходимо сделать, это спросить напрямую – метод анкетирования.



Рис. 1. Статистика по выпущенным компьютерным играм с платформы Steam за последние 5 лет

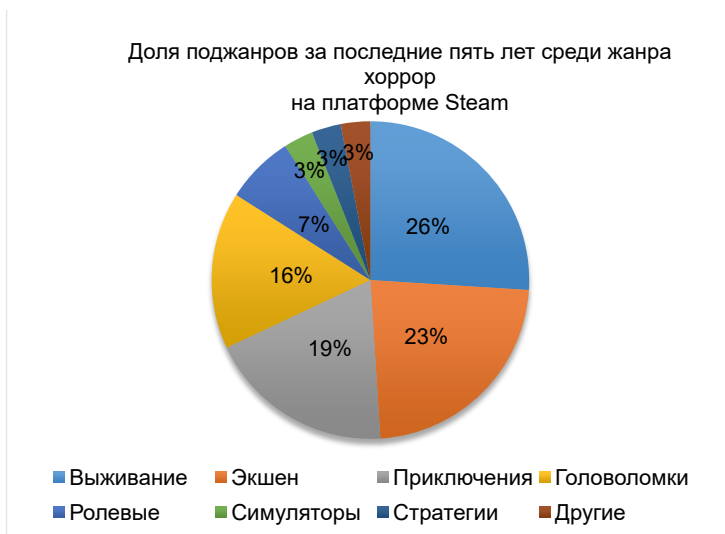


Рис. 2. Доля поджанров хоррора за последние пять лет на платформе Steam

Анкетирование – психологический вербально-коммуникативный метод, в котором в качестве средства для сбора сведений от респондента используется специально оформленный список вопросов – анкета [13]. Вопросы должны четко дать понять: личные предпочтения игрока, какие эмоции он испытал, что ему (не)понравилось.

Пример вопросов: какие эмоции вы испытывали во время прохождения игры, какие элементы вызвали у вас особенные эмоции, продолжались ли неприятные ощущения после прекращения игрового процесса продолжительное время, какой уровень ужаса вы предпочитаете.

Таблица 1

**Аналоги**

Cuphead	Hollow Knight	Little Nightmares 2	Rayman Legends	Заявляемый объект
Платформер	+	+	+	+
Уровни сложности	+	–	+	–
Атака	+	+	+	+
Инвентарь	+	+	+	+
	Хоррор	+	-	+
		Кооператив	+	–
			Ритм-уровни	+
				Разные концовки
				Выбор фраз в диалоге

Для реализации прототипа были рассмотрены и выбраны данные инструменты:

Unity 3D – кроссплатформенная среда разработки компьютерных игр.

Blender 3D – приложение для создание трехмерной графики.

Visual Studio – среда разработки программного обеспечения на языке C#.

*Разбор аналогов*

Среди наиболее близких по механике игр были выделены следующие:

Cuphead – сложная игра жанра платформер, стреляй и беги, стрельба из разного оружия, которое можно менять, выбор сложности, возможность играть с другом.

Little Nightmares – нагнетающая жуткая атмосфера, хоррор-выживание.

Rayman Legends – музыкальные уровни, проверяющие игрока на чувство ритма.

Inside – как бы намекающее повествование, напряженная атмосфера.

**Предлагаемое решение**

Звук – громкие, резкие звуки, звуки с определенной частотой. Накатывающиеся звуки часто имеют усеченные и неравномерные спектры, что приводит к возбуждению строго определенных групп нейронов, а не всех в определенной степени. Для таких шумов спектр быстро меняется во времени и в каждый момент времени остается усеченным, но в новом частотном диапазоне. Это заставляет мозг бить тревогу [12].

Графически – темное время суток, ограничение обзора, темные цвета, сочетать несочетаемое, устрашающие образы.

Искусственный интеллект – резкое появление предметов, запуск диалогов.

Геймплей – строгое ограничение ресурсов, прятаться, атаковать, возможность биться с «боссом» лишь 1 раз.

Для реализации прототипа выделены следующие этапы:

1. Создание ландшафта.
2. Создание персонажа.

3. Создание 3D игровых предметов.
4. Скриптинг управления и анимации персонажа.
5. Создание игрового меню.

На данный момент реализовано:

- Дискон.
- Концепт-документ.

Для реализации различных механик хоррор игр необходимо тщательно изучить психологическую составляющую эмоционального воздействия на игроков. Исследования в этой области помогают разработчикам лучше понять, какие элементы графики и дизайна вызывают самые сильные эмоции, такие как страх и тревога, и почему.

### Библиографический список

1. Blender 3D. URL: <https://media.contented.ru/glossary/blender-3d/> (дата обращения: 22.04.2023).
2. Cuphead Review. URL: <https://www.ign.com/articles/2017/10/02/cuphead-review> (дата обращения: 22.04.2023).
3. Hollow Knight Review. URL: <https://www.ign.com/articles/2018/06/22/hollow-knight-review> (дата обращения: 22.04.2023).
4. Rayman Legends. URL: [https://www.igromania.ru/article/23707/Rayman\\_Legends.html](https://www.igromania.ru/article/23707/Rayman_Legends.html) (дата обращения: 22.04.2023).
5. VK. Базовые механики в платформере. URL: <https://vk.com/@justplayittoo-bazovye-mehaniki-v-platformere> (дата обращения: 22.03.2023).
6. What is Unity 3D? URL: <https://www.juegostudio.com/blog/what-is-unity-3d-a-comprehensive-guide-to-unitys-features-and-uses#:~:text=Unity%20is%20a%20powerful,iOS%2C%20Android%2C%20and%20more> (дата обращения: 22.04.2023).
7. What is Visual Studio? URL: <https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022#:~:text=Visual%20Studio%20is%20a%20powerful,and%20then%20deploy%20your%20app> (дата обращения: 22.04.2023).
8. Базовые эмоции: что такое страх. URL: <https://theoryandpractice.ru/posts/18773-bazovye-emotsii-chto-takoe-strakh> (дата обращения: 22.04.2023).
9. Дискуссия: почему мы играем в хорроры. URL: <https://df.ru/flood/9149-diskussiya-pochemu-my-igraem-v-horroru> (дата обращения: 22.04.2023).
10. Жанр хоррора в играх – страшнее, чем в кино. URL: <https://club.dns-shop.ru/blog/t-64-videoigr/y/58011-janr-horror-a-igrah-strashnee-chem-v-kino/> (дата обращения: 22.03.2023).
11. Индустрия страха: почему зрителей привлекает то, что должно пугать? URL: <https://ki-news.ru/news/industrialiya-strakha-pochemu-zritelei-privlekaet-to-chto-dolzno-pugat/> (дата обращения: 22.04.2023).
12. Какие звуки вызывают у нас тревогу. URL: <https://www.psychologies.ru/articles/kakie-zvuki-vyzyvayut-chuvstvo-trevogi-i-kak-ot-nikh-mozhno-zashititsya/> (дата обращения: 22.04.2023).
13. Метод анкетирования. URL: [https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4\\_%D0%B0%D0%BD%D0%BA%D0%B5%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%B0%D0%BD%D0%BA%D0%B5%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) (дата обращения: 22.04.2023).
14. Рецензия на Little Nightmares 2. URL: <https://kanobu.ru/reviews/retsenziya-na-little-nightmares-2-376172/> (дата обращения: 22.04.2023).

УДК 004.738.5

**В. М. Цыгляев**

студент кафедры вычислительных систем и сетей

**Н. А. Балонин** – доктор технических наук, профессор – научный руководитель

### СПОСОБЫ ДОЛГОВРЕМЕННОГО СОХРАНЕНИЯ ИНФОРМАЦИИ НАУЧНЫХ САЙТОВ

Современный мир полон научной информацией, доступной через Интернет. Этой информацией могут быть сайты с научными статьями или сайты людей, которые занимаются наукой и загружают в открытый доступ свои научные труды. Однако не все эти научные сайты после потери активной поддержки своих создателей имеют гарантию своего существования в будущем. Это связано со следующими проблемами [1]:

- Изменение технологий. Когда сайты создаются, они могут быть оптимизированы для работы в конкретных браузерах или с определенными версиями программного обеспечения. Когда технологии меняются, сайты могут перестать работать, что может привести к их исчезновению.

- Недостаточное финансирование.

- Изменение приоритетов организаций, которые поддерживали эти сайты.

- Изменения в политике правительства или научных организациях.

- Недостаточная для автора сайта активность.

Так за последние десятилетия из Интернета исчезли десятки научных журналов и личных сайтов различных ученых, что привело к потере ценной информации для ученых и общественности. Данная потеря, в свое время, отразится в худшую сторону на научном наследии, поэтому эта проблема требует срочного решения, так как она угрожает научному развитию и образованию. Все это наводит на мысль о сохранении научных сайтов от исчезновения.

Для сохранения научных сайтов доступно несколько инструментов, однако самыми эффективными является создание зеркальных сайтов и выделение специального сервера для ученых, который будет основываться на «книгах нового поколения» (КНП).

Зеркальный сайт [2] – это полная копия оригинального сайта, которая размещается на другом сервере. Это позволяет сохранять сайт в его текущем состоянии и обеспечивает доступ к нему в случае, если оригинальный сайт становится недоступным или исчезает.

Создание зеркальных сайтов может быть произведено с использованием специализированных инструментов или программ, таких как HTTrack или WinHTTrack, которые позволяют загружать и сохранять сайты полностью на локальный диск или на другой сервер. Создание зеркальных сайтов также может быть выполнено вручную, путем загрузки и копирования всех файлов сайта на другой сервер.

Одним из главных преимуществ создания зеркальных сайтов является сохранение полного контента сайта и доступность сайта в случае, если оригинальный сайт становится недоступным или исчезает. Создание зеркальных сайтов также может ускорить загрузку сайта, так как они могут быть размещены на серверах с быстрым интернет-соединением.

Однако, следует отметить, что создание зеркальных сайтов может быть затратным процессом, особенно если требуется сохранить большой объем данных. Также, создание зеркальных сайтов не решает проблему обновления сайта, так как зеркальный сайт будет точной копией оригинального сайта на момент его создания, и любые изменения, сделанные на оригинальном сайте после этого, не будут отображаться на зеркальном сайте без совершения специальных действий.

Выделение сервера хранения КНП, основывается на хранении математических HTML-JS книг, что позволяет сохранять научную информацию в универсальном формате.

Данные книги легко скачиваются с серверов, и столь же легко возвращаются на них. Эта отделимость сайтов создает условия для их инъекции в окружающий нас мир с целью сохранения. Кроме того, данные книги легко обеспечивают интеграцию различных формул, расчетов, графиков, библиотек, трехмерных рисунков и анимаций. КНП своими 3D возможностями может способствовать созданию специфической ветви "виртуальной электроники" и робототехники, поскольку она обладает нужной для экспериментов средой.



Хоть сохранение научных сайтов является важной задачей для научного сообщества, однако существует несколько проблем, связанных с этим самым сохранением, которые затрудняют эту задачу:

1. Ограничения авторских прав.

Одна из главных проблем при сохранении научных сайтов – это ограничения авторских прав. Как правило, авторские права принадлежат владельцам сайтов, что означает, что без их разрешения сохранение или использование содержимого сайта может нарушать закон об авторском праве. Более того, авторские права могут быть сложным и изменчивым, что затрудняет определение того, какую информацию можно сохранить и какую нельзя. Так в гражданском кодексе Российской Федерации [1, 6] написано, что исключительное право действует в течении всей жизни автора и семидесяти лет, считая с 1 января года, следующего за годом смерти автора. Этот срок действия исключительного права делает невозможным зеркальное копирование сайта, если при своей жизни автор сайта не дал согласие на копирование.

2. Изменение формата сайтов и технологий.

Другой проблемой, связанной с сохранением научных сайтов, является изменение формата сайтов и технологий. Например, сайты, созданные на основе старых версий программного обеспечения, могут перестать работать после обновления ОС или браузера. Кроме того, новые технологии могут быть несовместимы с предыдущими версиями, что может привести к тому, что часть информации на сайте станет недоступной для сохранения.

Данная проблема почти полностью исчезает при использовании КНП, так как КНП реализуют единый универсальный формат HTML-JS страничек.

3. Ограниченные ресурсы для сохранения сайтов.

Третья проблема связана с ограниченными ресурсами для сохранения сайтов [3]. Это может быть связано с ограничениями на выделение бюджета на сохранение сайтов, недостаточной экспертизой в области сохранения сайтов, а также с отсутствием достаточно мощной инфраструктуры для хранения и поддержки сайтов на длительный период времени.

Для решения второй и третьей проблемы можно применять различные подходы. Например, можно разрабатывать новые технологии и форматы, которые бы увеличивали совместимость сайтов с различными версиями программного обеспечения. Кроме того, можно проводить обучение и развивать экспертизу в области сохранения сайтов для сотрудников библиотек, архивов и других организаций, которые занимаются сохранением научных данных.

К большому сожалению, проблему с авторским правом решить практически невозможно, так как для этого потребуется переписать данный пункт гражданского кодекса.

После определения проблем необходимо рассмотреть некоторые лучшие практики, которые помогут обеспечить эффективное сохранение научных сайтов:

Одной из лучших практик сохранения научных сайтов является поддержка от правительственных и некоммерческих организаций. Эти организации могут предоставлять финансовую поддержку и технические ресурсы для сохранения сайтов. Они также могут помочь в обеспечении правовой защиты для сохранения авторских прав на содержимое сайтов. В некоторых случаях правительственные организации могут также обеспечить юридическую поддержку для разрешения споров по поводу прав на сохраняемую информацию [4].

Другой важной практикой является распространение информации о сохранении научных сайтов [5]. Это может включать информацию о том, как сохранять сайты и как использовать сохраненную информацию. Распространение такой информации может помочь повысить осведомленность научного сообщества об этой проблеме и привлечь дополнительные ресурсы для сохранения сайтов.

Третьей важной практикой является сотрудничество между научными организациями и библиотеками. Библиотеки могут сыграть важную роль в сохранении научных сайтов, предоставляя хранилище для сохранения информации. Они также могут помочь в обеспечении доступности сохраненной информации, предоставляя доступ к сохраненным сайтам и обеспечивая сохранение необходимой информации для будущих поколений.

На основании всего вышесказанного мы можем констатировать, что сохранение научных сайтов является важным процессом, поскольку это позволяет сохранить научные знания и данные для

будущих поколений, тем самым повышая научный потенциал той страны, где это сохранение реализуется.

Лучшие практики, такие как поддержка от правительственных [6] и некоммерческих организаций, распространение информации о сохранении сайтов и сотрудничество между организациями, помогут эффективно сохранять научные сайты. Причем эффективнее всего этими практиками могут заниматься академии наук при использовании КНП, так как они имеют в своих руках достаточное количество ресурсов и финансирование, которые позволяют выделять сервера, заниматься издательством КНП и популяризацией идеи сохранения научных работ в эффективном и универсальном формате, организовывать юридическую поддержку, быть посредником между научными организациями и библиотеками.

#### Библиографический список

1. Десятки научных журналов с открытым доступом исчезли из интернета. URL: <https://22century.ru/society/91316> (дата обращения: 18.03.2023).
2. Mikael Laakso, Lisa Matthias, Najko Jahn. Open is not forever // A study of vanished open access journals. 2021. 14 p.
3. Десятки научных журналов исчезли из интернета за последние 20 лет, и никто их не сохранил. URL: <https://habr.com/ru/news/t/518594/> (дата обращения: 18.03.2023).
4. Как прожить без Sci -Hub или 10 полезных ресурсов Open Science, которые вас спасут. URL: <https://waksoft.susu.ru/2018/11/27/kak-zhit-bez-sci-hub-ili-10-poleznyih-resursov-open-science-kotoryie-vas-spasut/> (дата обращения: 18.03.2023).
5. Балонин Н. А. Перспективы хостинга для ученых в России. URL: <http://artspb.com/> (дата обращения: 18.03.2023).
6. Гражданский кодекс Российской Федерации. Часть 4. URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_64629/](http://www.consultant.ru/document/cons_doc_LAW_64629/) (дата обращения: 18.03.2023).

# ПРИКЛАДНАЯ МАТЕМАТИКА, ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ ТАМОЖЕННЫЕ ТЕХНОЛОГИИ

УДК 004.42

**Ю. В. Бульгин**

студент кафедры прикладной математики, информатики и информационных таможенных технологий  
ИФ ГУАП

**Р. А. Коваленко** – старший преподаватель – научный руководитель

## АРХИТЕКТУРА ОБРАЩЕНИЯ К ДАННЫМ В СОВРЕМЕННЫХ WEB-ПРИЛОЖЕНИЯХ

Базы данных используются в большинстве современных приложений, в том числе и в большинстве проектов веб-разработки. В их число входят программы как простейших блогов и каталогов, так и серьезных социальных веб-проектов. Какой бы не была сложность сайта и разработанной для него базы данных, для эффективной и надёжной работы каждый из них требует тщательного проектирования [1]. Проблемой использования языков объектно-ориентированного программирования в сочетании с базами данных является сложность их согласования между собой.

Реляционные базы представляют данные в табличном формате, в то время как объектно-ориентированные языки представляют их как связанный граф объектов, что влечет несоответствия во время сохранения этого графа объектов в реляционную базу или его загрузки из нее. Использование реляционной базы для хранения объектно-ориентированных данных приводит к семантическому разрыву, заставляя программистов писать программное обеспечение, которое должно обрабатывать данные в объектно-ориентированном виде, а также сохранять эти данные в реляционной форме. Эта постоянная необходимость в преобразовании между двумя разными формами данных не только сильно снижает производительность, но и создает трудности для программистов, так как обе формы данных накладывают определенные ограничения [2]. Следует также отметить, что проектирование информационных систем требует как описания логики взаимодействия компонентов, так и описание логики взаимодействия с базой данных, такими описаниями служат паттерны объектно-реляционной логики, которые активно применяются при обращении к данным в современных приложениях.

Рассмотрим, как происходит взаимодействие программы с базой данных на «низком уровне» т. е. без применения каких-либо упрощающих эту процедуру средств. В общем случае такое взаимодействие подразумевает сам SQL-запрос с учетом особенностей используемого СУБД, установление связи между приложением и данными из БД, возможные различные проверки на существование таких данных, их доступность, валидность и т. д., их последовательное считывание и их представление соответствующее модели конечного приложения

С подходом, использующим подобное взаимодействие связано довольно большое количество ограничений, дополнительных требований и трудозатрат. Например, какие-либо изменения в БД (типы данных, таблицы, их поля, связи) или переход на другую СУБД, связаны с необходимостью изменения соответствующего программного кода, исполняющего операции с доступом и представлением данных из БД. Таким образом, к недостаткам подхода, использующего прямое взаимодействие с базой данных SQL, можно отнести необходимость написания большого количества шаблонного кода, что в свою очередь требует большого количества времени на его реализацию и отладку, а также накладывает дополнительные требования в сопровождении и поддержании системы в рабочем и актуальном состоянии.

В современных приложениях основной технологией, решающей подобные трудности, является ORM (Object-relational mapping). Её использование решает проблему так называемой парадигмы «несоответствия». Это инструмент, решающий проблему семантического провала между реляционной и объектной моделями данных который создаёт слой между приложением и БД. ORM связывает реляционную базу данных с концепцией объектно-ориентированных языков программирования осуществляя перевод данных в свойства ООП-объектов. Основной задачей данной технологии является испол-

нение задач взаимодействия с БД посредством инкапсуляции её логики, что позволяет не учитывать особенностей содержимого БД при разработке приложения. Для этого ORM обеспечивает разработчика необходимым инструментарием и интерфейсом для CRUD-операций над данными [3].

```

1  static void Main(string[] args)
2  {
3      string connectionString = @"Data Source=.\SQLEXPRESS;Initial Catalog=usersdb;Integrated Security=True";
4
5      string sqlExpression = "SELECT * FROM Users";
6      using (SqlConnection connection = new SqlConnection(connectionString))
7      {
8          connection.Open();
9          SqlCommand command = new SqlCommand(sqlExpression, connection);
10         SqlDataReader reader = command.ExecuteReader();
11
12         if (reader.HasRows) // если есть данные
13         {
14             // выводим названия столбцов
15             Console.WriteLine("{0}\t{1}\t{2}", reader.GetName(0), reader.GetName(1), reader.GetName(2));
16
17             while (reader.Read()) // построчно считываем данные
18             {
19                 object id = reader.GetValue(0);
20                 object name = reader.GetValue(1);
21                 object age = reader.GetValue(2);
22
23                 Console.WriteLine("{0} \t{1} \t{2}", id, name, age);
24             }
25         }
26
27         reader.Close();
28     }
29
30     Console.Read();
31 }
    
```

Рис. 1. Обращение к данным с использованием нативных выражений языка SQL и C# классов для работы с БД

```

1  using System;
2  using System.Linq;
3
4  namespace HelloApp
5  {
6      public class Program
7      {
8          public static void Main(string[] args)
9          {
10             using (ApplicationContext db = new ApplicationContext())
11             {
12                 // создаем два объекта User
13                 User user1 = new User { Name = "Tom", Age = 33 };
14                 User user2 = new User { Name = "Alice", Age = 26 };
15
16                 // добавляем их в бд
17                 db.Users.Add(user1);
18                 db.Users.Add(user2);
19                 db.SaveChanges();
20                 Console.WriteLine("Объекты успешно сохранены");
21
22                 // получаем объекты из бд и выводим на консоль
23                 var users = db.Users.ToList();
24                 Console.WriteLine("Список объектов:");
25                 foreach (User u in users)
26                 {
27                     Console.WriteLine($"{u.Id}.{u.Name} - {u.Age}");
28                 }
29             }
30             Console.Read();
31         }
32     }
33 }
    
```

Рис. 2. Работа с базой данных с использованием ORM EntityFramework

В сочетании с ORM, или в качестве альтернативы в решении проблемы «несоответствия» также используется денормализация реляционных таблиц – намеренное приведение структуры базы данных в состояние, не удовлетворяющее требованиям нормализации [4]. А именно добавление избыточных данных в таблицу для ускорения доступа к данным. А также концепция реляционного отображения коллекций CoRM (Collection-Relational Mapping) которая опирается на то, что таблица как долговременное хранилище состояний объекта, имеет вполне адекватный образ в любой развитой объектно-ориентированной среде – коллекцию (Collection).

На представленном в рис. 2 программном коде рассматривается работа с базой данных через ORM. На котором можно увидеть несколько основных отличий от предыдущего подхода: отсутствие выражений на языке SQL – взаимодействие с БД происходит посредством уже существующих методов, отсутствие каких-либо проверок, последовательный построчный доступ к данным заменён на работу с объектом через его методы.

Таким образом ORM позволяет повысить абстракцию и скрыть логику БД предоставляя возможность работать с БД через ООП-классы.

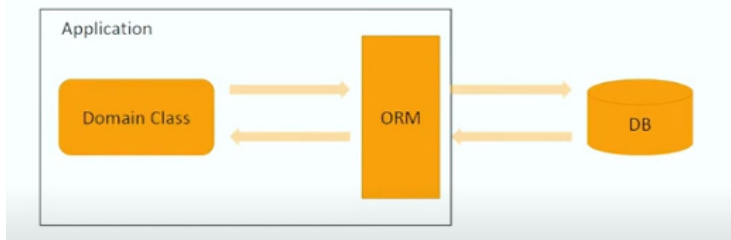


Рис. 3. Общая схема работы с базой данных с использованием ORM

При таком подходе изменяется и общая логика работы с данными. Один класс может объединять сразу несколько таблиц с данными которые были бы использованы в аналогичных задачах при использовании реляционных СУБД. В работе ORM используется ООП-концепция наследования, которая меняет как принцип доступа к данным, так и изменяет принцип связывания данных реляционной базы посредством внешних ключей. За счет этого появляется возможность создания интерфейсной модели, а также более быстрого и удобного доступа и обработки данных через связанные с объектами методы.

Среди паттернов управления выделена большая группа шаблонов, описывающих организацию связи с базой данных.

RowDataGateway (шлюз к данным записи) – последовательное, построчное сопоставление данных из БД с их последующим переводом в структурные типы данных. Поскольку Row Data Gateway представляет собой отдельную строку таблицы, для выбора необходимого ряда необходим компонент Finder.

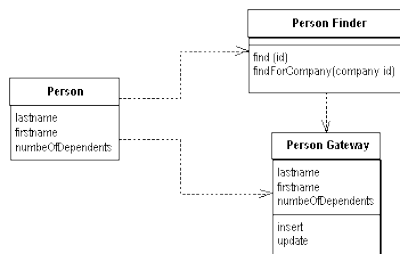


Рис. 4. UML-диаграмма паттерна RowDataGateway

TableDataGateway, который так же можно встретить под названием DAO (Data Access Object) – паттерн, в котором объект выступает в качестве шлюза между данными в приложении и определенной таблицей в базе данных, где работает сразу со всеми записями, и содержит все методы, работающие с данной таблицей.

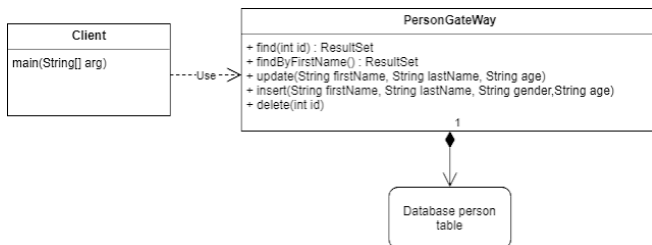


Рис. 5. UML-диаграмма паттерна TableDataGateway

ActiveRecord – один объект управляет и данными, и поведением Active Record в целом является развитием идеи Row Data Gateway, но позволяет хранить логику доступа к данным в объекте.

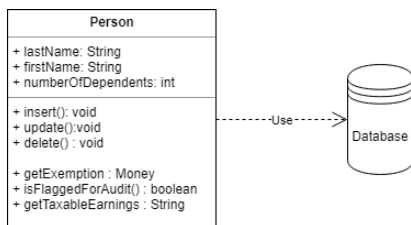


Рис. 6. UML-диаграмма паттерна ActiveRecord

Mapper – создает слой доступа к данным, который предоставляет двунаправленный «маппинг» данных между постоянным хранилищем данных и хранением данных в памяти. В отличие от Active Record, в Data Mapper появляется еще один слой или тип сущности такой как entityManager. Именно этот слой будет отвечать за перенос состояния модели в базу данных и обратно [5].

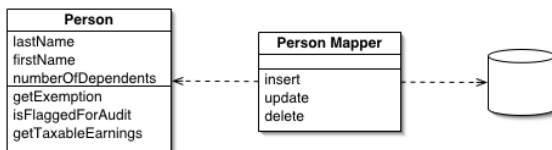


Рис. 7. UML-диаграмма паттерна DataMapper

Также технологией ORM используются паттерны объектно-реляционной логики:

IdentityMap (карта соответствия) – обеспечивает однократную загрузку объекта, сохраняя данные об объекте в карте соответствия. Паттерн Карта Соответствия хранит записи о всех объектах, которые были считаны из БД за время выполнения одного действия. Когда происходит обращение к объекту, проверяется карта соответствия, чтобы узнать, загружен ли объект. Таким образом, отказ от загрузки одних и тех же данных дважды не только обеспечивает корректность информации, но и ускоряет работу приложения.

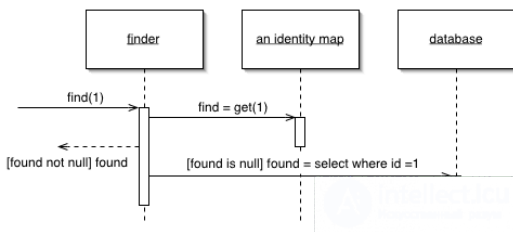


Рис. 8. Схема IdentityMap

Unit of Work (единица работы) – паттерн направлен на сохранение изменений в базу данных. Создает объект, отслеживающий изменения, вносимые в базу данных. Контролирует, какие объекты считываются, какие модифицируются, и производит операции сохранения изменений содержимого базы данных [5, 6].

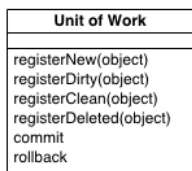


Рис. 9. Функции UnitofWork

При внедрении работы с данными через ORM выделяются следующие этапы: посредством классов описываются «сущности» – объекты, которые будут храниться в базе данных; определяется контекст – класс, который описывает и осуществляет взаимодействие с программы с базой данных; создается база данных посредством миграции – функции, позволяющей применять изменения к базе данных, посредством которой генерируется реляционная база данных со всеми описанными классами сущностями.

Применение архитектурных паттернов обращения к данным и технологии объектно-реляционных преобразований существенно упрощают внедрение баз данных в современных приложениях. Позволяя работать исключительно с ООП-моделями данных, их обновление и поддержка обеспечивается значительно меньшими затратами времени и усилий. Большинство операций выполняется автоматически. Можно работать без знаний и навыков в SQL программировании. Позволяет использовать ООП принципы как инструмент для организации долгосрочного хранения данных. Из недостатков можно отметить снижение производительности так как функции ORM библиотеки выступают в качестве «обертки» для SQLзапросов, а тот факт, что БД находится в абстрактном состоянии может помешать отладке и оптимизации процессов, происходящих с данными на низком уровне.

### Библиографический список

1. Основы правильного проектирования баз данных в веб-разработке. URL: <https://habr.com/ru/articles/115777/> (дата обращения: 14.04.2023).
2. ORM object-relational mapping (объектно-реляционное отображение). Варианты реализации (Active Record и Data Mapper) и альтернативы. URL: <https://intellect.icu/orm-object-relational-mapping-obektno-relyatsionnoe-otobrazhenie-varianty-realizatsii-active-record-i-data-mapper-i-alternativy-4701?ysclid=igm339zr28530163552> (дата обращения: 14.04.2023).
3. Руководство по фреймворку Blazor // METAINIT.COM. URL: <https://info-comp.ru/denormalizing-database?ysclid=lgoy97cj6r217394733> (дата обращения: 14.04.2023).

4. Что такое ORM? Почему стоит использовать ORM? Базы данных: [лекция] / Захар Лучинин. URL: [https://www.youtube.com/watch?v=9wONUQCMOQQ&list=PLAyGZSpiecFrZx\\_qX6em16P9IIuoaoOG-Y&index=12](https://www.youtube.com/watch?v=9wONUQCMOQQ&list=PLAyGZSpiecFrZx_qX6em16P9IIuoaoOG-Y&index=12) (дата обращения: 14.04.2023).

5. Паттерны проектирования: справочник. URL: [design-pattern.ru](http://design-pattern.ru) (дата обращения: 14.04.2023).

6. Паттерны объектно-реляционной логики // Lazy Load, Identity Map, Unit of Work. URL: <https://intellect.icu/patterny-obektno-relyatsionnoj-logiki-lazy-load-identity-map-unit-of-work-7704?ysclid=lgtnlv55v796140281> (дата обращения: 14.04.2023).

7. Catalog of patterns of Enterprise Application Architecture. URL: <https://martinfowler.com/> (дата обращения: 14.04.2023).



## УДК 004.65

**А. П. Викторова**студентка кафедры прикладной математики, информатики и информационных таможенных технологий  
ИФ ГУАП**Р. А. Коваленко** – старший преподаватель – научный руководитель**ОБЗОР И СРАВНЕНИЕ ЛОКАЛЬНЫХ ИСТОЧНИКОВ ХРАНЕНИЯ ДАННЫХ  
ДЛЯ ИНФОРМАЦИОННЫХ СИСТЕМ****Введение**

На сегодняшний день разработчики сайтов заинтересованы в том, чтобы удержать внимание пользователя. В этом, помимо красивого графического дизайна им помогает сохранение информации о состоянии страницы. Для этого используются различные инструменты, такие как cookie, локальное хранилище, сессионное хранилище, IndexedDB, WebSQL. Эти средства в общем выполняют одну и ту же задачу, однако различаются в деталях и могут быть использованы в разных ситуациях. Локальное хранение информации позволяет сократить время, необходимое для загрузки страницы и повышает безопасность и конфиденциальность за счёт того, что данные не отправляются на сервер, а хранятся на устройстве [1].

**Cookie**

Куки являются распространённым способом хранения небольших (4 кБ) строковых данных внутри документа. Куки пересылаются с сервера клиенту и обратно, управляя сессиями аккаунта или отслеживая информацию пользователя. В них может содержаться такая информация, как логин и пароль, язык, геолокация, состояние страницы [2]. Также куки могут собирать данные о

Файлы куки бывают следующих типов:

1. Временные. Действительны во время текущей сессии и очищаются каждый раз при выходе из браузера.
2. Постоянные. Остаются на устройстве даже после перезахода в браузер. Очищаются вручную через специальные настройки.
3. Сторонние. Эти куки принадлежат сторонним доменам. К примеру, рекламный баннер на сайте с другим доменом позволяет браузеру, кроме основных cookie, отправлять стороннего сайта. Это необходимо для оценки того, насколько эффективна реклама.
4. Зомби-cookie. Представляют из себя весьма трудноудаляемые файлы, находящиеся в других хранилищах, которые связаны с процессами браузера, например, в API Java, HTTP ETag, Microsoft Silverlight, Indexed DB. Лучший способ не допустить попадания таких файлов на компьютер – это использование исключительно HTTPS-соединения. Такие файлы считаются запрещёнными, во многих поисковых системах существует блокировка сайтов, которые пытаются внедрить зомби-cookie на ваш компьютер [3].

**Локальное хранилище**

Локальное хранилище относится к Web Storage API. Это позволяет хранить данные в браузере в формате ключ-значение. Разработчик сохраняет информацию в браузере пользователя до тех пор, пока пользователь не удалит их вручную. Данный API является более безопасной и интуитивной альтернативой куки.

Сессионное хранилище также относится Web Storage API, но в отличие от локального хранилища, данные пользователя сохраняются только для текущей сессии (вкладки) браузера. При закрытии вкладки и завершения работы браузера, информация о пользователе удаляется.

**IndexedDB**

Для хранения большого объёма информации в браузере используется IndexedDB (NoSQL database), которая основана на JavaScript. Несмотря на свою сложность по сравнению с Web Storage API, IndexedDB имеет ряд преимуществ:

- 1) обработка более сложных упорядоченных данных;
- 2) возможность работы с различными базами данных и таблицами внутри базы данных;
- 3) большой объём хранения информации.

Любые операции в БД осуществляются в виде атомарных транзакций. Данные транзакции дают возможность при необходимости сделать бэкап данных [4].

### WebSQL

База данных WebSQL используется при разработке оконных приложений, что отличает её от IndexedDB, которая применяется для создания сайтов или веб-сервисов. Размер базы WebSQL ограничен свободным местом на диске [5]. WebSQL сохраняет данные в таблице локальной БД, а не в формате ключ-значение, как это реализовано в LocalStorage.

База данных WebSQL поддерживается такими браузерами, как Opera, Google Chrome, а также Android. В браузерах Safari 13, Firefox и Internet Explorer поддержка не осуществляется. Отсутствие поддержки WebSQL не означает полный отказ от этой технологии, однако интерес разработчиков к ней снижается. Причина такого отношения к WebSQL заключается в том, что база данных основана на принципах SQL-стандарта – сторонней технологии. К тому же, WebSQL может быть небезопасной для хранения данных пользователя, так как она предназначена для работы с БД на клиенте [6].

### Вывод

Подводя итоги, можно сказать, что в настоящее время существует множество различных способов локального хранения пользовательских данных. Эти источники локального хранения, ввиду своих особенностей, применяются для различных целей и задач. Куки удобно использовать на сайтах интернет-магазинов или других страниц, для посещения которых требуется авторизация. Среди минусов куки можно выделить малый размер памяти для хранения (4 Кб) и незащищённость от действий злоумышленников. Локальное хранилище, по сравнению с куки, обладает большей памятью хранения (от 5 до 10 Мб) и работает исключительно на стороне клиента. Однако LocalStorage уязвим перед атаками на основе межсайтовых сценариев, т. е. XSS. IndexedDB работают с упорядоченными данными, однако эта технология более сложная, чем Local Storage. WebSQL сохраняет данные в полноценной таблице локальной БД, однако на данный момент не поддерживается. С учётом приведённой выше информации, разработчик может выбрать любое средство для реализации нужных задач.

### Библиографический список

1. База данных для сайта // Инфографика. URL: <https://wiserv.ru.turbopages.org/wiserv.ru/s/blog/web/database-for-website> (дата обращения: 20.04.2023).
2. Обзор средств хранения данных на клиентской стороне // Прогрессор. URL: <https://prgssr.ru/development/obzor-sredstv-hraneniya-dannyh-na-klientskoj-storone.html> (дата обращения: 20.04.2023).
3. Что такое cookie в браузере и почему на многих сайтах предупреждают об их использовании? // DNS Клуб. URL: <https://club-dns-shop.ru.turbopages.org/club.dns-shop.ru/s/blog/t-326-internet/47805-chtotakoe-cookie-v-brauzere-i-pochemu-na-mnogih-saitah-preduprejdi/> (дата обращения: 20.04.2023).
4. Изучаем IndexedDB // webdevblog.ru. URL: <https://webdevblog.ru.turbopages.org/webdevblog.ru/s/izuchаем-indexeddb/> (дата обращения: 20.04.2023).
5. WEB SQL в JavaScript // Дзен. URL: [https://dzen.ru/a/Xsw0C1frWhcl4nU\\_](https://dzen.ru/a/Xsw0C1frWhcl4nU_) (дата обращения: 20.04.2023).
6. WebSQL – реляционная база данных на веб-странице // Учебник HTML5. URL: <http://html5ru.com/websql-relyacionnaya-baza-dannyx-na-veb-stranice.html> (дата обращения: 20.04.2023).

**Е. Н. Долгов**

студент кафедры прикладной математики, информатики и информационных таможенных технологий  
ИФ ГУАП

**А. А. Сорочкин** – старший преподаватель – научный руководитель

### **ПЕРСПЕКТИВЫ ПРИМЕНЕНИЯ МЕХАНИЗМОВ НА ОСНОВЕ ШАРОВЫХ ШЕСТЕРНЕЙ В НАСТОЯЩЕМ ВРЕМЕНИ**

Сегодня мир вошел в новое десятилетие, когда робототехника не только является нетомленной частью серийного производства, но и входит в повседневную жизнь. Современное состояние искусственного интеллекта позволяют повысить степень автоматизации, что актуально в условиях: всплеск пандемий, для баланса производительности и энергоэффективности, автоматической коррекции при прототипировании и процесса производства посредством 3D-печати.

Другая причина – экономическая. Заменяв роботом работника-человека, предприятия минимизируют человеческий фактор – основную проблему контроля качества, тем самым повышая производительность за счет непрерывной работы и качество продукции, что немало важно в условиях быстро растущей мировой экономики, а также повышая безопасность технологических процессов, устраняя предпосылки аварии по вине человека.

Требуемые для этого различные манипуляторы требуют массового производства, различные мобильные робототехнические комплексы тоже нашли свое применение жизни и несмотря на высокие цены выходят на рынок. Поэтому сейчас идёт процесс поиска новых нестандартных решений.

Последние достижения в технологиях проектирования и производства посредством 3D печати делают возможным практическое изготовление сложных форм, в том числе массовое. То, что раньше оставалось чертежами и единичными прототипами сложными в производстве, теперь можно производить и вносить правки в каждое следующее изделие проще, быстрее и дешевле.

Примером такой вещи являются шаровые приводы. Несмотря на то, что появились они еще во второй половине прошлого века. Примером может быть патент СССР от 1989 года [3]. Проектирование и фрезеровка таких шестерней были сложными процессами, требующими большой точности, следовательно, их применение нерациональным. Такой механизм позволяет собирать суставы робота в одной точке (т. е. на активном шаровом шарнире), снижая потребление ресурсов и повышая экономичность работы сустава робота или протеза.

Развитие технологий проектирования и роста качества производства посредством 3D печати, использование современных материалов на основе композитных видов пластика, например, стеклонанополненный полиамид, вкпе позволяют возобновить эксперименты с применением шаровых приводов.

Такие приводы состоят из главного шарнира и специально расположенных направляющих колес. А самым совершенным на момент написания статьи является механизированный сустав на основе шаровой шестерни под названием ABENICS созданный Университетом Ямагата в Японии. Пользователи из Интернета уже успели повторить его на 3D-принтере и высказали собственное мнение.

Данный привод состоит из шаровой шестерни с выгравированными по всей поверхности зубьями и двух направляющих шестерней-приводов. Каждая направляющая шестерня приводится в действие двумя электродвигателями (в прототипе посредством червяковой передачи) – один крутит шестерню вокруг своей оси и главную вокруг первой, а второй двигатель за счет зацепления зубьев главной шаровой и направляющей поворачивает главную.

Особенностью механизма является широкий спектр поворота, высокая точность за счет конструкторских особенностей, большая функциональность за счет 3-х степеней свободы в одном суставе и свободного расположения направляющих шестерней. Например, при расположении друг напротив друга она допускает диапазон движения по трем осям [360, 90, 360] градусов, при применении напечатанного фиксатора изменен на [180, 180, 360] градусов (рис. 1).

Каждая направляющая шестерня, входит в зацепление с конструкцией зуба главной, ограничивает её вращение, за исключением вращения вокруг оси. Другими словами, один привод всегда управляет двумя из трех степеней свободы, за счет некоммутативности сложения углов поворота шар можно

перевести в любую точку пространства. Вместе шестерни имеют 4 физических степени свободы, одна из которых заблокирована, поскольку оси обоих направляющих шестерней ортогональны, две шестерни могут ограничивать или управлять всеми тремя степенями свободы шаровой шестерни. Механическая система исходя из формул имеет 12 контролируемых степеней, что делает её избыточной.

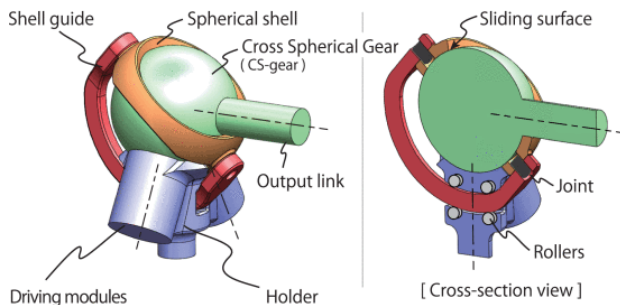


Рис. 1. Одна из возможных конфигураций [2]

Притом взаимосвязь между двумя приводными модулями должна удовлетворять следующим двум условиям:

- 1) оси вращения всех соединений пересекаются в центре главной шестерни;
- 2) угол между осями приводов равен не менее 60 градусов.

Чтобы понять, насколько такой механизм может заменить классические решения на практике и при серийном производстве следует сравнить прототип создателей ABENICS с аналогом на классической схеме. Рассмотрена конфигурация, которая может воспроизводить аналогичный диапазон движений что и сустав с шаровой шестерней (рис. 2).

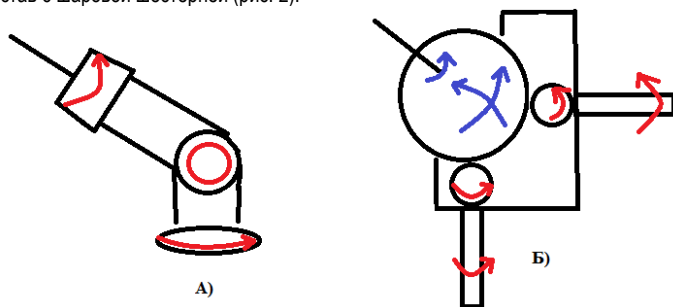


Рис. 2. Диапазон движения схем: а – классическая; б – шаровая шестерня

Во-первых, в классической схеме повышенные требования надежности и мощности выдвигаются к двигателям, расположенным ближе к основанию плеча, так как на него ложится вес плеча, содержащего ещё два двигателя, для такого же диапазона подвижности. В шаровом суставе такого не происходит, можно поставить одинаковые двигатели, зафиксированные эргономично и неподвижно. Вследствие отсутствия тяжелого плеча шаровой сустав меньше подвержен инерции. Дополнительно за счет встроенной точности его система управления не требует дополнительных вычислительных ресурсов.

Во-вторых, за время своего существования классические манипуляторы избавились от своих проблем, в то время как шаровой сустав обладает множеством нерешенных проблем, что сводит экономические преимущества на нет если таковые есть. Основной источник проблем шаровой шестерни. К проблемам прототипа можно отнести:

1) высокие требования к точности изготовления и постобработке, особенно прочности поверхностей трения, следствие дороговизна, сложность обслуживания и производства;

2) требуется масляный картер для постоянной смазки зубьев шестерни, совмещенный с механизмом удержания и защиты шаровой шестерни от внешних воздействий. Без фиксации теряется жесткость и возможность переносить нагрузки, а с нагрузками растёт трение;

3) повышенный износ и склонность к заеданию всей конструкции: шаровой шестерни и червяковой передачи, вследствие истирания появляется люфт и ошибки позиционирования. Для уменьшения износа шаровой шестерни можно использовать в качестве материала стеклонаполненные полипропилен или полиэтилен, прочный и устойчивый к истиранию;

4) пользователи Интернета, успевшие повторить механизм на 3D принтере обратили внимание на 4 "мёртвых" точки шаровой шестерне, совпадающих с осями кольцевых проточек. В этих точках она не цепляется зубами модулей. И множество движений этому суставу недоступны;

5) малая площадь контакта зубьев и соответственно относительно не высокие максимальные нагрузки на зуб.

Так как червяковая передача в данном случае включена в прототип с целью уменьшать люфт и улучшать точность позиционирования её особенности заслуживают отдельного упоминания. Так к достоинствам можно отнести плавность работы и повышенную кинематическую точность;

Также, как и большое передаточное отношение одной пары, – червячные редукторы с большим передаточным числом значительно компактнее и легче, чем эквивалентные шестеренчатые.

Самоторможение – при некоторых передаточных отношениях.

Очевидный недостаток – это существенное взаимное проскальзывание рабочих поверхностей, отсюда:

1. Высокие требования к геометрической точности и прочности поверхностей трения, необходимость точной регулировки.

2. Малый КПД с большими потерями на трение с тепловыделением, необходимость теплоотвода и компенсации осевых усилий, возникающих в опорах червячного вала.

3. Быстрый износ, ограничивающий контактные напряжения, при малейшем превышении расчетной нагрузки или проблемах со смазкой.

Как итог можно сказать, что созданный узел на основе шаровой шестерни, имеет интересные особенности благодаря чему найдет применение в своей нише. Но имеет слишком много недостатков для массового применения и производства, так как не решены проблемы удержания и защиты шаровой шестерни в суставе под нагрузкой и его общей надежности и жесткости. Применение технологий 3D-печати позволяет изготавливать комплекс шестерней достаточно точно, но требует некоторых затрат.

В настоящий момент суставы на основе классической схемы дешевы, ремонтпригодны и достигли за последнее десятилетие определенного совершенства благодаря чему превосходят механизмы на основе шаровой шестерни.

### Библиографический список

1. Технология ABENICS: революция в области механики? // Хабр. URL: <https://habr.com/ru/company/first/blog/711292/> (дата обращения: 01.02.2023).

2. ABENICS: Active Ball Joint Mechanism with Three-DoF Based on Spherical Gear Meshings / K. Abe, K. Tadakuma, R. Tadakuma // IEEE Transactions on Robotics. 2021. V. 37. No. 5. Pp. 1806–1825. DOI: 10.1109/TRO.2021.3070124.

3. Авторское свидетельство № 1646863 А1 СССР, МПК В25J 17/00. Приводной сферический шарнир: № 4642825: заявл. 30.01.1989: опубл. 07.05.1991. EDN LADDHB.

4. Wear-Resistant Glass-Filled Composites Based on Ultrahigh-Molecular-Weight Polyethylene. Role of Adhesion Varied with Coupling Agents / S. V. Panin, S. A. Bochkareva, B. A. Lyukshin [et al.] // Physical Mesomechanics. 2021. Vol. 24. No. 5. P. 548–560. DOI 10.1134/S1029959921050064. EDN PNJVTC.

УДК 004

**М. С. Смирнов**студент кафедры прикладной математики, информатики и информационных таможенных технологий  
ИФ ГУАП**А. А. Сорочкин** – старший преподаватель – научный руководитель**ВЛИЯНИЕ ОБЪЕКТОВ ДИНАМИЧЕСКОГО ОКРУЖЕНИЯ НА ПРОИЗВОДИТЕЛЬНОСТЬ UNITY**

Динамические объекты окружения позволяют создать из статичной сцены, реалистичную симуляцию реального мира. В основном используются методы, основанные на взаимодействии физических объектов, так и элементов с использованием анимаций.

Современные игровые движки позволяют без всяких проблем создавать и взаимодействовать с физическими объектами. В Unity за физику отвечает модифицированный и разработанный Nvidia – PhysX, путем использования компонента Rigidbody [1].

Rigidbody – это компонент в Unity, который добавляет физическое поведение к объекту на сцене. Когда объект имеет этот компонент, он подчиняется законам физики, таким как гравитация, столкновения, силы, вращение и т. д. Это позволяет создавать более реалистичные и интерактивные игровые механики или объекты окружения. Для оптимизации производительности у Rigidbody есть функция засыпания. Если Rigidbody имеет векторную скорость ниже пороговой определенный промежуток времени – он засыпает, и в этом состоянии он не рассчитывает физическое взаимодействие и не обновляет свое положение, пока не произойдет столкновение или не будет принудительно разбужен. Это позволит снизить нагрузку на процессор, при большом количестве физических объектов на сцене [2].

Взаимодействие с объектом с физической стороны происходит при помощи такой же основанной на физике силы или установки векторной скорости. Например, при помощи метода AddForce по определенному вектору можно создать силу в ньютонах [3].

Для тестирования, как на производительность влияют физические элементы окружения, на заготовленной сцене будет создано различное количество физических объектов, и будет замерено количество кадров в секунду.



Рис. 1. Количество мелких объектов на сцене 180

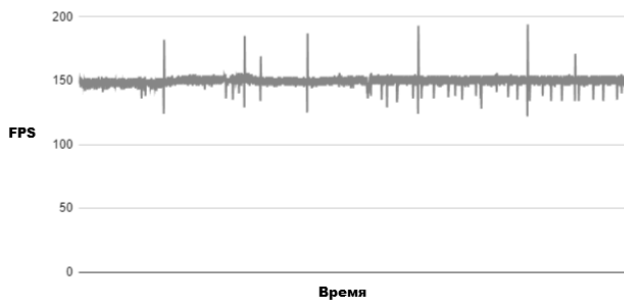


Рис. 2. График кадров в секунду за единицу времени при 200 динамических объектах на сцене

При таком количестве физических объектов, среднее количество кадров стало равно 150.



Рис. 3. Количество мелких объектов на сцене 1000

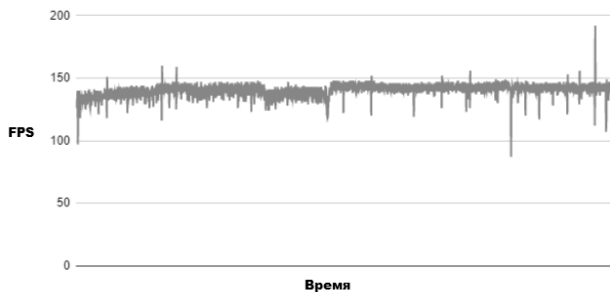


Рис. 4. График кадров в секунду за единицу времени при количестве объектов 1000

140. Можно заметить, что среднее количество кадров упало примерно на 10 единиц, и стало равно



Рис. 5. Количество мелких объектов на сцене 2000

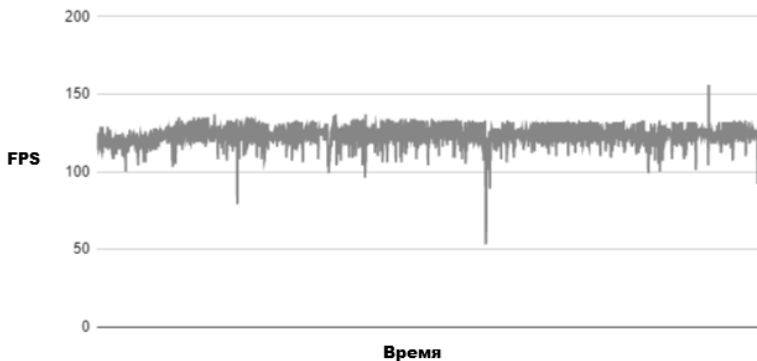


Рис. 6. График кадров в секунду за единицу времени при количестве объектов 2000

Количество кадров постепенно снижается, и с 2000 объектов среднее количество стало уже равно 130 кадрам.

Как можно заметить, при количестве объектов больше хоть и всего в 5 раз, количество кадров снизилось более чем в 30 раз. Это связано с тем, что процессорных мощностей уже не хватает для обработки такого количества объектов в реальном времени, из-за чего возникает "бутылочное горлышко" в системе. Даже с учетом того, что они могут уходить в режим сна. При этом, ситуация может быть еще хуже, если эти объекты будут одновременно находиться в динамике.





Рис. 7. Количество мелких объектов на сцене 10000



Рис. 8. График кадров в секунду за единицу времени при количестве объектов 10000

Но при этом, стоит отметить, что 10 тысяч одновременно работающих с физикой объектов является избыточным для видеоигры, работающей в реальном времени.

При большом количестве физических объектов, особенно ригдолла, могут возникнуть проблемы с производительностью, но возможно ли исправить эту проблему альтернативным методом – анимациями?

Помимо физики, динамику в окружении могут добавлять анимированные объекты. Для анимаций на движке Unity используется компонент Animator, который основан на Mecanim animation system. Эта система основана на концепции конечных автоматов, где каждая анимация представляет собой состояние, а переходы между анимациями контролируются условиями. Конечный автомат состоит из набора состояний, каждое из которых представляет анимацию, и набора переходов между этими состояниями. Каждое состояние может иметь свой собственный анимационный клип, настройки смешивания и другие параметры [4].

При использовании ригдолла, т. е. физического поведения тел, на 50 экспериментальных персонажах в рамках сцены, мы получаем порядка 30–40 кадров в секунду.



Рис. 9. Демонстрация физического поведения персонажей

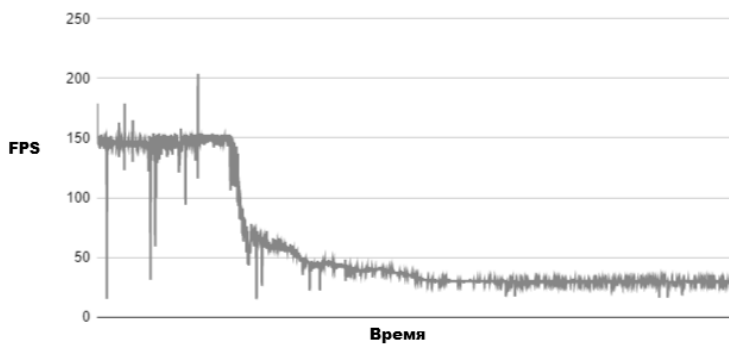


Рис. 10. График кадров в секунду за единицу времени при физической модели персонажей



Рис. 11. Демонстрация анимированной смерти персонажей

При анимированной “смерти” персонажа, теряется уникальность поведения, и персонажи падают в одинаковую позу. При этом, количество кадров не становится выше, а наоборот – ниже. Можно сказать, что система анимаций Unity не является высокопроизводительной, для работы с большим количеством объектов, и использование ее в рамках оптимизации вместо физической модели будет бесполезным.

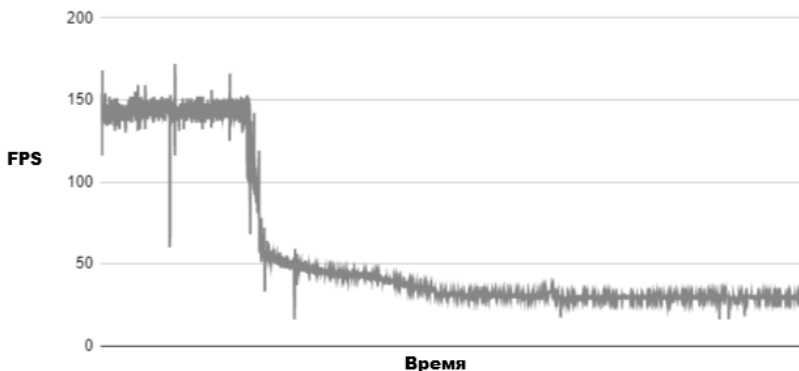


Рис. 12. График кадров в секунду за единицу времени при анимированной смерти персонажей

По итогу было измерено влияние на производительность динамических объектов на движке Unity, и рассмотрены потенциальные методы решения проблемы с производительностью при помощи системы анимаций. Динамические объекты, если не контролировать их количество на сцене, и их состояние сна, могут вызывать серьезные проблемы с производительностью.

#### Библиографический список

1. Unity: Официальная документация. URL: <https://docs.unity3d.com/ru/2019.4/Manual/class-Rigidbody.html> (дата обращения: 01.03.2023).
2. Unity: Официальная документация. URL: <https://docs.unity3d.com/ru/2019.4/Manual/RigidbodyOverview.html> (дата обращения: 01.03.2023).
3. Unity: Официальная документация. URL: <https://docs.unity3d.com/ScriptReference/Rigidbody.AddForce.html> (дата обращения: 01.03.2023).
4. Unity: Официальная документация. URL: <https://docs.unity3d.com/462/Documentation/Manual/MecanimAnimationSystem.html> (дата обращения: 01.03.2023).

## УДК 004.4'24

**А. М. Халтсонен**

студент факультета СПО

**А. А. Сорокин** – старший преподаватель – научный руководитель**ИГРОВЫЕ ДВИЖКИ С ПОДДЕРЖКОЙ МОБИЛЬНОЙ РАЗРАБОТКИ:  
ПРЕИМУЩЕСТВА И НЕДОСТАТКИ****Введение**

Игровые движки с поддержкой мобильной разработки – это программные инструменты, которые помогают разработчикам создавать игры для мобильных устройств. Они предоставляют набор инструментов, библиотек и компонентов, которые упрощают процесс разработки игр и позволяют создавать высококачественные проекты за более короткий промежуток времени.

**Индустрия мобильных игр**

Индустрия мобильных игр продолжает свое быстрое развитие, становясь одной из самых перспективных и динамично развивающихся отраслей в сфере игровых технологий. Она представляет собой весьма крупный рынок, который насчитывает сотни миллионов пользователей и генерирует миллиарды долларов выручки каждый год.

Среди главных причин такого успеха можно выделить:

1. Рост количества смартфонов и планшетов. С развитием технологий и повышением уровня жизни, все больше людей получают доступ к мобильным устройствам. Это открывает новые возможности для разработки и использования мобильных приложений, в том числе и игр.

2. Быстрое развитие технологий. Современные мобильные устройства обладают мощными процессорами, большим объемом памяти и высокоскоростным интернет-соединением. Это позволяет разработчикам создавать более сложные и красивые игры, которые становятся более привлекательными для пользователей.

3. Легкость скачивания. Большинство мобильных игр доступны для бесплатного скачивания из онлайн-магазинов приложений, таких как Google Play и App Store. Это делает их доступными для всех, кто имеет смартфон или планшет.

4. Возможности монетизации. Многие мобильные игры бесплатны для скачивания, но содержат внутриигровые покупки, которые позволяют игрокам получать дополнительные возможности в игре. Это позволяет разработчикам получать прибыль от игры, а игрокам – улучшать свой опыт и получать дополнительное удовольствие.

**Движки для мобильных игр**

Движок для мобильной разработки – это программное обеспечение, которое помогает разработчикам создавать мобильные приложения для разных платформ, таких как iOS, Android, Windows Phone и другие. Они облегчают процесс разработки, позволяя экономить время и снижают затраты на проекты.

**1. Движок Unity**

Unity – это самый популярный движок для создания 3D-видеоигр профессионального уровня. Школа программирования для детей в Калифорнии описывает Unity как «и слишком сложный, и слишком простой». В то же время Unity «более чем достаточно для создания потрясающих 3D-игр».

Преимущества Unity:

- Один из самых быстрых движков для создания игр.
- Поддерживает около 27 платформ, включая iOS и Android.
- Обеспечивает высококачественные видео- и аудио эффекты.
- Есть множество обучающих курсов, которые позволят вам быстро разобраться с созданием игр и приступить к работе.
- Огромное сообщество делает Unity предпочтительным движком.

– Простая отладка.

У Unity есть бесплатная версия, но предлагаются и расширенные варианты Pro, Plus или Enterprise за отдельную плату.

Несмотря на все положительные особенности, некоторые разработчики отмечают Unity как продукт, с которым легко начать работать. Но чтобы стать опытным разработчиком игр, нужно время и усилия. Они отмечают, что этот движок подходит не для всех проектов, особенно, когда речь идёт о разработке пользовательских интерфейсов или 2D-игр. Кроме того, для работы и сборки игр на Unity нужен мощный компьютер.

## 2. Движок Unreal Engine

Unreal Engine впервые появился в одноимённом шутере от первого лица, который вышел ещё в 1998 году. Первоначально разработанный Epic Games, он стал open source проектом в 2005 году. Вы можете бесплатно скачать Unreal Engine на GitHub.

Популярные игры, разработанные с помощью Unreal: Mortal Kombat, Batman: Arkham Asylum & City, A Way Out, King's Bounty II и Biomutant.

Unreal также поддерживает Windows, iOS, Linux, Oculus Rift, PlayStation, Xbox и многие другие платформы.

Плюсы Unreal Engine:

– Простое начальное обучение для разработчиков с небольшим опытом программирования или без него.

– Поддерживает быстрое тестирование.

Но по сравнению с Unity у него меньше плагинов расширения, процент с продаж для больших проектов и, как и в предыдущем случае, профессионалом в Unreal Engine стать довольно сложно.

## 3. Движок Solar 2D

Solar 2D (бывший Corona SDK) – один из самых известных и популярных игровых движков для создания 2D-игр. Он работает с языком сценариев Lua, предназначенным для создания лёгкого, но быстрого кода.

Преимущества:

– Поддерживает множество платформ.

– Разработчики ценят открытое сообщество Solar 2D, которое чрезвычайно полезно для тех, кто только начинает работать в игровой сфере.

– Ценная особенность Solar 2D – моделирование в реальном времени, дающее возможность сразу узнать, как ваша игра будет выглядеть после изменений.

– Эта платформа не содержит скрытых платежей, сборов и лицензионных отчислений.

Как понятно из названия, Solar 2D предназначен для разработки только двухмерных игр, и это одно из главных ограничений для него. Впрочем, если вам не нужен объём, то в первую очередь присмотритесь к этой платформе.

## 4. Движок PhoneGap

PhoneGap – это фреймворк для разработки мобильных приложений на основе веб-технологий. Это означает, что разработчик может использовать HTML, CSS и JavaScript для создания приложений, которые могут быть запущены на разных платформах, включая iOS, Android и Windows Phone.

Одно из главных преимуществ PhoneGap заключается в том, что он обеспечивает быструю и простую разработку мобильных приложений. Будучи основанным на веб-технологиях, PhoneGap уменьшает необходимость изучать новые языки программирования или фреймворки. Это делает его одним из самых доступных инструментов для тех, кто хочет начать разрабатывать мобильные приложения.

Ещё одним преимуществом PhoneGap является возможность создания мобильных приложений, которые могут работать в оффлайн-режиме. Приложения, построенные на базе PhoneGap, могут сохранять весь свой контент на устройстве, а не в облаке, что позволяет пользователю получить доступ к приложению в любое время, даже если он не имеет подключения к Интернету.

Также наиболее важным преимуществом PhoneGap является его способность стремительно создавать мобильные приложения на разных платформах. Вместо того, чтобы сосредотачиваться на разработке отдельных приложений для каждой платформы, разработчики могут использовать PhoneGap для создания единственного кодовой база, который может работать на всех платформах.

В целом, PhoneGap – это мощный и простой фреймворк для разработки мобильных приложений с использованием веб-технологий. Быстрая разработка, возможность работы в офлайн-режиме и возможность создания единого кода для разных платформ делают его привлекательным для многих разработчиков.

### **5. Движок Appcelerator Titanium**

Appcelerator Titanium – это открытая платформа для разработки мобильных приложений, которая позволяет разработчикам использовать один язык программирования (JavaScript) для создания приложений для iOS, Android и других мобильных платформ.

Преимущества Appcelerator Titanium:

- Кроссплатформенность: разработчики могут создавать приложения для разных мобильных платформ с помощью одного языка программирования, что значительно упрощает разработку и экономит время.

- Высокая скорость разработки: с помощью Appcelerator Titanium разработчики могут создавать мобильные приложения в разы быстрее, чем с использованием традиционных методов разработки.

- Единое API: Titanium предоставляет единое API для различных платформ, что позволяет разработчикам использовать отдельные API платформ на многих платформах.

- Нативные компоненты: многие UI-элементы в Titanium могут быть созданы с использованием нативных компонентов для каждой платформы, что обеспечивает более высокую производительность приложений.

- Большое количество плагинов: Titanium имеет множество плагинов, которые ускоряют процесс разработки и позволяют добавлять новые функции и возможности в мобильные приложения.

Сообщество разработчиков: Appcelerator Titanium имеет активное сообщество разработчиков, которые создают различные инструменты, библиотеки и плагины для этой платформы, что обеспечивает поддержку и развитие проекта.

- Открытый исходный код: Appcelerator Titanium распространяется под лицензией Apache 2.0, поэтому разработчики могут использовать исходный код для своих проектов без ограничений.

### **Преимущества и недостатки игровых движков с поддержкой мобильной разработки**

Игровые движки с поддержкой мобильной разработки имеют свои плюсы и минусы.

Преимущества:

- Ускоренная разработка. Использование готовых и оптимизированных движков позволяет сократить время на разработку игры и сосредоточиться на создании уникального геймплея и содержания.

- Мобильная совместимость. Мобильные движки обладают поддержкой различных платформ и устройств, что позволяет создавать игры для крупных мобильных операционных систем, таких как Android и iOS.

- Готовая экосистема. Большинство мобильных движков имеет свои SDK и элементы пользовательского интерфейса, что облегчает создание привлекательных игровых механик и интерфейсов для пользователей.

Недостатки:

- Ограничения возможностей. Использование готовых движков может создавать ограничения в функциональности и производительности для игрового приложения.

- Сложность кастомизации. Игровые механики, которые не поддерживаются мобильным движком по умолчанию, могут оказаться затруднительными в реализации.

– Высокая стоимость. Некоторые мобильные движки могут быть очень дорогостоящими и требовать значительных затрат на их покупку и поддержку.

В целом, использование игровых движков с поддержкой мобильной разработки предлагает ряд преимуществ, но и имеет свои ограничения, требующие внимательного взвешивания.

### **Заключение**

В целом, индустрия мобильных игр продолжает быстро развиваться, предлагая пользователям все более качественные и увлекательные игры. Она остается одной из наиболее перспективных и высокодоходных отраслей в сфере развлечений и технологий, и, скорее всего, будет продолжать расти и развиваться в ближайшие годы.

### **Библиографический список**

1. Движки для мобильных игр. URL: <https://appbooster.com/blog/podborka-luchshikh-dvizhkov-dlia-mobilnykh-igr/> (дата обращения; 24.04.2023).
2. Движки для мобильных игр. URL: <https://asomobile.net/blog/20-luchshih-dvizhkov-dlya-mobilnyh-igr-na-ios-i-android-platformy-i-instrumenty-dlya-razrabotki-v-2021-godu/> (дата обращения; 24.04.2023).
3. Движки для мобильных приложений. URL: <https://artjoker.ua/ru/blog/luchshie-instrumenty-dlya-sozdaniya-mobilnykh-prilozheniy/> (дата обращения; 24.04.2023).

# ВОЕННЫЙ УЧЕБНЫЙ ЦЕНТР

УДК 629.76.04

**Д. В. Анохин**

студент военного учебного центра

**А. А. Латровкин** – начальник кафедры военного учебного центра – научный руководитель

## ПРОБЛЕМА ФИЗИЧЕСКОЙ ЗАЩИТЫ КАБЕЛЬНОЙ СЕТИ АВТОНОМНОГО ИСТОЧНИКА ПИТАНИЯ

Предлагается техническое решение, позволяющее уменьшить количество неисправностей на кабельной сети на системах автономного электроснабжения (САЭ) в автономном источнике питания ДГ-30М. Проведён анализ статистики неисправностей на САЭ, предложен вариант защиты кабельной сети проходящей вдоль рамы дизельного генератора (ДГ) в виде защитного кабельного канала, построены 3D модели ДГ-30М и защитного кабельного канала, проведено компьютерное моделирование напряженно-деформированного состояния защитного кабельного канала.

### Актуальность

В настоящее время существует проблема, связанная с эксплуатацией ДГ-30М, во время проведения технического обслуживания, по технологическим картам, работам по устранению неисправностей на ДГ-30М, при снятии капота. Персонал при проведении работ наступает на кабельную сеть, расположенную вдоль рамы ДГ-30М, что приводит к повреждению оплётки, короткому замыканию и разрыву кабелей, возможны проливы технических жидкостей при проведении работ, которые разрушающе воздействуют на изоляцию кабельной сети. Также при установке капота после проведённых работ есть вероятность неправильной установки (опускания) капота на кабельную сеть, что может привести к её повреждению.

В табл. 1 представлена статистика неисправностей на системах автономного электроснабжения (САЭ) с октября 2020 по октябрь 2022 г.

Таблица 1

Наименование	2020-2021	2022	Итого
Неисправности на САЭ:	186	179	365
из них на ДГ:	49	52	101
из них на кабельной сети:	4	6	10

Защита от повреждения кабельной сети не была предусмотрена заводом-изготовителем. При отсутствии модернизаций и доработок возможен выход из строя ДГ-30М, что может привести к аварийным ситуациям. В ходе реализации работы предлагается решение поставленной проблемы путём изготовления защитного кабельного канала для кабельной сети ДГ-30М.

**Цель работы:** усовершенствование конструкции ДГ-30М для снижения количества возникновения неисправностей на САЭ.

#### Задачи работы:

1. Предложить форму защитной конструкции кабельной сети.
2. Подобрать геометрические параметры и материал изготовления защитной конструкции кабельной сети.
3. Провести компьютерное моделирование напряженно-деформированного состояния защитной конструкции кабельной сети из предложенных материалов.

**Объект исследования:** кабельная сеть ДГ-30М.

**Предмет исследования:** защитное приспособление для кабельной сети ДГ-30М.

**Обзор предметной области**



Автономный источник питания (АИП) – это комплекс технических средств, для обеспечения объекта электроэнергией без его подключения к централизованным электросетям. Автономный источник питания ДГ-30М предназначен для питания потребителей электроэнергией переменного трёхфазного тока напряжением 380В, частотой 50Гц по трёхпроводной схеме, а также собственных потребителей АИП-30М электроэнергией переменного трёхфазного тока напряжением 380/220В, частотой 50Гц по четырёхпроводной схеме с выведенной изолированной нейтралью. Номинальная мощность АИП-30М 30кВт.

**Состав АИП-30М:**

1. Капот типаУ175М С672 – 1 шт.
2. Компьютерная система управления АИП-30М – 1 шт.
3. Электроагрегат, в том числе:
  - блок с аппаратурой – 1 шт.;
  - дизель-генератор ДГ-30-Т400-3Р – 1 шт., в том числе:
    - а) дизель Д-246.1М – 1 шт.;
    - б) генератор 1 ГС-30-БП-ЗУ-КМ – 1 шт.;
4. Комплект ЗИП-О – 1 комплект.

Технические характеристики АИП-30М предоставлены в табл. 2.

Таблица 2

Наименование параметра	Норма	
Род тока	Переменный трёхфазный	
Номинальная мощность, кВт	30	
Номинальное напряжение на выходе электроагрегата, В	400	
Номинальная частота, Гц	50	
Номинальный ток нагрузки, А	54	
Удельный расход топлива при номинальной мощности, г/кВт*ч	220 <sup>+11</sup>	
Габаритные размеры, мм, не более	длина	3060
	ширина	885
	высота	1170
Масса, кг, не более	1300	

В настоящее время существует проблема, связанная с эксплуатацией АИП, заключается в том, что во время проведения технического обслуживания, в соответствии по технологическим картам снимается капот АИП, персонал при проведении работ наступает на кабельную сеть (рис. 2, 3), что приводит к внутреннему повреждению оплётки кабелей, короткому замыканию и разрыву. Также при установке капота после проведённых работ есть вероятность неправильной установки (опускания) капота на кабельную сеть, что может привести к их повреждению.

Согласно данным за последние 2 года, из 365 неисправностей на САЭ было 101 неисправностей, связанных непосредственно с ДГ, из которых 10 приходится на кабельную сеть представлены в табл. 1. Возможные места повреждения кабельной сети показаны на рис. 1.



Кабельная сеть. Возможные места повреждения

Рис. 1. ДГ-30М без капота

В рамках *первой* задачи была построена 3D модели ДГ-30М с предлагаемым защитным кабельным каналом в системе трехмерного проектирования КОМПАС-3D, представлены на рис. 2 и 3.

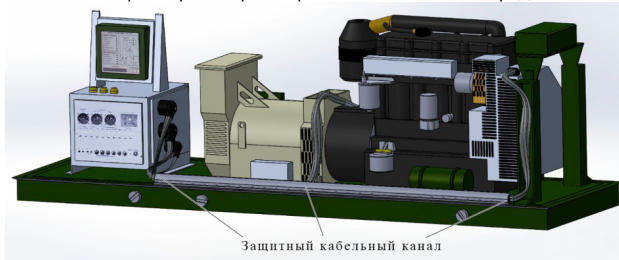


Рис. 2. 3D-модель ДГ-30М с защитным кабельным каналом

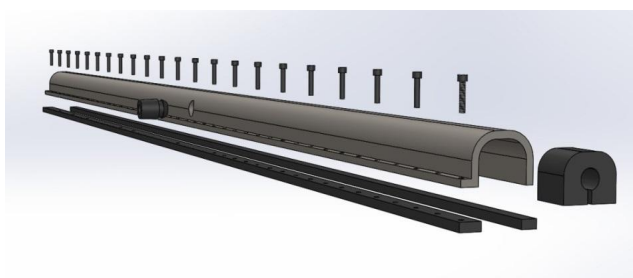


Рис. 3. 3D-модель защитного кабельного канала

В рамках *второй* задачи исходя из протяженности и максимального диаметра кабельной сети предложены геометрические параметры и материал изготовления защитного кабельного канала предоставлены в табл. 3.

Таблица 3

Объект	Геометрические параметры	Значения, мм
Защитный кабельный канал кабельной сети ДГ-30М	длина	1560
	высота	53
	ширина	83
	толщина	8

**1 этап** – выбор материалов для элементов сборки.

Выбор материала изготовления защитного кабельного канала его формы и толщины влияют на механические свойства предмета исследования. Предлагается провести расчёты из таких материалов, как пластик ABS, ПВХ, сплав алюминия 1060, оцинкованная сталь.

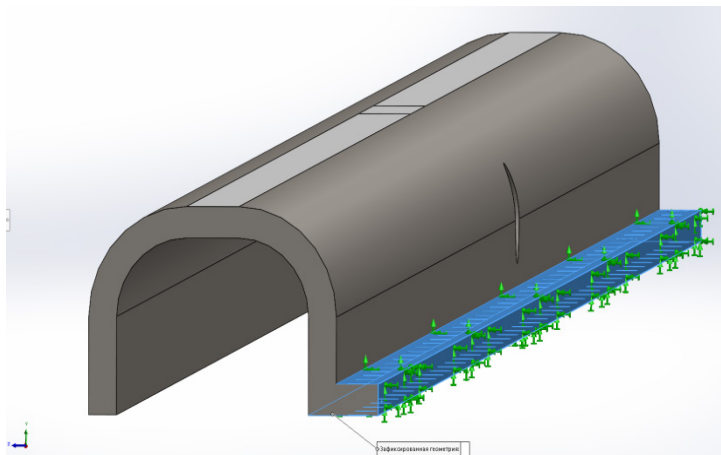
Данное исследование основывается на механических свойствах, указанных в табл. 4.

Таблица 4

Материал	Пластик ABS	ПВХ	Алюминий 1060 Сплав	Оцинкованная сталь
Модуль упругости, Н/м <sup>2</sup>	2e+9	2,41e+9	6,9e+10	2e+11
Коэффициент Пуассона	0,394	0,3825	0,33	0,29

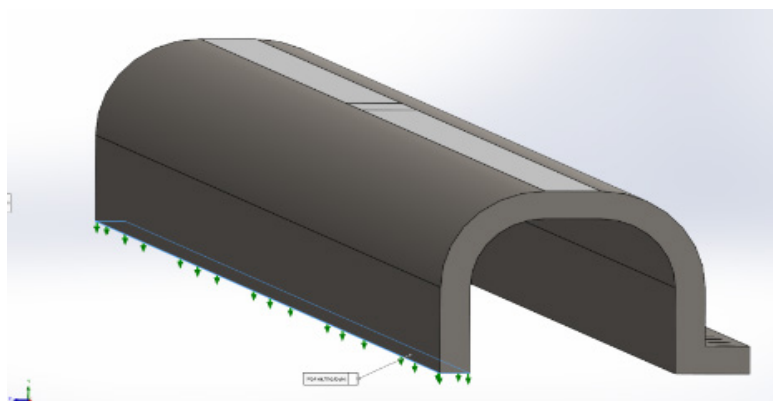
**2 этап** – выбор мест и типов креплений.

Тип крепления – зафиксированная геометрия. Данный тип крепления предполагает, что грани намертво зафиксированы и не смещаются в процессе анализа (рис. 4).



*Рис. 4. Место крепления, зафиксированная геометрия*

Тип крепления – ролик/ползун. Данный тип крепления предполагает, что грани зафиксированы и не смещаются в процессе анализа во всех плоскостях, кроме одной, в нашем случае параллельной горизонту (рис. 5).



*Рис. 5. Место крепления, ролик/ползун*

**3 этап** – выбор типа нагрузки.

Тип нагрузки – Сила. Прилагаемая нагрузка – 150 кгс, что эквивалентно давлению, оказываемому одним человеком поперёк защитного кабельного канала. В исследовании нагрузка прикладывалась вертикально вниз на площадь равную 50 см<sup>2</sup>, в самом уязвимом месте кабельного канала – над местом вывода бокового кабеля (рис. 6).

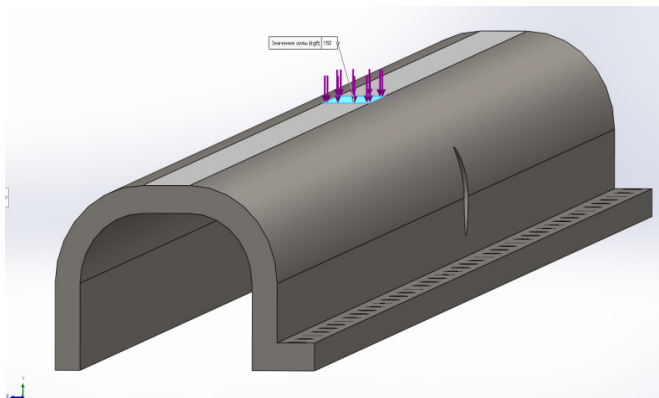


Рис. 6. Место нагрузки

В рамках *третьей* задачи проведено компьютерное моделирование напряженно-деформированного состояния защитного кабельного канала, выполненного из предложенных материалов. Используется статическое исследование, следовательно, упругого последействия нет. Материалы модели – однородные, изотропные, сплошные.

Был проведён анализ предлагаемого защитного кабельного канала из пластика ABS, ПВХ, сплава алюминия 1060, оцинкованной стали на напряжение и результирующее перемещение. Результаты исследования, а также свойства материалов, сведены в табл. 5.

Таблица 5

Материал	ABS	ПВХ	Алюминий 1060 Сплав	Оцинкованная сталь
Предел прочности при растяжении, Н/м <sup>2</sup>	3e+07	4,07e+07	6,89356e+07	3,56901e+08
Модуль упругости, Н/м <sup>2</sup>	2e+09	2,41e+09	6,9e+10	2e+11
Коэффициент Пуассона	0,394	0,3825	0,33	0,29
Массовая плотность, кг/м <sup>3</sup>	1 020	1 300	2 700	7 870
Модуль сдвига, Н/м <sup>2</sup>	3,189e+08	8,667e+08	2,7e+10	–
Напряжение VonMises, кгс/см <sup>2</sup>	89,210	89,876	92,607	94,442
Результирующее перемещение, мм	0,507	0,425	0,015	0,005
Объем изделия, м <sup>3</sup>	0,00262662			
Плотность кг/м <sup>3</sup>	1 020	1 300	2 700	7 870
Масса, кг	2,67915	3,4146	7,09187	20,6715
Цена за 1 кг, руб.	от 631	от 85	от 345	от 80
Стоимость материала изделия, руб.	от 1690,54	от 290,24	от 2446,7	от 1653,72

Задав форму профиля – полусфера, производим расчёт коэффициентов модели защитного кабельного канала в рис. 4.

В результате проведенного исследования для создания защитного кабельного канала по своим механическим защитным свойствам, при выбранной толщине канала, подходят все анализируемые материалы. Но стоит учесть, что материалы ABS и ПВХ подвержены более сильной деформации в срав-

нении с металлами, в связи с чем предлагается использовать оцинкованную сталь или алюминиевый сплав толщиной 8 мм.

Из табл. 4 видно, что оцинкованная сталь является наиболее дешёвым металлом, а алюминиевый сплав – наиболее легким.

Представлены стоимости материалов изделия без учёта затрат на изготовление защитного кабельного канала.

### **Выводы**

В ходе работы предложен вариант усовершенствования конструкции для увеличения износостойкости от внешней нагрузки и защиты кабельной сети ДГ-30М от пролива технических жидкостей. С помощью современных материалов и средств визуализации были построены 3D модели ДГ-30М и защитного кабельного канала, было проведено моделирование напряженно-деформированного состояния защитного кабельного канала.

Проведение доработки с установкой защитного кабельного канала на ДГ-30М позволит уменьшить количество неисправностей на САЭ.

В дальнейшей работе планируется изучить воздействие технических жидкостей, используемых в ДГ-30М на защитный кабельный канал.

### **Библиографический список**

1. Руководство по эксплуатации Электроагрегат ДГ-30М. М. 2015. 143 с.

УДК 004.9; 621.317.08

**Н. А. Арефьев**

студент студент кафедры эксплуатации и управления аэрокосмическими системами

**Д. Е. Селедкин**

студент кафедры аэрокосмических компьютерных и программных систем

**В. Д. Вахабов**

студент кафедры прикладной информатики в инновационной деятельности

**А. А. Парфенов**

студент кафедры прикладной информатики в инновационной деятельности

**В. В. Котович** – кандидат технических наук, доцент военного учебного центра – научный руководитель

### АППАРАТНАЯ ЧАСТЬ ОБУЧАЮЩЕГО ПРОГРАММНОГО КОМПЛЕКСА

С целью формирования у студентов-метрологов необходимых знаний и навыков в вопросах цифровой обработки и анализа электрических сигналов в военном учебном центре при ГУАП разрабатывается обучающий программный комплекс для изучения параметров детерминированных сигналов [1]. Комплекс включает в себя обучающую программу и аппаратную часть, которая включает в себя преобразователь аналогового сигнала в цифровой код (адаптер), цифровой генератор специальной формы, преобразователь интерфейсов UART/USB, UART/FT-232 и устройство отображения информации (жидкокристаллический дисплей). Комплекс позволит студентам исследовать частотные и временные параметры как виртуальных детерминированных сигналов различных форм, формируемых с использованием математических функций обучающей программой, так и сигналов, поступивших от внешних аналоговых источников (измерительных генераторов, блоков питания) и преобразованных в аппаратной части комплекса в цифровой код.

Адаптер на основе двух микроконтроллеров имеет два основных режима работы:

1. Прием и оцифровка внешнего сигнала.

2. Генерация собственного сигнала с возможностью его настройки.

Первый основной режим внешнего сигнала делится на два этапа:

1. Нормирование входного сигнала с целью обеспечения работы АЦП (аналого-цифрового преобразователя) в рабочем диапазоне от 0 до 5 вольт.

2. Оцифровка сигнала с помощью АЦП.

В качестве основного рабочего микроконтроллера используется микроконтроллер ATmega16, имеющий встроенный 10 битный АЦП. Допустимый диапазон тактовой частоты АЦП микроконтроллера  $F_{ADC}$  составляет 50–200 кГц, количество тактов, требуемое для преобразования  $N = 13$ . Максимальная частота преобразований АЦП вычисляется по формуле:

$$F_{max} = \frac{F_{ADC}}{N} = \frac{200000 \text{ Гц}}{13} \cong 15384 \text{ Гц.} \quad (1)$$

Результат преобразования АЦП записывается в регистры ADCH, ADCL микроконтроллера. Объединение этих регистров дает двухбайтовое число  $V_{digital}$  в диапазоне от 0 до 1023 ( $2^{10} - 1$ ). Для получения итогового значения напряжения используются формулы:

$$V_{digital} = ADCH * 256 + ADCL, \quad (2)$$

$$V = \frac{V_{digital}}{1023} * 5 \text{ В.} \quad (3)$$

Формула (3) используется для вывода значения напряжения на дисплей. Для передачи данных на компьютер используется значение  $V_{digital}$  вычисленное по формуле (2).

Значение напряжения  $V_{digital}$  можно преобразовать из 10 в 8-ми битовое значение, отбросив два младших бита и получившееся, таким образом, значение можно вывести на ножки порта В микроконтроллера. Это значение с при помощи 8-ми битового ЦАП (цифро-аналоговый преобразователь) можно преобразовать в сигнал, который будет можно считать с помощью осциллографа.

Для передачи данных из адаптера на компьютер с запущенной программой «Обучающая программа для изучения параметров детерминированных сигналов» используется протокол UART (Universal Asynchronous Receiver-Transmitter). Со стороны компьютера к порту USB подключен адаптер FT232RL, который создает виртуальный COM-порт на компьютере для передачи данных.

Для передачи данных был разработан протокол, схема которого приведена на рис. 1.

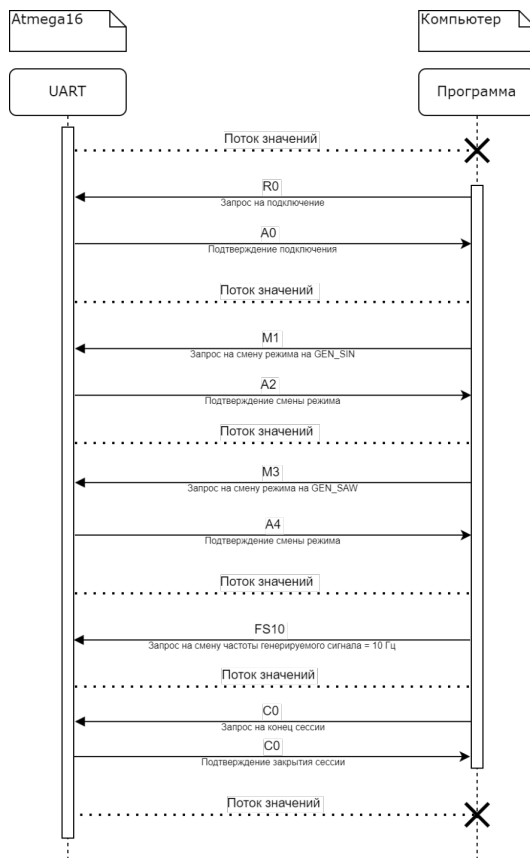


Рис. 1. Схема передачи данных

Данные передаются построчно, коды символов соответствуют таблице ASCII, конец строки – символ  $\backslash n$  (номер в таблице ASCII 10 – 0xA). Поток значений – строки со значениями напряжения  $V_{digital}$  в момент отправки данных. Запросы на начало соединения и конец нужны, чтобы микроконтроллер понимал, что компьютер подключен и программа начала прием и запись данных.

Микроконтроллер имеет 6 режимов работы, их список приведен в табл. 1.

Список сигналов

Режим	Описание
RECORDING	Чтение внешнего сигнала
GEN. SIN	Генерация синусоидального сигнала
GEN. SAW	Генерация пилообразного сигнала
GEN. TRIANGLE	Генерация треугольного сигнала
GEN. RECTANGLE	Генерация прямоугольного сигнала
GEN. TRAPEZOID	Генерация трапециевидного сигнала

С помощью прерывания по таймеру, микроконтроллер изменяет переменную  $x$  от 0 до 1000 за  $1 \times F_{ген}$  секунду. Меняя настройки таймера можно задавать частоту  $F_{ген}$  сигнала.

Изображение синусоидального сигнала, генерируемого по формуле (4) показано на рис. 2:

$$y = 512 \cdot \sin\left(\frac{x \cdot 2\pi}{1000}\right). \tag{4}$$

Изображение сигнала-пилы, генерируемого по формуле (5) показано на рис. 3:

$$y = \frac{1023 \cdot x}{1000} \% 1023, \tag{5}$$

где % – операция деление по модулю. Результатом этой операции является остаток от целочисленного деления

Изображение сигнала-треугольника, генерируемого по формуле (6) показано на рис. 4:

$$\begin{cases} y = 2 \cdot \left(\frac{1023x}{1000}\right) \% 1023; x \leq 500 \\ y = -2 \cdot \left(\frac{1023x}{1000}\right) \% 1023; x > 500 \end{cases} \tag{6}$$

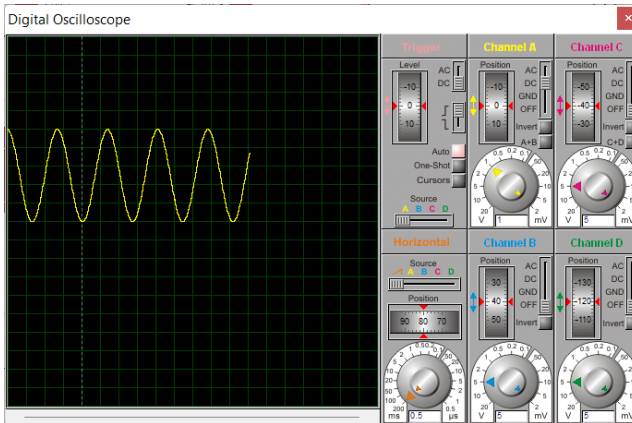


Рис. 2. Вывод в режиме GEN. SIN



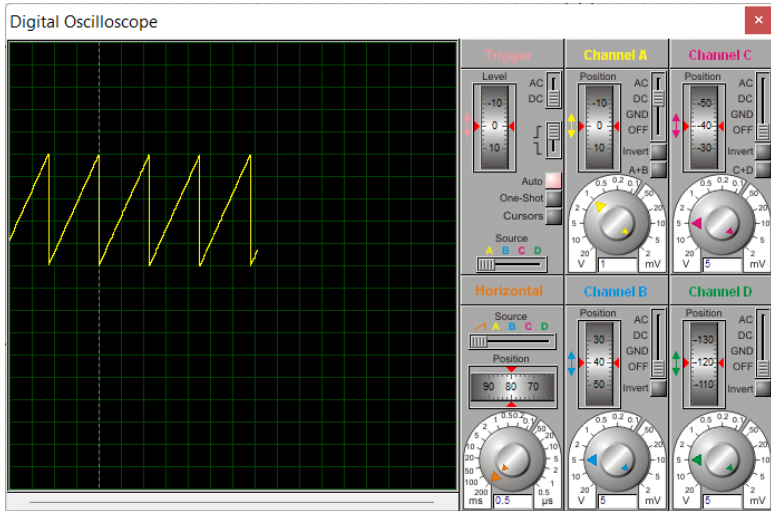


Рис. 3. Вывод в режиме GEN. SAW

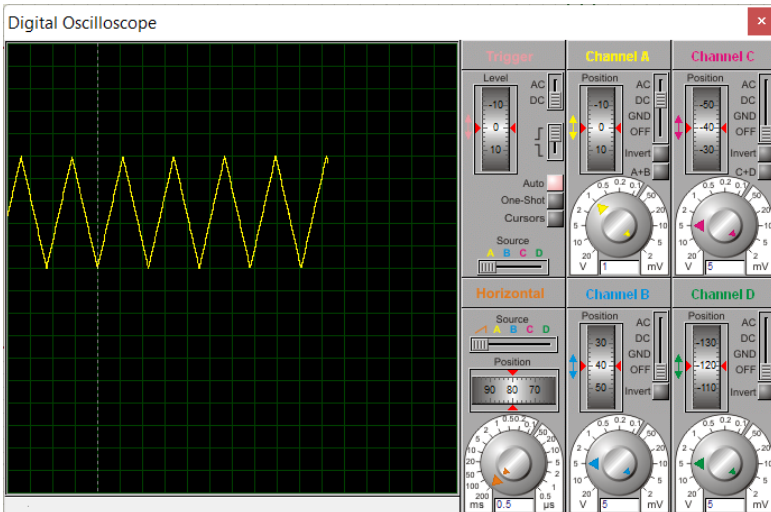


Рис. 4. Вывод в режиме GEN. TRIANGLE

Изображение сигнала-прямоугольника, генерируемого по формуле (7) показано на рис. 5:

$$\begin{cases} y = 0; x < 500 \\ y = 1023; x > 500 \end{cases} \quad (7)$$

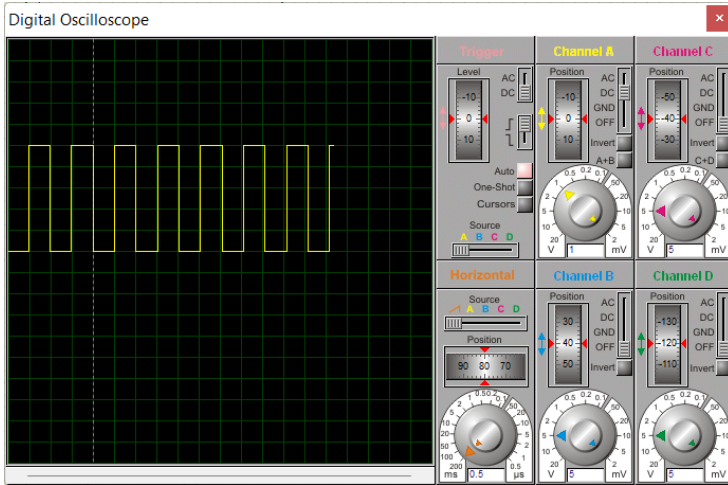


Рис. 5. Вывод в режиме GEN. RECTANGLE

Изображение сигнала-трапеции, генерируемого по формуле (8) показано на рис. 6:

$$\begin{cases} y = 5 * \left( \frac{1023x}{1000} \right) \% 1023; x < 500 \\ y = -5 * \left( \frac{1023x}{1000} \right) \% 1023; x > 500 \end{cases}$$

(8)

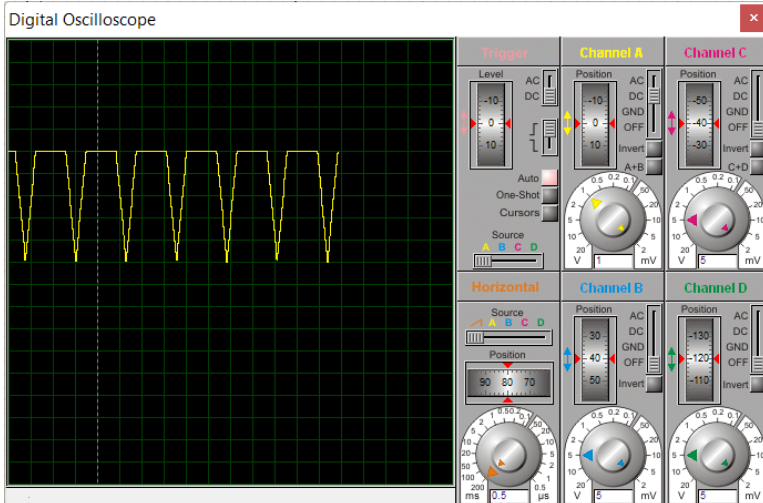


Рис. 6. Вывод в режиме GEN. TRAPEZOID

Для управления основным рабочим микроконтроллером адаптера к UART соединению (UART микроконтроллера – FT232RL) параллельно подключается дополнительный микроконтроллер – управляющий микроконтроллер ATmega16 (рис. 7), который перехватывает передаваемые данные, и отправляет свои данные, по тому же протоколу.

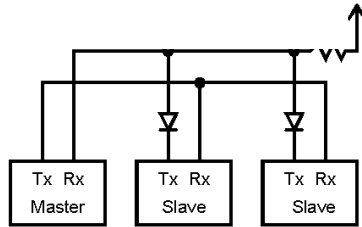


Рис. 7. Схема подключения нескольких интерфейсов UART

Соединение работает по принципу Multi-Drop, где роль главного интерфейса (Master) выполняет адаптер, а роль подчиненных (Slave) USB интерфейс компьютера и UART управляющего микроконтроллера. Ножки TxD (Transmit Data) и приема данных RxD (Receive Data) подчинённых интерфейсов, которые в обычном режиме имеют напряжение 5В, изолируются с помощью диодов, чтобы на RXD главного микроконтроллера сохранялось напряжение он линия RXD-TXD запитывается отдельно, через резистор номиналом 10 кОм. Таким образом два интерфейса TXD не создают помех друг для друга своим выходным напряжением.

К управляющему микроконтроллеру подключены три кнопки, и LCD (Liquid-crystal display) дисплей для вывода информации и меню. Модель схемы управляющего микроконтроллера с подключенным LCD дисплеем и кнопками выполнена в редакторе Proteus и представлена на рис. 8.

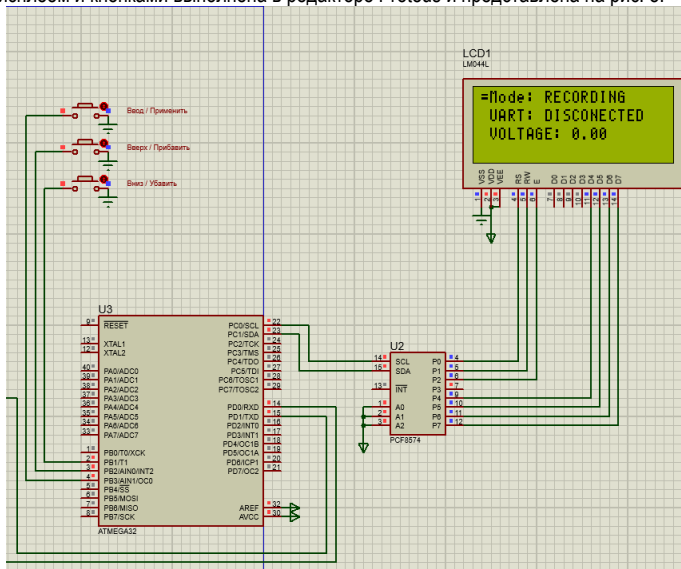


Рис. 8. Схема управляющего микроконтроллера

Для сокращения количества задействованных портов МК, отправка данных на LCD осуществляется через интерфейс I2C.

Полная схема модели адаптера, сделанная в программе Proteus, показана на рис. 9.

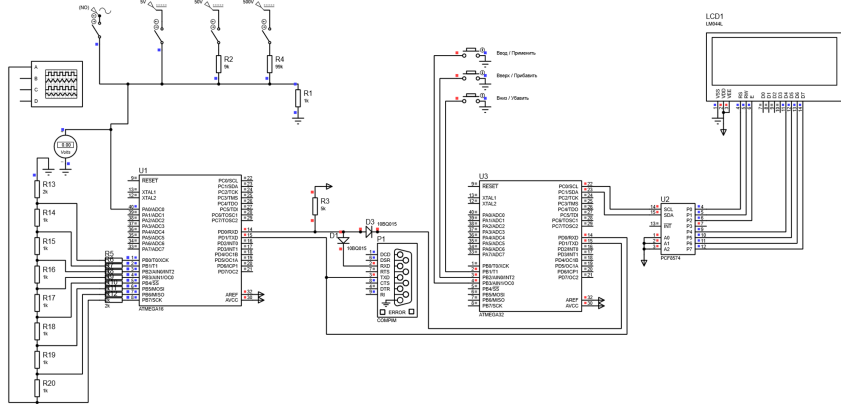


Рис. 9. Схема адаптера

Разработка аппаратной части обучающего программного комплекса является сложной технической задачей и в настоящий момент ещё не закончена. Готовый комплекс позволит автоматизировать процесс обучения студентов, которые интересуются измерением и преобразованием детерминированных сигналов различной формы. Работа аппаратной части подтверждается выводом сигнала

### Библиографический список

1. Котвич В. В. Автоматизация процесса изучения параметров детерминированных сигналов. Радиотехнические, оптические и биотехнические системы. Устройства и методы обработки информации // Третья Всероссийская научная конференция: сб. докладов. 14-22 апреля 2022 г. СПб. 2022. 167 с.

УДК 621.317.757

**Р. А. Ахмадов**

студент военного учебного центра

**В. В. Котович** – кандидат технических наук, доцент учебного центра – научный руководитель

### ФАЗОВРАЩАТЕЛЬ НА ОСНОВЕ МИКРОКОНТРОЛЛЕРА

В статье рассмотрен принцип сдвига фазы аналогового сигнала и принцип измерения сдвига фазы в диапазоне от 1 до 10 кГц. Предложена структурная схема измерителя фазы с фазовращателем и его компьютерная модель.

Фазовращатели широко применяются как в составе разнообразных устройств, так и самостоятельно. Например, фазовращатель входит в состав компенсационных измерителей сдвига фазы, согласующих устройств и т. д. Также фазовращатели используются в учебных целях, например, при проведении лабораторных и практических работах по исследованию параметров сигналов.

В основе предложенного устройства для сдвига фазы используется метод фазового фильтра. Фазовый фильтр 1-го порядка (рис. 1) позволяет посредством изменения значения сопротивления  $R$  установить необходимую величину фазового сдвига в диапазоне от 0 до 180°.

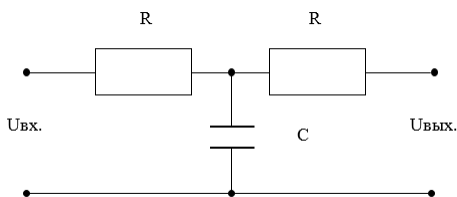


Рис. 1. Пассивный фазовый фильтр 1-го порядка

В такой схеме значение уровня выходного сигнала зависит от устанавливаемого значения сопротивления  $R$ . Для того чтобы уровень сигнала не изменялся в процессе изменения угла сдвига фазы (при изменении  $R$ ) в цепь следует добавить операционный усилитель (рис. 2). Такая схема позволяет с помощью резисторов  $R_2$  и  $R_3$  установить коэффициент передачи близким к единице [1].

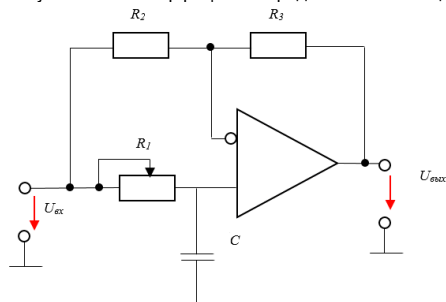


Рис. 2. Активный фазовый фильтр 1-го порядка

Величина фазового сдвига, при использовании такого фильтра, рассчитывается по формуле:

$$\varphi = 2 \cdot \arctg(2\pi \cdot \omega \cdot R_1 \cdot C), \quad (1)$$

где  $\omega$  – частота входного сигнала, рад/с;  $R_1$  – переменный резистор, Ом;  $C$  – емкость, Ф.

Предлагаемый фазовращатель позволяет измерить сдвиг фазы от 1 до 10 кГц. В качестве метода измерения сдвига фазы используется метод дискретного счета. Суть которого заключается в преобразовании количества тактовых импульсов между сигналами (исходным и сдвинутым по фазе) во время и расчет полученного значения по формуле 2.

$$\varphi = \frac{\tau}{T} * 360^{\circ}, \quad (2)$$

где  $\tau$  – время задержки одного сигнала относительно другого, с;  $T$  – период следования сигнала, с.

Предлагаемый фазовращатель позволяет в диапазоне частот от 1 Гц до 1 МГц обеспечить сдвиг фазы от 0 до 180 градусов, а также может измерить величину сдвига фазы сигналов в диапазоне от 1 Гц до 10 кГц.

Структурная схема предлагаемого устройства представлен на рис. 3.

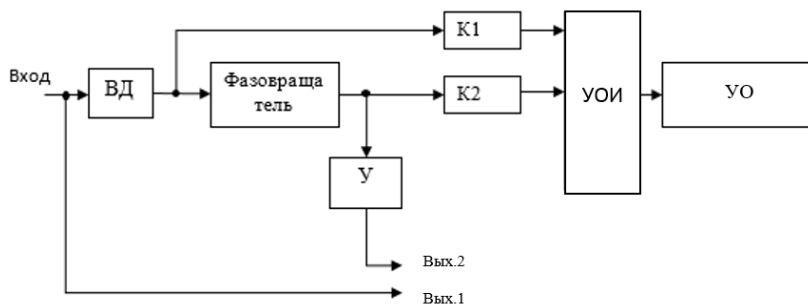


Рис. 3. Структурная схема с фазовращателя

В состав предлагаемого фазовращателя входят: входной делитель напряжения (ВД) с коэффициентом деления  $K$ ; фазовращатель на основе фазового фильтра; компараторы ( $K1$  и  $K2$ ) на основе операционных усилителей; усилитель ( $У$ ) с коэффициентом усиления  $K$ ; устройство обработки информации (УОИ) на основе микроконтроллера (МК); устройство отображения (УО) в виде жидкокристаллического (LCD) дисплея.

На вход микроконтроллера через компараторы  $K1$  и  $K2$  поступают два сигнала. Первый сигнал – исходный сигнал. Он поступает с входа фазовращателя. Второй сигнал – сдвинутый сигнал. Он поступает с выхода фазовращателя. В микроконтроллер с помощью алгоритма реализуется метод дискретного счета и осуществляется вычисление сдвига фазы между двумя сигналами [2]. Затем результат вычисления выводится на жидкокристаллический LCD дисплей.

Для защиты микроконтроллера и устройства в целом от выхода из строя, в случае превышения предельно допустимого входного напряжения, на входе устройства находится делитель напряжения с коэффициентом деления  $K=10$ , что позволяет расширить диапазон входного напряжения до 50 В.

Для реализации метода дискретного счета входные синусоидальные сигналы, поступающие на микроконтроллер, преобразовываются в импульсные с помощью компараторов  $K1$  и  $K2$ .

В состав фазовращателя на основе фазового фильтра, входит блок конденсаторов (БК) с дискретным переключателем. С его помощью осуществляется выбор нужного конденсатора, что в свою очередь обеспечивает выбор необходимо диапазона частот входного сигнала от 1 Гц до 1 МГц.

Предлагаемое устройство сдвига фазы имеет два выхода (Вых.1 и Вых. 2) (см. рис. 3). На первый выход поступает исходный сигнал, на второй выход поступает сигнал – сдвинутый по фазе относительно от первого.

Поскольку на входе устройства стоит делитель напряжения с коэффициентом деления  $K$  то, перед вторым выходом находится усилитель с коэффициентом усиления  $K$  для восстановления исходного уровня сигнала.

Моделирование работы устройства осуществлялось в системе автоматизированного проектирования (САПР) Proteus Professional. Схема модели представлена на рис. 4.

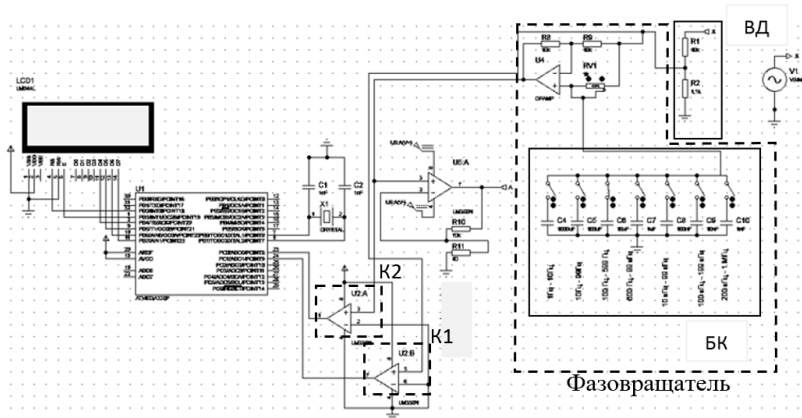


Рис. 4. Модель фазовращателя в Proteus

Модель фазовращателя в Proteus (рис. 4) работает следующим образом:

- 1) на блоке конденсаторов (БК) вручную устанавливается один из переключателей в положение «вкл.» для подключения конденсатора номинальное значение которого зависит от частоты входного сигнала;
- 2) на вход подается синусоидальный сигнал от генератора V1 с частотой в диапазоне от 1 до 10 МГц и с напряжением от 0,01 до 50 В. Поступая на делитель блока ВД, входной сигнал ослабляется в 10 раз и разделяется на два одинаковых сигнала. Один из них поступает на компаратор К1 где из синусоидального преобразуется в импульсный, а другой сигнал поступает на фазовращатель;
- 3) с помощью переменного резистора RV1 устанавливается нужный сдвиг фазы (от 0 до 180 градусов) для второго сигнала;
- 4) сдвинутый по фазе сигнал поступает на компаратор К2, где из синусоидального преобразуется в импульсный;
- 5) микроконтроллер осуществляет расчет временной задержки этих двух сигналов между собой и преобразование полученного значения в значение сдвига фазы в градусах. Полученное значение МК передает на экран LCD1.

Часть измерений, полученные в ходе моделирования, были сведены в табл. 1.

В табл. 1 в столбце 5 записаны результат измерений угла сдвига фазы с использованием встроенного в Proteus осциллографа. В столбце 4 записаны значения сдвига фазы, полученные расчетным путем по формуле 1. В столбце 6 записана погрешность измерения с помощью встроенного осциллографа. В столбце 7 записаны значения сдвига фазы, измеренные с помощью микроконтроллера в градусах и выводимые на экран LCD1. В столбце 8 записаны значения погрешности измерений микроконтроллера. Эти погрешности определяется дискретностью периода счетных импульсов и длительности обработки команд микроконтроллера. Погрешность дискретности связана с тем, что интервал времени можно измерить с точностью до одного периода счетных импульсов, т.е. точность измерений зависит от тактовой частоты микроконтроллера. Погрешность, вызванная длительностью обработки команд, означает, что микроконтроллер тратит время в процедуре обработки прерывания на расчет времени между сигналами в ms, расчет периода следования сигналов и вычисление сдвига фазы в градусах.

Погрешность в диапазоне частот от 1 Гц до 9 кГц не превышает трех градусов.

Результаты моделирования

F, Гц	R, Ом	C, Ф	Сдвиг фазы, полученный расчетным путем	Сдвиг фазы, определенный по осциллографу	Погрешность	Сдвиг фазы, опред. методом дискр. счета	Погрешность
1	2	3	4	5	6	7	8
1	1	1000мк	0,719626	0	0,719626	0	0,719626
1	10	1000мк	7,186912	7,187	-8,8E-05	7,2	-0,01309
1	100	1000мк	64,25764	64,943	-0,68536	64,4	-0,14236
1	250	1000мк	115,0104	116,951	-1,9406	115,2	-0,1896
1	500	1000мк	144,6696	145,045	-0,3754	144,9	-0,2304
1	1000	1000мк	161,9049	163,012	-1,1071	162,1	-0,1951
1	10000	1000мк	178,1754	178,366	-0,1906	177,2	0,9754
10	1	100мк	0,719626	0	0,719626	0,2	0,519626
10	10	100мк	7,186912	7,31	-0,12309	6,1	1,086912
10	100	100мк	64,25764	65,466	-1,20836	63,2	1,05764
10	250	100мк	115,0104	115,20	-0,1896	114,5	0,5104
10	500	100мк	144,6696	144,36	0,3096	145,7	-1,0304
10	1000	100мк	161,9049	162,36	-0,4551	161,9	0,0049
10	10000	100мк	178,1754	177,05	1,1254	176,8	1,3754
1к	1	1мк	0,719626	0	0,719626	0	0,719626
1к	10	1мк	7,186912	5,88	1,306912	6,8	0,386912
1к	100	1мк	64,25764	64,68	-0,42236	66,7	-2,44236
1к	250	1мк	115,0104	114,99	0,0204	115,5	-0,4896
1к	500	1мк	144,6696	143,73	0,9396	142,6	2,0696
1к	1000	1мк	161,9049	162,03	-0,1251	162,5	-0,5951
1к	10000	1мк	178,1754	177,05	1,1254	176,6	1,5754
10к	1	100н	0,719626	0	0,719626	0,7	2,02725
10к	10	100н	7,186912	7,05	0,136912	7,2	0,01095
10к	100	100н	64,25764	65,33	-1,07236	72	-7,742236

При оценке абсолютной методической погрешности предлагаемого устройства сдвига фазы было установлено, что разработанное устройство обладает точностью достаточной для использования в учебных целях в диапазоне частот от 1 Гц до 1 МГц.

Недостатком данного устройства является то, что оно может выводить на дисплей значения сдвига фазы в диапазоне частот от 1 Гц до 10 кГц, что обусловлено низкой тактовой частотой микроконтроллера, использованного в предложенной модели. В предлагаемом устройстве предполагается использовать микроконтроллеры из семейства AVR – Atmega, которые отличаются высокой производительностью, меньшим потреблением энергии, доступностью, и главное сравнительной дешевизной, но работающие на низких частотах. При необходимости измерять сдвиг фаз в диапазоне частот от 1 Гц до 1 МГц следует заменить на микроконтроллер серии Atmega на STM.



Разработанное устройство обладает рядом достоинств таких как, простота и дешевизна конструкции, малые массогабаритные параметры и сравнительно высокая точность. Предлагаемый фазовращатель можно будет использовать в учебном процессе при подготовке специалистов-метрологов.

**Библиографический список**

1. Алексаков Г. Н. Активные RC-фильтры на операционных усилителях. Энергия, 1974. 66 с.
2. Евстифеев А. В. Микроконтроллеров AVR семейств Т1ру и Мега фирмы АТМЕL. Додэка-XXI, 2008. 558 с.

УДК 681.5: 53.08

**А. П. Волкова**

студент военного учебного центра

**В. В. Котович** – кандидат технических наук – научный руководитель

### РАЗРАБОТКА ДАТЧИКА УГЛА НА ОСНОВЕ АКСЕЛЕРОМЕТРА

Для производства во многих областях как в гражданской, так и в военной областях предприятия следуют нормальным условиям при проверке, установке и регулировке приборов. В данной работе рассмотрен вариант модернизации оборудования, применяемого для установки необходимого положения устройства. Предлагается бюджетный и мобильный вариант датчика, который позволяет определять угол тангажа и крена положения прибора.

Технические характеристики данного устройства:

- разрабатываемое устройство обеспечивает точность измерений до 6 угловых минут;
- измерения проводятся в автоматизированном режиме под управлением ЭВМ (электронной вычислительной машиной) или под управлением микроконтроллера без ЭВМ.

Данный прибор основан на МЭМС-акселерометре, который позволяет уменьшить габариты, а также способен измерять несколько видов напряжения, что позволяет обеспечить необходимую точность измерения. Принцип работы акселерометра заключается в воздействии измеряемого углового ускорения на чувствительный элемент, возникает момент инерции  $M$ , который вместе с моментом обратной связи  $M_{oc}$  образуют сигнал  $\Delta M$ , вызывающий перемещение подвижного узла на угол  $\alpha$ . ПУ перемещается относительно обкладок измерительных емкостей датчика перемещений. В результате изменения величины зазора между обкладками происходит изменение емкостей на выходе мостовой схемы, в которую они включены, возникает напряжение  $U_{дп}$ , которое затем поступает на усилитель, преобразующий его в выходное напряжение датчика  $U_{вых}$ , сигнал которого используется в обратной связи.



Рис. 1. Структурная схема акселерометра

Конструктивно данный прибор состоит из трех блоков: блок датчика, в который входит акселерометр и аналого-цифровой преобразователь, блок обработки информации, состоящий из микроконтроллера и блок вывода информации представляет собой LCD дисплей, и интерфейс UART.

При изменении положения объекта акселерометр измеряет значения углов наклона (по оси X и Y) и передает их в виде аналогового сигнала на 16-битный АЦП. Преобразованные значения напряжения с помощью SPI интерфейса с АЦП отправляются на микроконтроллер ATmega32PU. Он принимает код от АЦП и преобразовывает его в градусы. Затем с микроконтроллера значения углов передаются на LCD дисплей и по интерфейсу UART на компьютер, где происходит её дальнейшая обработка.

Алгоритм работы устройства:

1. Каждую секунду микроконтроллер опрашивает 16-ти разрядный АЦП по двум каналам, и получает преобразованные значения напряжения, соответствующие углам тангажа и крена. После заполнения буфера памяти результатами опросов микроконтроллер усредняет полученные значения.
2. Усредненный результат измерения выходного напряжения акселерометра преобразуется в код и передаётся по интерфейсу UART на компьютер для дальнейшей обработки.

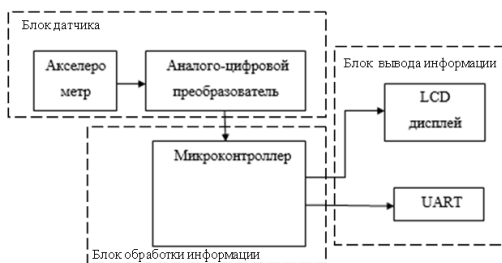


Рис. 2. Структурная схема датчика угла



Рис. 3. Алгоритм работы программы микроконтроллера

Симуляция модели устройства велась, с целью оценки работоспособности модели в САПР Proteus.

Для этого была выполнена схемотехническая модель и написана программа для симуляции работоспособности установки, которая при создании устройства, будет записана в память микроконтроллера ATmega32.

Поскольку на выходе акселерометра формируется код, который содержит информацию об изменении угла, а в программе Proteus отсутствует модель акселерометра, то предлагается в схему модели датчика угла включить другое устройство, модель которого работает аналогично модели акселерометра. То есть, предлагается вместо акселерометра использовать АЦП, который на выходе формирует код, аналогичный коду на выходе акселерометра. Предлагаемый АЦП должен измерять значение падения напряжения на переменном резисторе. Это падение напряжения имитирует изменение угла акселерометра.

На рис. 4 изображено:

- LCD1 – жидкокристаллический дисплей LM016L на основе контроллера HD44780, используется как устройство вывода информации;
- U1 – плата Atmega32A-PU;
- X1 – кварцевый резонатор;
- U4 – АЦП AD7705;
- J1 – интерфейс UART;
- RV1 – переменный резистор, использован в качестве элемента, который имитирует изменение угла наклона.

Так же для разрабатываемого устройства была разработана электрическая принципиальная схема с помощью САПР Altium Designer, представленная на рис. 5, руководствуясь техническими описаниями выбранных элементов, в схему были добавлены необходимые для их функционирования элементы.

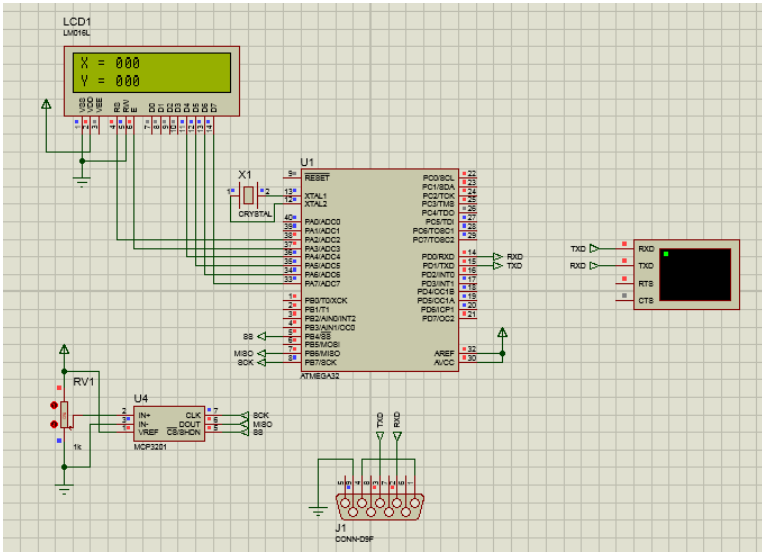


Рис. 4. Моделирование устройства в САПР Proteus

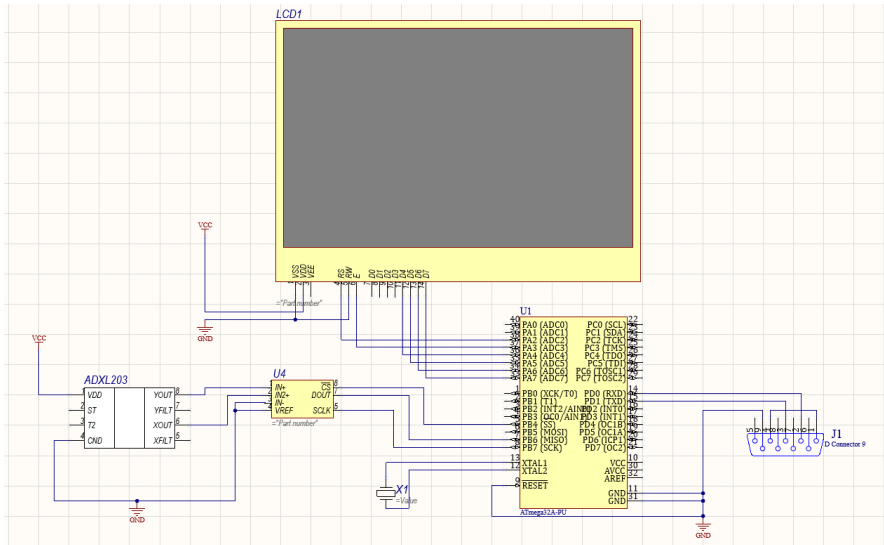


Рис. 5. Схема электрическая принципиальная

Была проведена проверка работоспособности собранного устройства, изменяя значения на резисторе, имитирующего работу акселерометра, можно заметить, что при этом меняются значения угла наклона по двум осям – X и Y и выводятся на LCD по интерфейсу UART.

Исходя из результатов, полученных в программе Proteus 8 Professional, можно сделать вывод о том, что разработанная модель работает корректно, результаты, которые были получены в ходе работы, соответствуют требованиям технического задания.

#### **Библиографический список**

1. Микромеханические системы и датчики: учеб. пособие / М. Е. Калинкина, О. И. Пирожникова, В. Л. Ткалич, А. В. Комарова. СПб.: ИТМО, 2020. 77 с.
2. Евстифеев А. В. Микроконтроллеров AVR семейств Tiny и Mega фирмы ATMEL; 5 изд. М.: Додэка-XXI, 2008. 558 с.

УДК 623.454.8

**Г. С. Гришкин**

студент кафедры конструирования и технологий электронных и лазерных средств

**Ю. В. Скороспехов** – кандидат технических наук – научный руководитель

### **АНАЛИЗ ИСПОЛЬЗОВАНИЯ ЛАЗЕРНЫХ ТЕХНОЛОГИЙ ПРИ РЕШЕНИИ ЗАДАЧ БОРЬБЫ С БПЛА**

В первые годы XXI века задача противодействия беспилотным летательным аппаратам (БПЛА) [1] решалась исключительно средствами поражения (ракетами и снарядами) зенитных ракетных комплексов (ЗРК) противовоздушной обороны (ПВО). С появлением средних и малых БПЛА, стало очевидно, что прямое отражение их массированного налета средствами ЗРК ПВО нерационально. Такой подход экономически не оправдан, из-за использования дорогостоящих ракет по большому числу относительно дешевых БПЛА. К тому же это ведет к быстрому расходу боевого запаса ЗРК и последующей неспособности отразить удар уже пилотируемой авиации, а также крылатых ракет высокоточного оружия.

В связи с этим, в настоящее время широко исследуются новые способы противодействия БПЛА. Одним из таких способов является использование средств направленного электромагнитного излучения, в том числе и лазерного.

Целью данной работы является систематизация и анализ различных видов БПЛА, как объекта поражения, а также предложение и обоснование способов поражения БПЛА, как эффективного средства решения данной задачи.

В настоящее время БПЛА зарекомендовали себя, как эффективное военное средство многопланового использования:

- разведывательные действия;
- передача сведений в наземный пункт в реальном времени;
- боевая атака;
- наведение авиации на наземные цели;
- корректировка огня ракетных войск;
- ретрансляция связи.

Современные БПЛА отличаются высококлассным техническим оснащением, превосходя классические пилотируемые аппараты по совокупности факторов:

- автономность;
- способность одновременно выполнять группу поставленных задач;
- снятие проблемы «человеческого фактора»;
- отсутствие риска для жизни человека;
- экономия на эксплуатации.

БПЛА принято делить по таким взаимосвязанным параметрам, как масса, время, дальность и высота полета. Выделяют следующие классы аппаратов [2]:

- «микро» (условное название) массой до 10 кг, временем полета около 1 ч. и высотой полета до 1 км;
- «мини» массой до 50 кг, временем полета несколько часов и высотой до 3...5 км;
- средние до 1000 кг, временем 10...12 ч. и высотой до 9...10 км;
- тяжелые с высотами полета до 20 км и временем полета 24 ч и более.

На сегодняшний день целый ряд стран создает боевые лазерные комплексы, предназначенные для противодействия БПЛА. В том числе и Россия, которая со времен СССР занимает лидирующие позиции в разработке лазерных комплексов вооружений.

Свойства лазерного излучения позволяют создавать узкий направленный луч с мощностью, достаточной для поражения объектов на расстоянии, путем фокусировки и удержания лазерного луча на цели в течение короткого времени. В результате достижения высокой плотности энергии в пятне излучения на цели происходит нагрев, расплавление или испарение материалов.

Лазерное оружие обладает следующими преимуществами: лазерный луч распространяется со скоростью света, он не подвержен влиянию гравитации; не имеет отдачи; не требует боеприпасов, кроме источника энергии.

Однако использование лазерного оружия имеет и свои недостатки: рассеивание и поглощение лазерного излучения в атмосфере; зависимость от метеоусловий; потребность в мощном источнике энергии; высокая стоимость производства лазерных комплексов.

При оценках порога разрушения лазерным излучением элементов конструкции БПЛА можно ограничиться более простыми соотношениями для тонких пластин, толщиной  $h$  [м]. Плотность энергии лазерного излучения  $E$  [Дж/м<sup>2</sup>], необходимая для разрушения произвольного элемента конструкции БПЛА можно оценить по следующей формуле:

$$E = [C \cdot (T_{пл} - T_0) + L_{пл} + C \cdot (T_{кип} - T_{пл}) + L_{исп}] \cdot \rho \cdot h, \quad (1)$$

где  $T_0$  – начальная температура, К;  $T_{пл}$  – температура плавления, К;  $T_{кип}$  – температура кипения, К;  $L_{пл}$  и  $L_{исп}$  – удельная теплота плавления и испарения соответственно, Дж/кг,  $C$  – теплоемкость материала, Дж/(кг·К);  $\rho$  – плотность материала, кг/м<sup>3</sup>.

С учетом длительности воздействия потребная для разрушения плотность мощности лазерного излучения  $I$  [Вт/м<sup>2</sup>] рассчитывается по формуле (2):

$$I = \frac{E}{\tau}, \quad (2)$$

где  $\tau$  – время воздействия лазерного излучения, с.

Под разрушением элемента конструкции БПЛА можно понимать либо доведение материала до плавления, либо до его испарения. Рассмотрим оба варианта:

Когда целью воздействия лазерного излучения является доведение материала конструкции БПЛА до плавления, выражение 1 принимает следующий вид:

$$E_{пл} = [C \cdot (T_{пл} - T_0) + L_{пл}] \cdot \rho \cdot h. \quad (3)$$

В случае, если целью воздействия лазерного излучения является испарение материала с поверхности БПЛА, то  $L_{исп} \gg C \cdot (T_{кип} - T_0) + L_{пл}$ . Тогда выражение (1) примет вид:

$$E_{исп} \approx L_{исп} \cdot \rho \cdot h. \quad (4)$$

Используя данные табл. 1, в которой указаны термодинамические характеристики некоторых конструкционных материалов [2], а также выражения (3) и (4), можно оценить требования к энергетике лазеров для каждого случая. Значение плотности энергии лазерного излучения, необходимое для разрушения стенки элемента конструкции БПЛА толщиной 0,1 см из перечисленных материалов, но без учета потерь излучения на трассе и на отражение от поверхности БПЛА, приведены в двух последних столбцах табл. 1.

Таблица 1

**Термодинамические характеристики материалов**

Материал	Температура плавления $T_{пл}$ , К	Теплота плавления $L_{пл}$ , Дж/кг	Температура кипения $T_{кип}$ , К	Теплота испарения $L_{исп}$ , Дж/кг	Плотность энергии при плавлении $E_{пл}$ , Дж/м <sup>2</sup>	Плотность энергии при испарении $E_{исп}$ , Дж/м <sup>2</sup>
Алюминий	933	$393 \cdot 10^3$	2772	$10,9 \cdot 10^6$	$2,6 \cdot 10^6$	$29,4 \cdot 10^6$
Титан	1941	$315 \cdot 10^3$	3560	$10,0 \cdot 10^6$	$5,3 \cdot 10^6$	$45,0 \cdot 10^6$
Сталь	1720...1790	$205 \cdot 10^3$	3135	$6,4 \cdot 10^6$	$7,3 \cdot 10^6$	$49,9 \cdot 10^6$

Время удержания лазерного пятна на одном месте быстродвижущейся цели для разрушения элементов ее конструкции по различным оценкам составляет от одной до нескольких десятков секунд. Для упрощения вычислений используем непрерывный режим работы установки. Подставив получен-

ные данные в выражение (2), можно найти плотность мощности лазерного излучения  $I$  [Вт/м<sup>2</sup>] для каждого случая.

Важным условием достаточности ущерба является размер пятна. Для не слишком больших целей непоправимый ущерб возникнет при размере пятна  $d$  [м] не менее 15 см [2]. Мощность  $I_{треб}$  [Вт], требуемую для разрушения стенки элемента конструкции БПЛА из перечисленных материалов без учета потерь излучения на трассе и на отражение от поверхности БПЛА для каждого случая, можно вычислить следующим выражением:

$$I_{треб} = I \cdot \frac{\pi \cdot d^2}{4} \quad (5)$$

Полученные результаты приведены в табл. 2.

Таблица 2

Требуемые мощности лазерного излучения

Материал	Мощность при плавлении $I_{треб\ пл}$ , кВт			Мощность при испарении $I_{треб\ исп}$ , кВт		
	при $\tau = 1$ с	при $\tau = 5$ с	при $\tau = 10$ с	при $\tau = 1$ с	при $\tau = 5$ с	при $\tau = 10$ с
Алюминий	45,9	8,83	4,59	519	104	51,9
Титан	93,6	19,4	9,36	795	159	79,5
Сталь	129	26,5	12,9	881	175	88,1

Анализируя полученные значения можно прийти к выводу, что для разрушения элемента конструкции БПЛА, путем доведения материала его поверхности до плавления, подходят лазеры, мощность которых способна достигать 100 кВт. Для данных целей могут быть использованы современные волоконные или химические лазеры.

В случае, когда скорость полета и материалы конструкции БПЛА позволяют удерживать на нем луч более 10 с, этого же лазера будет достаточно для испарения материала обшивки, что может привести к существенному изменению аэродинамических характеристик цели и к потере управляемости.

В промышленных лазерах для достижения высокой плотности мощности в области резания применяют фокусирование с помощью линз или зеркал. Это позволяет при сравнительно малой мощности лазера получить в фокусе луча плотность мощности до десятков ГВт/м<sup>2</sup>, что, как уже было показано, в разы превышает порог испарения некоторых конструкционных материалов. Диаметр таких линз находится в диапазоне до 3 см [4].

Оценим характеристики односторонне выпуклой линзы лазера с фокусным расстоянием 5 км. Величина коэффициента преломления материала линз для ближнего инфракрасного диапазона находится в интервале 1,4...1,7. Радиус кривизны поверхности линзы находится по формуле (6):

$$\frac{1}{f} = (n-1) \left[ \frac{1}{R_1} - \frac{1}{R_2} + \frac{(n-1)l}{nR_1R_2} \right], \quad (6)$$

где  $f$  – фокусное расстояние линзы, м;  $n$  – показатель преломления материала линзы;  $l$  – толщина линзы, м;  $R_1$  и  $R_2$  – радиусы кривизны первой и второй поверхностей линзы соответственно, м.

Подставив имеющиеся значения в выражение (6) получаем значения радиусов кривизны поверхности линз в диапазоне от 4 до 7 км. При диаметре линзы 3 см величина выпуклости в центре линзы относительно ее края/вогнутости центра зеркала относительно его края составляет несколько нм [4]. Создание оптических элементов с такими характеристиками невозможно при использовании современных технологий.

Однако существует возможность фокусирования лазерного излучения на дистанциях до 5 км. Если увеличить диаметр линзы или зеркала до 0,5 м, то величина выпуклости в центре относительно краев линзы/вогнутости центра зеркала относительно его края возрастает, что делает возможным их изготовление по существующим технологиям. Пример применения такого подхода показан на рис. 1.



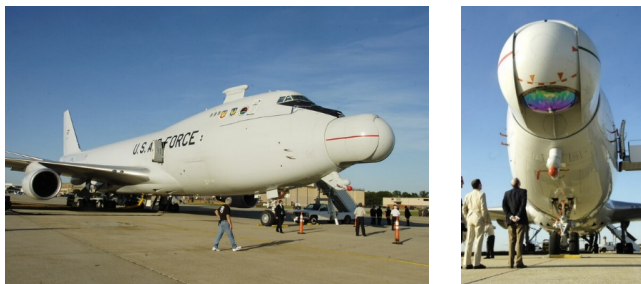


Рис. 1. Опытный образец на базе Боинга 747, оснащенный химическим лазером

Вероятность поражения БПЛА без отражателей и защитных экранов  $P_{пор}$  с помощью лазерного оружия можно определить по выражению (7) [5]:

$$P_{пор} = P_{обн} \cdot P_{нав} \cdot P_{уд} \cdot P_{разр}, \quad (7)$$

где  $P_{обн}$  – вероятность обнаружения БПЛА в четверти сферы. В рамках данной работы задача обнаружения цели не рассматривается, и для упрощения ее можно считать решенной. В таком случае, вероятность обнаружения БПЛА  $P_{обн} = 1$ ;  $P_{нав}$  – вероятность наведения луча на цель. Для механической следящей системы этот показатель может доходить до уровня 0,87 [5];  $P_{уд}$  – вероятность удержания лазерного пятна в заданной точке в течение заданного времени. Для БПЛА, летящего прямолинейно с постоянной скоростью,  $P_{уд} \approx 0,9$ . Для маневрирующего БПЛА с перегрузкой  $n_y \geq 1,7$  со сменой курса  $P_{уд} \leq 0,3$  [5];  $P_{разр}$  – вероятность того, что воздействие лазерного луча на конструкцию приведет к ее разрушению, взрыву горючего или боеприпаса. При возможности точной идентификации БПЛА эта величина может достигать значения 1 [5].

На основе выражения (7) и имеющихся данных можно сделать вывод, что БЛК способны справиться с задачей противодействия БПЛА с вероятностью  $P_{пор} \leq 0,78$ .

По результатам проведенного анализа можно сделать следующие выводы:

- применение БЛК экономически оправдано, поскольку затраты на генерацию лазерного излучения требует меньших материальных затрат, чем при использовании снарядов ЗРК ПВО при противодействии малоразмерным БПЛА;
- развитие лазерных технологий делает их не только перспективным способом противодействия БПЛА, но и позволяет успешно справляться с этой задачей уже в наши дни;
- БЛК способны справиться с задачей противодействия БПЛА на дистанции до 5 км с вероятностью до 0,78;
- БЛК не эффективны в неблагоприятных метеорологических условиях.

### Библиографический список

1. ГОСТ Р 57258-2016 Системы беспилотные авиационные. Термины и определения.
2. Лазеры: применения и приложения: Учебное пособие / Под ред. А. С. Борейшо. — СПб.: Издательство «Лань», 2016. — 520 с.
3. Макаренко С. И. Противодействие беспилотным летательным аппаратам: монография. СПб.: Наукоемкие технологии, 2020. 204 с.
4. Вакс Е. Д. [и др.]. Резание металлов излучением мощных волоконных лазеров. М.: Техносфера, 2019. 344 с.
5. Ростолчин В. В. Ударные беспилотные летательные аппараты и противовоздушная оборона – проблемы и перспективы противостояния // Беспилотная авиация. URL: [https://www.researchgate.net/publication/331772628\\_Udamnye\\_bespilotnye\\_letatelny\\_e\\_apparaty\\_i\\_protivovozdusnaa\\_oborona\\_-\\_problemy\\_i\\_perspektivy\\_protivostoania](https://www.researchgate.net/publication/331772628_Udamnye_bespilotnye_letatelny_e_apparaty_i_protivovozdusnaa_oborona_-_problemy_i_perspektivy_protivostoania) (дата обращения: 24.03.2023).

УДК 62.752.6

**М. А. Желавский**

студент кафедры эксплуатации и управления аэрокосмическими системами

**Д. С. Карпушин**

студент военного учебного центра

**Д. В. Коликов** – старший преподаватель – научный руководитель

## СПОСОБЫ СТАБИЛИЗАЦИИ УСТРОЙСТВ НАВЕДЕНИЯ И ЦЕЛЕУКАЗАНИЯ ВЕРТОЛЕТА

Системы наведения и целеуказания, расположенные на летательных аппаратах, обладают рядом преимуществ: высокая мобильность, высокая степень автономности, возможность использования на нескольких летательных аппаратах (ЛА) [1]. Воздушные суда с такими системами применяются для решения задач разведки, мониторинга территории и при выполнении поисково-спасательных операций. В последнее время подобные системы стали активно разрабатываться и применяться в составе беспилотных авиационных систем на основе беспилотных летательных аппаратов (БАС с БГЛА).

Размещение такой системы на вертолете позволяет значительно повысить ее эффективность. Благодаря режиму зависания ЛА возможно более точное и быстрое определение ряда параметров: расстояния до цели, её координат, а также скорости и эффективной площади рассеяния.

Комплексы наведения и целеуказания обычно строятся на основе оптико-электронных блоков, включающих в себя камеру высокого разрешения с функцией матричной стабилизации, лазерный дальномер для определения расстояния до цели и угла визирования матрицы, для распознавания цели в сложных метеоусловиях, а также на основе радиолокационных устройств – простых целевых антенн или фазированных антенных решёток (ФАР, АФАР) [2].

Задача наведения и сопровождения сводится к точному определению положения вектора нормали фазового центра радиолокационной антенны или направления линии визирования оптико-электронного устройства. Связь между направлениями осуществляется с помощью расширенных матриц поворота, записанных с помощью углов Эйлера-Крылова в переносных или обобщенных системах координат [3].

При решении поставленной задачи необходимо учитывать пространственные эволюции ЛА. В случае ЛА типа вертолет – это низкочастотные колебания по крену тангажу и рысканью и высокочастотные вибрации, вызванные работой несущего винта. Также необходимо учитывать собственные колебания платформы, на которой расположены устройства наведения и целеуказания, и собственные колебания опорно-поворотных устройств, вызванные особенностями работы двигателей линейных актуаторов: погрешностью мертвого хода, несоосностью посадочных мест валов, биением тел качения подшипников и т. д. Таким образом, колебания напрямую влияют на работу системы: уменьшается точность распознавания цели, уменьшается частота смены кадров, уменьшается точность и скорость срабатывания электромеханических поворотных устройств, увеличивается скорость износа механических звеньев.

При длительном вибрационном воздействии возможно появление катастрофических ошибок округления, которые существенно снижают точность и достоверность получаемой информации. Такие ошибки возникают вследствие наличия погрешности записи информации в формате числа с плавающей точкой [4]. Многократные операции с числами, содержащими такие погрешности (примером такой операции в данном случае может служить раскрытие матричного определителя, составленного из направляющих косинусов), приводят к появлению невязки между истинным и рассчитанным значением измеряемой величины, сравнимой с порядком самой величины.

Для гашения низкочастотных колебаний вертолета целесообразно использовать двухосную гиросtabilизированную платформу [5]. Структурная схема такой системы может выглядеть следующим образом (рис. 1).

В ее состав входит трехстепенной гироскоп, датчик углового положения, снимающий угол поворота гироскопа вокруг оси прецессии (ДУП), блок согласования и регулировки, необходимый для формирования разностного сигнала  $\pm \Delta\gamma$  (БСР), усилитель, необходимый для формирования управляющего напряжения (У), моментный двигатель (МД) и опорно-поворотное устройство (ОПУ) для приведения платформы в исходное положение.



Рис. 1. Структурная схема гиростабилизированной платформы

При возникновении возмущающего момента  $M_B$  гироскоп начинает прецессировать с некоторой угловой скоростью  $\dot{\gamma}$ , в результате чего возникает гироскопический момент  $M_r$ , компенсирующий внешнее возмущение. Из-за инерционности гироскопа момент  $M_r$  не позволяет полностью компенсировать отклонение от заданного положения платформы; возникает угол поворота гироскопа  $\gamma$ , который снимается с помощью ДУП, преобразуется в разностный сигнал, усиливается и подается на МД в виде напряжения. МД создает момент коррекции  $M_k$ , сонаправленный с гироскопическим моментом  $M_r$ . В ходе работы двигателя увеличивается создаваемый им коррекционный момент, уменьшается скорость прецессии гироскопа. В итоге внешнее воздействие оказывается скомпенсированным моментом двигателя; при этом гироскоп в процессе стабилизации повернется на угол  $\gamma_{ст}$ .

Преимущество такой схемы – возможность коррекции при длительном воздействии внешних возмущений. Однако при этом необходимо учитывать накопление статической ошибки в виде угла поворота гироскопа. Также такой подход не обеспечивает приемлемую стабилизацию при кратковременных воздействиях и малых отклонениях.

Одним из эффективных способов снижения амплитуды вибраций является метод динамического гашения. Этот метод заключается в подключении к объекту защиты специальных устройств, которые локально изменяют его вибросостояние.

Подавление вибраций объекта может происходить путем перераспределения энергии колебаний от объекта к гасителю (инерционные гасители), путем преобразования механической энергии колебаний в тепловую и ее рассеяния (поглотители колебаний) или объединением этих способов (динамические гасители с трением) [6].

Физическая сущность первого способа заключается в следующем: при определенной настройке параметров системы объект-гаситель вблизи настроечной частоты демпфер воздействует на основную массу с силой, равной по величине вынуждающей, но противоположной ей по направлению, что является частным случаем антирезонанса [7].

Второй способ основан на повышении диссипативных свойств системы путем присоединения дополнительно специально демпфируемых элементов. При этом не нарушаются свойства устойчивости системы в целом, исходя из уравнений Лагранжа II-го рода [8].

Конструктивно динамические гасители можно выполнить на основе пассивных элементов, активных, имеющих собственные источники энергии, а также их комбинаций. Для увеличения диссипативных свойств колебательной системы применяются демпферы: сухого (тросовые, силиконовые, резиновые) и вязкого трения.

Состав системы активной виброзащиты включает в себя следующие компоненты: элементы обнаружения и фиксации вибраций на основе акселерометров, микропроцессорные системы анализа колебаний и формирования управляющих сигналов – систему управления, а также механизм реализации активной компенсации, генерирующий противофазные колебания – исполнительное устройство.

Сигнал датчика вибраций в виде напряжения  $U_d$  сравнивается с опорным напряжением  $U_{оп}$ . Разностный сигнал  $\Delta U$  поступает на вход управляющей системы, где происходит формирование управляющего сигнала  $U_{упр}$ , который поступает на вход электромеханического исполнительного устройства. Исполнительное устройство формирует механическое воздействие  $F_{исп}$ , которое уменьшает или полностью устраняет колебания защищаемой системы.

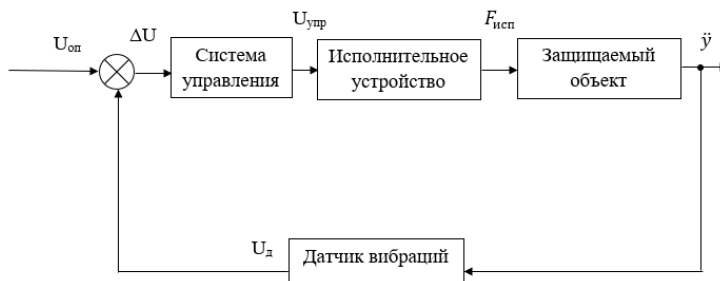


Рис. 2. Структурная схема системы активной виброзащиты

Одним из перспективных путей уменьшения влияния вибрационного воздействия является метод, основанный на генерации опорных колебаний постоянной частоты с помощью авиационного вибратора. Амплитуда и частота генерируемых колебаний априори известны, что позволяет применять методы цифровой фильтрации сигнала для нивелирования влияния случайных механических взаимодействий программным путем.

Таким образом, предложенные методы могут использоваться для гашения колебаний при синтезе высокоточных позиционных систем, применяемых на вертолетах. Стоит отметить, что современные системы целеуказания не способны в автономном режиме осуществлять наведение на цель с известными координатами и ее сопровождение в течение выполнения полетного задания. Существующие оптико-электронные системы не обеспечивают достаточную точность позиционирования системы в целом [9].

#### Библиографический список

1. Донсков Ю. Е., Богословский А. В., Матвеев Д. С. Способы применения беспилотных летательных аппаратов радиотехнической разведки в ходе ведения боевых действий в тактической зоне // Военная мысль. № 8. 2021.
2. Проектирование излучающей апертуры АФАР самолетной РЛС переднего обзора для системы наведения и посадки в автоматическом режиме / В. В. Чистюхин, К. С. Лялин, В. И. Орешкин, С. С. Меркулов, Ю. М. Мелёшин // Известия вузов. Электроника. Т. 20. № 5. 2015.
3. Методика проектирования аппаратуры наведения, целеуказания и стабилизации в составе беспилотного летательного аппарата / В. И. Бабичев, В. С. Фимушкин, О. В. Горячев, С. В. Феофилов // Известия ТулГУ. Технические науки. 2012. № 11.
4. Slyusarenko A. S. To the Problem of Rounding Errors Evaluation // International Scientific Review of Problems and Prospects of Modern Science and Education: Collection of Scientific Articles. XIV International Correspondence Scientific and Practical Conference. Boston, USA. May 24-25, 2018). Pp.12–26.
5. Лысов А. Н., Лысова А. А. Теория гироскопических стабилизаторов: учеб. пособие. Челябинск: Издательский центр ЮУрГУ, 2009.
6. Караваев В. Г., Прядко Ю. Г., Черногоров Е. П. Малые колебания механических систем: учеб. пособие. Челябинск: Издательский центр ЮУрГУ, 2017.
7. Вульфсон И. И. Краткий курс теории механических колебаний / Библиотека ВНТР. М.: ВНТР, 2017.
8. Гантмахер Ф. Р. Лекции по аналитической механике. М.: Наука, 1966.
9. Волков В. Г. Вертолетные оптико-электронные системы наблюдения и разведки. Специальная техника, 2002.

УДК 004.9; 681.3

**А. В. Жмури**

студент кафедры конструирования и технологий электронных и лазерных средств

**Е. М. Домбровский** – преподаватель – научный руководитель

### РАЗРАБОТКА ОБУЧАЮЩЕГО ПРИЛОЖЕНИЯ «КОРРЕЛЯЦИОННАЯ ОБРАБОТКА СИГНАЛОВ»

Корреляционная обработка сигналов – это один из оптимальных способов обработки сигналов, обеспечивающий максимальное соотношение сигнал/помеха на выходе приемного устройства радиолокационной системы.

Суть корреляционной обработки состоит в сравнении входных сигналов с эталонным сигналом, параметры которого соответствуют параметрам ожидаемого полезного сигнала, отраженного от цели. Степень подобия двух сигналов (функций) оценивается корреляционной функцией (1), представляющей собой интеграл по времени от произведения принятого сигнала с помехами и эталонного сигнала [1]:

$$R(x, y, z) = \int (S(t) + N(t)) \cdot S^*(t) dt = \int S(t) \cdot S^*(t) dt + \int N(t) \cdot S^*(t) dt, \quad (1)$$

где  $S(t)$  – полезный или входной сигнал,  $N(t)$  – помеха,  $S^*(t)$  – эталонный сигнал,  $x, y, z$  – параметры, характеризующие отличие функций  $S(t)$ ,  $N(t)$  и  $S^*(t)$ . Обычно в качестве аргументов  $x, y, z$  выступают соответственно следующие характеристики:  $\tau$  – разность времени входного и эталонного сигналов,  $\Delta F$  – разность частот входного эталонного сигналов,  $\Delta \varphi$  – разность фаз входного и эталонного сигналов.

Выражение (1) можно представить в следующем виде:

$$R_S(x, y, z) = \int S(t) \cdot S^*(t) dt \quad (2)$$

$$R_N(x, y, z) = \int N(t) \cdot S^*(t) dt \quad (3)$$

Выражение (2) называют автокорреляционной функцией сигнала, которая имеет максимальное значение в случае, когда входной сигнал  $S(t)$  и эталонный сигнал  $S^*(t)$  имеют одинаковые параметры. При различии параметров входного и эталонного сигнала значение автокорреляционной функции будет отлично от максимального значения, причем чем больше различие в параметрах, тем меньше значение функции [2].

В связи со сложностью вычисления автокорреляционной функции и трудностью восприятия данного материала возникла необходимость наглядного представления процесса корреляционной обработки сигналов. Необходимо отметить, что разрабатываемое приложение должно удовлетворять следующим требованиям: приложение должно быть удобно в использовании; включать в себя блок теоретического материала по теме, и интерактивный блок, позволяющий задавать требуемые параметры анализируемого сигнала с визуализацией результатов.

С учетом предъявляемых требований, была выбрана среда разработки MATLAB, позволяющая выполнять инженерные и математические расчеты, и имеющая широкий пакет прикладных программ: MATLAB App Designer, MATLAB Runtime, MATLAB Compiler и MATLAB WebAppServer. Перечисленные пакеты прикладных программ позволяют создать автономное приложение, для запуска которого будет достаточно использование любого установленного браузера на ПЭВМ, что в свою очередь обеспечит удобство в использовании [3].

Исходя из вышесказанного определим минимальные требования к предустановленному программному обеспечению: операционная система Windows 10, пакет MATLAB WebAppServer, наличие доступа к Интернет, для обмена данными с сервером MATLAB.

Как уже упоминалось выше, приложение будет содержать теоретическую часть в виде текстового материала по данной теме и интерактивную часть, которая позволит наглядно увидеть изменения сигналов и формирование автокорреляционной функции в графическом виде. Рассмотрим этапы разработки подробнее.

В процессе разработки блока интерактивной части приложения разработана блок-схема (рис. 1), которая при реализации позволит выводить на отображение обучающемуся три формы сигналов: полезный сигнал, эталонный сигнал и автокорреляционную функцию, формируемую на основе двух предыдущих сигналов. Вид отображаемых сигналов будет соответствовать введенным параметрам, которые так же будут отображаться в соответствующих полях.

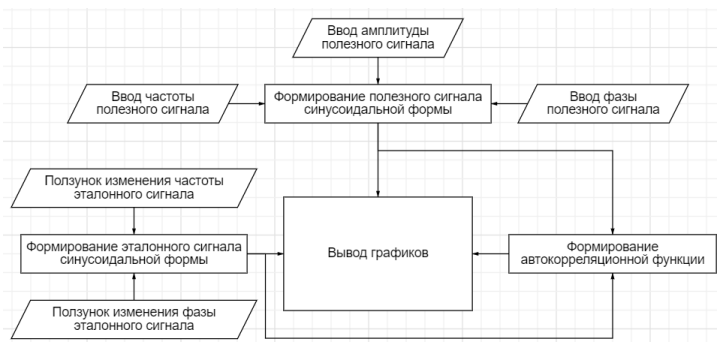


Рис. 1. Блок-схема интерактивной части приложения

В ходе реализации интерактивного блока созданы: окно для вывода результатов, исполнительный элемент для начала формирования результатов, три поля ввода начальных параметров и два элемента, позволяющих оперативно изменять начальные параметры эталонного сигнала с оперативным отображением изменений результатов (рис. 2). Логика работы основных элементов соответствует этапам корреляционной обработки.

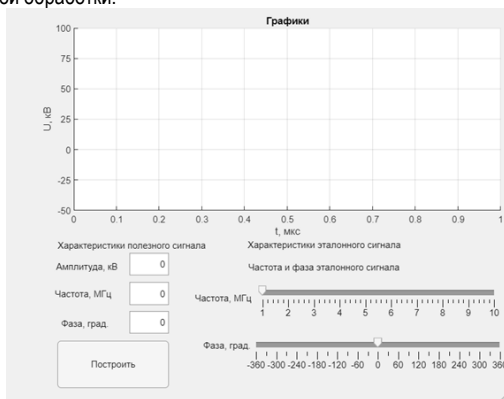


Рис. 2. Интерактивная часть приложения

Теоретический блок представляет из себя окно (рис. 3), в котором материал по теме, представлен с использованием языка гипертекстовой разметки HTML, что обеспечивает его отображение в используемом браузере [4].

### Корреляционная обработка сигналов

Корреляционная обработка сигналов - это один из оптимальных способов обработки сигналов. Оптимальным способом обработки сигналов называют способы, обеспечивающие максимальное соотношение **сигнал/помеха** на выходе приемного устройства.

Суть корреляционной обработки состоит в сравнении входных сигналов с эталонным сигналом, параметры которого соответствуют параметрам ожидаемого полезного сигнала, отраженного от цели. Эталонный сигнал это копия ожидаемого полезного сигнала. Эталонный сигнал формируется в приемном устройстве.

Рис. 3. Теоретическая часть приложения

Для реализации приложения выполнена сборка блоков в единую структуру, путем формирования *ctf* файла в пакете MATLAB Compiler, что позволит в дальнейшем открыть его в браузере через приложение MATLAB WebAppServer. Необходимо отметить, что для реализации данного действия на ПЭВМ обязательно должны быть предустановлены пакеты MATLAB WebAppServer и MATLAB Runtime.

Процесс сборки блоков в приложении осуществляется путем загрузки файла *ctf* в папку "Open App Folder" (рис. 4) и выполнением "Open Home Page". Выполнив данные действия приложение открывается в новой вкладке установленного браузера компьютера. В дальнейшем перенос *ctf*-файла уже не требуется.

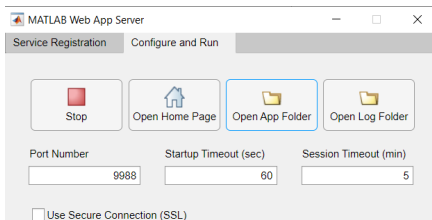


Рис. 4. Интерфейс MATLAB WebAppServer

Общий вид приложения, отображаемого в браузере (рис. 5).



Рис. 5. Окно приложения

Таким образом, разработано приложение, позволяющее наглядно представить сложные математические вычисления, что в свою очередь способствует более качественному усвоению материала по выбранной теме. Приложение отвечает всем современным требованиям и может быть использовано студентами и преподавателями при изучении принципов построения радиотехнических систем в разделе «Обработка радиолокационных сигналов».

*На приложение получено свидетельство о государственной регистрации программы для ЭВМ № 2022683949 «Приложение «correlation signal processing» для обучения студентов по теме «Корреляционная обработка сигналов», правообладатель: ФГАОУВО «Санкт-Петербургский государственный университет аэрокосмического приборостроения», авторы: Жмурин А. В., Смирнов А. Д., заявка № 2022684020 от 9 декабря 2022 года.*

#### **Библиографический список**

1. Радиотехнические устройства и элементы радиосистем: учеб. пособие. М.: Изд-во ФГУП Выш. Шк., 2005. 292 с.
2. Каганов В. И. Радиотехника: учеб. пособие. М.: Издательский центр «Академия», 2006. 352 с.
3. Сайт по MATLAB – [exponenta.ru](http://exponenta.ru), раздел «appdesigner». URL: <https://docs.exponenta.ru/matlab/ref/appdesigner.html?searchHighlight=matlab%20app%20designer> (дата обращения: 20.02.2023).
4. Сайт – [htmlbook.ru](http://htmlbook.ru), раздел «теги». URL: <http://htmlbook.ru/html> (дата обращения: 20.02.2023).



УДК 004.032.26

**И. А. Зорин**

студент кафедры прикладной информатики

**А. С. Швайко**

студент кафедры прикладной информатики

**Д. В. Жданов**

студент кафедры прикладной информатики

**Ю. В. Скороспехов** – кандидат технических наук – научный руководитель

## **РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА БАЗЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ОЧИСТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ ОТ ВРЕДНЫХ ФАКТОРОВ**

В настоящее время в текстовых материалах многих авторов содержится большое количество вредных факторов, отрицательно влияющих на логику изложения текста, а также на общее понимание содержания излагаемого материала.

Поэтому возникает потребность в разработке гибких и универсальных средств и методов, позволяющих устранять вредоносные факторы, что в настоящее время является актуальной задачей.

**Предварительная обработка данных.** Каждое предложение разбивается на список из символов, затем для каждого символа определяется его код в кодировке ASCII (American standard code for information interchange, Американский стандартный код для обмена информацией). Все числа меньше 1040 заменяются нулями, а из каждого числа с кодом 1040 и больше вычитается 1039. Это позволяет все символы, являющиеся шумами и не влияющими на содержание текста, представить нулями и уменьшить их влияние на результат работы нейросети. Так же это позволяет представить символы русского алфавита кодами от 1 до 66, что сильно облегчает чтение преобразованных данных.

После посимвольного представления предложения в цифровой формат, количество символов в предложении ограничивается 200. Если их было больше, то остальные удаляются, если символов меньше, то список дополняется нулями. В итоге предварительной обработки, предложение представлено списком из 200 чисел, каждому из которых соответствует буква русского алфавита. Это представление имеет существенный недостаток, так как для обработки текста большого объёма придётся разбивать его на части по 200 символов, что приведет к замедлению работы алгоритма.

Для реализации модели был использован язык программирования Python и три библиотека Keras – для создания модели нейронной сети, NumPy – для обработки данных и Matplotlib – для визуализации результатов обучения и тестирования модели.

В качестве входного слоя модели выступает слой Embedding, преобразующий каждое число в плотный семидесятимерный вектор, представляющий из себя список ненулевых значений, длиной в семьдесят элементов. На каждом этапе обучения модели данный слой меняет значения в плотных векторах, что позволяет представить данные в удобном для поиска связей виде [1].

Первым скрытым слоем является слой Dropout, который отключает определённое количество случайных нейронов, чтобы избежать переобучения нейронной сети, ситуации, когда модель показывает хорошие результаты на обучающем наборе данных, но плохие на тестовом [2].

Второй и третий скрытые слои – слои LSTM (Long short-term memory – долгая краткосрочная память), на которых происходят обнаружение и формирование признаков, необходимых для определения наличия или отсутствия obscene лексик в подаваемой последовательности [3].

В качестве основных слоёв были выбраны слои LSTM, потому что их архитектура лучше всего подходит для используемого представления данных, так как она позволяет обрабатывать связи между элементами подаваемой последовательности, что позволяет учитывать влияние символов по отдельности и в сочетании с соседними символами [3].

Выходным слоем является слой Dense, с одним нейроном и сигмовидной функцией активации, на вход принимает массив из 64 элементов, на выходе выдаёт число от “0” до “1”, которое показывает вероятность того, что полученное предложение содержит obscene лексику [4].

В качестве ошибки используется бинарная кросс-энтропия, измеряющая расстояние между двумя вероятностными распределениями, чем меньше данная величина, тем больше данные похожи

друг на друга, и тем лучше модель предсказывает наличие или отсутствие угрожающих факторов в последовательности. Для метода градиентного спуска был выбран алгоритм Adam, который меняет настраиваемые параметры нейронной сети в сторону минимума градиента, при этом скорости изменения параметров различная, чтобы с большей вероятностью получить наилучший результат обучения [5, 6].

Результат обучения и тестирования нейронной сети представлен на рис. 1–3.

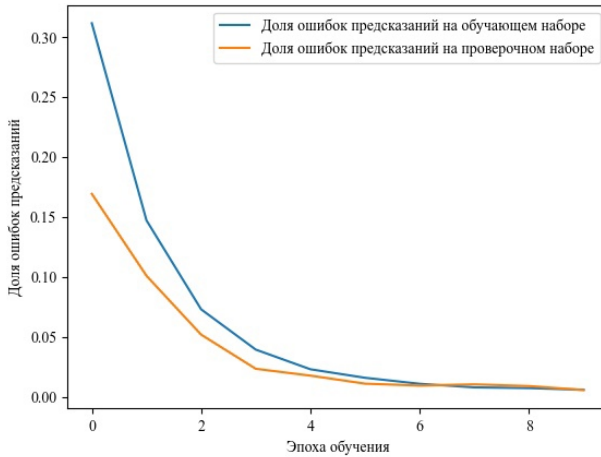


Рис. 1. Ошибка предсказания

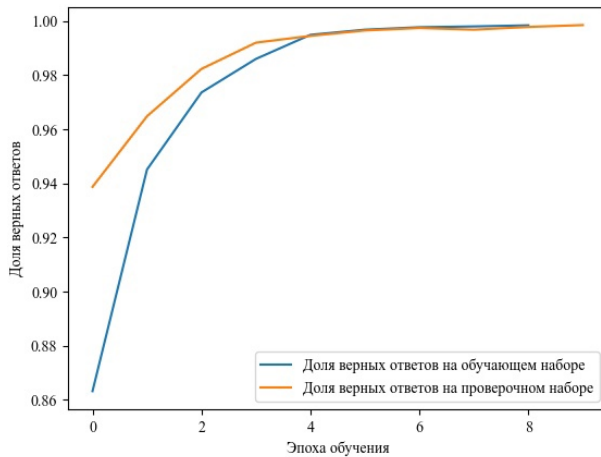


Рис. 2. Точность верных ответов

```

Epoch 1/10
625/625 [=====] - 23s 30ms/step - loss: 0.3115 - accuracy: 0.8632 - val_loss: 0.1692 - val_accuracy: 0.9387
Epoch 2/10
625/625 [=====] - 20s 31ms/step - loss: 0.1471 - accuracy: 0.9451 - val_loss: 0.1009 - val_accuracy: 0.9648
Epoch 3/10
625/625 [=====] - 20s 31ms/step - loss: 0.0729 - accuracy: 0.9736 - val_loss: 0.0518 - val_accuracy: 0.9823
Epoch 4/10
625/625 [=====] - 19s 30ms/step - loss: 0.0393 - accuracy: 0.9860 - val_loss: 0.0233 - val_accuracy: 0.9920
Epoch 5/10
625/625 [=====] - 20s 31ms/step - loss: 0.0229 - accuracy: 0.9923 - val_loss: 0.0176 - val_accuracy: 0.9945
Epoch 6/10
625/625 [=====] - 20s 32ms/step - loss: 0.0158 - accuracy: 0.9949 - val_loss: 0.0109 - val_accuracy: 0.9965
Epoch 7/10
625/625 [=====] - 19s 31ms/step - loss: 0.0108 - accuracy: 0.9968 - val_loss: 0.0093 - val_accuracy: 0.9974
Epoch 8/10
625/625 [=====] - 19s 30ms/step - loss: 0.0078 - accuracy: 0.9977 - val_loss: 0.0103 - val_accuracy: 0.9968
Epoch 9/10
625/625 [=====] - 19s 30ms/step - loss: 0.0072 - accuracy: 0.9980 - val_loss: 0.0089 - val_accuracy: 0.9978
Epoch 10/10
625/625 [=====] - 19s 30ms/step - loss: 0.0058 - accuracy: 0.9984 - val_loss: 0.0056 - val_accuracy: 0.9985
    
```

Рис. 3. Результаты обучения на каждой из эпох

Из показанных результатов обучения модели видно, что на протяжении 12 эпох точность нейросети возрастает, а ошибка уменьшается, из рис. 3 видно, что модель имеет ошибку ниже 1%, что отражает отличную точность в поставленной задаче.

Несколько примеров предсказания нейросети и время, за которое осуществляется предсказание показаны в табл. 1

Таблица 1

**Результаты предсказаний нейронной сети**

Текст	Время предсказания, с	Результат обнаружения нецензурной лексики
П***@ как не сладко мне сегодня пришлось	0,073	Есть
Отлично так и сделаю в следующий раз	0.096	Нет
Та*к держать! Кто-то ещё хо*чет поучаствовать?	0.070	Нет

Результаты проведённых исследований позволяют судить о том, что для посимвольного представления данных, модель на базе LSTM слоёв отлично показала себя, так как архитектура данных слоёв позволяет искать закономерности не только на основе каждого символа в отдельности, но и на основе связей между символами, что позволило получить такие высокие результаты.

**Библиографический список**

1. Embedding layer. URL: [https://keras.io/api/layers/core\\_layers/embedding/](https://keras.io/api/layers/core_layers/embedding/) (дата обращения: 12.03.2023).
2. Dropout layer. URL: [https://keras.io/api/layers/regularization\\_layers/dropout/](https://keras.io/api/layers/regularization_layers/dropout/) (дата обращения: 12.03.2023).
3. LSTM layer. URL: [https://keras.io/api/layers/recurrent\\_layers/lstm/](https://keras.io/api/layers/recurrent_layers/lstm/) (дата обращения: 10.03.2023).
4. Dense layer. URL: [https://keras.io/api/layers/core\\_layers/dense/](https://keras.io/api/layers/core_layers/dense/) (дата обращения: 12.03.2023).
5. Probabilistic losses. URL: [https://keras.io/api/losses/probabilistic\\_losses/](https://keras.io/api/losses/probabilistic_losses/) (дата обращения: 10.03.2023).
6. Adam. URL: <https://keras.io/api/optimizers/adam/> (дата обращения: 10.03.2023).

УДК 623.613

**Д. А. Зырянов**

студент кафедры автоматизированных и инфокоммуникационных систем

**О. В. Колупаев** – преподаватель – научный руководитель

### ИСПОЛЬЗОВАНИЕ БПЛА ДЛЯ ВЫБОРА ТРАССЫ ПРИ ПРОКЛАДКЕ ПОЛЕВОЙ КАБЕЛЬНОЙ ЛИНИИ С УЧЕТОМ ОПЫТА СВО

С начала проведения СВО на территории ЛНР, ДНР и Украины становится более яркой тенденция к использованию беспилотных летательных аппаратов (далее БПЛА). Они находят свое применение при выполнении не только стратегических операций, разведки, но и при тактических действиях. Потому вопрос более обширного внедрения БПЛА во все области военных действий становится как никогда актуальным.

В отношении применения БПЛА на поле боя с начала боевых действий более активными были противники. Это объясняется как внешними поставками устройств, так и опытом применения беспилотников странами НАТО. Они активно использовали аппараты для разведки и уничтожения точечных целей с помощью бомб и ракет.

На данный момент можно увидеть следующие варианты применения БПЛА:

- разведка;
- уничтожение точечных целей;
- уточнение местоположения противника;
- боевое охранение;
- поиск неисправностей и повреждений.

Разведка сейчас представляет собой комплекс мероприятий. Он включает разведку местности, уточнение координат и ориентиров, разведку противника, разведку сил и средств противника.

Уничтожение целей с начала проведения СВО приобрело неожиданный характер. Если ранее применялись тяжелые и сверхтяжелые беспилотники с ракетным или бомбовым вооружением, то теперь наоборот применяются легкие и сверхлегкие беспилотники с автоматическими сбрасывателями гранат и легких бомб. Это обусловлено сильной нехваткой БПЛА на поле боя, проблемой быстрого обнаружения и уничтожения БПЛА, опасностью обнаружения своих позиций. Эти условия необходимо вкладывать при проектировании новых беспилотных средств.

Уточнение местоположения противника показало наивысшую эффективность при применении БПЛА. Этот метод применения позволяет точно и оперативно уточнять координаты противника и своевременно уничтожать его.

БПЛА могут помочь в патрулировании территории, при боевом охранении. Их скорость передвижения позволяет своевременно засечь врага и дать возможность для быстрого реагирования.

С помощью БПЛА можно искать места потенциального повреждения полевой кабельной линии и других узлов связи.

На данный момент на территории боевых действий применяются: «Москит», «Ланцет», «Куб», «Орлан», «Орлан-10», «Элерон», «Иноходец», «Форпост-Р», «Герань-2». Со стороны противника применяются: Bayraktar TB2, Puma, Quantix Recon, Switchblade, MQ-9 Reaper, FlyEye, H10 Poseidon II, «Фурия» и «Лелека-100». Данные беспилотники представляют класс летающих крыльев. Они имеют высокую скорость, дальность полета. Однако главный их недостаток – невозможность зависания на одной точке. Ввиду этого особую популярность среди рядовых военных завоевали гражданские БПЛА, такие как: DJI Mavic, Phantom, Matrice 30T, Autel EVO II, Xiaomi Fimi X8, а также самодельные аппараты. Они имеют коптерную конфигурацию и возможность зависания на одной точке.

Министерство обороны при анализе проблем с поставкой БПЛА и при анализе выполняемых задач с использованием БПЛА пришло к выводу о необходимости создания новых коптерных беспилотников. Сейчас эту задачу выполняет концерн «Алмаз-Антей», разрабатывающий аппарат «Добрыня». Потому стоит ожидать и дальнейшего развития этого класса беспилотных летательных аппаратов.

Главными преимуществами БПЛА являются: большой радиус действия, скрытность, скорость, широкий диапазон применения. Однако есть и крупные недостатки, такие как:

- простота подавления: подавление связи управления или GPS;
- относительная простота уничтожения;
- раскрытие позиций оператора и подразделения.

В ходе СВО были найдены частные решения для борьбы с этими недостатками. Например, были разработаны дроны-камикадзе, цена которых позволяет использовать их при потере связи как управляемые или неуправляемые ракеты и бомбы.

Для предотвращения уничтожения БПЛА их делают меньше и маневреннее. В данном классе хорошо себя показали самодельные гоночные дроны с задаваемыми характеристиками.

Для предотвращения раскрытия позиций противником применяются специальные средства радиоэлектронной борьбы, применяются правила смены дислоцирования, дистанционного пуска БПЛА, автономного полета дронов по координатам. Однако вопрос защиты операторов остается открытым.

В отношении прокладки полевой кабельной линии связи будет интересно рассматривать БПЛА для разведки и рекогносцировки.

Рекогносцировка включает в себя уточнение трассы прокладки полевой кабельной линии. В ее ходе изучаются или отмечаются:

- точки поворотов, изгибов трассы;
- способы прокладки кабеля на различных участках;
- точки переходов через препятствия: железнодорожные пути, автотрассы, водные преграды, болотистая местность;
- обход неблагоприятных участков: химически, радиационно зараженных;
- места оборудования переходов;
- ориентиры, прокладка в лесах и труднопроходимых местностях;
- точки установки НУП, ОУП, оконечных, выделительных пунктов;
- места для общего резерва;
- пути обхода минных полей и крупных населенных пунктов.

При этом рекогносцировочная команда состоит из 3-4 человек и возглавляется опытным офицером роты. В ее распоряжении выделяются: средства передвижения, связи, приборы радиационной, химической и биологической разведки, миноискатели, бинокли, вооружение и необходимое имущество (флажки, указатели, таблички, колышки).

Почти все современные БПЛА оснащены компасами, GPS модулями, датчиками угла, камерами прямого транслирования. Это позволяет в ходе боевых действий быстрее проводить рекогносцировку, также позволяет защитить карты от попадания к противнику. БПЛА позволяют также оперативно уточнять трассу командирам взводов, начальникам линейных команд, аппаратных каналов образования и ротных участков без их непосредственного выдвижения к точкам местности.

БПЛА также могут снабжаться датчиками химического и радиационного заражения. А модули сброса или магнитометрические датчики могут обнаружить минные заграждения. Не исключаются и диверсионные действия противника в территории прокладки линии. Все это в совокупности может позволить не подвергать опасности личный состав.

Важным элементом рекогносцировки является обозначение точек прокладки линии, для дальнейшего прохода командой прокладки. Этот этап занимает значительную часть времени и может выполняться дистанционно с помощью беспилотников. Для этого предлагается внедрение специальной сбросовой системы с колышками со светоотражателями или геометками.

Система представляет собой специальное захватное устройство, с закрепленными колышками (рис. 1).

Двигатели приводят в движение специальный вал, с двумя внутренними дополнительными спицами для соединения отдельных толкателей. При зацеплении толкателя и внутреннего зубца начинает вращаться захват. Благодаря шестереночной передаче начинает вращаться в вторая половина захвата. Пружина позволяет в свободном состоянии удерживать 2 захвата между собой. Рис. 2А и 2Б демонстрируют процесс вращения вала с зацеплением зуба.

Вал представляет собой смещенные относительно друг друга толкатели (рис. 3). Сами толкатели являются отдельными деталями (рис. 4). Это позволяет в случае повреждения быстро заменить их, а также позволяет правильно настроить относительные углы между ними. Для синхронизации их

вращения и закрепления используются 2 внутренних спицы. Смещение необходимо для неодновременного вращения захватов (отпускания колышков по очереди).

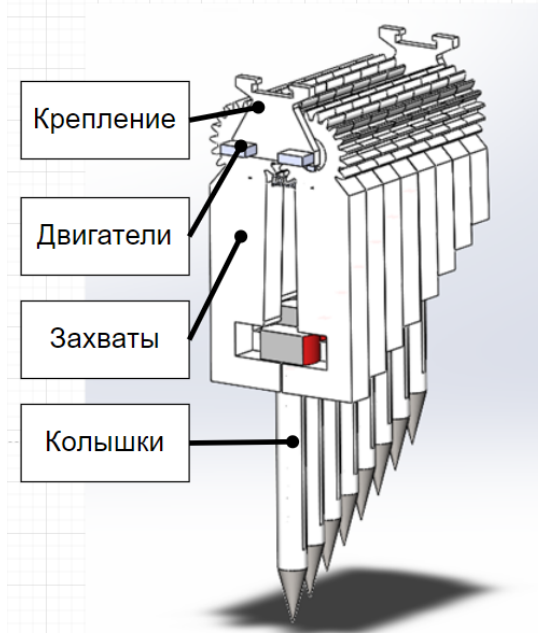


Рис. 1. Захватное устройство

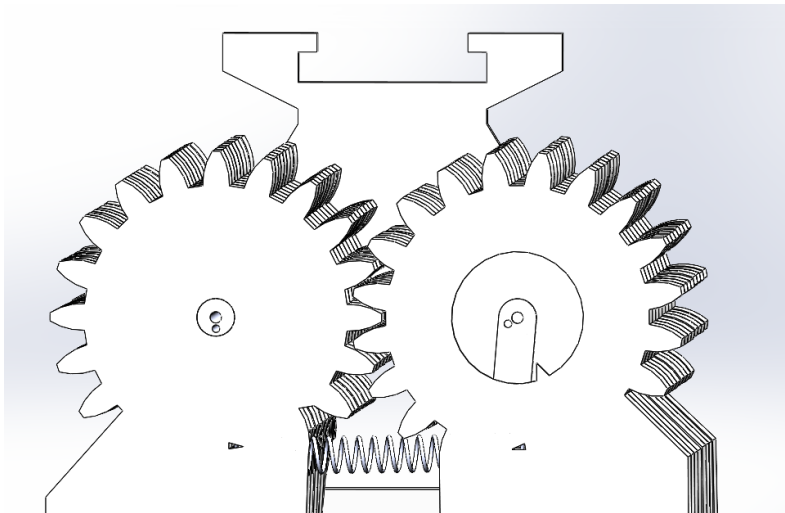


Рис. 2А. Начальное положение вала и захватов

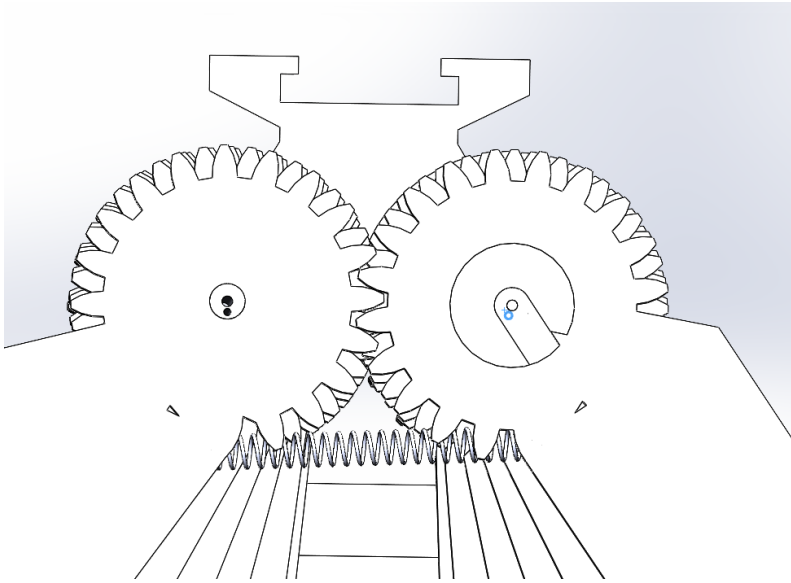


Рис. 2Б. Вращение вала и захватов

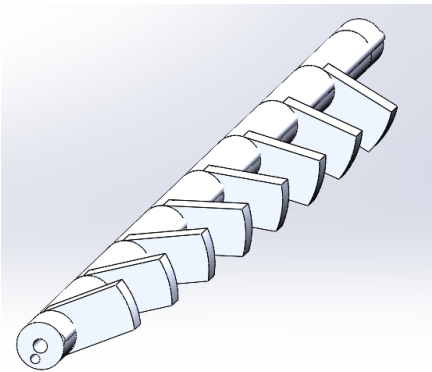


Рис. 3. Вал

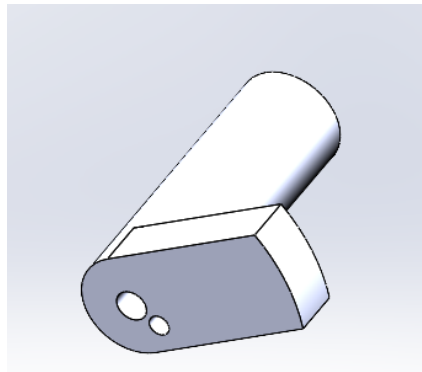
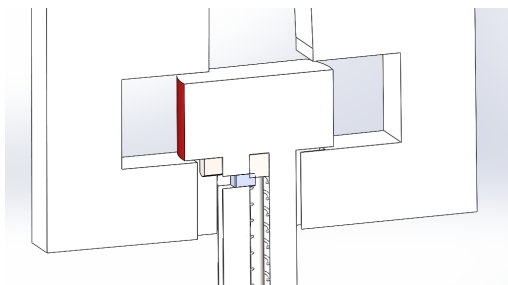


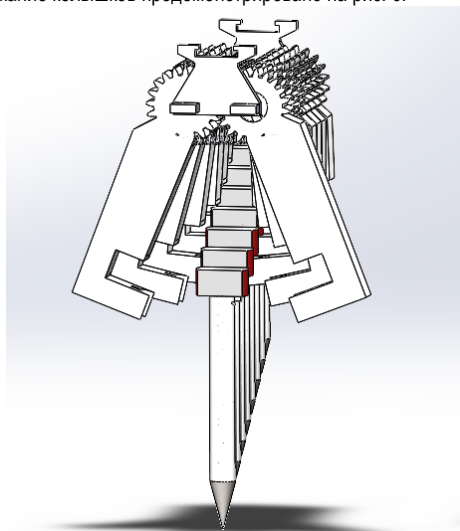
Рис. 4. Толкатель

Кольшек надежно закрепляется в захватном устройстве (рис. 5). Сверху над шляпкой кольшка установлены упоры, чтобы при приземлении БПЛА кольшек под его весом вбивался в землю. Для очередного вбивания длина захватных устройств меняется. Благодаря этому следующие кольца не препятствуют установке предыдущих.



*Рис. 5. Удержание кольщика*

Поэтапное отпущание кольщиков продемонстрировано на рис. 6.



*Рис. 6. Поэтапное отпущание*

Для успешного закрепления кольщика внутри него смонтирован механизм пружинного вбивания. На чеке закреплен ударник. В момент разжатия захватов пружина начинает тянуть ударник вниз. Когда ударник столкнется с основанием кольщика, он передаст всю кинетическую энергию и заставит кольщик вбиться в землю.

Благодаря отдельным толкателям устройство является масштабирующимся: возможно закрепление от 1 до 16 кольщиков.

Этот прибор может найти применение и для уничтожения точечных целей гранатами и другими взрывными устройствами.

Другим важным применением данного прибора будет срочная доставка элементной базы к местам повреждения или строительства ПКЛ. В совокупности со своевременной разведкой повреждений этим же БПЛА будет решаться задача устойчивости и надежности связи. При этом могут применяться популярные беспилотный типа «летающее крыло», самолетного типа, которые распространены и часто используются в зоне боевых действий.



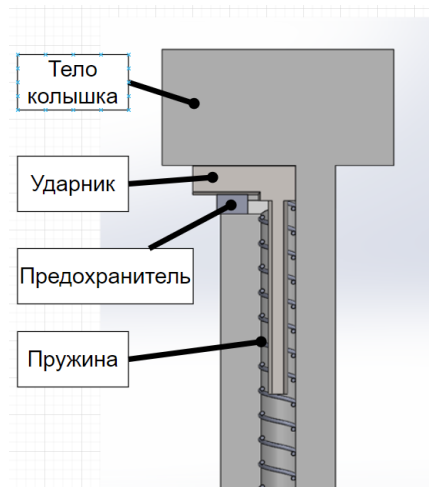


Рис. 7. Устройство кольца

Главным фактором в боевых действиях является время. Именно по данному фактору можно оценить эффективность данного устройства. Длина трассы ПКП от одного НУП до другого не превышает 15 км. Средняя скорость движения коптерных БПЛА 40 км/ч, средняя скорость пешего марша – 4 км/ч. Предположим, что в ходе рекогносцировки понадобится разместить все 16 колец, при этом беспилотнику необходимо будет вернуться обратно. Тогда время выполнения задачи беспилотником будет 0,75 часа, а группой рекогносцировки – 3,75 часа, что показывает огромный выигрыш во времени, а также по личному составу, так как теперь возможно участие всего одного человека – опытного офицера.

Однако стоит отметить и недостатки устройства:

- большая масса (примерно 5 кг для устройства с 16 кольцами), что ограничивает количество беспилотников, способных поднять устройство;
- пригодность использования только с коптерными беспилотниками;
- стоимость колец и сложность их производства (этот недостаток можно исключить при использовании простых колец, сделанных из подручных средств, однако эффективность фиксации тогда будет значительно хуже);
- слабое закрепление колец в грунт.

Боевые действия в зоне проведения специальной военной операции показывают эффективность применения и расширение спектра действий БПЛА. Потому необходимо внедрение новых устройств, крепящихся к беспилотникам, стратегий использования БПЛА. Устройство, предложенное в данной статье, для выполнения задач рекогносцировки демонстрирует огромный выигрыш во времени выполнения боевой задачи. Это может стать преимуществом в ведении войны. Особенно актуально это для нынешнего этапа, когда возникают задачи создания новой сети связи на новых территориях.

#### Библиографический список

1. Зырянов Д. А. Использование БПЛА для выбора трассы при строительстве полевой кабельной линии связи // Семьдесят пятая Международная студенческая научная конференция ГУАП (18–22 апреля 2022 г.): сб. докл.: в 4 ч. Ч. 1: Технические науки. СПб.: ГУАП, 2022. С. 333–335.
2. Николаев В. Б., Рубан Е. Н., Колупаев О. В. Развертывание и эксплуатация полевых кабельных линий связи: учеб. пособие. СПб.: ГУАП, 2021. 163 с.

3. Какие беспилотники используют армии России и Украины: полный список // ФедералПресс. URL: <https://fedpress.ru/article/3119779> (дата обращения: 10.04.2023).

4. Правила прокладки кабеля П-274 на различной местности // Студопедия URL: [https://studopedia.ru/12\\_10108\\_pravila-prokladki-kabelya-p-na-razlichnoy-mestnosti.html](https://studopedia.ru/12_10108_pravila-prokladki-kabelya-p-na-razlichnoy-mestnosti.html) (дата обращения: 05.04.2023).

5. Общие положения по планированию развертывания ПКП -296/302. Документы, разрабатываемые в ЛКР при планировании развертывания ПКП -296/302 // Инструкция, руководство по применению. URL: <https://rykovodstvo.ru/exspl/38776/index.html?page=7> (дата обращения: 05.04.2023).

УДК 621.45

**А. С. Иванов**

студент военного учебного центра

**Д. В. Коликов** – начальник учебной части – научный руководитель

### ИССЛЕДОВАНИЕ ВОЗМОЖНОСТИ ИСПОЛЬЗОВАНИЯ НА БОРТУ ВОЗДУШНОГО СУДНА ГИБРИДНОЙ СИЛОВОЙ УСТАНОВКИ

В современном мире происходит непрерывное развитие технологий, которые затрагивают многие отрасли, в том числе и авиацию. В настоящее время тенденции развития авиационной отрасли нацелены на обеспечение безопасности полётов, сокращение стоимости одного самолётового вылета, повышение экологичности, одним из направлений решения данных задач является использование электрических или гибридных двигателей на борту летального аппарата.

Гибридные силовые установки (ГСУ) – это система, сочетающая в себе два или более источника энергии для создания тяги. Можно выделить следующие схемы построения гибридных силовых установок:

- параллельная система;
- последовательная система.

В параллельной системе (рис. 1) вращение воздушного винта осуществляется с помощью электрического и теплового двигателя, которые могут работать как совместно, так и раздельно. Реализация параллельной системы предусматривает использование механического соединения, позволяющего объединить в единую систему тепловой двигатель, электродвигатель и воздушный винт.

Достоинствами параллельной системы являются:

- возможность переключения между силовыми установками, что позволяет снизить расход топлива и повысить экологические характеристики воздушного судна;
- компактность, что позволяет снизить массу и повысить маневренные возможности воздушного судна;
- автоматическая регулировка мощности силовой установки в зависимости от нагрузки, что позволяет экономить топливо и поддерживать эффективную работу.



*Рис. 1. Структурная схема параллельной системы гибридной силовой установки летального аппарата*

В последовательной системе (рис. 2) вращение воздушного винта осуществляется с помощью электрического двигателя, а тепловой двигатель используется для работы бортового электрогенератора. Вырабатываемая электроэнергия используется для питания всех потребителей на борту воздушного судна, в том числе и электродвигателя, который создает необходимую тягу и для зарядки бортовых аккумуляторных батарей. Последовательная система отличается большой гибкостью, что значительно расширяет диапазон применения данной системы в авиационной отрасли.

Одним из преимуществ последовательной системы является то, что тепловой двигатель внутреннего сгорания будет работать с максимальной термодинамической эффективностью. Данная система проста в реализации и позволяет освободить приводную систему от использования сложных и неэффективных редукторов.

Однако в случае с последовательной системой самым большим недостатком является тот факт, что в ней необходимо иметь сложную систему для преобразования механической энергии в электрическую.

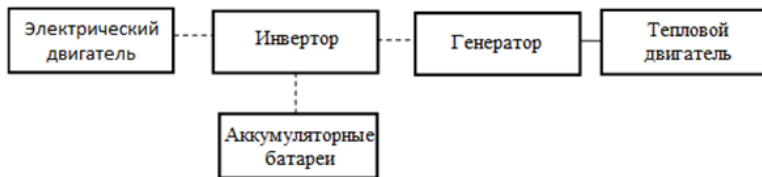


Рис. 2. Структурная схема последовательной системы гибридной силовой установки летательного аппарата

Для проведения исследования выбрана параллельная система, которая в сравнении с последовательной, является наиболее подходящей для использования на борту малых воздушных судов.

В качестве исследуемого воздушного судна выбран самолёта малой авиации Ан-2 с реализацией гибридной силовой установки на базе звёздообразного 9-цилиндрового двигателя АШ-62ИР и электродвигателя REB-90 производства компании «MGM COMPRO». Выбор электродвигателя осуществлен на основе анализа условий совместного использования электрического и теплового двигателя, с учетом технических характеристик.

В процессе исследования необходимо выполнить расчёты следующих характеристик:

– индикаторной мощности по условию:

$$N_i = p_i \frac{V_h * n * i}{900}, \quad (1)$$

где  $p_i$  – среднее индикаторное давление, кг/см<sup>2</sup>,  $n$  – число оборотов цилиндра, об/мин,  $i$  – число цилиндров двигателя,  $V_h$  – рабочий объем цилиндра, дм<sup>3</sup>.

– индикаторного КПД по условию:

$$\eta_i = \frac{632N_i}{G_T * H_u}, \quad (2)$$

где  $N_i$  – индикаторная мощность, л.с.;  $G_T$  – часовой расход топлива, кг/ч.;  $H_u$  – рабочая теплотворная способность топлива, кал/кг.

– эффективной мощности по условию:

$$N_e = N_i - N_T - N_H, \quad (3)$$

где  $N_T$  – мощность трения, л.с.,  $N_H$  – мощность нагнетателей, л.с.

– эффективного КПД по условию:

$$\eta_e = \eta_i \times \frac{N_e}{N_i}; \quad (4)$$

– удельного эффективного расхода топлива по условию:

$$C_e = \frac{632N_i}{\eta_e * H_u}. \quad (5)$$

Выполнив расчеты, получены следующие значения характеристик для теплового двигателя:

$$N_i = 17 \times \frac{3,31 \times 2200 \times 9}{900} = 1125,4 (\text{л.с.});$$

$$\eta_i = \frac{632 \times 1125,4}{300 \times 10500} = 0,23 = 23\%;$$

$$N_e = 1125,4 - 120 - 70 = 935,4(\text{л.с.});$$

$$\eta_e = 0,23 * \frac{935,4}{1125,4} = 0,19 = 19\%;$$

$$C_e = \frac{632 \times 1125,4}{10500 \times 0,19} = 356,5(\text{кг/лс} \times \text{ч}).$$

Для упрощения расчётов параметров гибридной силовой установки необходимо сделать некоторые допущения. Так как электродвигатель и двигатель внутреннего сгорания будут связаны между собой через редуктор, передаточное число которого возьмем равное 1.

Расчеты показали, что эффективная мощность гибридной силовой установки будет иметь следующие значения, с учетом эффективно мощности электрического двигателя ( $N_{el}$ ).

$$N_{el} = \frac{80000}{735} = 108,8(\text{л.с.}).$$

$$N_{acy(e)} = N_e + N_{el} = 935,4 + 108,8 = 1044,2(\text{л.с.}).$$

Значения эффективного КПД и удельного эффективного расхода топлива для гибридной силовой установки имеют следующие значения.

$$\eta_e = 0,23 \times \frac{1044,2}{1125,4} = 0,21 = 21\%;$$

$$C_e = \frac{632 \times 1125,4}{10500 \times 0,21} = 322,6(\text{кг/лс} \times \text{ч}).$$

Таким, образом на основе проведенного исследования реализации системы гибридных силовых установок для летательных аппаратов, произведены расчёты, которые в свою очередь показали, что внедрение гибридного двигателя на борт летательного аппарата позволит добиться сокращения расхода топлива, но негативно скажется на общей массе воздушного судна. Разработка и применение гибридной силовой установки на борту летательного аппарата могла бы существенно снизить стоимость одного вылета, что отразилось бы на сфере коммерческих рейсов гражданской авиации из-за уменьшения стоимости одного вылета. Данная технология может стать основой для улучшения условия развития и обеспечения безопасности малой авиации в России.

#### Библиографический список

1. Старцев Н. И., Виноградов А. С., Новиков Д. К. Конструкция и проектирование авиационных двигателей и энергетических установок. Самара: Изд-во Самар. гос. аэрокосм. ун-та, 2013. 82 с.
2. Захарченко В. С., Селиванов О. Д., Сунцов П. С., Рахманкулов Д. Я. Расчетные исследования по оценке эффективности применения электрической силовой установки на базе топливных элементов применительно к легкому вертолету // Вестник УГАТУ. 2018. Т. 22. № 1. С. 83–89.
3. Аль-Дарабсе А. М. Ф., Маркова Е. В., Денисова Т. В. Использование гибридного электрического летательного аппарата для скайдайвинга // Российский электронный научный журнал. 2019. № 3(33). С. 11-42.
4. Поиск нового – гибридные силовые установки. URL: <https://aviation21.ru/poisk-novogo-gibridnye-silovye-ustanovki/> (дата обращения: 22.03.2023).

УДК 623.76

**В. А. Пластунов**

студент кафедры радиотехнических систем

**А. В. Старостин** – кандидат технических наук, доцент – научный руководитель

**ИДЕНТИФИКАЦИЯ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ  
НА ОСНОВЕ РЕЗУЛЬТАТОВ ТРАЕКТОРНЫХ ИЗМЕРЕНИЙ**

Обнаружение БПЛА без вскрытия факта полета, траектории его движения, других значимых характеристик, важных для целеуказания средству противодействия, противодействие БПЛА невозможно.

Такие цели могут обнаруживаться современными средствами разведки с использованием существующих алгоритмов селекции целей. При этом, как показано в работах, для современных средств обнаружения наиболее сложными целями являются именно малогабаритные и малоскоростные БПЛА [1].

Для повышения достоверности результата классификации воздушного объекта (ВО) оптимально использовать метод сличения радиолокационного портрета цели с эталоном. При этом следует отметить, что не все источники РЛИ позволяют определять спектральный портрет цели, а задача определения типа (класса) воздушного объекта не снимается. Соответственно решение этих задач возлагается на человека, исходя из его субъективной способности анализа исходной информации. Определённые затруднения в принятии решений о типе БПЛА возникают в случаях неопределённости (неполноты) входной информации об анализируемом объекте. В определенной ситуации лицо принимающее решение (ЛПР), используя свой опыт и знания, дополняет нехватку объективной информации своими прогнозами. В этой деятельности ЛПР, могут оказать ценную помощь специальные аналитические методы поддержки принятия решений.

Предлагается следующий подход к решению задачи идентификации БПЛА на основе результатов траекторных измерений, суть которого заключается в анализе формируемых гипотез принадлежности БПЛА к соответствующему классу, формируемых методом интервальных оценок.

Для принятия решения о типе БПЛА необходимо выполнить выборку из базы знаний параметров БПЛА, в соответствии с результатами траекторных измерений. Элементами базы знаний ( $\Pi_i; T_j$ ) являются числовые аргументы, определяющие степень уверенности нахождения объекта в точке, характеризующей положение объекта в пределах измерения ( $S_{\Pi_i}^{T_j}$ ), соответствующие варианту  $\Pi_i$ , при условии  $T_j$ , рассчитываемых с учетом весовых коэффициентов на основе математического метода оценок Фишборна [2]. База знаний представлена в табличном виде (табл. 1), строками которой выступают параметры с учетом ошибок измерения ( $p_i \pm \Delta p_i$ ), анализируемого параметра  $\Pi_i$ , а столбцами – анализируемые типы объектов  $T_j$ .

Таблица 1

База знаний

$\Pi_i$	$T_1$	$T_2$	$T_3$	...	$T_j$
$p_1$	$S_{\Pi_1}^{T_1}$	$S_{\Pi_1}^{T_2}$	$S_{\Pi_1}^{T_3}$	...	$S_{\Pi_1}^{T_k}$
$p_2$	$S_{\Pi_2}^{T_1}$	$S_{\Pi_2}^{T_2}$	$S_{\Pi_2}^{T_3}$	...	$S_{\Pi_2}^{T_k}$
...	...	...	...	...	...
$p_i$	$S_{\Pi_i}^{T_1}$	$S_{\Pi_i}^{T_2}$	$S_{\Pi_i}^{T_3}$	...	$S_{\Pi_i}^{T_k}$

В качестве проверок  $\Pi_i$  выступают параметры, которыми характеризуются БПЛА (в частности высота, скорость, эффективная отражающая поверхность и т. д.). Результирующим значением элементов выборки является результат нормировки  $S_{\Pi_i}$  для каждого типа БПЛА (табл. 2).

Таблица 2

**Результирующие значения выборки**

Значение $\Pi_i$	Степень уверенности $S_{\Pi_i}^{T_k}$
$p_1$	$S_{\Pi_i}$
...	...
$p_i$	$S_{\Pi_i}$

Входная информация в виде зарегистрированных значений траекторных измерений параметров БПЛА поступает на вход системы принятия решения, что позволяют осуществить выборку значений из результирующих таблиц базы знаний, тем самым формируется интервалы выборки по правилу:

$$\Pi_i \text{изм} \pm \Delta \Pi_{\text{ош}}, \tag{1}$$

где  $\Pi_i \text{изм}$  – измеренное значение параметра  $\Pi_i$  проверки,  $\Delta \Pi_{\text{ош}}$  – ошибка измерения параметра  $\Pi_i$  проверки.

В результате формируется множество значений  $S_{\Pi_i}^{T_k}$  для каждого типа БПЛА, которое представляется в виде:

$$\Gamma = \{S_{\Pi_i}^{T_1}, S_{\Pi_i}^{T_2}, \dots, S_{\Pi_i}^{T_k}\}, \tag{2}$$

где  $(S_{\Pi_i}^{T_1}, S_{\Pi_i}^{T_2}, \dots, S_{\Pi_i}^{T_k})$  – интервал, в пределах которого принимается решение о существовании гипотезы  $\Gamma$ .

Множество взаимно исключающих гипотез, определим согласно теории Демпстера-Шеффера (ТДШ) по условию [3]:

$$N_{\Gamma} = 2^j \tag{3}$$

где  $j$  – количество типов объекта.

При анализе рассчитанного интервала выборки выбираются те гипотезы, которые имеют смысл при заданных условиях проверки  $\Pi_i$ , путем анализа интервалов степеней уверенности. Полученные множества позволяют осуществить формирование гипотез о выборке типа БПЛА:

$$\Gamma_{\partial} = \{T_1; T_2; \dots; T_k\}. \tag{4}$$

Гипотеза  $\Gamma_{\partial}$  трактуется следующим образом: в интервале  $(S_{\Pi_i}^{T_1}, S_{\Pi_i}^{T_2}, \dots, S_{\Pi_i}^{T_k})$   $\Pi_i$  проверки имеют место следующие типы  $(T_k)$ .

Для каждой гипотезы  $\Gamma_{\partial}$ , рассматриваемой на интервале  $\{T_1; T_2; \dots; T_k\}$ , рассчитывается субъективная вероятность  $P(\Gamma_{\partial})$ :

$$P(\Gamma_{\partial}) = \frac{M^{\Gamma_k} - M^{\Gamma_s}}{\sum_1^n M^{\Gamma_k} - M^{\Gamma_s}} \tag{5}$$

где  $M^{\Gamma_k} - M^{\Gamma_s}$  – величина подынтервала, в пределах которого принимается решение о существовании гипотезы  $\Gamma_j$ ;  $\sum_1^n M^{\Gamma_k} - M^{\Gamma_s}$  – сумма величин подинтервалов всех существующих гипотез.

Подынтервалы, на которых гипотезы не определяются, в расчет не берутся.

Рассчитанные по формуле (5) значения субъективных вероятностей гипотез  $P(\Gamma_j)$  заносятся в столбцы проверки  $\Pi_i$  табл. 3. Для всех остальных гипотез значения субъективных вероятностей  $P(\Gamma_j)$  будут иметь нулевые значения.

Таблица 3

**Результаты расчета**

	$\Pi_1$	$\Pi_2$	$\Pi_3$	...	$\Pi_i$
$\Gamma_0$	$P_r(0;1)$	$P_r(0;2)$	$P_r(0;3)$	...	$P_r(0;i)$
$\Gamma_1$	$P_r(1;1)$	$P_r(1;2)$	$P_r(1;3)$	...	$P_r(1;i)$
...	...	...	...	...	...
$\Gamma_j$	$P_r(n;1)$	$P_r(n;2)$	$P_r(n;3)$	...	$P_r(n;i)$

По данным табл. 3 вычисляется обобщенный показатель каждой гипотезы  $P(\Gamma_j)$  по условию:

$$P(\Gamma_j) = \sum_1^i P_r(n;i) \tag{8}$$

Для каждой ненулевой гипотезы рассчитываются ее субъективную вероятность по следующей формуле:

$$P_{Tj} = \frac{P(\Gamma_j)}{\sum_1^n P(\Gamma_j)} \tag{9}$$

Решение о типе беспилотного летательного аппарата принимается по условию:

$$P_{Tj} = \operatorname{argmax}_{P_{Tj}} \tag{10}$$

Таким образом, предложенный подход позволяет решить задачу определения типа (класса) беспилотного летательного аппарата (БПЛА) на основе результатов траекторных измерений, полученных по данным радиолокационных систем. При этом на достоверность результата не оказывает значительное влияние субъективный фактор экспертного оценивания на этапе формировании базы знаний, за счет использования методики интервального оценивания.

**Библиографический список**

1. Макаренко С. И. Протivoдействие беспилотным летательным аппаратам: монография. СПб.: Научное издание, 2020. 204 с.
2. Сигал А. В. Использование последовательностей Фишборна в подходящем моделировании, используемом для выборочных данных // Бизнес-информатика. 2019. Т. 15. № 4. С. 50–60.
3. Современные направления развития и области приложения теории Демпстера-Шафера / В. К. Иванов, Н. В. Виноградова, Б. В. Палюх, А. Н. Сотников // Искусственный интеллект и принятие решений. 2018. С. 32-42.



УДК 629.762.2

**В. А. Радионов**

студент военного учебного центра

**О. Д. Кудряшев**

студент военного учебного центра

**П. М. Безняков** – кандидат технических наук, доцент военного учебного центра – научный руководитель

## СОСТОЯНИЕ И ПЕРСПЕКТИВЫ ПРОЕКТА ПО СОЗДАНИЮ СТРАТЕГИЧЕСКОЙ РАКЕТЫ LGM-35A

### Введение

Создание новых ракетных систем для обеспечения национальной безопасности является одной из основных задач государственной политики США. В данном докладе рассматривается причина создания новой ракеты LGM-35A и отказа от использования LGM-30. Первые ракеты LGM-30 Minuteman («Минитмен») были разработаны в 1960-х годах и используются в настоящее время в качестве стратегического ядерного оружия сдерживания в вооруженных силах США.

Однако в последнее время технологии сильно шагнули вперед и старые ракеты США стали уступать новым моделям ракетных систем стран вероятных противников. Кроме того, эксплуатация старых ракет требует значительных затрат на обслуживание и модернизацию, что делает их экономически неэффективными.

В связи с этим руководством США было принято решение о создании новой ракетной системы LGM-35A. Она будет разработана с использованием новейших технологий и обладать значительно более высоким боевыми и эксплуатационными характеристиками, чем ее предшественница.

Новая ракета будет способна преодолевать защитные системы противоракетной обороны противника и обеспечивать стратегическую стабильность США в условиях возможной ядерной угрозы. Создание новой ракеты LGM-35A приведет к отказу от использования LGM-30.

### История предшественника LGM-35A

LGM-30 Minuteman – это ракетная система, созданная в 1960-х годах для использования в качестве стратегического ядерного оружия. Она была разработана для замены устаревших ракет LGM-25 Titan.

Ракета LGM-30 была разработана компанией Boeing в 1958 году, а первый успешный запуск состоялся в 1961 году. Название «Minuteman» было дано в честь минитменов, людей из вольного ополчения североамериканских колонистов, которые получили романтический образ благодаря активному участию в Семилетней войне и Войне за независимость США [1].

Система LGM-30 состоит из трех основных компонентов: ракеты, подземной пусковой установки и командного пункта управления. Ракета имеет дальность полета до 13000 км и может доставлять ядерные боеголовки с точностью до 200 метров. Всего было выпущено около 1000 ракет этой модели.

Одним из ключевых преимуществ системы LGM-30 ее высокая готовность к запуску. Ракеты могут быть запущены в течение нескольких минут после получения приказа, что делает их эффективным средством для ответа на внезапные угрозы.

LGM-30 Minuteman остается в истории США как одна из самых значимых и эффективных ракетных систем. Ее высокая готовность к запуску и точность доставки ядерных боеголовок позволяла обеспечить надежную защиту страны в период холодной войны.

### Процесс разработки новой ракеты LGM-35A

Первые идеи о замене устаревших ракет возникли в США ещё в начале 2010-х годов, однако история не выходила за рамки бумажной волокиты. Во многом это было обусловлено самоуспокоением американцев на счёт российского ядерного потенциала. С другой стороны, это было очень дорого – несколько десятков миллиардов взять просто так было неоткуда.

В 2010 году Коалиция по межконтинентальным баллистическим ракетам, состоящая из законодателей штатов, в которых размещены ядерные ракеты, заявила президенту Обаме, что они не поддержат ратификацию Нового договора СНВ с Россией, если Обама не согласится обновить ядерную триаду США:

ядерные ракеты, которые могут запускаться с суши, моря и воздуха. В письменном заявлении президент Обама согласился «модернизировать или заменить» все три элемента триады [4].

Запрос предложений по разработке новых ракет был сделан Центром ядерного оружия ВВС США в июле 2016 года. За контракт боролись корпорации Boeing и Northrop Grumman [1].

В августе 2017 года военно-воздушные силы заключили с Boeing и Northrop Grumman трехлетние контракты на разработку. Работы шли в рамках эскизного проектирования под шифром Ground Based Strategic Deterrent (Стратегический «сдерживатель» наземного базирования) – GBSD. Одна из этих компаний должна была быть выбрана в 2020 году для производства ракет.

25 июля 2019 года Boeing объявил, что не будет подавать заявку на участие в программе, сославшись на недавнее приобретение Northrop компании Orbital ATK (ныне Northrop Grumman Innovation Systems), поставщика твердотопливных ракетных двигателей Boeing. Northrop подписала соглашение о защите конфиденциальных данных Boeing после приобретения Orbital ATK. С тех пор военно-воздушные силы прекратили финансирование проекта Boeing, оставив Northrop Grumman единственным участником торгов.

С декабря 2019 года именно эта компания официально признана головным разработчиком преемника Minuteman. 8 сентября 2020 года Министерство военно-воздушных сил заключило с Northrop Grumman контракт на разработку межконтинентальной баллистической ракеты GBSD [2]. Первые ракеты, получившие громкое имя LGM-35A Sentinel («Страж»), должны быть готовы к 2029-2030 годам и останутся на вооружении до 2075 года.

Компания-исполнитель с привлечением ряда субподрядчиков начала полноценную разработку нового ракетного комплекса и его отдельных компонентов. Идут активные работы, но их подробности остаются неизвестными. При этом официальные лица утверждают, что программа продолжается в соответствии с графиком и будет завершена в установленные сроки.

В настоящее время ведется разработка боевого блока Mk 21A, которая дошла до стадии летных испытаний. В ближайшее время опытные блоки должны показать свои возможности и подтвердить расчетные характеристики. Однако процесс их проверки уже столкнулся с непредвиденными трудностями [1]. ВВС США сообщили о проведении испытательного запуска нового боевого блока Mk 21A, и, вероятно, это было первое такое мероприятие. Пуск состоялся 6 июля 2022 года на авиабазе Ванденберг (шт. Калифорния) и осуществлялся при помощи ракеты-носителя Minotaur II+ [1]. Сообщается, что пуск оказался неудачным. Всего через 10-11 сек. после старта ракета-носитель взорвалась. Соответственно опытный блок Mk 21A не смог выйти на расчетную траекторию и выполнить полетное задание. Его обломки вместе с фрагментами ракеты упали вблизи стартовой площадки. Причины аварии выясняются, но вряд ли они связаны с опытной полезной нагрузкой ракеты.

Авария при запуске боевого блока нарушает график испытаний и негативно сказывается на всем ходе программы GBSD/LGM-35A. В то же время, эти события показывают текущий прогресс в деле разработки нового боевого оснащения. Впрочем, остается неясно, сколько времени займут испытания и доводка боевого блока и сопутствующих изделий [4].

«Стражи» планируется оснащать устаревшими боеголовками W-87, часть которых были сняты с вооружения в 2005 году вместе с ракетами LGM-118A Peacekeeper («Миротворец»). Каждая такая боеголовка несёт ударный потенциал от 300 до 475 килотонн. Сейчас W-87 размещены в носовых отсеках Minuteman III.

Известно, что в дальнейшем LGM-35A Sentinel будет оснащен модернизированной термоядерной ракетной боевой частью W87-1. Она появится только к 2030 году. Вариант W-87-1 тяжело назвать новым – первые работы по этой теме начались ещё в 1988 году. Использование старых наработок обусловлено тем, что за годы холодной войны США построили несколько тысяч различных вариантов боеголовок, которые в боевых условиях не используются, а утилизация и модернизация крайне проблематична [1].

### Заключение

Таким образом, создание новой ракеты LGM-35A это необходимый шаг для обеспечения национальной безопасности США в условиях возможной ядерной угрозы. Отказ от использования старых ракет LGM-30 обусловлен их устареванием и экономической неэффективностью.

**Библиографический список**

1. Федоров Е. Два шага назад, один – вперёд: новая баллистическая ракета LGM-35A Sentinel. URL: <https://topwar.ru/195726-dva-shaga-nazad-odin-vpered-novaja-ballisticheskaja-raketa-lgm-35a-sentinel.html?ysclid=lhnywqiku431205547> (дата обращения: 06.04.2023).
2. Northrop Grumman Corporation. LGM-35A. URL: <https://www.northropgrumman.com/> (дата обращения: 06.04.2023).
3. Airforce-Technology. URL: <https://www.airforce-technology.com/projects/lgm-35a-sentinel-intercontinental-ballistic-missile-usa/> (дата обращения: 06.04.2023).
4. Defense Primer: LGM-35A Sentinel Intercontinental Ballistic Missile URL: <https://crsreports.congress.gov/product/pdf/IF/IF11681> (дата обращения: 06.04.2023).

УДК 621.317.734

**Т. Х. Рахматов**

студент военного учебного центра

**В. В. Котович** – кандидат технических наук, доцент – научный руководитель

### ЦИФРОВОЙ ОММЕТР

В статье рассмотрен принцип измерения сопротивления методом вольтамперметра, предложена структурная схема цифрового омметра и проведён анализ компьютерной модели омметра.

Для измерения сопротивления широко применяются приборы – омметры. Существуют различные методы измерения сопротивления. В данной статье рассматривается омметр на основе микроконтроллера, который измеряет падение напряжения на резисторе  $R_x$  и реализует метод вольтамперметр.

Действие омметров основано на косвенном измерении сопротивления через измерение силы тока, проходящего через измеряемое сопротивление от стабилизированного источника постоянного напряжения [1].

Функциональная схема предлагаемого омметра, представлена на рис. 1. В данной схеме делитель, состоящий из  $R_0$  и  $R_x$ , питается от стабилизированного источника опорного напряжения  $U_0$ . Падение напряжения  $U_{Rx}$  на резисторе  $R_x$  усиливается усилителем с большим входным сопротивлением.

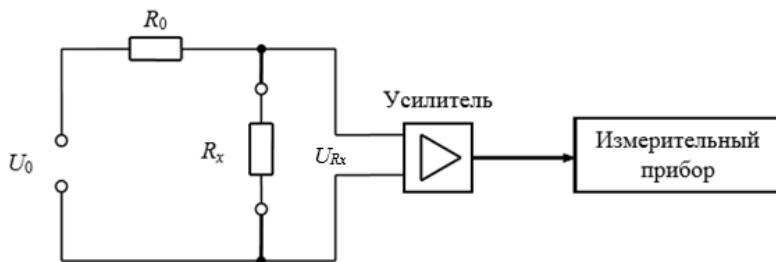


Рис. 1. Схема измерения сопротивления

Выходное напряжение усилителя зависит от величины измеряемого сопротивления  $R_x$ .

Одним из основных условий использования данного метода измерения, является выполнение требования чтобы  $R_x \ll R_0$ . Однако при малом  $R_x$  сказывается влияние сопротивления соединительных проводов, что вызывает потребность непосредственного подключения прибора к измеряемому сопротивлению.

Структурная схема предлагаемого омметра представлена на рис. 2.

С источника стабилизированного напряжения на делитель напряжения подается 5 В.

Делитель напряжения – это цепь, состоящая из двух резисторов, которые соединены последовательно. Самый простой делитель напряжения, состоящий из двух резисторов [2].

Сила тока в делителе напряжения (см. рис. 1) вычисляется по формуле:

$$I = \frac{U_{Rx}}{R_x} = \frac{U_0}{R_0 + R_x}, \quad (1)$$

где  $U_0$  – опорное напряжение в вольтах;  $U_{Rx}$  – падение напряжения на резисторе  $R_x$  в вольтах;  $R_0$ ,  $R_x$  – значение сопротивления резистора в омах.

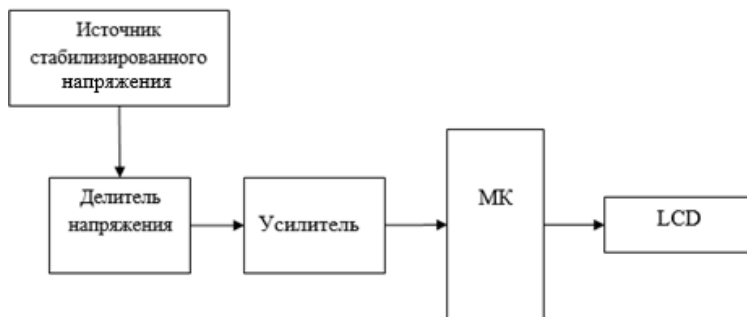


Рис. 2. Структурная схема омметра

Из формулы (1) получим значение сопротивления резистора  $R_x$  :

$$R_x = \frac{U_{Rx} * R_0}{U_0 - U_{Rx}} . \quad (2)$$

Измерение значения падения напряжения  $R_x$  производится с помощью аналогового цифрового преобразования микроконтроллера (МК). Затем по формуле (2) микроконтроллер вычислит значение  $R_x$  и выводит результат измерения на жидкокристаллический экран LCD.

Для микроконтроллера написана программа, алгоритм, который представлен на рис. 3.

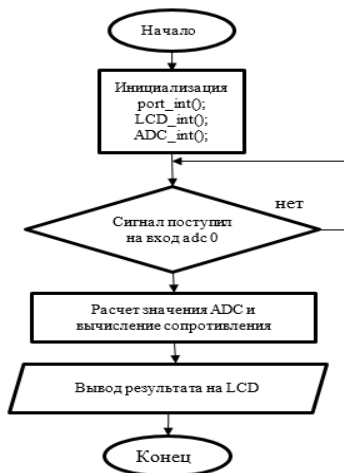


Рис. 3. Алгоритм программы

При включении устройства происходит настройка портов на ввод-вывод данных, инициализация модуля АЦП (ADC) и LCD. После чего начинается бесконечный цикл. В бесконечном цикле микроконтроллер опрашивает канал `adc0`, измеряет значение напряжения и вычисляет значение сопротивления. Полученное значение сопротивления резистора выводится на экран LCD-дисплея.

Для предлагаемого цифрового омметра разработана модель и осуществлено моделирование его работы в программе Proteus VSM. На рис. 4 представлена электрическая принципиальная схема, выполненная в редакторе Proteus. На схеме изображено:

LCD1 – УГО жидкокристаллического экрана LM016L на основе;

U1 – микроконтроллер ATmega16;

U3 – стабилизатор напряжения;

U2 – операционный усилитель;

R1, R2, R3 – резисторы.

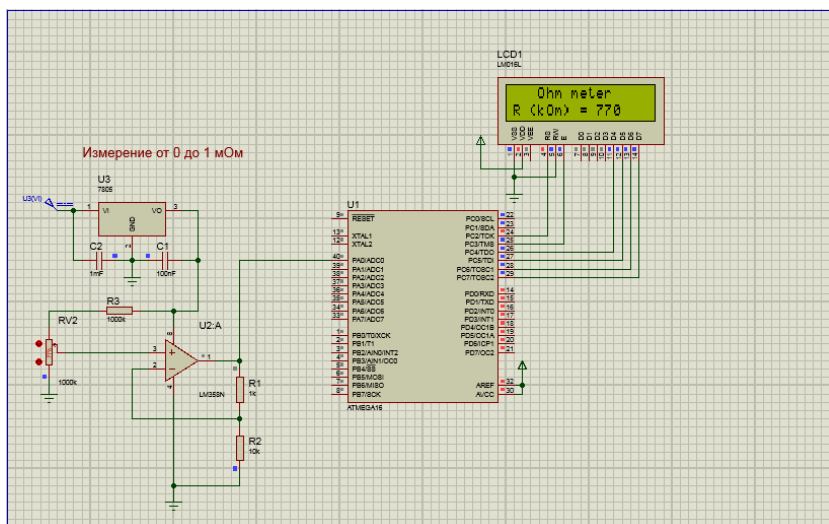


Рис. 4. Схема измерителя в Proteus

Проведённые исследования модели позволяют разработать компактный, недорогой цифровой прибор для измерения сопротивления. В дальнейшем прибор можно практически использовать в учебном процессе при подготовке специалистов-метрологов.

### Библиографический список

1. ГОСТ 16263-70 ГСИ. Метрология. Термины и определения.
2. Котур В. И. Электрические измерения, и электроизмерительные приборы. М.: Энергоатомиздат, 2005.

УДК 004.9; 621.317.08

**А. А. Фабрика**

студент кафедры компьютерных технологий и программной инженерии

**И. И. Божков**

студент кафедры высшей математики и механики

**В. В. Котович** – кандидат технических наук, доцент – научный руководитель

## ОБУЧАЮЩАЯ ПРОГРАММА ДЛЯ ИЗУЧЕНИЯ ПАРАМЕТРОВ ДЕТЕРМИНИРОВАННЫХ СИГНАЛОВ

Современная система обучения специалистов-метрологов невозможна без информационных технологий и интереса к ним со стороны, в первую очередь, обучающихся. Этот интерес мотивирует их к освоению наукоёмких дисциплин, связанных с измерением параметров и характеристик электрических сигналов. Опыт преподавания дисциплин, связанных с измерением параметров и анализом характеристик детерминированных сигналов, у специалистов-метрологов в нашем учебном заведении показывает, что студенты, даже прослушав курсы радиоизмерений, цифровой обработкой сигналов и других аналогичных дисциплин, испытывают сложности при решении теоретических и прикладных задач, связанных с представлением сигналов в спектральной и временной форме и анализа их характеристик. В значительной степени это связано с тем, что полученные знания в этой области носят больше теоретический характер, чем прикладной. Недостаток знаний и навыков в вопросе обработки и анализа электрических сигналов можно частично устранить путём разработки и применения в учебном процессе программно-аппаратного комплекса для изучения параметров детерминированных сигналов [1]. Комплекс позволит, в первую очередь, ознакомить студентов с математическим аппаратом прямого и обратного преобразования Фурье и графическим представлением детерминированных сигналов, используемых в радиотехнике, во временной и частотной областях. Во-вторых, сформировать у студентов навыки измерения и анализа сигналов, поступающих от преобразователя аналогового сигнала в цифровой код (адаптера) на последовательный USB-порт компьютера.

Приступая к разработке такого комплекса, были проанализированы возможности существующих обучающих программ, виртуальных измерительных приборов и ряд пользовательских приложений для анализа спектра сигналов, а также рассмотрены редакторы программ для создания виртуальных приборов и приложений для обработки электрических сигналов.

Результаты анализа некоторых бесплатных, наиболее близких по теме исследования программ представлены ниже.

**Программа SpectrAn.** Эта программа позволяет отображать в графическом редакторе сигналы различных форм, изменять их основные параметры, анализировать спектральную и временную области представления этих сигналов. С точки зрения прикладного обучения, программа имеет ряд недостатков: во-первых, она анализирует только созданные в ней математические модели сигналов, а не работает с реальными электрическими сигналами; во-вторых, в программе не показан математический аппарат, поясняющий процесс преобразования сигнала, например, вычисления его спектра.

**Программа SoundCardOscilloscope.** Это виртуальный измерительный прибор, включающий в себя двухканальный осциллограф, двухканальный генератор низкой частоты и анализатор спектра. Виртуальный генератор позволяет формировать гармонические, прямоугольные, треугольные и пилообразные сигналы в полосе частот от 20 Гц до 20 кГц, т. е. в звуковом диапазоне частот, на выход стереонаушников по двум каналам. Два входа виртуального осциллографа подключены к разъему стереомикрофона компьютера. Таким образом, виртуальный прибор способен анализировать временные и спектральные характеристики сигналов как внутренних, созданных виртуальным генератором, так и внешних, поступающих от звуковой карты компьютера. Недостатками этой программы можно считать то, что виртуальные приборы работают в узком звуковом диапазоне частот и то, что она не предназначена для обучения.

**Программа SpectrumLab** предназначена для генерации гармонических модулированных и немодулированных сигналов, анализа спектра аудио сигнала, поступающего на вход виртуального анализатора спектра через звуковую плату компьютера. Эта программа является универсальным средством для отображения и анализа, поступающего на вход звуковой карты сигнала. Программа может анали-

зировать звуковой поток, поступающий либо на COM-порт компьютера через внешний АЦП, либо поступающий по Ethernet-сетям по протоколам TCP/IP или UDP в компьютер, либо непосредственно из программы аудио проигрывателя WinAmp, минуя звуковую карту. Недостатком программы можно считать тот же, что и для программы SoundCardOszilloscope – она не предназначена для обучения.

**Программа FourierScope.** Это программа позволяет пользователю в графическом редакторе нарисовать радиосигнал любой формы и получить его спектр. Программа включает в себя генератор сигналов, осциллограф и спектральный анализатор. Программа позволяет моделировать сигналы, как простой геометрической формы, так и отображаемые составными функциями, а также применять к ним модуляцию. С помощью быстрого преобразования Фурье программа получает частотный, фазовый и энергетический спектры исследуемого сигнала. Программа предназначена для обучения и ознакомления со спектрами сигналов, наиболее часто используемых в теоретической и практической радиотехнике, и она больше всего подходит в качестве прототипа для разработки обучающего программно-аппаратного комплекса. К недостатку можно отнести то, что в программе не показан математический аппарат и отсутствует теоретический материал, поясняющий студенту процесс преобразования сигнала.

Таким образом, программная часть программно-аппаратного комплекса (ПАК) для изучения параметров детерминированных сигналов различной формы в настоящее время разрабатывается с учётом достоинств и недостатков проанализированных программ. Программа реализуется на языке программирования Python с использованием библиотек scipy, numpy, matplotlib для выполнения различных математических операций (преобразования Фурье) и вывода данных в графическом виде. Для удобства работы пользователя программа имеет простой и интуитивно понятный визуальный оконный интерфейс. Представленный комплекс позволяет обучающимся как создавать и исследовать виртуальные сигналы, так и работать с реальными аналоговыми сигналами из различных источников.

В режиме работы с виртуальными сигналами пользователь может создать сигнал одной из заданных форм с произвольными параметрами. По уже созданному сигналу пользователь может построить графики амплитуды и спектра выбранного сигнала (рис. 1). На рисунке представлен синусоидальный сигнал с амплитудой 1 вольт, частотой 1 Гц. В левом окне изображено временное представление сигнала, в правом – спектральное представление этого же сигнала.

В обучающей программе поддерживается возможность масштабирования формы сигнала по амплитуде и длительности по аналогии с реальным прибором – осциллографом.

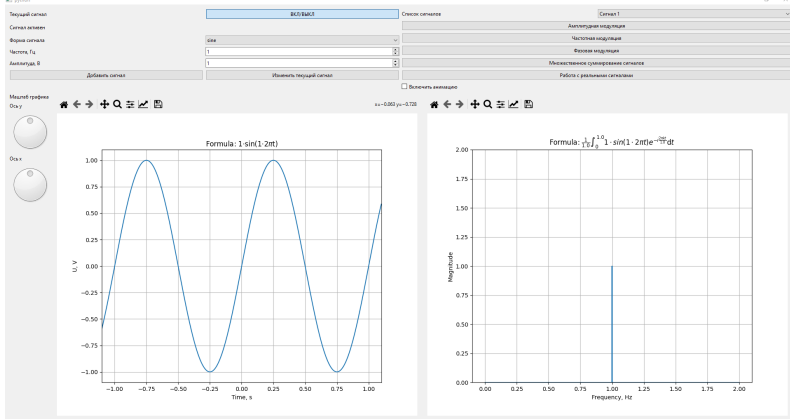


Рис. 1. Главное окно программы

Пользователь также может работать с набором сигналов, выполняя амплитудную, частотную, фазовую модуляцию и множественное суммирование сигналов. На рис. 2, в качестве примера, пред-



ставлен интерфейс окна, отображающего амплитудно-модулированный сигнал. Для других видов модуляции интерфейс окон выглядит аналогично.

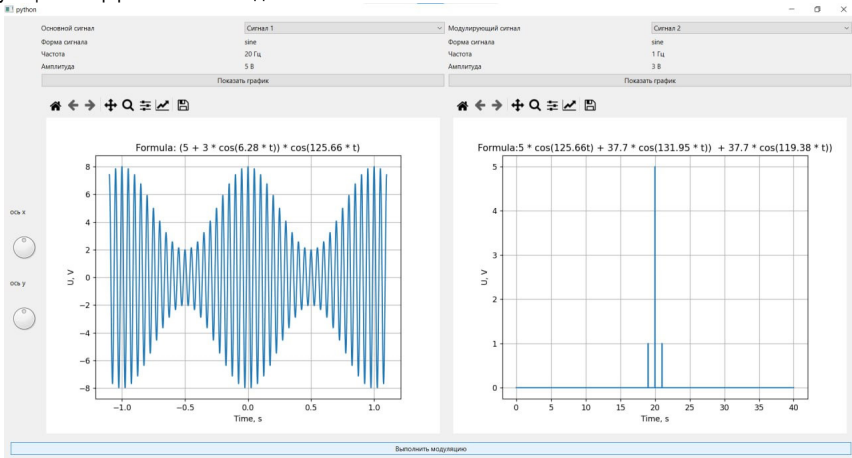


Рис. 2. Окно с амплитудно-модулированным сигналом

Кроме создания различных видов модуляций, комплекс также поддерживает возможность множественного суммирования сигналов. Для выполнения данного действия пользователь последовательно выбирает сигналы, которые хочет просуммировать. Результат суммирования сигналов отображается в окне. На рис. 3 представлен интерфейс окна множественного суммирования сигналов.

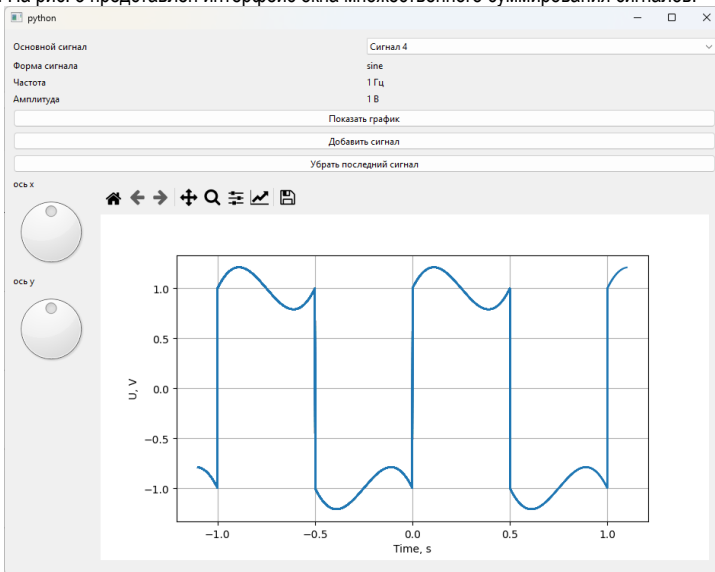


Рис. 3. Окно множественного суммирования сигналов

Работа с реальными сигналами заключается в подаче аналоговых сигналов различных форм от внешних источников на адаптер, преобразование этих сигналов в адаптере в цифровой код и выдача его на компьютер. Затем обучающая программа обрабатывает полученные данные и строит графики временного и спектрального представления сигнала. Адаптер в своём составе содержит микроконтроллер ATmega8, который имеет 10-ти разрядный АЦП и интерфейс UART для передачи цифровых данных. Адаптер дополнительно содержит преобразователь интерфейсов FT232 для согласования данных, поступающих от микроконтроллера к компьютеру.

Дополнительно, обучающая программа поддерживает возможность взаимодействовать с системами проектирования электронных схем, например, с Proteus. Обмен данными между обучающей программой и программой Proteus осуществляется с помощью программы управления виртуальными портами com0com. Кроме того, обучающей программе можно передавать данные через виртуальный порт вручную, используя программу Terminal. Пример взаимодействия с Proteus предоставлен на рис. 4 и 5. На рис. 4 представлена схема адаптера, выполненная в редакторе схем Proteus. Микроконтроллер ATmega8 формирует на выходе TXD цифровой код синусоидального сигнала, который поступает с виртуального порта COM1M на виртуальный вход компьютера через com0com. На рис. 5 представлен результат обработки обучающей программой цифрового кода, поступившего через виртуальный последовательный вход компьютера.

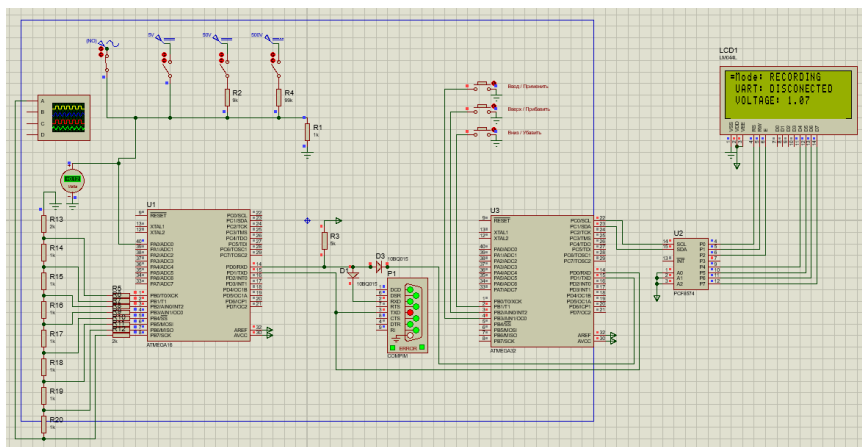


Рис. 4. Схема цепи в Proteus

Для работы с реальными сигналами необходимо подключить источник сигнала к адаптеру, в обучающей программе выбрать соответствующий порт и нажать кнопку «Начать получение сигнала от МК». Для завершения передачи данных необходимо нажать на кнопку «Завершить получение сигнала от МК».

Программа поддерживает два варианта вывода сигнала: «блочный» и «в режиме реального времени». В «блочном» режиме обучающая программа считывает определенное количество точек в специальный буфер. При этом отрисовка изображения сигнала в окне не производится. После заполнения буфера его содержимое отрисовывается в окне программы, а буфер начинает записывать новую порцию данных. В режиме «реального времени» обучающая программа считывает каждое значение сигнала и сразу же изображает его на графике.

Обмен данными между микроконтроллером и компьютером осуществляется с помощью специально разработанного протокола для предотвращения потерь данных. Протокол используется для открытия и закрытия соединения, и для изменения режима работы микроконтроллера.

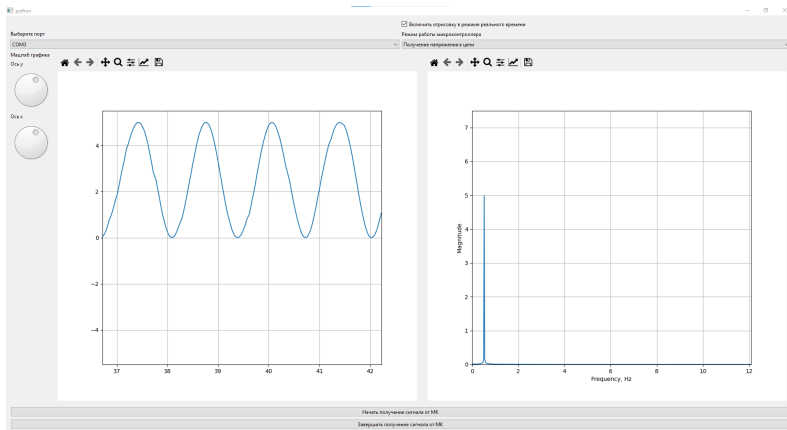


Рис. 5. Получение данных из Proteus

Разработка программной части обучающего программного комплекса является сложной технической задачей и в настоящий момент ещё не закончена. Готовый комплекс позволит автоматизировать процесс обучения студентов, которые интересуются измерением и преобразованием детерминированных сигналов различной формы.

#### Библиографический список

1. Котович В. В. Автоматизация процесса изучения параметров детерминированных сигналов // Радиотехнические, оптические и биотехнические системы. Устройства и методы обработки информации. Третья Всероссийская научная конференция, 14–22 апреля 2022 г.: сб. докладов. Спб.: ГУАП, 2022. 167 с.

УДК 53.082.52

**Д. Д. Федюнин**

студент военного учебного центра

**В. В. Котович** – кандидат технических наук, доцент – научный руководитель

### ЦИФРОВОЙ МАНОМЕТР

В статье рассмотрен принцип действия цифрового манометра, предложена структурная схема цифрового манометра и проведён анализ компьютерной модели измерителя давления на основе датчика давления с токовым выходным сигналом.

Часть измерений, которые выполняются в ходе проведения научных исследований, в промышленности и в сельском хозяйстве, связаны с измерениями давления и расхода различных веществ. Средства измерений, которые контролируют давление, должны соответствовать требованиям ГОСТ 2405-88 [1]. Этот ГОСТ устанавливает нормативные требования к проектированию и эксплуатации приборов измерения давления и методы их измерений. Невыполнение этих требований может привести к серьезным последствиям.

Для измерения избыточного давления используют устройства, которые называют манометрами. Принцип работы манометра заключается в уравнивании измеряемого давления силой упругой деформации трубчатой пружины или более чувствительной двухпластинчатой мембраны, один конец которой запаян в держатель, а другой через тягу связан с трибно-секторным механизмом, преобразующим перемещение упругого чувствительного элемента в круговое движение показывающей стрелки. В настоящее время не существует дешевых, малогабаритных и при этом точных манометров. Поэтому задача разработки такого прибора для измерения избыточного давления остаётся актуальной. Предлагается в качестве манометра использовать цифровой манометр на основе датчика давления, чувствительным элементом, которого является тензодатчик. Принцип работы тензодатчика основан на изменении сопротивления чувствительного элемента под воздействием поступающего на вход давления. Высокая чувствительность и линейная характеристика тензодатчика позволяет создать манометр высокого класса точности и использовать его в качестве эталона. Таким образом, предлагаемый цифровой манометр позволит поверять манометры с более низкими классами точности. Кроме того, он позволяет устанавливать пределы измерений давления и класс точности для поверяемых манометров с помощью кнопок и выводить значения рассчитанной приведенной погрешности поверяемого манометра на жидкокристаллический дисплей.

Структурная схема предлагаемого цифрового манометра представлена на рис. 1.



Рис. 1. Структурная схема цифрового манометра

Структурная схема цифрового манометра состоит из блоков. Выходной ток с датчика давления проходит через низкоомную нагрузку, создавая на ней падение напряжения, пропорциональное измеряемому давлению. Данное падение напряжения измеряется с помощью АЦП микроконтроллера. Результат измерения давления выводится на ЖК-дисплей. В памяти микроконтроллера хранится библио-

тека значений погрешностей, пределов шкалы измерения манометров и их классов точности. Выбор данных из библиотеки осуществляется пользователем в зависимости от типа поверяемого манометра путем нажатия кнопки.

В качестве элементной базы данного устройства предлагается использовать:

- микроконтроллер ATmega16, обладающий малым энергопотреблением, достаточным объемом памяти, быстройдействием и дешевизной;
- жидкокристаллический дисплей LM041L, простой в эксплуатации;
- датчик давления МИДА-ДИ-13П-У2-0.25/16МПа, содержащий в себе тензопреобразователь и усилитель. Датчик в зависимости от давления на входе обеспечивает значение выходного тока в диапазоне от 4 до 20 мА.

Алгоритм работы программы для микроконтроллера представлен на рис. 2.

В соответствии с алгоритмом, при включении устройства происходит настройка портов на ввод-вывод данных, инициализация дисплея, а затем его очистка и инициализация модуля АЦП (ADC).

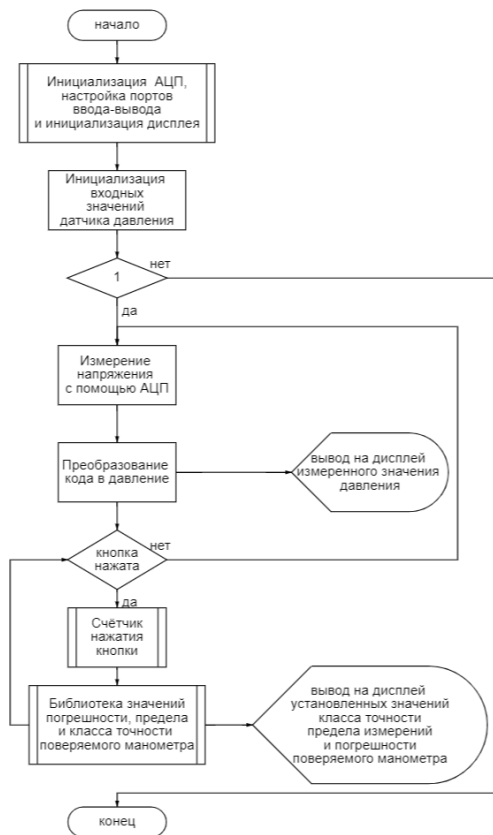


Рис. 2. Алгоритм работы программы

В блоке «Инициализация входных значений датчика» находятся данные о диапазоне измерений датчика и переменных, которые используются при нормировании напряжения в давление.

В подпрограмме «Счетчик нажатия кнопки» находится функция PEREDACHA, с помощью которой происходит смена данных выводимых на дисплей. При нажатии кнопки счетчик увеличивается на единицу.

В блоке «Расчет по формуле» происходит расчет измеренного давления по формуле преобразования выходного аналогового сигнала, измеряемого в вольтах. Затем рассчитанное значение давления выводится на ЖК-дисплей.

В подпрограмме «библиотека значений погрешности, предела и класса точности поверяемого манометра» находятся значения погрешности, предела и класса точности поверяемого манометра. По нажатию кнопки на ЖК дисплее изменяются эти значения в соответствии с поверяемым прибором.

Все необходимые расчеты и моделирование работы цифрового люксметра производилось в программах Proteus 8 Professional.

На рис. 3 представлена упрощенная схемотехническая модель устройства для измерения давления.

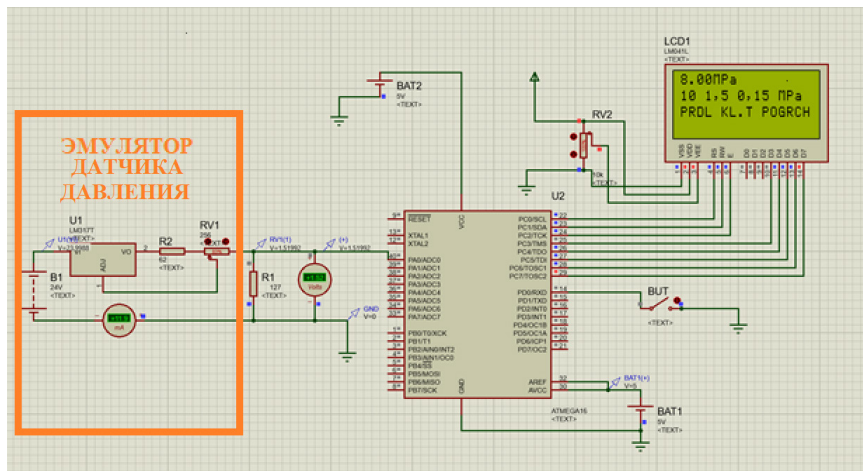


Рис. 3. Модель устройства в программе Proteus 8 Professional

В данной модели вместо датчика давления МИДА-ДИ-13П-У2-0.25/16МПа используется его эмулятор. Подстроечный резистор RV1 номиналом 256 Ом, используется в эмуляторе датчика давления для регулировки выходного тока в диапазоне от 4 мА до 20 мА. На схеме микроконтроллер Atmega16 обозначается как U2, жидкокристаллический дисплей LM041L – LCD1. Кнопка BUT – однокнопочный переключатель, с помощью которого происходит выбор значения погрешности, предела и класса точности поверяемого прибора из библиотеки микроконтроллер.

После построения модели было проведено моделирование работоспособности измерительного устройства, определены диапазоны изменения внешних и внутренних параметров схемы и подобрана элементная база.

Проведённые исследования модели и анализ элементной базы позволяют разработать компактный, точный, недорогой цифровой прибор для измерения давления, который можно использовать в качестве эталона для проверки рабочих манометров. Данный прибор также можно практически использовать в учебном процессе при подготовке специалистов-метрологов.

### Библиографический список

1. ГОСТ 2405-88 Манометры, вакуумметры, мановакуумметры, напорометры, тягомеры и тягонапорометры.

УДК 629.7.052

**М. Н. Фролов**

студент военного учебного центра

**А. П. Григорьев** – старший преподаватель – научный руководитель

### РАЗРАБОТКА И ИССЛЕДОВАНИЕ АЛГОРИТМОВ АДАПТИВНОГО ИНТЕРФЕЙСА «ПИЛОТ-ЛЕТАТЕЛЬНЫЙ АППАРАТ»

В данной работе рассматривается возможность использования на борту летательного аппарата (ЛА) алгоритмов и методов автоматизированной адаптивной настройки системы отображения информации (СОИ) в целях повышения эффективности управления. Целью работы является предложение структурной схемы работы адаптивного интерфейса, а также разработка алгоритмов оценки полетной ситуации и состояния пилота для их дальнейшего использования при настройке СОИ.

Известно, что до 30% всех ошибочных действий, приводящих к летным происшествиям, связано с инженерно-психологическими недостатками бортового оборудования [1]. Следовательно, повышая интуитивность и эффективность взаимодействия пилота с ЛА, можно добиться положительного влияния человеческого фактора на безопасность полета и выполнение полетного задания.

Кроме того, с развитием возможностей современной информационно-измерительной техники проблема противоречия между количеством и формой предоставляемой информации, с одной стороны, и ограничениями по обработке и восприятию этой информации человеком-оператором, с другой, приобретает все более актуальное значение. Такая проблема может быть решена путем разработки адаптивного интерфейса «пилот-ЛА», подстраивающегося под потребности пилота и учитывающего психофизическое состояние человека и полетную ситуацию в конкретный момент времени.

В упрощенном виде классическое взаимодействие пилота и ЛА можно представить так, как показано на рис. 1.

Изображенная на рисунке область, разделяющая человека и ЛА, представляет собой исследуемый интерфейс «пилот-летательный аппарат».

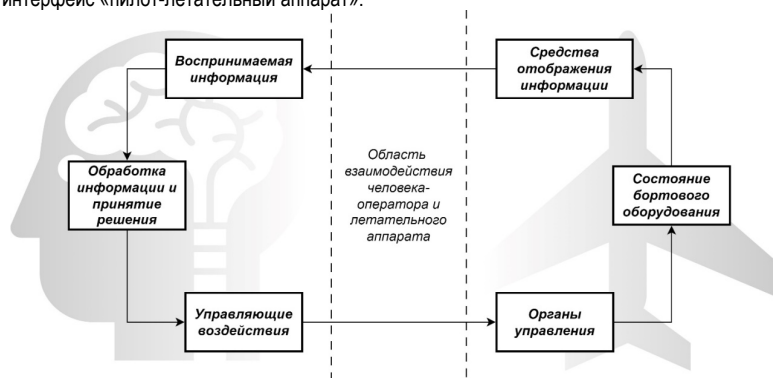


Рис. 1. Информационный контур пилот-ЛА

При разработке адаптивного интерфейса должны учитываться следующие требования:

- интерфейс не остается неизменен в течение всего полета, а подстраивается под полетную ситуацию;
- интерфейс производит мониторинг функционального состояния пилота;
- настройка интерфейса касается объема предоставляемой информации, способов ее представления, а также уровня автоматизации функций;

- информация передается по этому интерфейсу в обоих направлениях, обязательно наличие обратной связи от пилота;
- пилот должен постоянно находиться в контуре управления ЛА;
- в основе реализации должен находиться деятельностный подход, ориентированный на пилота.

**Разработка адаптивного интерфейса.** Структурная схема работы адаптивного интерфейса представлена на рис. 2.

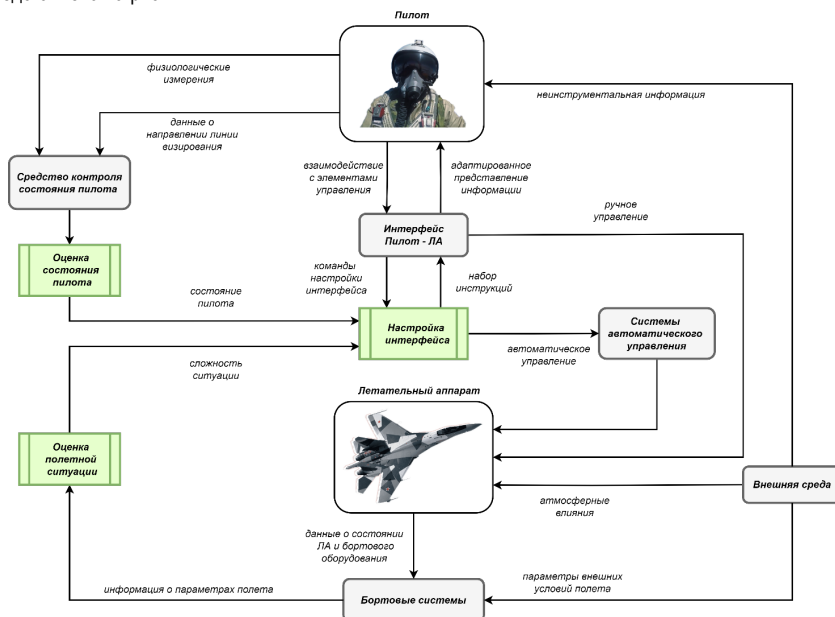


Рис. 2. Структурная схема работы адаптивного интерфейса

Алгоритм оценки полетной ситуации на основе информации, полученной от бортовых систем ЛА, классифицирует текущую полетную ситуацию на четыре типовые категории, характеристика которых приведена в табл. 1.

Таблица 1

**Типовые полетные ситуации для настройки интерфейса**

Полетная ситуация	Характеристика
Аварийная	Характеризуется значительным повышением психофизиологической нагрузки на пилота, ухудшением летных характеристик ЛА. Сопровождается аварийным предупреждением. Возможно потенциальное столкновение с землей или другим летательным аппаратом, а также угроза поражения (для военных ЛА). Приоритет – незамедлительное устранение опасности
Сложная	Характеризуется значительным повышением психофизиологической нагрузки на пилота, а также значительным ухудшением летных характеристик. Сопровождается предупредительной сигнализацией. Приоритет – осуществление своевременных действий пилотом



Полетная ситуация	Характеристика
Усложнение условий полета	Характеризуется незначительным увеличением психофизиологической нагрузки на экипаж. Усложнение условий полета не приводит к необходимости немедленного изменения плана полета и не препятствует его благоприятному окончанию. Усложнение условий может быть связано с полетом в сложных метеоусловиях (СМУ), ночью, на предельно низких или больших высотах
Штатный режим	Спокойный установившийся полет. Полет в автоматическом или ручном режиме. Пилот самостоятельно управляет режимами бортового оборудования. Предусмотрена возможность использовать альтернативные каналы информации.

Оценка полетной ситуации происходит по алгоритму, представленному на рис. 3.

Алгоритм оценки состояния пилота на основе данных, полученных от средства контроля состояния пилота, производит классификацию текущего состояния пилота, определяя его функциональную способность к управлению ЛА, а также возможный уровень загрузки.

Известны системы контроля функционального состояния оператора, реализованные в виде цифрового биометрического браслета (ЦББ) с интегрированным пульсоксиметрическим модулем, проводящим измерения различных параметров организма человека в режиме реального времени [2]. Так, измеряя данные человека о его частоте сердечных сокращений (ЧСС), уровне периферической сатурации крови (SpO2), а также определяя электрическую активность кожи (или кожно-гальваническую реакцию, КГР) как реакцию на стресс, можно с помощью методов статистического анализа данных (например, *t*-критерий Стьюдента для независимых выборок) определять значимые различия в состоянии функциональных систем организма пилота.

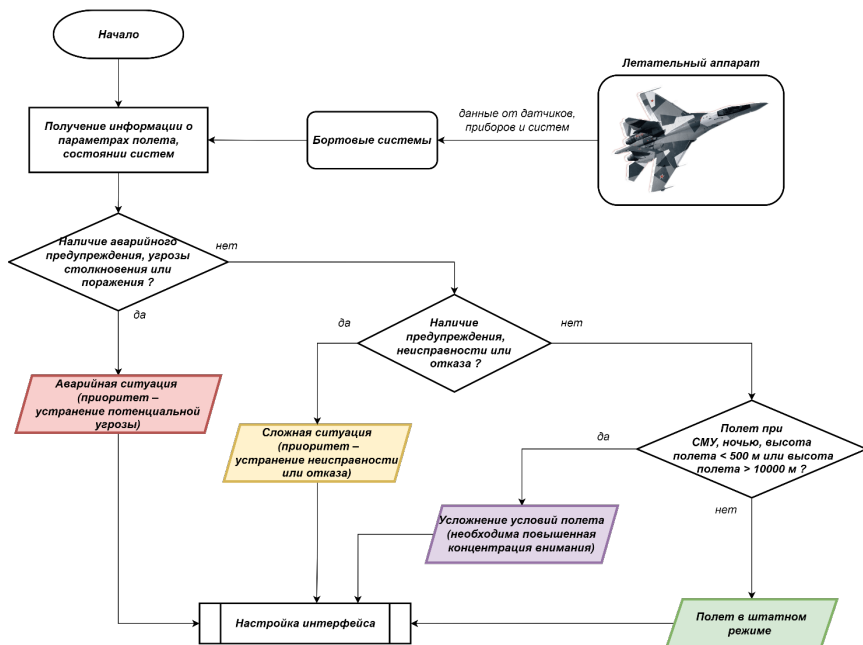


Рис. 3. Блок-схема алгоритма оценки полетной ситуации

После обработки данных, полученных от ЦББ, алгоритм определяет состояние пилота, классифицируя его по следующим категориям: минимальный уровень загрузки – соответствует нахождению всех контролируемых параметров в пределах нормальных значений для данного пилота; повышенный уровень загрузки – свидетельствует о наличии статистически значимых отклонений параметров пилота, возможном проявлении стресса или развивающейся гипоксии (рис. 4).

Дополнительным средством контроля состояния пилота является устройство, регистрирующее направление линии визирования пилота. Это может быть особенно важно для военных летчиков, подвергающихся высоким нагрузкам в полете. Устройство определения линии визирования реализовано в составе штатной наשלемой системы целеуказания (НСЦ) и работает по принципу передачи сигналов между излучателями и приемниками инфракрасного излучения. Кроме основного предназначения – наведения на цель, использование НСЦ позволяет регистрировать продолжительную задержку взгляда пилота в нерабочем положении, что свидетельствует либо о повышенном внимании пилота к объекту, находящемуся вне кабины, либо о неспособности управлять ЛА (например, в случае потери сознания при высоких продольных положительных перегрузках). В таком случае состояние пилота определяется как предельный уровень загрузки, и возникает необходимость получения от пилота обратной связи для подтверждения его участия в процессе управления.

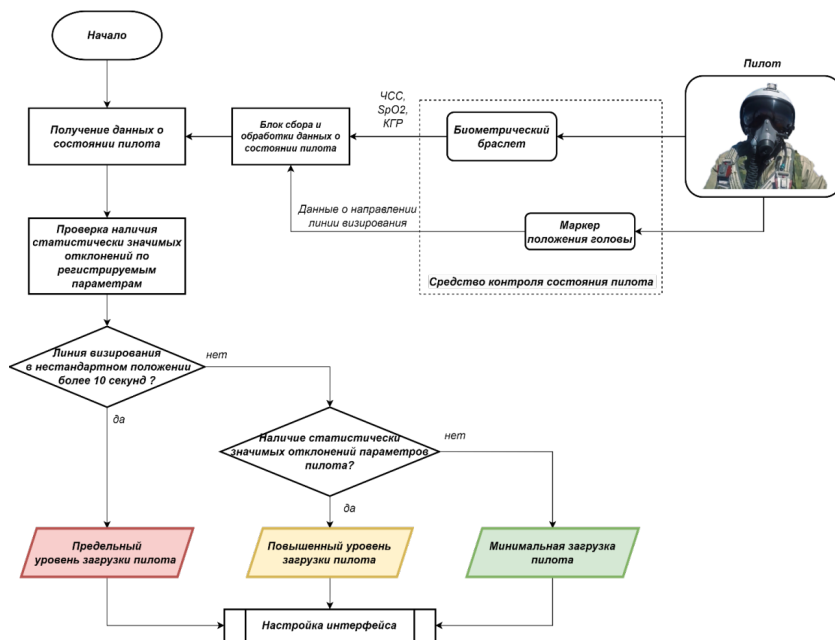


Рис. 4. Блок-схема алгоритма оценки состояния пилота

В данной работе были рассмотрены теоретические предпосылки разработки алгоритмов адаптивного интерфейса «пилот-ЛА», а также представлены конкретные требования, которым должен удовлетворять проектируемый интерфейс взаимодействия между пилотом и ЛА.

В результате, была предложена структурная схема работы адаптивного интерфейса, приведено описание основных ее блоков. Разработанные алгоритмы оценки полетной ситуации и оценки состояния пилота, используемые для адаптации интерфейса, в конечном итоге позволят повысить качество и безопасность управления, а также значительно снизить функциональную нагрузку на пилота ЛА.

**Библиографический список**

1. Лапа В. В. Эволюция методологии обеспечения взаимодействия человека с авиационной техникой. М.: Истоки, 2004. 48 с.
2. Коптев Д. С., Мухин И. Е. Характеристика концепции разработки интегрированных бортовых систем обеспечения безопасности полетов летательных аппаратов, включая системы контроля функционального состояния пилота // Вестник ФГБОУ ВО «Уфимский государственный авиационный технический университет». 2020. № 4. С. 57–68.

## СОДЕРЖАНИЕ

<b>ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММИРОВАНИЕ.....</b>	<b>3</b>
<b>Прикладная информатика.....</b>	<b>3</b>
<i>Барболин С. П., Автушко А. С., Бубнов А. Н., Жаданова К. С., Порядина К. А.</i> РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ КОММУНИКАЦИИ УЧАСТНИКОВ РЫНКА УСЛУГ В СФЕРЕ ОРГАНИЗАЦИИ МЕРОПРИЯТИЙ .....	3
<i>Виноградов Д. Ф.</i> РЕШЕНИЕ ПРОБЛЕМ, ВОЗНИКАЮЩИХ ПРИ КОМПЬЮТЕРНОЙ ПОСТОБРАБОТКЕ ВИДЕОДАНЫХ .....	7
<i>Гарилов А. Р.</i> МИКРОФРОНТЕНДЫ В СОВРЕМЕННОЙ WEB-РАЗРАБОТКЕ .....	13
<i>Господынюк В. В.</i> ИНСТРУМЕНТЫ И ПОДХОДЫ К ТЕСТИРОВАНИЮ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ .....	17
<i>Душеба А. В.</i> СКАНЕР BLOCKCHAIN ТРАНЗАКЦИЙ ДЛЯ СЕТЕЙ НА БАЗЕ EVM .....	20
<i>Душеба А. В.</i> WEB-АВТОРИЗАЦИЯ НА ОСНОВЕ ТЕХНОЛОГИИ BLOCKCHAIN .....	23
<i>Зорин И. А., Жданов Д. В.</i> НЕЙРОННАЯ СЕТЬ ДЛЯ ОБРАБОТКИ ИДИОМАТИЧЕСКИХ ВЫРАЖЕНИЙ ЕСТЕСТВЕННОГО ЯЗЫКА .....	26
<i>Ильиных М. В.</i> АВТОМАТИЗАЦИЯ РАБОТЫ В ГРАФИЧЕСКИХ РЕДАКТОРАХ .....	30
<i>Кирчевская П. В.</i> НЕЙРОННАЯ СЕТЬ ДЛЯ ОПРЕДЕЛЕНИЯ СТЕПЕНИ ЦВЕТОВОЙ СОЧЕТАЕМОСТИ ЭЛЕМЕНТОВ ОДЕЖДЫ .....	33
<i>Коновалов А. Н.</i> ЦИФРОВАЯ РЕАЛИЗАЦИЯ МЕХАНИЗМА ШИФРОВАЛЬНОЙ МАШИНЫ «ЭНИГМА» .....	35
<i>Коптев Д. В.</i> СОЗДАНИЕ WEB-ПРИЛОЖЕНИЯ «ПОМОЩНИК СТУДЕНТА» .....	39
<i>Кулешо М. М.</i> ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТЕЙ В DIGITAL-МАРКЕТИНГЕ .....	43
<i>Летунов И. Ю.</i> ОРГАНИЗАЦИЯ КЛИЕНТ-СЕРВЕРНОГО ВЗАИМОДЕЙСТВИЯ В АВТОМАТИЧЕСКОЙ СИСТЕМЕ ТЕХНИЧЕСКОЙ ДИАГНОСТИКИ МОСТИКОВЫХ СИСТЕМ СУДНА .....	46
<i>Летунов И. Ю.</i> КОМПОНЕНТЫ АВТОМАТИЧЕСКОЙ СИСТЕМЫ ТЕХНИЧЕСКОЙ ДИАГНОСТИКИ МОСТИКОВЫХ СИСТЕМ СУДНА .....	52
<i>Ляхович М. С., Немилостив И. Ю.</i> ФОРМИРОВАНИЕ ДИНАМИЧЕСКИ КОНФИГУРИРУЕМОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ НА ПРИНЦИПАХ ОРГАНИЗАЦИИ VPS/VDS .....	56
<i>Немилостив И. Ю., Ляхович М. С.</i> МЕТОДЫ ИНТЕГРАЦИИ ИНФОРМАЦИОННЫХ СИСТЕМ... ..	59
<i>Пономаренко В. В.</i> МЕТОДЫ ПОЛУЧЕНИЯ ИНФОРМАЦИИ С WEB-РЕСУРСОВ .....	62
<i>Сапунов Д. В.</i> ИНТЕГРАЦИЯ ТЕХНОЛОГИЙ БЛОКЧЕЙН И ОДНОСТРАНИЧНЫХ WEB-ПРИЛОЖЕНИЙ .....	66
<i>Сенина Я. И., Лацко А. В.</i> ПРОГРАММНЫЙ РЕНДЕРИНГ МЕТОДОМ ТРАССИРОВКИ ЛУЧЕЙ ... ..	69
<i>Сигачева Д. Н.</i> ИССЛЕДОВАНИЕ ВЛИЯНИЯ ПАРАМЕТРОВ СТРИМИНГОВЫХ СЕРВИСОВ НА ВОСПРИЯТИЕ ЧЕЛОВЕКОМ МУЗЫКАЛЬНЫХ ПРОИЗВЕДЕНИЙ .....	72
<i>Челурнова С. С.</i> РЕАЛИЗАЦИЯ АЛГОРИТМА ПОДБОРА СПАРРИНГ-ПАРТНЕРА В СПОРТИВНЫХ ЕДИНОБОРСТВАХ .....	74
<i>Чулочникова Н. И.</i> ПРОГРАММНЫЕ СРЕДСТВА СОЗДАНИЯ ВИДЕОКОНТЕНТА В СФЕРЕ EDUTECH .....	78
<i>Швайко А. С.</i> СОЗДАНИЕ ОДНОРАНГОВОЙ ЛОКАЛЬНОЙ СЕТИ С СЕТЕВЫМ ХРАНИЛИЩЕМ .....	81

<b>Информационные системы и технологии .....</b>	<b>88</b>
<i>Буслова О. А.</i> МОДЕЛИРОВАНИЕ ЛОГАРИФИЧЕСКИ НОРМАЛЬНОГО АНИЗОТРОПНОГО ПОЛЯ .....	88
<i>Ветрова Ю. В.</i> ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ПРОГРАММНЫХ РЕШЕНИЙ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ В РАМКАХ СВОБОДНО РАСПРОСТРАНЯЕМЫХ АНАЛИТИЧЕСКИХ ПЛАТФОРМ .....	93
<i>Григорьева Е. Д.</i> ПРИМЕНЕНИЕ ЦИФРОВЫХ ТЕХНОЛОГИЙ В СФЕРЕ ОБРАЗОВАНИЯ .....	99
<i>Закиева К. А.</i> ЗАЩИТА ВЕБ-ПРИЛОЖЕНИЙ ОТ ХАКЕРСКИХ АТАК И УГРОЗ СЕТЕВОЙ БЕЗОПАСНОСТИ .....	102
<i>Зацепин И. С.</i> ТЕХНОЛОГИИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ПРИ ПРОЕКТИРОВАНИИ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ .....	106
<i>Знаменская Д. Д.</i> СРАВНЕНИЕ СРЕДСТВ FLUTTER И XAMARIN ДЛЯ СОЗДАНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА В РАЗРАБОТКЕ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ .....	109
<i>Князева А. А.</i> СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ КЛАСТЕРИЗАЦИИ КАТЕГОРИАЛЬНЫХ ДАННЫХ .....	113
<i>Комаров И. М.</i> ОСОБЕННОСТИ РАЗРАБОТКИ ПРОГРАММНОГО БОТА ДЛЯ ПОМОЩИ СОТРУДНИКАМ КОМПАНИИ, ИСПОЛЗУЮЩЕГО REST-API СОЦИАЛЬНЫХ СЕТЕЙ .....	121
<i>Кудрин И. М.</i> РАЗРАБОТКА ИНТЕРФЕЙСА ПРИБОРНОЙ ПАНЕЛИ ЭЛЕКТРОМОБИЛЯ .....	125
<i>Куст А. П.</i> МОДЕЛЬ АНТИБЛОКИРОВОЧНОЙ СИСТЕМЫ АВТОМОБИЛЯ .....	129
<i>Микрюков И. И.</i> СОЗДАНИЕ ВЕБ-ЧАТА С ПОМОЩЬЮ WEBSOCKET НА ЯЗЫКЕ GOLANG .....	139
<i>Микрюков И. И.</i> СОЗДАНИЕ ПРЯМОЙ ТРАНСЛЯЦИИ С ИСПОЛЬЗОВАНИЕМ NGINX-RTMP-MODULE .....	143
<i>Остапченко Д. А.</i> МЕТОДЫ ВЕКТОРИЗАЦИИ ТЕКСТА НА ЕСТЕСТВЕННОМ ЯЗЫКЕ .....	147
<i>Половников А. С.</i> ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ ДЕЯТЕЛЬНОСТИ В ФИРМЕ ПО СБОРКЕ ПК .....	150
<i>Радаева А. С.</i> ПРИЕМЫ НЕЙРОДИЗАЙНА И ИХ ВЛИЯНИЕ НА УСПЕШНОСТЬ БИЗНЕСА .....	153
<i>Савельев М. А.</i> ИСПОЛЬЗОВАНИЕ МЕТОДОВ НАИМЕНЬШИХ КВАДРАТОВ И НАИМЕНЬШИХ МОДУЛЕЙ В ЭКОНОМИКЕ .....	156
<i>Свеженин И. Д.</i> ИССЛЕДОВАНИЕ АЛГОРИТМОВ СЛЕЖЕНИЯ ЗА ОБЪЕКТОМ НА ОСНОВЕ АДАПТИВНЫХ КОРРЕЛЯЦИОННЫХ ФИЛЬТРОВ .....	160
<i>Трифонов В. Н.</i> СТРУКТУРА ПРОСОПОГРАФИЧЕСКИХ БАЗ ДАННЫХ И ОБЛАСТИ ИХ ПРИМЕНЕНИЯ .....	166
<i>Угольников М. В.</i> ПОДСИСТЕМА ДЛЯ СОПОСТАВЛЕНИЯ НОМЕНКЛАТУРЫ ПОСРЕДСТВОМ ТОКЕНИЗАЦИИ ТЕКСТА .....	170
<i>Ускова Ю. А.</i> КОМБИНАЦИЯ ШАБЛОНОВ ПРИ СОЗДАНИИ КОРПОРАТИВНОГО ПРИЛОЖЕНИЯ С МИКРОСЕРВИСНОЙ АРХИТЕКТУРОЙ .....	174
<i>Хвойновский В. Д.</i> ОБЗОР МЕТОДОВ СОЗДАНИЯ КАРТ ГЛУБИНЫ .....	177
<i>Шабунин А. П.</i> ПРИМЕНЕНИЕ МЕТОДОВ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ В МОБИЛЬНОМ ПРИЛОЖЕНИИ ДЛЯ ВЫБОРА ПРЕДМЕТОВ МЕБЕЛИ .....	181
<i>Шаров И. С.</i> РАЗРАБОТКА ПЛАТЫ-НОСИТЕЛЯ МИКРОМЕЗОНИННЫХ ПЛАТ .....	189
<i>Шиманский И. С.</i> АЛГОРИТМЫ АНАЛИЗА ФОТОДАННЫХ ДЛЯ ОПРЕДЕЛЕНИЯ РАСПОЛОЖЕНИЯ СКОТА .....	194
<b>Компьютерные технологии и программная инженерия .....</b>	<b>198</b>
<i>Васильев В. А.</i> АПРОБАЦИЯ РЕЛЕЙНОГО УПРАВЛЕНИЯ РОБОТОМ-МАНИПУЛЯТОРОМ В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ .....	198
	409

<i>Глушкова Е. М.</i> АНАЛИЗ ПОВЕДЕНИЯ БИОЛОГИЧЕСКОГО ОБЪЕКТА НА ОСНОВЕ МОДЕЛИ С ВНУТРИВИДОВОЙ КОНКУРЕНЦИЕЙ .....	204
<i>Горбунов А. А.</i> FRIS-ФУНКЦИЯ В ЗАДАЧЕ КЛАСТЕРИЗАЦИИ .....	213
<i>Губайдулин В. Ф.</i> ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ОПРЕДЕЛЕНИЯ ПАРАМЕТРОВ ОБЪЕКТОВ СПОРТИВНЫХ МЕРОПРИЯТИЙ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ ТРЕКИНГА .....	215
<i>Дулимов А. Е.</i> ФОРМАЛИЗАЦИЯ ЗАДАЧИ ФОРМИРОВАНИЯ ЛЕКЦИОННЫХ ПОТОКОВ ПРИ ПЛАНИРОВАНИИ УЧЕБНОГО ПРОЦЕССА ВУЗА .....	219
<i>Зулкашев Р. С.</i> АЛГОРИТМ СИНТЕЗА ДАТАСЕТА ДЛЯ ЗАДАЧИ ДИАРИЗАЦИИ .....	223
<i>Калинин А. П.</i> РАЗРАБОТКА АЛГОРИТМОВ И ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ ПЛАНИРОВАНИЯ ПУТЕШЕСТВИЙ НА ОСНОВАНИИ ПЕРЕЧНЯ ОСНОВНЫХ ПАРАМЕТРОВ .....	227
<i>Котин Д. Р.</i> МАТЕМАТИЧЕСКАЯ МОДЕЛЬ УПРАВЛЕНИЯ ТЕХНИЧЕСКИМ СОСТОЯНИЕМ СЛОЖНОГО ТЕХНИЧЕСКОГО ОБЪЕКТА .....	232
<i>Кустова Е. В.</i> АЛГОРИТМИЧЕСКИЕ И МАТЕМАТИЧЕСКИЕ ПОДХОДЫ ПРИ ИССЛЕДОВАНИИ ИММУНОЛОГИЧЕСКИХ МОДЕЛЕЙ .....	237
<i>Малинов Н. А.</i> РАЗРАБОТКА ВИДЕОИГР С ПОМОЩЬЮ ОНТОЛОГИЙ .....	242
<i>Павлов А. А.</i> МОДЕЛИ И АЛГОРИТМЫ ОЦЕНИВАНИЯ ТЕХНИЧЕСКОГО СОСТОЯНИЯ СЛОЖНЫХ ТЕХНИЧЕСКИХ ОБЪЕКТОВ НА ОСНОВЕ СТРУКТУРНО-СТОХАСТИЧЕСКИХ МОДЕЛЕЙ .....	247
<i>Парфишов П. Ю.</i> НЕЙРОСЕТЕВОЙ МЕТОД РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ НА ПРИМЕРЕ ОБНАРУЖЕНИЯ ОРУЖИЯ .....	252
<i>Поляков В. В.</i> ПРОГРАММА ДЛЯ ФАКТОРНОГО АНАЛИЗА ПРИЧИН ДТП НА ПЕРЕКРЕСТКАХ САНКТ- ПЕТЕРБУРГА .....	257
<i>Савельева Д. Д.</i> ОБЗОР КИБЕРУГРОЗ ЗА 2022 ГОД .....	261
<i>Секунов А. Н.</i> ПРИМЕНЕНИЕ АЛГОРИТМА ОТРИЦАТЕЛЬНОГО ОТБОРА ДЛЯ АНАЛИЗА ПОВЕДЕНИЯ СЕТЕВОГО ТРАФИКА НА ОСНОВЕ К-D ДЕРЕВА .....	264
<i>Сениченкова Я. О.</i> СОЗДАНИЕ КОНВЕЙЕРА ОБРАБОТКИ ДАННЫХ В АРАСНЕ NIPI И AIRFLOW ДЛЯ РЕШЕНИЯ ETL-ЗАДАЧ .....	269
<i>Фомин Д. А.</i> АЛГОРИТМ АНАЛИЗА МУЗЫКАЛЬНОГО ВРЕМЕННОГО РЯДА ДЛЯ ФОРМИРОВАНИЯ ПРОГНОЗА .....	273
<i>Шквиро С. А.</i> ОПРЕДЕЛЕНИЕ РЯДОВ ВИНОГРАДА ПО ДАННЫМ ДЗЗ .....	278
<i>Щеголева А. А.</i> ПРИМЕНЕНИЕ МЕТОДА АНАЛИТИЧЕСКОГО КОНСТРУИРОВАНИЯ АГРЕГИРОВАННЫХ РЕГУЛЯТОРОВ К ТРЕХКОМПОНЕНТНЫМ МОДЕЛЯМ ВЗАИМОДЕЙСТВИЯ ПИТАНИЯ, ФИТОПЛАНКТОНА И ЗООПЛАНКТОНА .....	281
<i>Яппаров А. М.</i> ТЕХНОЛОГИЯ ИНТЕРНЕТА ВЕЩЕЙ В ЛОГИСТИКЕ И ТРАНСПОРТНОЙ СИСТЕМЕ .....	286
<b>Вычислительные системы и сети .....</b>	<b>291</b>
<i>Агеев А. С., Девятов А. А., Сафина К. Ф.</i> О МЕТРИКАХ КАЧЕСТВА МАСКИРОВАНИЯ ИЗОБРАЖЕНИЯ .....	291
<i>Азаренок М. М.</i> МЕХАНИКА ПОДСКАЗОК В ИГРАХ ГОЛОВОЛОМКАХ .....	294
<i>Андреев А. А.</i> СОХРАНЕНИЕ НАУЧНЫХ САЙТОВ ОТ ИСЧЕЗНОВЕНИЯ .....	298
<i>Антонов А. А.</i> СОЗДАНИЕ СЕРВЕРА ПОД ХРАНЕНИЕ ДОСТИЖЕНИЙ РОССИЙСКОЙ НАУКИ .....	301
<i>Еременок Е. В.</i> ПРИЛОЖЕНИЯ ДЛЯ ТУРИЗМА И ПУТЕШЕСТВИЙ НА ОСНОВЕ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ .....	303
<i>Кибардин Д. А., Соя И. Н.</i> ВЛИЯНИЕ ВЗАИМОДЕЙСТВИЯ ИГРОВЫХ МЕХАНИК НА ОПЫТ ИГРОКА .....	305
<i>Климов М. В.</i> VR-СИМУЛЯТОР ПОВЕДЕНИЯ ЧЕЛОВЕКА ПРИ ПОЖАРЕ .....	309

<i>Мурашко М. А.</i> МЕХАНИКИ СТРАХА В ИГРАХ ТИПА ХОРРОР .....	312
<i>Цыглев В. М.</i> СПОСОБЫ ДОЛГОВРЕМЕННОГО СОХРАНЕНИЯ ИНФОРМАЦИИ НАУЧНЫХ САЙТОВ .....	316
<b>ПРИКЛАДНАЯ МАТЕМАТИКА, ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ ТАМОЖЕННЫЕ ТЕХНОЛОГИИ .....</b>	<b>319</b>
<i>Бульгин Ю. В.</i> АРХИТЕКТУРА ОБРАЩЕНИЯ К ДАННЫМ В СОВРЕМЕННЫХ WEB-ПРИЛОЖЕНИЯХ .....	319
<i>Викторова А. П.</i> ОБЗОР И СРАВНЕНИЕ ЛОКАЛЬНЫХ ИСТОЧНИКОВ ХРАНЕНИЯ ДАННЫХ ДЛЯ ИНФОРМАЦИОННЫХ СИСТЕМ .....	325
<i>Долгов Е. Н.</i> ПЕРСПЕКТИВЫ ПРИМЕНЕНИЯ МЕХАНИЗМОВ НА ОСНОВЕ ШАРОВЫХ ШЕСТЕРНЕЙ В НАСТОЯЩЕМ ВРЕМЕНИ .....	327
<i>Смирнов М. С.</i> ВЛИЯНИЕ ОБЪЕКТОВ ДИНАМИЧЕСКОГО ОКРУЖЕНИЯ НА ПРОИЗВОДИТЕЛЬНОСТЬ UNITY .....	330
<i>Халтсонен А. М.</i> ИГРОВЫЕ ДВИЖКИ С ПОДДЕРЖКОЙ МОБИЛЬНОЙ РАЗРАБОТКИ: ПРЕИМУЩЕСТВА И НЕДОСТАТКИ .....	336
<b>ВОЕННЫЙ УЧЕБНЫЙ ЦЕНТР .....</b>	<b>340</b>
<i>Анохин Д. В.</i> ПРОБЛЕМА ФИЗИЧЕСКОЙ ЗАЩИТЫ КАБЕЛЬНОЙ СЕТИ АВТОНОМНОГО ИСТОЧНИКА ПИТАНИЯ .....	340
<i>Арефьев Н. А., Селедкин Д. Е., Вахабов В. Д., Парфенов А. А.</i> АППАРАТНАЯ ЧАСТЬ ОБУЧАЮЩЕГО ПРОГРАММНОГО КОМПЛЕКСА .....	346
<i>Ахмадов Р. А.</i> ФАЗОВРАЩАТЕЛЬ НА ОСНОВЕ МИКРОКОНТРОЛЛЕРА .....	353
<i>Волкова А. П.</i> РАЗРАБОТКА ДАТЧИКА УГЛА НА ОСНОВЕ АКСЕЛЕРОМЕТРА .....	358
<i>Гришкин Г. С.</i> АНАЛИЗ ИСПОЛЬЗОВАНИЯ ЛАЗЕРНЫХ ТЕХНОЛОГИЙ ПРИ РЕШЕНИИ ЗАДАЧ БОРЬБЫ С БПЛА .....	362
<i>Желаевский М. А., Карлушин Д. С.</i> СПОСОБЫ СТАБИЛИЗАЦИИ УСТРОЙСТВ НАВЕДЕНИЯ И ЦЕЛЕУКАЗАНИЯ ВЕРТОЛЕТА .....	366
<i>Жмурин А. В.</i> РАЗРАБОТКА ОБУЧАЮЩЕГО ПРИЛОЖЕНИЯ «КОРРЕЛЯЦИОННАЯ ОБРАБОТКА СИГНАЛОВ» .....	369
<i>Зорин И. А., Швайко А. С., Жданов Д. В.</i> РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА БАЗЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ОЧИСТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ ОТ ВРЕДНЫХ ФАКТОРОВ .....	373
<i>Зырянов Д. А.</i> ИСПОЛЬЗОВАНИЕ БПЛА ДЛЯ ВЫБОРА ТРАССЫ ПРИ ПРОКЛАДКЕ ПОЛЕВОЙ КАБЕЛЬНОЙ ЛИНИИ С УЧЕТОМ ОПЫТА СВО .....	376
<i>Иванов А. С.</i> ИССЛЕДОВАНИЕ ВОЗМОЖНОСТИ ИСПОЛЬЗОВАНИЯ НА БОРТУ ВОЗДУШНОГО СУДНА ГИБРИДНОЙ СИЛОВОЙ УСТАНОВКИ .....	383
<i>Пластунов В. А.</i> ИДЕНТИФИКАЦИЯ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ НА ОСНОВЕ РЕЗУЛЬТАТОВ ТРАЕКТОРНЫХ ИЗМЕРЕНИЙ .....	386
<i>Радионов В. А., Кудряшев О. Д.</i> СОСТОЯНИЕ И ПЕРСПЕКТИВЫ ПРОЕКТА ПО СОЗДАНИЮ СТРАТЕГИЧЕСКОЙ РАКЕТЫ LGM-35A .....	389
<i>Рахматов Т. Х.</i> ЦИФРОВОЙ ОММЕТР .....	392
<i>Фабрика А. А., Божков И. И.</i> ОБУЧАЮЩАЯ ПРОГРАММА ДЛЯ ИЗУЧЕНИЯ ПАРАМЕТРОВ ДЕТЕРМИНИРОВАННЫХ СИГНАЛОВ .....	395
<i>Федюнин Д. Д.</i> ЦИФРОВОЙ МАНОМЕТР .....	400
<i>Фролов М. Н.</i> РАЗРАБОТКА И ИССЛЕДОВАНИЕ АЛГОРИТМОВ АДАПТИВНОГО ИНТЕРФЕЙСА «ПИЛОТ-ЛЕТАТЕЛЬНЫЙ АППАРАТ» .....	403
	411

Научное издание

СЕМЬДЕСЯТ ШЕСТАЯ  
МЕЖДУНАРОДНАЯ СТУДЕНЧЕСКАЯ  
НАУЧНАЯ КОНФЕРЕНЦИЯ ГУАП  
(17–21 апреля 2023 г.)

Часть 3

ТЕХНИЧЕСКИЕ НАУКИ

Сборник докладов

Ответственный за выпуск:  
А. В. Бобович

Редакторская подготовка *Е. В. Лазаревой*  
Компьютерная верстка *А. Н. Колешко*

---

Подписано к печати 14.09.23. Дата выхода в свет 20.09.2023. Формат 60x84 1/16.  
Усл. печ. л. 24,3. Уч.-изд. л. 47,8 Тираж 150 экз. Заказ № 355.

---

Редакционно-издательский центр ГУАП  
190000, г. Санкт-Петербург, ул. Большая Морская, д. 67, лит. А

Распространяется бесплатно