

# ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ: ТЕОРИЯ И ПРАКТИКА

Сборник научных  
трудов

Выпуск 14



САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
ЛЕСОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ИМЕНИ С. М. КИРОВА

Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЛЕСОТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ имени С. М. Кирова»

---

*Кафедра информационных систем и технологий*

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ:  
ТЕОРИЯ И ПРАКТИКА

Сборник научных трудов

Выпуск 14

Санкт-Петербург  
2022

Рассмотрено и рекомендовано к изданию  
учебно-методической комиссией Ученого совета  
Санкт-Петербургского государственного лесотехнического  
университета имени С. М. Кирова

Редакционная коллегия:

кандидат технических наук, доцент **М. Р. Вагизов** (отв. редактор),  
кандидат технических наук, доцент **Н. В. Меламед** (отв. секретарь),  
кандидат технических наук, доцент **С. П. Хабаров**

Составитель

кандидат технических наук, доцент **Н. В. Меламед** (отв. секретарь)

Рецензент

доктор технических наук, профессор **В. А. Григорьев**  
ООО «Лаборатория инфокоммуникационных сетей»

**Информационные системы и технологии: теория и практика:**  
сб. научн. тр. Вып. 14 / отв. ред. М. Р. Вагизов. – Санкт-Петербург:  
СПбГЛТУ, 2022. – 252 с.

ISBN 978-5-9239-1303-3

*Представлен кафедрой информационных систем и технологий.*

Сборник подготовлен по материалам кафедры вуза, представленным на научно-технической конференции института леса и природопользования СПбГЛТУ в феврале 2022 г., и практических работ, выполненных ее сотрудниками.

## Горбачеву Виктору Александровичу

### ПОСВЯЩАЕТСЯ



**Горбачев Виктор Александрович (1949 - 2021)** закончил московский экономико-статистический институт (ныне входит в состав Плехановского университета) по специальности инженер – экономист. Со студенческих времен начал заниматься программированием и до настоящего. После института с 1971 по 1973 годы служил в армии. Потом работал программистом в отделе математического обеспечения Октябрьской железной дороги (1973-1976 годы). В 1976 году пришел работать начальником группы математического обеспечения вычислительного центра Лесотехнической Академии. Занимался разработкой программ для обеспечения учебного процесса. В 1984 году защитил кандидатскую диссертацию на экономическом факультете Лесотехнической Академии и получил степень кандидата экономических наук. С 1985 года перешел на преподавательскую деятельность кафедры информатики. Прошел все должностные ступени от ассистента до доцента. Читал множество курсов: информатика, информационные технологии, базы данных, автоматизированные экономические информационные системы и другие курсы. С 2000-х годов подключился к внедрению автоматизированной системы управления учебным процессом в Лесотехническом Университете.

*Из воспоминаний Мальцева Игоря Михайловича (Научного руководителя Лаборатории ММИС)*

Виктор Александрович Горбачев, являясь высококлассным специалистом в области ИТ, прилагал максимальные усилия в сфере цифровизации управления учебным процессом, основываясь на возможностях информационных систем Лаборатории ММИС (Шахты). Он неоднократно участвовал в работе научно-практических семинаров, проводимых, в том числе и в Санкт-Петербурге, организации встреч сотрудников Лаборатории ММИС с руководством и ППС лесотехнической академии (ныне университета). Идеи Виктора Александровича, несомненно, оказали значительное влияние на функционал нашего ПО, используемого по всей России.

Знаю его не только как отличного организатора, но и как прекрасного, отзывчивого человека, с которым поддерживал дружеские отношения. Его оценки «шахтинской» команды вдохновляли нас на новые свершения.



*Из воспоминаний Куриловой М.Н.*

Виктор Александрович Горбачев пришел работать на кафедру вычислительной техники (так тогда называлась кафедра ИСиТ) в 1976 г. начальником группы математического обеспечения (МО). В свое подчинение он получил группу молодых программисток, с которой ему предстояло внедрять в учебный процесс вычислительную технику и постепенно приобщать профессорско-преподавательский состав к ее использованию.

Это была эпоха «больших» машин, в академии работала ЭВМ «Минск-32», которую потом заменили на ЕС-1022. Стандартное программное обеспечение имелось для статистической обработки данных, а вот программы для учебного процесса и научных исследований предстояло писать самостоятельно. Переход на новую технику также требовал замены программного обеспечения, Все эти проблемы решала группа МО и прежде всего ее руководитель. Программирование Виктору Александровичу было хорошо знакомо, он имел опыт работы программистом, группа МО – отлично понимала своего руководителя, который щедро делился идеями и был надежным помощником при всех возникающих проблемах.



Просторы машинного зала, ЭВМ ЕС-1022 (сейчас ауд.101)



В.А. Горбачев -  
руководитель группы  
МО

Любопытно было общение с заказчиками, преподавателями кафедр академии, которые впервые сталкивались с применением вычислительной техники в своих дисциплинах и только в процессе работы понимали, какие возможности давали создаваемые программы! Вот это и доставляло руководителю группы много хлопот: могла потребоваться серьезная переделка программ, да и вообще не всегда она была возможна, проще было начать все снова, а это отражалось на сроках работы, что было очень нежелательно. И эти проблемы Виктор Александрович всегда решал успешно!



Интересно, что студенты, тоже высоко оценивали его преподавательскую работу. В академии не раз проводилось анонимное тестирование среди студентов по оценке работы преподавателей, и, конечно, Виктор Александрович получал самые высокие баллы, как преподаватель с боль-

шим опытом практической работы и внимательным, доброжелательным отношением к студентам.

*Из воспоминаний Вагизова М.Р.*

Горбачёв Виктор Александрович - преподаватель, коллега, учёный и высококлассный специалист в области систем управления базами данных. Моё знакомство с Виктором Александровичем началось со студенческих лет, ещё будучи студентом я обратил внимание на то, как Горбачёв В.А. легко и непринужденно, но при этом с интересом рассказывал нам свой предмет Управление данными. Его отличали очень человеческие качества, как преподаватель он всегда с уважением относился к студентам, всегда доносил самую суть предмета, при этом он увлеченно проводил практические занятия, никогда не оставлял без внимания, студентов у которых что-то не получалось. Он всегда интересовался нашей успеваемостью, предлагал прохождение практик в IT-компаниях, мог дать напутствие или дельный совет.

Большой вклад Горбачёв В.А. внёс для работы автоматизации университета, многие из его разработок используются в образовательном процессе и по настоящее время. Особое внимание он уделял вопросам работы с документооборотом кафедры, им было разработано несколько программ для информационной поддержки кафедры, мне повезло поучаствовать в совместной работе с Горбачевым В.А. по разработке такой системы. Стоит отметить, что Виктор Александрович, не оставался безучастным в других вопросах жизни университета, всегда участвовал в дискуссиях по стратегическим вопросам развития ВУЗа, что отражало его глубокую вовлечённость в жизнь университета.

Виктор Александрович навсегда останется в нашей памяти!

**А.М.Заяц**, кандидат технических наук, профессор.  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
zamfta@yandex.ru

## АНАТОЛИЙ МИХАЙЛОВИЧ ПОЛОВКО – НАДЁЖНОСТЬ В НАУКЕ И ЖИЗНИ

*Аннотация:* В статье представлены яркие страницы из жизни А.М. Половко и воспоминания его учеников и коллег.

*Ключевые слова:* научная школа А.М.Половко, один из создателей теории надёжности.



Анатолий Михайлович Половко в этом году отмечал бы своё *СТОЛЕТИЕ*. Конечно, у истории нет сослагательного наклонения, но эта дата есть, а память об Анатолии Михайловиче переживет как минимум еще одно столетие.

Год 1921, четырнадцатое ноября в городке Ичня Черниговской области родился Анатолий Михайлович Половко. Детство, школа и она - Великая Отечественная война. Молодой Анатолий Половко призван воевать в составе 373-го артиллерийского полка 175-й стрелковой дивизии. Младший сержант

печеру. Тор. Половко на ружью гикал свое оружие под огнем противника на прямую наголку и в упор расстреливал гитлеровцев. В этом бою было уничтожено 2 ст. пулемета и до 30 гитлеровцев. В этом же бою он был ранен, но не ушел с поля боя, пока не была отбита контратака гитлеровцев.

- ЦАМО. Учётная карточка Анатолия Половко. <https://pamyat-naroda.su/awards/cards/1101400378>

За что и награждён орденом Славы III степени.

С 1942 года началась новая страница в жизни Анатолия Михайловича. Сержант А.М.Половко, поступает на учебу в Военный инженерный краснознаменный институт имени А.Ф.Можайского [1].

После завершения обучения уже академии, А.М.Половко два года служил на авиационной научно-испытательной базе и был рекомендован для поступления в адъюнктуру. После ее окончания и защиты кандидатской диссертации до 1966 г. преподавал на вновь созданной в 1949 г. кафедре автоматики и телемеханики академии.

В 1954 г. вышел первый сборник переводов зарубежных материалов, затрагивающих вопросы надёжности, под ред. академика В.И. Сифорова.

«Увлечшей его теорией надежности» А.М. Половко стал заниматься с 1957 года — на заре ее развития, и уже через пять лет добился больших успехов.

В 1958 г. состоялась первая Всесоюзная конференция по надежности. Ленинградскими специалистами уже в 1959г. в первом отечественном отделе надежности (в одном из Ленинградских НИИ Судпрома) была выпущена первая книга - «Основы теории и расчета надежности (авторы книги – одни из основоположников отечественной школы надежности: И.М. Маликов, А.М. Половко, Н.А. Романов и П.А. Чукреев).

В 1963 году Половко А.М. защитил докторскую диссертацию, а в следующем году была опубликована его монография — первая в стране книга, в которой были сформулированы основные положения теории надежности как науки. Материалы опубликованных А.М. Половко исследований широко использовались при создании высоконадежных технических систем самого различного назначения. В 1969 г. А.М. Половко совместно с А.Н. Свердликом и И.В. Вазингером был удостоен бронзовой медали ВДНХ за создание и внедрение специализированной вычислительной машины для исследования надёжности.



Вот как пишет об Анатолии Михайловиче профессор МФТИ и МЭИ; профессор Университета Джорджа Вашингтона, Университета Дж. Мейсона и Калифорнийского университета в Сан-Диего (США) Игорь Алексеевич Ушаков в своей статье «Откуда есть пошла надежность на Руси».

«Истинным главой ленинградской школы надежности нужно признать Анатолия Михайловича Половко. Он возглавил Ленинградский кабинет надежности, руководил Ленинградским городским семинаром по надежности. А.М. Половко — автор первой отечественной книги по надежности, переведенной на Западе. Ленинградская школа внесла много нового в теорию надежности».

Воспоминания профессора Ушакова И.А. из книги «Записки неинтересного человека».

«В Ленинграде я познакомился и на всю жизнь сблизился с Анатолием Михайловичем Половко - признанным главой ленинградской школы теории надежности. У меня хранится его книга с дарственной надписью, которую он подарил мне после моей защиты докторской диссертации, на которой он выступал третьим оппонентом. Потом пошли выступления оппонентов. Команда у меня была "ЧетыреКО": ГнеденКО, ПоловКО и КоваленКО - оппоненты и БусленКО - головная организация.

После защиты, как и положено, был банкет в гостинице "Москва". Тамадой, естественно, был Анатолий Михайлович Половко. Внешне и по манере говорить он был копия Тарапунько (тот, который со Штепселем), артистичный, с большим чувством юмора. Говорить он умел! Жаль, что он ушел с банкета раньше, чтобы успеть на "Красную стрелу" в Ленинград.

Я старался вовсю. Обзвонил всех знакомых докторов, подходящих по профилю. Были здесь многие "киты" и среди них даже членкоры и рес-

публиканские академики (пишу в алфавитном порядке): Н.П. Бусленко, Е.С. Вентцель, Б.В. Гнеденко, И.Н. Коваленко, Б.Р. Левин, А.М. Половко, В.С. Пугачев... Такой букет!».

Половко, Анатолий Михайлович - профессор, доктор наук, один из основоположников ленинградской школы надежности, соавтор первой в СССР книги по расчету надежности.



А.М. Половко возглавил Ленинградский кабинет надёжности куда входили такие известные ученые как И. А. Рябинин, Л. К. Горский, Н. М. Седякин, Г. Н. Черкесов, И. Б. Шубинский и др. и более 30 лет руководил Ленинградским общегородским семинаром по проблемам надёжности техники.

Значимость ленинградской школы надежности характеризуется тем, что в год десятилетия семинара был выпущен общесоюзный памятный знак.

Из воспоминаний профессора Гурова Сергея Владимировича «А.М. Половко - руководитель семинара по надежности технических систем, на котором присутствовало множество ученых, производственников, аспирантов и соискателей. Специалисты по надежности из других регионов Советского Союза часто приезжали в Ленинград, чтобы обменяться мнениями с представителями ленинградской школы, лично пообщаться с Анатолием Михайловичем. На этих семинарах Анатолий Михайлович выглядел как артист. На него невозможно было спокойно смотреть. Он заводил людей своими удачными высказываниями, своей энергией, шутками. Но все было направлено на одно – увлечь собравшийся коллектив вопросами надежности сложных систем».

Анатолий Михайлович не только внес огромный вклад в теорию надежности, но и тонко чувствовал новое и прогрессивное в ее развитии.

Вот как об этом вспоминает профессор Рябинин И.А. в своей статье «История возникновения, становления и развития логико-вероятностного анализа в мире» «...решение практических задач ОКБ «Импульс» по созданию высоконадежных систем гражданского и оборонного назначения привело к новым теоретическим результатам в области ЛВМ в кандидатской диссертации Черкесова Г.Н. (1967). Скепсис к вероятностным расчетам и недоверие к числам 0.999 серьезно усложнили эту защиту: - научный руководитель профессор Соколов Тарас Николаевич был категорически против ее защиты по техническим наукам, соглашаясь хоть на докторскую, но по математическим наукам; - первый официальный оппонент профессор Половко Анатолий Михайлович уговорил меня (в качестве второго официального оппонента), пойти, может быть и на конфликт с Соколовым Т.Н., но защитить Геннадия Николаевича по техническим наукам. Настойчивость Половко А.М. в выборе меня стали понятными, когда я обнаружил в диссертации Г.Н.Черкесова попытку разработки принципиально новых методов исследования надежности, названных нами логико-вероятностными методами. При защите диссертации не было задано ни одного вопроса со-



искателю, и никто не выступил в прениях. Складывалось впечатление, что дело шло к провалу защиты, но по предложению профессора Канторовича Леонида Витальевича приступили всё же к тайному голосованию, которое продемонстрировало стопроцентное понимание достоинств и этой «бумажной» диссертации. Опускаю здесь хвалебные слова Канторовича Л.В. об исключительной добросовестности официальных оппонентов, которые так подробно и убедительно пояснили научную и практическую значимость работы и даже возможности издания ее научной части в открытой печати».

Анатолий Михайлович создал свою широко известную в нашей стране научную школу по теории надежности автоматических и других сложных систем, имеющую многочисленных учеников и последователей. Его учениками, в частности, были Н.К.Зайнашев и В.А.Смагин, ставшие докторами технических наук.

Из воспоминаний доктора технических наук, профессора В.А.Смагина

Военно-космическая академия им. А.Ф. Можайского. «Анатолий Михайлович Половко – основоположник научной школы теории резервирования, его применения в системах управления ракет и космических аппаратов.

Природа редко дарит нам таких талантливых мыслителей и прекрасных людей. Огромное спасибо ей за это. Светлая, чистая и глубокая память о нём хранится в мечтах и делах его учеников.

Я, его первый учёный секретарь научного семинара по теории надёжности, с глубоким уважением храню в памяти множество событий из научной жизни моего учителя, связанных с успехами и трудностями в науке.

Благодарю судьбу за то, что она связала меня с ним, и отдаю низкий поклон его таланту и человечности».

Как отмечалось в историческом очерке Военного инженерного краснознаменного института имени А.Ф.Можайского [2] «Ученые академии одними из первых в Вооруженных Силах, предвидя огромное значение вычислительной техники, стали готовить инженеров-специалистов в этой области. Внедрением в учебный процесс вычислительной техники занимался коллектив, которым руководил А.М.Половко.



Материалы исследований, имевшие целью автоматизацию процесса программирования, обеспечение надежности программного обеспечения, были опубликованы А.М.Половко и др. Признанием огромного вклада ученых в науку и ускорение научно-технического прогресса явилось присвоение им в период их работы в институте звания заслуженного деятеля науки и техники РСФСР. Этого почетного звания был удостоен А.М.Половко.

Первоначально внедрением в учебный процесс вычислительной техники занимались коллективы, которыми руководили преподаватели Ю. И. Беззубов, И. В. Еременко, а затем А. М. Половко. Ю. И. Рыжиков, А. Н. Свердлин, П. А. Чукреев, В. А. Смагин. Ими были поставлены курсы по электронным вычислительным машинам и программированию, знание которых было необходимо для всех инженеров».



В 1973 г. А.М. Половко закончил службу в Вооруженных Силах и возглавил кафедру информатики и вычислительной техники Ленинградского лесотехнического института им. С.М. Кирова (академия, университет). Будучи в Лесотехнической академии, Анатолий Михайлович наряду со своей любимой наукой НАДЕЖНОСТЬ активно продолжал разработку методик применения вычислительной техники в учебном процессе и в исследованиях, о чем свидетельствуют его фундаментальные труды в этих направлениях [3,4].

В истории кафедры написано: «В 1973 г. на кафедру пришел новый заведующий – АНАТОЛИЙ МИХАЙЛОВИЧ ПОЛОВКО. заслуженный деятель науки и техники РФ, доктор технических наук, профессор. Крупный ученый, основоположник теории надежности, который после завершения службы в Вооруженных Силах возглавил кафедру академии. Кафедра расширялась, креп ее научный потенциал...».



С 1991 г. Анатолий Михайлович продолжал работать на кафедре в качестве профессора.



Всегда в работе

Воспоминания сотрудников кафедры Информационных систем и технологий Санкт-Петербургского государственного лесотехнического университета им. С.М. Кирова [5-7].

«Как ученый и методист ни разу не подписывался под чужой работой, даже если был ее научным руководителем.

Внимательно выслушивал мнение собеседника, но, если его мнение было другим, то он обрушивался на оппонента всей мощью научных доказательств. Причем делал это спокойно и так, чтобы человек понял свое заблуждение.

Разговор был очень лаконичный и понятный собеседнику. На конференциях и семинарах Анатолий Михайлович являл собой центральную фигуру, около которой группировались ученые, представляющие все основные школы по надежности технических систем».

Плакат с фразой Анатолия Михайловича "Экспонента – закон природы" продолжительное время висел на входе в вычислительный центр академии. Член нескольких диссертационных Советов по техническим и экономическим наукам. Влюбленность в науку. Изучение пакетов прикладных математических программ (Derive, Matlab, Maple, Mathematica).



Написание книг. Двухтомник совместно с С.В. Гуровым, по словам Анатолия Михайловича, – «дело всей моей жизни». Он был не только исполнитель, но, прежде всего, вдохновитель и организатор этих книг (ведь это в 83 года). Постоянные встречи с главным редактором Е.И. Кондуковой, с литературными и техническими редакторами и рецензентами. В книге содержится много новых идей и желаний, которые вполне могут служить темами диссертаций.

диссертаций.

Например, интересной является задача о надежности длинного трубопровода, как система с последовательно соединенными элементами.

Достижения А.М.Половко:

- спортсмен (шахматы, коньки);
- прекрасный исполнитель украинских и неополитанских песен;
- актер и замечательный оратор;
- педагог и методист высочайшего уровня;
- отец, дедушка, прадедушка;



- садовод – исследователь (хотел вырастить сорт "красных" огурцов);
- автомобилист ("весь бензин хорошо бы превратить в воду");
- ученый, основатель научного направления и ленинградской школы в теории надежности;
- добрый и хороший человек.

Недостатки А.М.Половко - шутка. Не любил «Excel».



Заседание диссертационного совета



Ученики и друзья А.М. Половко  
( Слева на право дтн, профессор Гуров С.В.,  
ктн, доцент Хабаров С.П., дтн, профессор  
Богатырев В.А. )



( Слева на право ктн, доцент Михайлов А.А.,  
дтн, профессор Богатырев В.А. )





Коллектив кафедры после очередной научной конференции 2005г.



Коллектив кафедры и ее ветераны

Как дороги нам всем эти фотографии зафиксировавшие миг с Анатолием Михайловичем!

Прошло вот уже почти 15 лет.

### **Библиографический список**

1. Ленинградская военно-воздушная академия РККА в Йошкар-Оле. <http://olacity.ru/showthread.php?bid=5&threadid=6508>.
2. Очерк истории. 1941-1980. Ленинград 1981 Военный инженерный краснознаменный институт имени А.Ф.Можайского.
3. Половко А.М. Derive для студента. Спб.: БХВ-Петербург, 2005



4. Половко А.М., Ганичев И.В. Mathcad для студента. Спб.: БХВ-Петербург, 2006.
5. Информационные системы и технологии: теория и практика. Сборник научных трудов. Санкт-Петербург: СПбГЛТА, 2008.
6. Заяц А.М. 100-летию со дня рождения доктора технических наук, профессора заслуженного деятеля науки и техники рсфср Анатолия Михайловича Половко. В сборнике: Информационные системы и технологии: теория и практика. Санкт-Петербургский государственный лесотехнический университет имени С.М. Кирова. Санкт-Петербург, 2021.
7. Анатолий Михайлович Половко (1921 - 2007). <http://gurov.vs58.net/Polovko/index.htm>

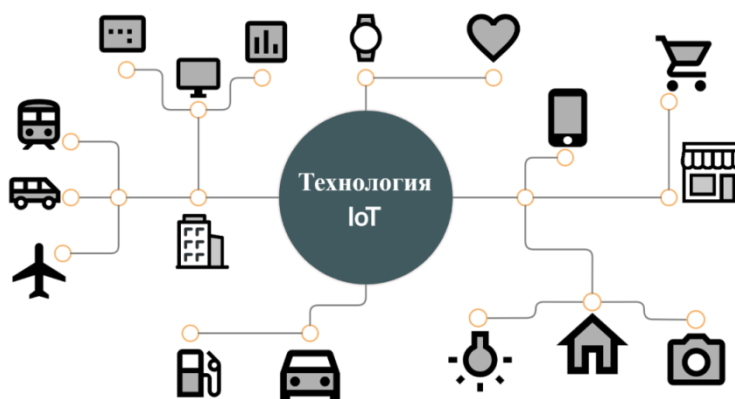
**А.К. Бойцов**, ассистент  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
[A.K.Boitsov@yandex.ru](mailto:A.K.Boitsov@yandex.ru)  
**А.А. Мокринский**, студент  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
[A.A.Mokrinsky@gmail.com](mailto:A.A.Mokrinsky@gmail.com)

## **ИОТ ДЛЯ СИСТЕМЫ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ АППАРАТНО-ПРОГРАММНОГО КОМПЛЕКСА С ИСПОЛЬЗОВАНИЕМ ДАТЧИКА УГЛЕКИСЛОГО ГАЗА В ЛЕСУ**

*Аннотация.* Для эффективного взаимодействия системы автоматического управления аппаратно-программного комплекса с использованием датчика углекислого газа в лесу требуется передача больших данных, где необходимо производить обмен информацией в режиме реального времени. Для решения данной проблемы разработана система автоматического управления аппаратно-программного комплекса с применением датчика углекислого газа в лесу. Требованию такой системы управления отвечает технология «Интернета Вещей» (IoT), где передача информации начинается с датчика, а заканчивается сервером и для их взаимодействия необходимы гибкие и надежные протоколы. Для выбора необходимых протоколов разработана схема взаимодействия элементов аппаратно-программного комплекса. Затем проанализированы участки схемы и подобраны протоколы: на участке от узла до узла – протокол DDS, от узла до брокера – XMPP, от брокера к серверу – MQTT, с брокером и с приложением - SOAP.

**Ключевые слова:** *IoT, интернет вещей, датчик углекислого газа, система автоматического управления, IoT в лесу, IoT в лесном хозяйстве, система автоматического управления аппаратно-программного комплекса.*

Современные системы и технологии при работе создают большие объёмы данных [1, с. 15]. Для повышения эффективного взаимодействия систем в работе с большими данными необходимо производить обмен информацией между системами в режиме реального времени. Требованиям такого взаимодействия отвечает технология «Интернета Вещей» (IoT), которая состоит из устройств, датчиков или приборов и осуществляет сбор, анализ и распределение данных, а также управление без прямого участия человека. Возможности данной технологии безграничны от сферы анализа больших данных до использования алгоритмов искусственного интеллекта (рисунок 1) [2, 3, 4].

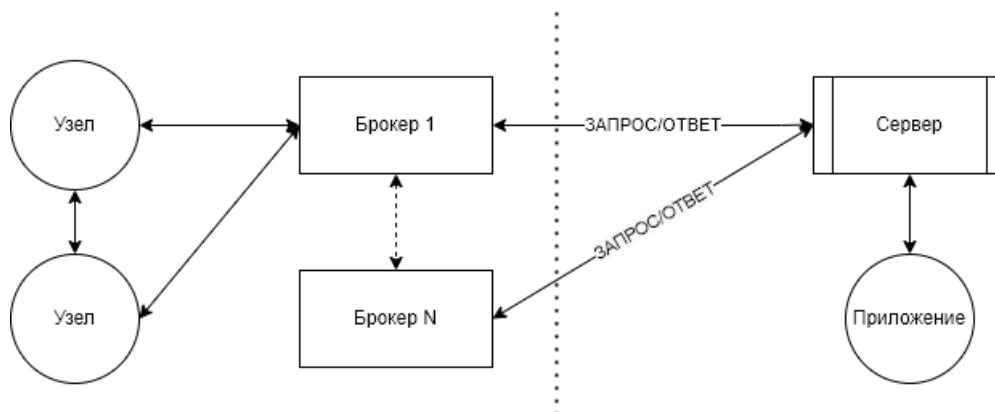


**Рис. 1.** Возможности технологии IoT

Стремительное развитие информационных технологий и вычислительной технике уже привело к интенсивной автоматизации многих технологических процессов. Многие сферы нашей жизни переходят от частичной автоматизации процессов до полной автоматизации, в которой общий уровень вычислительной мощности и информатизации открывает новые возможности в лесном хозяйстве [5], где уже используется технология IoT для мониторинга состояния лесных территорий, маркировке древесины, оценке действий оператора в полуавтоматических и многозадачных системах лесозаготовок, а также для проведения инвентаризации лесов [6, 7, 8]. Но впервые данная технология предлагается для системы автоматического управления аппаратно-программного комплекса с использованием датчика углекислого газа в лесу (результаты разработки датчика частично опубликованы [9]).

Цель данного исследования разработка системы автоматического управления аппаратно-программного комплекса с использованием датчика углекислого газа в лесу.

В системе автоматического управления аппаратно-программного комплекса будут задействованы датчики небольших размеров, которые используют мало электроэнергии и передают параметры показаний датчика в реальном времени, где необходим транспортный протокол, который будет гарантировать надежную передачу данных. Полученные результаты измерений будут обрабатывать узлы и передавать результаты на сервер (рисунок 2).



**Рис. 2.** Схема взаимодействия элементов аппаратно-программного комплекса

На рисунке 2 передача информации начинается с датчика, а заканчивается сервером, и для их взаимодействия необходимы гибкие и надежные протоколы, взаимодействующие с технологией IoT [10]. Для автоматизации процесса передачи информации необходимо сделать из датчиков единый узел, который будет передавать показатели метеоусловий и CO<sub>2</sub> на сервер. Узел, будет объединять данные, полученные с датчиков, передавая их дальше. Чтобы взаимодействовать с сервером возникает потребность в клиентском приложении, которое будет выводить показатели метеоусловий и CO<sub>2</sub> на персональном устройстве. Перед этим задействуется брокер, собирающий данные с узла, который позволит установить сбор и временное хранение данных, а также произвести их анализ и обработку. Хранение данных осуществляется до полной отправки на сервер. Для решения автоматизации процесса передачи информации требуется разбор схемы взаимодействия элементов аппаратно-программного комплекса по отдельным участкам.

Для решения задачи сбора информации, чтобы временно хранить и в дальнейшем перенаправлять её, между узлами будет использоваться протокол DDS, позволяющий эффективно и надежно доставлять данные (рисунок 3). DDS предварительно распределяет данные для временного хранения или перенаправления [11]. Чтобы не снижать эффективность DDS, стоит избегать запросов, которые требуют мощной сети, что в лесу сложно реализовать. Благодаря частому использованию в промышленной автоматизации и управлению телекоммуникационным оборудованием, сенсор-

ными сетями и системами управления сетями есть отличные примеры работы DDS [12].

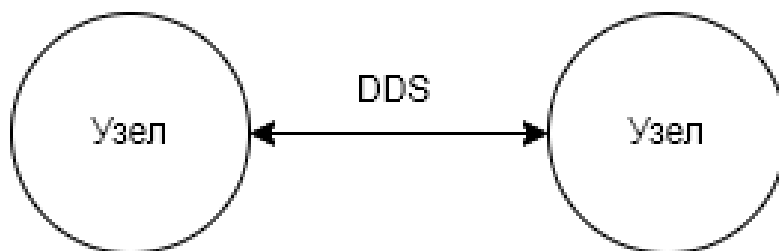


Рис. 3. Протокол DDS на участке между узлами

Перейдя к участку от узла до брокера, возникает задача настройки узлов, передача и распределение информации, где на сегменте могут использоваться два протокола: XMPP и COAP (рисунок 4).

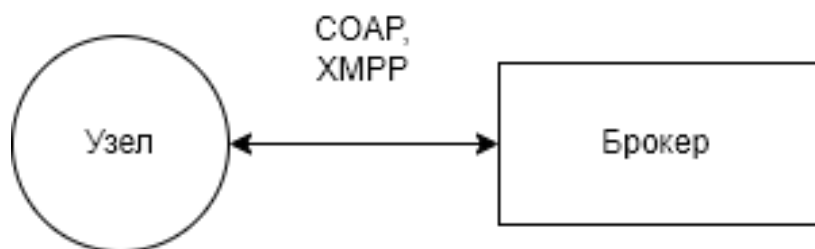
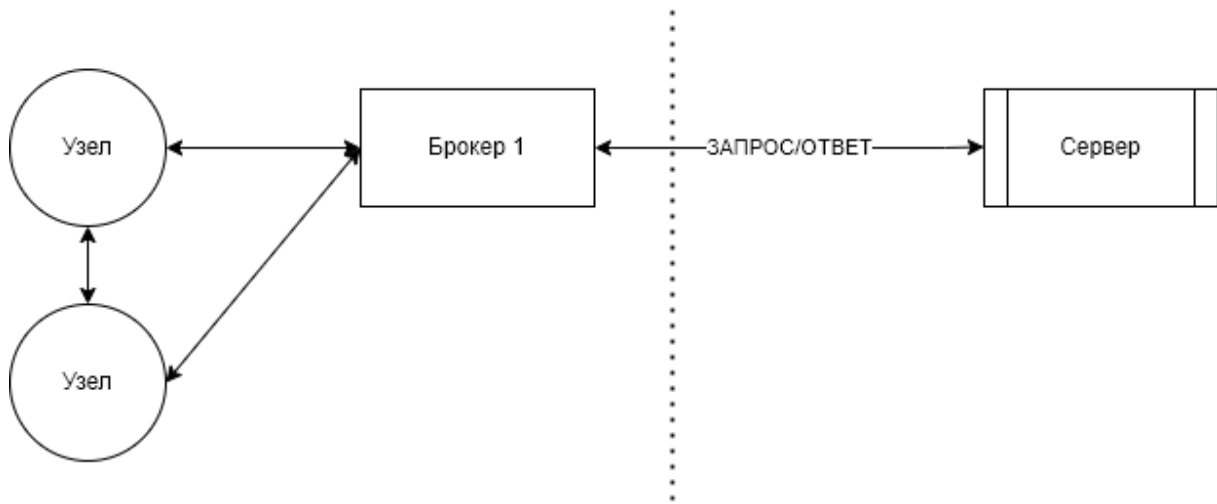


Рис. 4. Участок от узла до брокера

XMPP отвечает за структуру и настройку узлов, за передачу и распределение информации, а также обеспечивает легкий способ адресации устройств, используя надежный протокол транспортного уровня TCP [13, 14]. Для применения датчиков на большой и малой лесной территории, необходима масштабируемость, которой обладает XMPP. Но сеть будет обладать ограниченными ресурсами и низким энергопотреблением, с чем справляется протокол COAP, который предназначен для небольших мало-мощных датчиков [15]. Протокол позволяет наблюдать за работой датчиков, благодаря простым запросам, организовывая передачу изменений состояния датчиков.

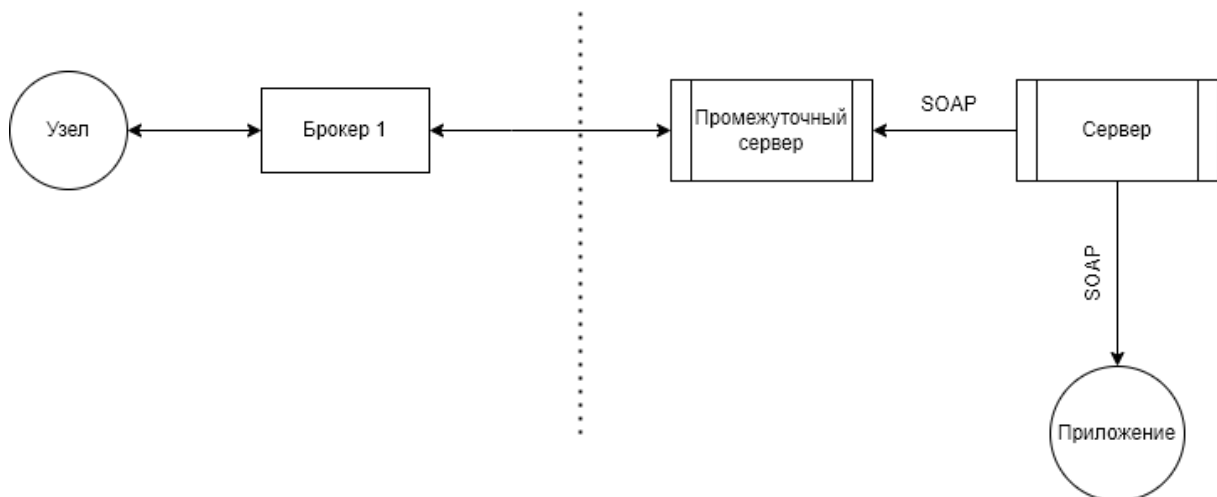
Следующий участок схемы взаимодействия элементов аппаратно-программного комплекса - участок с брокером, где помимо сбора, объединения и хранения полученных данных, ещё происходит формирование очередей запросов (рисунок 5). Для снижения нагрузки сети с большим количеством датчиков используется протокол MQTT, который предназначен для сетей с высокой задержкой и сетей с низкой пропускной способностью [16,17]. Сильная сторона протокола проявляется в простоте использования и совместимости с устройствами, которые будут использоваться в датчике. MQTT гарантировано доставляет сообщение от сервера, используя надежный транспортный протокол TCP [18].



**Рис. 5.** Участок с брокером

Участок с брокером допускает применение простого протокола передачи сообщений STOMP, который в сравнении с MQTT может взаимодействовать только от брокера к серверу, когда MQTT от брокера к узлу, так и от брокера к серверу. STOMP не занимается формированием очереди запросов, из-за чего могут возникнуть трудности с передачей сообщений [19].

Для решения задач получения результатов запросов с сервера и выставления требуемых параметров активации узлов или периода получения информации можно использовать протокол SOAP (рисунок 6).



**Рис. 6.** Участок с приложением

Распределяя информацию, может возникнуть необходимость в промежуточном сервере. Промежуточный сервер распределяет полученные результаты запросов, снижая нагрузку. Используя протокол SOAP, можно обеспечить согласованную передачу информации по запросу. Протокол будет использовать два механизма SOAP RPC и SOAP Message. SOAP RPC отвечает за удаленный запрос. SOAP Message используется для ответа на запрос. Использование простых механизмов позволит использовать любой

протокол прикладного уровня, которым можно регулировать передаваемую информацию по запросу [20].

Проведенный разбор схемы взаимодействия элементов аппаратно-программного комплекса с использованием датчика углекислого газа в лесу по отдельным участкам показал, что узлы, на которых установлены датчики, будут передавать собранные данные протоколом DDS. За взаимодействие узла с брокером будет отвечать XMPP, так как он использует надежный транспортный протокол TCP, в отличие от SOAP с транспортным протоколом UDP, который не гарантирует надежную передачу данных. За связь между брокером и сервером отвечает MQTT, соответствующий требованиям на сегменте. На участке с приложением протокол SOAP обеспечит согласованную передачу информации по запросу. Подобранные протоколы, взаимодействуя с технологией IoT, отвечают требованиям системы автоматического управления аппаратно-программного комплекса с использованием датчика углекислого газа в лесу.

### Библиографический список

1. Информационные технологии и системы: Учебное пособие для студентов высших учебных заведений / М. Р. Вагизов, М. А. Новикова, С. Ю. Степанов [и др.]. – Санкт-Петербург: Санкт-Петербургский государственный лесотехнический университет имени С.М. Кирова, 2020. – 78 с. – ISBN 978-5-00125-439-3.
2. Деревяго, Е. В. Умные вещи Интернета / Е. В. Деревяго // Автоматизация в промышленности. – 2020. – № 3. – С. 50-52. – DOI 10.25728/avtprom.2020.03.12.
3. Extensible Messaging and Presence Protocol (XMPP): Core, RFC 6120, March 2011
4. Селезнев, С. П. Архитектура промышленных приложений IoT и протоколы AMQP, MQTT, JMS, REST, CoAP, XMPP, DDS / С. П. Селезнев, В. В. Яковлев // International Journal of Open Information Technologies. – 2019. – Т. 7. – № 5. – С. 17-28.
5. Robotic systems in forestry / A. Boitsov, M. Vagizov, V. Pavlov [et al.] // IOP Conference Series: Earth and Environmental Science: 2, Saint Petersburg, Virtual, 18–19 марта 2021 года. – Saint Petersburg, Virtual, 2021. – P. 012034. – DOI 10.1088/1755-1315/806/1/012034.
6. Думов, М. И. "Умный лес" или IoT технологии мониторинга состояния лесных территорий / М. И. Думов // Актуальные вопросы в лесном хозяйстве: Материалы молодежной международной научно-практической конференции, Санкт-Петербург, 29–30 ноября 2017 года. – Санкт-Петербург: Санкт-Петербургский государственный лесотехнический университет им. С.М. Кирова, 2017. – С. 89-93.
7. Казанцев, Р. В. Особенности развития концепции Интернета вещей в лесной промышленности / Р. В. Казанцев, Д. В. Черник // Машиностроение: новые концепции и технологии: Всероссийская научно-практическая конференция студентов, аспирантов и молодых ученых,



- Красноярск, 23 октября 2020 года. – Красноярск: Федеральное государственное бюджетное образовательное учреждение высшего образования "Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева", 2020. – С. 61-65.
8. Чекотин, Р. С. Оценка действий оператора в полуавтоматических и многозадачных системах в лесозаготовках на основе использования IoT-технологий / Р. С. Чекотин, Н. О. Вербицкая // Системы. Методы. Технологии. – 2016. – № 4(32). – С. 188-196. – DOI 10.18324/2077-5415-2016-4-188-196.
  9. Мокринский А.А. Датчик исследований динамики углекислого газа в лесу / А.А. Мокринский, А.К. Бойцов, Н.В. Яготинцева, О.Н. Колбина // Актуальные вопросы в лесном хозяйстве: Материалы V международной научно-практической конференции молодых ученых, Санкт-Петербург, 11–12 ноября 2021 года. – Санкт-Петербург: Санкт-Петербургский государственный лесотехнический университет имени С.М. Кирова, 2021. – С. 118-122.
  10. ITU-T: Recommendations: Y Series: Y.2060/ International Telecommunication Union-Telecommunications /Overview of the Internet of things. 2012.
  11. OMG/ DDS v1.4 – the DDS specification / Object Management Group. 2015.
  12. Журавлева, Н. Г. Исследование возможностей применения технологии DataDistributionService для интеллектуальной обработки данных / Н. Г. Журавлева, М. В. Масюков, А. В. Радионов // Нейрокомпьютеры и их применение: тезисы докладов, Москва, 13 марта 2018 года. – Москва: Московский государственный психолого-педагогический университет, 2018. – С. 390-392.
  13. Пушкина, О. В. Влияние IoT (интернет вещей) и индустрии 4.0 на процессы снижения затрат и повышение эффективности производства / О. В. Пушкина, Я. П. Молоков // Вестник Московского университета им. С.Ю. Витте. Серия 1: Экономика и управление. – 2020. – № 4(35). – С. 87-92. – DOI 10.21777/2587-554X-2020-4-87-92.
  14. Брумштейн, Ю. М. Системный анализ совокупности концепций "умных объектов" городской среды в условиях развития информационно-коммуникационных технологий / Ю. М. Брумштейн, В. Ю. Гайфитдинова // Прикаспийский журнал: управление и высокие технологии. – 2017. – № 1(37). – С. 24-38.
  15. Constrained Application Protocol (CoAP) draft-ietf core-SoAP-18, June 28, 2013
  16. Дикий, Д. И. Протокол передачи данных MQTT в модели удаленного управления правами доступа для сетей Интернета / Д. И. Дикий, В. Д. Артемьева // Научно-технический вестник информационных технологий, механики и оптики. – 2019. – Т. 19. – № 1. – С. 109-117. – DOI 10.17586/2226-1494-2019-19-1-109-117.
  17. MQTT For Sensor Networks (MQTT-S) Protocol Specification, Version 1.2, November 14, 2013

- 18.MQ Telemetry Transport (MQTT) Specification v3.1, IBM, Eurotech
- 19.STOMP Protocol Specification, Version 1.2 // licensed under the Creative Commons Attribution v2.5 license. 2012. [Электронный ресурс]. Режим доступа: <https://stomp.github.io/stomp-specification-1.2.html> (дата обращения: 23.11.2021)
- 20.Simple Object Access Protocol (SOAP) [Электронный ресурс]. Режим доступа: <https://www.w3.org/TR/2000/NOTE-SOAP-20000508/> (дата обращения: 02.12.2021)

**А.О. Бутусова**, студентка 2 курса ИСиТ  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
nastbut2008@gmail.com

**А.Ю. Ничипорук**, студентка 2 курса ИСиТ  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
angelinanich@rambler.ru

**С.С. Колмогорова**, кандидат технических наук, доцент  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С. М. Кирова  
ss.kolmogorova@mail.ru

## **ПРОГРАММНАЯ ОБРАБОТКА «УМНОГО БРАСЛЕТА» ДЛЯ ЛЮДЕЙ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ**

**Аннотация:** В данной статье рассматривается проблема интеграции в общество людей с ограниченными возможностями. Предлагается создание умного браслета для людей с нарушением или полным отсутствием слуха, позволяющего чувствовать музыку через тактильные вибрации.

**Ключевые слова:** умный браслет, люди с ограниченными возможностями, нарушение слуха, микроконтроллер, электродинамический датчик вибрации.

«Технологии, на самом деле, о людях, а не об аппаратном или программном обеспечении» – Роберт Вейдей, глава департамента технологического развития Khemeia Consulting [4]. Действительно, в современном мире большая часть технических разработок направлена на то, чтобы упростить жизнь людей, избавить от рутинной работы, так или иначе, помочь человеку. Технические устройства стали неотъемлемой частью нашей повседневной жизни. Но в нашем обществе также присутствует категория людей, которая в большей мере нуждается в технических устройствах. Се-

годня проблема поддержки и помощи людям с ограниченными возможностями важна, как никогда раньше, так как с ускорением технического прогресса, появляется возможность устранения барьеров на пути к жизненным условиям без ограничений.

Данные исследования: [1], [2], [6], [7] свидетельствуют о проявлении интереса к музыке слабослышащих или совсем не воспринимающих звук людей. Однако данная категория людей всё ещё ограничена в возможности посещать массовые музыкальные мероприятия. Следует обратить внимание на то, что вышеупомянутый способ времяпрепровождения является одним из способов проведения досуга и социализации. Молодёжь с нарушением слуха не хочет быть исключением из сообщества единомышленников. Тем не менее, глобальная проблема аппаратных возможностей аналоговых приборов не позволяет в полной мере достичь подобной возможности.

Решение проблемы поддержки людей со сложными жизненными условиями важна в нашем обществе, о чем свидетельствует один из принципов третьей статьи Конвенции о правах инвалидов, принятой Генеральной Ассамблеей ООН 13 декабря 2006 года: «Принципами настоящей Конвенции являются: полное и эффективное вовлечение и включение в общество» [3]. Исходя из данного принципа, следует создать техническое устройство, способствующее социализации людей с ограниченными возможностями.

В результате изучения вышеупомянутой проблемы, было принято решение создать техническое устройство, способствующее интеграции особенных людей в общество.



**Рис.1.** Предполагаемый дизайн внешнего вида разработки

Умный браслет для помощи восприятия мелодий людям с нарушением или полным отсутствием слуха преобразует музыкальные произведения, загруженные в приложение «Vibic», в лёгкие вибросигналы и передаёт носителю устройства. Функционал оснащён базовым набором функций умного браслета: будильник, уведомления и часы. В разработку добавлен метроном для помощи обучения игре на музыкальных инструментах. Взаимодействие технического устройства с пользователем будет осуществляться при помощи тактильных вибраций и мобильного приложения.

Предлагается реализация проекта с помощью мобильных программных и аппаратных решений таких как:

- Наличие мобильного приложения: устройство включает в себя возможность использования людьми с ограниченными возможностями, учитывает спрос на простоту и доступность.
- Компактность аппаратной части, в данном случае капсулы умного браслета, обуславливается использованием в разработке нанотехнологий и анатомических возможностей восприятия вибрации человеком.

При реализации технического устройства следует уделить внимание таким составным элементам, как:

- Микроконтроллер с архитектурой ARM [5]. Стандартная архитектура микроконтроллеров, находящаяся в фитнес-браслетах, соединяет в себе все элементы и позволяет обеспечить связь с внешним устройством. Исходя из технических характеристик и аппаратной платформы к данной детали, предъявляются следующие требования: высокопроизводительное вычислительное ядро и наличие аналого-цифровой и цифро-аналоговой периферии.



**Рис.2.** Микроконтроллер с архитектурой ARM

- Электродинамический излучатель с параметрами от 100 Гр до 10 кГц [8].
- Необходимость использования дисплея с типом матрицы «AMOLED» [9], относящегося к категории «OLED». Преимущество над иным типом матрицы IPS заключается в том, что «AMOLED» дисплей можно использовать в изогнутом состоянии. Он довольно гибок.



**Рис.3.** Дисплей с типом матрицы «AMOLED»

Таким образом, при реализации устройства уделяется внимание особенностям характеристик используемых деталей: наличию различных номенклатур плат и модулей, что способствует уникальности разработки, и унифицированных программных модулей, то есть возможности подключения устройства к различным программным обеспечениям.

В результате создан прототип технического устройства с кроссплатформенным программным обеспечением для взаимодействия на основе сигнала Bluetooth с капсулой умного браслета, предназначенного для помощи восприятия мелодии людям с ограниченными возможностями.

В заключение рассмотрения данной проблемы, важно отметить, что ежедневно шанс максимально приблизить людей с ограниченными возможностями к безграничным довольно велик. Для его реализации уже существуют необходимые составные части, остаётся лишь грамотно сконфигурировать их. Улучшить жизненные условия, помочь людям – решаемая задача.

### **Библиографический список**

1. Ананьев Б.Г. Психология чувственного познания - М.: Изд-во АПН РСФСР, 1960. -277с.
2. Белик И.С. Музыка против глухоты. М., Владос, 2000г. - 157с.
3. Конвенция о правах инвалидов, Генассамблея ООН, 13 декабря 2006г [Электронный ресурс]. – Режим доступа: [https://www.un.org/ru/documents/decl\\_conv/conventions/disability.shtml](https://www.un.org/ru/documents/decl_conv/conventions/disability.shtml) (дата обращения 12.12.2021)
4. Корпоративное питание в эпоху Digital [Электронный ресурс]. – Режим доступа: <https://hr-portal.ru/cblog/korporativnoe-pitanie-v-epohu-digital>(дата обращения 12.12.2021)
5. Микроконтроллер ARM [Электронный ресурс]. – Режим доступа: <https://fivel.ru/imx/mimx8mm6cvtkzaa/pc14527169/>(дата обращения 11.12.2021)
6. Новости г. Реутов, 24 сентября 2018г [Электронный ресурс]. – Режим доступа: <https://reutovmedia.ru/novosti/item/2259-muzyka-dlya-slaboslyshashchikh>(дата обращения 12.12.2021)

7. Пономарёва, Е.В. Развитие музыкального восприятия слабослышащих школьников начальных классов /Е.В. Пономарёва// 2004г.
8. Электродинамический излучатель [Электронный ресурс]. – Режим доступа:<https://www.chipdip.ru/product/hc0901a>(дата обращения 11.12.2021)
9. AMOLED дисплей [Электронный ресурс]. – Режим доступа:<https://clck.ru/ahYG9>(дата обращения 11.12.2021)

**А.Ю. Ничипорук**, студентка 2 курса ИСиТ  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
angelinanich@rambler.ru

**А.О. Бутусова**, студентка 2 курса ИСиТ  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
nastbut2008@gmail.com

**С.С. Колмогорова**, кандидат технических наук, доцент  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С. М. Кирова  
ss.kolmogorova@mail.ru

## **РАЗРАБОТКА ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ "УМНОГО БРАСЛЕТА" ДЛЯ ЛИЦ СОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ**

***Аннотация:** В статье представлен процесс разработки пользовательского интерфейса “умного браслета”, также рассмотрено более подробно понятие пользовательского интерфейса и его необходимость в современном мире.*

***Ключевые слова:** пользовательский интерфейс, программа, фига, приложение, умный браслет.*

В современном мире интерфейсы окружают людей повсюду. Выбор услуг в банке через терминал, при снятии денег в банкомате, звонок или сообщение в смартфоне - все это пользовательский интерфейс. Пользовательский интерфейс часто понимают только как внешний вид программы. Но на самом деле пользователь воспринимает через него всю программу в целом, а значит, такое понимание является слишком узким. В действительности пользовательский интерфейс объединяет в себе все элементы и компоненты программы, которые способны оказывать влияние на взаимодействие пользователя с программным обеспечением (ПО). Этот термин означает внешнее программное обеспечение, которое помогает пользова-

телям взаимодействовать с платформой, иными словами, это проводник между человеком и программой. Пользовательский интерфейс является частью программного обеспечения и разработан так, чтобы пользователь понял, как работает программное обеспечение. Основная цель пользовательского интерфейса заключается в том, чтобы упростить взаимодействие со сложными техническими объектами. Интерфейсы помогли ускорить эти действия, сделав их доступными для всех. После получения команды человека с помощью интерфейса, устройство анализирует их и реагирует. Они передают информацию, отправляют команды, помогают обмениваться данными и выполняют другие полезные функции. Графические элементы действуют как индикаторы, ориентирующие пользователей. Если их убрать, то будет показан только набор символов. Одни из первых графических интерфейсов обеспечивали возможность с помощью клавиш или “мыши”:

- подводить курсор к любой части экрана;
- выделять на экране имя файла;
- оперировать выделенными данными независимо от других.

Интерфейсы пользователя бывают двух типов:

- процедурно-ориентированные: примитивные; меню; со свободной навигацией;
- объектно-ориентированные: прямого манипулирования.

Процедурно-ориентированный интерфейс основан на традиционной модели взаимодействия с пользователем - "процедура" и "операция". В рамках этой модели программное обеспечение предоставляет пользователю возможность выполнения некоторых действий, для которых пользователь определяет соответствие данных и следствием выполнения которых является получение желаемого результата.

Объектно-ориентированные интерфейсы основаны на модели взаимодействия с пользователем, ориентированной на манипулирование объектами предметной области. В рамках этой модели пользователю предоставляется возможность напрямую взаимодействовать с каждым объектом и инициировать выполнение операций, в процессе которых взаимодействуют несколько объектов. Задача пользователя формулируется как целенаправленное изменение некоторого объекта. Объект понимается в широком смысле слова - модель БД, системы и т.д. Объектно-ориентированный интерфейс предполагает, что взаимодействие с пользователем осуществляется посредством выбора и перемещения пиктограмм соответствующей объектно-ориентированной области.

Таким образом, процедурно-ориентированные интерфейсы:

1. Обеспечивают пользователю функции, необходимые для выполнения задач;
2. Акцент делается на задачи;
3. Пиктограммы представляют приложения, окна или операции;

4. Содержание папок и справочников отражается с помощью таблицы-списка.

Объектно-ориентированные интерфейсы:

1. Обеспечивает пользователю возможность взаимодействия с объектами;
2. Акцент делается на входные данные и результаты;
3. Пиктограммы представляют объекты;
4. Папки и справочники являются визуальными контейнерами объектов.

В настоящее время взаимодействие человека и компьютера строится на основе объектно-ориентированного графического интерфейса, в котором:

- объекты представляются в виде значков (иконок, пиктограмм), выбор которых с помощью, например, мыши приводит к активизации соответствующих объектов;
- оперирование объектами осуществляется в окнах – заранее заданных, очерченных рамками частях экрана;
- основным элементом программного управления является меню – выводимый на экран список команд, которые можно задать компьютеру;
- основным элементом аппаратного управления являются различные манипуляторы (мышь, трекбол, сенсорная панель и т.д.).

В пользовательском интерфейсе элементом управления компьютером является мышь, которая отображается на экране в виде указателя мыши - небольшого графического объекта, перемещающегося по экрану при движении мыши. Используется для того, чтобы указывать на различные объекты, запускать программы, выбирать пункт меню, перетаскивать объекты и т.д. Чаще указатель выглядит как маленькая стрелка, но его вид меняется в зависимости от операций, которые выполняет пользователь.

Основные задачи, для которых предназначен интерфейс: отображение и ввод информации; управление приложением; обмен данными с другими устройствами; взаимодействие с операционной системой.

На первых цифровых компьютерах проводились в основном расчёты. Эти компьютеры имели перфокарту, с помощью которой велось управление. Перфокарта — это листы тонкого картона, на которых были нанесены отверстия в определенном порядке. По ним считыватель определял двоичный код и выполнял соответствующую команду. Со временем стали усложняться вычислительные задачи, поэтому стали развиваться и устройства. Позже изобрели клавиатуру, которая стала посредником между человеком и программой. С ее помощью можно было разобратся в управлении информацией, иерархически организованной на жестких дисках первых компьютеров. В 1960-х годах Дуглас Энгельбарт изобрел компьютерную мышь. После этого появилась новая модель взаимодействия с ПК [1]. Дизайнеры первых пользовательских интерфейсов разработали меню. С по-



мощью мышки стало удобно буквально перетаскивать по экрану ПК значки.

Энгельбарт создал прототипы современных программ для работы с текстами и графикой, навигации по файловой системе. Идеи Энгельбарта воплотились в первом персональном компьютере с графическим интерфейсом — Xerox Alto. Появление этого девайса вдохновило Стива Джобса и Билла Гейтса на разработку собственных операционных систем.

Разработка, представленная в статье, предполагает собой интерфейс пользователя, которому принадлежит “умный браслет”, для дальнейшего взаимодействия с устройством. Если каждое приложение или программа, установленная на компьютере, планшете или смартфоне — это помощник, то интерфейс — это способ взаимодействовать с ней. Так как пользовательский интерфейс помогает двум объектам понимать друг друга и обмениваться информацией, нам необходимо было создать его в простой и понятной для пользователей структуре. Через пользовательский интерфейс мы имеем возможность получить доступ к новым возможностям, которые дает приложение или программа.

В процессе реализации поставленной задачи использованы вспомогательные инструменты из браузера. Таким инструментом служила программа для разработки прототипов приложений – Figma [2]. Именно в ней создается прототип нашего приложения для облегчения визуализации и оценки использования продукта. Figma доступна в браузере каждому пользователю, так как является бесплатной программой. Также она работает достаточно плавно и является кроссплатформенным инструментом, что дает возможность использовать приложение независимо от вида платформы. При разработке пользовательского интерфейса применяется определенный свод правил, который помогает сделать приложение удобным в использовании. Так, например, одним из требований является концепция “предсказуемости”. Пользователь заранее должен интуитивно понимать, какое действие выполнит программа после получения его команды. Также доступность – один из наиболее важных элементов дизайна. Человеку необходимо понять, как работает приложение, иначе у него не будет желания использовать эту программу. Все это старательно воплощено при создании нашего приложения. Также учтены такие важные черты удобного интерфейса, как: минимализм, уверенность, соответствие контексту, отзывчивость, привлекательность и эффективность. В разработке интерфейса использованы принципы атомарного дизайна [3,4]. В подходе атомарного дизайна мы создаем пользовательский интерфейс с нуля, то есть, начиная с мельчайших элементов, заканчивая дизайном конечных страниц. Это обеспечивает согласованность всей конструкции.

Результат разработки пользовательского интерфейса представлен на рисунке 1 в виде скриншотов экрана ПК:

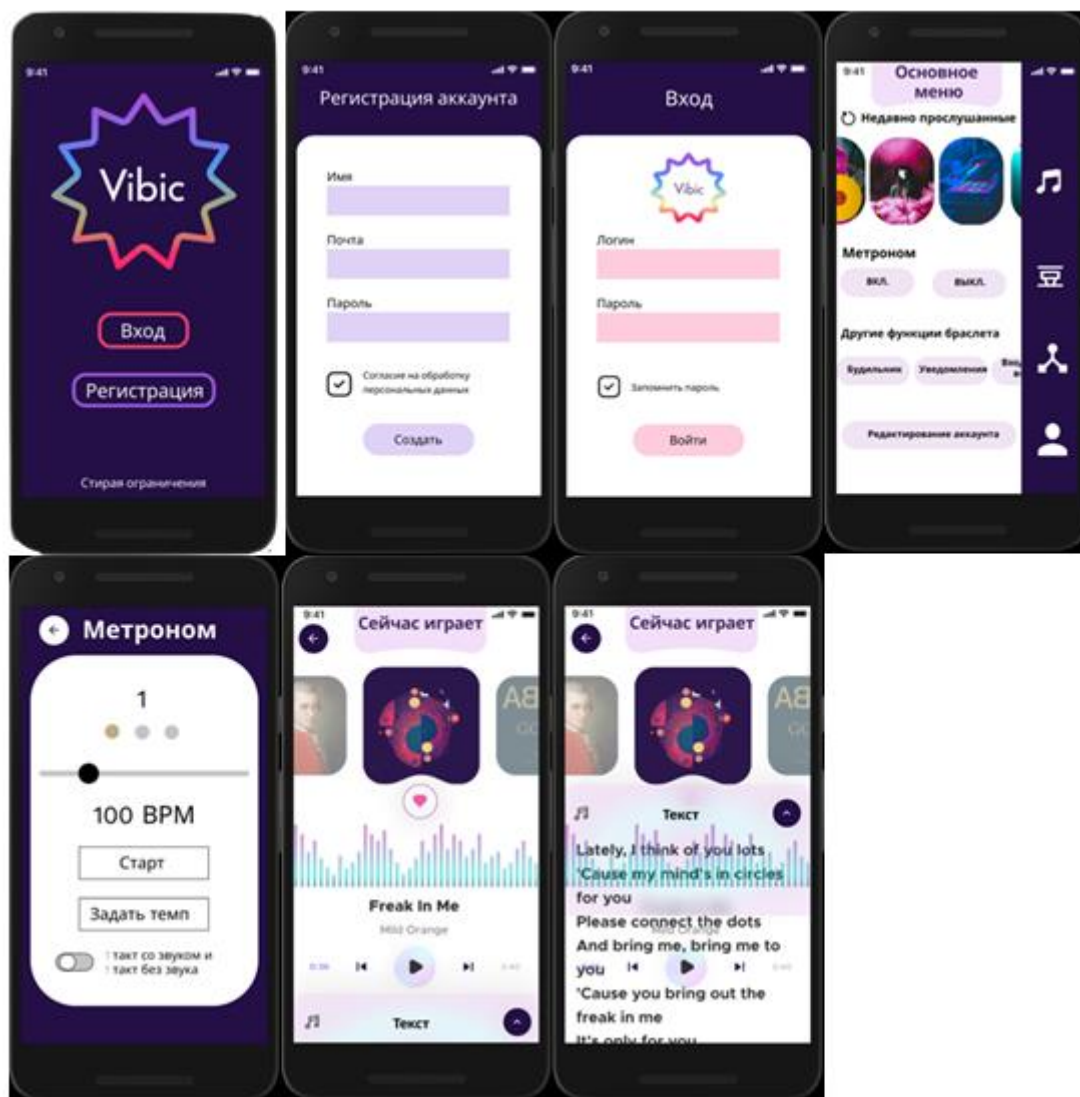


Рис. 1. Скриншот экрана работы в программе Figma

Чтобы оптимизировать приложение, обычно необходимо провести подробные тесты. Мы провели тестирование на уровне прототипирования, то есть сделали предварительный макет дизайна, оценили совместимость контента и синтаксиса. Также в нашем случае предполагается более широкое использование ассистивных технологий (под ними понимаются устройства, ПО, оборудование, направленные на улучшение функциональных возможностей людям с ограниченными возможностями) в оценке надежности проектируемого пользовательского интерфейса «умного браслета».

Таким образом, разработанный интерфейс выполнен с соблюдением представленных требований и задач, согласно направлению разработки. Учтены такие требования, как естественность, согласованность, понятность, простота, гибкость и привлекательность. При выполнении этих требований интерфейс становится удобным в процессе использования людям с ограниченными возможностями.

### Библиографический список

1. История пользовательского интерфейса [Электронный ресурс] URL : <https://www.iphones.ru/iNotes/brief-history-of-ui>
2. Figma: the collaborative interface design tool [Электронныйресурс] URL : [www.figma.com](http://www.figma.com)
3. Королькова А. Живая типографика / Александра Королькова. - М.: 2011. - 224 с.
4. Чихольд Ян. Новая типографика. Руководство для современного дизайнера / Ян Чихольд. - М.: "Студия Артемия Лебедева", 2011. - 853 с.

**Н.А. Карманова**, аспирант  
факультет Безопасности информационных технологий  
Кафедра информационных систем и технологий  
Университет ИТМО  
[karmanova.ifmo@gmail.com](mailto:karmanova.ifmo@gmail.com)

**А.Г. Карманов**, кандидат технических наук, доцент  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С. М. Кирова  
[karmanov.nip@gmail.com](mailto:karmanov.nip@gmail.com)

### ПОВЫШЕНИЕ НАДЁЖНОСТИ СИСТЕМ ДИСТАНЦИОННОГО ЗОНДИРОВАНИЯ ЗЕМЛИ

***Аннотация:** При решении специальных задач радиолокационные системы высокого разрешения обладают большей эффективностью, по сравнению с оптико-электронными, при использовании их в качестве полезной нагрузки на БЛА. Это объясняется всепогодностью, работой в условиях задымления, взрывов и пр., различного времени суток, использованием различной длины волны, возможностью зондирования земли, покрытой лесными массивами, комплексированием с различной информацией, селекцией и сопровождением подвижных объектов. Бортовые РЛС с АФАР сталкиваются с проблемой охлаждения. Решение этой задачи на борту легких и средних БЛА требует нового технологического решения.*

***Ключевые слова:** БЛА, АФАР, радиовидение, теплоотвод, АКК «Скелетон», теплопроводность*

Беспилотные авиационные системы и комплексы, способные решать в автоматическом режиме народно-хозяйственные, научно-исследовательские и специальные задачи, являются приоритетным направлением в прорывных мультиплатформенных технологиях. В гражданском секторе беспилотные авиационные системы большого радиуса действия

российского производства занимают лидирующие позиции на мировом рынке [1].

Эффективность использования беспилотного летательного аппарата (БЛА) напрямую зависит от уровня программно-аппаратных и технических средств полезной нагрузки, предназначенных для решения поставленных задач [2]. Использование БЛА в качестве носителя накладывает определенные требования на разрабатываемую бортовую аппаратуру. В первую очередь, ограничения касаются массогабаритных характеристик бортового оборудования и потребляемой мощности [3].

Использование беспилотной авиации в сложных условиях, например плохой видимости из-за погоды или пожара снижает риски, связанные с человеческим фактором. Но, с другой стороны, накладывает повышенные требования к полезной нагрузке.

Становится понятным, что наиболее эффективно использование бортовых радиолокационных станций с антенной фазированной решеткой, что позволяет достигать высокого разрешения [3]. При всех преимуществах антенных фазированных решеток существует ряд проблем, основная из которых связана с рассеиванием мощности из-за твердотельной элементной базы. По сравнению с обычным радаром беспилотного летательного аппарата с воздушным охлаждением, радар с антенной фазированной решеткой более надёжен, однако потребляет больше электроэнергии и требует более интенсивного охлаждения. Но антенная фазированная решетка может обеспечить намного большую передаваемую мощность, что необходимо для большей дальности обнаружения цели [4].

Таким образом, целью исследования стало повышение надежности систем дистанционного зондирования земли за счет разработки концепции теплоотвода антенных фазированных решеток для бортовых радиолокационных станций БЛА.

Концепция построения системы теплоотвода была найдена в новых материалах, а именно в алмаз-карбидокремниевом композите «Скелетон».

## 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ ПРОЕКТА

Рассмотрев такие вещества, как разновидности карбидокремниевых керамик и алмаза, становится понятным, что получение композитов (Усиление SiC-керамик алмазными частицами) должно приводить к получению композитов с уникальной комбинацией жесткости, теплопроводности, твердости и износостойкости. (Табл. 1)[5]

Таблица 1

Алмаз и SiC керамики

	SSiC	RBSiC	Алмаз
Плотность, г/см <sup>3</sup>	3.0	3.1	3.5
Модуль упругости, ГПа	400	400	1100
Твердость, ГПа	23	21	100

Теплопроводность Вт/м*К	80	150	2000
ТКЛР, $10^{-6}/\text{К}$	4.5	4.5	1.5

Изменение линейных размеров после преобразования заготовки в композит не превышает 0.2%. Это делает возможным изготовление деталей сложных форм, что обеспечивает требования различных конструкций АФАР (Рисунке 1).



Рис. 1. Формы деталей

АКК «Скелетон» имеет очень высокую жесткость: его модуль упругости достигает 830 ГПа, что является высшим результатом для конструкционных материалов. (Табл. 2)

Таблица 2

#### Упругие характеристики АКК «Скелетон»

Материал	Алюминий	Сталь	SiC	Волфрам	Твердый сплав	АКК
Е, ГПа	70	200	00	420	660	650-830

АКК «Скелетон» значительно превышает другие конструкционные материалы по удельной жесткости (отношению модуля упругости к плотности) и по скорости звука в нем. (Табл.3)[7]

Таблица 3

#### Упругие характеристики АКК «Скелетон»

Материал	Сталь	Алюминий	Твердый сплав	Бериллий	АКК
$E/\rho, (\text{км/с})^2$	25	26	43	160	200-240
$V_{зв}, \text{км/с}$	5	5,1	6,6	12,7	14-15,5

Согласно таблице 4, среди представленных материалов, рассматривая показатели по плотности, теплопроводности, коэффициенту линейного расширения, сопротивлению АКК «Скелетон» превосходит многие материалы и является диэлектриком. [7]

## Высокотеплопроводные материалы

Материал	$\rho$ , г/см <sup>3</sup>	$\lambda$ , Вт/м*К	$\alpha$ , см <sup>2</sup> /с	ТКЛР. $10^{-6}$ , 1/К	R, мкОм*м
Cu	8,9	390	1,15	17	0,017
Al	2,7	220	0,85	24	0,03
Si	2,3	150	0,9	2,3	$10^4$
AlN	3,1	160	0,85	4,5	$10^{15}$
BeO	3,0	250	0,85	7,0	$10^{13}$
SiC	3,2	150	0,73	4,3	$10^9$
АКК Скелетон»	3,3	300-600	1,8-3,2	2,0-2,3	$10^5$
Алмаз	3,5	2000	10,3	1,5	$10^{16}$

## 2. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ ПРОЕКТА

## 2.1 Аналитический расчет теплопроводности

Для аналитического расчета теплопроводности использовалась программа «Композит». Была получена зависимость концентрации алмазной крошки теплопроводных свойств от концентрации алмазной крошки. (Рисунок 2)

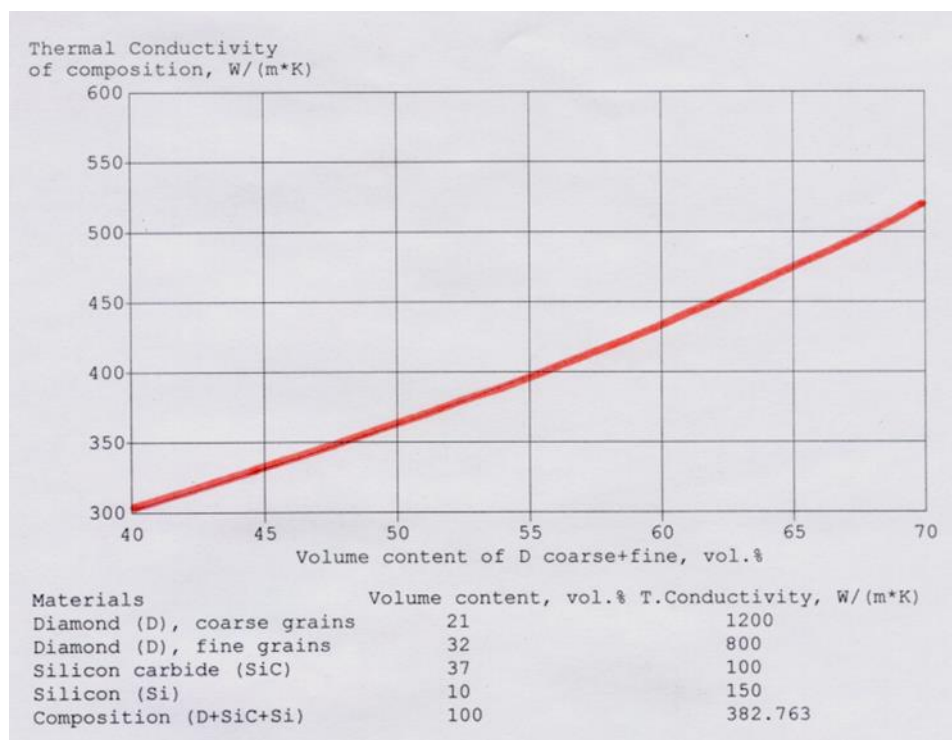
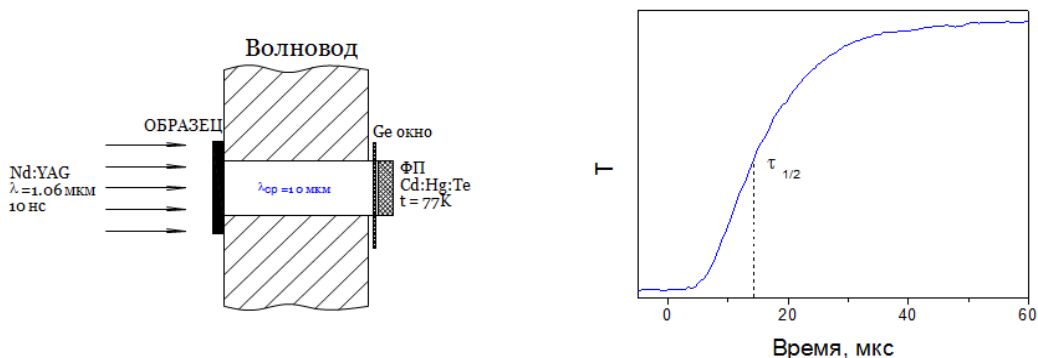


Рис. 2. Зависимость концентрации алмазной крошки теплопроводных свойств от концентрации алмазной крошки

## 2.2 Экспериментальный расчет теплопроводности

На представленной схеме (Рисунок 3) показывается, как экспериментально определяется теплопроводность композитов. Представлена упрощенная схема. На испытуемый образец подается тепловой удар лазером ближней ИК-области и фиксируется время достижения половины амплитуды теплового удара.



$$D = 1.38d^2 / \pi^2 t_{1/2}$$

$$\lambda = \rho C D$$

где  $Q$  - поглощенная энергия лазерного импульса (Дж/см<sup>2</sup>),  $C$  - теплоемкость,  $d$  - толщина образца,  $t_{1/2}$  - время достижения половины амплитуды  $T_{max}/2$ .

Рис. 3. Определение теплопроводности композитов

В таблице 5 представлены результаты нашего эксперимента. Представлены аналитические и экспериментальные данные при процентном соотношении материалов.

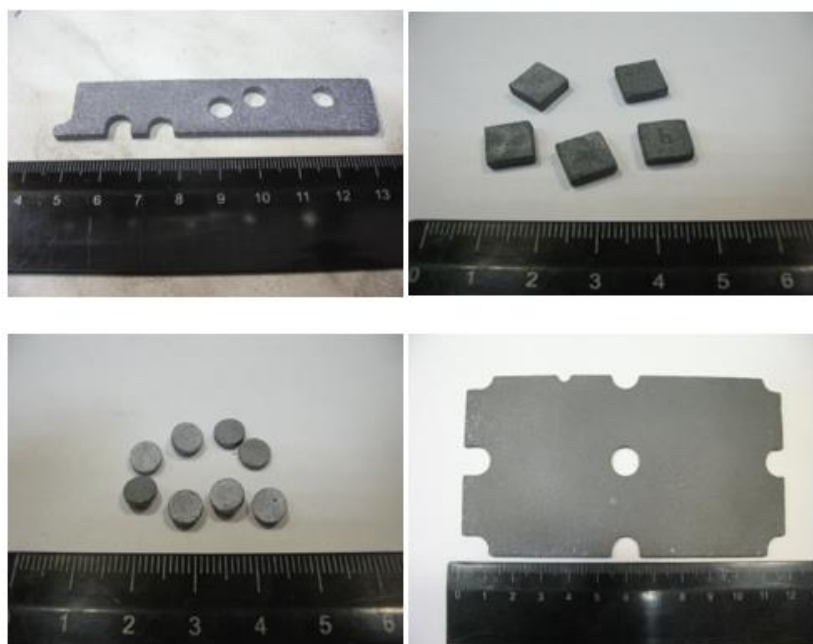
Таблица 5

### Результаты эксперимента

Состав материала, % об.			Теплопроводность, Вт/м К	
Алмаз	SiC	Si	эксперимент	расчет
43	46	11	280 ± 0,009	264
52	37	11	370 ± 0,010	365
58	30	12	490 ± 0,011	480
59	30	11	510 ± 0,020	480
68	25	7	610 ± 0,013	580
70	23	7	600 ± 0,023	605

Вид экспериментальных образцов теплоотводов электронных блоков и компонентов показан на рисунке 4.

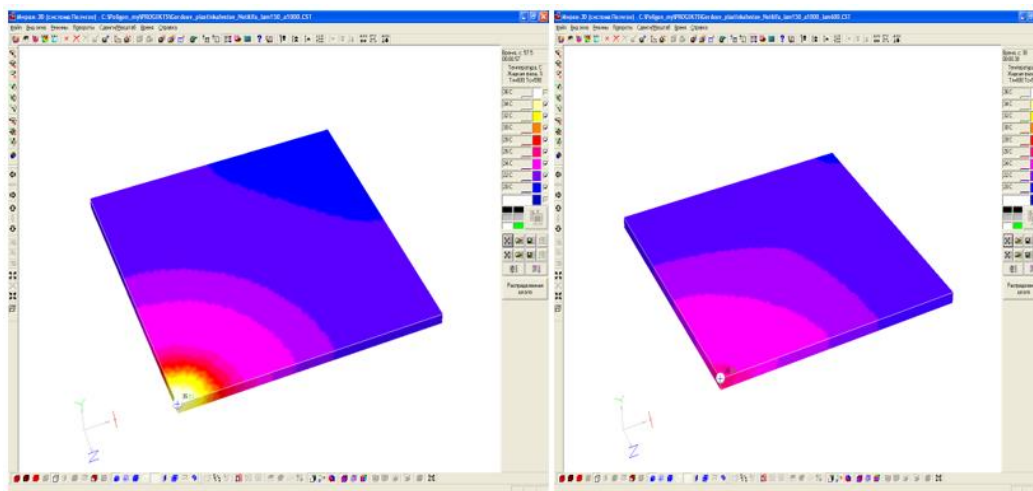




**Рис. 4.** Экспериментальные образцы теплоотводов электронных блоков и компонентов

### 2.3 Моделирование процесса теплоотвода

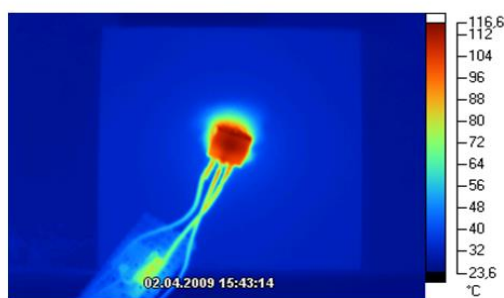
Далее представлены результаты моделирования теплопередачи пластин со вставками с разной теплопроводностью (рисунок 5)



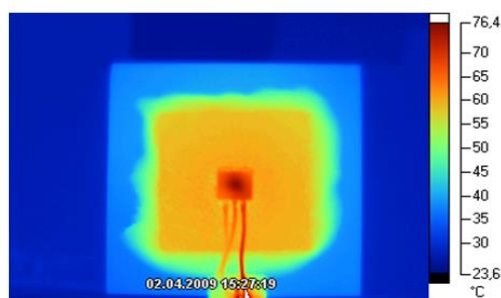
**Рис. 5.** Результаты моделирования теплопередачи пластин со вставками с разной теплопроводностью

Из рисунка 6 видно, что применение АКК «Скелетон» в качестве теплоразгрузочного элемента существенно снижает установившуюся температуру электронных компонент (в данном случае транзистора). Исходя из приведенных в таблице параметров видно, что применение АКК «Скелетон» позволит существенно снизить нагрев электронного компонента.





**Алюминиевая пластина (85x85 мм)  
+ тепловыделяющий элемент  
(транзистор)**



**Алюминиевая пластина (85x85 мм) + пластина АКК (44x44x3 мм)  
+ тепловыделяющий элемент  
(транзистор)**

**Рис. 6.** Изменение теплового поля при использовании АКК «Скелетон»

Исходя из приведенных в таблице 6 параметров видно, что применение АКК «Скелетон», по отношению к широко применяемому теплопроводящему материалу медь (Cu), позволяет существенно снизить тепловые нагрузки электронных компонентов, и как следствие, улучшить габаритно-массовые характеристики приборов.

Таблица 6

**Сравнение АКК «Скелетон» и Cu**

Материал	Плотность, г/см <sup>3</sup>	Теплопроводность l, Вт/м*К	Температурная проводимость а, см <sup>2</sup> /с	ТКЛР.10 <sup>-6</sup> , 1/К
Cu	8,9	390	1,15	17
АКК «Скелетон»	3,3	500-600	2,6-3,2	2,0-2,3

### Библиографический список

1. Вендик О. Г. Антенны с немеханическим движением луча. — М.: Советское радио, 1965
2. Воскресенский Д. И., Гостюхин В. Л., Максимов В. М., Пономарёв Л. И. Антенны и устройства СВЧ / Под ред. Д. И. Воскресенского. Учебник. — 2-е изд. — М.: Радиотехника, 2006. — 376 с. — ISBN 5-88070-086-0.
3. Антенны и устройства СВЧ. Проектирование фазированных антенных решёток. Учебное пособие / Под ред. Д. И. Воскресенского. — М.: Радио и связь, 1994. — 592 с. — ISBN 5-256-00404-2.
4. Сазонов Д. М. Антенны и устройства СВЧ. Учебник. — М.: Высшая школа, 1988. — 432 с. — ISBN 5-06-001149-6.
5. Gordeev S.K. Advanced Composite Materials on the Diamond Base. Diamond Based Composites. Dordrecht: Kluwer, 1997, p.1.
6. Гордеев С.К., Жуков С.Г., Данчукова Л.В., Экстрем Т.С. Особенности получения композиционных материалов на основе алмаза, карбида кремния при низких давлениях. — Неорганические материалы, 2001, т.37, №6, с.691–696.

7. Поляков В.П., Ножкина А.В., Чириков Н.В. Алмазы и сверхтвердые материалы.– М., 1990.

**М.Е. Кутузов**, студент 2 курса ИСиТ  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
optilik045@gmail.com

**С.С. Колмогорова**, кандидат технических наук, доцент  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С. М. Кирова  
ss.kolmogorova@mail.ru

## **ПРОГРАММНАЯ ОБРАБОТКА ЛАБОРАТОРНЫХ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА**

***Аннотация:** Реализован процесс обработки изображений легких для обнаружения повреждений COVID-19 с применением нейронных сетей написанных на языке Python с использованием библиотек TensorFlow и Numpy. Данный алгоритм выдает процентную вероятность наличия COVID-19 повреждений в легких.*

***Ключевые слова:** искусственный интеллект в медицине, нейронные сети, COVID-19, прогнозирование, Python.*

### **Введение.**

Сегодня к искусственному интеллекту (ИИ) относят программные средства с набором алгоритмов и методов, которые могут решать интеллектуальные задачи так же, как это сделал бы человек. К примеру, искусственный интеллект способен:

- прогнозировать различные ситуации;
- оценивать информацию и формулировать заключительную оценку;
- анализировать данные и искать скрытые закономерности.

### **Применение нейронных сетей**

Нейронные сети сегодня активно применяются в разработке интеллектуальных систем, в том числе и в медицине, благодаря их способности к обучению. Механизм работы искусственных нейросетей повторяет принцип биологических. В цифровом исполнении нейронная сеть представляет

собой граф с тремя и более слоями нейронов, которые соединяются между собой.

В процессе обучения входные нейроны получают данные, обрабатывают их на внутреннем слое нейросети, а на выход поступают результаты. Если полученный результат в процессе обучения не устраивает исследователей, они меняют вес соединений и заново обучают сеть. При этом успешность процесса и достоверность результатов зависит от количества входных данных – чем их больше, тем лучше.

Нейросети могут применяться в медицине разными способами. Например, пациент делает запрос «головная боль», «высокая температура», «озноб», а нейронная сеть анализирует тысячи или миллионы карточек других людей и на основе их диагнозов может предположить заболевание у человека, сделавшего запрос.

Естественно, нейросеть не может на 100% утверждать, что с названными симптомами у пациента, например, грипп, однако она предполагает такой диагноз в соответствии с заключениями врачей по другим медкартам.

## **Материалы и методы**

### **Используемые данные**

КТ-изображения легких пациентов с Covid-19, использованные в исследовании, были взяты из двух наборов данных, созданных и собранных в виде метаданных Bganga, который затем сделал его общедоступным через Github для открытого доступа. В связи с этим в исследовании было использовано в общей сложности 142 КТ-изображений легких. Размеры рассматриваемых изображений составляют от 115×98 до 2024×1523 и сильно различаются. Прежде всего, для уточнения области интереса на изображениях производилось кадрирование с включением области легких. В рамках исследования также было использовано 160 КТ-изображений легких, не связанных с Covid-19. Рассматриваемые изображения были аналогичным образом взяты из исследовательской базы данных открытого доступа LIDC-IDRI, ранее опубликованного набора данных. Процессы, примененные к изображениям Covid-19, также применялись к изображениям без Covid-19.

### **Локальный двоичный (бинарный) шаблон (LBP)**

В базовой версии для построения ЛБШ используется окно 3×3. Восемь соседних значений сравниваются со значением центрального пикселя, и в зависимости от результата сравнения центральному пикселю ставится в соответствие восьмибитное число – код ЛБШ (рис. 1).

$f_1$	$f_2$	$f_3$
$f_8$	$f_0$	$f_4$
$f_7$	$f_6$	$f_5$

Рис. 1. Нумерация пикселей

Определим изображение как функцию яркости пикселя  $f(x, y)$ . Код ЛБШ вычисляется в каждом пикселе с использованием значений пикселей в круговой окрестности  $f_i = f(x_i, y_i), i = 1, N$ , где  $N$  – количество пикселей, попадающих в окрестность, ограниченную окружностью радиуса  $R$  с центром в пикселе  $f_0$  (рис. 1), следующим образом:

$$LBP(N, R) = \sum_{i=1}^N I(f_i - f_0)2^i \quad (1)$$

В соответствии с формулой локального двоичного шаблона (2) значение центрального пикселя по очереди сравнивается со значениями соседних пикселей путем вычитания одного значения из другого. Если разность двух значений больше или равна нулю, то пикселю присваивается значение 1, если же меньше нуля, то присваивается значение 0:

$$s(m) = \begin{cases} 1, & m \geq 0 \\ 0, & m < 0 \end{cases} \quad (2)$$

На рисунке 2 показан пример работы локального бинарного двоичного шаблона.

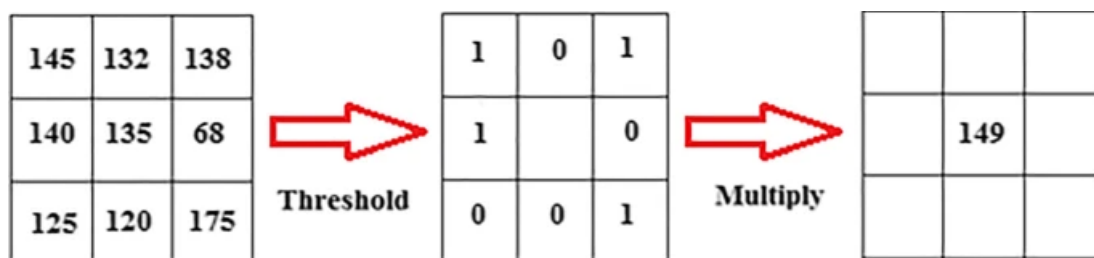


Рис. 2. Пример работы LPB

После этого двоичные значения пикселей записывается как двоичный код (пример: 10101001). Данный двоичный код переводится в десятичное число и присваивается данным 9 пикселям.

Следует также отметить, что оператор ЛБШ очень эффективен благодаря своей низкой вычислительной сложности. Главным недостатком данного вида шаблонов является чувствительность к шумовым искажениям.

### Подготовка изображений

В случае применения LBP к изображениям видно, что размеры изображений уменьшаются. Например, когда операция LBP с радиусом 2 применяется к изображению  $224 \times 224$ , получается изображение  $220 \times 220$ . Причина этого заключается в том, что эта операция не может быть применена к пикселям в начальных и конечных строках и столбцах. Диаметр LBP, используемого в этом исследовании, равен 2. Чтобы это уменьшение размера не вызвало каких-либо проблем при операциях, упомянутых выше, размеры изображения, полученного после операций LBP, были восстановлены до  $448 \times 448$ . Использование процесса LBP в исследовании позволяет нам получать новые изображения по сравнению с оригинальными, отражающими местные особенности. Таким образом, общая глубина изображения была увеличена.

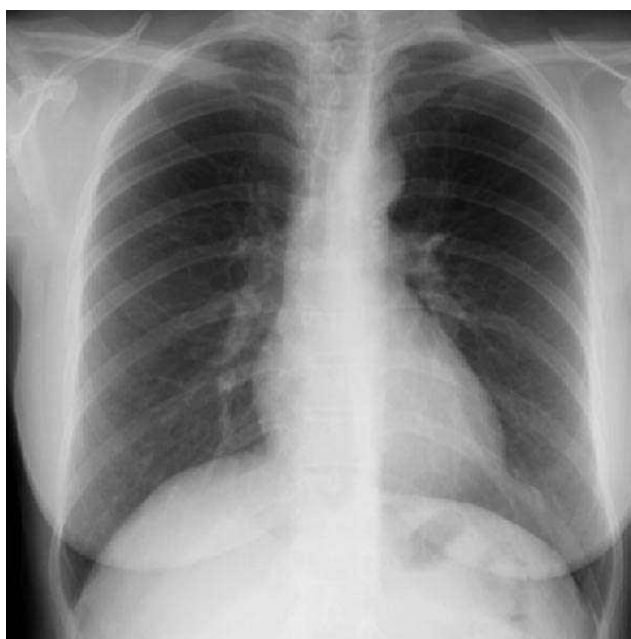


Рис. 3. Пример используемых изображений

### Сверточная нейронная сеть (CNN)

Глубокое обучение реализуется с использованием слишком большого количества уровней в реализации процесса обучения. Сверточная нейронная сеть (CNN) является наиболее часто используемой моделью глубокого обучения. Эта модель широко используется, особенно в приложениях обработки изображений в последние годы. CNN состоит из нескольких слоев, таких как сверточный слой, функция активации, объединение и полностью подключенный слой. Сверточные слои часто предназначены для следования друг за другом и позволяют нам получать шаблоны объектов от низкоуровневых объектов изображений до высокоуровневых объектов. Функции активации в архитектуре CNN могут быть определены как функции, которые приводят входящие входные данные в определенный диапазон или принимают некоторые входные значения, снова устраняя некоторые из них. С другой стороны, объединение слоев позво-

ляет уменьшить размер матриц объектов за счет выборки. Что касается полностью связанного слоя, то это слой, на котором выполняется процесс классификации в соответствии с признаками, полученными с помощью свертки, функции активации и объединения. Этот слой работает как классическая искусственная нейронная сеть. Перед процессом классификации выполняется преобразование матриц признаков в векторы признаков, то есть процесс сглаживания. В этом контексте на рисунке 4 показана общая архитектура классификатора CNN.

## Сверточные сети в TensorFlow

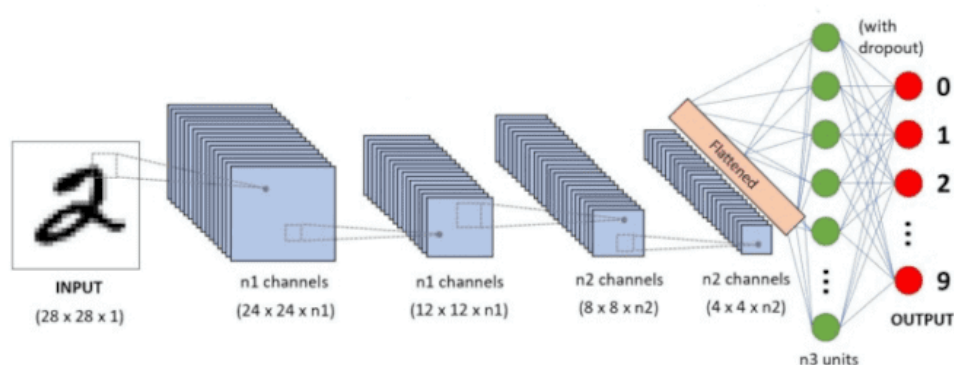


Рис. 4. Наглядная работа сверточной сети

На рисунке 5 показана программная реализация CNN.

```

model = Sequential()
model.add(Conv2D(32, kernel_size=(3,3), activation='relu', input_shape = (224,224,3)))
model.add(Conv2D(64,(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout(0.25))
model.add(Conv2D(64,(3,3), activation = 'relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))
model.add(Conv2D(128,(3,3), activation = 'relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss = keras.losses.binary_crossentropy, optimizer='adam', metrics = ['accuracy'])

```

Рис. 5. Программная реализация CNN

### Результаты исследования

В рамках исследования был реализован алгоритм искусственного интеллекта на основе глубокого обучения для автоматической диагностики компьютерных изображений легких для ранней диагностики заболевания Covid-19. В этом исследовании было использовано в общей сложности 400

изображений компьютерной томографии легких, 190 из которых принадлежали пациентам с диагнозом Covid-19 и 210 субъектам, не имеющим Covid-19.

Поскольку используемые изображения имеют очень разные размеры, изображения были изменены по размеру, и размеры изображения были установлены на  $448 \times 448$ . После чего на вход сверточной нейронной сети подавались изображения легких, а на выходе мы получали число от 1 до 0.01, которое является показателем наличия или отсутствия поражений.

```
Epoch 1/10
164/164 [=====] - 12s 73ms/step - loss: 0.6019 - accuracy: 0.7846 -
val_loss: 0.5413 - val_accuracy: 0.7115
Epoch 2/10
164/164 [=====] - 10s 62ms/step - loss: 0.3667 - accuracy: 0.8724 -
val_loss: 0.5326 - val_accuracy: 0.7532
Epoch 3/10
164/164 [=====] - 10s 62ms/step - loss: 0.2427 - accuracy: 0.9240 -
val_loss: 0.5632 - val_accuracy: 0.7596
Epoch 4/10
164/164 [=====] - 10s 60ms/step - loss: 0.1744 - accuracy: 0.9488 -
val_loss: 0.7198 - val_accuracy: 0.7147

<tensorflow.python.keras.callbacks.History at 0x7fc4a05c94d0>
```

Рис. 6. Пример полученных результатов работы алгоритма

На рисунке 6 показана работа нейронной сети и точность ее работы. На данный момент, точность работы нейронной сети состоящей из 15 слоев составляет примерно от 65 до 75 % в связи малых мощностей ПК и небольшого количества слоев сети.

Представленный алгоритм необязательно должен использоваться только в медицинской сфере, его можно применять в любых сферах деятельности, где нужно обрабатывать изображения с дальнейшими результатами.

## Стоимость

Создание и внедрение систем искусственного интеллекта требует серьезного финансирования. Высокая стоимость связана во многом с необходимостью обучать программу, настраивать ее под данные, накопленные в конкретном медицинском учреждении. Кроме того, она требует специального обслуживания, для которого потребуется квалифицированная и мотивированная команда.

## Библиографический список

1. Find, install and publish Python packages with the Python Package Index URL : <https://pypi.org/>
2. COVID-19 imagedatacollection



3. URL : [https:// github .com/ieee8023/ covid-chestxray-dataset](https://github.com/ieee8023/covid-chestxray-dataset)

**Е.В. Тетюев** магистрант 2 курса ИСиТ  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
unrou@mail.ru

**С.П. Хабаров**, кандидат технических наук, доцент  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С. М. Кирова  
SergHabarov@yandex.ru

### **РАЗРАБОТКА СЕТЕВОГО ПРИЛОЖЕНИЯ С КОНТРОЛЕМ ДОСТАВКИ И СИНХРОНИЗАЦИЕЙ ДАННЫХ С ПОМОЩЬЮ WEBSOCKET НА ОСНОВЕ PYTHON3**

***Аннотация.** В статье рассматриваются принципы обмена сообщениями на прикладном уровне между клиентами и серверной частью приложения для осуществления гарантированной регистрации клиентов на сервере, гарантированной доставки сообщений и возможности синхронизации отсутствующих данных. Работоспособность решения показана на примере создания клиент-серверного приложения на базе Python3.*

***Ключевые слова:** сетевые приложения, обмен сообщениями, клиент-серверная архитектура, websocket, python.*

**Введение.** Большинство разрабатываемых сегодня приложений не являются в полной степени автономными и, при наличии подключения к глобальной сети, используют её для обмена данными, чаще всего с сервером разработчика. Некоторые приложения могут обойтись без связи с сервером, однако клиент-серверные приложения целиком зависят от обмена информации по сети. Ключевыми вопросами при этом являются надёжность доставки, целостность и сохранность передаваемой информации. Поскольку шифрование – это отдельная и весьма объёмная тема, в данной статье рассматриваются именно вопросы доставки и синхронизации сообщений. Некоторые наивно полагают, что транспортный уровень сетевой модели OSI, а именно, протокол TCP обеспечивает достаточную надёжность передачи данных. Однако TCP и его производные отвечают за пересылку данных только при наличии соединения между хостами. В ситуации, когда хост временно не может принять сообщение или вообще отключен от сети, пакет TCP не дойдёт до адресата и будет уничтожен после заданного количества прыжков. В этом случае для обеспечения гарантированной доставки сообщений требуется использовать другие механизмы.

Данное исследование заключается в разработке модели обмена данными и создании приложения для её тестирования. Стоит отметить, что авторы уже проводили исследование подобной модели ранее [1], и новая модель является развитием первой с учётом полученных замечаний. В свою очередь, первая модель основана на примере клиент-серверного приложения с поддержкой множественных подключений [2]. Помимо усовершенствованной схемы передачи и синхронизации данных, в новом исследовании поменялась и техническая часть. В первой модели в роли сервера выступало приложение, написанное на Node.JS, а для данного исследования для серверного приложения был выбран язык Python3, по большей части ради разнообразия и для закрепления знаний об этом языке программирования у исследователя. Забегая вперёд, можно с уверенностью сказать, что, в рамках исследуемой модели, существенной разницы между подходами к разработке приложения нет, а все различия продиктованы разными конструкциями и синтаксисом языка программирования.

### **Постановка задачи.**

Модель описывает основные принципы передачи и синхронизации данных на прикладном уровне сетевой модели OSI, а приложение для тестирования состоит из сервера и клиентов, количество которых ограничено возможностями WebSocket модуля Python3. Клиенты получают некие сообщения (в данном случае используется ручной ввод от пользователя) и передают их на сервер, после чего тот рассылает сообщения всем подключённым клиентам. Графический интерфейс клиентов напоминает крайне упрощённый мессенджер, сервер не имеет графического интерфейса. Требуется обеспечить надёжную доставку сообщений между клиентами и сервером, а также возможность синхронизации уже имеющихся сообщений в случае, если клиент по каким-то причинам отключился от сети или выключился в принципе.

### **Построение модели.**

В качестве модели в данном исследовании рассматривается клиент-серверное приложение с возможностью множественных подключений и процедурной обработкой событий подключения клиентов и приёма сообщений. Обмен сообщениями происходит по протоколу WebSocket, сообщения представляют собой данные формата String, первые несколько символов которых зарезервированы под тип сообщения или команду. По сравнению с упомянутой ранее моделью, новая модель претерпела ряд изменений. В первую очередь, теперь сервер держит в памяти ID подключившихся клиентов, а также записывает их в поле отправителя для всех сообщений, хранимых на сервере. Очевидно, такой подход не позволил использовать для хранения сообщений обыкновенный массив и потребовал создания собственного класса [3] message, содержащего поля ID клиента, ID сообщения и непосредственно текста сообщения. Во-вторых, в связи с необходимостью регистрации клиента, увеличилось количество системных со-

общений между сервером и клиентом. Также была переработана схема пересылки сообщений между хостами.

В общем смысле принцип работы приложения описывается следующим набором сценариев:

После запуска клиент подключается к серверу и регистрируется.

После подтверждения регистрации клиент разрешает пользователю ввод сообщений, а также резервирует место в собственном пуле под известные сообщения.

Сообщения, полученные клиентом от пользователя, хранятся в его собственном буфере до подтверждения получения сервером. Клиент с заданной периодичностью отправляет эти сообщения на сервер.

Получив сообщения и записав их в пул, сервер отправляет клиенту подтверждение получения. Если сообщения уже находятся в пуле, сервер просто отправляет подтверждение приёма.

Периодически клиент опрашивает сервер о размере пула. При наличии отсутствующих у клиента сообщений, он адресно запрашивает их у сервера, после чего добавляет в свой пул.

При разрыве соединения клиент возвращается к своему исходному состоянию.

Как видно из представленных сценариев, в данной модели сервер выполняет пассивную роль, отвечая на сообщения клиентов. Единственный сценарий, когда сервер обращается к клиентам сам – при получении нового сообщения от клиента, когда у сервера обновляется количество известных сообщений, и он рассылает эту информацию всем подключённым клиентам. Ранее уже упоминалось, что информация, пересылаемая между клиентами и сервером, представлена в формате команда + данные. Распишем сценарии по командам (таблицы 1-2).

Таблица 1

#### Команды клиента

Команда	Описание команды	Действие сервера
L=NAME	Запрос на регистрацию клиента с именем NAME	(При отсутствии) добавить клиента в реестр, отправить подтверждение и количество сообщений на сервере.
MID=77:::TEXT	Новое сообщение с ID=77 и текстом TEXT	(При отсутствии) добавить сообщение в пул, отправить подтверждение
S	Запрос количества сообщений в пуле	Отправить клиенту количество сообщений
R=1;2;6	Запрос на получение сообщений с ID 1,2,6	Отправить запрашиваемые сообщения

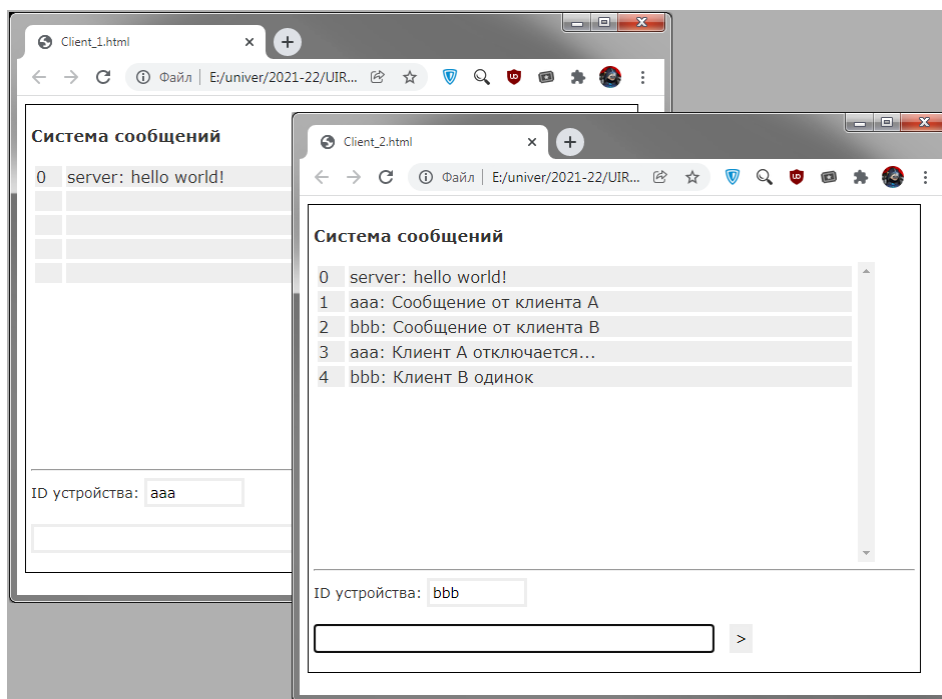
## Команды сервера

Команда	Описание команды	Действие клиента
L	Подтверждение регистрации	Разрешить приём и отправку сообщений
S=120	Количество сообщений в пуле на сервере	Задать длину массива сообщений, при необходимости запросить недостающие
CID=77	Подтверждение получения сообщения с ID=77	Исключить сообщение с ID=77 из буфера отправки
MID=11:::TEXT	Сообщение с ID=11 и текстом TEXT	Добавить сообщение в пул клиента

Следует отметить, что в данном случае ID сообщений, отсылаемых клиентом и сервером, не совпадают, поскольку сервер именуется сообщения по мере поступления от клиентов, а клиент – по мере ввода пользователем. Это не является проблемой, т.к. клиентский ID нужен только во время пересылки от клиента к серверу и для подтверждения приёма, во всех остальных случаях клиент и сервер опираются уже на ID сообщения, сгенерированный сервером. Тем не менее, клиентский ID тоже записывается в пул сообщений, чтобы исключить возможность добавления в пул уже полученных сообщений. Во избежание совпадений для клиентского ID, клиент начинает отсчёт с текущего количества сообщений в пуле на сервере. Таким образом, можно быть уверенным, что клиентский ID для сообщений ни разу не повторится за всё время жизни сервера.

**Тестирование.**

Для тестирования модели было создано клиент-серверное приложение на языке Python3 (серверная часть) и HTML/CSS/JS (клиентская часть). Для имитации работы в сети использовались виртуальные машины с ОС Windows 7, клиенты опирались на заранее известный IP-адрес и порт сервера. Для упрощения модели каждый клиент заранее получил уникальный идентификатор, хотя его получение можно было реализовать и автоматически при первом успешном подключении к серверу. После запуска сервера и клиентов было проведено базовое тестирование: клиенты подключались к серверу и регистрировались, отправляли сообщения, получали подтверждения отправки и т.д. Клиенты, загруженные после отправки сообщений другими клиентами, восстанавливали историю сообщений, предварительно получив данные об их общем количестве, подготовив под них место и адресно запросив их у сервера (рисунок 1).



**Рис. 1.** – Интерфейсы двух клиентов во время тестирования.

После проведения базового тестирования, были реализованы несколько сценариев для проверки надёжности модели (таблица 3).

Таблица 3

### Результаты тестирования надёжности модели

Сценарий	Результат
Отключение клиента от сети до передачи сообщения	Передаваемое сообщение сохранено у клиента и будет передано серверу после восстановления связи
Выключение клиента во время передачи сообщения	Передаваемое сообщение потеряно
Отключение клиента от сети после передачи сообщения, но до подтверждения получения	Сообщение сохранено на сервере, подтверждение приёма придёт при повторной отправке сообщения
Выключение клиента после передачи сообщения, но до подтверждения получения	Сообщение сохранено на сервере, синхронизация с клиентом произойдёт при его запуске и подключении к серверу
Отключение клиента от сети во время приёма сообщения	Данные будут синхронизированы после следующего подключения к серверу
Выключение клиента во время приёма сообщения	Данные будут синхронизированы после следующего подключения к серверу
Отключение сервера от сети во время приёма сообщения	Данные от клиента будут приняты после следующего подключения к серверу
Выключение сервера	Данные потеряны
Отключение сервера от сети во время отправки сообщения	Сообщение будет повторно запрошено клиентом при следующем подключении

При тестировании с избыточной нагрузкой на сеть реализовывался один из упомянутых выше сценариев: не получив ответ на своё действие, клиент просто повторял запрос с заданным интервалом времени до подтверждения результата.

**Выводы.** Исходя из данных таблицы 3, единственный вариант безвозвратной потери клиентом сообщения – это его выключение в период между вводом сообщения и отправкой его на сервер. Единственный вариант безвозвратной потери данных на сервере – его выключение. Теоретически, эта проблема решается, если обеспечить хранение сообщений, как для клиента, так и для сервера на энергонезависимом носителе, однако для упрощения исследуемой модели вопрос безопасного хранения данных не рассматривался. Касаемо обмена данными, сохранность передаваемой информации реализована посредством TCP и WebSocket, а гарантия доставки и синхронизация – посредством использования принципов, описанных ранее. Таким образом, можно утверждать, что используемая концепция, при всей своей простоте, обеспечивает достаточную надёжность пересылки данных.

#### **Библиографический список**

1. Тетюев Е.В., Хабаров С.П. Разработка сетевого приложения с контролем доставки и синхронизацией данных на основе WebSocket. с.196. / Актуальные вопросы в лесном хозяйстве// материалы V международной молодежной научно-практической конференции, 11–12 ноября 2021 г. /под ред. В.С. Павлова – Санкт-Петербург: СПбГЛТУ, 2021. – 236 с., С. 191-196.
2. Хабаров С. П. Построение распределенных систем на базе WebSocket : учебное пособие / С. П. Хабаров, М. Л. Шилкина. — Санкт-Петербург : Лань, 2020. — 216 с. — ISBN 978-5-8114-4739-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/142358> (дата обращения: 22.06.2020). — Режим доступа: для авториз. пользователей. С. 173-177.
3. Python Classes and Objects. — Текст : электронный // W3schools : [сайт]. — URL: [https://www.w3schools.com/python/python\\_classes.asp](https://www.w3schools.com/python/python_classes.asp) (дата обращения: 15.12.2021).



**К.Н. Черник**, магистрант  
Сибирский государственный университет науки и  
технологий имени М.Ф. Решетнева, г. Красноярск  
Kristi.Blueberry@yandex.ru

**Д.В. Черник**, кандидат технических наук, доцент  
Сибирский государственный университет науки и  
технологий имени М.Ф. Решетнева, г. Красноярск  
dionisu2@mail.ru

## ИССЛЕДОВАНИЕ ПАРАМЕТРОВ МАНИПУЛЯТОРА ЛЕСНОЙ МАШИНЫ С ПРИМЕНЕНИЕМ SOLIDWORKS И MATHCAD

*Аннотация.* В статье приведены результаты моделирования механизма подъема стрелы комбинированного манипулятора как механической системы в программе MathCAD, приведены результаты исследований прочностных характеристик конструкций стрелы комбинированного манипулятора в программе SolidWorks.

*Ключевые слова:* манипулятор, лес, кинематика, MathCAD, SolidWorks, прочность.

### Введение

На сегодняшний день в лесной отрасли широкое применение получили машины, оборудованные комбинированным манипулятором, состоящим из шарнирно-сочлененных и телескопических звеньев. Такие машины имеют ряд преимуществ перед другими типами машин начиная от широкого спектра технологических возможностей до обеспечения удобных и безопасных условий труда оператора. По вопросам оптимального проектирования манипуляторов посвящены работы отечественных ученых Александра В.А., Герасимова Ю.Ю., Кошуба С.М. и др. В настоящее время работы по созданию и внедрению манипуляторов высокого технического уровня очень актуальны.

Сейчас наблюдается устойчивая тенденция применения комбинированных манипуляторов, поскольку манипуляторы такого типа наиболее эффективно решают проблему расширения рабочего пространства при ограниченных размерах стрелы, поэтому исследование параметров комбинированных манипуляторов является актуальной темой.

Машины, у которых в качестве технологического оборудования установлены манипуляторы, широко применяются в лесной отрасли. На лесозаготовках такие машины используются на следующих операциях: валка, валка-пакетирование, подбор и трелевка ранее поваленных деревьев, штабелевка сортиментов, подача деревьев к сучкорезным устройствам, погрузка сортиментов, обрезка сучьев, раскряжевка хлыстов и т. д. [1].

## Материалы и методика исследования

Для исследования параметров механизма подъема стрелы манипулятора воспользуемся методикой, приведенной в работе [2].

Угол поворота стрелы ОС отсчитывается от оси ОХ (рисунок 1). Для начального положения  $\varphi_b = \varphi_{b0}$ , для верхнего  $\varphi_b = \varphi_{bmax}$ . Расстояние между осями проушин гидроцилиндра определяется выражением:

$$AB = S_{b0} + Vt, \quad (1)$$

где  $V$  – скорость движения штока гидроцилиндра, м/с, при установившемся движении принимаем  $V = \text{const}$ ;

$S_{b0}$  – расстояние между осями проушин при полностью втянутом штоке, м;

$t$  – текущее значение времени, с.

Остальные геометрические и кинематические характеристики системы определим, используя известные методики [3-5].

Усилие, возникающее на штоке гидроцилиндра стрелы, определим из выражения, Н:

$$F_c = \frac{g(ml + m_c l_c) \cos(\varphi_b + \delta) + (ml^2 + m_c l_c^2) \cdot \varepsilon}{b \sin(\beta_c)} \quad (2)$$

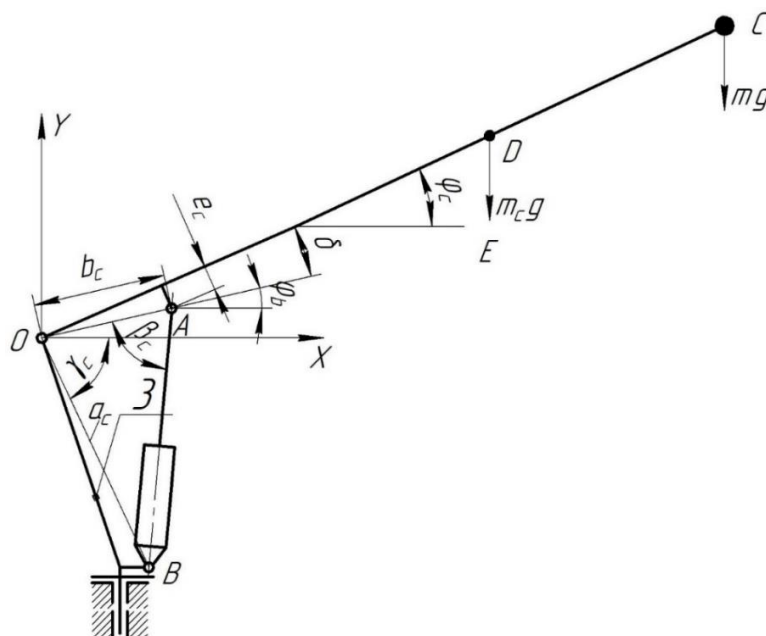


Рис. 1. Кинематическая схема механизма подъема стрелы

Полезный объем гидроцилиндра вычислим из формулы, м<sup>3</sup>

$$v_c = \frac{F_c^{\max}}{P} \cdot S_b, \quad (3)$$

где  $F_c^{\max}$  - максимальное усилие на штоке гидроцилиндра, Н;  
 $P$  – давление рабочей жидкости, Па.

### Результаты исследования

На основе полученных зависимостей была разработана математическая модель механизма подъема стрелы манипулятора в программе MathCAD (рисунок 2).

С помощью математической модели определим оптимальные параметры механизма подъема стрелы манипулятора. За критерий оптимальности примем полезный объем гидроцилиндра, который должен быть минимальным.

Варьируемыми параметрами данной механической системы будут являться  $b_c$ ,  $e_c$ ,  $\gamma_c$  (рисунок 1). Поиск оптимального решения будем производить следующим образом:

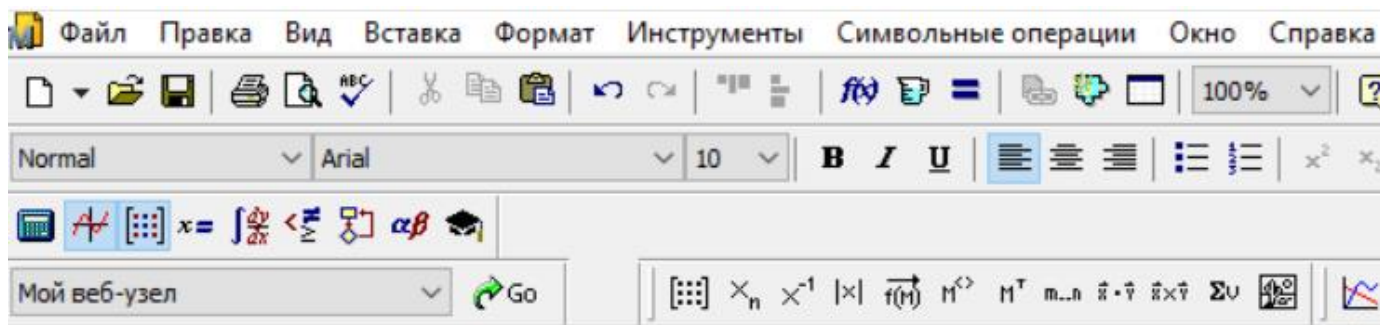
1. Установим значения параметров манипулятора, соответствующие параметрам манипулятора МА-100.

2. Принимая значения  $e_c$ ,  $\gamma_c$  постоянными, меняем значения параметра  $b_c$ , при котором полезный объем гидроцилиндра минимален, то есть  $b_c = b_{c.opt}$ .

3. Принимая значения  $\gamma_c$  постоянным, значение  $b_c = b_{c.opt}$ , меняем значения параметра  $e_c$  при котором полезный объем гидроцилиндра минимален, то есть  $e_c = e_{c.opt}$ .

4. Принимая значения  $b_c$  и  $e_c$  равными оптимальным и меняя угол  $\gamma_c$  определяем его оптимальное значение.

По результатам расчетов получаем графики зависимости объема от варьируемых параметров при разных значениях скорости движения штока гидроцилиндра подъема стрелы манипулятора. На рисунке 3 приведен один из полученных графиков.



**Входные параметры**

$\varphi_0 := -37\text{deg}$      $a := 1.4$      $m := 306 + 1115 = 1421$      $l := 7.530$   
 $\varphi_{\text{max}} := 41\text{deg}$      $b := 0.65$      $m_c := 375 + 292 = 667$      $l_c := 0.4 \cdot l = 3.012$   
 $\gamma := 80\text{deg}$      $e := 0.15$      $P := 20 \cdot 10^6$   
 $V := 0.15$

**Выходные параметры**

$\delta := \arcsin\left(\frac{e}{b}\right) = 0.233$      $\delta \cdot \frac{180}{\pi} = 13.342$      $\varphi := \begin{pmatrix} -37\text{deg} \\ -29.2\text{deg} \\ -21.4\text{deg} \\ -13.6\text{deg} \\ -5.8\text{deg} \\ 2\text{deg} \\ 9.8\text{deg} \\ 17.6\text{deg} \\ 25.4\text{deg} \\ 33.2\text{deg} \\ 41\text{deg} \end{pmatrix}$

$S_0 := \sqrt{a^2 + b^2 - 2a \cdot b \cdot \cos(\gamma + \varphi_0)} = 1.025$   
 $S_{\text{max}} := \sqrt{a^2 + b^2 - 2a \cdot b \cdot \cos(\gamma + \varphi_{\text{max}})} = 1.822$   
 $S := S_{\text{max}} - S_0 = 0.797$

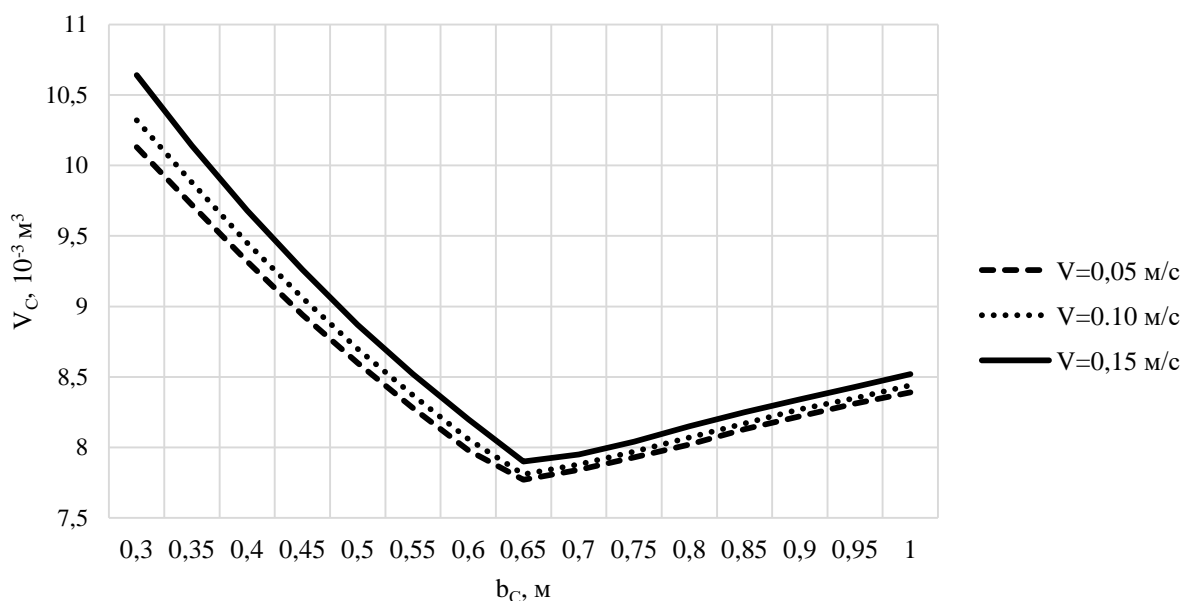
$t := \frac{-S_0 + \sqrt{S_0^2 - 2 \cdot a \cdot b \cdot (\cos(\gamma + \varphi) - \cos(\gamma + \varphi_0))}}{V}$   
 $\omega := \frac{(S_0 - V \cdot t) \cdot V}{a \cdot b \cdot \sin(\gamma + \varphi)}$   
 $\varepsilon := \frac{V^2 \cdot \sin(\gamma + \varphi) - V \cdot (S_0 + V \cdot t) \cdot \cos(\gamma + \varphi) \cdot \omega}{a \cdot b \cdot \sin(\gamma + \varphi)^2}$   
 $\beta := \arccos\left[\frac{b^2 + (S_0 + V \cdot t)^2 - a^2}{2 \cdot b \cdot (S_0 + V \cdot t)}\right]$   
 $F_c := \frac{9.81 \cdot (m \cdot l + m_c \cdot l_c) \cdot \cos(\varphi + \delta) + (m_c \cdot l_c^2 + m \cdot l^2) \cdot \varepsilon}{b \cdot \sin(\beta)}$

$F_{cst} := \frac{9.81 \cdot (m \cdot l + m_c \cdot l_c) \cdot \cos(\varphi + \delta)}{b \cdot \sin(\beta)}$   
 $\max(F_{cst}) = 194688$   
 $\max(F_c) = 197836$   
 $kd := \frac{\max(F_c)}{\max(F_{cst})} = 1.016$

**Полезный объем гидроцилиндра**

$v := \frac{\max(F_c)}{P} \cdot S = 0.00788$

Рис. 2. Математическая модель механизма подъема стрелы

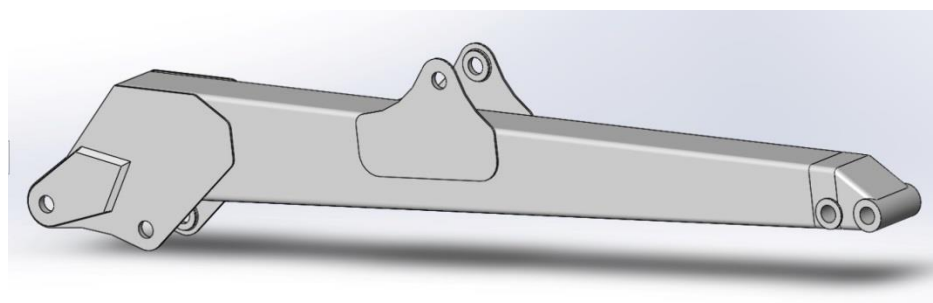


**Рис. 3.** Зависимости полезного объема гидроцилиндра подъема стрелы от параметра  $b_c$

Оптимальное значение параметра  $b_c$  манипулятора МА-100 составляет 0,65 м, в отличие от изначального значения, которое равняется 0,5 м. Полезный объем гидроцилиндра при этом уменьшился с  $8,87 \cdot 10^{-3} \text{ м}^3$  до  $7,9 \cdot 10^{-3} \text{ м}^3$ , то есть в 1,12 раз.

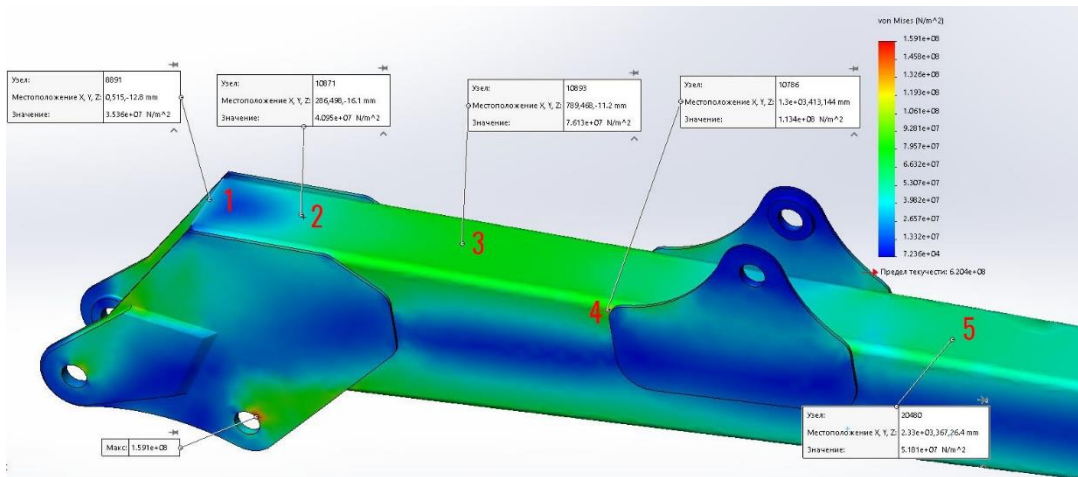
Прочностные характеристики конструкции стрелы манипулятора исследуем с помощью программы SolidWorks, сравнив два принципиально разных подхода к проектированию [6].

При стандартном подходе к проектированию конструкции, крепления для гидроцилиндров, выполняющих функцию подъема стрелы и функцию поворота рукояти, разрабатывают отдельными проушинами. Такой подход не всегда оправдан с точки зрения прочностных характеристик конструкций. Например, проанализируем возникающие напряжения в стреле на рисунке 4. Установим в качестве материала низколегированную сталь 09Г2С, наложим соответствующие ограничения в местах крепления стрелы к колонне и гидроцилиндру подъема стрелы, зададим внешние нагрузки, создадим сетку и запустим исследование.



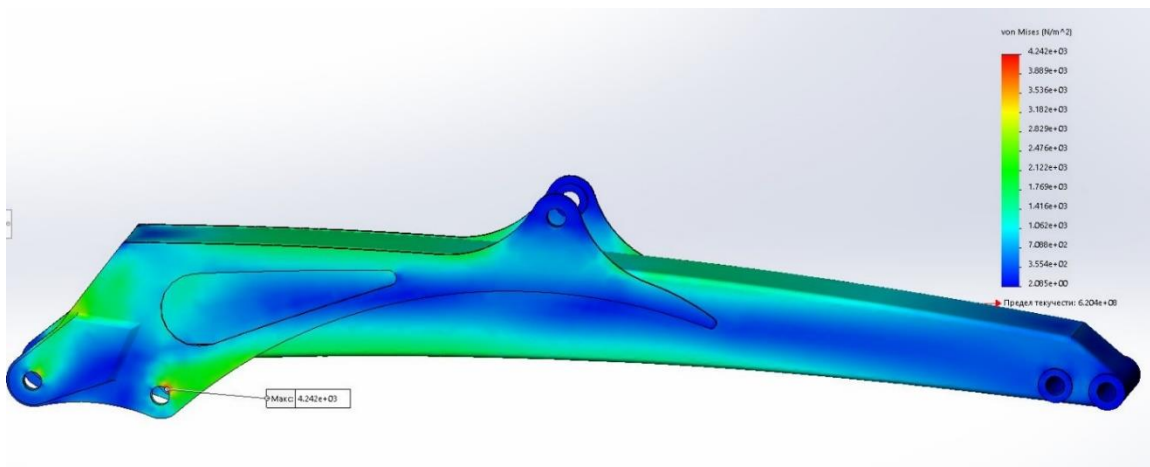
**Рис. 4.** Стандартная конструкция стрелы манипулятора в SolidWorks

Выполнив расчет, получаем цветную эпюру напряжений, на которой проводим зондирование в ключевых точках (рисунок 5).



**Рис. 5.** Зондирование напряжений в ключевых точках конструкции

На эпюре, вдоль стрелы, наблюдаются неравномерные, резкие, перепады напряжений, что может пагубно отразиться на надежности и долговечности конструкции. Для обеспечения более плавной границы напряжений предлагается связать проушины гидроцилиндра подъема стрелы и поворота рукояти в одну конструкцию и провести аналогичное исследование (рисунок 6).



**Рис. 6.** Эпюра напряжений стрелы после модернизации

Полученные результаты сведем в таблицу 1. По полученным данным построим график (рисунок 7).

Таблица 1

**Результаты исследований конструкций стрелы в SolidWorks**

Конструкция стрелы	Напряжения в точках, МПа						Масса, кг
	1	2	3	4	5	Max	
Стандартная	37,57	43,32	80,09	118,6	53,41	165,5	560,73
Модернизированная	2,64	1,70	0,14	1,77	3,23	59,4	57,75



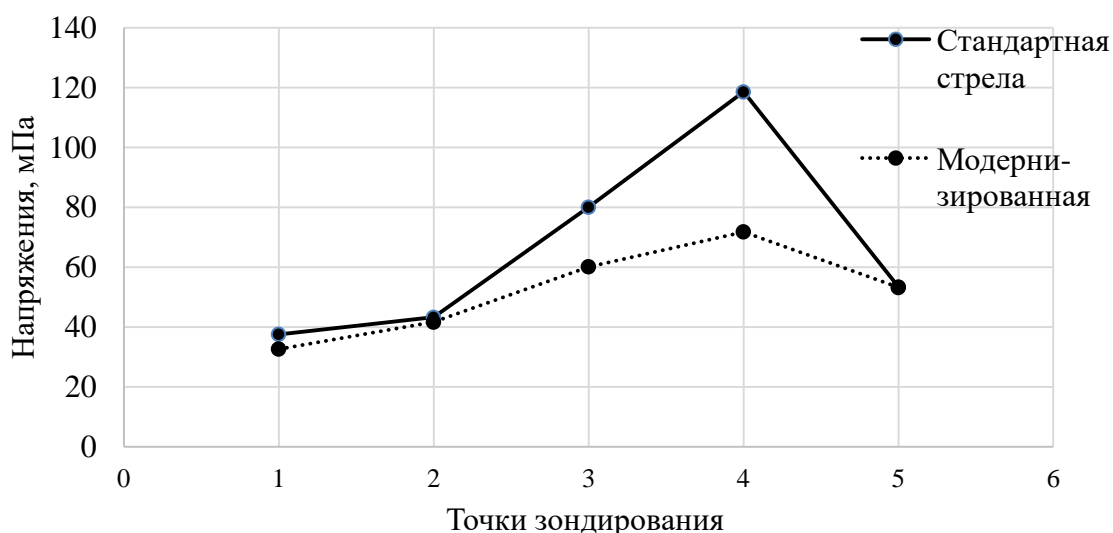


Рис. 7. График напряжений в точках зондирования

## Выводы

Согласно графику, кривая напряжений стрелы, у которой оптимизирована геометрия проушин, имеет более плавный характер чем кривая первоначальной конструкции стрелы. Это обеспечивается более мягким переходом геометрии общей конструкции проушин и позволяет снизить резкие перепады напряжений вдоль всей конструкции стрелы. Так, напряжения в точке 3 уменьшились в 1,33 раза, а в точке 4 - в 1,65 раз при одинаковых материале, ограничениях и внешних нагрузок. При этом масса конструкции у нас практически не изменилась.

В настоящее время современные системы автоматизированного проектирования, в том числе пакеты математических программ, позволяют значительно облегчить труд инженера-исследователя, автоматизируя рутинные задачи инженерной и научной практики, которые порой по трудоемкости занимают большую часть работы. Поэтому применение подобных систем открывает большие возможности инженеру-исследователю для творческой деятельности, в том числе при совершенствовании конструкций манипуляторов лесных машин.

## Библиографический список

1. Полетайкин, В. Ф. Комбинированные манипуляторы лесосечных и лесотранспортных машин. Динамика элементов конструкции: монография / В. Ф. Полетайкин, П. Г. Колесников. – Красноярск: СибГТУ, 2014. – 167 с.
2. Емтыль, З.К. Совершенствование кинематики, динамики и конструкции лесопромышленных гидроманипуляторов: автореферат дис. доктора технических наук : 05.21.01 / Воронеж. гос. лесотехн. акад. - Воронеж, 2002. - 36 с.
3. Артоболевский, Иван Иванович. Теория механизмов и машин [Текст]: учебник для студентов высших технических учебных заведений / И. И.

Артоболевский. - Изд. 6-е, стер., перепеч. с 4-го изд. 1988 г. - Москва: Альянс, 2011. - 639 с.

4. Добрачев А.А., Раевская Л.Т., Швец А.В. Кинематические схемы, структуры и расчет параметров лесопромышленных манипуляторных машин: монография. – Екатеринбург: Урал. гос. лесотехн. ун-т, 2014. – 128 с.
5. Герасимов Ю.Ю. Манипуляторные системы лесных машин: Проектирование и расчет: учебное пособие/ Ю.Ю. Герасимов, С.А. Кильпеляйнен, В.М. Костюкевич, В.С. Сюнев. – Петрозаводск: Петрозаводск. гос. ун-т., 1994. – 95 с.
6. Черник, К. Н. Анализ прочностных характеристик стрелы комбинированного манипулятора в среде solidworks / К. Н. Черник, Д. В. Черник // Машиностроение: новые концепции и технологии: Всероссийская научно-практическая конференция студентов, аспирантов и молодых ученых, Красноярск, 23 октября 2020 года. – Красноярск: Федеральное государственное бюджетное образовательное учреждение высшего образования "Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева", 2020. – С. 200-203.

**М.Р.Вагизов**, кандидат технических наук, доцент  
Зав. кафедрой информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
bars-tatarin@yandex.ru

**А.М.Заяц**, кандидат технических наук, профессор.  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
zamfta@yandex.ru

## **СОВРЕМЕННЫЕ ПРОБЛЕМЫ ЦИФРОВИЗАЦИИ ЛЕСНОГО ХОЗЯЙСТВА**

Потенциал ведения лесного хозяйства в Российской Федерации на сегодняшний день не раскрыт в том технологическом объёме, в котором лесная отрасль способна не только принести прибыль в бюджет страны и обеспечить рабочими местами специалистов лесного сектора, но и привлечь в отрасль дополнительно работников.

Важнейшим фактором, обеспечивающим эффективность управленческих процессов, является интеграция в отрасль современных цифровых технологий, использующих отечественное программное обеспечение и обновленную аппаратную базу основных компонентов, применяемых как при реализации специфических рабочих процессов в лесном хозяйстве, так и при обработке и передачи информации.

Существенным сдерживающим фактором здесь является отсутствие актуальной, достоверной и доступной информации о лесах, обеспечивающей ведение устойчивого и эффективного с точки зрения экономических, экологических и социальных аспектов лесного хозяйства.

Созданная автоматизированная информационная система «Государственный лесной реестр» (АИС ГЛР) по своей сути является реляционной базой данных, хранящей табличную информацию, в которой зачастую данные, публикуемые на сайтах государственных органов управления лесами, расходятся, а в последнее время сама система неработоспособна.

Функционал системы не обеспечивает выполнение процедур управления лесом и серьезной аналитики, необходимой для оценки состояния лесов, разработки планов их развития и управления лесным хозяйством.

Такое положение обусловлено тем, что при создании системы не был проведен серьезный анализ структуры и динамики такой предметной области как лесное хозяйство, не были выявлены ее текущие проблемы и системные требования, необходимые для поддержки автоматизации предметной области, не были определены потенциальные возможности совершенствования лесного хозяйства и функциональные границы создаваемой системы, а также при разработке графических интерфейсов потенциальных пользователей не были учтены их потребности, цели и квалификация.

В ряду проблем, влияющих на развитие лесного хозяйства, отвечающего современным требованиям, можно выделить: инфокоммуникацион-

ное и техническое обеспечение для цифровизации лесоустройства; подготовку специалистов и кадровое обеспечение; совершенствование законодательной базы.

В принятом распоряжении Правительства РФ от 11.02.2021 N 312-р «Об утверждении Стратегии развития лесного комплекса Российской Федерации до 2030 года» в пункте 8 отмечено: «...цифровой трансформацией лесной отрасли является переход от обмена бумажными документами к обмену данными, введение реестровых моделей, отказ от дублирующей и излишней информации».

Изменение подходов в развитии лесного хозяйства с применением цифровых технологий признано большинством экспертов и специалистов лесного хозяйства. При этом следует отметить, что цифровизация лесного хозяйства предполагает не только автоматизацию документооборота, но и активное применение современных информационных технологий для создания и использования уникальных бизнес-моделей, приводящих к эффективному изменению, ускорению и упрощению деловых процессов в лесной отрасли, ускорению проведения операций, устранения ненужных действий и упорядочению нужных, а также оптимизации использования ресурсов.

В лесном хозяйстве внедрение цифровых технологий сдерживается, прежде всего, применением вертикального (централизованного) проведения всех процедур лесоустройства. Необходимо социализировать систему лесоустройства, активнее использовать горизонтальное лесоустройство, допуская частные компании к ее проведению и снимая монополию профильного ведомства, зачастую предоставляющего некачественную лесоустроительную информацию и не активизирующего интерес и заинтересованность арендаторов, особенно долгосрочных, к применению лесных цифровых технологий на своей территории.

Перманентный процесс смены законодательной базы в лесном хозяйстве снижает его активность и не стимулирует арендаторов инвестировать в цифровизацию, так как в относительно короткий арендный период требуется большой объем необходимых затрат на внедрение таких технологий, а положительных примеров их использования в отрасли очень мало. Рынок информационных технологий и консалтинговых услуг пока не повернулся к лесной отрасли в развитии ее цифровизации, и не проявляет должной заинтересованности [1].

Интеграция государственного и частного секторов в лесоустройстве позволит активнее внедрять цифровые технологии при проведении лесоустроительных мероприятий и разработке нормативной базы, стимулировать активное применение передовых, для лесного бизнеса, информационных технологий, таких как: фотосъемка лесов беспилотными летательными аппаратами, лазерное сканирование лесных территорий, дистанционное зондирование лесов.

Так лазерное сканирование позволяет получить не только лесотехническую, но и дополнительную прикладную информацию для планирования

эффективного лесопользования: данные по рельефу местности ландшафту территории для проектирования и строительства дорог, данные для мониторинга и совершенствования гидромелиоративной сети, информацию по возможному распространению лесных пожаров на основе цифровой модели лесных территорий с определением рисков перехода низового пожара в верховой и распространения пожара на населенные пункты.

Социализация проведения лесоустройства, в рамках которой формируется приоритет развития работника и условий его производственной деятельности, приведет к открытости данных, их доступности и повышению качества информации о лесах. Оценивая текущую ситуацию, можно отметить, что современный арендатор (за исключением некоторых крупных компаний, которые в последнее время стали активно внедрять цифровые технологии) обычно работает с довольно разрозненной информацией о лесном фонде преимущественно в бумажном или растровом формате, а также с неструктурированными данными, представленными в разных системах. Планирование лесного хозяйства зачастую осуществляется на основе субъективных знаний местных специалистов, с минимальным привлечением объективных данных.

Цифровизация предполагает кардинальные изменения и в техническом обеспечении подразделений лесного хозяйства, и в создании новой инфраструктуры, обеспечивающей формирование и развитие информационного пространства лесной отрасли и средств информационного взаимодействия. Здесь важнейшим условием является так называемая цифровая зрелость. Она отражает готовность к digital-преобразованиям, наличие стратегии, определение тактики, уровень компетенции команды и другие аспекты. Не стоит, однако, идти по пути точечного внедрения модных технологий для решения отдельных задач — так называемой лоскутной цифровизации. Цифровая трансформация предполагает не использование локальных систем, а создание цифровой экосистемы, полностью меняющей бизнес-модель отрасли. Важно, чтобы лесничества имели доступ к современным системам автоматизации бизнес-процессов, располагали оборудованием необходимой мощности для эффективного хранения и обработки огромного количества данных.

Здесь важен стандартный централизованный подход в оснащении цифровой техникой всех уровней отрасли с учетом состава и номенклатуры функциональных задач, решаемых на каждом из них. В таком обеспечении возможно использование отечественных программных продуктов. Так, среди последних разработанных операционных систем положительные отзывы от пользователей получила ОС «Аврора» со встроенными средствами защиты каналов передачи данных, не имеющая серьезных системных требований к производительности процессорной платформы ПК.

Для обеспечения должной функциональности на полевых выездах, необходима координация работ и оперативное взаимодействие с центрами управления. Основной проблемой в этой связи является отсутствие оборудо-

дования, обеспечивающего устойчивое информационное взаимодействие в труднодоступных местах лесных территорий.

Классическим примером несовершенства технического обеспечения является необходимость передачи лесным инспектором различных правонарушений после их фиксации. В виду невозможности сделать это оперативно посредством, например, передачи информации через спутниковую связь, инспектору приходится осуществлять передачу данных на бумажных носителях, что замедляет процесс работы и принятия последующих решений.

Решением обозначенной проблемы является обеспечение всех сотрудников, участвующих в полевых работах, средствами спутниковой связи, к которым могут быть отнесены отечественные решения компании SatelliteTelecome, [2] где пользователям предоставляются различные услуги спутниковой связи: мобильная спутниковая связь (голосовая и факсимильная), мобильный Интернет, мобильная видеоконференцсвязь.

Существенным сдерживающим фактором в рассмотренном направлении является стоимость оборудования и тарифы трафика. Возможным решением этой проблемы является использование существующих механизмов частно-государственного партнёрства и лизинга, предполагающие как индивидуальные подходы, так и нахождение способов сотрудничества между государственными компаниями и коммерческим сектором, при котором каждый из участников данных отношений получает определённые выгоды.

Решить накопившиеся в лесном хозяйстве проблемы только за счет технического переоснащения удастся лишь частично. Важно организовать процесс включения сотрудников в цифровую трансформацию, донести до них необходимость изменений и освоения новых инструментов и технологий, обучить их нужным навыкам. Здесь важным является целевая подготовка кадров для лесного хозяйства. Задачей подготовки будущего digital-специалиста в лесной отрасли является формирование у него компетенций активного использования в своей работе цифровых инструментов и методов, современных информационных технологий. Негативным фактором, сдерживающим развитие цифровой трансформации в лесном хозяйстве, является низкий уровень заработных плат сотрудников, что мешает привлечению в отрасль высококачественных профильных IT и ГИС-специалистов.

Для подготовки в отрасли комплексных специалистов возможно два подхода. Первый подход заключается в целевой подготовке будущих специалистов лесного хозяйства на выпускающих IT кафедрах лесных вузов, где студенты третьих и четвертых курсов проходят производственные, технологические и другие виды практик на предприятиях лесного комплекса. Второй подход заключается в направлении организацией своих сотрудников как на получение профильного образования в случае его отсутствия, так и на повышение и переподготовку кадров разного уровня квалификации. Важно отметить, что подход, заключающийся в формировании



целевого кадрового обеспечения для организаций IT-профиля, при их тесном сотрудничестве с выпускающими кафедрами, позволяет достичь главной взаимовыгодной цели. Для кафедры такой целью является поддержка профессиональных компетенций в образовательной деятельности и сохранение учебных планов в актуальном состоянии, а для организации – получение на выходе готового специалиста с профориентацией в конкретной области информационных технологий.

Здесь важно поддерживать современный уровень цифровой трансформации в вузе, применяя передовые технологии и модели. Модели цифровой трансформации вуза должны обеспечивать получение необходимой интерактивной актуальной информации и знаний преподавателями и студентами на своих устройствах, как на занятиях, так и вне, что помогает им более эффективно взаимодействовать в образовательном процессе[3].

В части сокращения большого количества не актуальной и дублирующей информации необходима легализация данных о лесах, полученных с помощью цифровых технологий в новых цифровых форматах, ведь такие понятия как «цифровая аэрофотосъемка» и «лазерное сканирование» пока официально нигде не прописаны и не закреплены [4].

Новые подходы должны лечь в основу разработки стандарта лесохозяйственной информации, который обеспечит унификацию и сопоставимость разных массивов данных о лесах. При изменении и совершенствовании подходов к лесоустройству существует насущная необходимость пересмотра текущей лесоустроительной инструкции. Многие показатели, применяемые в лесоустройстве согласно инструкции, уже устарели или потеряли актуальность.

Внедрение цифровых технологий позволит обеспечить как внутренний, так и независимый контроль процессов лесопользования и сократить количество ошибок, приводящих к неэффективности экономики лесного хозяйства.

Так, лесохозяйственные мероприятия вовремя не фиксируются, и на момент появления данной информации в открытом доступе она уже не актуальна, хотя именно здесь проявляется наиболее общественно значимая и чувствительная информация о местах планируемых рубок и проведения других лесохозяйственных мероприятий. Даже информация о границах арендованных лесных участков по большей части недоступна в пригодном для использования формате.

Для устранения этого недостатка целесообразно внести в законодательную базу соответствующие изменения, обязывающие владельцев данных, полученных за государственный счет, открывать их, а также данные, поддерживающие создание и функционирование единого информационного пространства о лесах.

Развитие современных цифровых технологий получения и обработки данных и широкое внедрение цифровизации в планировании, заготовке, логистике, учете и мониторинге способствуют переходу к высокоточному, экономически эффективному и устойчивому лесному хозяйству.

### Библиографический список

1. Цифровая трансформация бизнеса: необходимая реальность. Оригинал статьи: <https://www.kp.ru/guide/tsifrovaja-transformatsija.html>
2. Материалы семинара Цифровые геотехнологии
3. Заяц А. М., Вагизов М.Р. Цифровая трансформация - вектор в высшее образование будущего // Цифровые технологии в лесном секторе: материалы Всероссийской научно-технической конференции, Санкт-Петербург, 26–27 марта 2020 года / Санкт-Петербургский государственный лесотехнический университет имени С. М. Кирова; Санкт-Петербургский политехнический университет Петра Великого. – Санкт-Петербург: Политех-Пресс, 2020. – С. 57-60.
4. А. Костенко. Цифровизация лесного сектора России ЛесПромИнформ №6 (160), 2021 г. <https://lesprominform.ru/jarticles.html?id=6028>

**Р.А. Анохин**, магистрант 1 курса ИСиТ  
anokhin.roman.27@gmail.com

**Б.М. Каримов**, магистрант 1 курса ИСиТ  
kbm99@mail.ru

**С.А. Кныш**, магистрант 1 курса ИСиТ  
monarch450@gmail.com

**К.А. Анохин**, студент 2курса  
Velorychblu666@gmail.com

Кафедра информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова

**В.Я. Шапиро**, доктор технических наук, профессор  
Кафедра высшей математики  
СПбГЛТУ им. С.М.Кирова  
shapiro54vlad@mail.ru

### АВТОМАТИЗАЦИЯ РАСЧЕТОВ ТЕХНОЛОГИЧЕСКИХ ПАРАМЕТРОВ ГРУППОВОЙ ОКОРКИ ЛЕСНЫХ БАЛАНСОВ

***Аннотация:** Целью работы было оптимизировать работу оператора окорочных барабанов, путём автоматизации расчётов технологических параметров. Было разработано приложение для операционной системы ANDROID, которое рассчитывает время окорки.*

***Ключевые слова:** Автоматизация, расчёт, технологические параметры, окорка, лесной баланс, групповая окорка, технологии.*

#### Введение

При групповой обработке лесных балансов в окорочных барабанах качество работ ниже, чем при штучной окорке, однако производительность труда значительно выше. Основное преимущество групповой окорки по

сравнению со штучной заключается в возможности обрабатывать мелкие и искривленные лесоматериалы, в том числе с поперечным сечением неправильной формы. В этих условиях штучная окорка, как правило, затруднена или вообще невозможна.

В установках для групповой обработки (окорочных барабанах) процесс отделения коры производится вследствие соударения и трения бревен между собой. В наиболее трудных условиях окорки мерзлых балансов на внутренней поверхности барабана располагают специальные ножи для дополнительного разрушения коры.

### **Математическое моделирование**

В основе математического моделирования технологических параметров групповой окорки лежит связь между качеством окорки и временем (Т) нахождения лесных балансов во вращающемся окорочном барабане. Параметр Т – основной выходной технологический параметр групповой окорки.

Количество ударов (N) по участку коры и время (Т<sub>1</sub>) обработки одного бревна являются основными регулируемыми промежуточными технологическими параметрами процесса окорки.

Математические модели разрушения массива коры в рамках механики контактного разрушения подробно рассмотрены в работах [1-5].

Основной механизм разрушения коры в зоне контакта – развитие сдвиговых напряжений от ударных нагрузок, приводящих к отделению коры от массива древесины.

Под действием силы удара за период времени t в массиве коры формируется тензор напряжений, одна из компонент которого  $\sigma_c$  – сдвиговое напряжение - при превышении предела прочности коры на скол  $\tau$  обеспечивает надежное отделение коры от поверхности бревна.

Механизм сдвиговых деформаций при многократных силовых нагрузках в массивах сплошных сред рассмотрен в работе [6].

Количество ударов по одному бревну определяется соотношением:

$$N = 360 / \left[ \arccos \left( 1 - \frac{K_d^2}{2(1-K_d)^2} \right) K_l \right] \quad (1)$$

где коэффициент  $K_d$  - отношение диаметра бревна d к диаметру барабана D; коэффициент  $K_l$  - отношение длины бревна l к длине барабана L [4].

Время t обработки одного бревна, в течение которого разрушение коры происходит на полную глубину h, определяется в [4], исходя из законов сохранения энергии и импульса движения со скоростью соударения v.

Так при окорке балансов летней сосны при положительной температуре  $T_b \geq 0^\circ\text{C}$  окружающей среды с влажностью коры W от 40 до 60% и параметрами окорки:

$$D=2,2\text{м}, d=0,12\text{м}, L=4,3\text{м}, l=2\text{м}, h=0,02, \tau=0,34\text{МПа} \quad (2)$$

расчет по формуле (1) позволят установить значение  $N=333$ , при этом получены следующие данные о времени  $t=9,62$ мс и скорости  $v=6,39$ м/с соударения участков бревен.

Таким образом, необходимое время обработки одного бревна составляет:

$$T_1 = N \cdot t = 3,75 \text{ с.} \quad (3)$$

В том случае, если окорка этих же балансов сосны осуществляется зимой при температуре  $T_b$  ниже  $0^\circ\text{C}$  значения параметра  $T_1$  умножают на поправочный температурный коэффициент  $K_T$  в соответствии с результатами исследований [6-9] о влиянии температуры  $T_b$  на разрушение массивов сплошной среды различной природы при многократных (циклических) динамических нагрузках:

$$K_T = -0,1916T_b + 1,17. \quad (4)$$

В конечном итоге, получен следующий основной технологический параметр окорки - время  $T$  групповой обработки всех загруженных в окорочный барабан лесных балансов, равное для данного примера расчета  $T=9$  минут.

При этом объем полученной коры  $V_k = 6,5 \text{ м}^3$ .

Для сравнения: при обработке таких же балансов мерзлой сосны ( $T_b = -10^\circ\text{C}$ ) при влажности  $W=30\%$  общее время окорки возрастает до  $T=40$  минут, т.е. почти в 4,5 раза.

### **Разработка мобильного приложения**

Автоматизация расчетов технологических параметров групповой окорки приводит к тому, что оператор данного процесса избавляется от необходимости лично анализировать все условия и факторы, которые необходимы при расчетах, и может воспользоваться разработанным приложением.

С помощью простого интерфейса предоставляется возможность получать и отслеживать все необходимые промежуточные и конечные технологические параметры групповой окорки лесных балансов непосредственно с экрана своего портативного устройства.

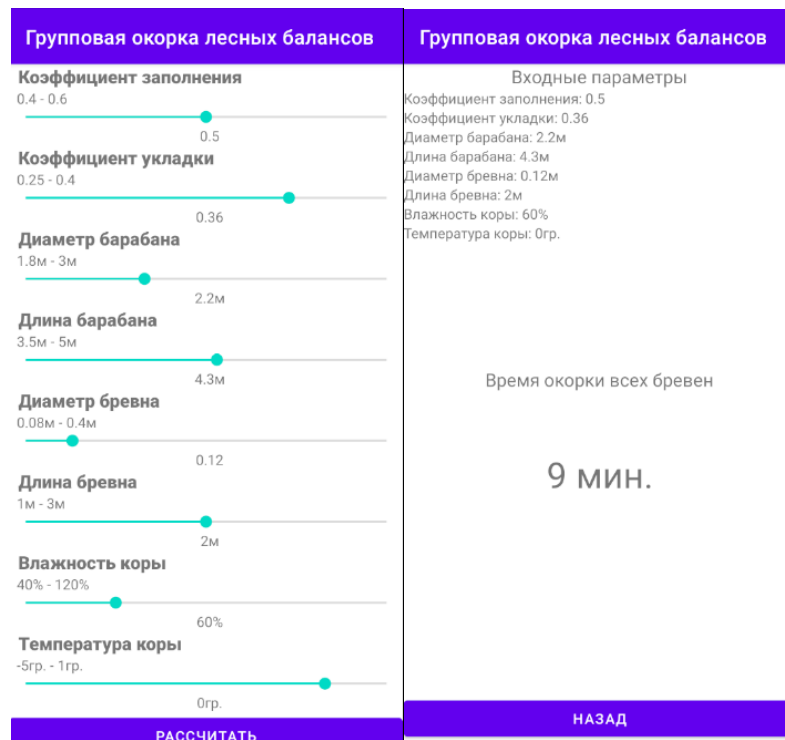


Рис.1. Интерфейс приложения и результат вычисления

Интерфейс создаваемого приложения выполнен в максимально упрощенном виде, для того чтобы оператор мог разобраться с его функциональными возможностями.

Программное обеспечение напрямую связано с формулами (1) - (4), которые лежат в основе расчетов, а предоставляемый пользователю интерфейс позволяет без особого труда выбрать из предлагаемых параметров ввода необходимые данные, что упрощает работу по сравнению с ручным методом.

	<b>Тв</b>			
<b>Кзап</b>	<b>Kd</b>	<b>Vк, м3</b>	<b>Vк, м3</b>	<b>T, мин</b>
<b>Кукл</b>	<b>Kl</b>	<b>Nбр</b>	<b>h, м</b>	
<b>D</b>	<b>Vбар, м3</b>	<b>N</b>	<b>τ, МПа</b>	
<b>d</b>	<b>Vбр, м3</b>	<b>T1, с</b>		
	<b>L</b>	<b>I</b>	<b>W</b>	
	<b>ИСХОДНЫЕ ДАННЫЕ</b>			
	<b>РАСЧЕТНЫЕ ДАННЫЕ</b>			
	<b>ТЕХНОЛОГИЧЕСКИЙ ПАРАМЕТР</b>			

Рис. 2. Принцип работы приложения

После ввода необходимых данных с помощью интерфейса приложения на основе разработанного функционала оператор на выходе получает проектные технологические параметры окорки.

Таким образом, с помощью данного приложения значительно упрощается задача оперативной оценки технологических параметров групповой окорки.

При этом оператору предоставляется возможность учесть геометрические параметры окорочного барабана и загружаемых бревен, их влажность и прочностные характеристики, а также температуру окружающей среды, что повышает надежность расчетов.

### **Библиографический список**

1. Газизов А.М., Григорьев И.В., Кацадзе В.А., Шапиро В.Я., Мурашкин. Повышение эффективности механической окорки лесоматериалов.- СПб: СПбГЛТА.-2009.-239 с.
2. Газизов А.М., Шапиро В.Я., Григорьев И.В. Влияние влажности на развитие процесса разрушения коры при роторной окорке// Вестник Московского государственного университета леса -Лесной вестник.-2008.-№6.-С.129-133.
3. Григорьев И.В., Шапиро В.Я., Гулько А.Е. Математическая модель групповой окорки лесоматериалов в окорочных барабанах// Научное обозрение.-2012.-№4.С.154-171.
4. Шапиро В.Я., Григорьев И.В., Гулько А.Е. Анализ методов расчета параметров и обоснование математической модели разрушения коры при групповой окорке древесины//Ученые записки Петрозаводского государственного университета.-2011.-№8(121).-С.92-96.
5. Газизов А.М., Шапиро В.Я., Григорьев И.В. Вариационный метод расчета и стабилизация параметров роторной окорки// Справочник. Инженерный журнал с приложением.-2009.-№7(148).-С.47-51.
6. Шапиро В.Я., Григорьев, Жукова А.И. Влияние сдвиговых деформаций на процесс циклического уплотнения почвы// Естественные и технические науки.-2006.-№1.-С.174-180.
7. Рудов С.Е., Шапиро В.Я., Григорьев И.В., Куницкая О.А., Григорьева О.И. Особенности взаимодействия трелевочной системы с оттаивающим почвогрунтом. Лесной вестник. ForestryBulletin.-2019.- Т.23.-№1.- С.52-61.
8. Шапиро В.Я., Григорьев И.В., Лепилин Д.В., Жукова А.И. Моделирование уплотнения почвогрунта в боковых полосах трелевочного волока с учетом изменчивости трассы движения //Ученые записки Петрозаводского государственного университета.-2010.-№ 6 (111).-С.61-64.
9. Шапиро В.Я., Григорьев И.В., Жукова А.И., Иванов В.А. Исследование механических процессов циклического уплотнения почвогрунта при динамических нагрузках. - Вестник КрасГАУ.-2008.-№1.-С.163-175.



**Р.А. Анохин**, магистрант 1 курса ИСиТ  
anokhin.roman.27@gmail.com

**Б.М. Каримов**, магистрант 1 курса ИСиТ  
kbm99@mail.ru

**С.А. Кныш**, магистрант 1 курса ИСиТ  
monarch450@gmail.com

**К.А. Анохин**, студент 2 курса  
Velorychblu666@gmail.com

Кафедра информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова

**В.Я. Шапиро**, доктор технических наук, профессор  
Кафедра высшей математики  
СПбГЛТУ им. С.М.Кирова  
shapiro54vlad@mail.ru

## **АВТОМАТИЗАЦИЯ РАСЧЕТОВ ТЕХНОЛОГИЧЕСКИХ ПАРАМЕТРОВ РОТОРНОЙ ОКОРКИ ЛЕСНЫХ БАЛАНСОВ**

***Аннотация:** Целью работы было оптимизировать работу оператора роторного окорочного станка, путём автоматизации расчётов технологических параметров. Было разработано приложение для операционной системы ANDROID, которое рассчитывает удельную силу прижима фрезы к лесному балансу.*

***Ключевые слова:** Окорка лесных балансов, автоматизация, расчёты, технологические параметры, роторный окорочный станок.*

Основное преимущество групповой окорки по сравнению со штучной заключается в возможности обрабатывать мелкие и искривленные лесоматериалы, в том числе с поперечным сечением неправильной формы. В этих условиях штучная окорка, как правило, затруднена или вообще невозможна.

Однако качество штучной окорки значительно выше, чем при групповой, в связи с чем, при высоких требованиях заготовки лесоматериалов предпочтение отдают роторно-скребковой технологии с использованием различных скребков (фрез) - режущего инструмента.

В основу этого метода окорки положены различия в прочности коры и древесины, что обуславливает снижение сцепления между ними. Для получения требуемого качества окорки скребкам придают соответствующую форму и устанавливают необходимую и достаточную силу их прижима  $F$  к поверхности бревна.

Величина силы прижима  $F$  в зависимости от условий окорки должна составлять в среднем от 15 до 25 кН на 1 м ширины скребка, поэтому измеряют силу  $F$  в кН/м и обозначают как удельную силу окорки.

Для летних и чрезвычайно влажных образцов, например, сосны (общая влажность  $W$  порядка 100% и более) величина  $F$  снижается до 10 и менее кН/м, тогда как окорку мерзлых сухих образцов ( $W$  менее 50%) осуществляют с удельной силой окорки 30 и более кН/м.

Уместно отметить, что при окорке мерзлых образцов осины или березы величина удельной силы окорки может достичь уровня  $F > 45-50$  кН/м.

Таким образом, параметр  $F$  выступает интегральной технологической характеристикой, обеспечивающей качественную обработку баланса на роторном станке.

Математические модели различных технологий окорки разработаны в работах [1-5].

Основной механизм разрушения коры в зоне контакта с режущим инструментом при роторной окорке – это развитие сдвиговых напряжений от статической сжимающей нагрузки, приводящей к отделению коры (сколу) от массива древесины.

Под действием силы  $F$  в массиве коры формируется тензор напряжений, максимальная сдвиговая компонента которого  $\sigma_c$  сравнивается с величиной  $\tau$  предела прочности на скол. Величина  $F$  должна обеспечить разрушение коры на полную ее толщину  $h$  с последующим сдвигом разрушенного слоя вдоль поверхности ослабления.

Базируясь на математических моделях [1,2], произведены расчеты по определению величины  $F$ .

Расчеты выполнены для условий окорки свежесрубленных (влажность порядка  $W = 100\%$ ) лесоматериалов сосны зимней заготовки при температуре воздуха  $T = -5^\circ\text{C}$ . Прочность коры на скол  $\tau = 0,65$  МПа, диаметр  $d$  бревна изменялся от 0,2 до 0,6 м.

Использован роторно-скребковый станок ОК-63 со следующими техническими характеристиками: скорость подачи кряжа  $v = 0,15-0,6$  м/с; угол заточки фрезы  $\alpha = 1,74-2,35$  радиан; частота вращения ротора  $\omega = 135-180$  об/мин, ширина скребка  $L = 0,366$  м; коэффициент перекрытия  $K_n = 1-4$ .

В таблицы 1 для бревен различного диаметра приведены данные о силе  $F$  при фиксированных остальных параметрах:  $v = 0,45$  м/с;  $\omega = 180$  об/мин,  $\alpha = 1,74$  радиан;  $K_n = 2$ .

Таблица 1

**Величина  $F$  при окорке балансов сосны различного диаметра**

Диаметр бревна, м	Удельная сила окорки, кН/м
0,20	10,77
0,30	13,17
0,45	16,42
0,50	19,37
0,55	21,07

0,60	22,28
------	-------

При влажности образцов  $W$ , отличной от 100%, данные таблицы 1 корректируются путем их умножения на поправочный коэффициент  $K_w$ , равный:

$$K_w = \left(\frac{W}{100}\right)^{-1,3} \quad (1)$$

Если окорка осуществляется при отличной от  $-5^\circ\text{C}$  температуре воздуха, данные таблицы 1 умножают на коэффициент  $K_T$ :

$$K_T = 0,0243T^2 - 0,0629T + 1,01 \quad (2)$$

Окорка при скоростях  $v$  подачи кряжа, отличающихся от значения  $v=0,45\text{м/с}$ , требует корректировки данных таблицы 1 умножением на коэффициент  $K_v$ :

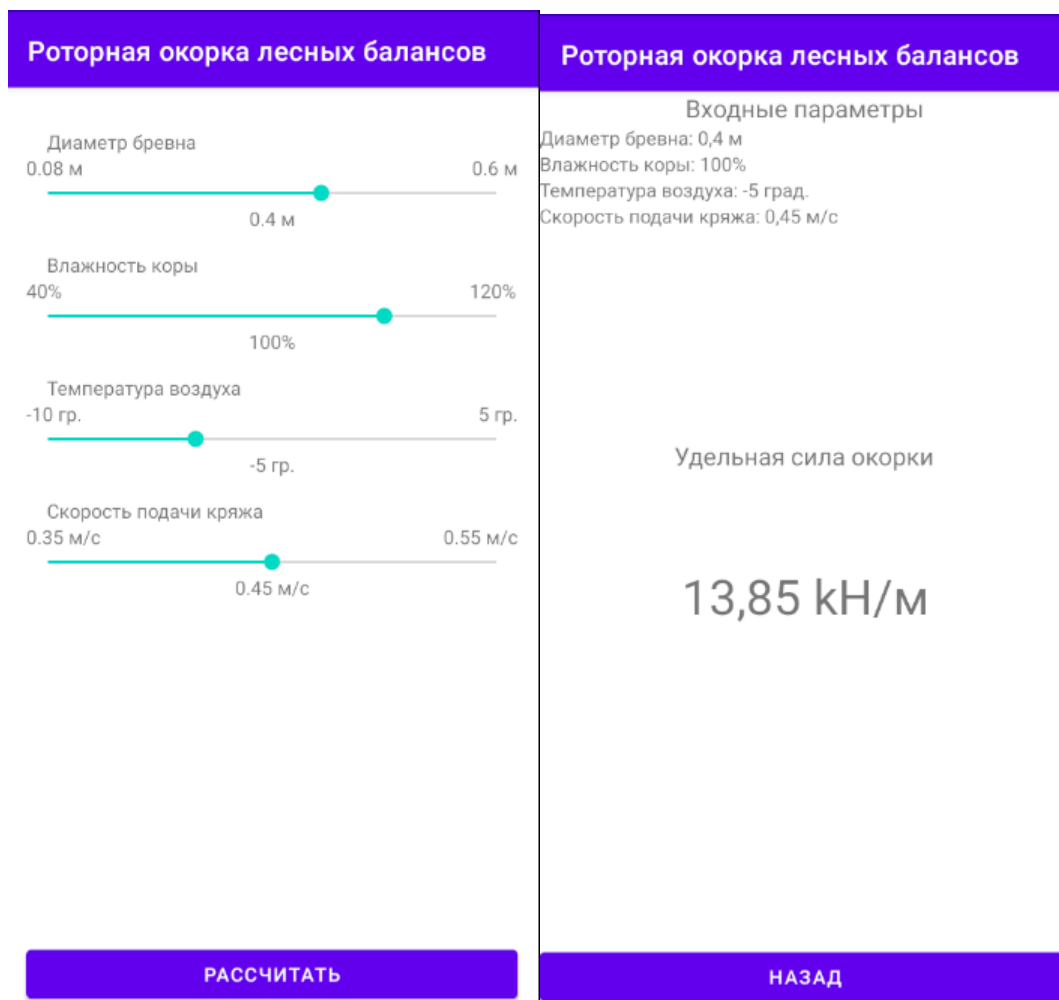
$$K_v = \sqrt[3]{-9,5v + 5,25} \quad (3)$$

Соотношения (1)-(3) основаны на закономерностях контактного разрушения массивов сплошной среды различной природы под действием инденторов (штампов) различной формы [6-9]. В ходе моделирования были установлены необходимые регрессионные уравнения с высоким (не менее 0,95) коэффициентом детерминации, которые были положены в основу при определении поправочных коэффициентов (1)-(3).

Автоматизация расчетов технологических параметров роторной окорки позволяет оператору станка избежать необходимости анализировать все условия и факторы, учитываемые при расчетах. Вместо этого ему предлагается воспользоваться разработанным приложением.

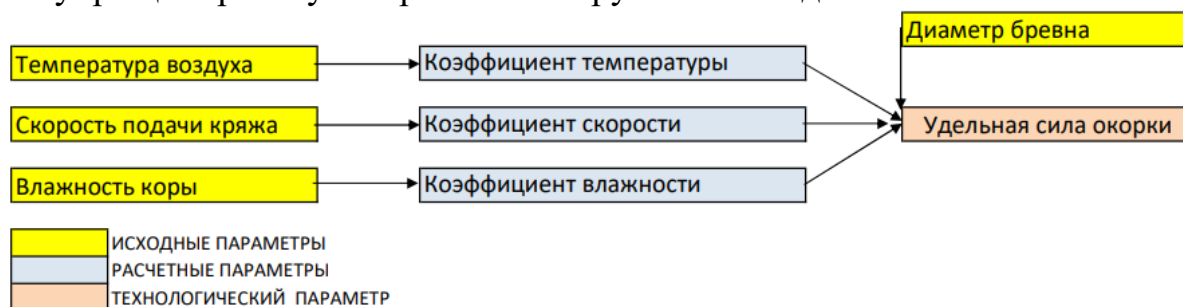
С помощью простого интерфейса оператор может отслеживать все необходимые технические и технологические параметры роторной окорки лесных балансов непосредственно с экрана своего портативного устройства.

Интерфейс создаваемого приложения выполнен в максимально упрощенном виде, для того чтобы оператор мог разобраться с его функционалом.



**Рис. 1.** Интерфейс приложения и результат вычислений

Функционал приложения напрямую связан с данными таблицы 1 и формулами (1)-(3), которые лежат в основе определения удельной силы окорки  $F$ , а предоставляемый пользователю интерфейс позволяет без особого труда выбрать из предлагаемых параметров ввода необходимые данные, что упрощает работу по сравнению с ручным методом.



**Рис.2.** Принцип работы приложения

После ввода исходных данных с помощью интерфейса приложения на основе разработанного функционала оператор на выходе получает основной технологический параметр - силу  $F$ .

Для примера, приведенного на рисунке 1, получено значение удельной силы окорки  $F=13,85$  кН/м. Если температура  $T$  понизится в 2 раза до

$T = -10^{\circ}\text{C}$ , величина  $F$  возрастет до значения  $F=29,9$  кН/м, т.е. более, чем в 2 раза, что свидетельствует о нелинейной связи параметров  $F$  и  $T$ .

Именно этим обусловлена необходимость пропарки мерзлых образцов древесины перед окоркой. Так, при прочих равных условиях повышение температуры образцов до  $0^{\circ}\text{C}$  приводит к снижению величины  $F$  до уровня  $F=7,29$  кН/м.

Таким образом, с помощью данного приложения значительно упрощается задача оперативной оценки технологических параметров роторной окорки, используя портативное устройство.

При этом оператору предоставляется возможность учесть технические параметры роторного станка, условия окорки и основные физико-механические свойства кряжа, что повышает надежность расчетов.

### Библиографический список

1. Газизов А.М., Григорьев И.В., Кацадзе В.А., Шапиро В.Я., Мурашкин. Повышение эффективности механической окорки лесоматериалов.- СПб: СПбГЛТА.-2009.-239 с.
2. Газизов А.М., Шапиро В.Я., Григорьев И.В. Влияние влажности на развитие процесса разрушения коры при роторной окорке// Вестник Московского государственного университета леса -Лесной вестник.-2008.-№6.-С.129-133.
3. Григорьев И.В., Шапиро В.Я., Гулько А.Е. Математическая модель групповой окорки лесоматериалов в окорочных барабанах// Научное обозрение.-2012.-№4.С.154-171.
4. Шапиро В.Я., Григорьев И.В., Гулько А.Е. Анализ методов расчета параметров и обоснование математической модели разрушения коры при групповой окорке древесины//Ученые записки Петрозаводского государственного университета.-2011.-№8(121).-С.92-96.
5. Газизов А.М., Шапиро В.Я., Григорьев И.В. Вариационный метод расчета и стабилизация параметров роторной окорки// Справочник. Инженерный журнал с приложением.-2009.-№7(148).-С.47-51.
6. Шапиро В.Я., Григорьев, Жукова А.И. Влияние сдвиговых деформаций на процесс циклического уплотнения почвы// Естественные и технические науки.-2006.-№1.-С.174-180.
7. Рудов С.Е., Шапиро В.Я., Григорьев И.В., Куницкая О.А., Григорьева О.И. Особенности взаимодействия трелевочной системы с оттаивающим почвогрунтом. Лесной вестник. ForestryBulletin.-2019.- Т.23.-№1.- С.52-61.
8. Шапиро В.Я., Григорьев И.В., Лепилин Д.В., Жукова А.И. Моделирование уплотнения почвогрунта в боковых полосах трелевочного волока с учетом изменчивости трассы движения //Ученые записки Петрозаводского государственного университета.-2010.-№ 6 (111).-С.61-64.
9. Шапиро В.Я., Григорьев И.В., Жукова А.И., Иванов В.А. Исследование механических процессов циклического уплотнения почвогрунта при динамических нагрузках.-Вестник КрасГАУ.-2008.-№1.-С.163-175.

**Н.П. Васильев**, кандидат технических наук, доцент  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
nikpv@mail.ru

**М.Р. Вагизов**, кандидат технических наук, доцент  
Зав. кафедрой информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
bars-tatarin@yandex.ru

## WEBGL ДЛЯ НАГЛЯДНОГО ПРЕДСТАВЛЕНИЯ ГЕОПОЛЕЙ

***Аннотация.** В статье рассматривается разработка технологии моделирования и визуализации поверхности рельефа на основе данных WebGL и данных SRTM.*

***Ключевые слова:** технологии WebGL, моделирование геополей, данные SRTM, рендеринг.*

### **Введение**

Если распределение некоторого параметра определяется географическими координатами, то такое пространственное распределение часто называют геополем. Это может быть, например, давление, температура, плотность химических элементов, геофизические поля и, конечно, высота рельефа местности над уровнем моря. Формально геополе — это скалярная функция от двух пространственных координат. Наиболее наглядным представлением геополя является его графическое представление в виде поверхности, совмещённое с картографическим отображением соответствующего участка земной поверхности.

В данной статье предлагается технология визуального представления геополей в браузерах на основе WebGL и Web-технологий карт Google (или Yandex). Технология тестируется на данных высот рельефа местности. В настоящее время высоты точек земной поверхности доступны из различных источников. Наиболее популярными являются результаты, полученные с помощью спутников NASA — это SRTM-данные (SRTM — Shuttle radar topographic mission) [1] и данные ASTER GDEM [2] (ASTER — Advanced Spaceborne Thermal Emission and Reflection Radiometer, GDEM — Global Digital Elevation Model). Независимо от источника структура данных одинакова — это регулярная сетка высот точек, координаты которых варьируются с фиксированным шагом по широте и долготе, в одну или в три секунды. Возникает классическая задача построения математической модели поверхности, определяемой высотами сетки, с последующим визуальным воспроизведением этой поверхности. Решение возможно с помощью известных программных средств. Например, можно выполнить импорт SRTM-данных в ГИС Arcinfo Workstation [3]. Теоретически также можно представить использование популярных программ для работы с

трёхмерной графикой, например, Blender или 3dsMax, если предварительно конвертировать сетку исходных высот в требуемый формат. В любом случае требуется промежуточная стадия подготовки и импорта данных, которая обычно выполняется вручную.

С появлением WebGL [4, 5] стало доступно взаимодействие с программируемыми графическими процессорами непосредственно с HTML-страницы. С помощью клиентского JavaScript теперь можно добраться до низкоуровневого API библиотеки OpenGL ES (for Embedded Systems) и её колоссальных возможностей рендеринга трёхмерной графики. Заметим, что кроссплатформенная библиотека OpenGL ES является частью современных мобильных платформ, поэтому и WebGL может быть использован для мобильных приложений, разрабатываемых с помощью Web-технологий, например, Cordova [10, 11, 12].

Воспользуемся WebGL и клиентским JavaScript для рендеринга сетки высот. Воспроизводить будем триангулированную поверхность — результат непосредственной кусочно-линейной интерполяции треугольниками по заданным точкам. Очевидно, это непрерывная гранёная поверхность, которая имеет разрывы первых частных производных (а значит и нормалей) на рёбрах между гранями.

Предварительное сглаживание с помощью классических составных поверхностей Фергюсона (фактически это вариант двумерной сплайн-интерполяции) [6] или иных методов [7, 8] не применялись. Заметим, что поскольку WebGL работает только с треугольниками, то после любого варианта построения гладкой поверхности всё равно потребуется триангуляция. Триангуляция может быть выбрана достаточно мелкой, с тем чтобы грани визуально не были заметны. Без сглаживания можно ожидать такого эффекта только при достаточно большом размере сетки, а значит и треугольных граней. Другое преимущество сглаживания — это однозначное определение нормалей в вершинах треугольных граней.

Достоинство непосредственной триангулированной поверхности — это отсутствие объёмных вычислений особенно при большом количестве точек сетки. С увеличением размера сетки следует ожидать увеличения погрешности и неустойчивости вычислений. С другой стороны, неоднозначность нормалей в точках стыковки граней, то есть в вершинах триангулированной поверхности, влечёт увеличение размеров буферов вертексов WebGL, поскольку общие вершины соседних граней и соответствующие им нормали должны храниться отдельно. В результате, координаты общих вершин смежных граней дублируются.

В настоящей статье последовательно излагается технология визуального воспроизведения SRTM-данных с помощью WebGL и интеграция этой технологии с картами Google. Технология опробована на примере построения WEB-инструмента для воспроизведения произвольных участков карты, выбранных с помощью мыши, в «режиме реального времени». Инструмент позволяет манипулировать рядом параметров, например, размером сетки и визуально оценивать качество результата. Ключевые положе-



ния сопровождаются фрагментами кода JavaScript, отражающими суть технологии с подробными пояснениями. Вместо SRTM-данных могут выступать любые данные, представляющие некоторое геополе.

### Переход к декартовым координатам

Для рендеринга триангулированной поверхности в WebGL требуется задать декартовы координаты вершин каждого треугольника. Предположим, что имеется сетка точек размером `nrows` (количество строк) на `ncols` (количество столбцов). Эта сетка является регулярной в географических координатах — шаг по широте и долготе фиксирован и равен трём секундам. Однако, в плане эта сетка уже не будет регулярной, поскольку соответствующую часть сферы нельзя без искажений спроецировать на плоскость. Если учесть, что длины параллелей зависят от широты, то шаг по оси  $x$  в каждой строке будет различаться. Длины меридианов постоянны, поэтому шаг по оси  $y$  в колонках будет постоянным. Сохраним длины параллелей, точнее их частей, соответствующих сетке и длину участка центрального меридиана (или центральных меридианов, если `ncols` — чётное). Начало координат по оси абсцисс совместим с нижним левым пикселем сетки, а по оси  $y$  с первой строкой. Результат представлен на рисунке 1.

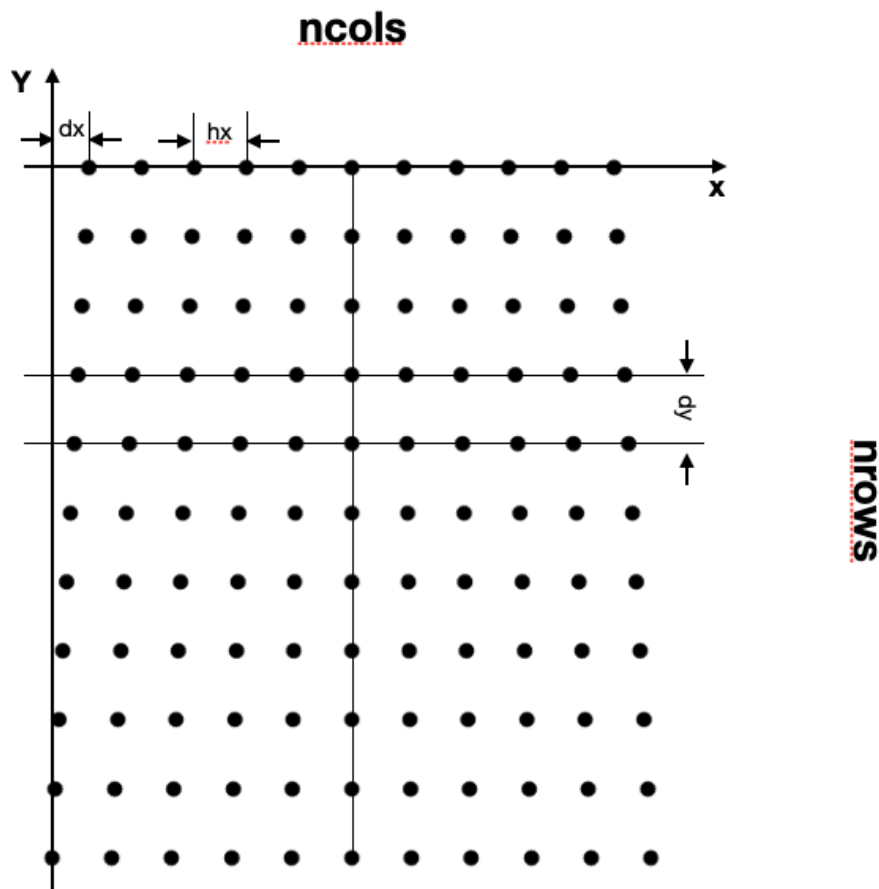


Рис. 1. Сетка точек размером `nrows` на `ncols` в декартовых координатах

Параметры сетки вычисляются следующим образом. Шаг в три секунды по оси  $x$  в ряду точек, соответствующему некоторой широте  $\varphi$ :

$$hx = hx_0 * \cos\left(\frac{\varphi\pi}{180}\right)$$

Здесь  $hx_0 = 11321.377778/1200$  — это шаг на экваторе — длина экватора в метрах, соответствующая трём секундам по долготе.

Шаг в три секунды по оси  $y$  является постоянным, поскольку длина меридианов одинакова:

$$hy = 3 * 30.8707947531 \text{ м.}$$

Для сохранения симметрии относительно центрального меридиана сетки каждый ряд точек сдвигаем относительно нижнего ряда на величину:

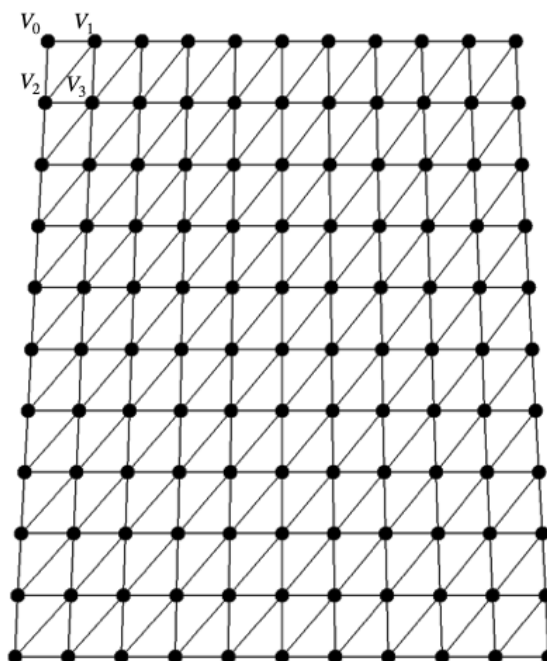
$$dx = (hxBottom - hx) * (ncols - 1) / 2$$

Здесь  $hxBottom$  — это шаг по оси  $x$  в нижнем ряду сетки.

Очевидно, что если задать широту верхнего ряда сетки, то можно вычислить декартовы координаты (абсциссу  $x$  и ординату  $y$ ) всех точек сетки. Аппликата  $z$  суть высота точки.

## Триангуляция

Один из возможных вариантов триангуляции представлен на рисунке 2.



**Рис. 2.** Вариант триангуляции. Для каждой ячейки сетки формируем два треугольника:  $V_0V_2V_1$  и  $V_2V_3V_1$ . Важно — координаты вершин в буфер вертексов должны быть загружены именно в таком порядке

Для того чтобы воспроизвести соответствующую триангулированную поверхность с помощью WebGL требуется сформировать буфер вертексов — буфер координат вершин каждого треугольника. Например, для первой ячейки требуется сформировать два треугольника:  $V_0V_2V_1$  и  $V_2V_3V_1$ , то есть в буфер вертексов загрузить шесть вершин, по три координаты

наты на каждую вершину, всего — восемнадцать чисел. Заметим, что вершины будут продублированы.

В WebGL есть механизм индексации вертексов. Этот механизм позволяет сформировать буфер без дублирования вершин, а порядок формирования треугольников указать в другом буфере с помощью индексов (номеров) вершин. Поскольку индексы — целые числа, которые в памяти занимают меньше места, чем координаты — числа с плавающей запятой, то возможна экономия памяти.

Однако, такой механизм в случае гранёной поверхности невозможен по следующей причине. Дело в том, что для расчёта того или иного варианта освещения поверхности в WebGL требуется для каждой вершины указать нормаль. Обычно, для нормалей формируется отдельный буфер, который строго соответствует буферу вершин: каждой тройке координат в буфере вершин соответствует тройка координат в буфере нормалей. По этой причине общие вершины приходится дублировать в строгом соответствии с количеством связанных граней, нормали которых различаются. Например, вершина  $V_3$  задействована в пяти треугольных гранях с разными нормальями. Таким образом, вершина  $V_3$  будет продублирована пять раз в буфере вертексов. Очевидно, если предварительно построить гладкую поверхность, а затем выполнить триангуляцию, то в каждой вершине нормаль будет определена однозначно.

## Формирование буферов вершин и нормалей

Работу с данными SRTM на языке JavaScript удобнее всего организовать в виде одноимённого класса. Положим, что высоты точек представлены в глобальном массиве SRTMDATA. Как они туда попадают, обсудим далее. Глобальными являются константы  $hx0$  (шаг в три секунды на экваторе) и  $hy$  (три секунды по меридиану), а также константа  $cellsize = 0.0008333333333333333 = 1/1200$ , содержащая значение три секунды, выраженное в градусах. Для формирования буферов вершин и нормалей достаточно знать широту верхнего ряда сетки  $lat0$ . Реализуем вычисления в виде метода класса calcGeometry:

```
classSRTM {
  constructor() {
    this.vrtxBuffer = null; this.normalBuffer = null;
    this.initProgram();
  }
  ... ..
  // прямоугольник:
  // v0 - v1
  // v2 - v3
  // треугольники: v0 - v2 - v1 v2 - v3 - v1;
  calcGeometry(lat0) {
    letnrows = SRTMDATA.length, ncols = SRTMDATA[0].length;
    lethxBottom =
    hx0 * Math.cos( Math.PI/180 * (lat0 - cellsize * (nrows - 1)) );
    letxyz = [], normals = [], cnt = 0;
    letlat = lat0, // строкеi=0
    hx = hx0 * Math.cos( Math.PI/180 * lat ),
```

```

y = 0.0,
dx = (hxBottom - hx) * (ncols - 1) / 2;
for(let i = 0; i < nrows - 1; i++) {
    // в (i+1)-ой строке
    let lat1 = lat - cellsize,
    hx1 = hx0 * Math.cos( Math.PI/180 * lat1 ),
    y1 = y - hy,
    dx1 = (hxBottom - hx1) * (ncols - 1) / 2;
    for(let j = 0; j < ncols - 1; j++) {
        let x = dx + j * hx, x1 = dx1 + j * hx1;
        // v0 - v2 - v1;
        xyz.push(x); xyz.push(y); xyz.push( SRTMDATA[i][j] ); // v0
        xyz.push(x1); xyz.push(y1); xyz.push( SRTMDATA[i+1][j] ); // v2
        xyz.push(x + hx); xyz.push(y); xyz.push( SRTMDATA[i][j+1] ); // v1
        this.calcNormal(xyz, normals);
        cnt += 3;
        // v2 - v3 - v1;
        xyz.push(x1); xyz.push(y1); xyz.push(SRTMDATA[i+1][j]);
        xyz.push(x1 + hx1); xyz.push(y1); xyz.push(SRTMDATA[i+1][j+1]);
        xyz.push(x + hx); xyz.push(y); xyz.push( SRTMDATA[i][j+1] );
        this.calcNormal(xyz, normals);
        cnt += 3;
    }
    lat = lat1; hx = hx1; y = y1; dx = dx1;
}
this.cnt = cnt;
this.vrtxBuffer =
this.initArrayBuffer(newFloat32Array(xyz), this.vrtxBuffer);
this.normalBuffer =
this.initArrayBuffer(newFloat32Array(normals), this.normalBuffer);
if( !this.vrtxBuffer || !this.normalBuffer ) return -1;
return cnt;
}
...
}

```

В представленном фрагменте кода формируется массив с именем `xyz` координат вершин треугольников. Треугольники загружаются в порядке, в котором они будут воспроизводиться WebGL. Параллельно формируется массив координат нормалей `normals`.

Вычисление нормалей для очередных трёх вершин выполняется стандартным образом — в результате вычисления векторного произведения векторов, образованных сторонами треугольника. Важно, чтобы направление нормали, которое указывает лицевую сторону грани, соответствовало движению винта при его откручивании в направлении обхода вершин. На рисунке 3 показано формирование нормали для треугольника  $V_0V_2V_1$ .

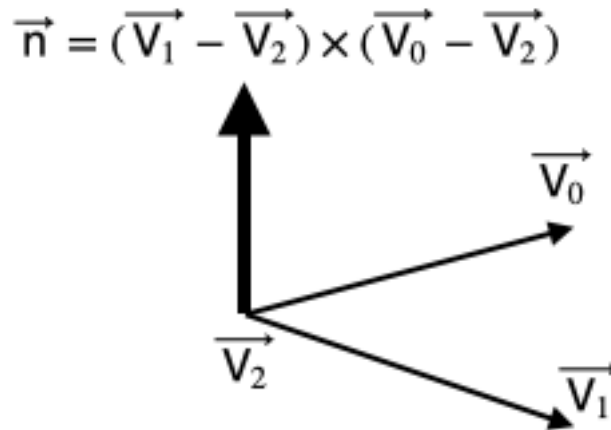


Рис. 3. Нормаль для треугольника  $V_0V_2V_1$

Вычисления нормалей для вершин очередного треугольника, сформированного в массиве xyz, реализуем в методе calcNormal:

```
calcNormal(xyz, normals) {
let last = xyz.length - 1;
let v1 = [
xyz[last-2] - xyz[last-5],
xyz[last-1] - xyz[last-4],
xyz[last] - xyz[last-3]
];
let v2 = [
xyz[last-8] - xyz[last-5],
xyz[last-7] - xyz[last-4],
xyz[last-6] - xyz[last-3]
];
// векторное произведение
let v1xv2 = [
v1[1] * v2[2] - v2[1] * v1[2],
v2[0] * v1[2] - v1[0] * v2[2],
v1[0] * v2[1] - v2[0] * v1[1]
];
// нормализация
let g = Math.sqrt(
v1xv2[0] * v1xv2[0] + v1xv2[1] * v1xv2[1] + v1xv2[2] * v1xv2[2]
);
let x = v1xv2[0]/g, y = v1xv2[1]/g, z = v1xv2[2]/g;
// каждой вершине треугольника по нормали
normals.push(x); normals.push(y); normals.push(z);
normals.push(x); normals.push(y); normals.push(z);
normals.push(x); normals.push(y); normals.push(z);
}
```

Метод получает в качестве аргумента массив xyz, в котором уже загружены координаты вершин очередного треугольника. Вычисляются вектора V1 и V2, совпадающие со сторонами этого треугольника, их векторное произведение, которое затем нормализуется. Вычисленные таким образом координаты нормали дублируются три раза в массиве normals для каждой вершины.

Массивы `xyz` и `normals` передаются в метод `initArrayBuffer`, где окончательно формируются буфера вертексов и нормалей для использования шейдерами WebGL:

```
initArrayBuffer(data, buf) {
    // создадим буферный объект
    let buffer = buf || gl.createBuffer();
    if(!buffer) {
        console.log('Failed to create the buffer object');
        return null;
    }
    // загрузим данные в буфер
    gl.bindBuffer(gl.ARRAY_BUFFER, buffer);
    gl.bufferData(gl.ARRAY_BUFFER, data, gl.DYNAMIC_DRAW);
    return buffer;
}
```

Здесь глобальная переменная `gl` представляет контекст WebGL [4,5].

### Выборка произвольной сетки высот размером `nrows` на `ncols`

Для исследования результатов воспроизведения с помощью WebGL триангулированной поверхности необходимо иметь возможность оперативно формировать сетку высот произвольного размера и заданными географическими координатами центра. Например, центр можно задавать непосредственно на карте по клику мышкой в рамках выбранного SRTM-участка размером 5 на 5 градусов. Следует отметить, что SRTM-данные поставляются участками размером 5 градусов по широте и долготе (5x5-tile). Таким образом, всё пространство по долготе (от -180 до +180) разбито на 72 участка (360/5). Диапазон от 60 градусов северной широты до 60 градусов южной широты разбивается на 24 участка (120/5). Северные широты больше 60 градусов и южные широты больше 54 исключаются. Участки нумеруются, начиная с единицы. Номера участков указываются в названии файла с данными. Например, файл с названием `srtm_48_07` содержит 48-й участок по долготе и 7-й участок по широте. По указанным номерам можно определить долготу и широту левой верхней точки сетки, содержащейся в файле. Например, определяем долготу:

$$(48 - 1) * 5 - 180 = 55$$

И широту:

$$60 - (7 - 1) * 5 = 30$$

Таким образом, первая точка сетки высот размером 6000 на 6000, содержащаяся в файле `srtm_48_07` имеет координаты: 55 градусов восточной долготы и 30 градусов северной широты. В общем случае размер SRTM-участка по широте и долготе определяется следующим образом:

```
// Название файла данных соответствует номеру квадрата
// из расчета 72 квадрата по горизонтали (360/5)
// и 24 квадрата по вертикали (120/5)
// srtm_48_07
```

```
function getLatLng(num1, num2) { // num1 = 48, num2 = 07
  // долгота
  let lonLeft = (num1 - 1) * 5 - 180, // 55
  lonRight = num1 * 5 - 180; // 60

  let latTop = 60 - (num2 - 1) * 5; // 30 широта верхнего угла
  // 25 + 5 / 6000 широта нижнего угла
  latBottom = 60 - num2 * 5 + 0.00083333333333333333;

  return [[latBottom, lonLeft], [latTop, lonRight]];
}
```

Зная размеры выбранного участка, можно представить его на карте Google, используя JavaScript и API, предлагаемый Google для работы с картами.



**Рис. 4.** Файл srtm\_36\_02 соответствует представленному участку размером 5 на 5 градусов. Левая верхняя точка сетки размером 6000 на 6000 имеет следующие географические координаты:  $(36 - 1) * 5 - 180 = -5$  градусов по долготе и  $60 - (2 - 1) * 5 = 55$  градусов по широте

Границы, представленные на рисунке 4, получены с помощью универсальной функции для любого SRTM-участка с произвольным порядковым номером по долготе (num1) и широте (num2):

```
function draw5x5(num1, num2) {
  let pnts = getLatLng(num1, num2);
```



```

    let latBottom = pnts[0][0], latTop = pnts[1][0], lonLeft =
pnts[0][1], lonRight = pnts[1][1];

    let areaPath = [
        new google.maps.LatLng(latBottom, lonLeft),
        new google.maps.LatLng(latTop, lonLeft),
        new google.maps.LatLng(latTop, lonRight),
        new google.maps.LatLng(latBottom, lonRight)
    ];

    let polygonOptions = {
        paths: areaPath,
        strokeColor: '#FF0000',
        strokeOpacity: 0.9,
        strokeWeight: 3,
        fillColor: '#FFFF00',
        fillOpacity: 0.25
    };

    let polygon = new google.maps.Polygon(polygonOptions);
    polygon.setMap(map);
}

```

Здесь границы участка вычисляются с помощью функции `getLatLng(num1, num2)`, представленной выше, а глобальная переменная `map` представляет карту.

Для реализации исследований необходимо иметь возможность получать сетку высот заданного размера `nrows` (количество строк) на `ncols` (количество столбцов), которая, является частью сетки 6000 на 6000 точек выбранного участка. Центр сетки будем определять кликом мышкой на карте. С этой целью, после инициализации карты добавляем необходимые обработчики событий мыши:

```

functioninitMap() {
    varmapOptions = {
        zoom: 10,
        mapTypeId: google.maps.MapTypeId.TERRAIN
    };
    letmapElement = document.getElementById('mapDiv');
    map = newgoogle.maps.Map(mapElement, mapOptions);

    letparseRes = parseSrtmFnm(fnm);

    fitBounds( parseRes[0], parseRes[1] );

    letcontrolDiv = document.createElement('div');
    controlDiv.className = 'mapControl'; controlDiv.id = 'mapCoordinates';
    controlDiv.innerHTML = 'Lat/Lng: 0.00 / 0.00';

    map.controls[google.maps.ControlPosition.LEFT_BOTTOM].push(controlDiv);
    google.maps.event.addListener(map, 'mousemove', function(e) {
        letlat = e.latLng.lat(), lon = e.latLng.lng();
        letcoordinateText = 'Lat/Lng: ' +
        lat.toFixed(6) + ' / ' + lon.toFixed(6);
        letsrtmInfo = getNum12(lat, lon);
        controlDiv.innerHTML = coordinateText + '<br/>' + srtmInfo[2] +
        '<br/>Zoom: ' + map.getZoom();
    });
}

```

```

google.maps.event.addListener(map, 'click', onMapClick);
draw5x5line( parseRes[0], parseRes[1] );
}

```

В представленном фрагменте кода масштаб и границы карты подгоняются под выбранный SRTM-участок с помощью функции fitBounds:

```

function fitBounds(num1, num2) {
  let pnts = getLatLng(num1, num2);
  let latBottom = pnts[0][0], latTop = pnts[1][0],
      lonLeft = pnts[0][1], lonRight = pnts[1][1];
  map.fitBounds({
    north: latTop, south: latBottom,
    east: lonRight, west: lonLeft
  });
}

```

Функции fitBounds передаются номера участка по широте и долготе, которые получаются в результате разбора названия файла участка (fnm=srtm\_36\_02.tif) с помощью функции parseSrtmFnm:

```

function parseSrtmFnm(fnm) {
  let ret = [0, 0];
  let res = fnm.match(/srtm_(\d+)_(\d+)/);
  if(res) ret = [ parseInt(res[1]), parseInt(res[2]) ];
  return ret;
}

```

Отметим также, что на карте (рис. 4) отображается информационное окно с географическими координатами курсора мыши, названия соответствующего файла SRTM и параметра zoom. Указанная информация необходима для более точного выбора сетки точек rows на cols и формируется в обработчике событий перемещения мыши mousemove. Здесь также решается обратная задача: по широте и долготе определяются номера SRTM-участка:

```

function getNum12(lat, lon) {
  let num1 = 0, num2 = 0;
  if(-54 <= lat && lat <= 60) {
    num2 = Math.floor((60 - lat) / 5) + 1;
  }
  num1 = Math.floor((lon + 180) / 5) + 1;
  return [num1, num2, 'srtm_' + pad(num1, 2) + '_' + pad(num2, 2)];
}

function pad(num, size) {
  let s = num + "";
  while (s.length < size) s = "0" + s;
  return s;
}

```

### Формирование сетки высот по заданным координатам центра

Для выборки высот из SRTM-файла требуется решить следующую задачу: по заданным географическим координатам центра участка сетки,

заданного размера (`nrows` на `ncols`) определить номера крайней левой и крайней правой колонки (`c1l`, `clr`), а также номера верхнего и нижнего ряда (`rwt`, `rwb`) в рамках шеститысячной SRTM-сетки.

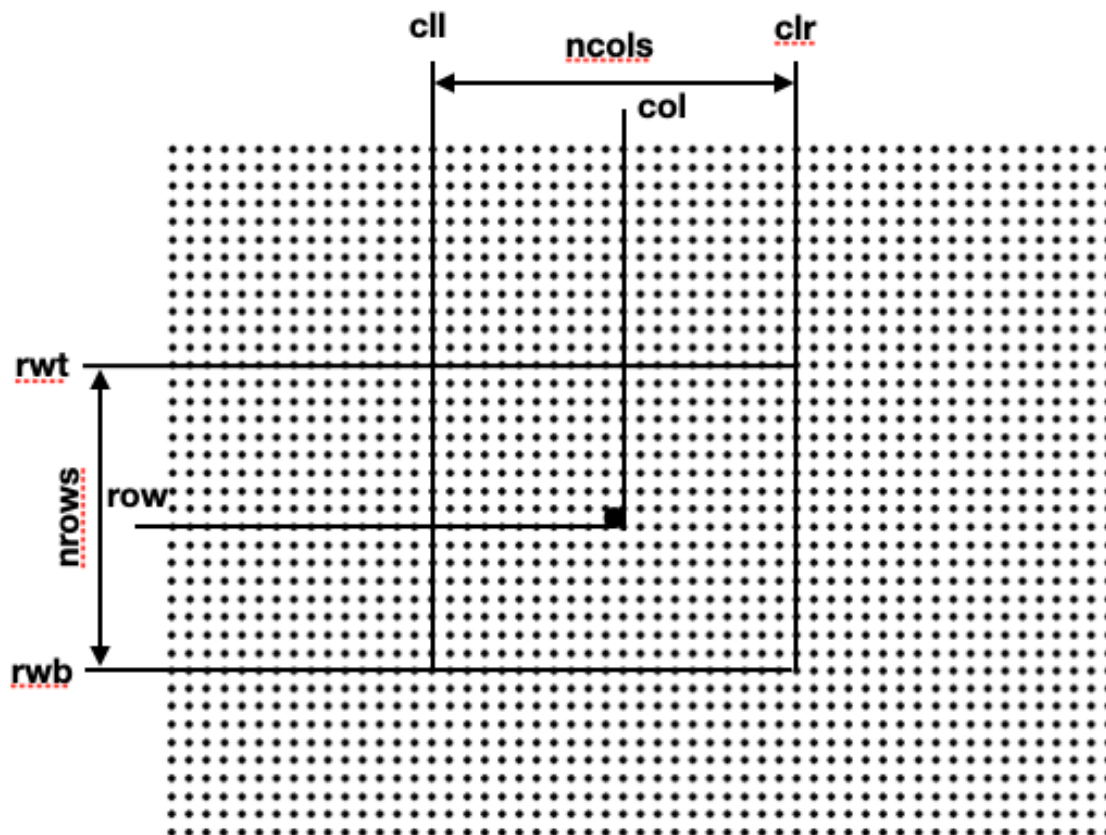


Рис. 5. Выборка участка сетки точек

Географические координаты центра определяются в обработчике события — клика мыши и здесь же решается поставленная задача:

```
function onMapClick(e) {
  getNRwsNCLs(); // вспомогательная функция для загрузки значений переменных NCLS и NRWS с размерами сетки (они же nrows и ncols) из предусмотренных в интерфейсе полей ввода

  let lat = e.latLng.lat(), lon = e.latLng.lng();
  let num12 = getNum12(lat, lon);
  let num1 = num12[0], num2 = num12[1];
  let pnts = getLatLng(num1, num2);
  let latBottom = pnts[0][0], lonLeft = pnts[0][1];
  // номер колонки и строки точки ближайшей к точке клика мышкой
  let col = Math.floor( (lon - lonLeft) * 1200 ),
  row = Math.floor( (lat - latBottom) * 1200 );
  row = 6000 - row + 1;

  // номера левой и правой колонки определяются по разному в зависимости
  // от четности числа колонок в сетке
  c1l = 0; clr = 0;
  let atLeft = 0, atRight = 0; if( NCLS % 2 == 0 ) { // чёт
    atLeft = NCLS / 2; atRight = atLeft - 1;
  }
  else {
```

```

atLeft = (NCLS - 1) / 2; atRight = atLeft;
}

if(col<atLeft) { cll = 1; clr = NCLS; }
elseif(6000 - col<atRight) { cll = 6000 - NCLS + 1; clr = 6000; }
else { cll = col - atLeft; clr = col + atRight; }

// номера верхнего и нижнего ряда определяются аналогично
rwt = 0; rwb = 0;
letfromBelow = 0, above = 0; if( NRWS % 2 == 0 ) { // чёт
above = NRWS / 2; fromBelow = above - 1;
}
else {
above = (NRWS - 1) / 2; fromBelow = above;
}

if(row<above) { rwt = 1; rwb = NRWS; }
elseif(6000 - row<fromBelow) { rwt = 6000 - NRWS + 1; rwb = 6000; }
else { rwt = row - above; rwb = row + fromBelow; }
//*****

lettop = latBottom + (6000 - rwt) / 1200,
bottom = latBottom + (6000 - rwb) / 1200;
letleft = lonLeft + cll / 1200, right = lonLeft + clr / 1200;

LatTopOfRectangle = top; // запоминаем широту верхнего ряда для
// calcGeometry

// граница вокруг участка
if( rectPolyline ) rectPolyline.setMap(null);

letlinePath = [
newgoogle.maps.LatLng(top, left), newgoogle.maps.LatLng(top, right),
newgoogle.maps.LatLng(bottom, right), newgoogle.maps.LatLng(bottom,
left),
newgoogle.maps.LatLng(top, left)
];

letlineOptions = {
path: linePath,
strokeWeight: 3,
strokeColor: '#00FF00',
strokeOpacity: 0.8
};

rectPolyline = newgoogle.maps.Polyline(lineOptions);
rectPolyline.setMap(map);

loadData();
}

```

Загрузка данных осуществляется в результате Ajax-запроса:

```

functionloadData() {
varxhr = newXMLHttpRequest();
xhr.open('GET', 'read.php?fnm=' + fnm + '&c11=' + c11 + '&clr=' + clr
+'&rwb=' + rwb + '&rwt=' + rwt, true);
xhr.onload = function () {
if(xhr.response) {
SRTMDATA = JSON.parse(xhr.response);
}
}
}

```

```

srtmapp.drawScene();
    }
};
xhr.onerror = function() {
console.log("Anerroroccurred");
};
xhr.send();
}

```

После успешной загрузки (с помощью read.php) высот сетки точек формируется массив SRTMDATA и выполняется прорисовка триангулированной поверхности (drawScene).

Не будем подробно обсуждать выборку высот из SRTM-файла с помощью php, заметим только, что данные SRTM поставляются в двух форматах: в текстовом (расширение файла asc) и в формате растровых данных GeoTIFF (расширение файла tif). Второй формат предпочтительнее, поскольку имеет меньший размер, однако, его интерпретация является более сложной. Подробно работа с GeoTIFF форматом на php представлена в работе BobOsola, результаты которой выложены на github [9]. Здесь представим без дополнительных комментариев код read.php для выборки участка высот с заданными номерами крайних значений рядов (rwt, rwb) и колонок (c1l, clr):

```

<?php
// standard TIFF constants
$TIFF_ID = 42; // magic number which identifies a TIFF file
$TAG_STRIPOFFSETS = 273; // identifying code for 'StripOffsets' tag
$TAG_IMAGE_WIDTH = 256;
$TAG_IMAGE_LENGTH = 257;
$LEN_IFD_FIELD = 12; // the number of bytes in each IFD entry
$BIG_ENDIAN = "MM"; // byte order identifiers located at bytes 0-1
$LITTLE_ENDIAN = "II";
// *****

$fnm = $_REQUEST['fnm']; //$fnm = 'srtm_42_01.tif';
$fp = fopen("./GeoData/$fnm", 'rb');
if($fp === false) die("could not open the file '$fnm'");

// go to the file header and work out the byte order (bytes 0-1)
// and TIFF identifier (bytes 2-3)
fseek($fp, 0);
$dataBytes = fread($fp, 4);
$data = unpack('c2chars/vTIFF_ID', $dataBytes);

// check it's a valid TIFF file by looking for the magic number
$TIFF = $data['TIFF_ID'];
if($TIFF != $TIFF_ID) die("the file '$fnm' is not a valid TIFF file");

// convert the byte order code to ASCII to get Motorola or Intel ordering // identifiers
$byteOrder = sprintf('%c%c', $data['chars1'], $data['chars2']);

// the remaining 4 bytes in the header are the offset to the IFD
fseek($fp, 4);
$dataBytes = fread($fp, 4);
// unpack in whichever byte order was identified previously

```

```

// - this seems to be always 'II' but whether this is always the case
is // not specified
// so we do the check each time to make sure
if ($byteOrder == $LITTLE_ENDIAN) $data = unpack('VIFDoffset', $data-
Bytes);
elseif ($byteOrder == $BIG_ENDIAN) $data = unpack('NIFDoffset', $data-
Bytes);
else die("could not determine the byte order of the file '$fnm'");

// now jump to the IFD offset and get the number of entries in the IFD
// which is always stored in the first two bytes of the IFD
fseek($fp, $data['IFDoffset']);
$dataBytes = fread($fp, 2);
$data = ($byteOrder == $LITTLE_ENDIAN) ? unpack('vcount', $dataBytes) :
unpack('ncount', $dataBytes);
$numFields = $data['count'];

// iterate the IFD entries until we find the ones we need
for($i = 0; $i < $numFields; $i++) {
    $dataBytes = fread($fp, $LEN_IFD_FIELD);
    $data = ($byteOrder == $LITTLE_ENDIAN) ?
        unpack('vtag/vtype/Vcount/Voffset', $dataBytes) :
        unpack('ntag/ntype/Ncount/Noffset', $dataBytes);
    switch($data['tag']) {
        case $TAG_IMAGE_WIDTH :
            $numDataCols = $data['offset']; break;
        case $TAG_IMAGE_LENGTH :
            $numDataRows = $data['offset']; break;
        case $TAG_STRIPOFFSETS :
            $stripOffsets = $data['offset']; break;
    }
}

$rwt = $_REQUEST['rwt']; $c11 = $_REQUEST['c11']; $rwb =
$_REQUEST['rwb']; $clr = $_REQUEST['clr']; //$rwt = 10; $c11 = 10; $rwb
= 260; $clr = 260;
$cols = $clr - $c11 + 1;
$selev = Array();
for($row = $rwt; $row <= $rwb; $row++) {
    fseek($fp, $stripOffsets + $row * 4);
    $dataBytes = fread($fp, 4);
    $data = unpack('VdataOffset', $dataBytes);
    $firstColOffset = $data['dataOffset'];
    $requiredColOffset = $c11 * 2;
    fseek($fp, $firstColOffset + $requiredColOffset);
    $dataBytes = fread($fp, 2 * $cols);
    $data = unpack("s$cols", $dataBytes);
    $selevRow = Array();
    foreach ($data as $key => $value) $selevRow[] = $value;
    $selev[] = $selevRow;
}

echo json_encode($selev);
?>

```

В итоге формируется массив высот выбранного участка в формате json, который интерпретируется на клиентской стороне.

## Рендеринг триангулированной поверхности

Подведём промежуточный итог. С тем чтобы иметь возможность оперативно экспериментировать с различными участками с размерами сетки `nrows` на `ncols`, центр участка выбирается мышкой и выделяется маленьким прямоугольником в рамках большого шеститысячного прямоугольника (см. рисунок 6).

NROWS:  16.2км. ..27.7км. ..928м.  
NCOLS:



**Рис. 6.** Требуемый участок заданных размеров (`nrows` на `ncols`) выбирается мышкой и выделяется прямоугольником. Большой прямоугольник соответствует исходной шеститысячной сетке. Указывается размер участка и максимальная высота

Итак, участок выбран, сетка высот загружена в массив `SRTM`, широта верхнего ряда сетки задана в переменной `LatTopOfRectangle`. Таким образом, всё готово для формирования буферов вертексов и нормалей, с помощью методов класса `SRTM`, описанных выше. Остаётся передать буфера шейдерам и прорисовать поверхность с помощью `WebGL`. Все эти действия собраны в рамках метода `drawScene` класса `SRTMAPP`:

```
class SRTMAPP {
  constructor() {
    this.canvas = document.getElementById('webgl');
    gl = getWebGLContext(this.canvas);
    if(!gl) {
      console.log('Failed to get the rendering context for WebGL');
      return;
    }
    this.resizeCanvasToDisplaySize();
    gl.viewport(0, 0, gl.canvas.width, gl.canvas.height);
    // Set the clear color and enable the depth test
    gl.clearColor(0, 0, 0, 1);
    gl.enable(gl.DEPTH_TEST);
  }
}
```



```

this.srtm = newSRTM();
}

drawScene() {
    // Unbindthebufferobject
    gl.bindBuffer(gl.ARRAY_BUFFER, null);
    letsrtm = this.srtm;
    // создаём буфера вертексов и нормалей
    srtm.calcGeometry(LatTopOfRectangle);
    if (srtm.cnt < 0) {
        console.log('Failed to set the vertex information');
        return;
    }
    // размеры участка и размах по высоте
    let dx = srtm.maxx - srtm.minx, dy = srtm.maxy - srtm.miny,
        dz = srtm.elevMax - srtm.elevMin;
    let d = Math.sqrt(dx * dx + dy * dy + dz * dz) / 2;
    // показываем размеры участка в интерфейсе
    let fld = document.querySelector('#SIZES');
    fld.innerHTML = (dx/1000).toFixed(1) + "км. .." +
        (dy/1000).toFixed(1) + "км. .." +
        (dz).toFixed(0) + "м.";
    // Calculate the view projection matrix
    let mvpMatrix = newMatrix4(); // Model view projection matrix
    mvpMatrix.setPerspective(30, this.canvas.width/this.canvas.height,
        1, 10 * d);
    mvpMatrix.lookAt(d, -1.5 * d, 0.7 * d, 0, 0, 0, 0, 0, 1);
    let modelMatrix = newMatrix4();
    // переносим центр поверхности в начало координат
    modelMatrix.translate(-srtm.centerX, -srtm.centerY,
        -(srtm.elevMin + srtm.elevMax) / 2);
    mvpMatrix.multiply(modelMatrix);

    // Clear color and depth buffer
    gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);

    this.srtm.draw(mvpMatrix);
}

resizeCanvasToDisplaySize() {
    let multiplier = window.devicePixelRatio || 1;
    let width = this.canvas.clientWidth * multiplier;
    let height = this.canvas.clientHeight * multiplier;
    if (this.canvas.width !== width || this.canvas.height !== height) {
        this.canvas.width = width; this.canvas.height = height;
    }
    return true;
}
return false;
}
}

```

В конструкторе класса реализованы стандартные действия инициализации WebGL-контекста [5] в глобальной переменной `gl` и создаётся экземпляр класса `SRTM`. Наконец, в методе `drawScene` выполняется расчёт геометрии (`calcGeometry`) — буферов вертексов и нормалей. Далее формируется матрица `mvpMatrix` проецирования сцены в `ClipSpace` — именно это пространство (от -1 до +1 по всем трём координатам) воспроизводится

шейдерами [5]. Предварительно выполняется перенос центра поверхности в начало координат (translate). Расположение камеры (точки зрения) выбирается в зависимости от размера  $d$  диагонали параллелепипеда, ограничивающего поверхность. По оси  $x$  отступаем на величину  $d$ , по оси  $y$  на величину  $-1.5d$  и поднимаемся на высоту  $0.7d$  (lookAt). Камера направлена в начало координат. Библиотеку аффинных и проекционных матричных преобразований (lookAt, setPerspective) можно заимствовать, например, из книги Коичи Мацуда [5].

Сформированная таким образом матрица используется в методе draw класса SRTM, в котором выполняется запуск шейдеров:

```
draw(mvpMatrix) {
  let program = this.program; if( !program ) return;
  gl.useProgram(program);
  gl.uniformMatrix4fv(program.u_MvpMatrix, false, mvpMatrix.elements);
  // буфер вертексов связываем с атрибутом a_Position в шейдере
  this.initAttributeVariable(program.a_Position, this.vrtxBuffer);
  // буфер нормалей связываем с атрибутом a_Normal в шейдере
  this.initAttributeVariable(program.a_Normal, this.normalBuffer);
  gl.drawArrays(gl.TRIANGLES, 0, this.cnt);
}

initAttributeVariable(a_attribute, buffer) {
  gl.bindBuffer(gl.ARRAY_BUFFER, buffer);
  gl.vertexAttribPointer(a_attribute, 3, gl.FLOAT, false, 0, 0);
  gl.enableVertexAttribArray(a_attribute);
}
```

Шейдеры — свойство program, а также ссылки на атрибутивные переменные a\_Position и a\_Normal, цвета, направление на источник света инициализируются в методе initProgram класса SRTM:

```
initProgram() { // шейдеры + атрибутивные + uniform-переменные
  this.program = null;

  // Shaders for srtm
  let VSHADER_SOURCE = document.querySelector("#vertex-shader-3d").text,
      FSHADER_SOURCE = document.querySelector("#fragment-shader-3d").text;

  // Initialize shaders
  let program = createProgram(gl, VSHADER_SOURCE, FSHADER_SOURCE);
  if( !program ) {
    console.log('Failed to initialize srtm shaders');
    return;
  }
  //

  // Get storage locations of attribute and uniform variables in program
  // object for srtm drawing
  program.a_Position = gl.getAttribLocation(program, 'a_Position');
  program.a_Normal = gl.getAttribLocation(program, 'a_Normal');

  program.u_MvpMatrix = gl.getUniformLocation(program,
```

```

'u_MvpMatrix');
    program.u_Color = gl.getUniformLocation(program, 'u_Color');
    program.u_DiffuseLight =
        gl.getUniformLocation(program, 'u_DiffuseLight');
    program.u_AmbientLight =
        gl.getUniformLocation(program, 'u_AmbientLight');
    program.u_LightDirection =
        gl.getUniformLocation(program, 'u_LightDirection');

    if(
        program.a_Position < 0 || program.a_Normal < 0 ||
        program.u_MvpMatrix < 0 || program.u_Color < 0 ||
        program.u_DiffuseLight < 0 || program.u_AmbientLight < 0 ||
        program.u_LightDirection < 0
    ) {
        console.log(
'Failed to get the storage location of attribute or uniform varia-
bles');
        return;
    }
    // Set the colors
    gl.useProgram(program);
    gl.uniform4fv(program.u_Color, [0.1, 1., 0.1, 1.]); // green
    gl.uniform3f(program.u_DiffuseLight, 1.0, 1.0, 1.0);
    // Direction of light
    let lightDirection = new Vector3([1., 1., 1.]);
    lightDirection.normalize();
    gl.uniform3fv(program.u_LightDirection, lightDirection.elements);
    // ambient light
    gl.uniform3f(program.u_AmbientLight, 0.2, 0.2, 0.2);

    this.program = program;
}

```

Ряд стандартных действий по инициализации шейдеров скрыты в функции createProgram [5].

Вершинный (vertexshader) и фрагментный шейдеры (fragmentshader) соответствует общепринятой практике рендеринга освещённой сцены с добавлением диффузного и фонового освещения [5] и не требует дополнительных комментариев:

```

<!-- vertex shader -->
<script id="vertex-shader-3d" type="x-shader/x-vertex">
attribute vec4 a_Position;
attribute vec4 a_Normal;          // Normal
uniform mat4 u_MvpMatrix;
uniform vec4 u_Color;           // own color
uniform vec3 u_DiffuseLight;    // Diffuse light color
uniform vec3 u_LightDirection; // Diffuse light direction
uniform vec3 u_AmbientLight;   // Color of an ambient light
varying vec4 v_Color;
void main() {
    gl_Position = u_MvpMatrix * a_Position;
    vec3 normal = normalize(a_Normal.xyz);
    // The dot product of the light direction and the normal
    float nDotL = max(dot(u_LightDirection, normal), 0.0);
    // Calculate the color due to diffuse reflection
    vec3 diffuse = u_DiffuseLight * u_Color.rgb * nDotL;
}

```

```

// Calculate the color due to ambient reflection
vec3 ambient = u_AmbientLight * u_Color.rgb;
// Add the surface colors due to
// diffuse reflection and ambient reflection
v_Color = vec4(diffuse + ambient, u_Color.a);
}
</script>

<!-- fragment shader -->
<script id="fragment-shader-3d" type="x-shader/x-fragment">
precision mediump float;
// Passed in from the vertex shader.
varying vec4 v_Color;
void main() {
    gl_FragColor = v_Color;
}
</script>

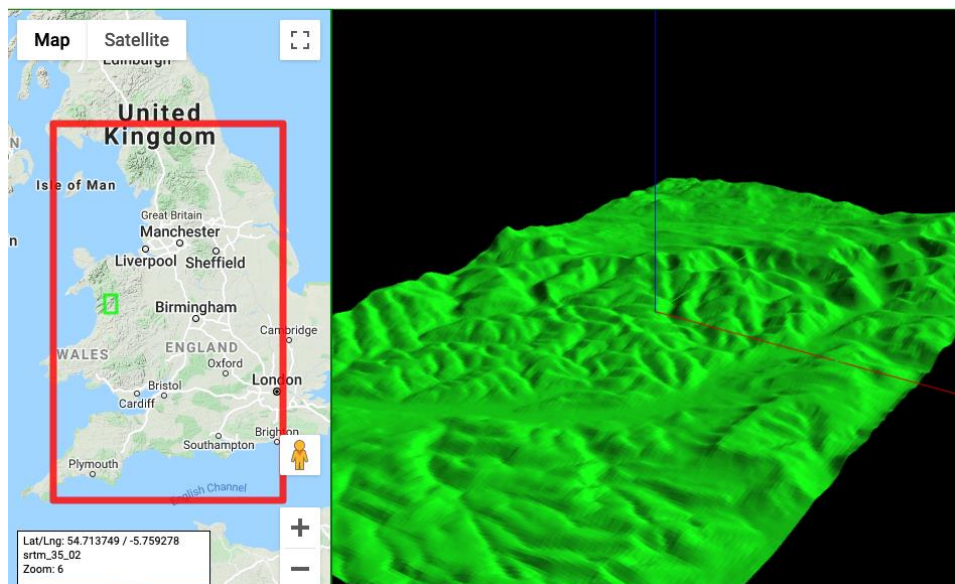
```

Заметим, что представленный код можно легко усовершенствовать, например, добавить возможность вращения поверхности вокруг координатных осей с помощью манипуляций мышью, снабдить рисунок координатными осями.

## Результаты экспериментов. Выводы

Общий вид интерфейса созданного WEB-инструмента для экспериментов и исследований представлен на рисунке 7.

NROWS:  16.9км. ..27.7км. ..883М.  
 NCOLS:



**Рис. 7.** В рамках используемого SRTM-фрагмента каждое нажатие мыши перерисовывает поверхность, соответствующую выбранному участку с заданными размерами сетки nrows на ncols. Интерфейс позволяет разворачивать поверхность с помощью манипуляций мышкой

Поставим следующую задачу: определить минимальный размер сетки, при котором изображение поверхности, построенной без предвари-

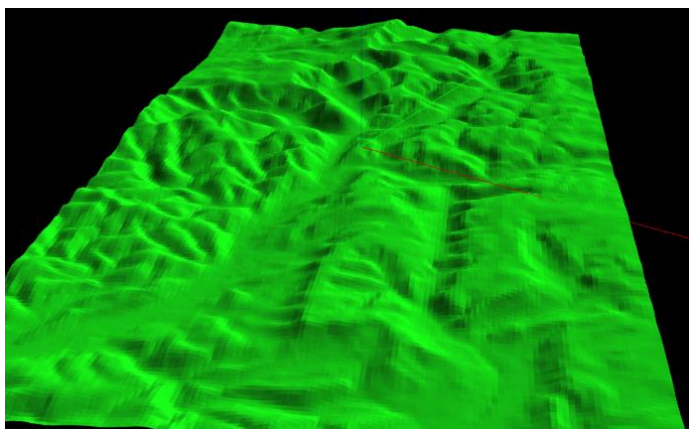
тельного сглаживания, остаётся приемлемым. Ответ можно получить экспериментально, с учётом того, что он, очевидно, носит частично субъективный характер и зависит от разрешения монитора, а также от размеров канвы. Эксперименты с рендерингом различных участков и разным размером сетки позволяют сделать следующие выводы:

Для размеров сетки `rows=ncols=200` и больше изображение является приемлемым, даёт полное визуальное представление о поверхности.

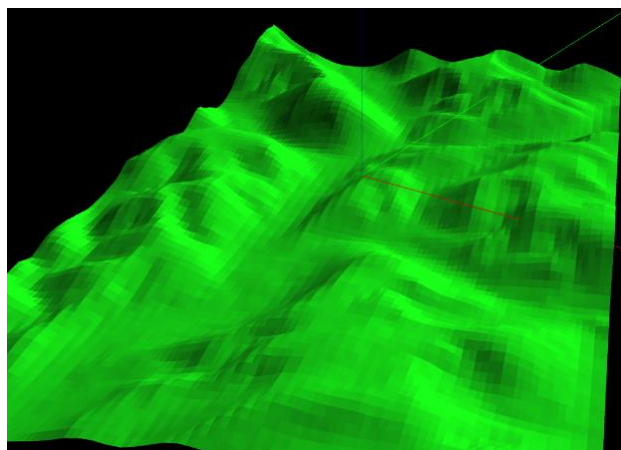
Для размеров сетки `rows=ncols=150` и до 50 на изображении появляются грани, тем не менее, изображение по-прежнему является приемлемым, даёт полное визуальное представление о поверхности.

На рисунках 8-10 представлены результаты воспроизведения участка с одинаковыми координатами центра (широта: 52.7, долгота: -3.7) и различными значениями размера сетки. Отметим, что камера (наблюдатель) смотрит на север, что соответствует положительному направлению оси  $u$  (то есть за рисунок).

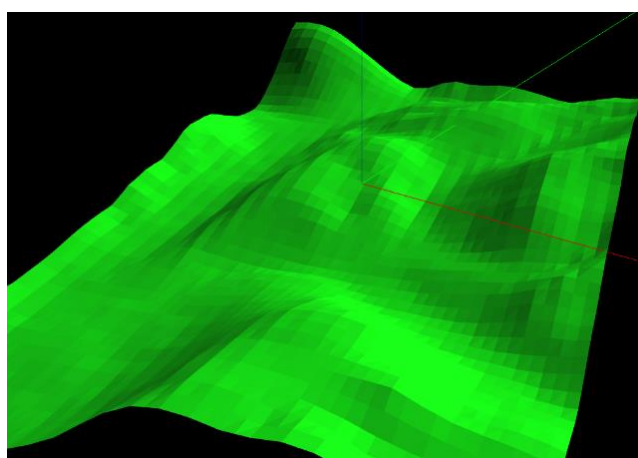
Таким образом, вычислительные затраты на сглаживание поверхности оправданы для размеров сетки меньше 50 точек. Для сетки больших размеров использование непосредственной триангуляции — оправдано, возможный эффект граней вполне окупается большой скоростью рендеринга и возможностью быстрой манипуляции изображением мышью даже для маломощных компьютеров без дискретной графической карты.



**Рис. 8.** Результаты воспроизведения участка `rows=250, ncols=250`.  
Размеры: 14.0км. 23.1км. 867м



**Рис. 9.** Результаты воспроизведения участка  $nrows=150$ ,  $ncols=150$ .  
Размеры: 8.4км., 13.8км., 737м



**Рис. 10.** Результаты воспроизведения участка  $nrows=50$ ,  $ncols=50$ .  
Размеры: 2.8км., 4.5км., 441м

Очевидные преимущества предложенной технологии воспроизведения геополей:

Исключается использование вспомогательных программ (например, ГИС).

Не требуется промежуточная стадия импорта данных, которая обычно выполняется вручную и поэтому воспроизведение графики происходит в реальном масштабе времени (online).

Задача решается исключительно средствами WEB.

Технология легко интегрируется с такими мощными инструментами, как Google карты.

#### **Библиографический список**

1. <https://srtm.csi.cgiar.org/srtmdata/>
2. <https://earthexplorer.usgs.gov>
3. <https://gis-lab.info/qa/srtm-ai.html>
4. <https://www.khronos.org/webgl/>

5. Коичи Мацуда, Роджер Ли WebGL: программирование трехмерной графики. — ДМК Пресс, 2015. — 494с.
6. Фокс А., Пратт М. Вычислительная геометрия. Применение в проектировании и на производстве: Пер. с англ. — М.: Мир, 1982. — 304с.
7. Тюкачев Н. А. Сгущение триангуляции поверхности со сглаживанием// Вестник воронежского государственного университета, серия: системный анализ и информационные технологии, 2010, №2 — Воронеж: ВГУ — с. 33-38.
8. Чекмарев Д.Т., Абузьяров М.Х., Cheng Wang. О гладкой интерполяции триангулированной поверхности// Проблемы прочности и пластичности. 2020, Том 82, № 2, — с.147-155.
9. <https://github.com/bobosola/PHP-SRTM>
10. Васильев Н.П. Мобильные Cordova-приложения сбора данных о состоянии лесных территорий с привязкой к геопозиции. // Известия СПбЛТА — СПб.:СПбГЛТУ, 2019. № 230.
11. Васильев Н.П. Универсальные технологии разработки мобильных приложений. // Информационные системы и технологии: теория и практика: сб. научн. тр. Вып. 10 Ч.1. — СПб.:СПбГЛТУ, 2018. — с.23-30.
12. Васильев Н.П. Гибридные технологии разработки приложений для мобильных платформ. // Информационные системы и технологии: теория и практика: сб. научн. тр. Вып. 9. — СПб.: СПбГЛТУ, 2017. — с.12-21.

**М.Р.Вагизов**, кандидат технических наук, доцент  
Зав. кафедрой информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
bars-tatarin@yandex.ru

**А.А. Гаврилюк**, студентка 2 курса ИСиТ  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С. М. Кирова  
Anna\_gvk@mail.ru

## УПРАВЛЕНИЕ ЛЕСНЫМИ ЗЕМЛЯМИ НА ОСНОВЕ ГЕОИНФОРМАЦИОННЫХ МОДЕЛЕЙ ЛЕСНЫХ ЭКОСИСТЕМ

***Аннотация.** В статье раскрываются основные направления по применению геоинформационных моделей в целях управления лесными землями.*

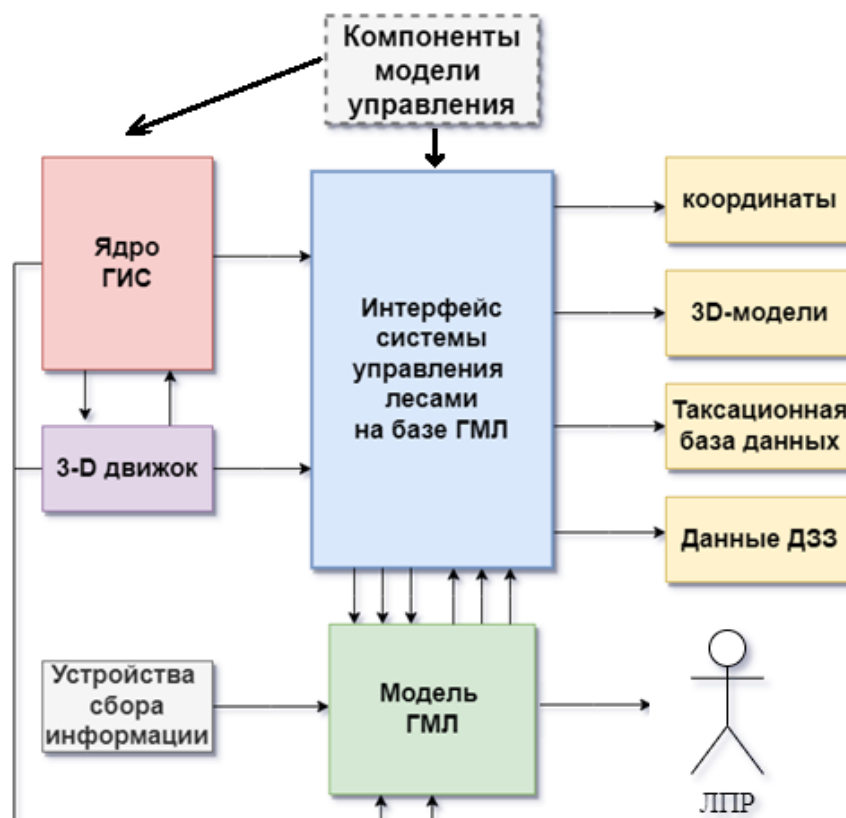
***Ключевые слова:** геоинформационное моделирование, лесные экосистемы, управление лесами.*



Цели управления – это явно и неявно выраженные предпочтения субъекта управления и других заинтересованных лиц. Решение проблемы целеполагания требует соблюдения определенного порядка действий, последовательности взаимосвязанных процедур, направленных на формирование системы целевых установок.[1]

В научных работах [2,3] предложена технология, раскрывающая визуальное представление покрытой лесом земли. С 1 января 2023 года должна заработать федеральная информационная система лесного комплекса. Она призвана обеспечить создание единого цифрового пространства для хранения сведений о лесах, а также для учёта древесины и сделок с ней. Конечно, основное представление такой информации – это достоверные количественные и качественные показатели древесной растительности, однако необходим ещё один инструмент, позволяющий исключить любые сторонние манипуляции, искажающие цифровую информацию и оценки лесных ресурсов. В качестве такого инструмента может выступить технология геоинформационного моделирования леса, интегрированная в состав федеральной информационной системы, фактически являющаяся точной копией лесного участка на основе данных дистанционного зондирования Земли, таксационных данных и трёхмерного отображения лесобразующих пород в программной среде.

Использование такого полезного инструмента позволит получить самое главное для всех заинтересованных лиц процесса управления лесной землей – возможность видеть ресурс управления, знать его состояние оценивать динамику его изменений. Второй важной составляющей является абсолютная понятность отображаемых характеристик, даже не для специалистов лесного профиля, что позволит лучше понимать основные процессы управления лесами. Третья составляющая – это контроль соответствия фактического состояния древесной растительности с его цифровым представлением в моделируемой среде. Данная функция в составе геоинформационной модели позволяет сделать важную практическую составляющую: не просто сформировать абстрактное представление количества лесных ресурсов на картографических материалах, а сформировать такое представление данных, при котором станет невозможным процесс его искажения, поскольку в данном случае процесс отображения материалов связан с их визуальным представлением и привязан к географическим координатам, а также связан с геопространственной информацией - таксационными данными, которые привязаны к каждому дереву. (Рис.1)



**Рис.1.** Способ представления данных основных компонентов ГМЛ  
(ГМЛ - геоинформационная модель леса)

Здесь возникает проблема интеграции разнородных данных. заключается она в том, что таксационные описания составлены на лесотаксационный выдел и представляют собой, достаточно условное отображение древесных пород в той точке сбора информации, в которой находился таксатор в момент её фиксации. При этом не исключён человеческий фактор в ошибках определения точного возраста, диаметра и породного состава лесов при длительных полевых работах и глазомерном способе измерения таксационных показателей насаждений.

Наиболее точным подходом в части методологического обеспечения сбора информации о лесе является подеревная таксация лесов, где учёту подлежит каждое дерево. На сегодняшний день технологии, описанные в работах [1,2], позволяют проводить данные манипуляции. Более того, увеличивается количество различных орбитальных спутников, в том числе спутников Российского производства и спутников специального назначения, что позволяет получать комплексную информацию в части количественных показателей древесного покрова на больших площадях и в труднодоступных территориях.

Так, вполне возможно интегрировать в состав модели координатную привязку к каждому моделируемому дереву (цифровому двойнику), где координаты определяются как точка в центре группы пикселей кроны дерева, после чего отправляются в базу данных ГМЛ. В данной процедуре происходит процесс соотнесения фактического дерева с его цифровым двойником, после чего дереву автоматически присваивается уникальный

ID, который будет его сопровождать с момента фиксации данными ДЗЗ до начала процесса лесопользования, вывозки, складирования и поступления на склад, то есть фактически до момента ликвидации дерева как товара. Для контроля процесса всех компонентов возможно использование технологии блок-чейн, предложенной в работе [6]. В таком случае невозможно внести изменения ни на одном из этапов транспортировки древесины, поскольку в основе технологии блок-чейн лежит такая структура данных программного кода, в которой используется и шифрование, и децентрализованное хранение цепочки блоков. После связывания данных каждого дерева с его цифровой копией в ГМЛ происходит фиксация всех компонентов. Представление данных в таком виде возможно на примере прототипа интерфейса (Рис.2)

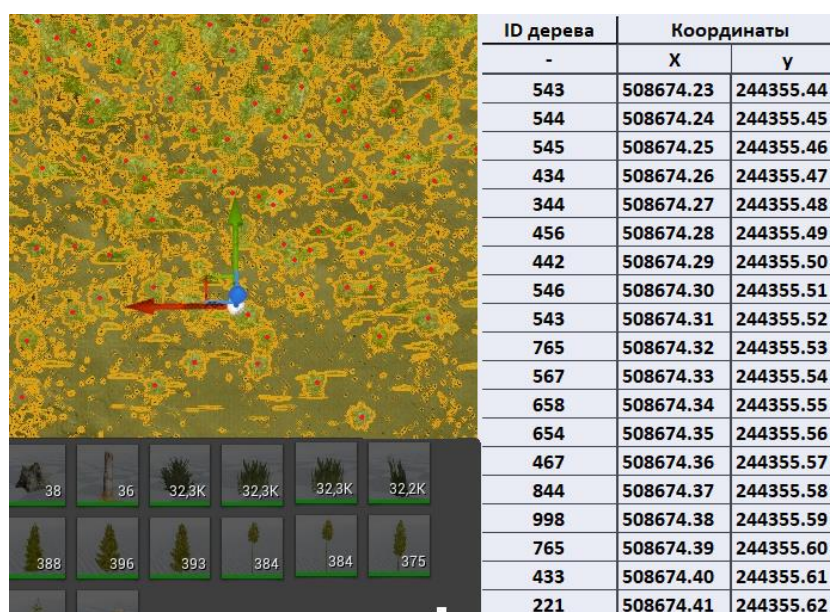


Рис. 2. Прототип отображения разнородных данных в ГМЛ

Последним моментом фиксации данных в ГМЛ является полное прослеживание жизненного цикла леса на корню и его логистического пути при использовании человеком, таким компонентом является компонент визуализации каждого дерева с присвоенным ID. Данный компонент в сочетании с вышеуказанными технологиями ставит точку в вопросе незаконного использования древесины, искажения информации в государственном лесном реестре, намеренного изменения запасов древесины и других искажений численной информации, поскольку геопространственная привязка к координатам в геоинформационной модели отражает не только численные характеристики, но и физическую копию дерева в цифровой среде.

В данном контексте стоит отметить, что термин «геоинформационная модель» отражает привязку к позиции дерева, а вариант его отображения всегда сопряжен с реальной местностью, в которой важными составляющими является отображение ландшафта местности и рельефа, а при необходимости и микрорельефа территории.

В развитии методов сбора информации о лесе может также выступать группировка беспилотных летательных аппаратов, в том числе крупных: «Геоскан Агро» и «Орлан» [4,5], а также аппараты класса микро «DjiMavicmini и DjiAir 2S». Перспективными выглядят технологии использования группы БЛА или роя БЛА. В таком случае необходимы технологии совершенствования траектории полётов БЛА предложенных в работе учёных [7]. Поскольку при необходимости снижения БЛА на удалённых территориях требуется высокая степень автономности аппаратов, а также решение части проблем с питанием устройств, среди которых можно выделить использование технологий заряда на базе солнечных батарей и изготовление целого корпуса БЛА, поглощающего солнечную энергию, впоследствии преобразованную в электрическую.

Управлять лесными ресурсами необходимо с учётом современных технологий (трёхмерного моделирования, инструментом VR/AR технологий, беспроводных сенсорных сетей [8]), поскольку пренебрежение современными технологиями, способными обеспечить поддержку в системах принятия решений, влечёт за собой снижение конечной цели управления и рентабельности лесного хозяйства. Необходимо отметить, что геоинформационная модель леса – это модель природно-технической системы. В этом случае природно-техническая система, представленная в виде модели, должна обладать такими свойствами и инструментами отображения данных, управление которыми, позволит обеспечить развитие лесного хозяйства и эффективность государственной политики в части устойчивого управления лесами. Геоинформационное управление природно-техническими системами, на базе геоинформационных моделей представляет особый вид управления, реализующий целенаправленное воздействие на систему с учетом обработки и визуализации пространственно-временных характеристик леса, для формирования у лица принимающего решение наиболее комплексную позицию о состоянии исследуемой покрытой лесом земли. Жизненный цикл древесных растений изучен в достаточной мере для интеграции всех моделируемых процессов в цифровую среду в качестве вспомогательного инструмента управления покрытыми лесом землями.

### **Библиографический список**

1. Вагизов М.Р., Истомина Е.П., Колбина О.Н., и др. Введение в геоинформационное управление. Учебное пособие // СПб.: НП «БИУ» 2021 352 с.
2. Хитрик Л. А., М. Р. Вагизов Разработка методики геоинформационного моделирования лесного квартала // Актуальные вопросы лесного хозяйства: материалы V международной молодежной научно-практической конференции, Санкт-Петербург, 11–12 ноября 2021 года / Санкт-Петербургский государственный лесотехнический университет имени С.М. Кирова. – Санкт-Петербург: 2021. – С. 191-195.
3. Вагизов М. Р. Цифровое геомоделирование лесов: новый этап анализа данных о лесных экосистемах / М. Р. Вагизов // Сборник научных тру-

- дов Совета молодых ученых СПбГЛТУ. – Санкт-Петербург : 2021. – С. 4-8.
4. Бабашкин Н. М. Исследовательские испытания программно-аппаратных комплексов Геоскан 101 и Геоскан 201 / Н. М. Бабашкин, С. А. Кадничанский, С. С. Нехин // Геодезия и картография. – 2020. – Т. 81. – № 1. – С. 19-25. – DOI 10.22389/0016-7126-2020-955-1-19-25.
  5. Н. А. Карманова, А. Г. Карманов. Разработка модели радара с синтезированной апертурой беспилотного летательного аппарата для дистанционного зондирования лесных массивов // Актуальные вопросы лесного хозяйства: материалы V международной молодежной научно-практической конференции, Санкт-Петербург, 11–12 ноября 2021 года / Санкт-Петербургский государственный лесотехнический университет имени С.М. Кирова. – Санкт-Петербург: 2021. – С. 174-181.
  6. Лобовиков М. А. Блокчейн как инструмент борьбы с незаконным оборотом древесины / М. А. Лобовиков, Н. К. Прядилина, И. Н. Щербак // Цифровые технологии в лесном секторе : Материалы II Всероссийской научно-технической конференции-вебинара, Санкт-Петербург, 18–19 февраля 2021 года. – Санкт-Петербург: Санкт-Петербургский государственный лесотехнический университет имени С.М. Кирова, 2021. – С. 75-78.
  7. Хабаров С. П., Шилкина М. Л. Формирование программных траекторий движения БПЛА с учетом ограничений на их управляемость / // Цифровые технологии в лесном секторе : Материалы II Всероссийской научно-технической конференции-вебинара, Санкт-Петербург, 18–19 февраля 2021 года. – Санкт-Петербург: Санкт-Петербургский государственный лесотехнический университет имени С.М. Кирова, 2021. – С. 141-143.
  8. А. М. Заяц, С. П. Хабаров Подход к моделированию беспроводных сенсорных сетей системы мониторинга лесных территорий // Актуальные проблемы и перспективы развития лесопромышленного комплекса : Материалы IV Международной научно-практической конференции, Кострома, 08–11 сентября 2021 года / Отв. редакторы А.А. Титунин, Т.Н. Вахнина. – Кострома: Костромской государственный университет, 2021. – С. 156-158. – EDN ZFVRSN.

**А.А. Гаврилюк**, студентка 2 курса ИСиТ  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С. М. Кирова  
Anna\_gvk@mail.ru

**А.О. Бутусова**, студентка 2 курса ИСиТ  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С. М. Кирова  
nastbut2008@gmail.com

**А.Ю. Ничипорук**, студентка 2 курса ИСиТ  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С. М. Кирова  
angelinanich@rambler.ru

**С.С. Колмогорова**, кандидат технических наук, доцент  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С. М. Кирова  
ss.kolmogorova@mail.ru

## **РАЗРАБОТКА ПРОГРАММЫ УПРАВЛЕНИЯ И ОЦЕНКА ВОЗМОЖНОСТИ ДВИЖЕНИЯ В ОДНОМЕРНОЙ ПРОЕКЦИИ КВАДРОКОПТЕРА В СРЕДЕ MATLAB**

***Аннотация.** Развитие направления внедрения автономного процесса управления беспилотными воздушными системами устанавливает ряд проблем, связанных с математическим и программным моделированием. В статье представлен процесс моделирования управления движением одномерной проекции квадрокоптера. Реализованы математические функциональные зависимости управления квадрокоптером с учетом физических законов в одномерной реализации, а также программа управления на платформе MATLAB, учитывающая такие влияющие факторы как параметры контроллера высоты, пропорциональное и производное усиления, время стабилизации. Предложен сравнительный анализ по времени стабилизации при разных коэффициентах в рамках моделируемых практических задач.*

***Ключевые слова:** квадрокоптер, беспилотный летательный аппарат, MATLAB, математическая модель, одномерная система координат, поступательное движение, управление высотой.*

### **Введение**

Задача проектирования управления полетом в системе беспилотных воздушных судов (БВС) является важной и не полностью реализованной. Проблема управления квадрокоптером, без участия пилота, решается современными учеными и исследователями. Основная задача, которая ставится перед разработчиками БВС — это сложность программирования микроконтроллеров и реакция БВС на условия применения. Проектирова-

ние управления, реализующее концепцию автономной координации квадрокоптера, в одномерной размерности разработано и представлено в статье.

При проектировании систем управления квадрокоптерами, предварительно авторами разрабатывалась математическая модель, в том числе на основе физических законов. Важно отметить, что программная реализация включает себя параметры не только математической модели, но предполагаемой аппаратной реализации, с конфигурационными характеристиками на примере PID (ПИД) регулятора, для настройки эффективности полета, влияющие на стабильность обработки полета.

Для достижения стабильности и идеальности полета, важно экспериментальное моделирование с подбором значения регулятора в полетном микроконтроллере квадрокоптера, что и позволяет сделать вывод о влияющих параметрах функции P, I и D и других важных коэффициентов, таких как вибрация, ветер, температура и другие климатические условия.

### **Обзор и формулировка проблемы**

Началом истории беспилотных летательных аппаратов (БЛА) можно считать 19 век [1]. И уже в начале 21 века благодаря удешевлению электроники и внедрению компактных навигационных приборов стало возможным создание недорогих и небольших по размеру беспилотных летательных аппаратов. Беспилотные летательные аппараты классифицируются по назначению, типу управления и взлётной массе [2,3].

Для управления беспилотным летательным аппаратом оператору необходимо точно знать положение летательного аппарата в пространстве в определённый момент времени. Для выполнения этого условия каждый беспилотный летательный аппарат оснащён навигационной системой.

Самой распространённой навигационной системой и одной из самых важных считается инерциальная навигационная система. Такой навигационной системой оснащено большинство беспилотных летательных аппаратов, так как она позволяет решить задачу навигации без использования дополнительных устройств. Инерциальная навигационная система вычисляет положение в пространстве беспилотного летательного аппарата через интегрирование его ускорения, определённого с помощью акселерометра и гироскопа (датчика угловых скоростей). Есть также радионавигационные и спутниковые системы навигации. Чаще всего в современных летательных аппаратах комбинированно работают инерциальные и спутниковые навигационные системы.

*Основы системы управления.* Система управления представляет собой неотъемлемую составляющую любой робототехнической системы. Для того, чтобы беспилотный летательный аппарат следовал по заданной траектории и имел возможность совершать определённые контролируемые действия, ему необходима система управления. Управлением называется выполнение совокупности действий, совершённых над управляемым объектом и направленных на достижение поставленной цели [4]. Самыми



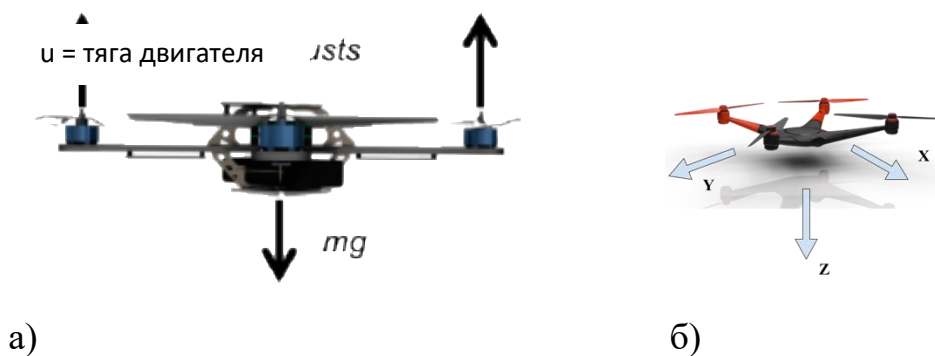
важными задачами, которые выполняет система управления, являются стабилизация, слежение за аппаратом и выполнение заданной программы.

В статье рассматривается задача составления математической модели беспилотного летательного аппарата в одномерной плоскости и его программная реализация с учетом влияющих коэффициентов от производных движения в математическом уравнении БЛА. Также представлены и учтены особенности стабилизации при программировании микроконтроллера в режиме автономного управления БЛА.

## Моделирование

Имитационное моделирование (ИМ) позволяет разработчику быстро и оперативно проверить алгоритмическую реализацию, заранее сформировать систему, позволяет произвести тестирование без искажений в аппаратной части. Существует множества вариантов систем ИМ, используемые под различные цели БЛА. Такие как: моделирование динамики летательного аппарата, проверка системы технического зрения, разработка полёта для группы аппаратов и т.д. Например, моделирование конструкции летательного аппарата, которое производится в таких программах как: SOLIDWORKS, NX - моделирование и проектирование конструкции летательного аппарата; для моделирования различных физических процессов внутри аппарата, например, теплопроводности, используется StarFlow; в MATLAB [5] проводится моделирование движение летательного аппарата [6].

*Математическое моделирование управления квадрокоптером в 1D.* Разрабатываемая математическая модель учитывает поступательного движения летательного аппарата относительно одномерной оси координат 1D – Z, соответственно она же и будет являться стартовой системой координат для БЛА. Поэтому задачей математического моделирования в данном случае является описание набора высоты квадрокоптером вдоль оси z и его стабилизация на предполагаемой высоте (рисунок 1).



**Рис. 1.** а) 1D модель квадрокоптера. б) расположение осей связанной системы координат квадрокоптера (направление в математическом моделировании (направление координатной оси Oz строго вниз (к центру Земли)



Начало связанной системы координат находится в центре тяжести квадрокоптера. Ось  $X$  указывает направление вдоль носовой части и ось  $Y$  направлена вправо, ось  $Z$  направлена вниз, следуя правилу правой руки. При переходе в одномерную систему координат, сохраним направление оси  $Oz$  [5].

Фундаментальной формулой, на которой строится последующее описание математической модели, является основное уравнение динамики – второй закон Ньютона

$$F = m \cdot a, \quad (1)$$

где  $F$  — сила, вызывающая ускорение тела,  $m$  — масса тела,  $a$  — приобретенное телом ускорение.

Для модели, описываемой в статье, формула преобразуется в следующем виде (2):

$$m \cdot \frac{d^2x}{dt^2} = \sum_i F_i \quad (2)$$

где:

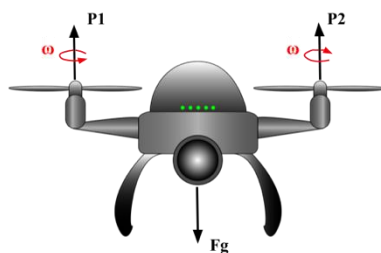
$m$  – масса системы,

$x$  – положение системы,

$t$  – время, за которое происходит перемещение,

$F$  – сумма всех сил, действующих на систему.

Рассмотрим подробнее силы, действующие на обобщённую модель квадрокоптера (рисунок 2):



**Рис. 2.** Графическое изображение сил, действующих на квадрокоптер

Силы  $P_1$  и  $P_2$  - силы тяги двигателей квадрокоптера, силы рассчитываются по формуле (2):

$$P = b \cdot \omega^2 \quad (3)$$

где:

$P$  – сила тяги двигателя,

$b$  – коэффициент тяги,

$\omega$  - угловая скорость.

В системе 1D, где основное уравнение динамики (1) с учетом (2) будет выглядеть следующим образом:

$$\sum_i F_i = \sum_i P_i + mg = m \frac{d^2x}{dt^2} \quad (4)$$

где:  $mg$  – это сила тяжести  $Fg$ , а  $m \frac{d^2x}{dt^2}$  – вторая производная положения по времени, являющаяся ускорением,  $P_i$  – сумма тяг всех двигателей.

Так как движение аппарата производится вдоль одной оси Oz, запишем уравнение динамики в скалярном виде:

$$m \frac{d^2 x}{dt^2} = \sum_i b \omega_i + mg \quad (5)$$

Скорость есть интеграл по времени от функции ускорения. Проинтегрируем уравнение (5), получим:

$$m dx = \int (\sum_i b \omega_i + mg) dt \quad (6)$$

Положение в пространстве есть интеграл от скорости, следовательно, двойной интеграл от ускорения. Чтобы получить положение системы проинтегрируем полученное выражение (6) и получим выражение (7):

$$m x = \int \int (\sum_i b \omega_i + mg) dt dt \quad (7)$$

Чтобы получить координату квадрокоптера, следует уравнение (7) разделить на массу БЛА.

Моделирование динамики движения беспилотного летательного аппарата в одномерной системе координат заключается в получении приращения скорости и положения аппарата. Далее необходимо решить дифференциальное уравнение любым удобным способом, например: методом Эйлера, то есть проинтегрировать его и получить нужный результат.

Вышеизложенные уравнения используются в программе-симуляторе для моделирования динамики БЛА.

*Система управления высотой.* Задачей управления является минимизация ошибки системы, а именно уменьшение разности между текущим и рассматриваемым в качестве целевого параметрами системы. При рассмотрении алгоритма расположим систему координат привычным образом, ось Oz направлена вверх.

Задачей является удержание квадрокоптера в воздухе. В качестве решения предложен поэтапный алгоритм для системы управления высотой беспилотного летательного аппарата:

1. Вычисление ошибки проводится по формуле (8):

$$e_j = x_{tar} - x_{j-1} \quad (8)$$

где  $e_j$  – ошибка по положению на  $j$ -той итерации,  $x_{tar}$  – целевое положение аппарата,  $x_{j-1}$  – положение квадрокоптера на прошлой итерации.

2. Расчет управляющего воздействия исходя из пропорционального интегрально-дифференцирующего закона регулирования:

$$u_j = K_p e_j + K_i \int_{t_0}^T e_i dt + K_d \frac{e_j - e_{j-1}}{dt} \quad (9)$$

где  $u_j$  – управляющее воздействие, созданное ПИД-регулятором на  $j$ -той итерации,  $K_p$ ,  $K_i$ ,  $K_d$  – коэффициенты ПИД-регулятора,  $dt$  – шаг интегрирования.

Принцип работы ПИД-регулятора представлен на рисунке 4:

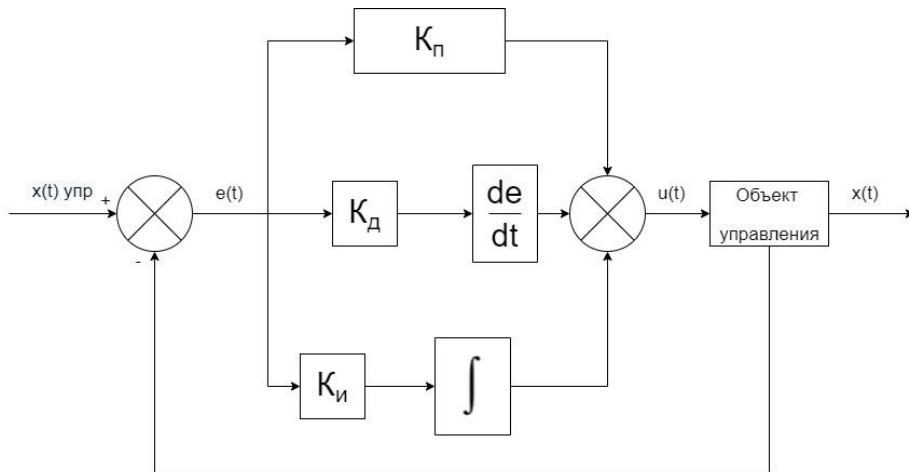


Рис. 4. Математическая блок-схема работы ПИД-регулятора

3. Производится замена угловых скоростей двигателя управляющим воздействием, так как управление угловой скоростью летательного аппарата нацелено на движение в направлении целевого положения.

$$\omega_i = u_j, \quad (10)$$

4. Суммирование всех сил, действующих на систему:

$$F_{sum} = b \sum_{j=1}^n \omega_j^2 - F_g, \quad (11)$$

где  $b$  - коэффициент тяги двигателя

5. Вычисление скорости летательного аппарата:

$$V_j = V_{j-1} + \frac{F_{sum}}{m} dt \quad (12)$$

6. Определение положения системы на каждой итерации:

$$x_j = x_{j-1} + V_j dt, \quad (13)$$

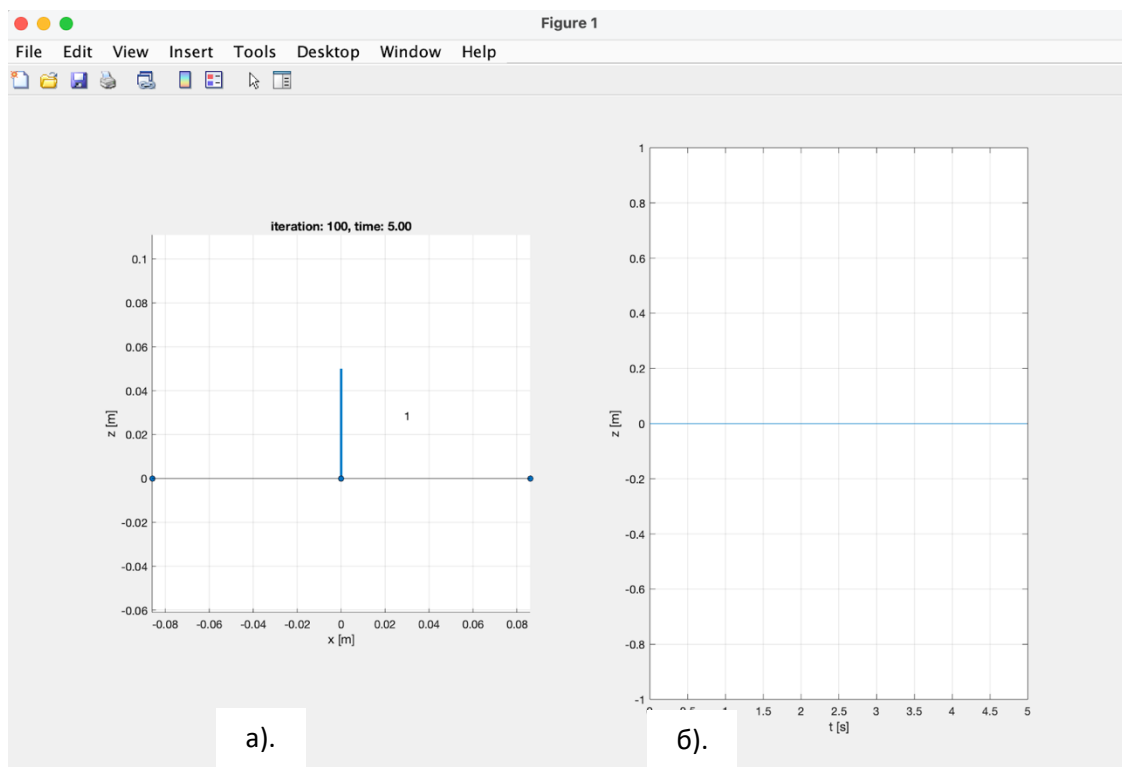
Полученный алгоритм работает по итерациям от 0 до определенной конечной операции с шагом  $dt$ .

В разработанной авторами математической модели используются выражения и формулы (1-13) для моделирования динамики движения квадрокоптера, то есть совершать управляемой движение. Полученные координаты БВС используются в системе управления для создания управляющего воздействия, используемого для расчета новых данных и влияющих факторов.

*Моделирование в среде Matlab.* Для оценки математической модели необходимо построить графики зависимости требуемого положения системы от времени. На графике (рисунок 5) представлена стабилизация квадрокоптера на высоте 0. На рисунке 5а представлено 100 итераций анимационного квадрокоптера по оси  $z$  относительно оси  $x$  в метрах за время  $t$ . На рисунке 5б части показана проверка удержания высоты на нулевом уровне на всем промежутке времени.

На рисунке 6 показан график шага квадрокоптера в 1 метр. То есть квадрокоптер поднимается на высоту 1 метр. Результат на этот ввод имеет нарастание менее одной секунды и максимальное превышение менее 5%.

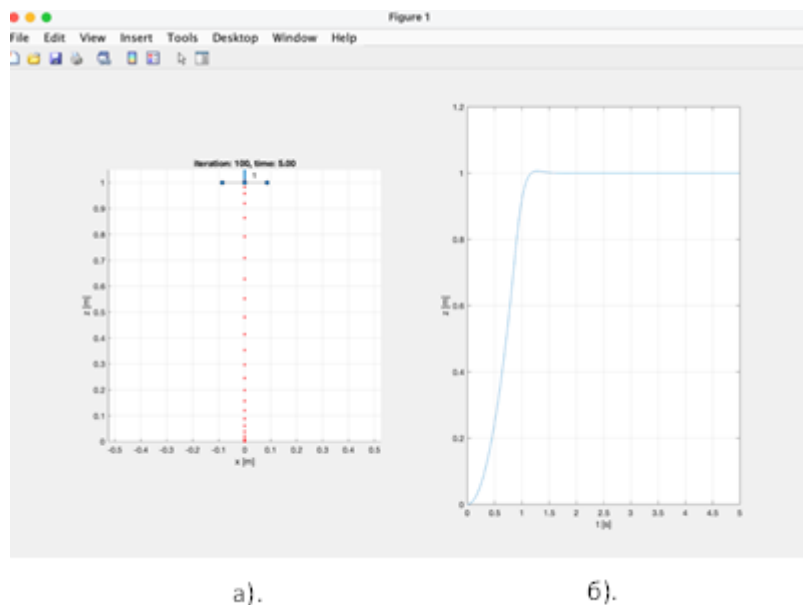
Время нарастания – это время, необходимое для достижения 90% от установившегося значения, поэтому в этом случае квадрокоптер достигает 0,9 метра менее, чем за одну секунду.



**Рис. 5.** График стабилизации квадрокоптера на одном уровне

Набор файлов в моделируемом программном коде имеет несколько уровней: универсальные файлы нижнего уровня, которые могут применяться для всех режимов (это файлы математических преобразований углов, алгоритмов работы с передатчиком и двигателями и т.д.), и узкоспециализированные файлы (в частности, файл контроллера стабилизации положения), которые опираются на ряд файлов низшего уровня. На рисунке 7а представлен текстовый скрипт, который будет вызван для тестирования полета квадрокоптера с заданием шага. В процессе разработки алгоритма БВС определяем параметры конструкции, влияющие непосредственно на динамику аппарата: массу, моменты инерции, положение центра масс и т.д. На рисунке 7б показан программный код моделирования контроллера БВС, в котором описан алгоритм управления поведением квадрокоптера в процессе полета, использующий заданные входные параметры. В данном случае представлен код стабилизации квадрокоптера на высоте 0. На рисунке 8а показан код, после запуска которого квадрокоптер будет подниматься на высоту 1 метр, на рисунке 8б представляет код, который будет вызван для генерации отправленных файлов, где представлены входные значения. Важно отметить важные введенные коэффициенты:  $K_v$  - коэффициент производного усиления дифференцирующей составляющей ПИД-регулятора. Данный коэффициент влияет на колебания системы. При его

увеличении квадрокоптером плавнее достигается целевое значение, при подлёте к заданной точке БЛА замедляет скорость и стабилизируется на заданной высоте практически без колебаний.  $K_p$  - коэффициент пропорциональной составляющей регулятора. Данный коэффициент влияет на скорость достижения заданной высоты квадрокоптером. Чем больше коэффициент, тем быстрее аппарат достигнет заданную высоту. Значение  $K_p$  должно быть положительным.



**Рис. 6.** График тестирования полета квадрокоптера

```

close all;
clear;

% Целевое положение
z_targ = 1;

% Генератор данной траектории
pathhandle = @(t) fixed_set_point(t, z_targ);

% Контроллер
controlhandle = @controller;

% Запуск тестирования с заданным генератором траектории и контроллером
[t, z] = height_control(pathhandle, controlhandle);

```

а).

```

function [ u ] = pd_controller(~, s, s_targ, params)
%PD_CONTROLLER PD-контроллер высоты
%
% s: вектор 2x1 текущего состояния [z; v_z]
% s_targ: вектор 2x1 желаемого состояния [z; v_z]
% params: параметры робота

u=0;

end

```

б).

**Рис. 7.** а). Скрипт тестирования полета квадрокоптера с заданием шага б). Скрипт контроллера БВС на стабилизацию уровня высоты

```

function [ u ] = pd_controller(~, s, s_targ, params)
%PD_CONTROLLER PD-контроллер высоты
%
% s: вектор 2x1 текущего состояния [z; v_z]
% s_targ: вектор 2x1 желаемого состояния [z; v_z]
% params: параметры робота

zdot_targ = 0;
e = s_targ(1) - s(1);
edot = s_targ(2) - s(2);
Kp = 100; %пропорциональное усиление
Kv = 30; %производное усиление(16)
u = params.mass*(zdot_targ + Kp*e + Kv*edot + params.gravity);

end

```

а).

```

close all;
clear;

% Контроллер
controlhandle = @controller;

evaluate(controlhandle);

```

б).

**Рис. 8.** а). Программный код запуска квадрокоптера на высоту 1 м. б). Генерация от-  
правленных файлов на представление входных значений.

```

function [t_out, z_out] = height_control(pathhandle, controlhandle)

addpath('utils');
video = false;
video_filename = 'height_control.avi';

params = sys_params;

% real-time
real_time = true;

%% ===== FIGURES =====
disp('Initializing figures...')
if video
    video_writer = VideoWriter(video_filename, 'Uncompressed AVI');
    open(video_writer);
end
h_fig = figure;
sz = [1000 600]; % figure size
screensize = get(0, 'ScreenSize');
xpos = ceil((screensize(3)-sz(1))/2); % center the figure on the screen horizontally

```

а).

```

function [ s_targ ] = fixed_set_point(t, z_targ)
%FIXED_SET_POINT

if t == 0
    s_targ = [0;0];
else
    s_targ = [z_targ;0];
end

end

```

б).

**Рис. 9.** а). Программный код, реализующий функции контроля высоты БВС. б).

На рисунке 9а представлен код моделирования (функция контроля высоты), а на рисунке 9б представлена функция, которая выводит постоянное желаемое состояние, за исключением, когда  $t = 0$ , где возвращается в точку  $[0;0]$ . Иными словами – это функция результата алгоритма с шагом  $t$ . Также в комплексе программ обязателен вспомогательный файл, в котором реализуются вспомогательные функции для проведения тестирования полета квадрокоптера.

## Результаты

В процессе исследования проведено моделирование нескольких реализация программного кода на разных начальных условиях и категориальных коэффициентах.

В первом тестовом примере рисунок 10 квадрокоптеру нужно стабилизироваться на высоте 0. Коэффициент управляющего воздействия  $u = 0$ . Стабилизация на нулевой высоте реализована в связи с необходимостью определения начального положения БВС в пространстве. На рисунке 11 квадрокоптер поднимается на высоту 1 метр. Коэффициент  $u = \text{params.mass} * (z2\text{dot\_des} + K_p * e + K_v * e\text{dot} + \text{params.gravity}); (z2\text{dot\_des} = 0)$ . На рисунке 12 продемонстрировано увеличение шага квадрокоптера (на рисунке ниже представлен шаг – 2 метра). Соответственно с увеличением высоты полета увеличивается и скорость БВС. На рисунке 13 представлен полет квадрокоптера с заданным шагом – 5 метров. Цель состоит в том, чтобы наш управление контроллеров давал желаемый ответ для каждого теста.

В таблице 1 представлена сравнительная характеристика поведения модели квадрокоптера на разных высотах. Моделирование динамику посредством интегрирования уравнения движения возможно в программе Matlab. Для моделирования системы управления использовалась система уравнений управления движения БВС. Стоит отметить, что система управления, настроенная для конкретного квадрокоптера и решения определенных задач, моделировалась с учетом специфики аппарата.

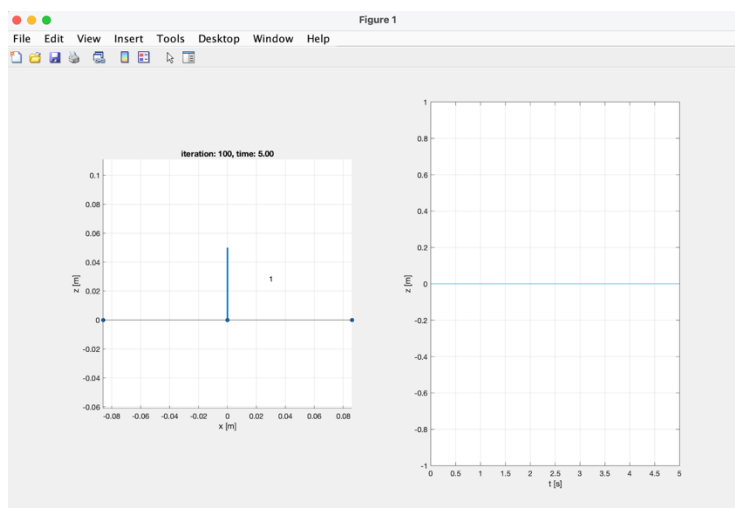


Рис. 10. Тестовый пример стабилизации на одном уровне

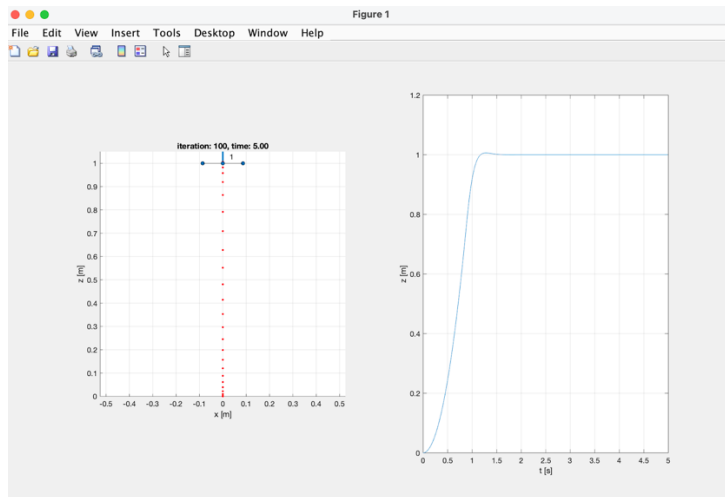


Рис. 11. Тестовый пример стабилизации на высоте полета - 1 метр

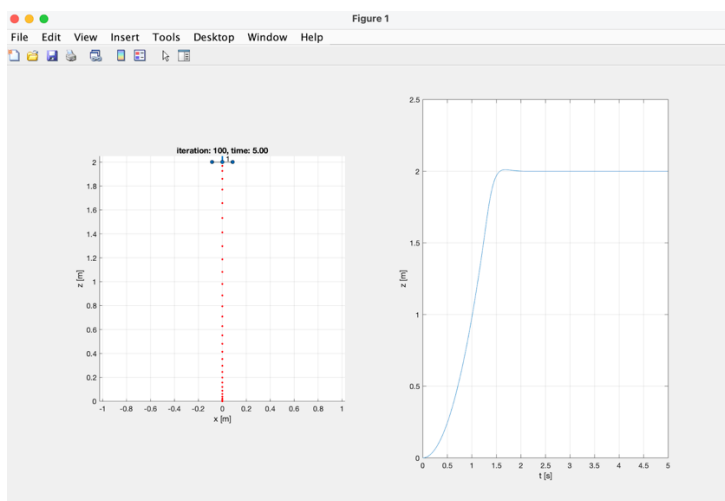


Рис. 12. Тестовый пример стабилизации на высоте полета - 2 метр

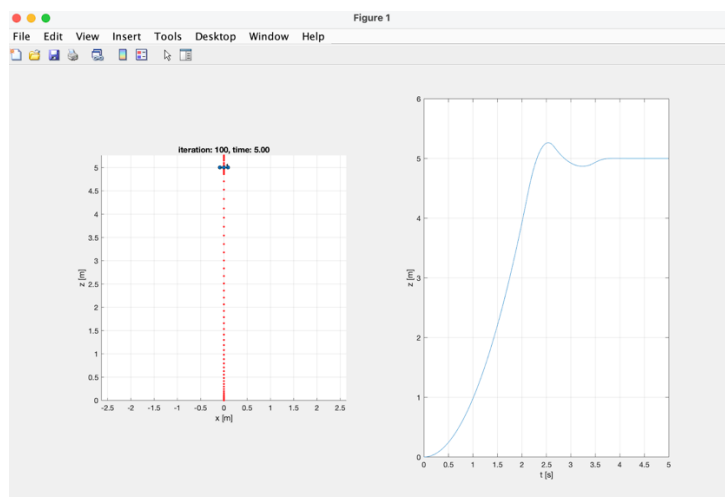


Рис. 13. Тестовый пример стабилизации на высоте полета - 5 метров



**Сравнительная характеристика поведения модели квадрокоптера на разных высотах**

Заданная высота (шаг)	1 м.	2 м.	5 м.	7 м.	10 м.	15 м.
Верхнее отклонение от заданной высоты	незначительное колебание		5.3 м.	7.5 м.	10.9 м.	16.6 м.
Нижнее отклонение от заданной высоты			4.9 м.	6.4 м.	8.4 м.	11 м.
Время первичного достижения заданной высоты	1.2 сек	1.5 сек	3.8 сек	2.7 сек	3.2 сек	3.9 сек
Время стабилизации	стабилизация модели квадрокоптера за 0,4 сек.		1.5 секунды	2.45 секунд	3.58 секунды	5.9 секунды

### Выводы и заключение

В статье реализованы задачи математического и программного моделирования поведения квадрокоптера в размерности одномерной плоскости. В качестве инструмента для выполнения задачи проекта сформирована программа в среде MATLAB, в результате выполнения которой построены графики требуемого положения системы в зависимости от времени и стабилизации высоты. Рассмотрены основные варианты задач, возможные в одномерной плоскости, среди которых рассмотрены такие случаи, как: подъем аппарата на определенную высоту вдоль вертикальной оси за заданный промежуток времени и его стабилизация на одной высоте. Данные графики подтверждают работоспособность разработанной математической модели поведения квадрокоптера в одномерной плоскости.

Универсальное значение ПИД - коэффициентов для задачи подъёма БВС на заданную высоту подобрать возможно с определёнными сложностями. Так как для каждого значения высоты собственные оптимальные значения коэффициентов. Исходя из результатов тестирования, в качестве примера, приводятся оптимальные значения коэффициентов для наиболее эффективного выполнения задачи подъёма БВС на высоту 5 метров:  $K_p = 100$ ,  $K_v = 30$ .

Таким образом, представленный в статье вычислительный алгоритм имеет возможность управлять поведением летательного аппарата в процессе полета в разных условиях.

В заключении важно отметить, что проектирование управления полетом квадрокоптера – одно из наиболее сложных, важных, перспективных и быстроразвивающихся направлений робототехники. Во всем мире стремительную популярность набирают беспилотные технологии, наиболее

распространённой из которых и является полеты квадрокоптеров. Беспилотные технологии — это уже не будущее, а настоящее, которое следует изучать и совершенствовать с высокоточным и безопасным полетным управлением.

### Библиографический список

1. История развития дронов [Электронный ресурс]. – Режим доступа: <https://dronomania.ru/faq/istoriya-razvitiya-dronov.html> (Дата обращения 07.01.2021)
2. Беспилотные авиационные системы. Циркуляр 328. – ИКАО. – 2011. Р. 66.
3. ГОСТ Р 59517-2021 Беспилотные авиационные системы. Классификация и категоризация. М., 2021. 12 с. (Система стандартов по информ., библи. и изд. делу).
4. “Программирование дронов” [Электронный ресурс]. – Режим доступа: <https://skillbox.ru/course/drone-programming/> (Дата обращения 18.01.2021)
5. Пакет прикладных программ “MATLAB” [Электронный ресурс]. – Режим доступа: <https://ww2.mathworks.cn/en/products/matlab.html> (Дата обращения 22.01.2021)
6. Modelling and control of quadcopter/ Teppo Luukkonen/School of Science Mat-2.4108/ Independent research project in applied mathematics/ 22.08. 2011
7. AIRCRAFT FLIGHT PATH PREDICTION UNDER COMPLETE PARAMETRIC UNCERTAINTY [Electronic resource] / V. V. Kosyanchuk [et al.] // IZVESTIYA SFedU. ENGINEERING SCIENCES. — 2021. — № 1. — P. 101-112.
8. M. W. Mueller and R. DAndrea, “Stability and control of a quadcopter despite the complete loss of one, two, or three propellers,” in IEEE International Conference on Robotics and Automation (ICRA), 2014.
9. M. Saska, V. Vonasek, J. Chudoba, J. Thomas, G. Loianno, and V. Kumar, “Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles,” *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1, pp. 469–492, 2016.
10. T. Ryan and H. J. Kim, “Probabilistic correspondence in video sequences for efficient state estimation and autonomous flight,” *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 99–112, Feb 2016.
11. M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza, “Autonomous, vision-based flight and live dense 3d mapping with a quadrotor micro aerial vehicle,” *Journal of Field Robotics*, vol. 33, no. 4, pp. 431–450, 2016.
12. J.-J. E. Slotine, W. Li, et al., *Applied nonlinear control*. prentice-Hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.

## КОНЦЕПЦИЯ WEB 3.0 И ЕЁ СОВРЕМЕННАЯ РЕАЛИЗАЦИЯ, ОТЛИЧИЯ ОТ WEB 1.0 И WEB 2.0

*Аннотация.* Данная статья о концепции web 3.0 – концепции, которая в ближайшем будущем может изменить весь интернет.

*Ключевые слова:* web 3.0, эволюция интнета, технология блокчейн.

### Актуальность

Термин web 3.0 был придуман в 2014 году соучредителем Ethereum Гэвином Вудом, и с тех пор интерес к технологии понемногу рос – и в последние пару лет он начал резко расти. На рисунке 1 можно видеть количество разработчиков, работающих над технологиями, реализующими данную концепцию. Этот график – фрагмент отчета ElectricCapital, который основан на активности разработчиков на GitHub.

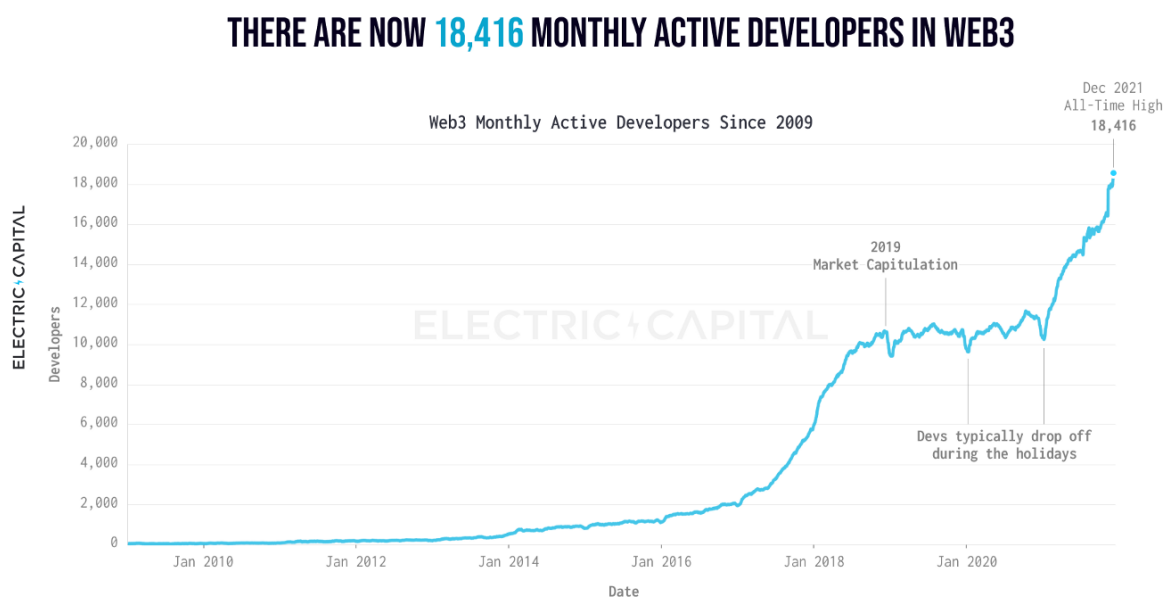


Рис. 1. Исследование ElectricCapital

Это исследование показывает, что интерес к данной теме растет. Более того, многое говорит о том, что интерес будет расти и дальше – ведь web 3.0 решает многие технические проблемы от отображения данных, до надёжности передачи информации и предоставляет много новых возможностей. Использование технологии web 3.0 будет востребовано и при раз-

работке пользовательских web-приложений для задач лесной и природоохранной деятельности. [4]

## Эволюция интернета: web 1.0, web 2.0

Чтобы объяснить, что такое web 3.0, стоит сначала рассказать про концепции web 1.0 и web 2.0.

На заре интернета, когда браузеры только набирали популярность, веб-страницы являлись простыми html страницами без какой-либо интерактивности, объединенные между собой гиперссылками. В то время только администраторы этих страниц могли добавлять или изменять содержимое сайтов, а обычные пользователи - только просматривать это содержимое. Такое состояние интернета и описывает концепция web 1.0.

Web 2.0 же описывает то состояние, в котором интернет находится сейчас: теперь на сайтах существует интерактивность, и часто не администратор сайта, а сам пользователь является автором контента. Это было обеспечено с помощью клиент-серверной архитектуры. Данные от пользователей отправляются на сервер, затем они сохраняются в базу данных.

Визуализацию данных концепций можно видеть на рисунке 2.

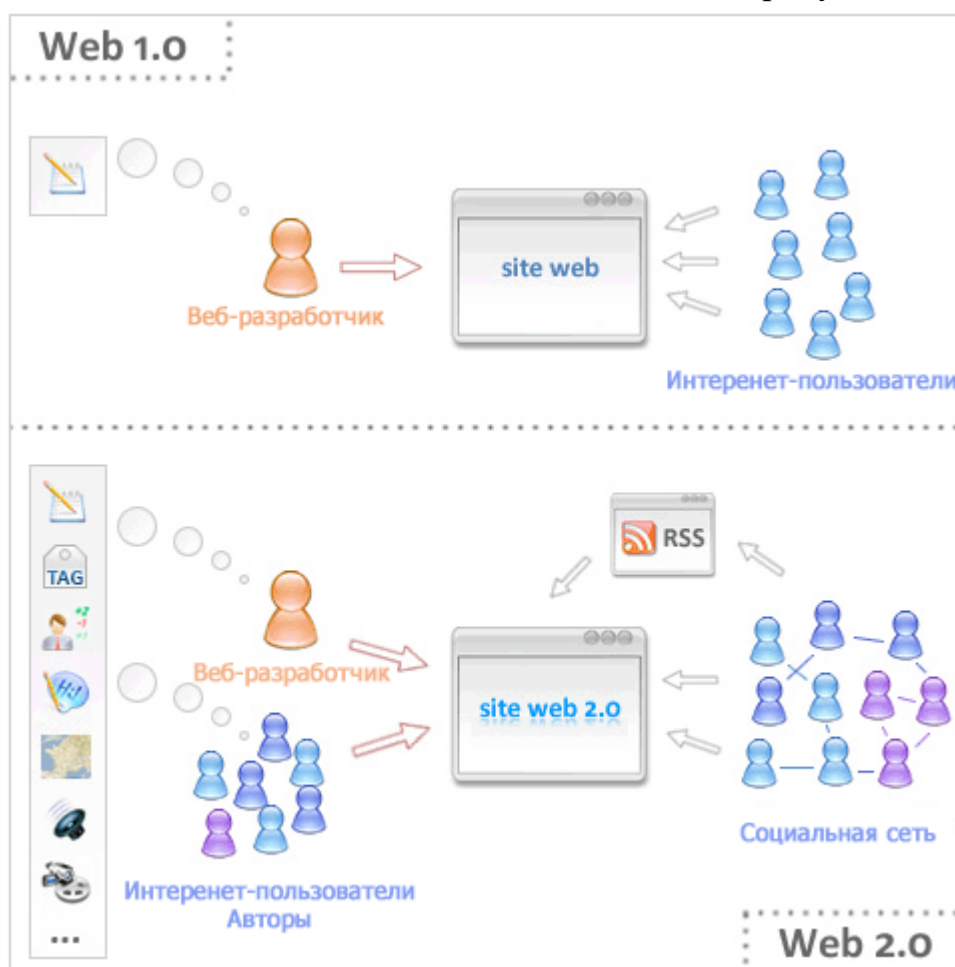


Рис. 2. Различия web 1.0 и web 2.0

Однако, зачастую вместе с полезными данными приложения собирают различные аналитические данные, и это является проблемой безопасности. Также, проблемой является то, что компания, владеющая сервером, может управлять контентом и доступом пользователей, и, например, заблокировать тех, кто по каким-то причинам ее не устраивает. Эти проблемы и призвана решить концепция web 3.0.

### Web 3.0

Данная концепция предполагает создание приложений не по клиент-серверной архитектуре. Вместо того, чтобы взаимодействовать с сервером, пользователи должны взаимодействовать друг с другом в общей мировой сети, построенной на основе блокчейн технологий (Рис. 3).

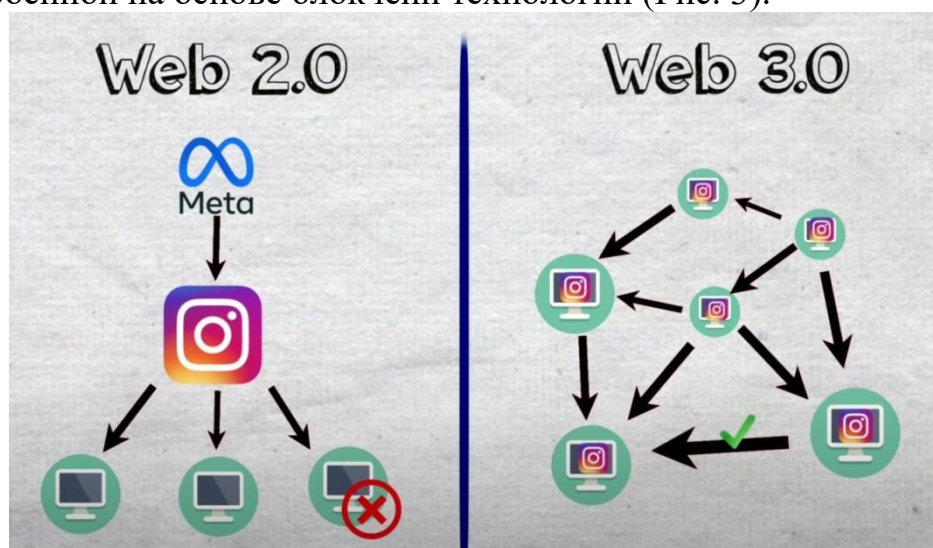


Рис. 3. Различия архитектур web 2.0 и web 3.0

Для дальнейшего пояснения, необходимо рассказать о технологии блокчейн (цепочке блоков). Эта технология, частично является, базой данных и заключается в следующих преимуществах:

- все действия всех пользователей блокчейна записываются в единый файл (блок);
- когда файл начинает превышать определенный объем (например, 1МБ), он считается заполненным и ему присваивает его хэш - адрес, по которому этот файл можно будет найти в блокчейне;
- блокчейн создает новый файл, и в нем указывается хэш предыдущего блока, после чего он продолжает писать данные в этот файл, и затем процесс повторяется.

Таким образом, через последний созданный файл можно просмотреть всю цепочку файлов.

Хэш блока высчитывается по его содержимому, изменение содержания изменит его хэш, следовательно, придется поменять хэш в следующем блоке, это повлечёт изменение и его хэша, и в итоге придется изменить хэши всех последующих блоков. Таким образом, легко достигается неизменяемость - достаточно просто хранить хэш последнего готового

блока, и будет понятно, изменил кто-то или что-то в конкретной базе данных или нет.

Но одного факта использования такой базы данных недостаточно для решения проблем web 2.0. Следующая ступень - децентрализация, и она также реализуется с помощью технологии блокчейн.

Чтобы обеспечить децентрализацию, применяется следующая схема: блокчейн выкладывается в открытый доступ, и любой желающий может его скачать. Таким образом, если скачать его на несколько компьютеров, все они будут иметь одинаковый набор данных. И все желающие, выкачавшие его, будут участвовать в процессе подтверждения запросов, с помощью чего и обеспечится децентрализация.

Этот процесс подтверждения запросов работает таким образом: если что-то необходимо записать в блокчейн (например, информацию о новом сообщении от пользователя), опрашиваются все участники блокчейна. Если в ходе этого опроса хотя бы на 51% всех машин выяснится, что пользователь не может совершить данную операцию (исходя из информации в блокчейне), то операция будет отклонена. В противном же случае она будет занесена на все прочие машины.

Все участники, предоставляющие для этого вычислительные мощности, будут представлять собой администраторов блокчейна, и смогут влиять на принимаемые решения (например, о блокировке кого то или удалении чего то). При этом, любой желающий сможет стать участником, и каждое решение будет принимать большое количество людей. Достигается свойство демократии в интернете.

Кроме этого, вся информация будет храниться в зашифрованном виде, и доступ к ней будет только у того пользователя, кому эта информация принадлежит. Доступ будет осуществляться с помощью приватного ключа, которым является пароль пользователя.

### **Примеры web 3.0.**

Какими же будут приложения в web 3.0? На сегодняшний день, уже несколько приложений используют данную концепцию. Чтобы ими воспользоваться, вам не нужен особый браузер. Внешне все выглядит так же как в обычных клиент - серверных приложениях. Внутренне, существуют технологические отличия.

На рисунке 4 можно видеть некоторые примеры уже реализованных приложений в различных сферах.



Рис. 4. Примеры приложений на web 3.0 в различных сферах

### Как можно построить приложение на web 3.0

Концепцией web 3.0 может воспользоваться каждый - в том числе, и разработчик. Чтобы построить свое приложение, входящее в данную концепцию, можно создать свой блокчейн, или воспользоваться существующим.

Некоторые разработчики баз данных уже разработали блокчейн базу данных, которую можно развернуть у себя, например, MongoDB. Но еще проще использовать уже существующий блокчейн, и самое популярное решение на данный момент - блокчейн Ethereum. Совершенно бесплатно можно начать хранить там данные своего приложения.

### Отрицательные стороны

В заключении стоит указать на минусы технологии. Со временем, использования приложений, объем данных в блокчейне будет расти. В итоге поддерживать его будет все сложнее - и весь его уже не будет возможности выкачать себе, например, на смартфон. Через какое то время и пары серверных машин не хватит - и обычный пользователь уже не сможет стать участником сети, просто выкачав его себе на ПК. Конечно, он сможет пользоваться приложением, и иметь полную анонимность, но при этом он не сможет влиять на работу сети, принимать участие в голосованиях по принятию каких либо решений для всей сети. В итоге, полной децентрализации будет сложно достичь.

Также, такая система потребляет много электроэнергии - на данный момент, блокчейн Ethereum потребляет такое же количество энергии, сколько потребляет Сербия.

Данные проблемы, возможно, решить, и над этим уже трудятся разработчики, но пока технологии не идеальны.

## **Заключение**

Несмотря на то, что технологии для концепции web 3.0 еще недостаточно эффективны, энтузиасты уже создают в ней свои приложения. Со временем технологии будут развиваться, и это сможет привести к значительному упрощению нашей жизни - не нужно будет оформлять множество документов, чтобы купить себе дом, например. Все это сможет произойти автоматически, информация о владельце дома будет записана в блокчейн.

## **Библиографический список**

1. Shermin Voshmgir. Token Economy: How the Web3 reinvents the Internet // BlockchainHub Berlin (June 5, 2020), pp 1-360
2. Laura Shin. The Cryptopians: Idealism, Greed, Lies, and the Making of the First Big Cryptocurrency Craze // PublicAffairs (February 22, 2022), pp 1-497
3. Brad Stone. The Everything Store // Little, Brown and Company; Illustrated edition (October 15, 2013), pp 1-386
4. Никитина А. Д., М. Р. Вагизов Прототип web-приложения «Линдуловская роща - жемчужина отечественного лесокультурного дела» / Актуальные вопросы лесного хозяйства : материалы V международной молодежной научно-практической конференции, Санкт-Петербург, 11–12 ноября 2021 года / – Санкт-Петербург: Санкт-Петербургский государственный лесотехнический университет имени С.М. Кирова, 2021. – С. 162-165.

**А.М. Заяц**, кандидат технических наук, профессор  
Кафедра информационных систем и технологий  
СПб ГЛТУ им. С.М.Кирова  
zamfta@yandex.ru

## **ГРАФОВЫЕ МОДЕЛИ В АНАЛИЗЕ И СИНТЕЗЕ БЕСПРОВОДНЫХ СЕНСОРНЫХ СЕТЕЙ**

***Аннотация.** Рассматриваются графовые математические модели для исследования беспроводных сенсорных сетей на этапах анализа для получения количественных оценок их параметров и синтеза для определения топологии и метрик сетей до их реального развертывания.*

***Ключевые слова:** беспроводные сенсорные сети, графовые модели, технология digimesh, конфигурация сети.*



Исследование беспроводных сенсорных сетей (БСС) с целью получения количественных оценок их параметров на этапе анализа и определения топологии и метрик на этапе синтеза, до процесса реального развертывания сети, можно провести на математических графовых моделях.

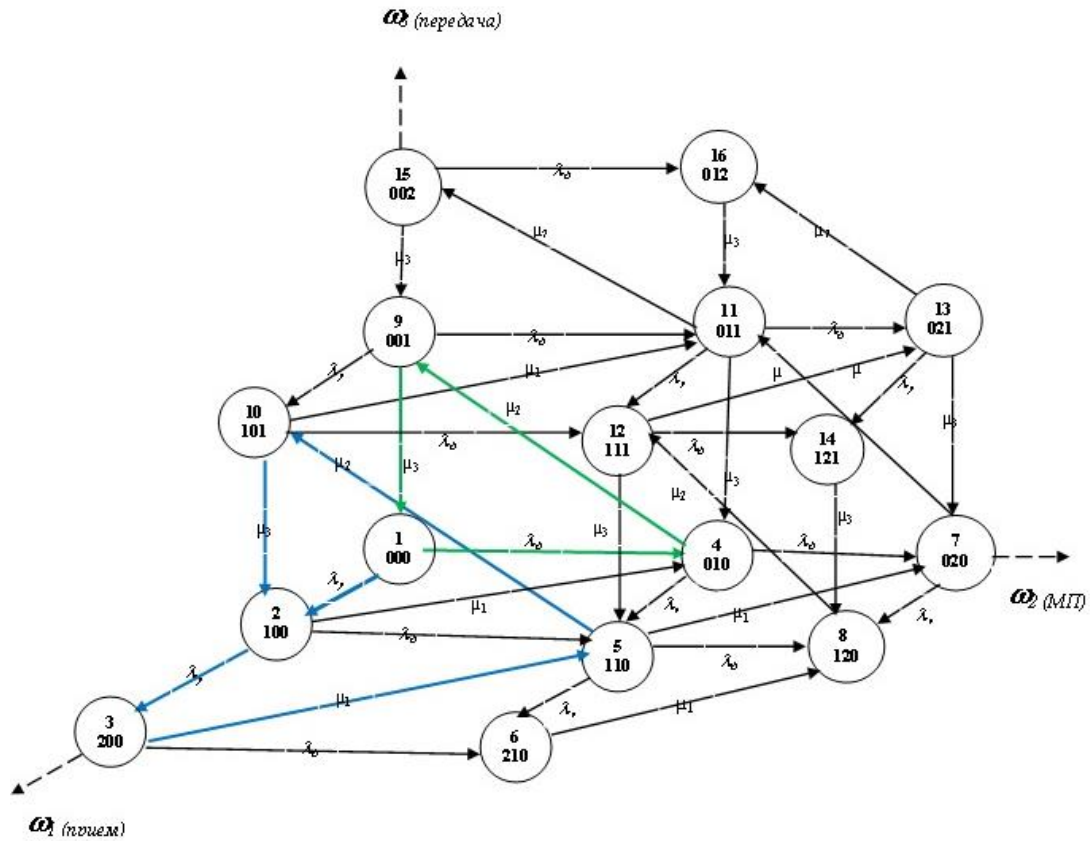
Получение оценок пропускной способности, загрузки узлов и надежности доставки сообщений в ходе анализа, можно провести на математической модели функционирования беспроводной сенсорной сети в составе системы мониторинга лесных территорий, возможные состояния и переходы в которой описываются ориентированным графом, подробно рассмотрены в работах [1-3].

В этой модели состояния процесса обслуживания заявок различаются по количеству обслуживающих приборов (приемопередатчики и микропроцессоры узлов), одновременно занятых обслуживанием и по номеру этапа, на котором находится каждая обслуживаемая заявка (под этапами понимается нахождение заявки на соответствующем обслуживающем приборе).

Граф модели (рис.1) представляет процесс обслуживания заявок в сети digimesh, каждый из узлов которой обладает полной функциональностью и может решать как задачи сбора, так и маршрутизации при доставке пакетов на шлюз.

Каждому состоянию  $e_\xi$  процесса из множества  $E = \{e_\xi\}_{\xi=1}^E$  состояний поставлен в соответствие вектор занятости  $\omega_\xi = (\omega_{1\xi}, \omega_{2\xi}, \omega_{3\xi})$ , каждый компонент  $\omega_{g\xi}$  которого является случайной целочисленной величиной, определяющей количество приборов, одновременно занятых обслуживанием заявок, а индекс  $g$  соответствует этапу, на котором она обслуживается.

Переходы в модели эволюционируют как, под воздействием поступающих заявок ретранслируемых пакетов интенсивностью  $\lambda_r$  и заявок от датчика собственных узлов с интенсивностью  $\lambda_o$ , так и в соответствии с процессами их обслуживания в узлах сети: приемниками с параметром  $\mu_1$ , микропроцессором (МП) с параметром  $\mu_2$  и передатчиком с параметром  $\mu_3$ .



**Рис. 1.** Ориентированный граф процесса функционирования беспроводной сенсорной сети

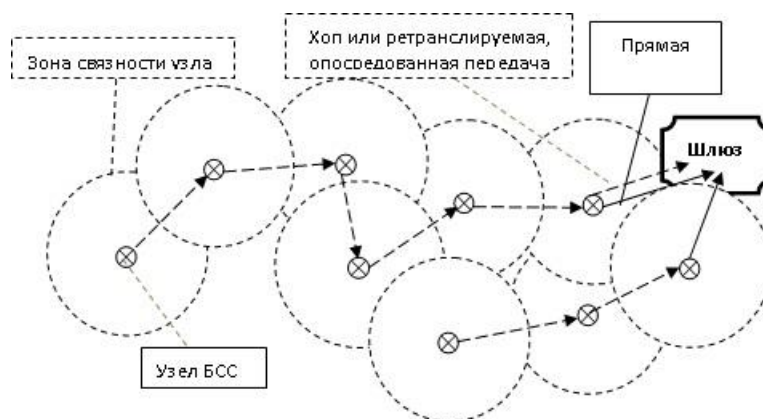
Процесс обслуживания заявок представляется как блуждание точки по узлам  $E$  – мерной решетки, пространственные координаты состояний которой задаются компонентами соответствующих векторов  $\omega_{g\xi}$ .

Для примера на рисунке 1 представлен граф для БСС, включающий два узла. Следует подчеркнуть, что конфигурация графа моделируемого процесс функционирования сети и направлений возможных переходов останутся прежними при любом количестве узлов сети.

Для произвольного количества узлов в [2] получены математические выражения для вычисления пропускной способности БСС, коэффициента занятости микропроцессоров и приемопередатчиков.

В ходе анализа графовой модели процесса функционирования сети и оценки ее основных характеристик варьируя исходные данные, такие как количество  $N$  узлов сети, интенсивности потоков заявок различных абонентов  $\lambda_i$ , интенсивностей обслуживания  $\mu_i$  можно определить рациональные, с точки зрения потерь информации и занятости приемопередатчиков (ПП) и МП узлов, характеристики сети.

Так при анализе можно предположить что, увеличение интенсивности  $\lambda_y$  ретранслируемого потока заявок характеризуется снижением значения такого критерия как связность узлов (рис. 2).



**Рис. 2.** Упрощенная схема БСС с выделенными зонами связности

При этом достижение пакетом шлюза происходит при большем количестве хопов, что увеличивает время его доставки.

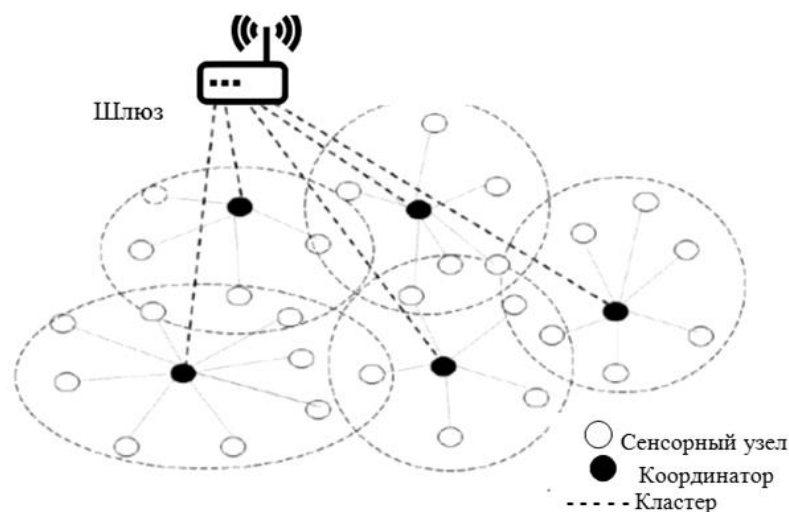
Обслуженный поток заявок поступающих на шлюз и характеризуемый параметром  $\mu_3$  определяет пропускную способность сети.

На рассмотренные характеристики сети существенное влияние оказывают параметры топологической структуры, конфигурация сети, выбор и оптимизация которой решается в ходе *синтеза*.

Технология digimesh позволяет строить беспроводные сенсорные сети с самоорганизацией, обеспечивающей процесс самообразования, настройки и поддержания работы сети. При такой организации сеть способна изменять свои параметры и менять логику работы из-за: изменения её загрузки, конфигурации, выхода из строя узлов, критического снижения энергии источников питания и др. Сеть считается работоспособной до тех пор, пока существуют маршруты доставки сообщений от всех исправных конечных устройств до координатора.

Так как развертывание сети системы мониторинга лесов осуществляется на значительных территориях то, как показано в [4] digimesh сеть строится в виде кластерной архитектуры (рис. 3).

Данный подход заключается в том, что некоторое количество узлов сенсорного поля сети объединяется в группы-кластеры. Из узлов этой группы выбирается координатор, который принимает данные от узлов кластера, в котором между парой узлов может быть несколько маршрутов, обрабатывает полученные данные и передает их приложению верхнего уровня на шлюз по радиоканалу.



**Рис. 3.** Схема кластерной беспроводной сенсорной сети

Кластерная топология при большом количестве узлов сети требует оптимизации метрики, как для кластеров, так и для всей сети в целом.

Под метрикой будем понимать параметр, выраженный числом переходов (так называемых "хопов" или "скачков"), которые влияют на выбор маршрута в сети. В таблице маршрутизации предпочтительным является тот маршрут, у которого меньшая метрика.

Метрики позволяют выбрать лучший маршрут из нескольких возможных маршрутов от узлов к координатору кластера. Маршрут в направлении шлюза от координаторов кластеров сети также должен быть с наименьшей метрикой.

Кластерная организация считается эффективной и масштабируемой, но лишь при условии рационального, с точки зрения наименьшей метрики, выбора координатора в кластерной сети в конкретный момент времени.

Задача, выбора места расположения координатора в кластере может быть решена на графовой модели в ходе *синтеза*.

В общем случае как топологическую структуру БСС, так и конфигурацию каждого кластера можно представить в виде неполного графа с произвольным расположением вершин и ветвей, соединяющих эти вершины.

В системах мониторинга одним из важнейших требований является его непрерывность, т.е. обеспечение сбора параметров окружающей среды на всем пространстве или на всем процессе за счет лучшего покрытия заданной области лесной территории в течение достаточно длительного периода времени.

Под синтезом конфигурации структуры будем понимать процесс построения сети с минимальной суммарной протяженностью линий радиосвязи и с меньшей метрикой. Для решения этой задачи используем модели взвешенных графов [5].

Графы с заданной числовой функцией только на вершинах или только на дугах или одновременно на дугах и вершинах получили название взвешенных графов. Введем следующие обозначения:

$$\begin{aligned} p(v_i, v_j) &- \text{значение ветви } (v_i, v_j); \\ q(v_i) &- \text{значение вершины } v_i, \end{aligned} \quad (1)$$

Задача размещения координатора в структуре кластера БСС сводится к задаче минимизации суммарной длины его линий радиосвязи с узлами.

Эту задачу можно сформулировать следующим образом. Имеется граф сети  $G_s=(V_s, B_s)$ , где  $V_s$  множество вершин графа,  $B_s$  множество ветвей графа. Необходимо среди множества вершин графа выбрать такую  $V_i$  вершину, чтобы сумма значений путей от нее до всех остальных вершин была бы минимальной. Вершина, удовлетворяющая решению поставленной задачи, называется медианой.

Введем обозначения для формального описания медианы:

$P(V_i - V_j)$  – значение минимального пути с началом в вершине  $V_i$ ;

$F(V_i)$  – сумма значений минимальных путей от вершины  $V_i$  до всех остальных вершин;

$\Psi(V_i)$  – сумма значений минимальных путей от всех вершин графа до вершины  $V_i$ .

Каждую вершину графа характеризует своя пара чисел  $F(V_i)$  и  $\Psi(V_i)$ , получивших название внешнего передаточного числа и внутреннего передаточного числа соответственно и их значения определяются по следующим формулам:

$$F(V_i) = \sum_{V_k \in V'} P(V_i - V_k), \quad \Psi(V_i) = \sum_{V_k \in V'} P(V_k - V_i) \quad (2)$$

Выражения (2) не учитывают особенностей узлов БСС. Часто в соответствии с «важностью» узлов вершины графа, описывающего сеть, могут иметь определенный вес. Например, численное значение связности, объем передаваемой и принимаемой информации и т.п.

Итак, с учетом весов вершин графа, вышеприведенные формулы примут вид:

$$F(V_i) = \sum_{V_k \in V'} q_k (V_i - V_k), \quad \Psi(V_i) = \sum_{V_k \in V'} q_k (V_k - V_i) \quad (3)$$

где  $q_k$  значение вершины  $V_k$ .

Внешней медианой графа  $G_s=(V_s, B_s)$  называется вершина  $V_j^*$ , для которой внешнее передаточное число минимально.

Внутренней медианой графа  $G_s=(V_s, B_s)$  называется вершина  $V_j^*$ , для которой внутреннее передаточное число минимально.

После введенных выражений медиана графа  $G_s=(V_s, B_s)$  это вершина  $V^*$ , для которой сумма внутреннего и внешнего передаточного чисел минимальна. Для выбора координатора в кластере сети, по представленной

методике, рассмотрим пример (рис. 4) на графе сети состоящей из четырех узлов.

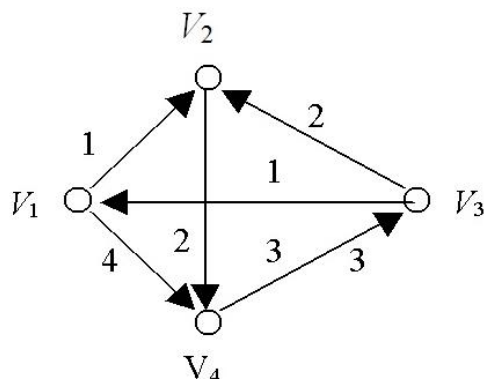


Рис. 4. Граф кластера сети

Определим медиану графа с изображенными направлениями переходов, значения вершин которого равны  $q_1=2$ ,  $q_2=2$ ,  $q_3=1$ ,  $q_4=4$ , а значения дуг  $p$  обозначены на рисунке 4.

Построим матрицу минимальных путей, в которой элемент в строке  $v_i$  и столбце  $v_k$  равен  $P(v_i - v_k)$ .

Для определения внешнего передаточного числа выбираем строку  $v_i$ . Каждый элемент этой строки умножаем на значение вершины, соответствующей столбцу, в котором находится данный элемент. Полученные произведения складываем.

$q_k$	→	2	2	1	4	
		$V_1$	$V_2$	$V_3$	$V_4$	
2	↓	$V_1$	0	1	6	3
	1	$V_2$	6	0	5	2
	2	$V_3$	1	2	0	4
	3	$V_4$	4	5	3	0
	4					

Определяем внутренние передаточные числа для вершин графа:

$$F(V_1) = 0 \cdot 2 + 1 \cdot 2 + 6 \cdot 1 + 3 \cdot 4 = 20$$

$$F(V_2) = 6 \cdot 2 + 0 \cdot 2 + 5 \cdot 1 + 2 \cdot 4 = 25$$

$$F(V_3) = 1 \cdot 2 + 2 \cdot 2 + 0 \cdot 1 + 4 \cdot 4 = 22$$

$$F(V_4) = 4 \cdot 2 + 5 \cdot 2 + 3 \cdot 1 + 0 \cdot 4 = 21$$

Для определения внешнего передаточного числа выбираем столбец  $V_i$ , каждый элемент этого столбца умножаем на значение вершины, соответствующей строки, в которой находится данный элемент. Получение произведения складываем.

Определим внешние передаточные числа для приведенного примера:

$$\Psi(V_1) = 0 \cdot 2 + 6 \cdot 2 + 1 \cdot 1 + 4 \cdot 4 = 29$$

$$\Psi(V_2) = 1 \cdot 2 + 0 \cdot 2 + 2 \cdot 1 + 5 \cdot 4 = 25$$

$$\Psi(V_3) = 6 \cdot 2 + 5 \cdot 2 + 0 \cdot 1 + 3 \cdot 4 = 34$$

$$\Psi(V_4) = 3 \cdot 2 + 2 \cdot 2 + 4 \cdot 1 + 0 \cdot 4 = 14$$

Выбирая среди передаточных чисел минимальное, определяем внешнюю и внутреннюю медианы. Внешней медианой является вершина  $V_4$ , внутренней медианой – вершина  $V_1$ .

Для определения медианы найдем суммы соответствующих передаточных чисел:

$$F(V_1) + \Psi(V_1) = 20 + 29 = 49$$

$$F(V_2) + \Psi(V_2) = 25 + 24 = 49$$

$$F(V_3) + \Psi(V_3) = 22 + 34 = 56$$

$$F(V_4) + \Psi(V_4) = 21 + 14 = 35$$

Наименьшее число 35, что соответствует вершине  $V_4$ . Таким образом, медианой графа и координатором кластера сети является вершина  $V_4^*$ .

Представленная методика применима для произвольного числа узлов кластера сети, однако при большом их количестве требуется использование приложений автоматизирующих получение результата.

Следует заметить, что процесс выбора координатора в кластере процедура не статическая, а динамическая (повторяющаяся, итерационная).

Так после очередного назначения координатора он выполняет свои функции до момента  $t_i$ . По прошествии некоторого времени равного  $\Delta t$  существующий координатор может, например, израсходовать значительное количество энергии на передачу сообщений от всех сенсорных узлов кластера к моменту времени  $t_j = t_i + \Delta t$ . Поэтому в момент времени  $t_j$  координатором в кластере может быть назначен другой исправный узел, сохранивший к этому времени наибольший энергетический запас.

Выбор и назначение координатора может происходить и по другим причинам. Так в беспроводной сенсорной сети на основе информации о количестве соседних узлов в кластере формируется граф связности с выбранным на определенный момент координатором. На основании графа связности и информации о запасе энергии узлов шлюз формирует расписание структур кластера, задающее их набор и продолжительность использования каждой из них. Расписание строится исходя из задачи максимальной продолжительности работы сети.

По окончании времени равного продолжительности использования выбранной структуры шлюз инициирует ее смену, оповещая все узлы соответствующей командой.

Шлюз передает каждому из узлов информацию о роли узла в новой структуре кластера и ID нового координатора выбранного по рассмотренной выше методике.

Рассмотрим формальное описание данной ситуации. Зададим реконфигурируемую сеть  $N$  в следующем виде:

$N = (G_s, G_{s,n}, \Pi)$ , где

$G_s = (V_s, B_s)$  - граф конфигурируемой сети.

$G_{s,n} = \{G_n(k), k \in V_s\}$  - множество сетевых графов, каждый элемент которого определяется вариантом  $k$  конфигурации сети.

$\Pi = (\langle \pi_1, t_1 \rangle, \langle \pi_2, t_2 \rangle, \dots, \langle \pi_k, t_k \rangle)$  - последовательность смены конфигураций, где  $\pi_i \in V_s$  - номер конфигурации,  $k$ -количество вариантов конфигураций,  $t_i$  - время ее использования.

Граф конфигураций  $G_s$  состоит из множества вершин  $V_s = \{1, 2, \dots\}$  и множества ребер  $B_s: B_s \subseteq V_s * V_s$ . Число вершин  $V_s$  в общем случае задает возможные состояния, в которых может находиться сеть. Состояния определяются положением координатора.

По истечении времени, равного продолжительности использования выбранной конфигурации или при других обстоятельствах, например отказ узла, шлюз выбирает следующую структуру в расписании и функционирование БСС на основании нового графа связности продолжается до новой смены координатора.

Представленные графовые модели позволяют получить предварительные оценки параметров беспроводных сенсорных сетей в ходе анализа и выбрать рациональную конфигурацию сети, с точки зрения расположения координатора в ходе ее синтеза.

### Библиографический список

1. Заяц А.М., Игнатъева Т.И. Модель беспроводной сенсорной сети системы мониторинга лесных территорий. В сборнике: Эколого-экономические и технологические аспекты устойчивого развития Республики Беларусь и Российской Федерации. сборник статей III Международной научно-технической конференции: в 3 т.. Минск, 2021. С. 24-31.
2. Заяц А.М., Игнатъева Т.И. Алгоритм определения показателей эффективности беспроводных сенсорных сетей. В сборнике: Информационные системы и технологии: теория и практика. Сборник научных трудов. Санкт-Петербургский государственный лесотехнический университет имени С.М. Кирова. Санкт-Петербург, 2021. С. 81-89.
3. Игнатъева Т.И., Заяц А.М. Модельное представление узлов беспроводной сенсорной сети. В сборнике: Актуальные вопросы в лесном хозяйстве. Материалы IV международной научно-практической конференции молодых ученых. 2020. С. 133-138.
4. Заяц А.М., Хабаров С.П. Беспроводные сенсорные сети в лесном хозяйстве: Учебное пособие / А. М. Заяц, С.П. Хабаров. – СПб.: СПбГЛТУ, 2021.
5. Панфилов, И. В. Информационные сети. Учеб. пособие для студ. 071900, 210200, 060800 / И. В. Панфилов, С. П. Хабаров, А. М. Заяц ;



**А.М. Заяц**, кандидат технических наук, профессор  
Кафедра информационных систем и технологий  
СПб ГЛТУ им. С.М.Кирова  
zamfta@yandex.ru

## **ОБЗОР ПРИМЕНЕНИЯ ТЕХНОЛОГИЙ ВИРТУАЛЬНОЙ И ДОПОЛНИТЕЛЬНОЙ РЕАЛЬНОСТИ ПРИ ПРОЕКТИРОВАНИИ И ИССЛЕДОВАНИИ СИСТЕМ МОНИТОРИНГА ЛЕСНЫХ ТЕРРИТОРИЙ**

***Аннотация.** В статье рассматриваются основные направления использования технологий виртуальной (VR) и дополненной реальностью (AR) на платформе цифровой двойник (Digital Twin) в системах мониторинга лесных территорий.*

***Ключевые слова:** технологии виртуальной и дополненной реальности, цифровой двойник, мониторинг лесных территорий, технологии искусственного интеллекта.*

В предлагаемой статье рассматриваются основные направления использования технологий виртуальной (VR) и дополненной реальностью (AR) на платформе цифровой двойник (Digital Twin) в системах мониторинга лесных территорий:

1. Решение задач проектирования системы мониторинга лесных территорий (СМЛТ) с использованием цифрового двойника беспроводной сенсорной сети (БСС) [1] на виртуальной модели лесного участка.

При проектировании цифровой двойник разрабатывается раньше будущей системы и тестируется в разных условиях и режимах. По результатам тестирования можно внести нужные изменения и доработать проект, если будут выявлены недочеты, как в штатном, так и в штатном режиме работы сети.

Цифрового двойника как виртуальное представление БСС на цифровой модели лесного участка, можно использовать для визуализации, прогнозирования и предоставления ответов об измеряемых параметрах и функциональных характеристиках сети, прогнозировать результаты мониторинга и выполнять итерации моделируемого процесса, причем с высокой точностью.

2. Симуляция алгоритмов функционирования БСС и СМЛТ на виртуальных моделях с целью накопления статистики возможных событий

для дальнейшего использования при принятии решений в реальных условиях [2].

Цифровые двойники позволяют выбрать нужные режимы работы, провести виртуальные эксперименты, которые в реальных системах могут быть сопряжены с риском повредить аппаратно-программные компоненты сети.

Эти виртуальные точные копии любых объектов обеспечивают взаимодействие пользователя с системой мониторинга лесных территорий, за счет реализации компьютерной модели, объединяющей информацию в единую среду, которая может меняться во времени. Например, двойники дают возможность проверить влияние различных негативных воздействий на объект при помощи симуляции.

Дополненная реальность обеспечивает встраивание отдельных цифровых компонент в реально существующую систему, что позволяет проведение проверки нововведений и выявить возможные проблемы на ранних этапах эксплуатации.

3. Анализ мониторинговой информации и получение результатов с использованием технологий BigData и искусственного интеллекта.

При мониторинге лесных территорий масштабных размеров собираются огромные объемы разнородной информации, которую нужно структурировать и анализировать. На цифровой модели лесной территории используя технологии искусственного интеллекта и Big Data можно быстро искать нужную информацию в большом объеме данных и анализировать их. Цифровой двойник местности позволит без полевых испытаний прогнозировать изменение состояния окружающей среды, исходя из прошлых данных, выявить закономерности погодных или климатических условий на мониторинговой территории для оценки возможного возникновения или определения негативных явлений.

Функциональные приложения с искусственным интеллектом обеспечивают поиск нужной информации в большом объеме мониторинговых данных, используя их для прогнозирования негативных воздействий, исходя из ранее полученных данных.

В самой идее цифрового двойника как эволюции ИТ-продуктов нет ничего нового: расчеты и модели того, как будет вести себя какая-нибудь некий объект, система или сущность, делались и раньше. Но лишь недавно появились достаточные вычислительные мощности, чтобы проводить такие расчеты в реальном времени, а также возможности для постоянного обновления моделей на основе данных, получаемых с реальных объектов.

Современный подход в этом направлении начинается с создания новой технологии, которая интересна и полезна большому числу участников, информационных сервисов и бизнес-процессов. Затем в неё встраиваются инновации и когда продукт готов, вокруг него создается полноценная экосистема из разработчиков, заинтересованных бизнесменов и партнёров.

Средства аналитики, получая определенного вида «телеметрию» от инфраструктуры и приложений, должны быть способны в режиме реально-

го времени осуществлять различные виды анализа о состоянии, поведении и тенденциях как самих ИТ-услуг, так и клиентов, потребляющих эти услуги, а также оценивать, насколько результат соответствует ожидаемому качеству предоставления сервиса.

Проведем обзор трехмерных 3D моделей, использующих виртуальную и дополненную реальность, которые имеют различные приложения: от анализа и проектирования систем мониторинга лесных территорий и беспроводных сенсорных сетей, обеспечивающих сбор данных, до понимания того, как использовать его результаты в случае негативных ситуации в лесу.

Как показано в работе [3] слияние трехмерного моделирования с VR и AR обеспечивает пользователям принимать решения, по поведению объекта, хотя и смоделированного.

Виртуальная реальность как компьютерное отображение реального мира позволяет совместить, например, 3D-модель лесной территории, использующей географические данные, и еще не существующую модель беспроводной сенсорной сети СМЛТ. Модель размещения БСС может быть встроена в существующую трехмерную сцену лесной территории с дальнейшим исследованием ее параметров, таких как: связность узлов и топология сети с учетом рельефа местности, таблицы маршрутизации, время и надежность доставки сообщений и т.п.

Использование этих моделей, наряду с AR и VR, открывает инновационные возможности для получения опыта взаимодействия пользователя с СМЛТ в режиме реального времени и это не просто некое нереальное видение, а технология принятия решения на основе цифровой модели среды и происходящих событий. Так, например, можно реализовать ситуацию по определению рисков распространения лесных пожаров на основе модели высот ветвей и крон (выявляются участки, где низовой пожар может переходить в верховой), риски перехода пожара на населенные пункты, вероятные направления распространения пожара и т.п.

Графическая система таких технологий позволяет отображать на мониторе во всех плоскостях объекты и их динамику, а также данные, которые к ним относятся (описание, характеристики, результаты расчетов и т.д.). В такой системе отсутствуют ограничения по масштабированию отображения 3D модели, и если это один «движок», то переход от одного масштаба 3D модели к другому будет бесшовным. Все остальное – это интерфейс и алгоритмы работы с объектами 3D модели, которые будут наполнять базу данных.

Для пользователя будет меняться только интерфейс при редактировании объектов – в зависимости от ситуации и состояния самого объекта. Такая платформа позволяет разрабатывать любые модели объектов без ограничений по размерам и сложности, а значит, не будет технологической проблемы проектировать новый объект, рассчитать его статические и динамические характеристики, проводить испытания – все в одной графиче-

ской среде с использованием единой базы данных. Нет необходимости переносить данные, конвертировать и использовать разное ПО.

Так в Финляндии эти технологии помогают бизнесу решить четыре важнейших задачи: узнать точно, что и где растет; что заготовлено и когда надо вывезти; что уже вывезли и как использовали [4]. Здесь разработчики ПО рассчитывают в ближайшее время представить сервис для создания полной цифровой модели леса. Сейчас для этого апробируются две технологии сбора данных — дроны, вооруженные камерами, и беспилотники с лазерами. В будущем каждое дерево, растущее в Финляндии, получится смоделировать, и участники экосистемы смогут узнать его местоположение, длину, диаметр, вид и моделировать различные ситуации.

С помощью гарнитуры виртуальной реальности можно будет посетить любой лес. Так что владелец лесного участка может совершать его «обход», сидя в гостиной, попутно получая консультацию специалиста по уходу за деревьями и примерный расчет прибыли.

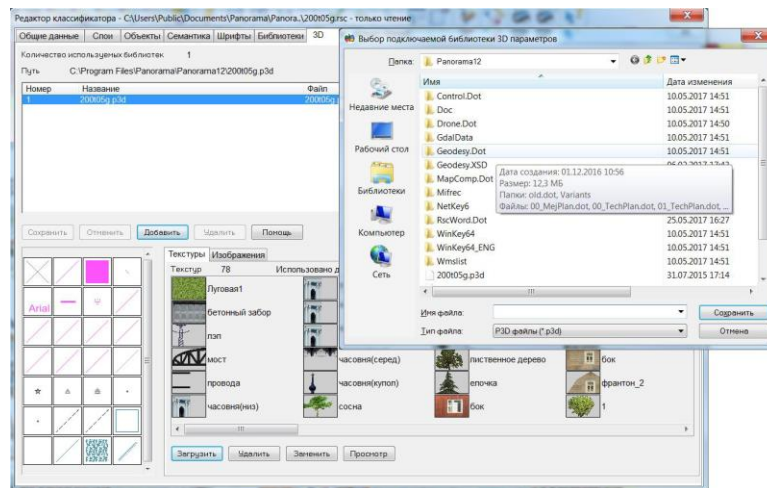
В других системах сейчас ведутся опыты VR, предоставляющие возможности кругового обзора со статичной точки и «телепортации» в другую панораму. Подобные опыты VR, позволяют оператору передвигаться, используют экспертные знания при работе с 3D-инструментами, осуществлять рендеринг в режиме реального времени с использованием движков и графического оборудования.

Дополненная реальность как связующее звено между виртуальной реальностью и действительностью отличается тем, что она включает размещение изображения некоего объекта на картине реального мира. Примером могут служить ситуации, при которых создается образ лесного участка, охваченного пожаром, а на него накладывается схематическое изображение выполняемых мер по его тушению и последовательность противопожарных действий.

Для построения 3D моделей соединенных с VR и AR может быть использована отечественная геоинформационная система «ПАНОРАМА»[5].

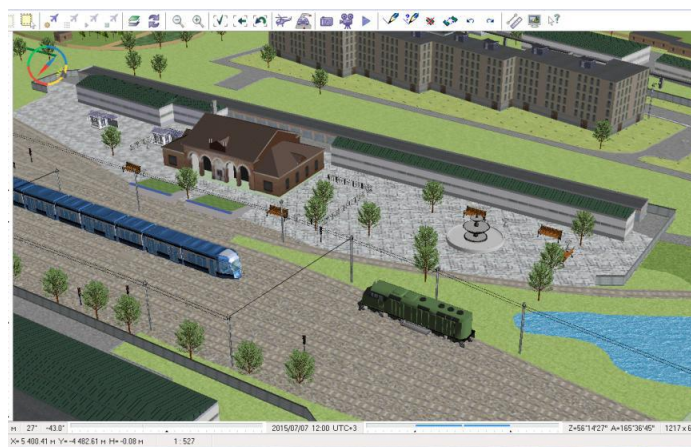
Это универсальная ГИС, имеет средства создания и редактирования цифровых карт и планов местности, в которых выполняются различные измерения и расчеты, оверлейные операции, построение 3D-моделей, обработка растровых данных, подготовка графических документов в электронном и печатном виде, а также использование инструментальных средств для работы с базами данных.

В комплекте ГИС Панорама вместе с классификаторами электронных векторных карт поставляются библиотеки трехмерных изображений для различных масштабов. Для получения объемного вида имеющихся электронных карт достаточно к классификатору карты подключить одну или несколько библиотек.



**Рис. 1.** Подключение к классификатору библиотеки 3D изображений

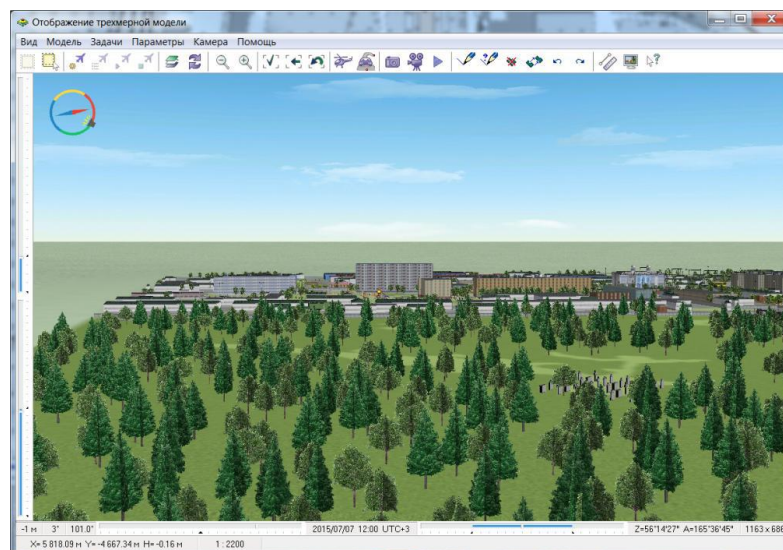
После подключения библиотеки выбираются типы объектов и назначается им подходящее трехмерное изображение из имеющихся в библиотеке. Такой подход дает возможность быстрой подготовки карты к отображению в трехмерном виде.



**Рис. 2.** Трехмерная модель на основе библиотеки трехмерных изображений

При необходимости более детального и индивидуального отображения местности можно создавать свои объемные изображения объектов.

В зависимости от настройки трехмерного вида отдельных объектов и по степени детализации трехмерные модели делятся на типовые, детальные, модели внутренних помещений и тематические.



**Рис.3.** Участок местности, построенный из типовых условных знаков

Построение типовой модели является самым быстрым способом получения качественной трехмерной модели местности. Типовые модели могут применяться для визуальной оценки взаимного расположения объектов, например, узлов БСС с учетом особенностей рельефа, анализа взаимного расположения деревьев и учетом различных ландшафтных особенностей местности.

Трехмерные модели детального вида описывают местность с объектами, вид которых настраивается индивидуально. Модели детального вида содержат поверхность рельефа местности, объекты, объемное изображение которых приближается к их реальному виду на местности (деревья различных пород, дороги и сооружения на лесной территории и многое другое).

В редакторе трехмерного вида объектов ГИС «Панорама» доступно импортирование данных из формата VRML, поэтому некоторые элементы индивидуального вида объектов могут быть созданы в трехмерных редакторах сторонних разработчиков и загружены в различные шаблоны отображения этих объектов. Тематические модели создаются по тематическим картам и могут использоваться для оформления статистических диаграмм, отображающих распределение некоторых показателей по территориям.

Для построения трехмерной модели местности используются векторная карта, матрица высот, триангуляционная (TIN) модель рельефа, облако точек (MTD-модель), классификатор карты, библиотека трехмерных моделей объектов, цифровые фотоснимки местности и цифровые фотографии объектов местности, трехмерные тайлы (DB3D), как небольшие изображения одинаковых размеров, служащие фрагментами большой картины.

Изображение местности может быть использовано для наложения на поверхность рельефа.





**Рис.4.** Трехмерная модель на основе цифрового снимка и матрицы высот

Параметры поверхности рельефа, такие как углы закрытия, крутизна склонов, учитываются при исследовании модели с различными вариантами размещения узлов беспроводной сенсорной сети на лесной территории.

Для обеспечения технологии, реализуемой в ГИС «Панорама» используется программное обеспечение для создания и отображения трехмерных моделей, которое может выполняться в 32-х и 64-х разрядной операционной среде Windows 7 и выше на компьютерах с архитектурой процессоров Intel (Core i5, Core i7 или старше).

Рекомендуемые требования к составу аппаратно-программных средств:

- оперативная память: от 8 Гбайт и выше;
- жесткого диска: от 2 Тбайт и более;
- размер экрана не менее 1024 на 768 точек;
- сетевой адаптер, поддерживающий технологию Gigabit Ethernet не менее 1 Гбит/с.

Видеокарта с аппаратной поддержкой OpenGL версии 1.4 и выше и параметрами не ниже, чем:

- объем памяти – 2048 Мбайт;
- шина памяти – 128 бит;
- количество универсальных конвейеров (SPU) – 640;
- текстурных блоков (TMU) – 80;
- блоков растеризации (ROP) – 16;
- быстродействие (SPU 1085 МГц, ядра 1020 МГц, памяти 5400 МГц).
- наличие USB-порта, доступного для ключа защиты приложения.

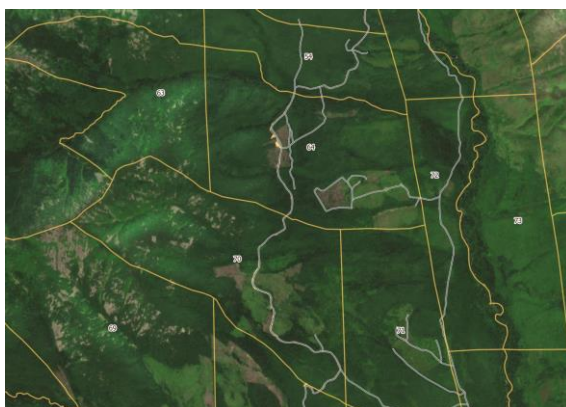
Другой платформой создания цифровой модели леса является геоинформационное решение АО «ТЕРРА ТЕХ» [6] созданное на базе источников пространственной информации, в первую очередь данных дистанционного зондирования Земли (ДЗЗ).

Здесь создание цифровой модели леса (ЦМЛ) осуществляется путем сопоставления ЦМР – цифровой модели рельефа и ЦММ – цифровой модели местности, определяемых по снимкам, после их согласования и приведения к единому пространственному разрешению.



**Рис.5.** Цифровая модель леса

Лесной массив представляется в ЦМЛ не в виде отдельных деревьев, а как непрерывная трехмерная структура, наслаивающаяся на рельеф местности, которой могут отображаться квартальная и дорожная сеть, водоохранные зоны вдоль рек, склоны и их крутизна и многое другое.

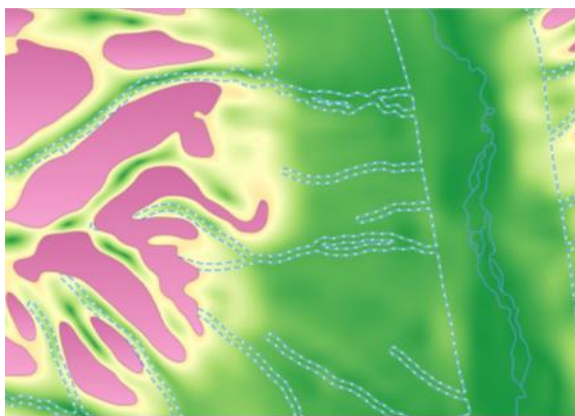


**Рис. 6.** Квартальная и дорожная сеть



**Рис. 7.** Водоохранные зоны





**Рис.8.** Склоны и их крутизна

Цифровые двойники как совместная платформа технологий VR, AR, ЦММ и 3D моделирование - это виртуальные представления любой системы даже такой сложной экосистемы как территории покрытые лесом. Они обеспечивают безопасную среду тестирования различных инноваций. За счет моделирования всей цепочки создания системы они могут повысить устойчивость сохранения функционала на протяжении всего жизненного цикла, от проектирования для повторного использования. Это среда, в которой можно в режиме реального времени постоянно вносить новые данные и изменения, смотреть, что при этом происходит, что можно улучшить, проводить различные симуляции продуктов. Это совмещение виртуального и реального мира. При этом можно задействовать и другие развивающиеся технологии, такие как искусственный интеллект и машинное обучение, большие данные, облачные вычисления.

#### **Библиографический список**

1. Заяц А.М., Хабаров С.П. Беспроводные сенсорные сети в лесном хозяйстве: Учебное пособие /А. М. Заяц, С.П. Хабаров. – СПб.: СПбГЛТУ, 2021. 92 с.
2. Заяц А.М., Хабаров С.П. Исследование алгоритма работы распределенной системы мониторинга лесных территорий. // Известия Санкт-Петербургской лесотехнической академии. 2019. Вып. 229. С. 243–254.
3. Миронова Ю.Н. Новые методы виртуального моделирования в геоинформационных технологиях//Интернет-журнал «НАУКОВЕДЕНИЕ» Том 8, №5 (2016) <http://naukovedenie.ru/PDF/03TVN516.pdf>.
4. Анна Орешкина. В цифровом лесу: Интернет вещей для лесорубов. <https://www.if24.ru/tsifrovoi-les-iot-dlya-lesorubov/>
5. Геоинформационная система «Панорама» (ГИС Панорама x64») Прикладные задачи. Построение 3D модели. ПАРБ.00046-06 98 07. <https://gistoolkit.ru/download/doc/model3d.pdf>.
6. АО «ТЕРРА ТЕХ» – дочернее предприятие АО «Российские космические системы», коммерческий оператор услуг ДЗЗ и геоинформационных сервисов на их основе. <https://terratech.ru/>

**Т.К. Екшикеев**, кандидат экономических наук, доцент,  
СПХФУ,  
tag2009spb@ya.ru

**И.А. Обухова**, кандидат технических наук, доцент,  
Кафедра информационных систем и технологий  
СПбГЛТУ им.С.М.Кирова  
IObukhova@inbox.ru

## **ИСПОЛЬЗОВАНИЕ БАЛЬНО-РЕЙТИНГОВОЙ СИСТЕМЫ ПРИ ОЦЕНКЕ ЗНАНИЙ СТУДЕНТОВ, ФОРМИРУЕМЫХ В ЭЛЕКТРОННОЙ ИНФОРМАЦИОННО-ОБРАЗОВАТЕЛЬНОЙ СРЕДЕ**

***Аннотация.** Балльно-рейтинговая система оценки результатов обучения студентов предусматривает накопление студентами оценок в баллах при изучении дисциплин в течении определенного периода времени. Актуальность оценки результативности учебной деятельности студентов за последние полтора года обусловлена пандемией. COVID-19 выступил катализатором цифровизации оценки, которая давно назрела. Приводится результат использования балльно-рейтинговой системы при оценке знаний студентов формируемых в электронной информационно-образовательной среде.*

***Ключевые слова:** балльно-рейтинговая система, электронная информационная образовательная среда, текущий и промежуточный контроль знаний.*

### **Введение – актуальность исследования.**

Пути реализации балльно-рейтинговой системы (БРС) имеют существенные различия детерминированные миссией конкретного образовательного учреждения, законами и правилами практического опыта построения учебного процесса. Механизмы внедрения балльно-рейтинговой системы в учебный процесс являются актуальной темой многих научных исследований. Высокую практическую значимость, как считают авторы – имеют исследования позволяющие сформулировать валидные рекомендации по реализации БРС в высших учебных заведениях. В данной статье авторами представляется «дорожная карта» изучения дисциплин в соответствии с БРС и представлены результаты исследования оценки эффективности разработки тестов в Moodle.

Пути реализации балльно-рейтинговой системы (БРС) имеют существенные различия детерминированные миссией конкретного образовательного учреждения, законами и правилами практического опыта построения учебного процесса. Механизмы внедрения надежной балльно-рейтинговой системы в учебный процесс являются актуальной темой многих научных исследований. Высокую практическую значимость, как счи-

тают авторы – имеют исследования позволяющие сформулировать валидные рекомендации по реализации БРС в высших учебных заведениях. В данной статье авторами представляется «дорожная карта» изучения дисциплин в соответствии с БРС и представлены результаты исследования оценки эффективности разработки.

В исследовании принимали участие студенты групп изучавших дисциплины без использования БРС и с использованием БРС. Используемая бально-рейтинговая система основана на календарно-тематическом плане изучения технических и гуманитарных дисциплин: информационных систем и технологий, физики (СПбГЛТУ), экономики (СПбФХУ). Студенты изучают дисциплины на основе набора баллов в процессе изучения теоретического материала, практических / лабораторных занятий, самостоятельной работы и экзамена.

Исследование эффективности изучения дисциплин в соответствии с БРС проводилось в период 2018-2022, с использованием электронной информационно-образовательной среде в 2019-2022.

Можно сформулировать – исследования по использованию БРС носили лонгитюдный характер и длились пять лет. В последние два года авторами был накоплен материал при оценке знаний студентов – на основе БРС формируемых в электронной информационно-образовательной среде.

Авторами использовались – учебно-методические материалы для основы бально-рейтинговой системы и наполнения электронной информационно-образовательной среды [2, 3].

В ходе исследований проводился контроль остаточных знаний студентов, сравнительный анализ результативности сдачи экзаменов в форме теста. Тестирование проводилось как при помощи анкет теста в очном формате и в электронной информационно-образовательной среде. Банк вопросов в СПбФХУ – по дисциплинам 3 зет достигал 450 тестовых вопросов множественного выбора, верно-неверно и расчетных. Пример характеристики экзаменационного теста – 80 вопросов с ограничением времени 60 минут.

В дисциплинах на 1-4 зет – определялась база тестов как минимум из 130 вопросов.

Рейтинговый балл по каждой дисциплине складывался из баллов, набранных студентами при выполнении всех видов учебных работ, а также баллов, полученных за знания по теоретической части дисциплины.

По итогам текущего контроля и результатам итоговой аттестации формировался рейтинговый балл по каждой дисциплине.

Для организации текущего контроля по каждому виду занятий дисциплин выделялись основные разделы дисциплины, подлежащие контролю (контрольные точки), и разрабатывались правила формирования рейтинговой оценки.

При формировании рейтинговой оценки по каждой дисциплине:

– общий балл распределялся между всеми контролируемымими видами занятий;

- для каждого вида занятий определить относительный вклад каждого контролируемого раздела (темы) в общий балл по виду контроля;
- устанавливались четкие критерии оценки по каждому виду занятий.

Выполнение индивидуальных заданий проводилось во время самостоятельной работы и предполагало выполнение отчетов загружаемых в электронную информационно-образовательную среду.

Минимальное значение рейтинговой оценки, набранной студентом по результатам текущего контроля по всем видам занятий, при котором студент допускался к сдаче экзамена, составлял определенное количество баллов.

Студент, получивший по результатам текущего контроля и экзамена рейтинговую оценку по дисциплине – менее требуемых баллов, аттестовывался неудовлетворительно, и ему предоставляется возможность ликвидировать задолженность по дисциплине в дни переэкзаменовок или по индивидуальному графику, установленному учебным отделом.

Полученные в ходе исследования данные позволили сделать вывод о качественном улучшении уровня освоения дисциплин на основе балльно-рейтинговой системы – студентами в электронной информационно-образовательной среде (в том числе, в Moodle) (ранее, до Covid-19, студенты тестировались в классе на ПК в разных тестовых системах).

Результаты тестирования в среде Moodle (СПбГЛТУ) (график оценок с участниками (Рисунок 1)) и пример таблицы результатов тестирования в МУДЛ по дисциплине Информационные технологии (табл. 1).

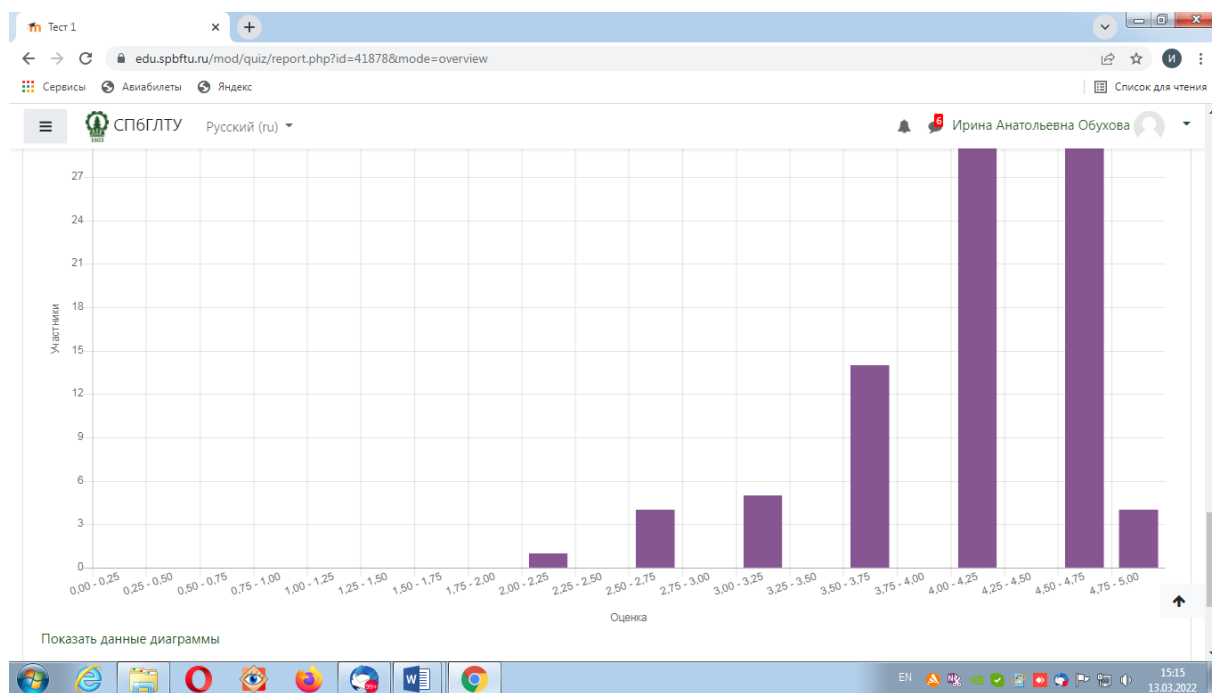


Рис. 1. Результаты тестирования в среде Moodle

Результаты тестирования в Moodle и средний балл внизу таблицы в Moodle (голубым цветом выделена наилучшая попытка):

Таблица 1

**Результаты тестирования в МУДЛ по дисциплине Информационные технологии**

Завершено	11 января 2022 09:58	11 января 2022 10:07	9 мин. 1 сек.	<u>4,00</u>
Завершено	11 января 2022 10:45	11 января 2022 10:52	6 мин. 35 сек.	<u>4,00</u>
Завершено	22 января 2022 09:18	22 января 2022 09:29	10 мин. 43 сек.	<u>3,50</u>
Завершено	22 января 2022 09:21	22 января 2022 09:30	9 мин. 6 сек.	<u>3,50</u>
Завершено	22 января 2022 09:22	22 января 2022 09:42	20 мин. 1 сек.	<u>4,00</u>
				3,91 (16)

В СПбГЛТУ (Лесотехнический университет) приняты следующие критерии оценки знаний студентов (указано в рабочих программах):

***Критерии оценки усвоения дисциплины (экзамен)***

Оценка (балл)	Критерии
Отлично	Студент показал творческое отношение к обучению, в совершенстве овладел всеми теоретическими вопросами дисциплины, показал все требуемые умения и навыки.
Хорошо	Студент овладел всеми теоретическими вопросами дисциплины, показал основные умения и навыки.
Удовлетворительно	Студент имеет недостаточно глубокие знания по теоретическим разделам дисциплины, показал не все основные умения и навыки.
Неудовлетворительно	Студент имеет пробелы по отдельным теоретическим разделам специальной дисциплины и не владеет основными умениями и навыками.

***Шкала оценивания***

Оценка	Неудовлетворительно	Удовлетворительно	Хорошо	Отлично
Баллы в БРС университета	50 и менее	51 -70	71-84	85-100
Уровень сформированности компетенций	Не сформированы	Пороговый	Базовый	Продвинутый

Результат промежуточной аттестации по дисциплине «Информатика и цифровые технологии» студент может получить в процессе сдачи экзамена или на основании итогового рейтинга по дисциплине.

Итоговый рейтинг служит для оценки знаний, умений и навыков студента по всему объему учебной дисциплины. Итоговый рейтинг изме-

ряется в баллах (от 0 до 100) и вычисляется на основании текущих рейтингов путем усреднения.

Преподаватель определяет качественные диапазоны соответствия рейтинга студента принятой четырехбалльной шкале оценок.

При определении итогового рейтинга преподаватель может добавить студенту от 1 до 10 баллов к окончательной рейтинговой оценке, что позволяет учесть активность студента.

Если итоговый рейтинг и оценка к концу семестра удовлетворяет студента, то он вправе не приходить на экзамен. Студент вправе прийти на экзамен в период сессии в соответствии с расписанием для изменения оценки.

К примеру, по дисциплине «Информатика и цифровые технологии»:

- ✓ **неудовлетворительной оценке** соответствует диапазон от 0 до 50 баллов;
- ✓ **удовлетворительной оценке** соответствует диапазон от 51 до 70 баллов;
- ✓ **хорошей оценке** – от 71 до 84 баллов;
- ✓ **отличной оценке** – выше 85 баллов.

Использование бально-рейтинговой системы более совершенно оценивает результаты обучения студентов, позволяет использовать активно тестирование студентов в среде МУДЛ и объективно определяет оценки, что позволяет снизить количество неуспевающих студентов.

Применение подобной системы позволило повысить надежность усвоения студентами материалов дисциплин, а также сократить число опозданий и пропусков занятий, повысило уровень вовлеченности студентов в учебный процесс за счет выбора индивидуального маршрута обучения в рамках темы или курса в электронной информационно-образовательной среде.

### Библиографический список

1. Гнеденко Б.В. Математические методы в теории надежности /Б.В. Гнеденко, Ю.К. Беляев, А.Д. Соловьев. –М.: Наука, 1965. -524 с.
2. Екшикеев Т.К. Реализация информационно-аналитических моделей инновационных фармацевтических процессов: сетевое планирование и управление /Т.К. Екшикеев. – М.: КноРус, 2019. -252 с.
3. Информационные технологии в создании программного обеспечения инновационных разработок: сетевое планирование и управление: методические указания к лабораторным работам для студентов всех форм обучения направления подготовки 38.04.04 «Государственное и муниципальное управление» (магистерская программа «Государственное и муниципальное управление в лесном секторе») / сост.: И. А. Обухова, Т. К. Екшикеев. – Санкт-Петербург: СПбГЛТУ, 2020. – 32 с.
4. Наркевич И.А. К столетию университета: «Мы уверенно смотрим в будущее» //Вестник высшей школы 8(151) октябрь 2019. С. 1-2.

5. Прямая линия с В.В. Путиным – 20 июня 2019: В.В. Путин считает главным способом решения проблемы уровня жизни рост производительности труда. Электронный источник: <https://tass.ru/obschestvo/6571104>. Дата обращения 15.09.2021.

**С.А. Иванов**, кандидат технических наук, доцент  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
kemsit@mail.ru

## **ОБЗОР ИНСТРУМЕНТОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ СОВРЕМЕННЫХ ИТ-ПЛАТФОРМ ДЛЯ РЕШЕНИЯ ЗАДАЧ ПРОГНОЗИРОВАНИЯ**

***Аннотация.** В статье рассмотрены стратегические и тактические задачи, которые решаются с помощью анализа данных. Приведены основные отличия в рамках классификации задач. Рассмотрены современные платформы, позволяющие решать задачи сбора, обработки данных, прогнозирования и поддержки принятия решений в условиях неопределенности.*

***Ключевые слова:** анализ данных, решение задач в условиях неопределенности, прогнозирование, law-code.*

На сегодняшний день перед всеми компаниями стоит большая задача разработки стратегических и тактических планов развития организации в быстроменяющихся, нестабильных условиях рынка, процессы которого протекают в условиях неопределенности [1]. Для решения этой задачи менеджментом компаний применяются различные инструменты анализа данных и прогнозирования дальнейшей ситуации. Основные планы можно разделить, как указывалось выше, на две основные категории: стратегические и тактические.

***Стратегические.** Для этого класса задач необходим достаточно большой горизонт планирования, который выходит за пределы одного года и часто варьируется в интервале 3-5 лет. В рамках анализа используются исторические данные, накопленные за большой промежуток времени, аналитические базы данных должны содержать как внутрикорпоративную информацию, так и агрегированные и очищенные данные из внешних источников. Так базы данных должны удовлетворять серьезным требованиям информационной безопасности, вплоть до предоставления доступа только к определенным столбцам и строкам таблиц, потому что в них содержится критически важная информация для организации.*

*Тактические.* Для оперативных баз данных горизонт событий, как правило, ограничивается несколькими месяцами. Они могут не содержать данных, полученных из внешних источников, ограничиваясь внутрикорпоративной информацией для принятия оперативных решений. Также в данном случае снижаются требования к обеспечению информационной безопасности – вполне достаточно разграничение доступа на уровне таблиц.

Для решения задач прогнозирования на основе анализа данных можно использовать R-программирование (язык для статистических вычислений и графики, который используется, в том числе для анализа больших данных) или Python, но этот вариант требует от аналитика достаточно серьезных навыков в программировании и понимании математических методов. Соответственно, оплачивать работу аналитиков высокого класса не всегда могут позволить себе организации малого и среднего бизнеса. Но на сегодняшний день на рынке представлено достаточно большое количество программных решений, которые позволяют использовать современный аналитический инструментарий практически без написания собственного кода.

Среди наиболее популярных платформ на российском рынке можно выделить линейку продуктов 1С. Внушительный пакет инструментов для бизнес-аналитики позволяет осуществлять поиск закономерностей в исходных данных информационной базы; управлять параметрами выполняемого анализа как программно, так и интерактивно; осуществлять программный доступ к результату анализа; автоматически выводить полученный результат анализа в табличный документ; создавать модели прогноза, позволяющие автоматически прогнозировать события или значения неких характеристик новых объектов [3, 4].

Механизм анализа данных представляет собой набор взаимодействующих друг с другом объектов встроенного языка, что позволяет разработчику использовать его составные части в произвольной комбинации в любом прикладном решении. Встроенные объекты позволяют легко организовать интерактивную настройку параметров анализа пользователем, а также позволяют выводить результат анализа в удобной для отображения форме в табличный документ.

Также на рынке большую нишу занимает *LoginomCompany*, которая специализируется на разработке систем для глубокого анализа, охватывающий вопросы сбора, интеграции, очистки данных, построения моделей и визуализации. Их продукт – платформа *Loginom* – позволяет: извлекать данные (источниками могут быть файлы, базы данных, веб-сервисы, бизнес-приложения); проводить очистку извлеченных данных, их преобразование, связывание, систематизацию и консолидацию; есть возможности машинного обучения, прогнозирования, сложных расчетов; обеспечена интеграция в ИТ-инфраструктуру, загрузка файлов базы, приложений, публикация веб-сервисов; реализованы широкие возможности визуализации, многомерного анализа и интерпретации результатов [2]. Основным отличием платформы от многих других инструментов является low-code, что



позволяет бизнес-аналитику не тратить время на написание кода, алгоритмов и формул – многие решения уже реализованы и внедрены на платформе.

В качестве заключения можно отметить, что языки программирования, например Python, предоставляют аналитику большие возможности и свободу действий, но предпочтительнее применять такой инструментарий при решении нетривиальных, нестандартных задач поддержки управленческих решений. В большинстве же случаев можно значительно сократить время от тестирования до создания работающего бизнес-процесса за счет использования готовых инструментов с функциональными возможностями интеграции в ИТ-инфраструктуру компаний.

### **Библиографический список**

1. The methodology for solving the multi-criteria problem of choosing a tool for the development of intrapreneurship for Russian corporates / S. A. Ivanov, M. K. Nemilentsev, I. V. Zementckii, E. F. Shchipanov // IOP Conference Series: Earth and Environmental Science : 2, Saint Petersburg, Virtual, 18–19 марта 2021 года. – Saint Petersburg, Virtual, 2021. – P. 012039. – DOI 10.1088/1755-1315/806/1/012039.
2. Евсюков, В. В. Аналитическая платформа Loginom - универсальный инструмент углубленной аналитики / В. В. Евсюков, А. В. Капустин, Ю. А. Ильина // Вестник Тульского филиала Финуниверситета. – 2020. – № 1. – С. 291-292.
3. Любавина, М. Н. Программные продукты фирмы "1С" в бизнес-аналитике предприятий / М. Н. Любавина, С. В. Данилова // Новые информационные технологии в образовании : Сборник научных трудов 20-й международной научно-практической конференции, Москва, 04–05 февраля 2020 года / Под общей редакцией Д.В. Чистова. – Москва: Общество с ограниченной ответственностью "1С-Публишинг", 2020. – С. 500-503.
4. Первухина, А. А. Возможности кадровой аналитики на базе программного обеспечения "1С" / А. А. Первухина // Альманах научных работ молодых ученых Университета ИТМО : XLVII научная и учебно-методическая конференция Университета ИТМО, Санкт-Петербург, 30 января – 2 февраля 2018 года. – Санкт-Петербург: федеральное государственное автономное образовательное учреждение высшего образования "Национальный исследовательский университет ИТМО", 2018. – С. 111-114.

**П.А. Кашо**, студент 2 курса ИСиТ  
Кафедра Информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
lighthousekx143@gmail.com

**Р.А. Яковлев**, студент 2 курса ИСиТ  
Кафедра Информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
hacks.max@yandex.ru

**С.С. Колмогорова**, кандидат технических наук, доцент  
Кафедра Информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
ss.kolmogorova@mail.ru

## **СИСТЕМА ВИЗУАЛЬНОГО ПРОЕКТИРОВАНИЯ NODE-RED КОМПОНЕНТОВ ИНТЕРНЕТА ВЕЩЕЙ В ЛЕСНОЙ ОТРАСЛИ**

***Аннотация.** В статье приводятся обзор технологии LoRaWAN, NodeRed, IoT. Акцентировано внимание на модуляции LoRa и на ее ключевых параметрах. Приведена методика расчета параметров модуляции LoRa и сети LoRaWAN. Описана модель системы получения данных сенсоров. Приведены необходимые формулы для расчета основных характеристик сети LoRa.*

***Ключевые слова:** Интернет вещей, LoRa, NodeRed, IoT, сенсоры, лесная отрасль.*

### **Введение**

Одним из ключевых направлений развития сетей связи стала концепция Интернета вещей (InternetofThings). В недалеком будущем к Интернету будет подключено несколько миллиардов устройств. Большинство устройств будет иметь батарейное питание. В связи с этим, одной из важных характеристик интернет вещей является продолжительная работа устройства без дополнительного обслуживания и зарядки. Для эффективного решения задач, связанных с энергопотреблением, появились новые типы сетей LPWAN (Low Power Wide Area Networks). Технологии, которые позволяют подключать автономные устройства к глобальной сети. Одна из наиболее популярных среди таких технологий являются LoRa (Long Range). Их появление обусловлено необходимостью подключения множества приборов учета и телеметрии для децентрализованного сбора данных на облачных серверах.

### **I. Обзор**

Технологии IoT были разработаны, чтобы облегчить и улучшить автоматизацию, удаленную работу и мониторинг в различных отраслях. За счет увеличения доступности информации с использованием сетевых датчиков. Это позволяет предпринимателям контролировать процесс отовсю-

ду. Технологии IoT позволяют управлять системами, расположенными по всему миру. В нашем случае лесу. Потенциал IoT технологий, к примеру заметила компания. Huawei которая ожидает 100 миллиардов. IoT-соединений к 2025 году.

### *Обзор LoRa*

Технология LoRa позволяет передавать данные на большие расстояния при минимальных затратах на передачу данных и организацию сети. Для многих приложений достаточно использования сети типа звезда или точка-точка. Например, сбор данных с датчиков (температура, влажность и т.д.), мониторинг состояния устройств, управление светом и проч. Стоит отметить, что при самостоятельном развертывании сети есть возможность использовать дальность технологии LoRa без использования протокола LoRaWAN. В этой статье будет рассмотрен пример использования протокола MiWi поверх модуляции LoRa.

LoRa (Long Range) - это технология модуляции сигнала физического уровня (PHY), ориентированная на устройства с питанием от батареи и передачу сигнала на большие расстояния. Технология LoRa разработана компанией Semtech и использует запатентованный метод модуляции сигнала с расширением спектра, где данные кодируются широкополосными импульсами с изменяемой частотой. Применение такого метода имеет следующие преимущества: значительно повышается чувствительность приемника (до -148дБм) и снижается критичность к расстройке по частоте между приемником и передатчиком. Технология LoRa обеспечивает связь на большие расстояния (до 20 км прямой видимости), низкое энергопотребление, высокую чувствительность приемника, низкие скорости передачи данных и защищенную передачу с шифрованием.

В сети, LoRa определяет физический уровень и верхний уровень. Уровни определяются открытым стандартом LoRaWAN — протоколом, созданным LoRa Alliance. Различные шлюзы соединяют конечные устройства к серверу IP-сети. Устройства отправляют сообщение одному или нескольким близким шлюзам. Далее шлюзы направляют пакеты на сервер, который удаляет дубликаты. Затем сервер управляет шлюзами и создает безопасность связи, после чего сервер приложения расшифровывает полученные данные и шифрует отправляемые данные на устройства. Сервер присоединения хранит root ключи устройств. Он отвечает за генерацию сеансовых ключей, используемых сетевым сервером и сервером приложения. В случае размещения на удалённом местоположении, сервер присоединения можно развернуть отдельно.

К ключевым свойствам модуляции LoRa относятся [1]:

1) Масштабируемая пропускная способность.

LoRa - модуляция может масштабироваться как по ширине полосы, так и по частоте. Он может использоваться как для узкополосного скачкообразного изменения частоты, так и для широкополосных приложений с прямой последовательностью.

2) Постоянная огибающая / малое энергопотребление.

Подобно FSK, LoRa - это схема модуляции с постоянной огибающей, которая означает, что одни и те же недорогие и маломощные высокоэффективные каскады могут быть повторно использованы без изменений.

3) Высокая надежность.

Из-за асинхронного характера сигнал LoRa очень устойчив к механизмам внутрисполосных и внеполосных помех.

4) Устойчивость к многолучевости и затуханию.

Импульс ЛЧМ является относительно широкополосным, поэтому LoRa обеспечивает устойчивость к многолучевому распространению и затуханию, что делает его идеальным для использования в городских и пригородных условиях.

5) Устойчив к эффекту Доплера.

Доплеровский сдвиг вызывает небольшой сдвиг частоты в импульсе LoRa, который вводит сравнительно незначительное смещение временной оси сигнала основной полосы частот.

6) Дальнодействие.

Для фиксированной выходной мощности и пропускной способности бюджет канала LoRa превышает бюджет обычного FSK.

7) Увеличенная пропускная способность сети.

LoRa модуляция использует ортогональные коэффициенты расширения, которые позволяют передавать несколько сигналов расширения в одно и то же время и по одному и тому же каналу без минимального ухудшения чувствительности приемника.

8) Ранжирование и локализация.

Тенденции развития науки

Неотъемлемым свойством LoRa является способность линейно различать частоту и время ошибок. LoRa является идеальной модуляцией для радиолокационных приложений и, таким образом, идеально подходит для приложений измерения дальности и локализации, таких как службы определения местоположения в реальном времени.

LoRa является реализацией уровня PHY и не зависит от реализаций более высокого уровня. Это позволяет LoRa сосуществовать и взаимодействовать с существующими сетевыми архитектурами.

LoRa-модуляция решает все проблемы, связанные с системами DSSS, чтобы обеспечить низкую стоимость, низкую мощность, но прежде всего надежную альтернативу традиционным методам связи с расширенным спектром [2].

### ***Обзор Node-RED***

Node-RED — это инструмент потокового программирования, позволяющий соединять разнообразные устройства, API-интерфейсы и онлайн-сервисы в рамках Интернета вещей («IoT») и в качестве ключевой составляющей Node-RED выступает парадигма потокового программирования. Потоковое программирование — это способ описания поведения приложения в виде сети черных ящиков или «узлов», как они называются в Node-RED. Каждый узел имеет четкую цель — к нему поступают некоторые

данные, он что-то делает с этими данными, а затем передает их на следующий узел. Сеть отвечает за поток данных между узлами. Эта модель отлично подходит для того, чтобы представить ее визуально: так она становится более доступной для широкого круга пользователей. Если кто-то пытается разобраться в проблеме, он может разбить задачу на отдельные шаги, взглянуть на поток и понять, что он делает, без необходимости разбираться в отдельных строках кода в каждом узле. Node-RED работает в среде исполнения Node.js, а для создания или редактирования потока («Flow») используется браузер [3].

Node-RED как система работает в основном в средах Linux. Присутствуют как готовые репозитории, так и наличествует возможность собственноручной компиляции. Не обойдены стороной и основные облачные платформы: IBM Cloud, AWS Cloud, Azure. При желании систему можно установить и на Android. На Windows последние редакции Node-RED работают не хуже, чем на родных ОС. Также Node-RED предлагает моделировать через консоль.

```

cmd. prn
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.

C:\Users\Kesh>node-red
22 Feb 12:42:27 - [info]

Welcome to Node-RED
=====
22 Feb 12:42:27 - [info] Node-RED version: v2.1.5
22 Feb 12:42:27 - [info] Node.js version: v13.14.0
22 Feb 12:42:27 - [info] Windows_NT 6.1.7601 x64 LE
22 Feb 12:42:31 - [info] Loading palette nodes
  
```

Рис. 1. Node-RED в консоли

## II. Расчет беспроводной линии связи

Проектирование беспроводных сетей практически невозможно без оценки пригодности линии связи, т.к. эта оценка имеет большое значение для выявления возможных проблем в ходе установки. Наличие хорошего энергетического потенциала является базовым условием для нормального функционирования линии связи [4].

Энергетический потенциал (Link budget) беспроводной линии связи учитывает все усиления и потери уровня сигнала при его распространении от передатчика к приемнику через беспроводную среду передачи, кабели, разъемы и различные препятствия (стены, потолки, деревья и т.д.).

Полное уравнение энергетического потенциала линии связи можно записать следующим образом:

$$P_{tr} - L_{tr} + G_{tr} - L_{bf} + G_{recv} - L_{recv} = SOM + P_{recv} \quad (1)$$

Где:  $P_{tr}$ – мощность передатчика, дБм (dBm);

$L_{tr}$ – потери сигнала в антенном кабеле и разъемах передающего тракта, дБ (dB);

*Gtr*– коэффициент усиления передающей антенны, дБ (dBi);  
*Lbf*– потери передачи в свободном пространстве, дБ (dB);  
*Grecv*– коэффициент усиления приемной антенны, дБ (dBi);  
*Lrecv*– потери сигнала в антенном кабеле и разъемах приемного тракта, дБ (dB);  
*SOM*– запас на замирание сигнала (SOM, System Operating Margin), дБ (dB);  
*Precv*– чувствительность приемника при данной скорости передачи, дБм (dBm).

$$20 - (-157.55) + 10 - 143.55 + 8 - 11 = 20 + (-61)$$

$$41 = 41$$

Энергетический потенциал линии связи у нас равен 41, что является стабильным результатом для беспроводной сети.

Потери в свободном пространстве для линии связи с изотропными антеннами можно рассчитать с помощью следующей формулы(2):

$$L_{bf} = 20 \lg F + 20 \lg D + K \text{ (дБ)} \quad (2)$$

$$143.55 = 20 \lg 71.76 + 20 \lg 5 + 92.45$$

143.55 это потери в свободном пространстве которые на порядок ниже чем значения у аналогов которые не используют LoRa

*Lbf*– основные потери передачи в свободном пространстве;

*F*– центральная частота канала, на котором работает система связи;

*D*– расстояние между двумя антеннами;

*K*– константа, которая зависит от единиц измерения частоты и расстояния.

Распространение любого сигнала неизбежно сопровождается его затуханием, причём величина затухания сигнала зависит как от расстояния от точки передачи, так и от частоты сигнала. При измерении в децибелах величины затухания сигнала (ослабление при распространении) пользуются формулой (3):

$$L_p = X \log \left( \frac{4\pi d f}{c} \right) \quad (3)$$

где:

*x* - коэффициент ослабления, равный 20 для открытого пространства

*d* - расстояние от точки передачи;

*f* - частота сигнала;

*c* - скорость света.

$$230 = 20 \log(4\pi * 5 * 71.76 / 299\,792\,458)$$

Коэффициент затухания 230 является нормальным значением, не сильно превосходящим аналоги.

### III. Моделирование системы ИОТ

На рисунке 2 представлена структура системы, которую мы более детально рассмотрим далее и на рисунке 3 представлена модель на платформе node-red предложенная IoT система.



Рис. 2. Модель структуры системы сенсорного покрытия леса



Рис. 3. Модель системы сенсорного покрытия леса

Модель на вид не сложна, но функциональна. Каждый заданный промежуток времени на сенсоры, расположенные в лесу отправляется запрос для получения данных. После ответа следует передача данных на сервер, где они записываются в файл. Данная функция может перезаписывать предыдущие данные или удалять их при надобности (например, при недостатке места на сервере). Чтобы получить данные с сервера будет разработан специальный интерфейс [5].

Мониторинговая сенсорная система работает при различных погодных условиях и при отсутствии электропитания. Сенсоры устанавливаются на стволе дерева на уровне 1.5 метра и передают данные на Сервер и далее

в облачную web-платформу. К web платформе можно подключиться со смартфона или при помощи специального интерфейса с персонального компьютера.

На рисунке 4 изображена БД, разработанная для хранения данных с сенсоров, Храниться будут данные с самого сенсора, время получения этих данных, типы сенсоров, а также сохраненная история полученных данных.

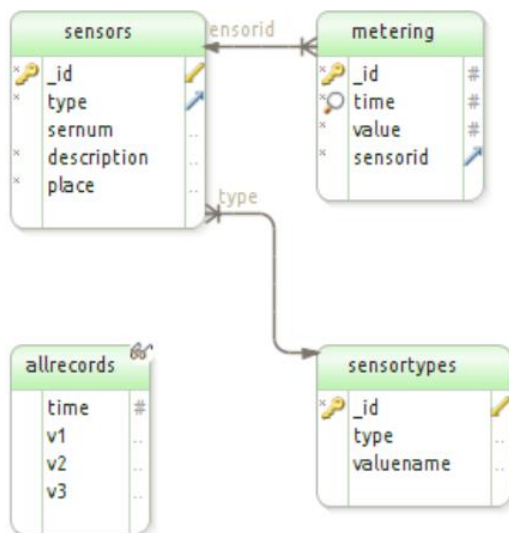


Рис. 4. Модель Базы данных сенсорного покрытия леса

## Результаты

Проведённые исследования позволили проанализировать систему LoRa. Выявить достоинства и недостатки использования в лесной отрасли. К достоинствам относится дальность передачи сигнала и низкое энергопотребление, недостатком является пропускная способность ниже, чем у конкурентов.

В результате расчётов получили значение основных потерь данных в системе IoT, значение которого составляет 143.55, можно отметить, что значение меньше, чем в среднем значение потерь сенсоров не в городской обстановке.

Смоделирована действующая сеть Интернета вещей, использующая платформу от Node-red. Приведен общий алгоритм работы с платформой. Описана структура платформы с указанием функций ее компонентов. Представлен алгоритм мониторинга информации на базе платформы Node-red.

## Заключение

Системы сенсоров в лесной отрасли становятся очень популярными. Увеличивается число лесных хозяйств, которые пользуются ими, соединенных как друг с другом, так и с сервером. Заставить все эти устройства работать в лесной среде без снижения производительности может стать сложной задачей. Инструменты симуляции помогают эффективно опреде-



лять и решать проблемы затухания сигнала, позволяя компаниям разрабатывать более надежные и энергоэффективные системы.

Смоделирована действующая сеть Интернета вещей, использующая платформу от Node-red. Приведен общий алгоритм работы с платформой. Описана структура платформы с указанием функций ее компонентов. Представлен алгоритм мониторинга информации на базе платформы Node-red.

### Библиографический список

1. Semtech Application Note AN1200.22 LoRa™ Modulation Basics [Электронный ресурс] – URL: <https://www.semtech.com/uploads/documents/an1200.22.pdf> [Дата обращения: 31.01.2022]
2. LoRaWAN™ Specification v.1.1, LoRa Alliance. [Электронный ресурс] URL: <https://lora-alliance.org/resourcehub/lorawantm-specification-v11> [Дата обращения: 1.02.2022]
3. Node-reduserguide. [Электронный ресурс] URL: <https://nodered.org/docs/user-guide/editor/workspace/nodes> [Дата обращения: 1.02.2022]
4. С. С. Владимиров БЕСПРОВОДНЫЕ СИСТЕМЫ ПЕРЕДАЧИ ДАННЫХ [Электронный ресурс] URL: [http://opds.spbsut.ru/data/\\_uploaded/mu/bspd/vlss19mu\\_bspd\\_prakt\\_802.pdf](http://opds.spbsut.ru/data/_uploaded/mu/bspd/vlss19mu_bspd_prakt_802.pdf) [Дата обращения: 1.02.2022]
5. KeysightTechnologies Моделирование систем NB-IoT. [Электронный ресурс] URL: <http://www.unitest.com/pdf/appnotes/5992-1734RURU.pdf> [Дата обращения: 1.02.2022]

**А.П. Кичук**, студентка 4 курса ИСиТ  
Кафедра информационных систем и технологий  
СПб ГЛТУ им. С.М. Кирова  
kichuk997@yandex.ru

### ВЗАИМОДЕЙСТВИЕ МИКРОСЕРВИСОВ ЧЕРЕЗ API НА ПРИМЕРЕ ERP СИСТЕМЫ

*Аннотация:* В связи с развитием информационных систем большое количество компаний и производств хотят оптимизировать и автоматизировать свои рабочие процессы с целью повышения скорости работы, сокращения персонала, а также повышения качества своей продукции, что, в итоге, приведет к увеличению прибыли. Вследствие этого, большинство компаний и, даже, учебных заведений внедряют на своих предприятиях ERP системы.

**Ключевые слова:** архитектура, внедрение, микросервисы, API.

За последнее время в России наблюдается рост IT-компаний, которые внедряют ERP системы в структуру работы своих подразделений.

ERP (EnterpriseResourcePlanning) переводится как "планирование ресурсов предприятия" и является программным обеспечением для управления бизнес-процессами, которое объединяет финансы, цепочки поставок, операции, торговлю, производство, кадры и позволяет управлять ими. Большинство компаний располагают системами управления финансами и бизнес-процессами, но возможности ПО, используемого на данный момент, как правило, ограничиваются и не взаимодействуют друг с другом [1].

На сегодняшний день представлены два вида организации приложений:

1. Монолитная архитектура.
2. Микросервисная архитектура.

### **Монолитная архитектура**

Монолитная архитектура – это архитектура, в которой приложение представлено в виде единого компонента и представляет собой разрез бизнес-логики, которая модульно прошита.

Такой тип организации приложения представлен в виде единого компонента, работа происходит в рамках одного приложения.

Плюсами такой системы являются:

1. Простота разработки и запуск готового продукта. В связи с тем, что разработка происходит в одной системе, легче внедрить инструменты для облегчения разработки, такие как библиотеки и каталоги. Также, если нужно отредактировать элементы кода, это можно сделать в одном месте, а не по отдельности, в различных местах.

2. Большое количество приложений имеют зависимость от задач, которые совершаются между компонентами программы: логи, ограничения скорости и контрольные журналы. При монолитной архитектуре эти проблемы практически отсутствуют, так как все сконцентрировано в одном коде и все работает в рамках одного приложения.

3. Если приложения были собраны корректно, то одно и то же приложение при монолитной архитектуре будет работать более производительнее, чем при микросервисах. Это обеспечивается единым кодом программы. [4]

Но, у монолитной системы есть значительные минусы:

1. Большая кодовая база. Разрабатываемый продукт довольно большой и постоянно масштабируется, со временем его код разрастается до огромных размеров. Это утяжеляет его понимание и дальнейшее обслуживание. В связи с этим может наступить момент, когда код будет «перегружен» и потеряет от этого свое качество. Также, новым разработчикам, намного сложнее погрузиться в код.

2. Когда нужно добавить в приложение какую-то новую «фичу» при монолитной архитектуре можно столкнуться с множеством препятствий, чтобы это реализовать. Связано это с тем, что в некоторых случаях добавить какую-то «фичу» означает полностью переписать приложение, что достаточно дорого и долго. При этом после деплоя необходимо проводить регресс всего приложения, а не конкретной «фичи» и смежного функционала. [4]

## **Микросервисы**

Чтобы оптимизировать работу и иметь более гибкий функционал была придумана микросерверная архитектура. [6]

Микросерверная архитектура представляет собой монолитный проект, разделенный на микросервисы, т.е., по сути, это небольшая программа, запущенная на своем сервере и работающая только над одним типом задач. Микросервисы позволяют по мере необходимости обновлять приложение по частям, а также интегрировать микросервисы в другие проекты.

Плюсами микросервисной архитектуры являются:

1. Высокая отказоустойчивость: при падении одного из сервисов, все остальные продолжают работать. Таким образом, неполадки в отдельных сервисах не мешают всему рабочему процессу.

2. Более простой код. Чем меньше код, а каждый микросервис, это отдельно работающая система, тем проще программистам разобраться, что и как работает. К тому же, на это уйдет меньше времени.

3. Небольшой размер кода обеспечивает быстрый деплой.

4. Необходимые и нужные сервисы можно дополнить и расширить, когда появится такая необходимость. Вся система при этом остается прежней.

Но, у микросервисной архитектуры тоже есть свои недостатки:

1. Так как каждый функциональный элемент изолирован, требуется особая тщательность при построении между ними грамотной коммуникации. При увеличении количества сервисов сложность в построении их общения значительно возрастает.

2. Рост числа сервисов также влечет за собой и рост числа баз данных, с которыми эти сервисы соотносятся, так как, в отличие от монолитной архитектуры, микросервисы используют не одну общую базу данных.

3. Имеются сложности при тестировании. Сложность заключается в том, что сначала нужно отдельно разобраться с каждым сервисом, а потом тестировать корректное взаимодействие его с другими микросервисами.

Микросерверная архитектура состоит из большого количества микросервисов, которые должны общаться между собой и передавать данные из одного микросервиса в другой, т.е. уметь запрашивать информацию с одного микросервиса и передавать информацию в другой микросервис. [5]

Сервисы могут общаться между собой разными способами:

1 - отправляя друг другу TCP или UDP-пакеты;

2 - через REST-интерфейс (методы по HTTP);

3 - через шину данных.

Рассмотрим подробнее REST API — принципы организации взаимодействия сервисов посредством протокола HTTP. Клиенты отправляют запросы, используя методы представленные протоколом HTTP, и выполняют операцию. Например: получение, добавление, модификация или удаление данных. [7]

API — это Application Programming Interface, или программный интерфейс приложения, с помощью которого одна программа может взаимодействовать с другой при помощи общедоступных методов, которые описаны в контрактах взаимодействия. API позволяет отправлять и получать информацию напрямую из одного микросервиса в другой, минуя интерфейс взаимодействия с пользователем.

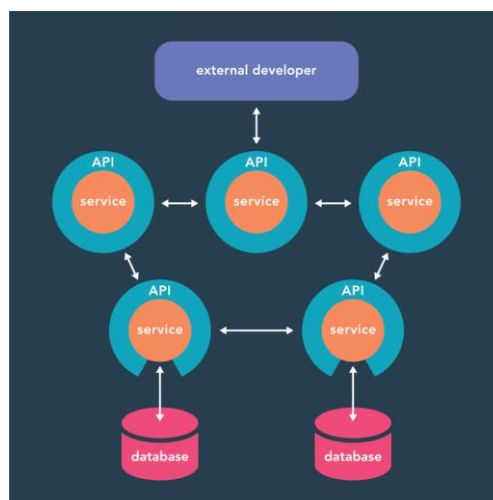
Внутри микросервиса у каждой службы есть собственный API, который определяет какие запросы она может получать и как она отвечает. Чаще всего, все сообщения между микросервисами используются в формате json. [2]

Таким образом, мы можем отправлять запросы от одного микросервиса в другой с помощью контрактов.

Сам контракт включает в себя:

- саму операцию, которую мы можем выполнить,
- данные, которые поступают на вход,
- данные, которые оказываются на выходе (контент данных или сообщение об ошибке).

На рисунке 1 представлено прямое взаимодействие микросервисов через API. Такая архитектура подходит для небольших приложений, где нет сложной логики.

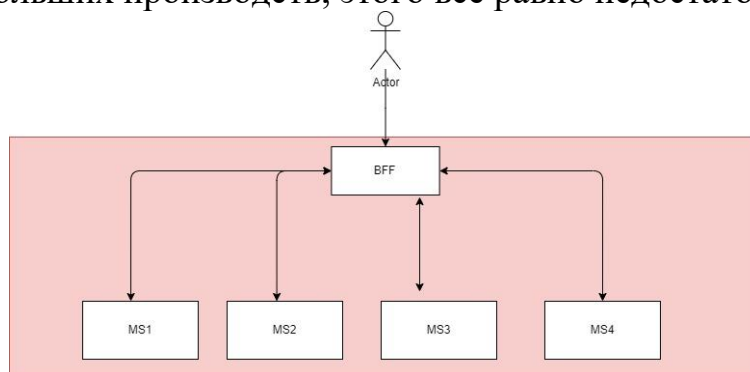


**Рис.1.** Прямое взаимодействие микросервисов через API

Главным недостатком является длительности обработки запросов. Например, клиент хочет отправить запрос на удаление товара. У товара есть цена и фотография, в таком случае, запрос отправляется в микросервис товаров, после обработки в сервисе товаров, уходит успешный результат в сервис фотографий, а потом поступает в сервис цен. Только после

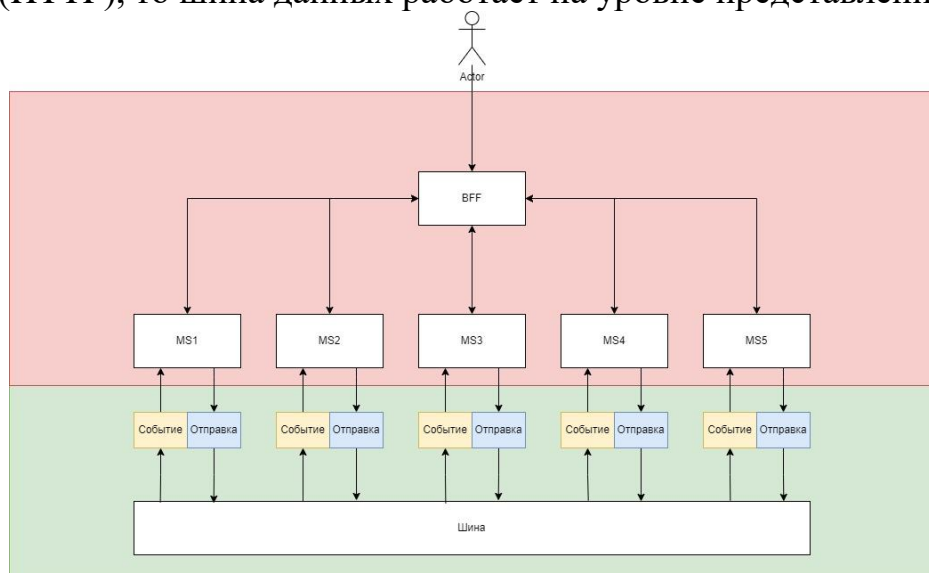
этого возвращается ответ клиенту с успешным результатом, либо с ошибкой. [2]

В связи с тем, что любое приложение должно быть быстродейственным, в архитектуру проекта добавляется прокси BFF (рисунок 2), который получает запросы и передает их в нужные сервисы. Таким образом, когда от клиента поступает запрос, он попадает в BFF, и отправляет запрос не в один микросервис, а сразу в несколько сервисов, которые обрабатывают запрос и возвращают ответ BFF. Но, BFF возвращает ответ только после обработки всех ответов микросервисов. По сравнению с первым способ он работает значительно быстрее, но для крупных приложений, интернет-магазинов или больших производств, этого все равно недостаточно. [6]



**Рис.2.** Архитектура проекта с прокси BFF

В таких случаях для достижения максимальной производительности, добавляется шина данных (рисунок 3). Шина данных является протоколом обмена информацией между сервисами без клиента и работает на другом уровне. Если BFF и запросы клиента работают на уровне транспортного протокола (HTTP), то шина данных работает на уровне представления.



**Рис.3.** Архитектура проекта с прокси BFF и шиной данных

Смысл работы заключается в том, что, когда клиент отправляет запрос, запрос попадает на BFF, BFF, в свою очередь, отправляет запрос на конкретный микросервис, например, нужно удалить товар. Запрос попада-

ет в микросервис товаров, после принятия запроса микросервисом, не дожидаясь обработки запроса, ответ отправляется в BFF, что товар удален и клиент дальше пользуется приложением, а в это время слушатель заданий микросервиса оповещает о том, что появилась новая задача. Но, так как наш товар связан с другими микросервисами, то через отправителя информация об удалении попадает в шину данных. В это время у всех остальных микросервисов есть подписка на события, и как только в шину данных попадает информация об удалении товара, у нужных микросервисов, которые взаимосвязаны с сервисом товаров, срабатывает триггер, что необходимо обработать такое событие и тоже удалить всю информацию о товаре. Причем благодаря подписке на события, все микросервисы обрабатывают события параллельно, что значительно повышает производительность, приложения. [6]

Когда система ERP может публиковать свои услуги (в сочетании с архитектурой микросервисов), она может обеспечить бесшовное соединение со сторонними системами и в то же время может расширить функции самой ERP.

Таким образом, при разработке ERP системы, необходимо четко понимать и продумывать архитектуру приложения, чтобы подобрать самый оптимальный и экономически выгодный вариант. Которые не будет требовать лишних затрат и будет обеспечивать бесперебойную и быструю работу приложения.[7]

### **Библиографический список**

1. Алгазинов, Э. К. Анализ и компьютерное моделирование информационных процессов и систем / Э.К. Алгазинов, А.А. Сирота. - М.: Диалог-Мифи, 2016. - 416 с.
2. Артамонов Ю.С. Разработка распределенных приложений сбора и анализа данных на базе микросервисной архитектуры / Ю.С. Артамонов, С.В. Востокин // Известия Самарского научного центра Российской академии наук, 2016 – т. 18.
3. Архитектуры и топологии многопроцессорных вычислительных систем. Курс лекций / А.В. Богданов и др. - М.: Интернет-университет информационных технологий, 2014. - 176 с.
4. Ильиных Г. Микросервисы или монолит: в поисках золотой – <https://blog.gramant.ru/2017/05/17/microservices-vs-monolith-compromise/> (дата обращения: 26.12.2021).
5. Костров, Б. В. Архитектура микропроцессорных систем / Б.В. Костров, В.Н. Ручкин. - М.: Диалог-Мифи, 2013. - 304 с.
6. *Ньюмен С.* Создание микросервисов = Building Microservices. — СПб.: Питер, 2016. — 304 с.
7. Шитько А.М. Проектирование микросервисной архитектуры программного обеспечения – Текст: электронный <https://cyberleninka.ru/article/v/proektirovanie-mikroservisnoy-arhitektury-programmnogo-obespecheniya> (дата обращения: 15.12.2021).

**С.С. Колмогорова**, кандидат технических наук, доцент  
Кафедра Информационных систем и технологий  
СПбГЛТУ им. С.М.Кирова  
ss.kolmogorova@mail.ru

**А.С. Колмогоров**, аспирант  
Кафедра Физики Омского ГТУ  
Омский Государственный технический университет  
ss.kolmogorova@mail.ru

**Д.С. Баранов**, аспирант  
Кафедра Физики Омского ГТУ  
Омский Государственный технический университет  
ss.kolmogorova@mail.ru

**С.В. Бирюков**, доктор технических наук, профессор  
Кафедра Физики Омского ГТУ  
Омский Государственный технический университет  
sbiryukov154@mail.ru

**О.Н. Колбина**, кандидат технических наук, доцент  
Кафедра Прикладной информатики  
Российский государственный гидрометеорологический университет  
lolikkolbik@gmail.com

**А.Д. Романовская**, студентка 3 курса  
Базовая кафедра оптоэлектроники  
Санкт-Петербургский государственный электротехнический  
университет «ЛЭТИ» имени В. И. Ульянова (Ленина)  
adromanovskaya@stud.etu

## **ПРОГРАММНО-ТЕХНИЧЕСКИЙ КОМПЛЕКС КОНТРОЛЯ И ПРОГНОЗИРОВАНИЯ ОПАСНЫХ СИТУАЦИЙ**

***Аннотация.** Развитие всех отраслей производства и глобальная цифровизация общества порождает ряд проблем, связанных с возникновением электромагнитных полей техногенной природы, их воздействие на технические и биологические объекты. В статье рассматривается измерение параметров электромагнитного поля в рамках практической задачи децентрализованной «облачной» информационно-аналитической системы с применением новых форм датчиков как IoT-элементов. Представлена программная реализация сбора данных с сенсоров новой формы, автоматическая оценка и прогнозирование. Предложена новая форма чувствительных элементов и математическая обработка, а также виртуальный прибор, реализующий платформу с последующей статистической оценкой. Интеллектуальная система обработки данных в распределенном измерительном комплексе позволяет в режиме реального времени производить оценку критических параметров электромагнитного поля.*

***Ключевые слова:** средства измерений, электрометрия, биологическая безопасность, промышленная безопасность, электромагнитная безопасность, напряженность, сенсоры электромагнитных полей, воздействие электромагнитных полей, интернет вещей, информационно-измерительная система.*

## **Введение**

Постоянный рост количества используемых электронных устройств, устойчивая вовлеченность человека в процессе использования устройств, являются характерными тенденциями современности. Техническое оснащение технологических промышленных процессов постоянно совершенствуется. Развитие всех отраслей производства и глобальная цифровизация общества порождает ряд проблем, связанных с возникновением электромагнитных полей техногенной природы, их воздействие на технические и биологические объекты. Электрометрия в Индустрии 4.0 (четвертая промышленная революция) является важным направлением. Область прикладного использования разработок электрометрии постоянно расширяется. Однако, существует большой разрыв между устоявшимися регламентирующими стандартами и нормативам ВОЗ [1], СанПиН [2, 3], ГОСТ [4] и текущим развитием измерений и информационно-измерительной техники.

Важное назначение измерителей электромагнитных полей – промышленная безопасность. Выход из строя элемента системы управления взрывоопасным производством, например нефтеперерабатывающий завод, может нести масштабные последствия, вплоть до уровня техногенной катастрофы. Все устройства автоматики, в определенной мере, подвержены влиянию помех. Помехи могут быть связаны с электропитанием, наводками в измерительных цепях, механическим воздействием, а также воздействием электромагнитных полей. Поэтому своевременное измерение электромагнитных полей влияет на промышленную безопасность.

Вопрос безопасной электромагнитной совместимости устройств рассматривается в различных нормативных документах. Современные регулирующие стандарты: европейский EN 55014-1 «Electromagnetic compatibility» и, официально используемый в Российской Федерации, технический регламент таможенного союза ТР ТС 020/2011 «Электромагнитная совместимость технических средств» [5]. Стандарты ограничивают различные виды излучений и помех от приборов, а также способность нормально функционировать при воздействии внешних излучений и помех. Использование приборов, соответствующих требованиям электромагнитной совместимости, в системах управления опасными производствами позволяет на этапе проектировки исключить риски, связанные с воздействием электромагнитных полей.

Необходимо улучшать не только метрологические и эксплуатационные свойства измерителей электромагнитных полей, но и обеспечивать их совместимость и возможность консолидирования с другими устройствами в составе IoT. Цифровизация привела к тому, что большинство управляю-



щих систем и комплексов оказываются все более высокого уровня интеграции.

Совершенствование и усложнение исследуемых приборов, наложение новых отраслевых требований постоянно повышают планку качества и гибкость измерительных приборов электромагнитных полей.

Целью данной работы является развитие направления интеграции измерителей электромагнитного поля в кластер IoT (InternetofThings)

## **1. Постановка задачи и обзор вопроса**

Электромагнитные поля присутствуют повсеместно. Естественные электрические поля создаются природными источниками, такими как накопление электрических зарядов в атмосфере, связанное с грозами. Искусственные источники: медицинское оборудование, использующее статические поля (например, МРТ), электрические приборы, использующие низкочастотные электрические и магнитные поля (50/60 Гц), а также различное беспроводное, телекоммуникационное и радиовещательное оборудование, использующее высокочастотные электромагнитные поля (100 кГц - 300 ГГц).

Требования к параметрам воздействия ЭМП (электромагнитное поле) предъявляются не только в российских [2 - 5], а также в международных стандартах. Таких как ICNIRPGuidelines [7], IEEE C95.1-2005 [8], DIRECTIVE 2013/35/EU [9]. Согласно [2 - 5] российские стандарты и нормы, регламентирующие требования к допустимым уровням напряженности электрических и магнитных полей на рабочих местах персонала электроустановок, для населения, находятся в определенном противоречии с международными [1, 6 - 9], особенно по магнитной составляющей и электромагнитному полю широкополосного спектра частот.

Существует большое многообразие измерителей ЭМП. В том числе и утвержденные типы, занесенные в Государственный реестр РФ [10]. В том числе и внесены в Федеральный информационный фонд по обеспечению единства измерений РФ некоторые импортные приборы, например серия Narda, PMM, SA, а также новые разработки на патентном этапе.

Требования ГОСТ и СанПиН предусматривает инструментальный контроль именно теми приборами, которые включены в Госреестр, в котором, основном представлены измерители с изотропными антеннами-преобразователями. Инструментальный контроль должен осуществляться приборами с допускаемой основной относительной погрешностью измерений  $\pm 10\%$ , при проведении измерений на минимальном расстоянии в 0,5 м от источника ЭМП.

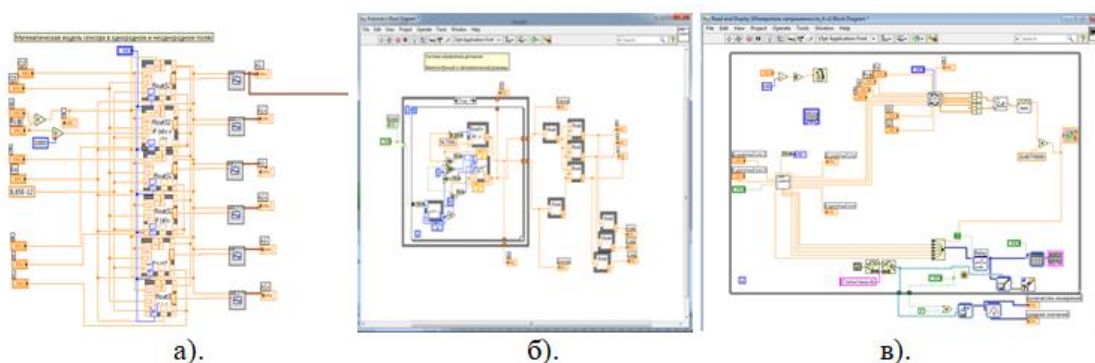
Не смотря на все многообразие приборов с различными видами датчиков напряженности ЭСП, существует необходимость в модернизации и разработке приборов с качественно лучшими параметрами и характеристиками, а также в учете существующих стандартов, регламентирующих требования к применяемым приборам и параметры воздействия ЭМП.

Кроме вышеизложенного, важно отметить, что современные тенденции развития техники диктуют применение новых технологий и материалов в конструкции чувствительных элементов измерителей ЭМП (в том числе используя технологии нанотехнологий), а также новые интеллектуальные методы получения и обработки данных в реальном времени. Необходимо создание кластер устройств IoT, постоянно оценивающий параметры ЭМП на соответствие актуальных стандартов, инструкции и программ контроля.

Концепция IoT контроля ЭМП, с использованием беспроводных датчиков, является одним из способов мониторинга, сбора, обработки и релевантного представления информации об ЭМП. Программная реализация сбора данных с сенсоров ЭМП, автоматическая оценка и прогнозирование будут предложены в настоящей статье.

## 2. Программная реализация сбора данных с сенсоров ЭМП IoT, автоматическая оценка и прогнозирование.

Структура исследовательской системы реализована в виде виртуального прибора ЭМП в среде объектно-графического программирования LabVIEW. Для апробации функционирования предложенной в настоящей статье системы выбрана среда LabVIEW, так как является удобной платформой моделирования устройств сбора и обработки измерительных данных. На рисунке 1 представлен пример реализации взаимосвязи и сбора данных [20] с нескольких сенсоров, расположенных в одной мониторинговой области. Виртуальный прибор включает алгоритмическую реализацию обработки сигналов от сенсоров и математическое моделирование средств измерения. Структура масштабируема и позволяет проводить виртуальные эксперименты как с одним, так и с несколькими сенсорами в одной области мониторинга.



**Рис. 1** а). Пример технической реализации блока нескольких сенсоров с ЧЭ в виде VI (часть блок-диаграмма) в программе LabVIEW б). система правления датчиками в). модуль автоматизации выбора и включения сенсора по заданному алгоритму

В результате проведения исследований получен ряд реляционных баз данных результатов измерений параметров ЭМП - представления форм

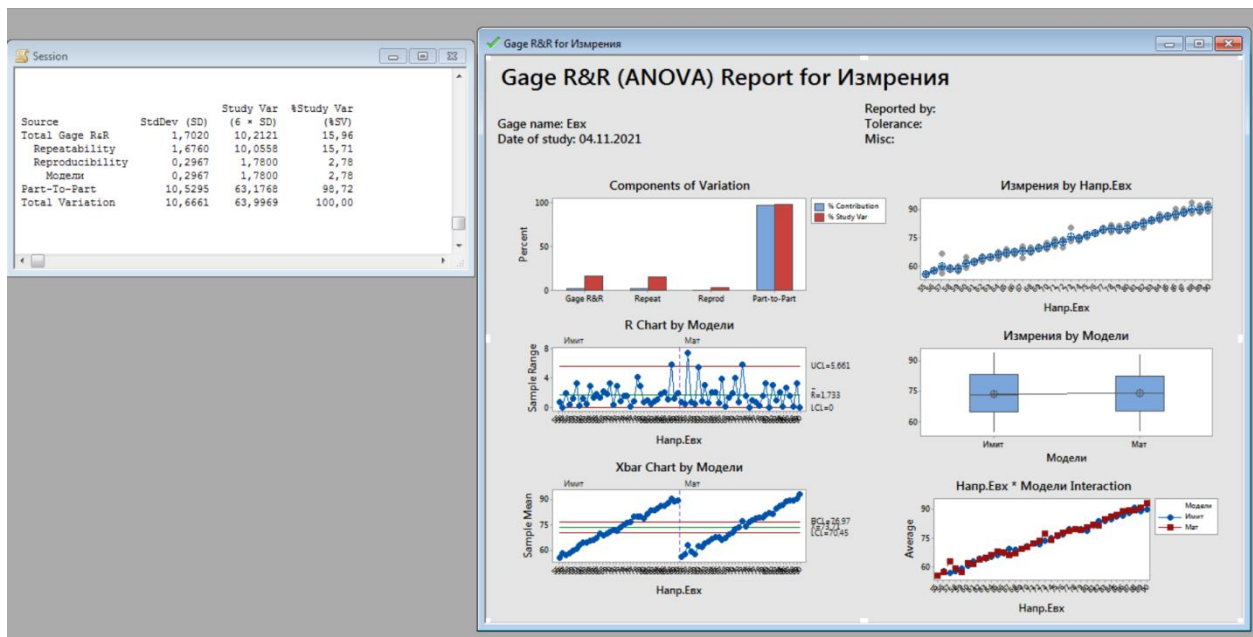
временных характеристик различными параметрами (в том числе уровень напряженности ЭМП). Данные являются достаточно информативными для кодирования, идентификации и распознавания сигналов и дальнейшей интеллектуальной обработки нейросетевыми технологиями обработки больших данных в составе информационно-измерительного IoT комплекса по оценке ЭМП на соответствие норма и правил промышленной безопасности. Результаты моделирования подтверждают возможность создания комплекса измерения параметров ЭМП на основе IoT на соответствие норма и правил промышленной безопасности

### **3. Сходимость результатов**

Проведем оценку информационной системе с конкретного применения данных с сенсоров ЭМП. Сбор данных с контролируемых датчиков о состоянии происходит в результате моделирования системы. Дадим статистическую оценку результатов имитационного моделирования электромагнитного поля с внесенным датчиком и математических расчетов. Сходимость и воспроизводилось позволит нам оценить влияние разных подходов к точности измерения.

Для оценки рассмотрим несколько аналитических серий (для разных показателей напряженности ЭМП). Проведем расчет и снимем показания измерений напряженности ЭМП на геометрической нейтрали. Проведем расчет коэффициента вариации и смещения, воспользовавшись статистическим программным продуктом *Minitab* (рисунок 2).

Для статистической характеристики методов провели оценку измеряемого напряжения точечного источника ЭМП в диапазоне [55...90] кВ, при этом получили значения с математической модели нормальной составляющей, рассчитанные в математическом редакторе по полученным новым зависимостям для ЧЭ в форме диска и значений нормальной составляющей напряженности ЭМП с расчетных данных модели Elcut. Статистическая обработка данных позволила авторам представить данные в удобном и интерактивном виде для оценки и интерпретации анализа сходимости и воспроизводимости результатов (Рисунок 2).



**Рис. 2** Модель анализа сходимости и воспроизводимости результатов моделирования экспериментальных данных новых форм ЧЭ в ЭМП

Дисперсионный анализ для расчета вариации полученных значений посредством Minitab представили в числовом и графическом виде. По диаграмме «Components of Variation» отмечены все источники вариабельности и по последней паре столбцов (а именно вариация измеренных значений) составляет 98,72 % общей вариабельности, что свидетельствует о 1,28 % влиянии методики на полученные результаты наблюдений, что не превышает и 10 % вариабельности результатов измерений. На диаграмме «R Chart по Модели» (в нашем случае выбор имитационная или математическая модели) отражены точки, которые соответствуют разнице между наибольшим и наименьшим значениями результатом измерений отдельным методом. В нашем случае наибольший размах равен 8 единиц, на графике, где разнесены методы получения данных, видим, что результаты согласованы между собой при каждом методе получения данных. На графике «Xbar Chart по Модели» сравниваем результаты с вариацией показаний. На графике «Измерения по Напр.» показаны все результаты измерений каждой отдельного значения в оцениваемом диапазоне. Измерения отображаются в виде box-диаграмм, а линией соединены средние значения полученных наблюдений. В нашем случае наблюдаются незначительные вариации между неоднократными измерениями входной величины, так как размер box-диаграммы минимален, а не закрашенные боксы каждого измерения расположены близко друг к другу. На графике «Измерения по Модели» позволяет сравнить разброс результатов, собранных разными моделями. Линией на графике отмечены средние значения всех измерений. Этот график оценивает, насколько вариаций измерений зависит от каждого метода. В нашем случае линия практически параллельна. И на графике «Напр. на Модели Interaction» показывает средние значения всех наблюдений, производимых моделями при анализе измеряемого напряжения.

Дополнительно было получено, что результаты согласованы между собой при каждом методе получения данных. Также стоит обозначить, что по полученным данным наблюдаются малые вариации между неоднократными измерениями входной величины

#### 4. Реализация архитектуры сбора данных

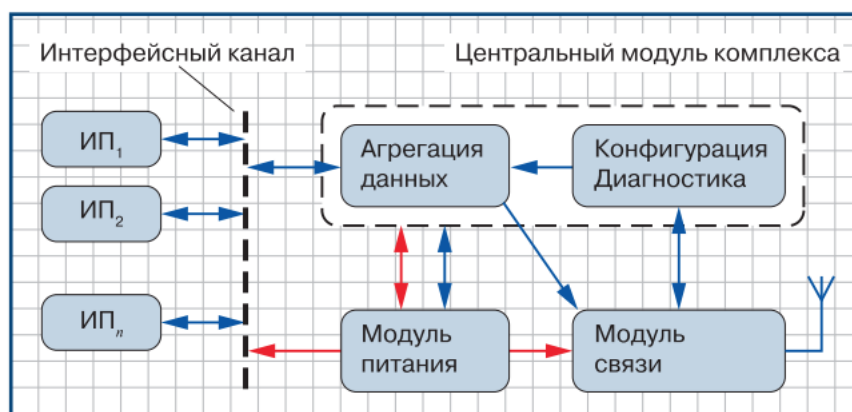
Контроль в задаче прогнозирования ЭМП на производстве для промышленной безопасности посредством децентрализованной беспроводной информационно-измерительной системы представлен на рисунке 3. Сбор данных с контролируемых беспроводных датчиков о состоянии происходит в результате взаимодействия серверной и измерительной части платформы.

Цель использования автоматизированной платформы с конструктивно улучшенными сенсорами ЭМП - измерение параметров ЭМП в режиме реального времени, с выводом целевых количественных результатов, с достаточным уровнем точности и прогнозирования опасных ситуаций.



Рис. 3 Графический пример реализации платформы контроля ЭМП

Сбор сигналов проводится с помощью автоматического диагностического комплекса- платформы сбора данных ЭМП (рисунок 4). Измерительная часть платформы принимает сигналы от датчиков, осуществляет диагностику, агрегирует данные и осуществляет обмен информацией с серверной частью платформы, по средствам беспроводной связи.



**Рис. 4** Общая архитектура сбора данных с сенсоров ЭМП. Структурная схема платформы сбора данных ЭМП

Преобразование данных измеряемых физических величин осуществляется в цифровом виде. Предполагается использовать типовые измерительные узлы, основанные на улучшенных датчиках ЭМП, с интерфейсным цифровым сигналом. Платформа, согласно заложенному алгоритму, запрашивает пакеты данных от датчика. Выполнив цикл опроса, данные с датчиков агрегируются в контейнер данных для передачи на внешний информационный ресурс. Конфигурация Платформы контроля ЭМП выполняется удаленно (или локально при подключении ноутбука или смартфона) и содержит: конфигурацию и параметры подключенных датчиков; период опроса каждого датчика; расписание суточной и календарной активности; параметры доступа к внешнему ресурсу; настройки связи; параметры электропитания. Данные с измерительной части Платформы контроля ЭМП поступают на серверную часть. Серверная часть осуществляется хранения массива данных и функционирование программных модулей: модуль мониторинга; модуль анализа; модуль сервиса Станции. Модуль мониторинга выводит данные в ранжированном, релевантном виде для пользователей. Модуль анализа осуществляет математическую обработку данных для прикладных задач, таких как статистика, прогнозирование. Модуль сервиса платформы осуществляет сбор диагностической информации о состоянии оборудования платформы и выполняет конфигурацию платформы по заданным сценариям.

Непрерывный поток данных с сенсоров в режиме онлайн и накопление данных позволяют реализовывать анализ различных параметров измерений, с формированием сигнализаций о наступившем опасном состоянии и прогнозировать потенциально неблагоприятных условий в области мониторинга, а также учет деградации оборудования измерительной части Платформы. Построение сенсорной сети и передача данных в «облачную» серверную часть Платформу легко интегрируются с различными Web сервисами.

Примирение современных телекоммуникационных систем передачи данных, совместно с автономными системами электропитания вызволяют



использовать решения автоматизированной платформы без ограничений инфраструктуры и локаций.

Обработка измерительных данных ЭМП происходит посредством алгоритмов искусственного интеллекта. Применение технологий нейросетевого анализа больших данных с сенсоров ЭМП происходит на базовых архитектурах нейросетей. Был разработан прототип информационно-аналитической системы с возможностями визуализации состояния датчиков, наблюдаемых объектов и доверительные окрестностями наблюдений сенсоров. Автоматизация процесса распределения целей сбора данных между сенсорами реализована на языке *Python*. На данный момент программный код по обработке данных информационно-аналитической системы ЭМП в региональном ФИПС.

## **5. Обсуждение результатов**

В результате проведенного исследования важно отметить востребованность задачи автоматизированной IoT платформы контроля ЭМП. Применение новых подходов к контролю ЭМП позволяет повысить уровень промышленной безопасности.

Конструкция чувствительных элементов датчика ЭМП позволяют создавать измерительный прибор в составе автоматизированной платформы с минимальной в своем классе погрешностью от неоднородности, а проведенное моделирование предложенных в статье чувствительных элементов по расчету электромагнитных задач подтверждается соотношением с математическими моделями. Сходимость и воспроизводимость результатов значений напряженности ЭМП в средах моделирования электромагнитного поля и результатов значений ЭМП новой блочной структуры математического описания предложенных ЧЭ являются удовлетворительными.

Направление IoT, с учетом тенденции развития технологий, является устойчивым для решения задачи контроля ЭМП. Возможность создавать малогабаритные измерительные узлы, на базе предложенных сенсоров, с большой автономностью и применением беспроводной связи позволяет значительно расширить зону контроля ЭМП.

Архитектура сбора и обработка данных в серверной части дает возможность применение сложных, ресурсоемких алгоритмов. Подход позволяет давать качественно лучшую информацию, по сравнению с автономными единичными измерительными приборами. Данные с сенсоров, являются непрерывными, что дает возможность их обработки с формированием сообщений об опасных состояниях превентивно.

## **Заключение**

Создание и эксплуатация автоматизированного сетевого комплекса измерения ЭМП позволяет значительно повысить быстродействия процесса измерений, повысив тем самым скорость реакции на критические, для здоровья людей и промышленных объектов, факторы.

Существует необходимость в улучшение не только метрологических и эксплуатационных свойства сенсоров и приборов, но и обеспечение их совместимость и возможности интеграции с другими устройствами в одной системе, а так же безопасность их использования для человека. Повсеместное внедрение информационных технологий привело к тому, что большинство информационных и управляющих систем и комплексов оказываются все более высокого уровня социальной и промышленной интеграции.

Применение предложенной автономной архитектуры дает возможность проводить измерения в местах, опасных для человека, и применять решение в новых, ранее не рассматриваемых, задачах. Независимость результатов от человеческого фактора повышают точность измерений.

### Библиографический список

1. World Health Organization [Электронный ресурс]: Electromagnetic fields. URL : [https://www.who.int/health-topics/electromagnetic-fields#tab=tab\\_1](https://www.who.int/health-topics/electromagnetic-fields#tab=tab_1) (дата обращения 06.02.22).
2. Электронный фонд правовых и нормативно-технических документов [Электронный ресурс]: СанПиН 2.2.4.3359-16 "Санитарно-эпидемиологические требования к физическим факторам на рабочих местах". URL : <https://docs.cntd.ru/document/420362948> (дата обращения 06.02.22).
3. Электронный фонд правовых и нормативно-технических документов [Электронный ресурс]: СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания». URL : <https://docs.cntd.ru/document/573500115> (дата обращения 06.02.22).
4. ГОСТ 12.1.006-84 Система стандартов безопасности труда. Электромагнитные поля радиочастот. Допустимые уровни на рабочих местах и требования к проведению контроля вед. 1986-01-01. [Электронный ресурс]. URL : <https://docs.cntd.ru/document/5200272> (дата обращения 06.11.21).
5. ТР ТС 020/2011 [Электронный ресурс]: Электромагнитная совместимость технических средств. URL : <https://docs.cntd.ru/document/902320551> (дата обращения 06.02.22).
6. EPRI (2010) [Электронный ресурс]: «Radio Frequency Exposure Levels from Smart Meters» Electric Power Research Institute, November 2010. URL : <https://www.bge.com/SmartEnergy/SmartMeterSmartGrid/Documents/RFSafetyWithCCST.pdf> (дата обращения 06.02.22).
7. ICNIRPGuidelines [Электронный ресурс]: руководство МКЗНИ по ограничению воздействия переменных электрических, магнитный и электромагнитных полей. URL : <http://www.icnirp.org/cms/upload/publications/ICNIRPEMFgdRus.pdf> (дата обращения 06.02.22).



8. IEEE C95.1-2005 [Электронный ресурс]: IEEE Standard for Safety Levels with Respect to Human Exposure to Radio Frequency Electromagnetic Fields. <https://ieeexplore.ieee.org/document/1626482> (дата обращения 06.02.22).
9. DIRECTIVE 2013/35/EU on the minimum health and safety requirements regarding the exposure of workers to the risks arising from physical agents (electromagnetic fields) (20th individual Directive within the meaning of Article 16(1) of Directive 89/391/EEC) and repealing Directive 2004/40/EC. [Электронный ресурс]: URL : <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32013L0035> (дата обращения 06.02.22).
10. ROSSTANDART: Federal information fund for ensuring the uniformity of measurements. Available at: <https://fgis.gost.ru/fundmetrology/registry> (accessed: February 6, 2022). (In Russ.).
11. Fang Yun-Tuan, Wang Yu-Ya, Xia Jing. Large-range electric field sensor based on parity-time symmetry cavity structure // *Acta Physica Sinica*. 2019. Vol. 68: 194201. DOI: 10.7498/aps.68.20190784.
12. When J., Yu Z., Zhang C., Mou Y., Huang Y., Zhen Z. Research of independent DC electric field sensor with wireless power supply circuit // *Journal of Engineering*. 2019, Vol. 2019 Iss. 16, pp. 929-932. DOI: 10.1049/joe.2018.8549.
13. Mou Y., Yu Z., Huang K., Ma Q., Zeng R., Wang Z. Research on a Novel MEMS Sensor for Spatial DC Electric Field Measurements in an Ion Flows Field // *Sensors*. 2018. Vol. 18. Iss. 6, Pages 1740. DOI: [org/10.3390/s18061740](https://doi.org/10.3390/s18061740).
14. Xuerong Zai, Ang Liu, Yuhua Tian, [Fanggang Chai](#) & [Yubin Fu](#). Oxidation Modification of Polyacrylonitrile-Based Carbon Fiber and Its Electro-Chemical Performance as Marine Electrode for Electric Field Test // *Journal of Ocean University of China*. 2020. Vol. 19. P. 361-368. DOI: [org/10.1007/s11802-020-4178-x](https://doi.org/10.1007/s11802-020-4178-x)
15. Barry M., Le Prado M. Device for measuring an electric field in a conducting medium and method of calibrating such a device. US patent 0238646A1; filed February 17th, 2016; published August 18th, 2016.
16. Mathieu Baicry, Matthieu Le Prado. Device for measuring an electric field in a conducting medium and method of calibrating such a device. US patent № US20160238646 A1, USA, / №US 15/045,445; filed 17.02.2016; published 18.08.2017.
17. Biryukov S.V., Kolmogorova S. S., Kolmogorov A.S. Sensor of the electric field strength vector components in the form of three mutually perpendicular square // [Journal of Physics: Conference Series](#), 2021, 1791(1), 012038. – <https://doi.org/10.1088/1742-6596/1791/1/012038>
18. S. Kolmogorova, S. Biryukov, V. V. Danshina and N. Eysmont, "Size Optimization of Sensitive Elements of an Electric Induction Disk of Electric Field Strength Sensor," 2021 International Conference on Electrical Engineering

and Photonics (EExPolytech), 2021, pp. 5-9, doi: 10.1109/EExPolytech53083.2021.9614690.

19. Biryukov, S.V. Multu-electrode sensors of the components of the electric field intensity vector in the form of the disk made of conductive material / Biryukov, S.V., Kolmogorova, S.S., Kolmogorov, A.S., Baranov, D.S / IOP Conf. Series: Journal of Physics: Conf/ Series 1441(1) (2020) p. 012037 doi: 10.1088/1742-6596/1441/1/012037.
20. Свидетельство о государственной регистрации программы для ЭВМ № 2019664188 Российская Федерация. Программный комплекс обработки показаний датчиков : № 2019663331 : заявл. 29.10.2019 : опубл. 01.11.2019 / Д. С. Баранов, С. В. Бирюков, А. С. Колмогоров, С. С. Колмогорова ; заявитель Федеральное государственное бюджетное образовательное учреждение высшего образования «Омский государственный технический университет».

**Н.В. Меламед**, кандидат технических наук, доцент  
Кафедра Информационных систем и технологий  
СПбГЛТУ им. С.М. Кирова  
natali@melamed.su

**А.А. Мамонтова**, магистрант 2 курса ИСиТ  
Кафедра Информационных систем и технологий  
СПбГЛТУ им. С.М. Кирова  
mamonto.anna@ya.ru

## **ИСПОЛЬЗОВАНИЕ МЕТОДОВ МНОГОКРИТЕРИАЛЬНОГО ПРИНЯТИЯ РЕШЕНИЙ В ЗАДАЧАХ ТРАНСПОРТНОЙ ЛОГИСТИКИ**

***Аннотация.** Статья посвящена разработке приложения поддержки принятия решения в задачах транспортной логистики на основе метода анализа иерархий.*

***Ключевые слова:** Многокритериальное принятие решений, транспортная логистика, метод анализа иерархий.*

### **Введение**

Увеличение потребности в качественных транспортно-логистических услугах, в том числе для грузовых перевозок характеризуются постоянным увеличением производственной и коммерческой деятельности предприятий.

На данный момент транспортно-логистическое обслуживание перевозок обеспечено полной технической, экономической, информационной и

правовой поддержкой для координации работы других звеньев логистической системы.

Необходимость в разработке эффективных схем доставки грузов, позволяющие сократить время доставки и затрат, связанных с доставкой, вызывается интенсивным ростом числа перевозок. Для обеспечения комплексного и качественного обслуживания грузовых перевозок, необходимо, соответствие темпа роста сопутствующих услуг и темпов роста транспортных услуг, иначе обеспечение комплексного и качественного обслуживания грузовых перевозок будет невозможно, из-за чего компании будут терять прибыль в связи с потерей клиентов.

Часть логистических операций при перемещении от источника до потребителя осуществляется при помощи разных транспортных средств. Работа транспортных компаний, как и разнообразных участников грузоперевозок, должна быть нацелена на получение единого экономического результата в логистической цепи. В данной цепи основной функцией транспорта является транспортировка груза, т.е. изменение местонахождения товаров при соблюдении принципа экономичности (сокращение стоимостных и временных затрат), при этом значимость временных затрат возрастает с появлением логистических концепций, требующих сокращения запасов, в том числе и запасов, находящихся в пути.

Огромный информационный всплеск в логистической деятельности и лесохозяйственной компонент стратегии устойчивого лесопользования выявил необходимость создания прикладных программных продуктов с функциями для реализации принятия решений при управлении логистическими процессами в лесопромышленном комплексе. Возможность реализации подобных программных продуктов тесно связана с методами многокритериального принятия решений[1-2].

В результате общего аналитического обзора проблемы многокритериального принятия решений были сделаны определённые выводы. Так, неоспоримым фактом является важность и актуальность использования методов анализа многокритериального принятия решений при управлении логистическими процессами.

Другим важным выводом является представление о том, что существующие на сегодняшний день программные средства для поддержки многокритериального принятия решения достаточно совершенны, однако не существует специализированных программных продуктов для лесопромышленного комплекса. В то же время существует потребность в таких программах, которые позволяли бы решать конкретные задачи поддержки многокритериального принятия решений для поддержки управления логистическими процессами, при этом не требовали бы дополнительного обучения персонала (как например, в случае с универсальными системами поддержки принятия решений).

По результатам анализа можно также сделать вывод о том, что наиболее удобными на сегодняшний день рабочими инструментами явля-

ются программы, рассчитанные на неподготовленного пользователя, которые не требуют дополнительного обучения.

Многокритериальное принятие решений обычно осуществляется с помощью универсальных программ. Их основные недостатки заключаются в том, что пользователю каждый раз требуется создавать иерархические структуры, критерии и альтернативы, и затем вводить их значения. Это требует определенных затрат времени на изучение программной документации. В данной работе создается специализированный программный продукт, который ориентирован на поддержку решения задач, связанных с управлением логистическими процессами, имеет специализированный интерфейс и предварительно определенные альтернативы и критерии [4-5].

Таким образом, задача создания системы поддержки многокритериального принятия решений в задачах управления логистическими процессами включает в себя следующие аспекты:

- разработка программных алгоритмов для иерархического метода поддержки многокритериального принятия решений;
- создание универсальной программы, реализующий этот метод;
- создание специализированного интерфейса для работы.

### **Формализация задачи в виде иерархической структуры**

В результате анализа методов поддержки многокритериального принятия решения был выбран метод иерархий, разработанный американским ученым Т. Саати [3].

Метод анализа иерархий (МАИ) является самым распространенным в решении многокритериальных задач в различных областях деятельности, в том числе и в управлении природными ресурсами, так как дает возможность представления проблем в виде некоторой иерархической структуры целей, подцелей, вариантов решений.

При использовании метода анализа иерархии возникает возможность рассмотрения объективной информации, экспертных знаний и субъективных предпочтений как единого целого, при сравнении альтернативных планов, в планировании лесов, возможно использование качественного критерия.

Важным фактором также является то, что МАИ прост для программной реализации, позволяет получить достоверные результаты. Применяемый в рамках иерархического метода, метод экспертных оценок не обладает избыточной сложностью, что позволяет применять его даже неподготовленному пользователю.

Математическая реализация начинается с постановки задачи и определения конечной цели – фокуса.

Строится иерархическая структура, начиная с вершины (конечной цели), через промежуточные уровни (критерии) к нижнему уровню (списку альтернатив).

Все критерии низшего уровня попарно сравниваются в матрице парных сравнений. Мы осуществляем построение множества матриц парных

сравнений для каждого из нижних уровней схемы. Элементы любого уровня сравниваются друг с другом относительно  $m$  воздействия на элемент более высокого уровня.

Парные сравнения проводятся в терминах доминирования одного элемента над другим. Эти суждения выражаются в целых числах. Если элемент А доминирует (важнее) над элементом Б, то клетка, соответствующая строке А и столбцу Б, заполняется целым числом, а клетка, соответствующая строке Б и столбцу А, заполняется обратным к нему числом (дробью). Если А и Б одинаковы, в обе позиции ставится единица.

Степень доминирования определяется по 10-бальной шкале от 1 (одинаковы) до 10 (крайняя степень превосходства).

Далее вычисляются собственные значения каждой матрицы парных сравнений и затем нормализуются.

Выполняется нормализация, целью которой является сделать равной единице сумму всех получаемых значений для каждой матрицы парных сравнений.

После проведения всех парных сравнений и вычисления собственных значений матриц определяется согласованность.

Для окончательной проверки согласованности необходимо индекс согласованности разделить на значение средней согласованности (ИСП) для случайных матриц при условии, что парные сравнения осуществляются по шкале от 1 до 10. Если отношение не превышает 20%, то можно считать, что экспертные суждения согласованны. Если оно превышает 20%, то необходимо корректировать анализируемую матрицу.

Все выше перечисленные операции проводятся для всех уровней и групп в иерархии. Если количество критериев равно  $M$ , то необходимо заполнить и обработать (получить собственные числа и индекс согласованности) ровно  $M$  матриц размерности  $N \times N$ .

Матрица парных сравнений критериев размерности  $M \times M$  обрабатывается таким же образом. В результате обработки получаем нормализованные собственные значения  $S(i)$ ,  $i = 1, \dots, M$  - матрицы критериев.

После получения нормализованных собственных значений  $M + 1$  матриц ( $M$  матриц парных сравнений альтернатив и одна матрица парных сравнений критериев) вычисляется вес каждой альтернативы.

*Постановка задачи:* определить наиболее выгодный выбор перевозчика для отправления лесоматериалов покупателю из пяти предложенных:

1. наем дополнительных работников и использование всего автопарка компании, штатных водителей и вспомогательных работников;
2. наем компании в аутсорсинг;
3. использование всего автопарка компании и штатных водителей;
4. использование всего автопарка компании, штатных водителей и вспомогательных работников;

- наем компании в аутсорсинг и использование всего автопарка компании, штатных водителей и вспомогательных работников.

Для удобства реализации метода зашифруем данные альтернативы как  $L_1, L_2, L_3, L_4, L_5$ , соответственно.

Определены критерии оценки альтернатив:

- надежность времени доставки (C1);
- тарифы доставки «от двери до двери» (C2);
- общее время транзита (C3);
- наличие дополнительного оборудования (C4);
- наличие услуг по комплектации и доставке груза (C5);
- сохранность груза (C6);
- экипирование отправок (C7);
- квалификация персонала (C8);
- сервис на линии (C9);
- отслеживание отправок (C10)
- процедура заявки транспортировки (C11);
- Расход (C12);
- Прибыль (C13).

На рисунке 1. приведена иерархическая структура элементов метода анализа иерархий для более выгодного выбора перевозчика.



Рис. 1. Иерархическая структура элементов МАИ для выбора поставщика

На первом этапе экспертами было выполнено парное сравнение критериев по методике, изложенной выше (табл.1).

Таблица 1

## Матрица парных сравнений критериев

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
C1	1,00	1,43	1,25	5,00	2,00	1,25	1,67	1,25	1,43	1,25	2,00	1,25	1,00
C2	0,70	1,00	0,88	3,50	1,40	0,88	1,17	0,88	1,00	0,88	1,40	0,88	0,70
C3	0,80	1,14	1,00	4,00	1,60	1,00	1,33	1,00	1,14	1,00	1,60	1,00	0,80
C4	0,20	0,29	0,25	1,00	0,40	0,25	0,33	0,25	0,29	0,25	0,40	0,25	0,20
C5	0,50	0,71	0,63	2,50	1,00	0,63	0,83	0,63	0,71	0,63	1,00	0,63	0,50
C6	0,80	1,14	1,00	4,00	1,60	1,00	1,33	1,00	1,14	1,00	1,60	1,00	0,80
C7	0,60	0,86	0,75	3,00	1,20	0,75	1,00	0,75	0,86	0,75	1,20	0,75	0,60
C8	0,80	1,14	1,00	4,00	1,60	1,00	1,33	1,00	1,14	1,00	1,60	1,00	0,80
C9	0,70	1,00	0,88	3,50	1,40	0,88	1,17	0,88	1,00	0,88	1,40	0,88	0,70
C10	0,80	1,14	1,00	4,00	1,60	1,00	1,33	1,00	1,14	1,00	1,60	1,00	0,80
C11	0,50	0,71	0,63	2,50	1,00	0,63	0,83	0,63	0,71	0,63	1,00	0,63	0,50
C12	0,80	1,14	1,00	4,00	1,60	1,00	1,33	1,00	1,14	1,00	1,60	1,00	0,80
C13	1,00	1,43	1,25	5,00	2,00	1,25	1,67	1,25	1,43	1,25	2,00	1,25	1,00

Для получения вектора приоритетов, содержащего коэффициенты приоритетов каждого из критериев необходимо выполнить расчеты нормализованных оценок векторов приоритетов согласно формулам, приведенным в таблице 2.

Таблица 2.

**Приближенное значение собственного столбца матрицы методом среднего геометрического**

	$L_1$	$L_2$	...	$L_n$	Оценка компонентов собственного вектора матрицы парных сравнений	Нормализованная оценка компонентов вектора приоритетов
$L_1$	$\frac{w_1}{w_1}$	$\frac{w_1}{w_2}$	...	$\frac{w_1}{w_n}$	$y_1 = \sqrt[n]{\prod_{j=1}^n \frac{w_1}{w_j}}$	$f_1 = \frac{y_1}{\sum_{i=1}^n x_i}$
...	...	...	...	...	...	...
$L_n$	$\frac{w_n}{w_1}$	$\frac{w_n}{w_2}$	...	$\frac{w_n}{w_n}$	$y_i = \sqrt[n]{\prod_{j=1}^n \frac{w_n}{w_j}}$	$f_n = \frac{y_n}{\sum_{i=1}^n x_i}$

Сумма	$\sum_{i=1}^n \frac{w_i}{w_1}$	$\sum_{i=1}^n \frac{w_i}{w_2}$	...	$\sum_{i=1}^n \frac{w_i}{w_n}$	$\sum_{i=1}^n y_i$ $= \sum_{i=1}^n \sqrt[n]{\prod_{j=1}^n \frac{w_n}{w_j}}$	1
-------	--------------------------------	--------------------------------	-----	--------------------------------	--------------------------------------------------------------------------------	---

Далее необходимо произвести оценку компонентов собственного вектора матрицы парных сравнений ( $y$ ) и нормализованную оценку компонентов вектора приоритетов ( $f$ ) каждой матрицы критериев по формулам, указанным в таблице 2.

Результаты расчета оценок для матрицы критериев приведены в таблице 3.

Таблица 3.

**Оценка компонентов собственного вектора матрицы и нормализованная оценка компонентов вектора приоритетов критериев**

Наименование критерия	$y_i$	$f_i$
Надежность времени доставки	1,51	0,11
Тарифы доставки «от двери до двери»	1,06	0,08
Общее время транзита	1,21	0,09
Наличие дополнительного оборудования	0,30	0,02
Наличие услуг по комплектации и доставке груза	0,75	0,05
Сохранность груза	1,21	0,09
Экипирование отправок	0,90	0,07
Квалификация персонала	1,21	0,09
Сервис на линии	1,06	0,08
Отслеживание отправок	1,21	0,09
Процедура заявки транспортировки	0,75	0,05
Расход	1,21	0,09
Прибыль	1,51	0,11

Наибольшее собственное значение матрицы парных сравнений может быть вычислено по формуле:

$$\lambda_{max} = L_1 \sum_{i=1}^n \frac{w_i}{w_1} + L_2 \sum_{i=1}^n \frac{w_i}{w_2} + \dots + L_n \sum_{i=1}^n \frac{w_i}{w_n}$$

$$\lambda_{max} = 13$$



Рассчитываем индекс согласованности:

$$CI = 0$$

Так как размер нашей матрицы равен 13, то случайный индекс согласованности  $CIS = 1,56$  для всех критериев.

Рассчитываем отношение согласованности:

$$CR = 0$$

Так как отношение согласованности не превышает 20%, то мы можем сделать вывод, что экспертные суждения и все матрицы парных сравнений критериев согласованы.

*На втором этапе* получаем матрицы парных сравнений альтернатив в соответствии с каждым критерием. В качестве примера рассмотрим матрицу парных сравнений по критерию «надежность времени доставки» (табл. 4).

Таблица 4

**Матрица парных сравнений для критерия «надежность времени доставки»**

Критерий «надежность времени доставки»	L <sub>1</sub>	L <sub>2</sub>	L <sub>3</sub>	L <sub>4</sub>	L <sub>5</sub>
L <sub>1</sub>	1,0	3,0	1,5	1,8	1,286
L <sub>2</sub>	0,333	1,0	0,5	0,6	0,429
L <sub>3</sub>	0,667	2,0	1,0	1,2	0,857
L <sub>4</sub>	0,556	1,667	0,833	1,0	0,714
L <sub>5</sub>	0,778	2,333	1,167	1,4	1,0

Расчет оценок для матрицы по критерию «надежность времени доставки» приведен в таблице 5.

Таблица 5

**Оценка компонентов собственного вектора матрицы и нормализованная оценка компонентов вектора приоритетов по критерию «надежность времени доставки»**

Критерий «надежность времени доставки»	$y_1$	$f_1$
L <sub>1</sub>	1.60	0.30
L <sub>2</sub>	0.53	0.10
L <sub>3</sub>	1.07	0.20
L <sub>4</sub>	0.89	0.17
L <sub>5</sub>	1.24	0.23

Всего было получено 13 матриц парных сравнений альтернатив. Все оценки можно увидеть в таблице 6.

Таблица 6

**Сводная таблица нормализованных оценок**

$L_i/C_i$	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
	0,11	0,08	0,09	0,02	0,05	0,09	0,07	0,09	0,08	0,09	0,05	0,09	0,11
L1	0,30	0,23	0,26	0,26	0,38	0,13	0,28	0,30	0,24	0,19	0,23	0,21	0,33
L2	0,10	0,19	0,12	0,14	0,04	0,19	0,10	0,10	0,09	0,29	0,19	0,26	0,04
L3	0,20	0,31	0,21	0,20	0,21	0,06	0,24	0,25	0,18	0,23	0,31	0,12	0,22
L4	0,17	0,15	0,18	0,17	0,08	0,25	0,17	0,15	0,27	0,13	0,15	0,18	0,15
L5	0,23	0,12	0,24	0,23	0,29	0,38	0,21	0,20	0,21	0,16	0,12	0,24	0,26

Используя аддитивную свертку, получены весовые коэффициенты для каждой альтернативы с учетом всех критериев (табл.7).

Таблица 7

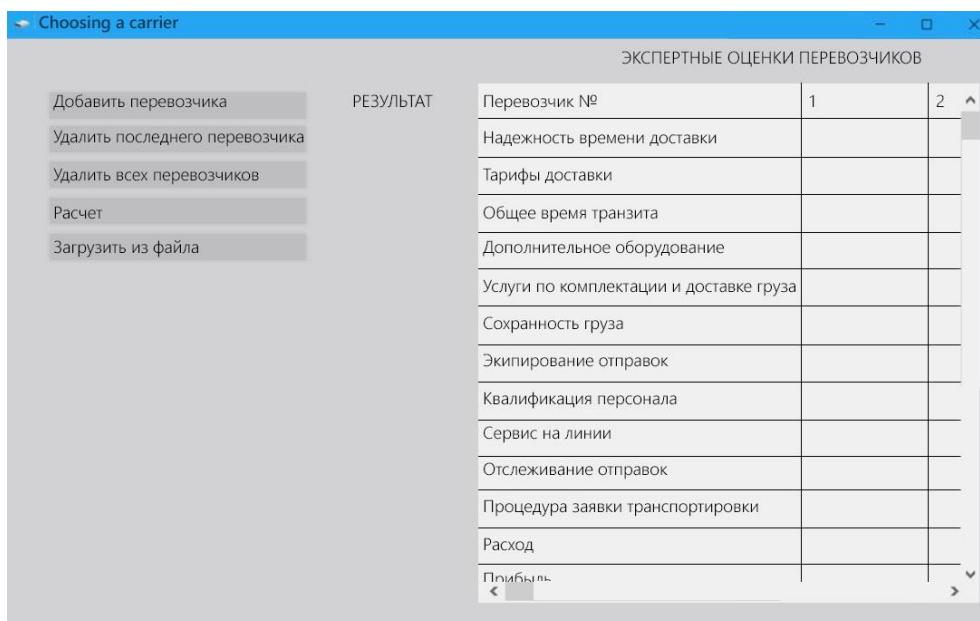
**Глобальные (обобщенные) весовые коэффициенты альтернатив**

Альтернативы	Глобальные приоритеты
Наем дополнительных работников и использование всего автопарка компании, штатных водителей и вспомогательных работников	0,26
Наем компании в аутсорсинг	0,14
Использование всего автопарка компании и штатных водителей	0,21
Использование всего автопарка компании, штатных водителей и вспомогательных работников	0,17
Наем компании в аутсорсинг и использование всего автопарка компании, штатных водителей и вспомогательных работников	0,22

### **Концепция программного средства**

Система включает в себя функции двух основных типов: интерфейсные и расчетные. Интерфейсные функции образуют интерфейсный модуль программы, основанный на двух типах диалогов: диалог на основе экранных форм и диалог типа "меню". Интерфейсные функции служат для ввода и вывода данных. Расчетные функции производят анализ исходных данных с помощью иерархического метода.

На рисунке 2. представлен общий вид главного окна программы.



**Рис. 2.** Главное окно программы

Элементами окна являются:

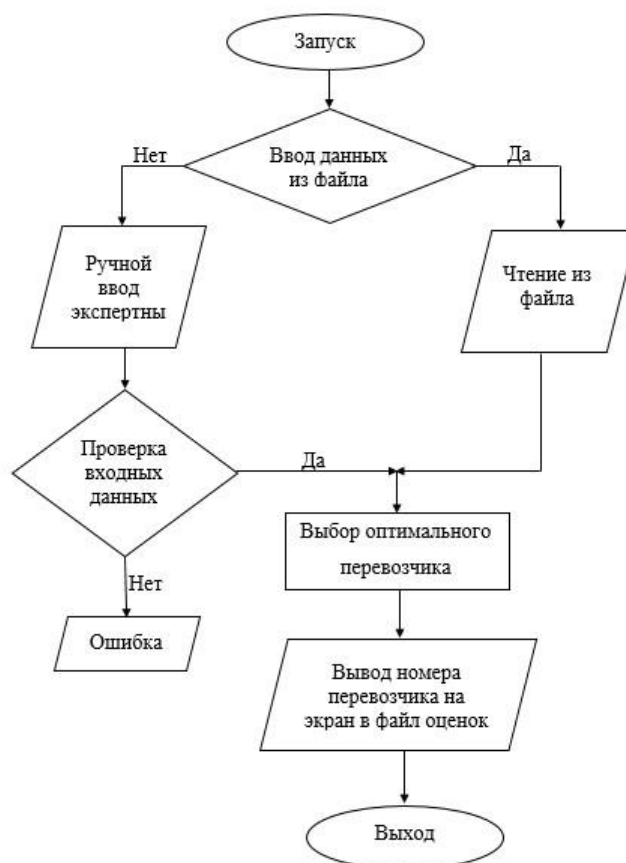
- таблица для ввода исходных данных;
- кнопка для добавления нового перевозчика (строки) в таблицу;
- кнопка для удаления последнего добавленного перевозчика из таблицы;
- кнопка для удаления всех перевозчиков (строк) из таблицы;
- кнопка запуска расчёта по данным из таблицы;
- кнопка для загрузки исходных данных из текстового файла;
- область вывода результата расчета.

#### *Функциональное назначение*

Программа предназначена для поддержки многокритериального принятия решения в задачах выбора перевозчика.

Пользователь имеет возможность выбрать наиболее лучший вариант, учитывая при этом большинство ключевых факторов, которые в свою очередь являются разнонаправленными (количественными и качественными). Выбор оптимального перевозчика производится с помощью иерархического метода поддержки принятия решений.

## Алгоритм программы



**Рис. 3.** Алгоритм работы программы

Программа в своей работе использует метод анализа иерархий.

Входными данными для программы является текстовый файл «evaluations.txt», имеющий структуру, представленную на рисунке 4.

По умолчанию текстовый файл заполнен названиями критериев через строчку. Ввод значения экспертной оценки осуществляется в пустую строку под названием критерия.

```
evaluations.txt – Блокнот
Файл  Изменить  Формат  Вид  Справка
Количество_перевозчиков
5
Надежность_времени_доставки
9 3 6 5 7
Тарифы_доставки_от_двери_до_двери
6 5 8 4 3
Общее_время_транзита
9 4 7 6 8
Наличие_дополнительного_оборудования
9 5 7 6 8
Наличие_услуг_по_комплектации_и_доставке_груза
9 1 5 2 7
Сохранность_груза
2 3 1 4 6
Экипирование_отправок
8 3 7 5 6
Квалификация_персонала
6 2 5 3 4
Сервис_на_линии
8 3 6 9 7
Отслеживание_отправок
6 9 7 4 5
Процедура_заявки_транспортировки
6 5 8 4 3
Расход
7 9 4 6 8
Прибыль
9 1 6 4 7
```

**Рис 4.** Пример заполнения файла с исходными данными

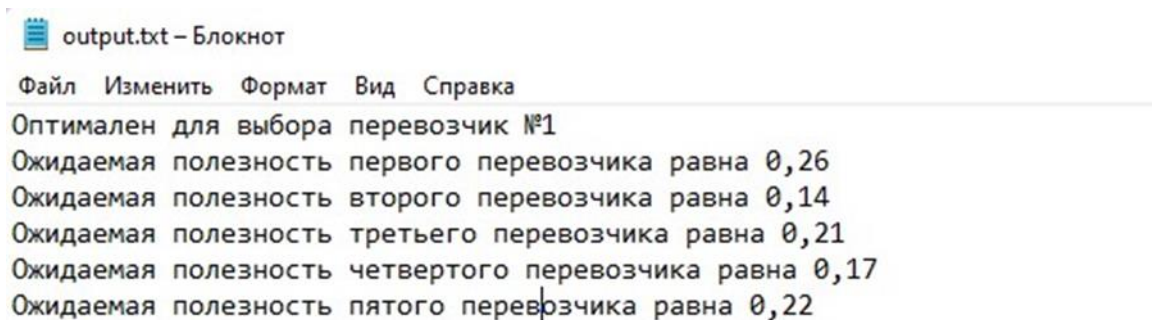
После заполнения файла «evaluations.txt» необходимо запустить программу и нажать на кнопку «Считать из файла» (элемент 4 в описании главного окна программы). После этого будет произведен расчет на основании экспертных оценок, введенных в файл.

Входные данные также можно водить непосредственно в окне программы. Для этого необходимо создать необходимое количество перевозчиков, задав их количество в ячейке, а затем нажать на кнопку «Добавить перевозчиков» чтобы получить таблицу для ввода значений нужного размера.

Выходными данными будут являться:

- выводимая текстовая информация на экране (результаты отбора оптимального перевозчика).
- файл в формате \*.txt – с номером оптимального перевозчика, в соответствии с выбранными критериями принятия решений и значениями ожидаемых полезностей для всех заданных перевозчиков.
- диагностика состояния системы и сообщения обо всех возникающих ошибках.

Для проверки работоспособности программы были введены экспертные оценки данных контрольного примера (рисунок 5).



```
output.txt – Блокнот
Файл  Изменить  Формат  Вид  Справка
Оптimalен для выбора перевозчик №1
Ожидаемая полезность первого перевозчика равна 0,26
Ожидаемая полезность второго перевозчика равна 0,14
Ожидаемая полезность третьего перевозчика равна 0,21
Ожидаемая полезность четвертого перевозчика равна 0,17
Ожидаемая полезность пятого перевозчика равна 0,22
```

**Рис. 5.** Пример файла с выходными данными

Как видно из рисунка 5 весовые коэффициенты ручного и программного расчета совпадают.

### **Заключение**

Разработанный программный продукт хорошо показал себя в решении логистической задачи выбора поставщика. Метод анализа иерархий, заложенный в его основу, легко позволит расширить функциональные возможности продукта и охватить другие задачи выбора лесопромышленной логистики.

### **Библиографический список**

1. Миротин Л.Б., Тышбаев Ы.Э. Системный анализ в логистике. М.: Экзамен, 2004. – 435с.
2. Транспортная логистика: Учебное пособие/ Под общей ред. Л.Б. Миротина. – М.: Экзамен, 2002. - 512 с.
3. Саати Т. Принятие решений. Метод анализа иерархий/Пер. с англ. Р.Г. Вачнадзе – М.: Радио и связь, 1993. – 278 с.
4. Симанова Н.В. Логистическая информационная система управления лесопромышленным предприятием// Известия Санкт-Петербургской лесотехнической академии. Сборник докладов молодых ученых на ежегодной научной конференции Санкт-Петербургской лесотехнической академии: Вып.9/ Под общей ред. А.В. Селиховкина, Э.М. Лаутнера. СПб: СПбГЛТА, 2005. – с. 62-65.
5. Симанова Н.В. Новые методы принятия многокритериальных решений при проектировании логистических цепей лесопромышленного комплекса// Сборник материалов Международной научно-практической конференции молодых ученых, проходившей 13-14 ноября 2007 года в Санкт-Петербургской государственной лесотехнической академии «Современные проблемы и перспективы рационального лесопользования в условиях рынка». – СПб.: Изд-во Политехн. ун-та, 2008. – с 221-227.

**Д.Е. Поваров**, магистрант 2 курса ИСИТ  
Кафедра Информационных систем и технологий  
СПбГЛТУ им. С.М. Кирова  
dima.povarov.91@inbox.ru

**Н.В. Меламед**, кандидат технических наук, доцент  
Кафедра Информационных систем и технологий  
СПбГЛТУ им. С.М. Кирова  
natali@melamed.su

## **РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ АНАЛИЗА И МОДЕЛИРОВАНИЯ ХОДА РОСТА ДРЕВОСТОЕВ НА БАЗЕ БД ПИК ГИЛ**

***Аннотация.** В статье рассматривается разработка программного комплекса для анализа и моделирования хода роста древостоев. Математическое моделирование ТХР осуществлено путем регрессионного анализа.*

***Ключевые слова:** моделирование хода роста древостоев, ГИЛ, регрессия.*

### **Цель исследования**

Целью данного исследования было:

- Изучение методов разработки таблиц хода роста нормальных насаждений.
- Разработка алгоритмов и программных кодов расчета коэффициентов уравнений регрессии.
- Разработка алгоритмов и программных кодов формирования таблиц хода роста нормальных насаждений.
- Получение тестовых таблиц хода роста нормальных насаждений с преобладанием основных лесообразующих пород (сосна, ель, береза, осина) Ленинградской области.
- Проведение анализа полученных тестовых таблиц путем их сравнения с общепризнанными.

### **Используемые данные**

Государственная инвентаризация лесов (ГИЛ) в Российской Федерации проводится с 2008 года. В 2020 году закончен первый цикл работ и в 2021 году начат второй.

Единицей работ (объектом измерений) по определению количественных и качественных характеристик лесов являются закладываемые в границах объектов работ постоянные пробные площади (ППП).

Количественные и качественные характеристики лесов определяются на ППП ГИЛ на основе выборочного метода наблюдений в соответствии с

методами математической статистики и взаимосвязями между таксационными показателями лесных насаждений.

Периодичность повторных наблюдений на ППП составляет 10 - 15 лет.

В состав работ по определению количественных и качественных характеристик лесов входит:

- получение и анализ исходных данных на объект работ (материалов последнего лесоустройства, данных государственного лесного реестра, иных материалов);
- получение (приобретение) материалов дистанционного зондирования Земли (цифровых космических снимков давностью съемки не более 3-х лет с учетом года проведения работ с пространственным разрешением, обеспечивающим необходимое качество проведения соответствующих работ) на 100% площади объекта работ;
- создание цифровой основы объекта работ на базе материалов лесоустройства, данных ДЗЗ, государственного лесного реестра, материалов из государственного фонда данных, полученных в результате проведения землеустройства;
- актуализация цифровой основы по данным ДЗЗ и данным государственного лесного реестра с целью внесения изменений, произошедших в лесах под воздействием антропогенных и природных факторов, после выполнения работ по таксации лесов;
- стратификация лесов;
- расчет количества ППП;
- создание генерализованной цифровой основы, изготовление актуализированных карт-схем лесных страт, вычисление площадей лесных страт;
- статистическое размещение ППП на актуализированной цифровой основе, определение местоположения (координат) ППП;
- закладка ППП и проведение на них исследований и измерений;
- проведение контрольных проверок качества полевых измерений на ППП;
- определение количественных и качественных характеристик лесов в результате комплексной обработки данных стратификации и ППП;
- анализ и обобщение результатов, подготовка ежегодных отчетов по субъектам Российской Федерации в разрезе объектов работ по результатам работ;
- определение количественных и качественных характеристик лесов посредством проведения специальных обследований в лесах, расположенных в зонах техногенного загрязнения, в которых ограничена хозяйственная деятельность.



Из перечисленных мероприятий наиболее ответственным и трудозатратным является закладка ППП и проведение на них исследований и измерений, которые и являются исходными данными для получения всей последующей информации о лесах.

В настоящей работе используются данные, полученные в ходе полевых работ при определении количественных и качественных характеристик лесов (работа базируется на предоставленных БД ГИЛ от зам. директора филиала ФГБУ «Рослесинфорг» «Севзаплеспроект» – С.Г. Кокорина).

### **Характеристика программно-измерительного комплекса ГИЛ и баз данных ГИЛ**

Программно-измерительный комплекс для осуществления государственной инвентаризации лесов [1] подразделяется на аппаратную часть и две программно-вычислительных части: полевую и камеральную. Программно-вычислительные части разработаны в среде СУБД Paradox.

Аппаратная часть состоит из набора электронных измерительных приборов, объединенных полевым компьютером в единый кластер и инструментов как основных, так и вспомогательных. В состав единого кластера входит: штатив, полевой компьютер; лазерный дальномер; электронный компас; bluetooth на компасе и компьютере; GPS; кабель для соединения GPS и компьютера, кабель для соединения лазера и компаса, кабель для соединения компаса и Bluetooth, кабель для соединения Bluetooth и компьютера; элементы питания или аккумуляторы.



**Рис. 1.** Общий вид аппаратно-измерительного кластера

Полевой частью программно-вычислительного комплекса является программное обеспечение заполнения, первичной обработки и полевого контроля БД ГИЛ и содержит следующие формы ввода информации: форма «Пробная площадь»; форма «Насаждение»; форма «Мероприятие»; форма «Почва»; форма «Деревья»; форма «Подрост»; форма «Детрит»;

форма «Привязки»; форма «Фотографии»; форма «Точки»; форма «Категории земель»; контроль базы данных и завершение измерений.

Камеральная часть программно-вычислительного комплекса является ПО тотального контроля БД, расчета производных показателей, формирование отчетных и произвольных табличных форм и обладает следующим функционалом и программными модулями: модуль проверки целостности базы данных; модуль контроля наполнения базы данных; модуль проверка логических связей в базе данных; модуль актуализация сети пробных площадей; модуль автоматизированного определения страт; модуль ввода и увязки площади страт; модуль установки размера инвентаризационной пробной площади; модуль расчета модельных высот; модуль расчета биомассы деревьев; модуль добавления виртуальных деревьев в таблицу Деревья; модуль расчета репрезентативной площади; модуль дополнения площадей отдельных страт; модуль классификации возраста по 10-летним периодам; модуль классификации деревьев по определенным ступеням толщины; модуль классификации деревьев по 4 см ступеням толщины; модуль классификации деревьев по классам размерности; модуль классификации древесных пород по группам; модуль расчета запасов углерода; модуль настройки представления статистических результатов; форма настройка графиков; модуль работы с графиками; модуль замены стратификатора в статистических задачах; модуль контроля статистических задач; модуль объединения проектов в центральную базу данных; модуль настройки веса пробной площади; модуль формирования отчетных табличных форм; модуль для формирования запросов получения произвольных табличных форм.

### **База данных ПИК ГИЛ**

В состав БД ПИК ГИЛ входит 38 содержательных, 32 справочных и произвольного количества вспомогательных таблиц.

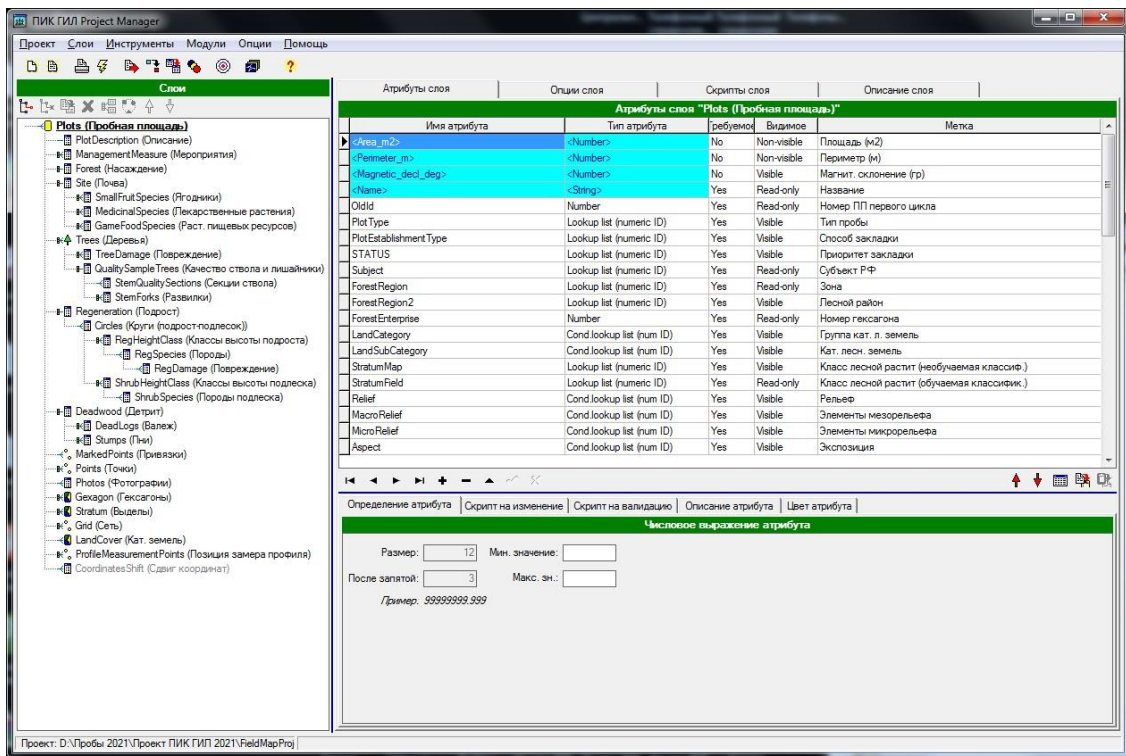


Рис. 2. Структура базы данных ПИК ГИЛ

База данных ПИК ГИЛ (рисунок 2) построена по стандартной модели организации реляционных БД, основанной на создании совокупности таблиц, распределенных в соответствии со строгой иерархией и связанных между собой отношением одна запись ко многим. С детальным описанием полей таблиц можно ознакомиться в работе [2].

При формировании ТХР по базам данных ГИЛ Ленинградской области, использованы алгоритмы, предложенные в работе ФГБУ «Рослесинфорг» «Алгоритмы обработки данных полевых измерений на постоянных пробных площадях государственной инвентаризации лесов» [3].

Математическое моделирование ТХР осуществлено путем регрессионного анализа, программные коды расчетов написаны в программной среде СУБД VisualFoxPro, графики построены в среде MicrosoftExcel, представленные фрагменты программных кодов сформированы в среде NotePad++.

Поскольку целью работ является формирование ТХР только по четырем основным лесообразующим породам Ленинградской области (сосна, ель, береза, осина) в расчет приняты ППП, в которых перечисленные породы являются доминирующими. Таких ППП – 1500 шт., в том числе с доминированием сосны – 532, ели – 398, березы – 381, осины – 189.

Первоначально для определения степени репрезентативности (представительности отобранной из всей изучаемой генеральной совокупности части в отношении общего запаса древесины) для получения статистически достоверных результатов необходимо рассчитать необходимое количество ППП полноценно отражающей свойства генеральной совокупности по статистической формуле:

$$y = \left( \frac{t \cdot \sigma}{q \cdot \bar{M}} \right)^2 \quad (1)$$

где:  $y$  – общий размер выборки;

$t$  – квантиль стандартного нормального закона ( $t = 1,96$  для 95%);

$\sigma$  – среднееквадратическое отклонение,  $\text{м}^3/\text{га}$ ;

$q$  – относительная погрешность оценки (0 ... 1);

$\bar{M}$  – среднее значение запаса древесины,  $\text{м}^3/\text{га}$ ;

Среднеквадратическое отклонение рассчитывается по формуле **Ошибка! Источник ссылки не найден.**:

$$\sigma = \sqrt{\frac{1}{(N-1)} \sum (M_i - \bar{M})^2} \quad (2)$$

где:

$N$  – количество ППП;

$M_i$  – запас на  $i$ -той ППП,  $\text{м}^2/\text{га}$ ;

$\bar{M}$  – средний по выборке запас,  $\text{м}^2/\text{га}$ .

```

PROCEDURE next02    && Расчет необходимого количества ППП в
выборочной совокупности

SELECT plots.idplots AS idplots, plots.dominpor, plots.bon AS bon,
;
  plots.area_ha AS area_ha t, 00000000000.0 AS area_ha s, ;
  SUM(trees.repre_volu*20) AS M_ha, SUM(trees.TreeNumber) AS
n_ekz ;
  FROM FORCE plots, trees;
  WHERE plots.idplots=trees.idplots AND ;

INLIST(plots.dominpor,100150,100215,302620,302688,304000)=.T.;
GROUP BY plots.landcat_zk,plots.idplots;
INTO TABLE temp\temp01

SELECT dominpor, SUM(1) AS N, 1.96 AS T, 0.1 AS q, ;
  AVG(M_ha) AS M_ha, 00000000.0 AS sigma, 000000 AS TOTAL ;
  FROM temp01 GROUP BY dominpor INTO TABLE temp\temp02

SCAN
SELECT temp01
CALCULATE STD(M_ha) FOR dominpor=temp02.dominpor TO m.sigma
SELECT temp02
REPLACE sigma WITH m.sigma
REPLACE TOTAL WITH ((T*sigma)/(q*M_ha))^2
ENDSCAN

BROWSE NOWAIT

```

**Рис. 3.** Программный код расчета необходимого количества ППП в выборочной совокупности

В таблице 1 показаны результаты расчета количества выборочной совокупности

Таблица 1

### Результаты расчета количества выборочной совокупности

Порода	Кол-во заложённых ППП	t	q	$\bar{M}$	$\sigma$	Кол-во выборочной совокупности
Сосна	532	1,96	0,1	223	138,5	148
Ель	398	1,96	0,1	293	155,8	109
Береза	381	1,96	0,1	220	146,0	169
Осина	189	1,96	0,1	408	210,0	102

ИТОГО	1500				528
-------	------	--	--	--	-----

Как видно из табл. 1, количество заложенных в Ленинградской области ППП более чем достаточно для статистических расчетов. В основе расчетов лежит изменчивость запасов древесины как показателя, имеющего наибольшую вариабельность в силу того, что данная величина зависит от всех показателей, приводимых в ТХР (возраст, высота, диаметр, количество деревьев, сумма площадей сечений, средний объем ствола, текущий прирост по запасу) и меняется при изменении любого из них.

Все дальнейшие статистические расчеты по формированию ТХР осуществляются на основе регрессионного анализа.

В соответствии с методическими рекомендациями [4] по проведению государственной инвентаризации лесов, высоты деревьев измеряются выборочно. В связи с этим в ПИК ГИЛ высоты не измеренных деревьев получают путем вычислений корреляционной зависимости между диаметром дерева и его высотой.

Зависимость высоты от диаметра, после анализа значительного объема исследовательских работ на примере таблиц хода роста из сборника А.З. Швиденко и др. [5], из стандартных функций наиболее оптимальной признана степенная функция, имеющая вид:

$$H = b \cdot D^a \quad (3)$$

где:  $H$  – высота, м;

$D$  – диаметр, мм;

$a$  и  $b$  – коэффициенты математической модели.

Прологарифмировав правую и левую части уравнения(3), приведем его к линейному виду:

$$\ln(H) = a \cdot \ln(D) + \ln(b) \quad (4)$$

Подставив вместо  $x \rightarrow \ln(D)$ , а вместо  $y \rightarrow \ln(H)$  получим:

$$a = \frac{N \cdot \sum [\ln(D) \cdot \ln(H)] - \sum \ln(D) \cdot \sum \ln(H)}{N \cdot \sum \ln(D)^2 - [\sum \ln(D)]^2} \quad (5)$$

$$\ln(b) = \frac{\sum \ln(D)^2 \cdot \sum \ln(H) - \sum \ln(D) \cdot \sum [\ln(D) \cdot \ln(H)]}{N \cdot \sum \ln(D)^2 - [\sum \ln(D)]^2} \quad (6)$$

$$b = \exp \left\{ \frac{\sum \ln(D)^2 \cdot \sum \ln(H) - \sum \ln(D) \cdot \sum [D \cdot \ln(H)]}{N \cdot \sum \ln(D)^2 - [\sum \ln(D)]^2} \right\} \quad (7)$$

Используя приведенные уравнения, получаем коэффициенты «а» и «b» уравнения регрессии зависимости высоты от диаметра.

Определение теоретических высот через регрессионный анализ осуществляется пятью итерациями, что связано с встречаемостью тех или иных родов пород при дифференциации массива деревьев по разному перечню индексных показателей.

Алгоритм процедуры сводится к следующему:

- методом SQL-запроса или другим способом (Рисунок 4) из дочерней таблицы второго уровня Trees формируется таблица накопления коэффициентов корреляционного уравнения в составе следующих полей: IdPlots – № ППП, Von – группа бони-

тета, Gen – род древесной породы, N – количество измеренных на ППП экземпляров деревьев по их родам, пустые поля a и b для наполнения их коэффициентами математической модели;

```

PROCEDURE Calc H
  USE (m.put3+'Trees') in select('Trees')           && Открытие файла
таблицы деревьев
  USE (m.put3+'Plots') in select('Plots')           && Открытие файла
таблицы ППП
  SELECT Trees                                     && Выбор раб.области
таблицы

  *** SQL-ЗАПРОС ФОРМИРОВАНИЯ ТАБЛИЦЫ ДЛЯ НАКОПЛЕНИЯ КОЭФ.РЕГРЕССИОННОГО
АНАЛИЗА
  use in select('DH1')
  select Trees
  select IdPlots, bon, substr(str(species,6),2,3) as gen, ;
         sum(1) as N, ;
         0000.000 as a, ;
         0000.000 as b, ;
         .f. as prizn ;
  from Trees ;
  where deadtree<2 and ;
         between(treebreak, 2,4)=.f. and ;
         treefork<2 ;
  group by IdPlots, bon, gen ;
  into table (put3+'diam_height1')

  use (put3+'diam_height1') in select('diam_height1') alias DH1

```

Рис. 4 Формирование таблицы для накопления коэффициентов регрессионного анализа

- при первой итерации из дочерней таблицы второго уровня Trees формируется (Рисунок 5) временная таблица MNK состоящая из следующих полей: N – количество экземпляров измеренных по высоте, D – измеренный диаметр приведенный к сантиметрам, H<sub>izm</sub> – измеренная высота, пустые поля для последующего заполнения: H – высота полученная по математической модели (для анализа), lnD – натуральный логарифм диаметра, lnH – натуральный логарифм высоты, A – коэффициент «a», B – коэффициент «b», coef – отношение измеренной высоты к полученной по MatMod;

```

*** Итерация 1-я, регрессионный анализ при наличии более 3 ;
*** измеренных по высоте экземпляров по роду в ППП
select Trees
index on str(IdPlots,9)+substr(str(species,6),2,3) to temp\Trees
select DH1
index on str(IdPlots,9)+gen to temp\DH1
set filter to n=>4  && фильтрация строк с кол-вом экземпляров от 4 и
более
m.nn11=0
sum 1 to m.nn22
m.mess=m.mess+chr(13)+chr(9)+'- 1-я итерация;'

SCAN                && Цикл перебора строк таблицы DH1
do prokr
select 1 as N, ;    && SQL-запрос формирования временной таблицы MNK
round(dbh_mm/10,1) as D, ;
height m as Hizm, ;
000.0 as H, ;
00000.000 as lnD, ;
00000.000 as lnH, ;
00000.000 as A, ;
00000.000 as B, ;
00.00 as koef ;
from Trees;
where dh1.IdPlots=IdPlots and ;
dh1.gen=substr(str(species,6),2,3) and ;
height_m>0 and ;
dbh_mm>0 and ;
deadtree<2 and ;
between(treebreak, 2,4)=.f. and ;
treefork<2 ;
into table temp\mnk
do mnk              && Переход на модуль расчета коэффициентов MatMod
go top
select dh1
replace a with mnk.a  && Перенос в накопительную таблицу коэф. A
replace b with mnk.b  && Перенос в накопительную таблицу коэф. B
replace prizn with iif(a+b#0,.T.,.F.)
ENDSCAN

```

Рис. 5. Итерация 1-я, регрессионный анализ при наличии более 3 измеренных по высоте экземпляров по роду в ППП

- во временную таблицу вносятся записи экземпляров деревьев по условию заложенному в операторе Where: № ППП в DH1 равен номеру ППП в Trees, род породы в DH1 равен роду породы в Trees три цифры из поля Species начиная со второй – substr(str(species,6),2,3), высота и диаметр в Trees больше 0, экземпляры полноценные и здоровые;
- временная таблица MNK передается на обработку в модуль MNK – метод наименьших квадратов (Рисунок 6);

```

*** -----Процедура расчета коэффициентов регрессионного уравнения
PROCEDURE mnk
replace lnD with log(D) all && заполнение поля lnD логарифмом диаметра
replace lnH with log(Hizm) all && заполнение поля lnH логарифмом высоты
sum 1, lnD, lnH, lnD*lnH, lnD^2 to m.n, m.D,m.H, m.xy,m.D2 && Переменные
линейной регрессии,
if m.n=>4 and (m.n*m.D2-m.D^2)>0
m.a=(m.n*m.xy-m.D*m.H)/(m.n*m.D2-m.D^2)
m.b=exp((m.D2*m.H-m.D*m.xy)/(m.n*m.D2-m.D^2)) && Для степенной функции
replace a with m.a all
replace b with m.b all
replace H with b*D^a all
endif
RETURN

```

Рис. 6. Процедура расчета коэффициентов регрессионного уравнения

- модуль MNK заполняет поля lnH & lnD натуральными логарифмами соответствующих показателей;



- далее рассчитывает переменные, являющиеся суммами для использования в формулах (5) и (6).
- при условии, что экземпляров деревьев более 3 и знаменатель формул (5) и (6) более 0, делается расчет коэффициентов А & В
- поле Н заполняется для возможности проведения анализа между измеренной высотой и полученной по математической модели;
- предыдущим кодом (Рисунок 5) коэффициенты «а» и «b» переносятся в накопительную таблицу DN1, где в поле prizn ставится отметка о заполненности полей А и В;
- отличительной особенностью второй итерации от первой заключается в отсутствии № ППП в качестве дифференцирующего массива деревьев показателя, т.е. дифференциация массива осуществляется по бонитету и роду породы всех деревьев без учета их принадлежности к той или иной ППП;
- при второй итерации от таблицы DN1, SQL-запросом формируется временная таблица-курсор temp01 содержащей следующие поля: Bon – группа бонитетов ППП, Gen – род породы, n1 – количество экземпляров деревьев, при формировании temp01, поля bon и gen являются группирующими;
- по аналогии с первой итерацией, из таблицы Trees формируется временная таблица MNK с тем отличием, что в операторе Where отсутствует условие № ППП в DN1 равно номеру ППП в Trees, то есть, в расчет отбираются деревья без учета их расположения в ППП с дифференциацией их массива по бонитету ППП и роду породы;
- дальнейший порядок действия аналогичен первой итерации;
- при третьей итерации от таблицы DN1, SQL-запросом формируется временная таблица-курсор temp02 содержащей следующие поля: Gen – род породы, n2 – количество экземпляров деревьев, при формировании temp02, поле Gen является группирующим;
- по аналогии со второй итерацией, из таблицы Trees формируется временная таблица MNK с тем отличием, что в операторе Where отсутствует условие по отбору бонитета ППП в Trees, то есть, в расчет отбираются деревья без учета их расположения в ППП и класса бонитета с дифференциацией их массива только по роду породы, таким образом итерация предназначена для определения коэффициентов MatMod для достаточно редко встречающихся в Ленинградской области родов пород;
- дальнейший порядок действия аналогичен первой и второй итерациям;



- при четвертой итерации от таблицы DN1, SQL-запросом формируется временная таблица-курсор temp02 содержащей следующие поля: bon – бонитет, n2 – количество экземпляров деревьев, при формировании temp02, поле bon является группирующим;
- по аналогии с третьей итерацией, из таблицы Trees формируется временная таблица MNK с тем отличием, что в операторе Where вместо условия по отбору рода породы, применяется условие по отбору бонитета ППП в Trees, то есть, в расчет отбираются деревья без учета их расположения в ППП и бонитета ППП с дифференциацией их массива только по роду породы, таким образом итерация предназначена для определения коэффициентов MatMod для единично встречающихся в данном регионе рода пород;
- дальнейший порядок действия аналогичен первой, второй и третьей итерациям;
- пятая итерация предназначена для тех родов пород, которые в массиве информации встречается не более 3-х раз
- при пятой итерации строки таблицы DN1 отбираются в расчет по условию поля przn имеющее логическое выражение False (.F.);
- по аналогии с предыдущими итерациями, из таблицы Trees формируется временная таблица MNK со всем массивом экземпляров деревьев без учета их рода, бонитета ППП и их привязки к № ППП, соответственно в операторе Where заложены условия по отбору здоровых без сломов и многовершинности экземпляров;
- дальнейший порядок действия аналогичен предыдущим итерациям;
- по результату пяти итераций определения коэффициентов математической модели, сформирована таблица Diam\_Height1 (alias – DN1) представляющая собой сбор коэффициентов степенной функции математических моделей по всем родам деревьев Ленинградской области дифференцированных по классам бонитета. Вид таблицы представлен на рисунке 7.

Idplots	Bon	Gen	N	A	B	Prizm
29138603	3	004	7	0.726	1.756	T
29138603	3	026	1	0.734	1.918	T
29138604	3	004	2	0.726	1.756	T
29138604	3	026	8	0.734	1.918	T
29140501	5	004	10	0.741	1.604	T
29140501	5	026	1	0.714	1.944	T
29140501	5	040	5	0.607	2.109	T
29140503	3	004	8	0.726	1.756	T
29140503	3	026	2	0.734	1.918	T
29140504	3	001	1	0.676	1.977	T
29140504	3	004	1	0.726	1.756	T
29140504	3	026	1	0.734	1.918	T

Рис. 7. Таблица Diam\_Height1 (alias – DH1)

- в цикле, путем перебора строк таблицы DH1 по диаметру, приведенному к сантиметрам (поле – DBH\_mm, таблицы Trees) рассчитывается теоретическая высота и вкладывается в поле CalcHeight, таблицы Trees для здоровых, без сломов и многовершинности экземпляров деревьев.

### Формирование тестовых таблиц хода роста нормальных насаждений

При формировании ТХР нормальных насаждений использован метод Баура (метод полосок) в виду того, что данный способ наиболее простой и технология закладки ППП ГИЛ полностью отвечает потребностям для этого метода. Для расчета автором разработан программный код приведенный ниже.

```

SELECT plots.idplots AS idplots, plots.bon AS bon, ;
INT (plots.dominpor/100) *100 AS dominpor, ;
INT (trees.species/100) *100 AS Gen, ;
SUM (ROUND (trees.age*trees.repre_volu*20,0)) AS age, ROUND (trees.age/10,0) AS pag_age, ;
SUM (trees.TreeNumber) AS kolvo, ;
SUM (ROUND (trees.TreeNumber/repretreen,0)) AS n_ekz, ;
plots.area_ha AS area_ha_t, ;
AVG (ROUND (trees.CalcHeight,1)) AS H, ;
0000.0 AS D, ;
00.000 AS V, ;
SUM (PI () * (trees.dbh_mm/2000) ^2) AS G, ;
SUM (ROUND (trees.repre_volu*20,0)) AS M_ha;
FROM FORCEplots, trees;
WHERE plots.idplots=trees.idplots AND ;
INLIST (plots.dominpor,100150,100215,302620,302688,304000)=.T. ;
AND trees.iar<4 AND plots.poln1>0.25 AND BETWEEN (plots.bon,3,7)=.T. ;
GROUP BY plots.idplots, plots.bon, Gen ;
INTO TABLE temp\temp01

REPLACE age WITH age/M_ha FORM M_ha>0
REPLACE pag_age WITH ROUND (age/10,0) ALL
REPLACE G WITH G/kolvo FOR kolvo>0
REPLACE D WITH (G/PI ()) ^0.5*200 ALL

SELECT plots.idplots, plots.poln1, INT (plots.dominpor/100) *100 AS dominpor, ;
SUM (n_ekz) AS n_ekz, ;
SUM (ROUND (G*n_ekz,1)) AS G, ;
SUM (M_ha) AS M_ha ;
FROM FORCEplots, temp01 ;
WHERE plots.idplots=temp01.idplots ;

```

```

GROUPBYplots.idplots ;
INTOCURScurs01

SELECTtemp01
BLANKFIELDn_ekz,G,M_haALL
DELETEDFORdominpor#Gen

SELECTcurs01
SCAN
SELEtemp01
LOCATEFORidplots=curs01.idplots
REPLACEn_ekzWITHcurs01.n_ekz/curs01.poln1
REPLACEGWITHcurs01.G/curs01.poln1
      REPLACEM_haWITHcurs01.M_ha/curs01.poln1
ENDSCAN
SELECTtemp01
REPLACEVWITHM_ha/n_ekzALL
REPLACEbonWITHbon-2ALL

&&----- Обработка
per1='H'
SELECTdominpor,por.N_Por,pag_age*10ASage, ;
ROUND(SUM(IIF(bon=1,&per1,0.0))/SUM(IIF(bon=1,1,0)),1)ASPok_1, ;
0000.000ASpok_1a, ;
ROUND(SUM(IIF(bon=2,&per1,0.0))/SUM(IIF(bon=2,1,0)),1)ASPok_2, ;
0000.000ASpok_2a, ;
ROUND(SUM(IIF(bon=3,&per1,0.0))/SUM(IIF(bon=3,1,0)),1)ASPok_3, ;
0000.000ASpok_3a, ;
ROUND(SUM(IIF(bon=4,&per1,0.0))/SUM(IIF(bon=4,1,0)),1)ASPok_4, ;
0000.000ASpok_4a, ;
ROUND(SUM(IIF(bon=5,&per1,0.0))/SUM(IIF(bon=5,1,0)),1)ASPok_5, ;
0000.000ASpok_5a, ;
FROMtemp01,por ;
WHEREtemp01.dominpor=por.SpeciesGIL ;
GROUPBYdominpor,pag_age ;
INTOTABLEtemp\temp02

REPLACEPok_1WITH0FORPok_1>999
REPLACEPok_2WITH0FORPok_2>999
REPLACEPok_3WITH0FORPok_3>999
REPLACEPok_4WITH0FORPok_4>999
REPLACEPok_5WITH0FORPok_5>999

SELECTdominpor,N_PorFROMtemp02GROUPBYdominporINTOCURScurs01
SCAN
SELEtemp02
FORi=0TO130STEP10
LOCATEFORdominpor=curs01.dominporANDage=i
IFFOUND()=.F.
APPENDBLANK
REPLACEdominporWITHcurs01.dominpor
REPLACEN_PorWITHcurs01.N_Por
REPLACEageWITHi
ENDIF
ENDFOR
ENDSCAN
SELECTtemp02
DELETEDFORage>130ANDINLIST(dominpor,100100,100200)=.T.
DELETEDFORage>100ANDdominpor=302600
DELETEDFORage>80ANDdominpor=304000

INDEXONSTR(dominpor,6)+STR(age,3)TOtemp\temp02
CREATETABLEtemp\mnk(ageN(3),PokN(5,1),Pok1N(5,1), ;
lnAN(10,3),lnPokN(10,3),aN(10,3),bN(10,3))
SELECTcurs01
SCAN
SELECTtemp02
m.nn33=0
FORM.nn22=4TO12STEP2
m.nn33=m.nn33+1
SELECTmnk
ZAP

```

```

SELECT temp02
per='temp02.'+FIELD(m.nn22)
per1='temp02.'+FIELD(m.nn22+1)
SETFILTER TO dominpor=curs01.dominpor AND &per>0
SCAN
SELECT mnk
APPEND BLANK
REPLACE age WITH temp02.age/10
REPLACE Pok WITH &per
ENDSCAN
DO mnk
GOTO P
SELECT temp02
SETFILTER TO dominpor=curs01.dominpor
REPLACE &per1 WITH mnk.b*(age/10)^mnk.a ALL
ENDFOR
ENDSCAN
SELECT temp02
SETFILTER TO
DO excel
RETURN
PROCEDURE excel

obj1=CREATEOBJECT('Excel.Application')
obj1.APPLICATION.VISIBLE=.T.
obj1.workbooks.ADD
SELECT temp02
m.nn11=1
SCAN
m.nn11=m.nn11+1
FOR i=1 TO 3
per=FIELD(i)
obj1.cells(m.nn11,i).VALUE=&per
ENDFOR
FOR i=4 TO 13
per=FIELD(i)
obj1.cells(m.nn11,i).VALUE=IF(&per>0,&per,'')
ENDFOR
ENDSCAN

GOTO P
BROWSE NOWAIT
RETURN

PROCEDURE mnk
SELECT mnk
BLANKFIELD lnA,Pok1,lnPok,a,b ALL

REPLACE lnA WITH LOG(age) ALL && key изменять в 3-х местах это 1-е.
&& Степенная функция используется с последующим
REPLACE lnPok WITH LOG(Pok) ALL

SUM1,lnA,lnPok,lnA*lnPok,lnA^2;
TO m.n,m.a,m.Pok,m.xy,m.A2 && Линейная регрессия, для степенной функции

IF m.n=>4 AND (m.n*m.A2-m.a^2)>0
m.a=(m.n*m.xy-m.a*m.Pok)/(m.n*m.A2-m.a^2)
m.b=EXP((m.A2*m.Pok-m.a*m.xy)/(m.n*m.A2-m.a^2)) && Для степенной функции
REPLACE a WITH m.a ALL
REPLACE b WITH m.b ALL
REPLACE Pok1 WITH b*age^a ALL
ENDIF
RETURN

```

### Алгоритм расчета:

1. Из базы данных ППП ГИЛ, SQL-запросом формируется рабочая таблица первичной информации (Рисунок 8)

Idplots	Bon	Dominpor	Gen	Age	Pag_age	Kolvo	N_ekz	Area_ha_t	H	D	V	G	M_ha
34501001	7	100200	100200	97	10	6	740	5407.00	13.95	19.8		0.030764	83
34501001	7	100200	302600	79	8	4	160	5407.00	16.15	24.1		0.045455	48
34501003	3	100100	100100	59	6	6	120	2478.00	20.60	25.6		0.051587	62
34501003	3	100100	100200	75	8	4	240	2478.00	17.30	21.4		0.036133	48
34501004	4	100200	100200	74	7	13	880	3500.00	17.74	22.9		0.041062	140
34501004	4	100200	302600	69	7	4	80	3500.00	21.35	24.5		0.047141	40
34501004	4	100200	304000	70	7	7	140	3500.00	22.49	25.7		0.051840	78
34501004	4	100200	511800	71	7	1	100	3500.00	16.00	17.4		0.023779	17
34501005	4	302600	100200	80	8	6	200	3587.00	20.80	25.9		0.052777	79
34501005	4	302600	302600	79	8	14	360	3587.00	24.66	25.9		0.052542	197
34501005	4	302600	304000	80	8	2	40	3587.00	32.90	42.0		0.136615	87
34501007	5	100100	100100	69	7	10	280	2478.00	17.78	23.2		0.042254	92
34501007	5	100100	302600	66	7	2	800	2478.00	8.90	7.5		0.004414	17
34501009	5	100100	100100	90	9	13	260	2101.00	19.89	26.1		0.053352	136
34501009	5	100100	100200	91	9	3	60	2101.00	17.13	22.8		0.040710	21
34501009	5	100100	302600	84	8	4	1102	2101.00	11.83	10.9		0.009251	39
34501009	5	100100	304000	89	9	3	60	2101.00	24.83	33.8		0.089955	65
34501014	5	100200	100100	42	4	1	100	3587.00	14.90	15.0		0.017671	13
34501014	5	100200	100200	74	7	7	220	3587.00	17.27	24.6		0.047377	73
34501014	5	100200	302600	26	3	3	1405	3587.00	8.75	4.6		0.001665	13
34501014	5	100200	304000	90	9	1	20	3587.00	22.00	61.8		0.299962	62
34501014	5	100200	304400	20	2	1	400	3587.00	10.00	9.3		0.006793	11
34501015	5	100100	100100	84	8	13	340	2101.00	17.34	24.7		0.047831	120
34501015	5	100100	302600	61	6	4	842	2101.00	13.45	16.5		0.021312	32
34501015	5	100100	304100	47	5	1	100	2101.00	12.80	12.6		0.012469	8
34501015	5	100100	304400	48	5	2	200	2101.00	13.25	13.8		0.014873	16
34501016	6	100200	100200	52	5	12	2160	3500.00	12.77	15.4		0.018565	125
34501016	6	100200	302600	90	9	3	60	3500.00	23.77	30.6		0.073335	49
34501016	6	100200	304000	90	9	1	20	3500.00	31.00	55.8		0.244545	71

Рис. 8. Фрагмент первичной рабочей таблицы

Перечень полей таблицы: Idplots – номер ППП; Bon – код класса бонитета; DominPor – код доминирующей на ППП порода; Gen – коды составляющих на ППП пород; Age – возраст составляющей породы; Pag\_Age – номер 10-летнего возрастного периода; Kolvo – количество деревьев на ППП; N\_ekz – количество деревьев приведенной к 1-му га; Area\_ha – весовая площадь ППП; H – средняя высота составляющей породы; D – средний диаметр составляющей породы; V – средний объем ствола (на этом этапе не заполнен) G – средняя площадь сечения ствола; M\_ha – запас древесины составляющей породы на 1 га.

1. Формируется вторая рабочая таблица-курсор (Рисунок 9)

Idplots	Poln1	Doming	N_ekz	G	M_ha
34501001	0.54	100200	900	30.1	131
34501003	0.32	100100	360	14.9	110
34501004	0.73	100200	1200	49.6	275
34501005	0.69	302600	600	35.0	363
34501007	0.38	100100	1080	15.3	109
34501009	0.77	100100	1482	31.9	261
34501014	0.57	100200	2145	23.2	172
34501015	0.66	100100	1582	40.6	176
34501016	0.74	100200	2360	51.9	260
34501017	0.80	302600	1460	36.2	281
34501020	0.42	302600	880	28.4	149

Рис. 9. Вторая рабочая таблица

В таблице по номеру ППП сгруппированы суммы: количества деревьев на 1 га, сумма площадей сечений на 1 га, запас древесины на 1 га и определена относительная полнота по доминирующей породе.



2. Вложение указанных показателей в первичную таблицу с приведением их величины к нормальному насаждению, то есть, к относительной полноте 1,0, удаление из первичной таблицы второстепенных пород, ППП с полнотой менее 0,25 и ППП с классами бонитета: 1б, 1а, 5а, 5б. В результате получаем преобразованную первичную таблицу (рисунок 10), в которой коды классов бонитета преобразуются непосредственно в классы и рассчитывается средний объем ствола доминирующей породы.

Idplots	Bon	Dominpc	Gen	Age	Pag_age	Kolvo	N_ekz	Area ha t	H	D	V	G	M ha
34501001	5	100200	100200	97	10	6	1667	5407.00	13.95	19.8	0.146	55.740241	243
34501003	1	100100	100100	99	6	6	1125	2478.00	20.60	25.6	0.306	48.562500	344
34501004	2	100200	100200	74	7	13	1644	3500.00	17.74	22.9	0.223	67.945205	377
34501005	2	302600	302600	79	8	14	870	3587.00	24.66	25.9	0.605	50.724638	526
34501007	3	100100	100100	69	7	10	2842	2478.00	17.78	23.2	0.101	40.263158	287
34501009	3	100100	100100	90	9	13	1925	2101.00	19.89	26.1	0.176	41.428571	339
34501014	3	100200	100200	74	7	7	3763	3587.00	17.27	24.6	0.080	40.701754	302
34501015	3	100100	100100	84	8	13	2397	2101.00	17.34	24.7	0.111	61.515152	267
34501016	4	100200	100200	52	5	12	3189	3500.00	12.77	15.4	0.110	70.125138	361
34501017	1	302600	302600	55	6	16	1825	3410.00	22.45	23.0	0.192	45.250000	351
34501020	1	302600	302600	56	6	5	2096	2589.00	23.94	25.8	0.189	67.619048	395
34501024	2	302600	302600	80	8	8	792	2400.00	21.51	26.7	0.352	36.041667	279
34501025	2	302600	302600	90	9	29	1431	3587.00	23.10	24.0	0.278	55.490196	398
34501027	1	304000	304000	75	8	4	1458	5845.00	26.40	34.4	0.257	46.949153	375
34501028	3	302600	302600	80	8	3	1643	2400.00	18.67	19.1	0.170	37.142857	279
34501031	3	100200	100200	90	9	15	706	5407.00	21.17	25.1	0.569	36.470588	402
34501037	4	302600	302600	51	5	4	3950	1457.00	11.75	9.6	0.041	29.722222	161
34501038	3	100200	100200	75	8	14	1970	3500.00	17.36	20.9	0.203	75.757576	400
34501040	4	302600	302600	90	9	5	1073	2400.00	19.42	19.5	0.346	42.682927	371
34501041	5	302600	302600	96	10	11	3034	2400.00	13.32	15.7	0.074	59.310345	226
34501042	1	100200	100200	110	11	2	333	5407.00	30.55	46.0	1.709	42.142857	569
34501043	4	302600	302600	89	9	6	3577	2400.00	15.75	19.0	0.063	38.360656	225
34501045	3	100200	100200	100	10	11	909	3112.00	19.65	25.4	0.413	47.045455	375
34501046	3	302600	302600	90	9	11	2000	3587.00	21.88	23.3	0.236	127.361111	471
34501049	2	302600	302600	90	9	12	1489	3587.00	22.49	26.1	0.241	50.106383	359
34501051	3	100100	100100	88	9	14	947	1500.00	20.11	24.5	0.356	40.677193	337
34501052	1	302600	302600	83	8	19	1333	3587.00	24.34	30.0	0.329	54.774775	439
34501053	1	304000	304000	70	7	13	897	3754.00	27.33	29.7	0.590	47.241379	493
34501054	4	100200	100200	94	9	9	1900	5407.00	17.38	21.0	0.251	61.675000	377

Рис. 10. Преобразованная первичная таблица

Первичная таблица является информационной основой для расчета всех показателей включаемых в ТХР. Расчет ход роста по высоте нормального насаждения.

3. Из первичной таблицы SQL-запросом формируется таблица средних показателей (в данном случае высот) по классам бонитета (рисунок 11).

Dominpor	N por	Age	Pok_1	Pok_1a	Pok_2	Pok_2a	Pok_3	Pok_3a	Pok_4	Pok_4a	Pok_5	Pok_5a
100100	Сосна	10							3.0			
100100	Сосна	30			9.9		9.0		4.7			
100100	Сосна	40	15.0		13.1				8.1		5.6	
100100	Сосна	50	18.3				13.2				7.5	
100100	Сосна	60	20.7		18.2						8.2	
100100	Сосна	70	21.3		20.4		17.9		10.1		8.5	
100100	Сосна	80	24.2		20.8		17.7		15.0		12.0	
100100	Сосна	90	25.1		22.2		19.7		17.0		14.4	
100100	Сосна	100	26.5		24.4		20.8		18.0		13.9	
100100	Сосна	110	27.6		25.7		21.1		18.2		15.1	
100100	Сосна	120	27.3		26.2		22.5		18.7		15.7	
100100	Сосна	130	32.0		23.7		22.2		19.0		15.8	
100100	Сосна	140			24.9		22.3		19.8		16.9	
100100	Сосна	150					22.9		19.8		16.3	
100100	Сосна	160					22.4		20.8		15.0	
100100	Сосна	170					18.0		19.9		16.0	
100100	Сосна	180							19.3			
100100	Сосна	190							18.5		16.6	
100100	Сосна	200							19.2			
100200	Ель	30			10.8		8.3		5.4			
100200	Ель	40	14.2		11.6		10.4				7.7	
100200	Ель	50	17.7		15.4		12.5		10.8		7.5	

Рис. 11. Таблица средних высот по классам бонитета

Показатели в таблице группируются по породам и 10-летним возрастным периодам (Pok\_1 – среднее значение показателя по 1-му классу

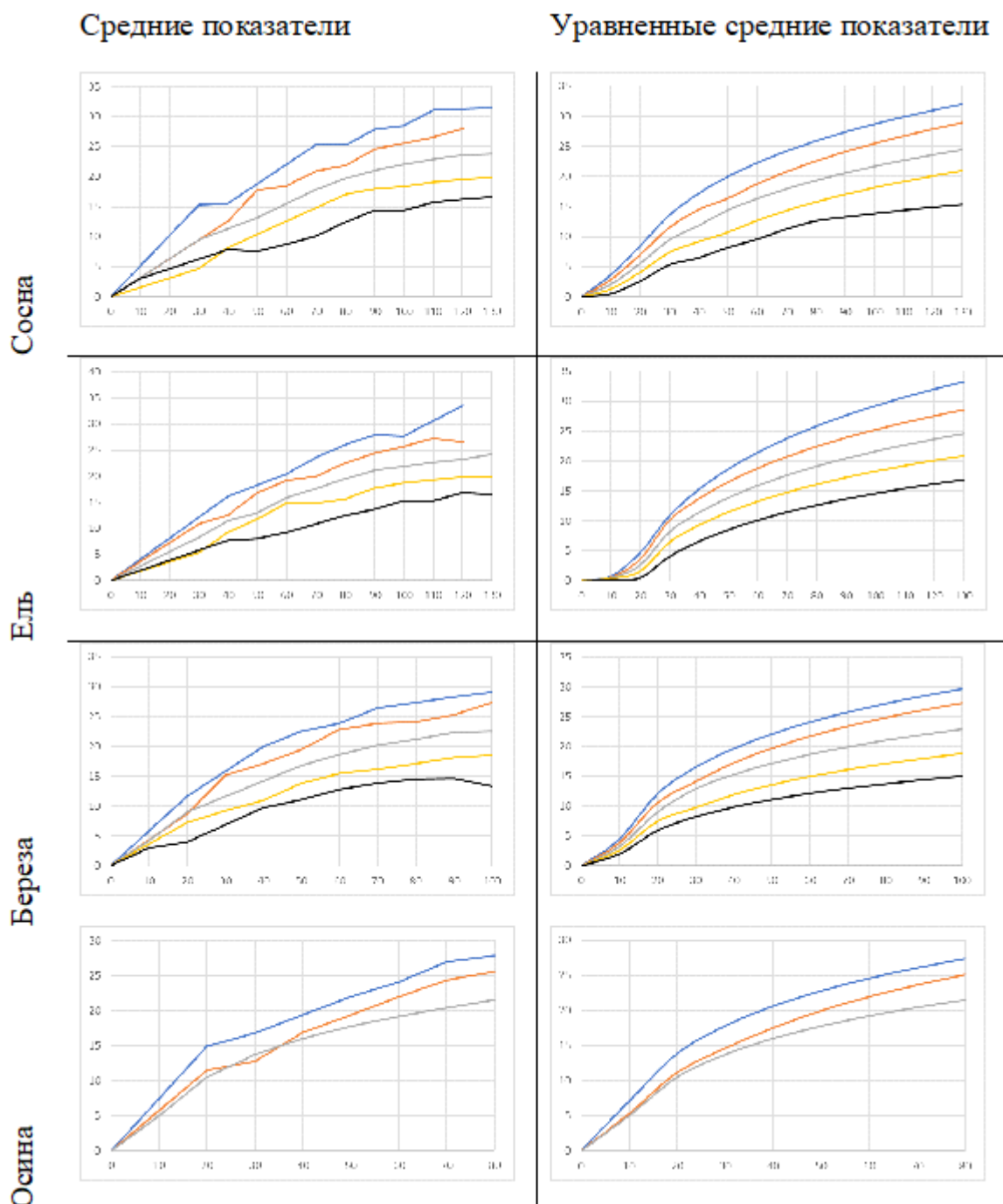
бонитета, ... , Pok\_5 – среднее значение показателя по 5-му классу бонитета, Pok\_1a – среднее значение показателя по 1-му классу бонитета после выравнивания, ... , Pok\_5a – среднее значение показателя по 5-му классу бонитета после выравнивания).

4. Данные таблицы средних показателей обрабатываются путем регрессионного анализа, в результате получаем уравненные данные средних показателей (рисунок 12).

	Dominpor	N_por	Age	Pok_1	Pok_1a	Pok_2	Pok_2a	Pok_3	Pok_3a	Pok_4	Pok_4a	Pok_5	Pok_5a
	100100	Сосна	0										
	100100	Сосна	10		7.197		5.416		4.767	3.0	2.620		1.603
	100100	Сосна	20		10.655		8.429		7.377		4.552		3.054
	100100	Сосна	30		13.403	9.9	10.918	9.0	9.525	4.7	6.288		4.453
	100100	Сосна	40	15.0	15.774	13.1	13.117		11.417	8.1	7.909	5.6	5.820
	100100	Сосна	50	18.3	17.897		15.124	13.2	13.141		9.448	7.5	7.162
	100100	Сосна	60	20.7	19.843	18.2	16.989		14.740		10.926	8.2	8.485
	100100	Сосна	70	21.3	21.652	20.4	18.745	17.9	16.243	10.1	12.355	8.5	9.793
	100100	Сосна	80	24.2	23.351	20.8	20.412	17.7	17.669	15.0	13.742	12.0	11.088
	100100	Сосна	90	25.1	24.961	22.2	22.005	19.7	19.030	17.0	15.094	14.4	12.371
	100100	Сосна	100	26.5	26.495	24.4	23.535	20.8	20.336	18.0	16.416	13.9	13.645
	100100	Сосна	110	27.6	27.964	25.7	25.011	21.1	21.595	18.2	17.712	15.1	14.910
	100100	Сосна	120	27.3	29.375	26.2	26.439	22.5	22.811	18.7	18.984	15.7	16.166
	100100	Сосна	130	32.0	30.737	23.7	27.824	22.2	23.991	19.0	20.235	15.8	17.416
	100200	Ель	0										
	100200	Ель	10		7.008		5.787		4.195		2.524		2.529
	100200	Ель	20		10.331		8.728		6.720		4.502		4.223
	100200	Ель	30		12.964	10.8	11.101	8.3	8.854	5.4	6.316		5.701
	100200	Ель	40	14.2	15.231	11.6	13.165	10.4	10.767		8.031	7.7	7.054
	100200	Ель	50	17.2	17.258	15.4	15.028	12.5	12.531	10.8	9.675	7.5	8.320
	100200	Ель	60	20.2	19.113	18.5	16.744	14.8	14.185		11.267	9.2	9.522
	100200	Ель	70	21.5	20.836	19.2	18.347	17.0	15.753	14.6	12.814	10.9	10.672
	100200	Ель	80	23.2	22.454	21.6	19.859	18.1	17.250	14.6	14.326	12.0	11.781
	100200	Ель	90	24.5	23.985	22.5	21.295	19.9	18.689	16.6	15.806	12.3	12.853
	100200	Ель	100	25.9	25.443	22.1	22.668	20.7	20.077	17.6	17.260	14.0	13.896
	100200	Ель	110	26.3	26.838	23.5	23.986	21.2	21.421	18.5	18.689	15.3	14.911
	100200	Ель	120	26.2	28.178	23.7	25.257	21.5	22.727	18.8	20.098	16.7	15.903
	100200	Ель	130		29.470	24.7	26.484	21.5	23.998	18.8	21.487	16.2	16.873
	302600	Береза	0										

Рис. 12. Таблица со средними и уравненными показателями

5. Данная таблица конвертируется в формат таблиц MicrosoftExcel, в среде которого формируются графики по методу Баура (Рисунок 13).



**Рис. 13.** Графики Баура на примере средних высот. Синий график – 1-й кл.бонитета, оранжевый – 2-й, серый – 3-й, желтый – 4-й, черный – 5-й. В Ленинградской области осина, как доминирующая порода встречается только в 3-м и более высоких бонитетах. По оси Y – высота, по оси X – возраст.

6. На основе таблицы (рисунок 12) и графиков (рисунок 13), таким образом и по такому алгоритму делается расчет по всем показателям ТХР. Результаты работ - Фрагмент таблицы хода роста полных сосновых насаждений Ленинградской области по сосновым насаждениям и 3 классу бонитета.



## Ход роста полных сосновых насаждений Ленинградской области

Возраст, лет	Высота, м	Диаметр, см	Количество деревьев, шт	Сумма площадей сечений, м <sup>2</sup> /га	Запас, м <sup>3</sup> /га	Прирост по запасу, м <sup>3</sup> /га в год	
						текущий	средний
3 класс бонитета							
10	2,1	1,8	42 168	10,7	17	1,7	1,7
20	5,5	7,0	4 861	18,8	64	4,8	3,2
30	9,4	11,5	2 297	23,8	124	6,0	4,1
40	11,8	15,4	1 408	26,4	165	4,1	4,1
50	14,4	18,9	1 030	28,9	211	4,6	4,2
60	16,3	21,7	828	30,6	248	3,7	4,1
70	17,9	24,1	705	32,1	280	3,2	4,0
80	19,3	26,1	621	33,3	309	2,9	3,9
90	20,6	28,0	560	34,4	335	2,6	3,7
100	21,7	29,6	514	35,4	359	2,4	3,6
110	22,7	31,1	478	36,2	382	2,2	3,5
120	23,6	32,4	449	37,0	402	2,1	3,4
130	24,4	33,6	424	37,7	422	1,9	3,2

Для анализа соответствия смоделированных показателей с исходными данными, используется коэффициент детерминации, который отражает меру качества уравнения регрессии, анализируемой регрессионной модели и имеет вид:

$$R^2 = 1 - \frac{\sum(Y_i - Y_x)^2}{\sum(Y_i - \bar{Y})^2} \quad (8)$$

Результаты работ - Фрагмент таблицы коэффициентов детерминации по породе сосна в разрезе классов бонитета и в среднем по всем показателям

Таблица 3

**Коэффициенты детерминации по породе сосна  
в разрезе классов бонитета и в среднем по всем показателям**

Порода	Коэффициенты детерминации по классам бонитета				
	1-й	2-й	3-й	4-й	5-й
По высоте					
Сосна	0,965	0,982	0,987	0,980	0,848
По диаметру					
Сосна	0,914	0,966	0,942	0,981	0,831
По количеству деревьев					
Сосна	0,821	0,928	0,834	0,761	0,731
По сумме площадей сечений					
Сосна	0,940	0,974	0,965	0,981	0,840
По запасу					
Сосна	0,936	0,974	0,947	0,987	0,872
По текущему приросту					
Сосна	0,938	0,974	0,956	0,984	0,856

По среднему приросту					
Сосна	0,936	0,974	0,947	0,987	0,872

## Сравнение тестовых таблиц хода роста лесных насаждений с общепризнанными

Для сравнения полученных ТХР с общепризнанными, последние взяты из сборника А.З. Швиденко и др. [5], ТХР полных сосновых насаждений Северо-Запада европейской части (северная и средняя таежные зоны) Российской Федерации (стр. 130-132), ТХР полных еловых насаждений Северо-Запада европейской части (северная и средняя таежные зоны) Российской Федерации (стр. 187-189), ТХР полных березовых древостоев Северо-Запада европейской части (экорегiónы южной тайги и зоны смешанных лесов) Российской Федерации (стр. 315-317)), ТХР полных осиновых древостоев Северо-Запада европейской части (экорегiónы южной тайги и зоны смешанных лесов) Российской Федерации (стр. 341-343). Результаты сравнения абсолютных значений показателей приведены ниже.

Таблица 4

### Сравнение тестовых таблиц хода роста лесных насаждений с общепризнанными

	Возраст, лет	Высота, м	Диаметр, см	Количество деревьев, шт	Сумма площадей сечений, м <sup>2</sup> /га	Запас, м <sup>3</sup> /га	Прирост по запасу, м <sup>3</sup> /га в год	
							текущий	средний
Итоговые результаты	автор	<b>207,8</b>	<b>279,4</b>	<b>14 176</b>	<b>374,6</b>	<b>3 303</b>	<b>40,5</b>	<b>45,1</b>
	сборник	198,7	225,1	20 446	379,3	3 320	41,4	43,6
	разница	9,1	54,3	-6 270	-4,7	-17	-0,9	1,5
	%	4,6	24,1	-31	-1,2	-1	-2,1	3,5

## Заключение

В целом в связи с изменением климатических условий и по мере накопления информации ГИЛ, действующие ТХР необходимо пересматривать, причем обязательно по единым, утвержденным в установленном порядке, алгоритмам и с использованием БД ГИЛ.

## Библиографический список

1. Программный измерительный комплекс государственной инвентаризации лесов Российской Федерации (ПИК ГИЛ). Руководство пользователя. ФГБУ «Рослесинфорг», М:2013

2. Поваров Е.Д. Анализ алгоритмов обработки данных полевых измерений на постоянных пробных площадях и порядок расчета количественных показателей лесных насаждений в ПИК ГИЛ. ФГБУ «Рослесинфорг», стр.67, Москва, 2018
3. Поваров Е.Д. Алгоритмы обработки данных полевых измерений на постоянных пробных площадях государственной инвентаризации лесов/ Е.Д. Поваров, А.И. Григорьев, В.Н. Манович. ФГБУ «Рослесинфорг», стр.159, Москва, 2019
4. Приказ Рослесхоза от 10 ноября 2011 г. № 472 «Об утверждении методических рекомендаций по проведению государственной инвентаризации лесов», в ред. приказа Рослесхоза от 15.03.2018 № 173
5. Швиденко А.З., Щепашенко Д.Г., Нильссон С., Булуй Ю.И. Таблицы и модели хода роста и продуктивности насаждений основных лесобразующих пород Северной Евразии (нормативно-справочные материалы). Издание 2-е, доп. — М.: Федеральное агентство лесного хозяйства, 2008. — С. 886

**Р.А. Рухлов**, магистрант 1 курса  
Кафедра информационных систем и технологий  
СПб ГЛТУ им. С.М.Кирова  
rom2000@mail.ru

**С.П. Хабаров**, кандидат технических наук, доцент  
Кафедра информационных систем и технологий  
СПб ГЛТУ им С.М.Кирова  
serg.habarov@mail.ru

## **ИССЛЕДОВАНИЕ ТЕХНОЛОГИИ VPN С ИСПОЛЬЗОВАНИЕМ СЕТИ ИЗ ВИРТУАЛЬНЫХ МАШИН НА БАЗЕ ОС WINDOWS**

***Аннотация.** В настоящее время актуальны проблемы, связанные с тем, что при передаче информации через интернет конфиденциальные данные могут попасть в руки злоумышленником. Чтобы защитить данные, передаваемые по сети можно использовать VPN-соединение, которое сможет гарантировать, что трафик, передаваемый через Интернет, будет защищен также, как и при передаче данных по локальной сети.*

***Ключевые слова:** Виртуальная частная сеть; VPN соединения; VPN протоколы; безопасность.*

## **Введение.**

По мере развития информационных технологий, тема безопасности информационного взаимодействия пользователей и защиты информации, как в обычной жизни, так и в профессиональной деятельности становится все более актуальной. Виртуальная частная сеть или VPN (Virtual Private Network) – это совокупность технологий, которые позволяют обеспечивать одно или несколько сетевых соединений поверх другой сети (например, Интернет). Эта технология позволяет защитить процесс обмена информацией по сети от вмешательства посторонних лиц. VPN соединение представляет из себя, так называемый, “туннель” между двумя узлами: компьютером-клиентом и компьютером-сервером, каждый из которых шифрует данные до их попадания в “туннель”.

Построение VPN-сетей основывается на базе VPN-протоколов, которые используются для реализации туннелирования между устройствами. Одними из самых известных протоколов являются:

- 1) PPTP (Point-to-Point Tunneling Protocol). Этот протокол от компании Microsoft не требует большой вычислительной мощности, поддерживается всеми операционными системами и имеет максимальную производительность, однако имеет устаревшие методы шифрования в отличии от современных протоколов.
- 2) OpenVPN. Этот протокол считается одним из самых безопасных и известных протоколов, который можно установить на любую операционную систему. Его шифрование реализуется за счет использования библиотек OpenSSL.
- 3) L2TP (Layer 2 Tunneling Protocol)/IPSec. Считает одним из самых эффективных протоколов для построения виртуальных сетей. Имеет поддержку всех современных ОС, но требует больше вычислительных ресурсов. L2TP чаще всего работает совместно с группой протоколов IPSec, которые поддерживают безопасный обмен информационными пакетами на IP-уровне. Данный протокол имеет производительность ниже, чем у OpenVPN, так как во время его работы происходит двойное инкапсулирование данных.
- 4) IKEv2 (Internet Key Exchange version 2)/IPSec. По безопасности протокол находит на том же уровне, что и предыдущий, однако производительнее и быстрее OpenVPN. Это один из не многих протоколов, который кроме популярных систем, также поддерживает BlackBerry ОС.
- 5) SSTP (Secure Socket Tunneling Protocol). Это надёжный протокол, находится на одном уровне с OpenVPN, однако его можно использовать лишь на компьютерах с операционной системой Windows и Linux.

В ходе работы VPN-протокола происходит аутентификация пользователя, после чего между сервером и клиентом происходит соединение, в ходе которого последнему предоставляются права на выполнение опреде-

ленных действий. Все эти права изначально прописаны на сервере для каждого пользователя или группы.

### Постановка задачи.

Для исследования работы технологии VPN необходимо организовать VPN-соединение. Для реализации данной задачи предполагается наличие определенного сетевого оборудования, с помощью которого можно смоделировать переход данных из одной сети в другую, а также конечных устройств (компьютеры, ноутбуки и т.п.)

Создать модель сети можно с помощью мощных программных продуктов (ПО), специальных симуляторов сети передачи данных, которые позволяют производить настройку маршрутизаторов, коммутаторов, взаимодействовать между пользователями. Примерами таких ПО могут быть CiscoPacketTracer, Dynamips и другие.

Другой подход к организации сети, приближенный к реальным условиям, заключается в использовании стандартных средств ОС Windows и среды виртуализации OracleVMVirtualBox. При помощи данного программного продукта есть возможность на одном компьютере, при наличии необходимых вычислительных ресурсов, создать упрощенную модель сети, которая будет содержать два конечных устройства и два маршрутизатора. Сеть между маршрутизаторами будет моделировать передачу данных через Интернет. Также в данной модели организован VPN-туннель, формирующий виртуальную сеть между узлами PC1 и PC2.

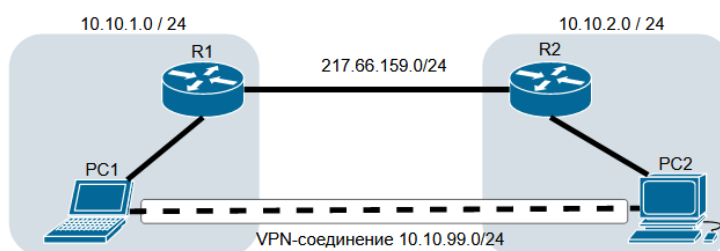


Рис. 1. Модель исследуемой сети.

### Формирование модели и запуск ее в работу.

Как видно на рисунке 1 исследуемая сеть состоит из 4 узлов. Узлы R1 и R2 на базе ОС WindowsXP настроены и используются в качестве маршрутизаторов, для передачи пакетов из одной сети в другую. Узлы PC1 и PC2 являются виртуальными машинами на базе ОС Windows 10, именно они будут выполнять роль VPN-сервера и клиента.

Перед тем как начать исследовать принцип работы VPN необходимо создать виртуальную частную сеть. Для этого надо провести настройку виртуальной машины PC2, выполняющую роль VPN-сервера. Для этого следует выполнить следующие действия:

- 1) Перейти в папку Сетевые подключения (Пуск -> Панель управления -> Центр управления сетями и общим доступом -> Изменение параметров адаптера);
- 2) Затем необходимо создать новое сетевое подключение. Для этого необходимо нажать клавишу Alt, после чего сверху появится строка, где нужно выбрать Файл -> Новое входящее подключение.
- 3) На первом шаге необходимо создать или выбрать пользователя, который в дальнейшем будет подключаться к серверу. Создадим нового пользователя, например, vpnuser. Для этого следует нажать на кнопку Добавить пользователя..., указав его имя и пароль. Если есть необходимость, то можно добавить несколько VPN-пользователей, так как это может быть полезно для дальнейшего контроля за ними. В результате будет создан список пользователей, которым будет разрешен удаленный доступ на конфигурируемый VPN-сервер.
- 4) В следующем окне следует установить флажок в поле «Через Интернет», иначе кнопка «Далее» будет недоступной.
- 5) На последнем шаге нужно выбрать и настроить протоколы. Для этого необходимо:
  - выделить пункт «Протокол Интернета версии 4 (TCP/IPv4)», щелкнув по его названию, а не по галочке;
  - нажать «Свойства»;
  - убедиться, что флажок пункта «Разрешить звонящим доступ к локальной сети» установлен;
  - выбрать опцию «Указать IP-адреса явным образом»;
  - указать диапазон IP-адресов.

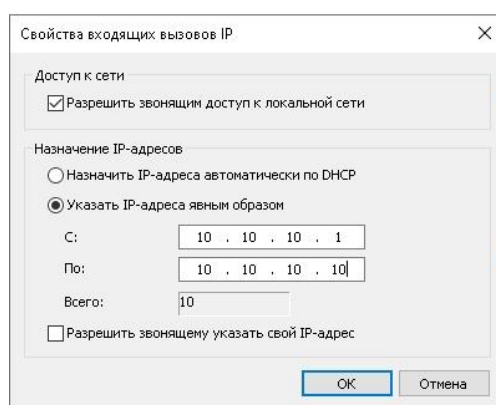


Рис. 2. Настройка свойств входящих вызовов IP

На рисунке 2 видно, что для создаваемого VPN-подключения выделено 10 IP-адресов, один из них будет выделен для сервера. Следовательно, к нему по VPN смогут подключиться до 9 клиентов.

После настройки протокола необходимо нажать кнопку «Разрешить доступ». Если все было выполнено верно, то должно появиться сообщение что «выбранные пользователи теперь могут подключаться к этому компь-

ютеру”. Также в папке сетевые подключения появится новый ярлык “Входящие подключения”.

Также в Windows 10 следует убедиться, что через брандмауэр разрешены VPN-соединения. Для этого необходимо через Пуск выполнить поиск “Разрешение взаимодействия с приложениями через брандмауэр Windows”. В списке разрешенных программ и компонентов нужно найти “Маршрутизация и удаленный доступ” и убедиться, что они разрешены для частного и общего доступа. При необходимости нужно поставить галочки и нажать “Изменить параметры”.

После успешной настройки сервера можно выполнить подключение к нему с виртуальной машины PC1, которая будет выступать в качестве VPN-клиента.

- 1) В “Центре управления сетями и общим доступом” нужно выбрать пункт “Создание и настройка нового подключения или сети”;
- 2) В следующем окне выбирается третий вариант подключения, а именно “Подключение к рабочему месту”;
- 3) На следующем этапе указывается первый пункт “Использовать мое подключение к Интернету (VPN)”;
- 4) На последнем шаге следует указать IP-адрес компьютера, на котором находится VPN-сервер. После нажатия кнопки “Создать” в папке “Сетевые подключения” компьютера PC1 появится VPN-подключение. Для установки соединения необходимо нажать правой кнопкой мыши на данный ярлык и выполнить подключение, после чего будет запрошен пользователь и пароль.

Также в свойствах подключения можно выбрать протокол, по которому будет производиться туннелирование между устройствами (рис. 3).

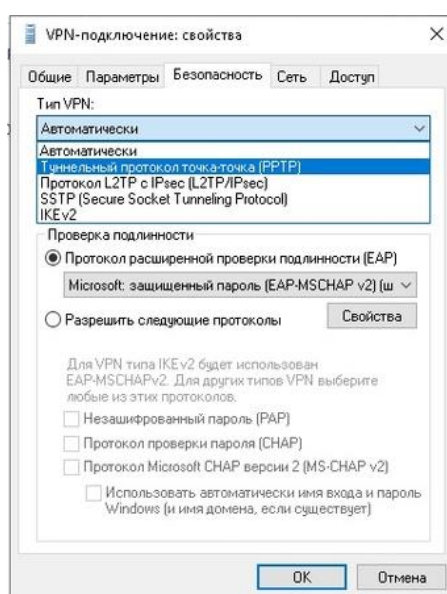


Рис. 3. Настройка свойств VPN-клиента

Если теперь обратить внимание на состояние VPN-подключения PC1, то можно убедиться в том, что пользователь `vpnuser` установил связь с VPN-сервером с использованием протокола PPTP.

Завершая установку соединения следует убедиться в его работоспособности. Выполнить проверку можно с помощью стандартного эхо-запроса, который будет выполнен успешно (рис. 4), если все сделано верно. Это подтверждает работоспособность канала на сетевом и канальном уровнях.

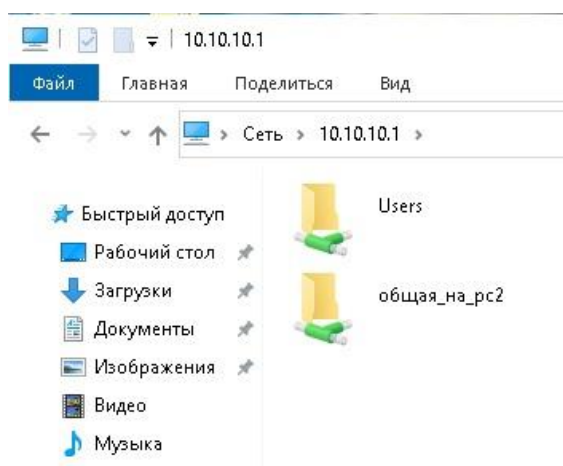
```
C:\Users\user>ping 10.10.10.1

Обмен пакетами с 10.10.10.1 по с 32 байтами данных:
Ответ от 10.10.10.1: число байт=32 время=3мс TTL=128
Ответ от 10.10.10.1: число байт=32 время=3мс TTL=128
Ответ от 10.10.10.1: число байт=32 время=1мс TTL=128
Ответ от 10.10.10.1: число байт=32 время=2мс TTL=128

Статистика Ping для 10.10.10.1:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
```

**Рис. 4.** Эхо-запрос с PC1 на VPN-сервер

Более того любое сетевое подключение создается для того чтобы получить некую информацию с удаленного узла, будь то другой компьютер или сервер. Соответственно если эта информация является конфиденциальной, то и предавать эту информацию лучше по зашифрованному VPN-каналу. В созданной модели после подключения клиент имеет возможность получить доступ к общим файлам и папкам (рис. 5), что подтверждает работоспособность соединения на прикладном уровне. Причем на сервере можно задать определенные права доступа как для конкретного пользователя, так и для группы.



**Рис. 6.** Доступ с PC1 к общедоступным ресурсам PC2

Следующим шагом в исследовании технологии VPN является маршрут, по которому передаются данные по сети. Чтобы разобраться в данном вопросе, можно воспользоваться командой `tracert`. Проведя трассировку маршрутов (рис. 6) от PC1 к реальному физическому интерфейсу PC2 и к



виртуальному интерфейсу VPN, видно, что маршрут к физическому интерфейсу прошел через маршрутизаторы R1 и R2, сделав 3 перехода между сетями. При трассировке маршрута к виртуальному интерфейсу PC2 можно наблюдать, что пакет не совершали ни одного перехода, а логически двигались по туннелю VPN, хотя на физическом уровне пакеты в виде сигналом могут перемещаться только по физическим каналам.

```

Трассировка маршрута к 10.10.2.1 с максимальным числом прыжков 30

 1 <1 ms <1 ms <1 ms R1 [10.10.1.2]
 2 2 ms 2 ms 2 ms 217.66.159.2
 3 3 ms 3 ms 3 ms 10.10.2.1

Трассировка завершена.

C:\Users\user>tracert 10.10.10.1

Трассировка маршрута к 10.10.10.1 с максимальным числом прыжков 30

 1 2 ms 2 ms 3 ms 10.10.10.1

Трассировка завершена.

```

**Рис. 6.** Маршруты передачи пакетов от узла PC1 к узлу PC2

Чтобы убедиться в том, что туннель проходит мимо маршрутизаторов, можно выполнить трассировку от одного из них (например, от R2) к виртуальной сети (рис. 7).

```

Командная строка

C:\Documents and Settings\admin>ping 10.10.10.1
Обмен пакетами с 10.10.10.1 по 32 байт:
Ответ от 217.66.159.2: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 217.66.159.2: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 217.66.159.2: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 217.66.159.2: Превышен срок жизни (TTL) при передаче пакета.

Статистика Ping для 10.10.10.1:
  Пакетов: отправлено = 4, получено = 4, потеряно = 0 (0% потерь),
 Приблизительное время приема-передачи в мс:
  Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек

C:\Documents and Settings\admin>ping 10.10.10.2
Обмен пакетами с 10.10.10.2 по 32 байт:
Ответ от 217.66.159.2: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 217.66.159.2: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 217.66.159.2: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 217.66.159.2: Превышен срок жизни (TTL) при передаче пакета.

Статистика Ping для 10.10.10.2:
  Пакетов: отправлено = 4, получено = 4, потеряно = 0 (0% потерь),
 Приблизительное время приема-передачи в мс:
  Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек

```

**Рис. 7.** Трассировка маршрутов от узла R2 к узлам VPN-сети

Чтобы отметить еще одну особенность при передаче данных по сети с использованием VPN, следует посмотреть маршруты от PC1 к физическим интерфейсам сети 10.10.2.0, а именно PC2 и R2.

```
C:\Users\user>tracert 10.10.2.1
Трассировка маршрута к 10.10.2.1 с максимальным числом прыжков 30
 1  <1 мс    <1 мс    <1 мс    R1 [10.10.1.2]
 2  1 ms     <1 мс    <1 мс    217.66.159.2
 3  1 ms     1 ms     1 ms     10.10.2.1
Трассировка завершена.
C:\Users\user>tracert 10.10.2.2
Трассировка маршрута к 10.10.2.2 с максимальным числом прыжков 30
 1  2 ms     1 ms     1 ms     10.10.10.1
 2  2 ms     1 ms     1 ms     R2 [10.10.2.2]
Трассировка завершена.
```

**Рис. 8.** Трассировка маршрутов от узла PC1 к узлам сети 10.10.10.0/24

Как видно на рисунке 8 маршрут от PC1 до PC2 прошел стандартно через 2 маршрутизатора, однако маршрут от PC1 к интерфейсу R2, который находится в сети 10.10.10.0 прошел через VPN канал, следовательно, PC1 имеет доступ к интерфейсу маршрутизатора R2 по защищенному каналу.

### **Заключение.**

В ходе исследования технологии VPN были проанализированы самые известные протоколы, на базе которых создается VPN-подключение. При помощи среды OracleVMVirtualBox на базе виртуальных машин под управлением ОС Windows 10 была реализована модель компьютерной сети, которая состоит из двух подсетей, а также канала, моделирующего Интернет подключение. При использовании стандартных средств операционной системы была произведена настройка виртуальной частной сети, с использованием протокола PPTP, что позволило исследовать маршруты передачи данных в такой сети.

### **Библиографический список**

1. Кравченкова И.С., Хабаров С.П. Исследование методов маршрутизации с использованием среды виртуальных машин. // Информационные системы и технологии: теория и практика. Сборник трудов научно-технической конференции. Том 12. – СПб.: СПбГЛТУ, 2020. – С. 72 – 75.
2. Хабаров С.П., Шилкина М.Л. Вычислительные машины, системы и сети: Учебное пособие. — СПб.: СПбГЛТА, 2017. — 240 с.
3. Волохов В.В. Исследование принципов работы VPN, Разработка политики безопасности VPN. Использование анонимайзеров [Электронный ресурс] Научная электронная библиотека “КиберЛенинка”. URL: <https://cyberleninka.ru/article/n/issledovanie-printsipov-raboty-vpn-razrabotka-politiki-bezopasnosti-vpn-ispolzovanie-anonimayzerov/viewer>. (дата обращения: 15.01.2022)
4. Авласевич Д.В., Дмитриев Н.А., Кириллов А.А., Бачинский А.Г. Использование технологии VPN для обеспечения информационной без-

опасности. [Электронный ресурс] Научная электронная библиотека “eLibrary.ru”. URL:

[https://www.elibrary.ru/download/elibrary\\_42857677\\_51751840.pdf](https://www.elibrary.ru/download/elibrary_42857677_51751840.pdf) (дата обращения 16.01.2022)

5. Хабаров С.П. Моделирование Ethernet сетей в среде OMNeT++ INET framework // Научно-технический вестник информационных технологий, механики и оптики. 2018. Т. 18. № 3. с. 465–471.

**И. Б. Соловьев**, магистрант 2 курса ИСиТ  
Кафедра информационных систем и технологий  
СПб ГЛТУ им. С.М.Кирова  
wera\_q@mail.com

**С. П. Хабаров**, кандидат технических наук, доцент  
Кафедра информационных систем и технологий  
СПб ГЛТУ им. С.М.Кирова  
serg.habarov@mail.ru

## **ОСНОВЫ РЕАЛИЗАЦИИ МНОГОПОТОЧНОЙ КЛИЕНТ-СЕРВЕРНОЙ АРХИТЕКТУРЫ ДЛЯ БПЛА**

*Аннотация.* В данной статье рассматривается подход реализации клиент-серверной архитектуры для БПЛА. Данный подход был рассмотрен на примере взаимодействия консольных приложений.

*Ключевые слова:* БПЛА, клиент-серверная архитектура, Python.

### **Введение.**

(БПЛА) - Современные беспилотные летательные аппараты - на своем борту имеют разнообразный набор оборудования от оптических систем высокой чувствительности до геофизических датчиков сканирования земной поверхности.

В условиях ограниченной автономности работы БПЛА затраты времени на приведение его в заданный район обследования, а также на перемещение между отдельными участками обследуемой территории могут существенно снижать эффективность использования БПЛА. А также увеличение системных средств на конструкции БПЛА приводит к дополнительным затратам.

Минимизация временных и экономических затрат на использование большего количества БПЛА может повысить эффективность их использования. Этого можно достичь за счет передачи данных между БПЛА и сервером. В качестве примера для решения этой задачи может быть использована клиент-серверная система, где сложные вычисления будут выполняться на сервере, а БПЛА будет выступать в виде клиента.

Клиент-сервер архитектура, является архитектурой, в которой сетевая нагрузка распределена между поставщиками услуг - серверами, и заказчиками услуг - клиентами. Обычно эти программы расположены на разных вычислительных машинах и взаимодействуют между собой через вычислительную сеть посредством сетевых протоколов.

Основными преимуществами данной архитектуры являются отсутствие дублирования кода, снижение требований к компьютерам, высокая защита данных на сервере и организованный контроль полномочий. А также у данной архитектуры присутствуют недостатки такие как зависимость всей сети от работоспособности сервера, необходимость в системном администраторе и высокая стоимость оборудования.

### **Постановка задачи.**

В данной статье рассматривается тестовый сценарий реализации клиент-серверной архитектуры. В качестве сервера, хранящего данные, выступает виртуальная машина на Ubuntu, с реализацией серверной стороны на высокоуровневом интерпретируемом языке программирования общего назначения - Python, в качестве клиента, передающего данные и играющего роль БПЛА, выступает консольное приложение, а также в качестве просмотрщика (viewer) используется консольное приложение, для мониторинга информации о состоянии всех активных клиентов.

### **Описание эксперимента.**

Реализуемая модель в первую очередь представляется в формате .ру-файлов, которые структурно представляют описываемую модель. В данном сценарии с целью снижения нагрузки на сеть канал передачи данных передает на сервер новые данные только при существенном их изменении по сравнению с ранее переданными. Особенность такой организации работы в том, что отсутствует необходимость в организации запросов со стороны сервера, передача выполняется только в случайные моменты времени, которые определяет сам клиент.

```
from random import *
import time
import socket
addr = ('127.0.0.1', 8081)
t_old = 0
try:
    with socket.create_connection(addr) as sock:
        for i in range(0, 30):
            t_new = randint(15, 20)
            if abs(t_new - t_old) > 2:
                t_old = t_new
                print(i, t_old)
                sock.send(str(t_old).encode('utf-8'))
            time.sleep(1)
```

```
sock.send(b'end')
```

```
except ConnectionRefusedError:  
    print('... Сервер недоступен')
```

При запуске программы происходит автоматическое подключение к серверу, если он находится в активном состоянии. В противном случае обрабатывается исключение и выводится сообщение о недоступности сервера. При успешном TCP соединении с сервером каждую секунду происходит изменение данных и их сравнение с ранее переданными, а в случае значительного различия выполняются преобразование измененных данных в байтовое представление и отправка его на сервер.

Об окончании текущего сеанса работы клиента с сервером будет свидетельствовать посылка ему специального сообщения, в данном примере это b'end'. Приняв это сообщение, сервер должен закрыть используемые для этого сеанса соединения сокетов.

В функцию клиента просмотрщика (viewer) входит установление соединения с сервером или вывод сообщения о его недоступности, а также регистрация клиента на сервере путем посылки ему некоторого стандартного идентификационного сообщения (b'viewer'), прием сообщений от сервера, их декодирование из байтового представления и вывод в консоль приложения.

```
import socket  
addr = ('127.0.0.1', 8081)  
try:  
    with socket.create_connection(addr) as sock:  
        sock.send(b'viewer')  
        while True:  
            data = sock.recv(1024)  
            if data == b"":  
                print("... Соединение разорвано")  
                break  
            print(data.decode('utf-8'))
```

```
except ConnectionRefusedError:  
    print('... Сервер не доступен')
```

Следует отметить, что просмотрщиков при работе в разрабатываемой системе может быть произвольное количество, которое может меняться в течение времени работы системы. Но все они, так или иначе, выполняют клиентские функции, устанавливая дуплексные каналы связи с сервером. При этом на сервер возлагается задача поддержки всех этих соединений и обеспечение взаимодействия между ними.

Эту задачу может решить только многопоточный сервер, обрабатывая каждое из соединений в своем потоке.

Одним из вариантов решения этой задачи может быть использование глобальной переменной clients типа словаря и формирование на ее основе

массива, каждый элемент которого будет содержать подробную информацию о каждом из установленных socket-соединений. Причем не всех соединений, а только тех из них, которые установлены для связи с просмотрщиком.

Глобальный массив установленных с клиентами соединений будет формироваться сервером при поступлении на его вход в одном из потоков бинарного сообщения вида b'viewer'. Данный поток признается работающим с просмотрщиком, и информация о сокете, который используется в данном потоке, будет занесена в глобальную переменную.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import socket
import threading
import time

def echo_svr(conn,addr):
    print('... Connected by', addr)
    with conn:
        while True:
            data = conn.recv(1024)
            print(addr, ' - ', data.decode('utf-8'))
            if data == b'viewer':
                clients[conn] = addr
            else:
                if data == b'end': break
                elif data == b"":
                    del clients[conn]
                break
            else:
                for client in clients:
                    msg = str(addr).encode('utf-8') + b'- '
                    t = time.ctime().encode('utf-8')
                    msg = msg + t
                    client.send(msg + b'-'+ data)
            print('... Disconnect by',addr)

clients = {}
with socket.socket() as sock:
    sock.bind(('', 8081))
    sock.listen(1)
    print('... Wait connection...')
    while True:
        conn, addr = sock.accept()
        th = threading.Thread(target=echo_svr, args=(conn,addr))
```

th.start()

Таким образом, в любой момент времени переменная `clients` содержит информацию обо всех активных соединениях сервера с просмотрщиками. Это позволяет во всех потоках, при поступлении любого нового сообщения:

- принять байтовый поток в переменную `data`;
- на ее основе сформировать сообщение, дополнительно включающее в себя адрес и время поступления;
- отправить вновь сформированное сообщение всем просмотрщикам, используя следующую программную конструкцию:

```
for client in clients:
```

```
    client.send(«бинарноесообщение»)
```

### **Вывод.**

Использование клиент-серверной архитектуры позволяет разделить сетевое приложения на несколько компонентов, каждый из которых реализует набор сервисов. Компоненты такого приложения могут выполняться на разных компьютерах, выполняя серверные и/или клиентские функции. Это позволяет повысить надежность, безопасность и производительность сетевых приложений и сети в целом. Исходя из этого мы можем увидеть, что данная архитектура идеально подходит для решения данной задачи.

### **Библиографический список**

1. Хабаров, С. П. Формирование программных траекторий движения БПЛА с учетом ограничений на их управляемость / С. П. Хабаров, М. Л. Шилкина // Цифровые технологии в лесном секторе: Материалы II Всероссийской научно-технической конференции-вебинара, Санкт-Петербург, 18–19 февраля 2021 года. – Санкт-Петербург: Санкт-Петербургский государственный лесотехнический университет имени С.М. Кирова, 2021. – С. 141-143.
2. Хабаров, С. П. Построение распределенных систем на базе WebSocket : учебное пособие / С. П. Хабаров, М. Л. Шилкина. — Санкт-Петербург : Лань, 2020. — 216 с. — ISBN 978-5-8114-4739-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/142358> (дата обращения: 18.01.2022). — Режим доступа: для авториз. пользователей.
3. Алексеев, А. С. Мониторинг лесопользования с применением открытых материалов дистанционного зондирования земли (ДЗЗ) и беспилотных летательных аппаратов (БПЛА) / А. С. Алексеев, А. А. Филатов // Цифровые технологии в лесном секторе : материалы Всероссийской научно-технической конференции, Санкт-Петербург, 26–27 марта 2020 года / Санкт-Петербургский государственный лесотехнический университет имени С. М. Кирова; Санкт-Петербургский политехнический университет Петра Великого. – Санкт-Петербург: Политех-Пресс, 2020. – С. 5-8.

**С. П. Хабаров**, кандидат технических наук, доцент  
Кафедра информационных систем и технологий  
СПб ГЛТУ им. С.М.Кирова  
serg.habarov@mail.ru

## ИСПОЛЬЗОВАНИЕ Notepad++ ДЛЯ ТЕСТИРОВАНИЯ ПРОСТЕЙШИХ СЕТЕВЫХ JAVA-ПРИЛОЖЕНИЙ

***Аннотация.** Рассмотрен подход к тестированию и исследованию простейших сетевых Java-приложений без применения сложных интегрированных сред разработки, а с использованием только одного текстового редактора Notepad++. Для этого предлагается расширить его функциональность за счет установки плагина NppExec, что позволит все операции по редактированию, компиляции и запуску программного кода выполнять в одной среде, которой будет являться данный редактор.*

***Ключевые слова:** распределенные системы, сетевое программирование на Java, протоколы UDP и TCP.*

### **Введение.**

При построении распределенных информационных систем на Java важным является вопрос тестирования сетевого взаимодействия между отдельными программными модулями. Обычно, их взаимодействие осуществляется на базе UDP или TCP протоколов, когда на один из модулей возлагают серверные функции, а на другие – клиентские, или совмещают эти функции. В этих условиях для полноценного сетевого тестирования нужна хотя бы локальная сеть с реальными временами передачи пакетов, и их задержками в коммутаторах и маршрутизаторах.

Однако на начальных этапах проектирования системы, когда тестируется только функционал, возлагаемый на отдельные ее модули, можно использовать и всего один компьютер, запуская на нем сразу несколько виртуальных машин или несколько процессов в рамках одной операционной системы. В последнем случае могут использоваться либо специализированные фреймворки тестирования Java, такие как JBehave, Selenium или JUnit, либо универсальные интегрированные среды разработки Java (IDE), такие как NetBeans, Eclipse и IDEA. Они допускают возможность отладки, кодирования, использование плагинов и расширений. Их различие в целях, которые стоят перед конкретными программистами. NetBeans легче изучить, чем Eclipse, но Eclipse может обрабатывать более крупные проекты.

Однако, IDE занимают больше ресурсов процессора и памяти, чем обычный текстовый редактор. А огромный объем функций и возможностей может поставить под угрозу эффективное редактирование текста. Поэтому ряд программистов предпочитают использовать простой текстовый редактор, особенно если речь идет о простейших, хотя и сетевых, кодах.

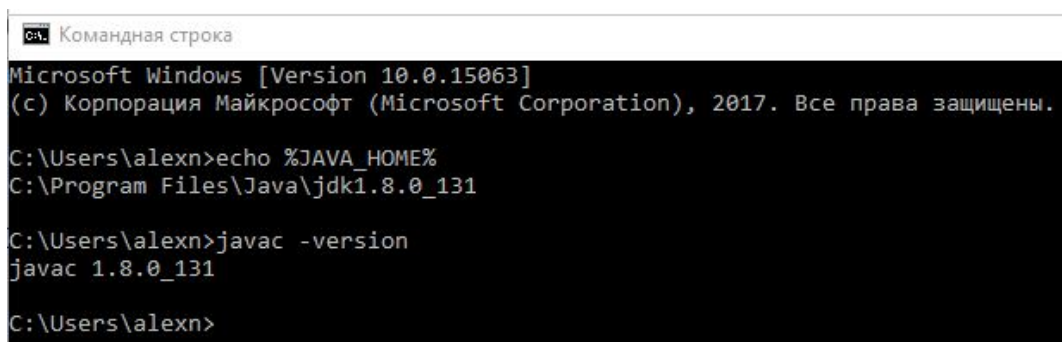


## Компиляция Java-классов в Notepad++.

Редактор Notepad++ – это один из самых популярных текстовых редакторов для программистов. Одной из причин такого успеха является возможность расширения его функционала многочисленными плагинами. В частности, он позволяет компилировать Java-классы, не покидая самого редактора, что очень удобно начинающим Java-программистам.

Для работы с Java с использованием Notepad++, надо убедиться, что на компьютере установлен Java-компилятор, входящий в поставку JDK от Sun. При его отсутствии следует выполнить его установку. При этом надо различать понятия JDK (JavaDevelopmentKit) и JRE (JavaRuntimeEnvironment). Их различие в том, что JDK представляет собой пакет инструментов для разработки программного обеспечения, а JRE – это пакет для запуска Java-кода. Он может использоваться, как отдельный компонент для простого запуска Java-программ, либо быть частью JDK. В любом случае JDK требуется JRE, потому что запуск программ является неотъемлемой частью их разработки.

После того, как JDK от Sun будет установлена на компьютере, надо в переменную среды окружения PATH добавить путь к файлам javac.exe и java.exe. Также необходимо создать переменную CLASSPATH, и в качестве ее значения указать «.;». Подобным же образом создается/изменяется переменная JAVA\_HOME. Её значение — это путь к каталогу, в который установлена Java (как для PATH), но без указания поддиректории bin. После задания всех переменных окружения, установку Java можно считать завершённой и для проверки ее работоспособности в окне командной строки следует ввести команду: `java -version`.



```
cmd. Командная строка
Microsoft Windows [Version 10.0.15063]
(c) Корпорация Майкрософт (Microsoft Corporation), 2017. Все права защищены.

C:\Users\alexn>echo %JAVA_HOME%
C:\Program Files\Java\jdk1.8.0_131

C:\Users\alexn>javac -version
javac 1.8.0_131

C:\Users\alexn>
```

Рис. 1. Результат тестирования успешной установки Java

Проверка успешности установки JDK позволяет перейти к настройке редактор Notepad++ для компиляции и запуска Java-программ. Для этого в него следует установить плагин NppExec. С этой целью надо:

В строке основного меню перейти в режим Плагины (Plugins) -> Plugin Manager -> Show Plugin Manager.

В открывшемся окне на вкладке Available следует выбрать NppExec и нажать Install.

После загрузки и установки плагина соглашаемся с предложением перезапустить Notepad.

Для проверки работоспособности выполненных настроек напомним какой-нибудь Java-класс. Пусть это будет тестовый класс типа HelloWorld. После сохранения его в файле с именем, аналогичным названию класса, следует нажать функциональную клавишу F6 и в окне "Execute ..." (рис. 2) ввести следующие строки:

```
cd "$(CURRENT_DIRECTORY)"
javac $(FILE_NAME)
java $(NAME_PART)
```

Поименовать эту группу команд (например, java\_comp\_run) и нажать кнопки Save и OK.

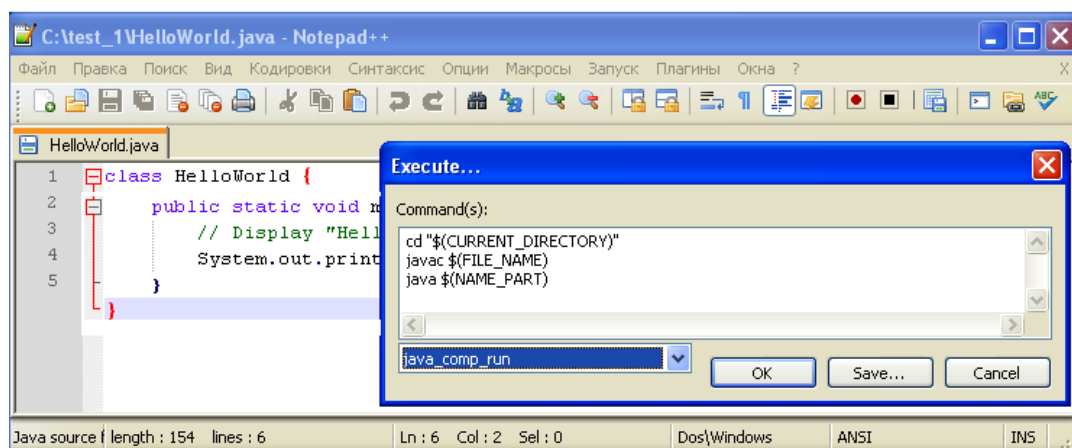


Рис. 2. Формирование команд компиляции и запуска Java-кода

Результат компиляции и выполнения Java-программы будет отображаться в консоли, которая выводится в нижней части окна Notepad++ под фреймом окна редактора файла (рис. 3).

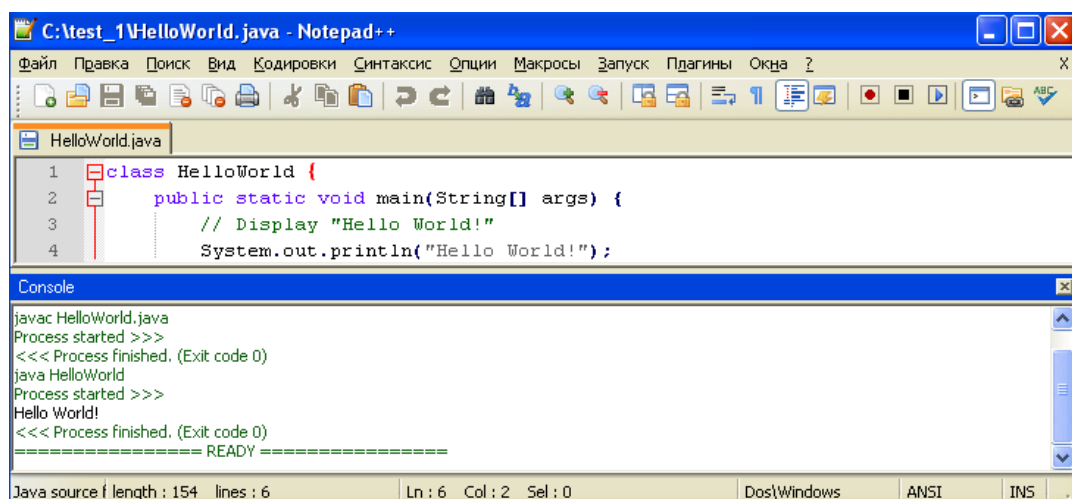


Рис. 3. Формирование выполнения Java-кода

Следует отметить, что в тех случаях, когда до момента компиляции текущего Java-файла надо выполнять его автоматическое сохранение, достаточно в первую строку командного скрипта добавить еще команду:

```
NPP_SAVE
```

Если проект включает в себя несколько файлов, которые подвергаются совместному редактированию, то в этом случае может оказаться полезной команда `NPP_SAVEALL`, которая будет обеспечивать сохранение всех модифицированных в Notepad++ файлов.

### Возможные проблемы с русификацией

Если приведенный ниже код был создан, например, в Блокноте и сохранен в текстовом файле `test.java` с кодировкой ANSI

```
class test {  
    public static void main(String[] args) {  
        // Display "Hello World!"  
        System.out.println("Привет, мир !!!");  
    }  
}
```

то после его компиляции в окне командной строки и запуска на выполнение, будет получен результат, аналогичный тому, что приведен на рис. 4.

```
C:\>javac test.java  
  
C:\>java test  
±Ештх, ьшЕ !!!  
  
C:\>chcp  
pxьe 7p ьюфютр ёёЕрэшЮр: 1251
```

Рис. 4. Результат компиляции и запуска кода в окне командной строки.

При этом проверка кодировки выдает результат, что окно работает с кодовой страницей 1251. В ряде случаев исключить ошибки вывода можно путем изменения типа шрифта, используемого этим окном. Если правой кнопкой мыши щелкнуть по заголовку окна, то появится меню, в котором надо выбрать опцию "Свойства" и на закладке "Шрифт" установить шрифт `Lucida Console` (рис. 5).

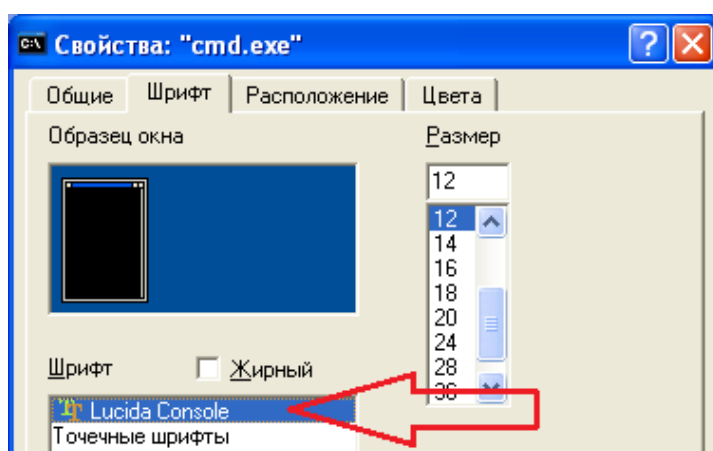


Рис. 5. Изменение типа шрифта в окне командной строки

После изменения типа шрифта и нажатия кнопки ОК, окно командной строки автоматически обновится и станет иметь вид, как на рис. 6.

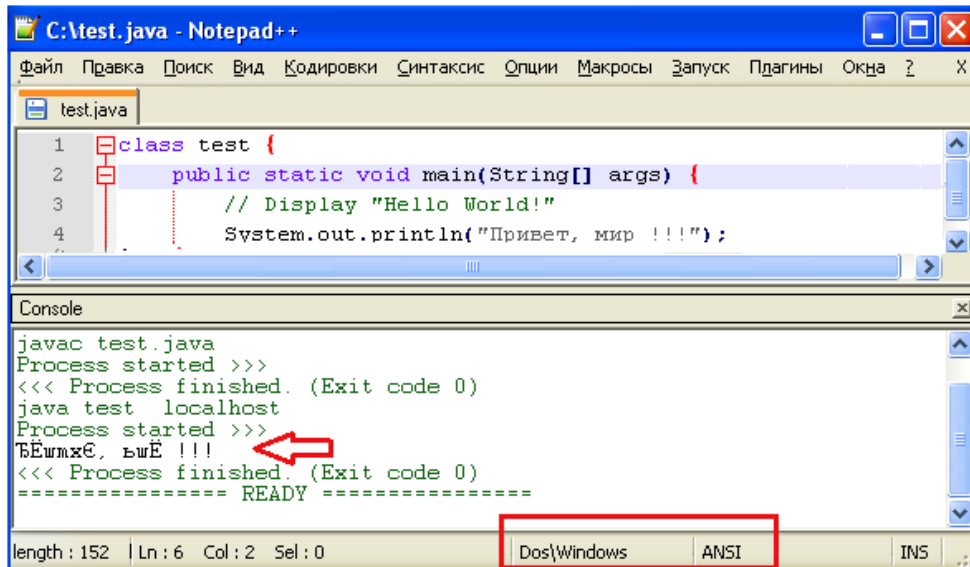
```

C:\>javac test.java
C:\>java test
Привет, мир !!!
C:\>chcp
Текущая кодовая страница: 1251

```

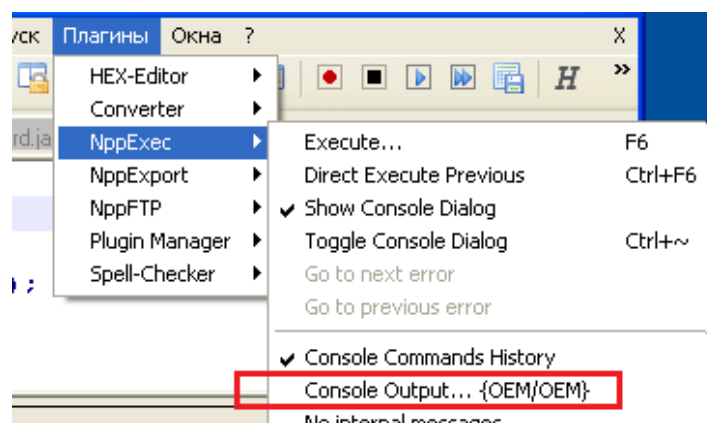
**Рис. 6.** Окно командной строки после изменение типа шрифта.

Ошибки в отображении кириллицы могут возникнуть и в Notepad++. Особенно, сразу после установки NppExec. При этом вариант выполнения кода, содержащего кириллицу, может быть такой, как показан на рисунке 7.



**Рис. 7.** Результат компиляции и запуска Java-кода в окне Notepad++

В этом случае следует, прежде всего, убедиться в том, что исходный файл сохранен в кодировке Windows ANSI. Если это так, то надо проверить кодировки консоли ввода/вывода самого Notepad++ (рис. 8).



**Рис. 8.** Проверка кодировки консоли ввода/вывода Notepad++

Для проверки или изменения кодировки, используемой в консоли Notepad++ надо в меню Плагины выбрать опцию NppExec, и в появившемся меню проверить, что рядом с опцией Console Output ... стоит обозначение

ние типа {ANSI/ANSI}. Если это не так, то следует кликнуть мышкой по этой опции и выполнить соответствующую настройку.

## Тестирования простейших сетевых Java-кодов

В рамках одного распределенного приложения передача данных от одного его компонента другому включает в себя: во-первых, получение данных тем узлом, где находится приложение-адресат, и, во-вторых, получение данных именно тем выполняющимся на узле-адресате приложением, которому они предназначены, так как на узле одновременно могут выполняться несколько приложений.

Простейший способ решения таких задач может основываться на API, входящем в состав пакета java.net. Этот пакет позволяет реализовывать сетевое взаимодействие с применением одного из двух транспортных протоколов: UDP или TCP. Рассмотрим процесс тестирования сетевого взаимодействия всего двух простейших сетевых компонентов, используя для этого всего один компьютер и редактор Notepad++.

Файл UDPServer.java содержит всего один класс **UDPServer**, который создает сокет и обслуживает запросы клиента.

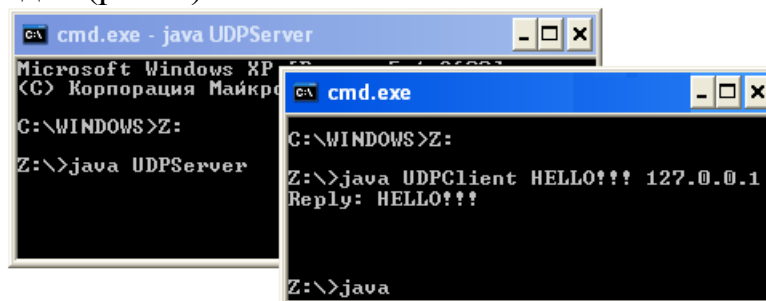
```
import java.net.*;
import java.io.*;
public class UDPServer{
    public static void main(String args[]) {
        DatagramSocket aSocket = null;
        try{
            aSocket = new DatagramSocket(6789); // create socket
            byte[] b = new byte[1000];
            while(true){
                DatagramPacket req = new DatagramPacket(b, b.length);
                aSocket.receive(req);
                DatagramPacket reply = new DatagramPacket(
                    req.getData(), req.getLength(),
                    req.getAddress(), req.getPort());
                aSocket.send(reply);
            }
        } catch (SocketException e){
            System.out.println("Socket: " + e.getMessage());
        } catch (IOException e) {
            System.out.println("IO: " + e.getMessage());
        } finally {if(aSocket != null)
            aSocket.close();
        }
    }
}
```

Файл UDPClient.java содержит класс **UDPClient**, который создает сокет, соединяется с сервером (порт 6789), пересылает ему сообщение и ждет от него ответа.

```
import java.net.*;
import java.io.*;

public class UDPClient{
    public static void main(String args[]){
        // args give message contents and destination hostname
        try {
            DatagramSocket aSocket = new DatagramSocket();
            byte [] message = args[0].getBytes();
            InetAddress aHost = InetAddress.getByName(args[1]);
            int serverPort = 6789;
            DatagramPacket req = new DatagramPacket(message,
                args[0].length(), aHost, serverPort);
            aSocket.send(req);           //send message
            byte[] buf = new byte[1000];
            DatagramPacket reply = new DatagramPacket(buf, buf.length);
            aSocket.receive(reply);     //wait for reply
            System.out.println("Reply:" + new String(reply.getData()));
            aSocket.close();
        } catch (SocketException e){
            System.out.println("Socket: " + e.getMessage());
        } catch (IOException e){
            System.out.println("IO: " + e.getMessage());
        }
    }
}
```

В том, что представленные коды работоспособны легко убедиться, если сначала выполнить компиляцию каждого из них, а затем, открыв две версии командной строки, выполнить в каждой из них запуск откомпилированных Java-кодов (рис. 9).



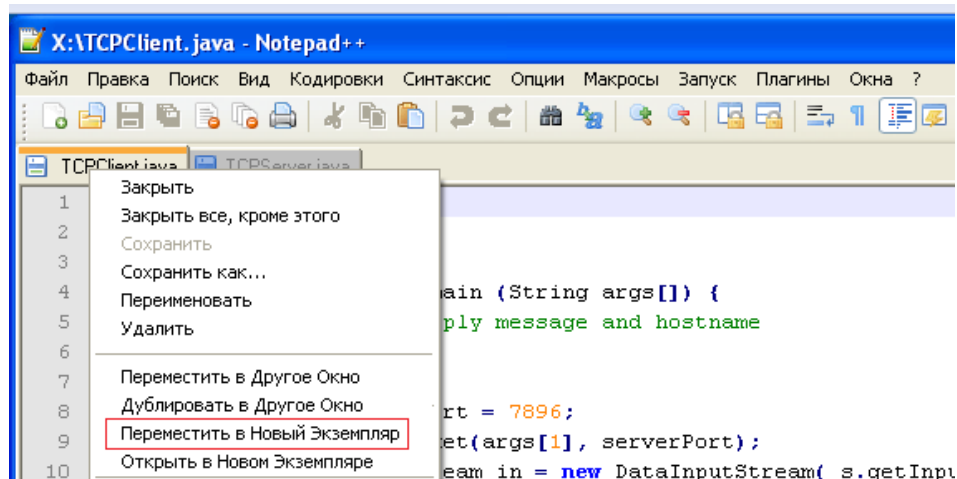
**Рис. 9.** Проверка взаимодействия двух сетевых java-кодов

Но если потребуются какие-либо изменения в исходных кодах, то все процесс редактирования, компиляции и выполнения надо повторить вруч-

ную. Что является довольно трудоемким процессом. Частично упростить его позволит использование уже настроенного редактора Notepad++.

В том случае, когда для редактирования, компиляции и выполнения простейших клиент-серверных программ используется среда Notepad++, представляется целесообразной следующая технология работы.

Запускается среда Notepad++, в которую загружаются сразу два файла. Например, TCPClient.java и TCPServer.java. Затем над закладкой одного из них правой кнопкой мыши вызывается всплывающее меню и выбирается опция "Переместить в Новый экземпляр" (рис. 10).



**Рис. 10.** Вызов второго экземпляра Notepad++ с перемещением в него файла TCPClient.java.

В результате выполнения этого действия будет выполнен вызов второго экземпляра Notepad++, куда будет перемещен выбранный файл. На экране будут присутствовать два окна Notepad++, в каждом из которых может редактироваться и сохраняться свой файл.

Для совместной работы клиент-серверных приложений достаточно последовательно в каждом из окон Notepad++ нажать F6 и ввести необходимую последовательность команд для компиляции и выполнения команд. Для компиляции и выполнения TCPServer.java надо ввести (или выбрать из ранее сохраненных наборов) набор команд, приведенный на рис. 11 в окне "Execute...".

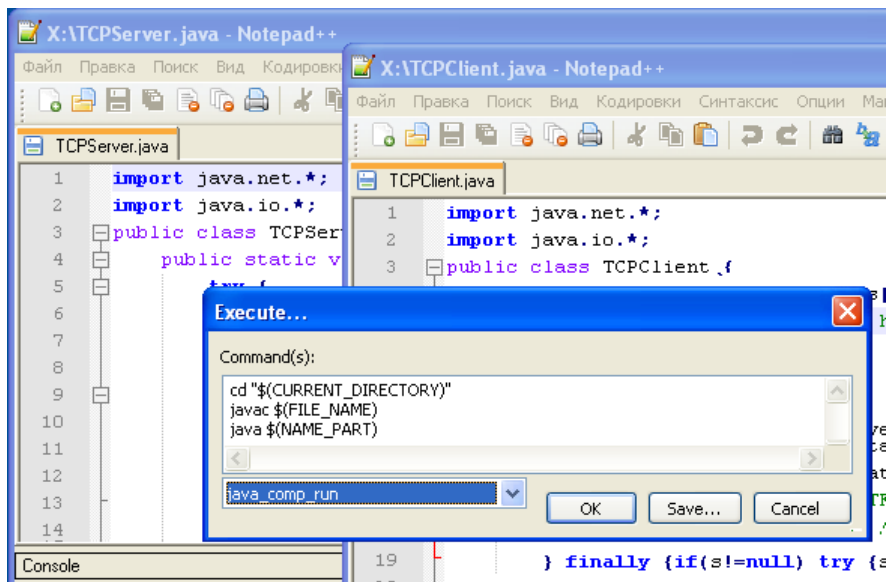


Рис. 11. Запуск java-программы на компиляцию и выполнение

Для запуска в работу программы TCPClient.java следует несколько изменить последнюю строчку, указав в ней IP-адрес сервера, к которому клиент будет подключаться. При работе этих двух программ на одном ПК эта строка может иметь вид:

```
java $(NAME_PART) "ПРИВЕТ !!!" 127.0.0.1
```

### Исследование взаимодействия UDP-модулей

Для тестирования процесса сетевого взаимодействия между двумя Java-модулями, которые обмениваются между собой UDP-сообщениями, следует выполнить ряд экспериментов. В частности, остановив сервер и запустив клиента можно убедиться, что в окне клиента ничего не происходит и после небольшой паузы закрыть окно консоли, чтобы прервать работу программы. Для выяснения работоспособности Java-клиента по отправке сообщения серверу, имеет смысл в исходный код добавить еще одну строку:

```
aSocket.send(req); //send message
System.out.println(" ... The message was sent to
the server " + aHost);
```

После запуска в работу можно убедиться, что клиент отправляет сообщения на сервер. Это подтверждает вывод в консоль Notepad++. Но результат работы клиента не изменился. Дело в том, что при использовании UDP соединение с сервером не устанавливается и единственная возможность контролировать обмен данными – это получение ответа от сервера. А если он не работает или не доступен, то контроль связи можно осуществить только по таймауту. Для контроля возможности связи UDP-клиента с сервером установим у него таймаут, например, в 3 секунды. С этой целью в исходный код надо добавить еще одну строку:

```
DatagramPacket reply=new DatagramPacket(buf, buf.length);
aSocket.setSoTimeout(3000); //время ожидания ответа 3сек
```



Вид консоли после запуска отредактированного UDP-клиента изменится, а программа завершится по срабатыванию исключения (рис. 12).

```

Console
-----
CD: Z:\
Current directory: Z:\
javac UDPClient.java
Process started >>>
<<< Process finished. (Exit code 0)
java UDPClient HELLO 127.0.0.1
Process started >>>
... The message was sent to the server /127.0.0.1
IO: Receive timed out
<<< Process finished. (Exit code 0)
===== READY =====
Java source file      length : 1261  lines : 31      Ln : 31  Col : 1

```

Рис. 12. Вид консоли UDP-клиента при не запущенном сервере

Если на компьютере, у которого есть доступ к Интернету, не запускаемая UDP-сервер, запустить UDP-клиента, используя команду

```
java UDPClient HELLO www.habarov.spb.ru
```

то результат выполнения этой команды в консоли Notepad++ будет иметь вид, аналогичный тому, что приведен на рис. 13. Из него видно, что класс InetAddress позволяет выполнить сопоставление логических и физических адресов сети.

```

java UDPClient HELLO www.habarov.spb.ru
Process started >>>
... The message was sent to the server www.habarov.spb.ru/95.140.148.234
IO: Receive timed out
<<< Process finished. (Exit code 0)
===== READY =====

```

Рис. 13. Вид консоли при запуске клиента с логическими адресами

Несмотря на свою простоту, UDPClient является полноценным Интернет-клиентом. В этом можно убедиться, если подключится к одному из серверов мирового времени. Например, сервер ptb.de оборудован высокоточными атомными часами и позволяет по порту 13 получать текущее время в символьном формате. Для реализации поставленной задачи надо временно изменить значение используемого UDP порта

```
intserverPort = 13; //6789;
```

а для запуска UDP-клиента использовать команду

```
java UDPClient " " ptbtime2.ptb.de ,
```

результат тестирования будет иметь вид, представленный на рис. 14.

```

java UDPClient " " ptbtime2.ptb.de
Process started >>>
... The message was sent to the server ptbtime2.ptb.de/192.53.103.104
Reply: 14 JUL 2020 17:37:10 UTC
.....
<<< Process finished. (Exit code 0)
===== READY =====

```

Рис. 14. Вид консоли с текущим значением текущего мирового времени.

Следует отметить, что к рассматриваемому серверу возможно подключение с разных узлов. Это могут быть виртуальные машины внутри

основного компьютера или другие сетевые компьютеры, у которых в брандмауэрах открыт порт 6789. Существенно не усложняя код UDP-сервера, потребуем от него выдачи информации о запуске сервера и регистрации входных сообщений с определением адреса их отправителей.

Пусть простейшим вариантом этой задачи будет вывод информационных сообщений в консоль сервера. Один из вариантов такой реализации, выполненный на компьютере, который имеет IP-адрес 192.168.200.201, представлен на рисунке 15.

```

Console
java UDPServer
Process started >>>
===== Server OK!
... /127.0.0.1 : 1686 ---> HELLO to 127.0.0.1
... /127.0.0.1 : 1687 ---> NEW HELLO to 127.0.0.1
... /192.168.200.201 : 1688 ---> NEW HELLO to 192.168.200.201
... /192.168.200.206 : 62713 ---> HELLO From main PC to VM
... /192.168.200.206 : 64956 ---> Last HELLO From main PC to VM
... /127.0.0.1 : 1691 ---> HELLO from localhost
  
```

**Рис. 15.** Вид консоли сервера при подключении разных клиентов

Первые три подключения выполнялись с локального компьютера, но в третьем вместо 127.0.0.1 использовался IP-адрес компьютера сервера. Следующие два подключения выполнялись из командной строки виртуальной машины, установленной на этом компьютере и имеющей IP-адрес 192.168.200.206. Один из вариантов такого вызова приведен на рисунке 16.

```

c:\For_Java_UM>java UDPClient "HELLO from main PC to UM" 192.168.200.201
... The message was sent to the server /192.168.200.201
Reply: HELLO from main PC to UM
  
```

**Рис. 16.** Запуск UDPClient на компьютере с IP = 192.168.200.206

Для более подробного тестирования можно выполнить модификацию кода UDP-сервера и провести аналогичное тестирование. Примером такой модификации может стать добавление в соответствующие места программы следующих строк. При тестировании особое внимание следует обратить на адреса узлов и портов, с которых получены дейтаграммы.

```

System.out.println("==== Server OK!");
// выделяем сообщение клиента из буфера приема
String mes= (new
    String(req.getData())).substring(0,req.getLength());
// выводим в консоль информационное сообщение
System.out.println(" ... " + req.getAddress() + " : " +
    req.getPort() + " ---> " +mes)
  
```

При передаче длинных сообщений с использованием протокола UDP, сообщения должны разбиваться на несколько блоков, каждый из которых будет передаваться отдельной дейтаграммой. Кроме этого, так как прото-

кол UDP не гарантирует доставки, то в наиболее ответственных случаях одно и то же сообщение посылается несколько раз.

Эту ситуацию можно смоделировать передачей на сервер несколько раз одного и того же сообщения, которое, как и раньше, задается параметром команды запуска. Для этого в программе клиента надо организовать цикл по отправке однотипных дейтаграмм. Примером может быть, например, следующая последовательность действий:

```
for (inti = 1; i<= 3; i++) {  
    System.out.print(i + " ");  
    ... фрагмент программы отправки и получения дейтаграмм ...  
}
```

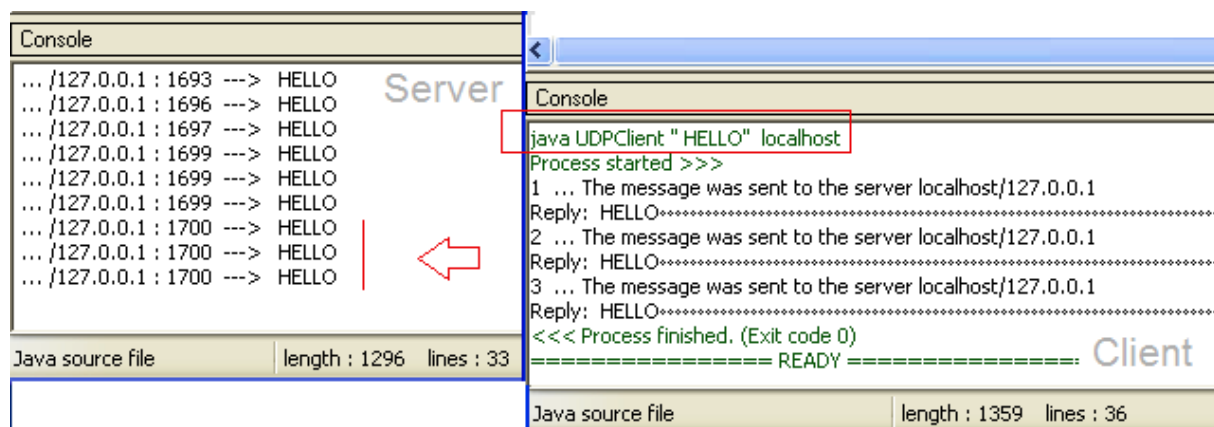


Рис. 17. Пакетная отправка с клиента трех однотипных запросов

Один из возможных вариантов реализации данной задачи приведен на рисунке 17. При этом особое внимание следует обратить на адреса и номера UDP портов зафиксированных на сервере сообщений, как при одиночной, так и при пакетной передаче.

### Библиографический список

1. Notepad++ [Электронный ресурс]. <https://notepad-plus-plus.org/> (дата обращения: 02.01.2022).
2. [Notepad++ User Manual](https://npp-user-manual.org/docs/plugins/) [Электронный ресурс]. <https://npp-user-manual.org/docs/plugins/> (дата обращения: 02.01.2022).
3. Хабаров С.П., Красовский И.А., Киев А.Х.-А. Удаленное управление на базе технологии websocket. // Сборник научных трудов "Информационные системы и технологии: теория и практика" — СПб.:СПбГЛТУ, 2017. № 9. — с.120-126
4. Соловей В.С., Хабаров С.П. Распределённое взаимодействие узлов сети в судовых системах ДАУ. В сборнике: Информационные системы и технологии: теория и практика. Сборник научных трудов. Санкт-Петербургский государственный лесотехнический университет имени С.М. Кирова. Санкт-Петербург, 2021. С. 135-141.

**М.Л. Шилкина**, кандидат технических наук, доцент  
Кафедра информационных систем и технологий  
СПбГЛТУ им. С.М. Кирова  
mchernobay@inbox.ru

## **АДАПТАЦИЯ МУЛЬТИАГЕНТНОГО ПРИЛОЖЕНИЯ JADE ПО ОБСЛУЖИВАНИЮ ЛЕСОПЕРЕВОЗОК ДЛЯ РАБОТЫ НА WEB-СЕРВЕРЕ**

***Аннотация.** Стандартная комплектация мультиагентной платформы JADE предполагает использование графической среды для запуска и уничтожения агентов администратором сервера, в условиях реального web-сервера это невозможно. Агенты должны динамически создаваться и уничтожаться в процессе работы приложения на сервере, без участия администратора. Метод решения проблемы автоматизации функционирования приложения JADE предлагается в данной работе.*

***Ключевые слова:** мультиагентная платформа, интеллектуальные агенты, стандарты FIPA, язык коммуникации агентов ACL, роевой интеллект*

Технология мультиагентных систем (МАС) с использованием специальных графических средств их проектирования позволяет создавать распределенные ИС, в которых автономные интеллектуальные агенты обмениваются сообщениями, чтобы эффективно решать сложные задачи в среде распределенных взаимодействующих объектов. В работе [1] рассматривался проект мультиагентного приложения для оптимизации перевозок леса на основе агентной платформы JADE. JADE (Java Agent Development Environment) – открытая платформа для разработки мультиагентных систем, поддерживающая стандарты FIPA. JADE включает runtime-окружение (динамическую среду), в котором могут жить агенты, библиотеку классов, облегчающую разработку агентов, и набор графических утилит для администрирования и мониторинга активных агентов. Рассмотренное в [1] логистическое приложение включало агентов-перевозчиков (ForestSellerAgent) и агентов-заказчиков (ForestBuyerAgent), подбирающих по поручению своих владельцев вариант перевозки груза древесины в соответствии с оптимальными значениями желаемых критериев. Параметрами перевозчика являются его грузоподъемность и длительность перевозки. Заказчику требуется подобрать перевозчика, значение грузоподъемности которого не меньше веса перевозимого груза, и обеспечивающего минимальную стоимость перевозки. Особенностью приложения [1] является то, что заказчику для ведения переговоров в динамической среде JADE необходимо было знать идентификатор агента, имеющего структуру имя@адрес, где адрес – IP адрес платформы, где находится агент. Это неудобно, так как если в исходном коде используются конкретные IP-адреса,

то это затрудняет переносимость приложения. Другим недостатком приложения [1] является использование графического интерфейса для модификации каталога перевозчика, так как при запуске приложения в условиях Web-сервера использование графического интерфейса в общем случае недопустимо. Методы преодоления перечисленных недостатков рассматриваются в настоящей работе.

Для того чтобы избавиться от необходимости указания IP-адресов агентов, участвующих в переговорах в динамической среде JADE, в среде разработки MAC на платформе JADE предусмотрен "сервис желтых страниц", чтобы позволить агентам-заказчикам динамически узнавать об агентах-перевозчиках, доступных в данный момент времени. Платформа JADE предлагает интерфейс агента-фасилитатора DF (Directory Facilitator) agent. Он позволяет взаимодействовать с другими DF-агентами и регистрировать, модифицировать, а также искать агентов по их характеристикам во всей сети объединения DF-агентов. DF-агент реализует сервис "yellow pages" для других агентов, которые могут зарегистрировать свои услуги с помощью DF-агента или сделать запрос с помощью DF-агента найти поддержку у других DF-агентов, которые, в свою очередь, могут объединяться и создавать федерации агентов-фасилитаторов. Таким образом, DF-агент – менеджер директорий (согласно спецификации FIPA), представляет службу, где агенты могут публиковать информацию о предоставляемых ими сервисах. С помощью DF агент может находить агентов, предоставляющих необходимые ему сервисы, и вступать с ними в переговоры. Внутри одной платформы может существовать несколько DF, например, предоставляющих информацию о различных группах сервисов или о сервисах различных групп агентов. Платформа JADE по умолчанию поддерживает агента DF (его локальное имя "df"). Могут быть запущены и объединены несколько DF-агентов (включая тех, что находятся в платформе по умолчанию), для создания единого распределённого каталога "жёлтых страниц". Так как DF является агентом, с ним можно взаимодействовать стандартным образом, обмениваясь ACL-сообщениями, используя, согласно спецификациям FIPA, соответствующий язык содержания (язык SLO) и соответствующую онтологию (онтологию FIPA-agent-management). Чтобы упростить эти взаимодействия, JADE предоставляет класс `jade.domain.DFService`, с помощью которого можно публиковать и выполнять сервисы через вызовы методов.

Агент, желающий опубликовать (сделать доступными) один или более сервисов, должен предоставить DF для каждого публикуемого сервиса описание, включающее тип сервиса (класс `DFAgentDescription`), его имя, языки и онтологии, необходимые для его использования (класс `ServiceDescription`), и свойства, специфичные для данного сервиса (класс `Property`), все классы входят в состав пакета входящие в состав пакета `jade.domain.FIPAAgentManagement`.

Применительно к классу `ForestSellerAgentDF` публикация сервисов агента-заказчика с помощью DF-агента сервиса "желтых страниц" выпол-

нена в методе setup() с помощью статического метода register() класса DFService. Значение типа публикуемого сервиса устанавливается равным "forest-selling". В данном случае описание языков, онтологии и свойств, специфичных для данного сервиса, не требуется.

```
import jade.core.Agent;
import jade.core.behaviours.*;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;
import jade.domain.DFService;
import jade.domain.FIPAException;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.ServiceDescription;
import java.util.*;

public class ForestSellerAgentDF extends Agent {
//Catalog of delivered weights for sale (maps weight to price)
    private Hashtable catalogue;

    protected void setup() {
        // Create the catalog
        catalogue = new Hashtable();

        // Register the forest-selling service in the yellow pages
        DFAgentDescription dfd = new DFAgentDescription();
        dfd.setName(getAID());
        ServiceDescription sd = new ServiceDescription();
        sd.setType("forest-selling");
        sd.setName("JADE-forest-trading");
        dfd.addServices(sd);
        try {
            DFService.register(this, dfd);
        }
        catch (FIPAException fe) {
            fe.printStackTrace();
        }

        // Add the behaviour serving queries from buyer agents
        addBehaviour(new OfferRequestsServer());

        // Add the behaviour serving purchase orders from buyer agents
        addBehaviour(new PurchaseOrdersServer());

        // Add the behaviour serving update catalog from catalog agents
        addBehaviour(new CatalogUpdateServer());
    }
}
```

При завершении работы агента надо отменить регистрацию опубликованных сервисов:

```
protected void takeDown() {
// Deregister from the yellow pages
try {
DFService.deregister(this);
}
catch (FIPAException fe) { fe.printStackTrace();
}
```

Агент, осуществляющий поиск некоторых сервисов, предоставляет агенту DF описание шаблона поиска. Результатом поиска является список всех описаний, которые удовлетворяют указанному шаблону. Описание соответствует шаблону, если все поля, указанные в шаблоне, присутствуют с теми же значениями и в описании. Для класса ForestBuyerAgentDF поиск сервисов агентов-продавцов с помощью DF-агента сервиса "желтых страниц" выполнен также в методе setup() с помощью статического метода search() класса DFService, поиск осуществляется по требованию соответствия типу сервиса "forest-selling". Обновление списка известных агентов-перевозчиков осуществляется каждый раз перед попыткой сделать заказ на перевозку, так как агенты-перевозчики в динамической среде в общем случае могут появляться и исчезать:

```
import jade.core.Agent;
import jade.core.AID;
import jade.core.behaviours.*;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;
import jade.domain.DFService;
import jade.domain.FIPAException;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.ServiceDescription;
public class ForestBuyerAgentDF extends Agent {
...
private AID[] sellerAgents;
protected void setup() {
...
addBehaviour(new TickerBehaviour(this, 10000) {
protected void onTick() {
DFAgentDescription template = new DFAgentDescription();
ServiceDescription sd = new ServiceDescription();
sd.setType("forest-selling");
template.addServices(sd);
try {
DFAgentDescription[] result = DFService.search(myAgent, tem-
plate);
sellerAgents = new AID[result.length];
```

```

        for (int i = 0; i < result.length; ++i) {
            sellerAgents[i] = result[i].getName();
        }
    }
    catch (FIPAException fe) {
        fe.printStackTrace();
    }
    addBehaviour(new RequestPerformer());
}
});
...

```

Следующим шагом по адаптации рассмотренного в [1] приложения к работе на Web-сервере является избавление от графического интерфейса для обновления хеш-каталога агента-перевозчика. Это удобнее всего осуществить добавлением еще одного агента-обновителя каталога, который будет при вызове в качестве параметров получать имя агента-перевозчика для обновления каталога и значения грузоподъемности, длительности и стоимости перевозки, которые надо добавить в хеш-каталог этого агента-перевозчика. Поведение агента-обновителя каталога ForestCatalogAgentDF должно осуществляться в режиме TickerBehaviour через интервал времени, например 10000мс, пока искомый каталог не станет доступен для обновления, после чего агент-обновитель каталога и агент-перевозчик вступают в переговоры, по завершении которых (при получении строки "YES" от агента-перевозчика) агент-обновитель каталога уничтожается:

```

import jade.core.Agent;
import jade.core.AID;
import jade.core.behaviours.*;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;
import jade.domain.FIPAException;
import jade.domain.FIPAAgentManagement.ServiceDescription;
public class ForestCatalogAgentDF extends Agent {
    // The weight needed to deliver
    private AID Deliver;
    private Integer CargoWeight;
    private Integer Price;
    private Integer Duration;

    // Put agent initializations here
    protected void setup() {

        // Get the weight needed to deliver as a start-up argument
        Object[] args = getArguments();
        if (args != null && args.length > 0) {
            String t_w = (String) args[0];

```



```

        String strArr[] = t_w.split("-");
        String NameOfAgent = strArr[0];
        CargoWeight = Integer.parseInt(strArr[1]);
        Price = Integer.parseInt(strArr[2]);
        Duration = Integer.parseInt(strArr[3]);
        Deliver = new AID(NameOfAgent);
        // Printout a welcome message
        System.out.println("Hello!  Catalog-agent  "+getAID().getName()+"  is
ready to update the catalog of " + Deliver.getName());
        Integer t_weight = CargoWeight;
        Integer t_price = Price;
        Integer t_duration = Duration;
        // Add a TickerBehaviour that schedules a request to the catalog
// of the agent every 10000 ms
        addBehaviour(new TickerBehaviour(this, 10000) {
            protected void onTick() {

                addBehaviour(new AddCatalogPerformer());
            }
        });
    }
    else {
        // Make the agent terminate immediately
        System.out.println("No price, duration, AID and cargo weight specified");
        doDelete();
    }
}
// Put agent clean-up operations here
protected void takeDown() {
    // Printout a dismissal message
    System.out.println("Catalog-agent  "+getAID().getName()+"  is terminat-
ing.");
}

/**
    Inner class AddCatalogPerformer.
    This is the behaviour used by Forest-catalog agents to request seller
agents to update the catalog.
*/
private class AddCatalogPerformer extends Behaviour {
    private boolean finished = false;
    private String AddToCatalog;
    private MessageTemplate mt; // The template to receive replies
    private int step = 0;
    public void action() {

```

```

switch (step) {
case 0:
// Send the cfp to the target seller
    ACLMessage cfp = new ACLMessage(ACLMessage.CFP);
    cfp.addReceiver(Deliver);
    AddToCatalog = String.valueOf(CargoWeight.intValue()) + "-" +
String.valueOf(Price.intValue()) + "-" + String.valueOf(Duration.intValue());
    cfp.setContent(AddToCatalog);
    cfp.setConversationId("forest-trade-catalog");
//Unique value
    cfp.setReplyWith("cfp"+System.currentTimeMillis());
    send(cfp);
    mt = MessageTemplate.and(MessageTemplate.MatchConversationId(
"forest-trade-catalog"), MessageTem-
plate.MatchInReplyTo(cfp.getReplyWith()));
    step = 1;
    break;
case 1:
// Receive answer from seller agents
    ACLMessage reply = receive(mt);
    if (reply != null) {
// Reply received
        String seller_ans = (String)reply.getContent();
        if (seller_ans.lastIndexOf("YES") > -1) {
            finished = true;
            doDelete();
        }
    }
    else {
        block();
    }
    break;
}
}
public boolean done() {
return finished;
}
} // End of inner class AddCatalogPerformer
}

```

К описанию класса агента-перевозчика ForestSellerAgentDF добавляется описание его нового поведения для ведения переговоров с агентом ForestCatalogAgentDF:

```

private class CatalogUpdateServer extends CyclicBehaviour {
    public void action() {

```

```

        MessageTemplate m1 =
MessageTemplate.MatchPerformative(ACLMessage.CFP);
        MessageTemplate m2 =
MessageTemplate.MatchConversationId("forest-trade-catalog");
        MessageTemplate mt = MessageTemplate.and(m1, m2);
        ACLMessage msg = receive(mt);
        if (msg != null) {
            // CFP Message received. Process it
            String answer = (String)msg.getContent();
            String strArr[] = answer.split("-");
            Integer t_w = Integer.parseInt(strArr[0]);
            Integer t_p = Integer.parseInt(strArr[1]);
            Integer t_d = Integer.parseInt(strArr[2]);
            String pr_and_dur = t_p.toString() + " " + t_d.toString();
            updateCatalogue(t_w, pr_and_dur);
            ACLMessage reply = msg.createReply();
            reply.setPerformative(ACLMessage.PROPOSE);
            reply.setContent("YES");
            send(reply);
        }
        else {
            block();
        }
    }
} // End of inner class CatalogUpdateServer

public          void          updateCatalogue(
                final          int          weight,          final          String          price)
{
    addBehaviour(new OneShotBehaviour() {
        public void action() {
            catalogue.put(new Integer(weight), price);
            Integer t_w = weight;
            String strArr[] = price.split(" ");
            Integer t_p = Integer.parseInt(strArr[0]);
            Integer t_d = Integer.parseInt(strArr[1]);
            System.out.println("Delivery service of " + t_w.toString() + " kg for
Price " + t_p.toString() + " during " + t_d.toString() + " days inserted into cata-
logue of " + getAID().getName());
        }
    });
}

```

Теперь необходимо определиться с тем, как будут агенты всех трех типов запускаться на сервере. Для их совместной работы на платформе должны быть запущены контейнеры – динамические среды исполнения

МАС, в которых находятся агенты. Каждый контейнер может содержать несколько агентов. Набор активных контейнеров и образует платформу JADE. Первым контейнером при старте платформы должен быть главный (Main-container), а все остальные являются неглавными контейнерами и должны обладать информацией, как найти главный контейнер, на котором они будут регистрироваться, то есть должны иметь данные о хосте и порте. Пользователю не обязательно знать, как работает динамическая среда JADE, но она должна быть запущена перед началом выполнения агентов.

Таким образом, для функционирования системы администратором должна быть запущена динамическая среда JADE вместе с агентами-перевозчиками в главном контейнере. В условиях рассматриваемой задачи можно принять, что агенты-перевозчики должны быть постоянно активными на сервере, а если необходимо удалить или добавить нового перевозчика, это может сделать администратор при помощи команды [2]:

```
java -cp ... jade.Boot -container Josef: ForestSellerAgentDF ()
```

Но подобный метод для запуска агентов-покупателей и агентов ForestCatalogAgentDF не подходит, так как в этом случае при запуске для каждого агента будет создаваться свой контейнер (опция -container), но при уничтожении агента контейнер не уничтожается, а остается на платформе пустым. Так как запуски агентов-заказчиков и агентов ForestCatalogAgentDF будут многократны, то это быстро приведет к замусориванию платформы пустыми контейнерами. Поэтому проблему динамического запуска агентов МАС надо решать через создание отдельного Java-класса, который будет создавать и уничтожать контейнеры на запущенной платформе, а также запускать и уничтожать агентов в созданных контейнерах.

В JADE [3] реализован интерфейс, который позволяет внешним Java-приложениям использовать JADE в качестве своего рода библиотеки и для динамического запуска JADE Runtime из самого приложения.

Экземпляр среды выполнения JADE можно получить с помощью статического метода `jade.core.Runtime.instance()`, который предоставляет два метода: первый – для создания основного контейнера JADE (Main Container) и второй – для создания контейнера, который присоединяется к существующему Main Container. Оба метода требуют передачи в качестве параметра объект `jade.core.Profile`, в котором хранятся параметры конфигурации (например, имя хоста и номер порта основного контейнера), необходимого для запуска среды выполнения JADE.

Оба этих метода среды выполнения возвращают объект-обертку, принадлежащий пакету `jade.wrapper`, который обортывает высокоуровневой функциональностью контейнеров агентов, такой как установка и удаление MTP (протокола передачи сообщений), уничтожение контейнера (когда только контейнер уничтожается, а внешнее приложение остается) и создание новых агентов.

Метод `createNewAgent` этого контейнера-оболочки также возвращает объект-оболочку, который оборачивает агента некоторыми функциями, но все же сохраняет некоторую автономию агентов. В частности, приложение

может управлять жизненным циклом агента, но не может получить прямой доступ к нему по ссылке на объект `agent` и, как прямое следствие, из внешнего приложения нельзя выполнять вызовы методов для этого объекта (агента). Следует обратить внимание, что после создания агента его еще нужно запустить с помощью метода `start()`.

Для динамического запуска агента-заказчика надо создать следующий класс, который получает при запуске в качестве аргумента строку с параметрами заказа (вес груза и предельная стоимость) и второй аргумент – постфикс имени контейнера, в котором будет запущен агент-заказчик.

```
import jade.core.Runtime;
import jade.core.Profile;
import jade.core.ProfileImpl;
import jade.wrapper.*;

public class StartBuyerAgent{
    public static void main(String[] arg) {
        Object args[] = new Object[]{arg[0]};
        // Get a hold on JADE runtime
        Runtime rt = Runtime.instance();
        // Create a default profile
        Profile p = new ProfileImpl();
        p.setParameter(Profile.CONTAINER_NAME,          "Buyer-Container-
"+arg[1]);
        // Create a new non-main container, connecting to the default
        // main container (i.e. on this host, port 1099)
        ContainerController cc = rt.createAgentContainer(p);
        // Create a new agent, a ForestBuyerAgentDF
        // and pass it a reference to an Object
        try
        {
            AgentController    cat_ag    =    cc.createNewAgent("BAg",
                "ForestBuyerAgentDF", args);
            cat_ag.start();
        }
        catch(StaleProxyException ex)
        {
            System.out.println("Creating Agent Error");
        }
        try
        {
            cc.kill();
        }
        catch(StaleProxyException ex)
        {
            System.out.println("Killing Container Error");
        }
    }
}
```

```

    }
  }
}

```

В приведенном коде команда `cc.kill()` уничтожает временный контейнер `cc`, который был создан для динамического запуска агента-заказчика.

Аналогично можно написать код Java-приложения для динамического запуска агента `ForestCatalogAgentDF`, которое в качестве аргумента при запуске получает строку, содержащую полное системное имя агента-перевозчика, чей каталог надо обновить, и данные, которые надо добавить в хеш-каталог агента-перевозчика (грузоподъемность, длительность перевозки и ее стоимость), второй аргумент – постфикс имени контейнера, в котором будет запущен `ForestCatalogAgentDF`:

```

import jade.core.Runtime;
import jade.core.Profile;
import jade.core.ProfileImpl;
import jade.wrapper.*;
public class StartCatalogAgent{
    public static void main(String[] arg) {
        Object args[] = new Object[]{arg[0]};
        Runtime rt = Runtime.instance();
        Profile p = new ProfileImpl();
        p.setParameter(Profile.CONTAINER_NAME,          "Catalog-Container-
"+arg[1]);
        ContainerController cc = rt.createAgentContainer(p);
        try
        {
            AgentController    cat_ag    =    cc.createNewAgent("CAg",
"ForestCatalogAgentDF", args);
            cat_ag.start();
        }
        catch(StaleProxyException ex)
        {
            System.out.println("Creating Agent Error");
        }
        try
        {
            cc.kill();
        }
        catch(StaleProxyException ex)
        {
            System.out.println("Killing Container Error");
        }
    }
}

```

Результат работы приложения в командном окне, из которого был запущен Main Container, можно видеть на рисунке 1. Желтым выделены сообщения, соответствующие моментам запуска новых агентов из второго командного окна, ответные сообщения агента-перевозчика при обмене сообщениями по протоколу приложения между этими новыми агентами и агентами-перевозчиками, и сообщения, соответствующие уничтожению агентов, запущенных из второго командного окна. На рисунке 2 показан скриншот второго командного окна с командами запуска агента-обновителя каталога для обновления агента-перевозчика John (выделено желтым вверху) и агента-покупателя Katya (выделено внизу), сообщениями, соответствующими протоколу их работы, и сообщения об их уничтожении. Для запуска агента-покупателя Katya, который хочет перевезти 280кг за 3000руб используется команда следующего вида:

```
java -cp lib\jade.jar;<lib\..;> StartBuyerAgent 280-3000 Katya
```

В дальнейшем рассмотренное в настоящей работе приложение необходимо расширить новым функционалом, чтобы заказчик указывал не только вес и стоимость перевозки, но и пункт назначения.

Таким образом, для запуска МАС, обслуживающую логистику лесоперевозок, необходимо на сервере запустить динамическую среду JADE, в Main Container которой запущены и постоянно активны агенты-перевозчики. Затем по запросу из клиентской части Web-приложения (сайта) на сервере запускаются Java-приложения, динамически запускающие и уничтожающие на работающей платформе агентов-заказчиков или агентов-обновителей хеш-каталогов агентов-перевозчиков, код которых приведен в настоящей работе.

Эти Java-приложения могут быть запущены из серверного кода Web-приложения (сайта) с использованием функции `php exec()`, которая позволяет запустить на выполнение Java-приложение с переданными значениями аргументов, значения которых, в свою очередь, были переданы на сервер по запросу из клиентской части Web-приложения. Между тем, так как на сервере уже запущены и функционируют агенты перевозчики, то они будут реализовать свое поведение в соответствии с протоколом переговоров в ответ на запросы агентов-заказчиков и агентов-обновителей каталогов. Выходной поток функции `exec()` нужно правильно декодировать (парсить) с тем, чтобы сервер мог передать в браузер пользователю результаты работы запущенных им агентов.

```

Командная строка - StartForestTradingDFCatalog1.bat
c:\JADE>StartForestTradingDFCatalog1.bat
c:\JADE>java -cp lib\jade.jar;FTC_classes;lib\Forest_DeliveringServiceDFCat.jar
jade.Boot -agents John:ForestSellerAgentDF
эт. 02, 2022 10:30:09 PM jade.core.Runtime beginContainer
INFO:
-----
This is JADE 4.5.0 - revision 6825 of 23-05-2017 10:06:04
downloaded in Open Source, under LGPL restrictions,
at http://jade.tilab.com/
-----
эт. 02, 2022 10:30:10 PM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://192.168.0.20:1099
эт. 02, 2022 10:30:11 PM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
эт. 02, 2022 10:30:11 PM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
эт. 02, 2022 10:30:11 PM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
эт. 02, 2022 10:30:11 PM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
эт. 02, 2022 10:30:11 PM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
эт. 02, 2022 10:30:11 PM jade.mtp.http.HTTPServer <init>
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParser
Impl$JAXPSAXParser
эт. 02, 2022 10:30:11 PM jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://Pavilion:7778/acc
эт. 02, 2022 10:30:11 PM jade.core.AgentContainerImpl joinPlatform
INFO:
-----
Agent container Main-Container@192.168.0.20 is ready.
-----
эт. 02, 2022 10:31:40 PM jade.core.PlatformManagerImpl localAddNode
INFO: Adding node <Catalog-Container-John> to the platform
эт. 02, 2022 10:31:40 PM jade.core.PlatformManagerImpl$1 nodeAdded
INFO: --- Node <Catalog-Container-John> ALIVE ---
Delivery service of 300 kg for Price 3000 during 3 days inserted into catalogue
of John@192.168.0.20:1099/JADE
эт. 02, 2022 10:32:00 PM jade.core.nodeMonitoring.BlockingNodeFailureMonitor ru
n
INFO: PING from node Catalog-Container-John returned [EXIT]
эт. 02, 2022 10:32:00 PM jade.core.PlatformManagerImpl removeTerminatedNode
INFO: --- Node <Catalog-Container-John> TERMINATED ---
эт. 02, 2022 10:32:00 PM jade.core.PlatformManagerImpl localRemoveNode
INFO: Removing node <Catalog-Container-John> from the platform
эт. 02, 2022 10:32:17 PM jade.core.PlatformManagerImpl localAddNode
INFO: Adding node <Buyer-Container-Katya> to the platform
эт. 02, 2022 10:32:17 PM jade.core.PlatformManagerImpl$1 nodeAdded
INFO: --- Node <Buyer-Container-Katya> ALIVE ---
Delivery of 300 kg will be done for buyer-agent BAg@192.168.0.20:1099/JADE
эт. 02, 2022 10:32:37 PM jade.core.nodeMonitoring.BlockingNodeFailureMonitor ru
n
INFO: PING from node Buyer-Container-Katya returned [EXIT]
эт. 02, 2022 10:32:37 PM jade.core.PlatformManagerImpl removeTerminatedNode
INFO: --- Node <Buyer-Container-Katya> TERMINATED ---

```

Рис.1 Скриншот командного окна с запуском Main Container с агентами-перевозчиками внутри него



```

Командная строка
INFO: Service jade.core.event.Notification initialized
эт. 02, 2022 10:31:40 PM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Catalog-Container-John@192.168.0.20 is ready.
-----
Hello! Catalog-agent CAg@192.168.0.20:1099/JADE is ready to update the catalog of
f John@192.168.0.20:1099/JADE
Catalog-agent CAg@192.168.0.20:1099/JADE is terminating.
эт. 02, 2022 10:32:00 PM jade.core.messaging.messageManager shutdown
INFO: MessageManager shutting down ...

c:\JADE>Start_Buyer_Agent1.bat

c:\JADE>"C:\Program Files\Java\jdk1.8.0_221\bin\javac.exe" -classpath lib\jade.j
ar;lib\Forest_DeliveringServiceDFCat.jar -d FTC_classes src\StartBuyerAgent.java

c:\JADE>"C:\Program Files\Java\jdk1.8.0_221\bin\jar.exe" -cmf new\META-INF\MANIF
EST.MF lib\My_StartBuyerAgent.jar -C FTC_classes .

c:\JADE>java -cp lib\jade.jar;FTC_classes;lib\My_StartBuyerAgent.jar;lib\Forest
DeliveringServiceDFCat.jar StartBuyerAgent 280-3000 Katya
эт. 02, 2022 10:32:16 PM jade.core.Runtime beginContainer
INFO: -----
This is JADE 4.5.0 - revision 6825 of 23-05-2017 10:06:04
downloaded in Open Source, under LGPL restrictions,
at http://jade.tilab.com/
-----
эт. 02, 2022 10:32:17 PM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://192.168.0.20:54941

эт. 02, 2022 10:32:17 PM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
эт. 02, 2022 10:32:17 PM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
эт. 02, 2022 10:32:17 PM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
эт. 02, 2022 10:32:17 PM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
эт. 02, 2022 10:32:17 PM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
эт. 02, 2022 10:32:17 PM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Buyer-Container-Katya@192.168.0.20 is ready.
-----
Hello! Buyer-agent BAg@192.168.0.20:1099/JADE is ready.
Trying to order the delivery of 280 kg for 3000 RUR
Be found next agents:
John@192.168.0.20:1099/JADE with price 3000 during 3 days.
Delivery of 280 kg will be done by John@192.168.0.20:1099/JADE
My Cost of delivery service = 3000
Buyer-agent BAg@192.168.0.20:1099/JADE is terminating.
эт. 02, 2022 10:32:37 PM jade.core.messaging.MessageManager shutdown
INFO: MessageManager shutting down ...

c:\JADE>

```

Рис.2. Скриншот второго командного окна с запуском агента-покупателя и агента-обновителя каталога для агента-перевозчика, запущенного в Main Container

### Библиографический список

1. Разработка мультиагентного приложения для оптимизации перевозок леса на основе агентной платформы JADE. Шилкина М.Л., Васильев Н.П. В сборнике: Информационные системы и технологии: теория и практика. сборник научных трудов. Санкт-Петербургский государственный лесотехнический университет имени С.М. Кирова. Санкт-Петербург, 2021. С. 171-184.
2. <https://jade.tilab.com/doc/administratorsguide.pdf>
3. <https://jade.tilab.com/doc/programmersguide.pdf>

**М.А.Шубина**, кандидат технических наук, доцент  
Кафедра информационных систем и технологий  
СПбГЛТУ им.С.М.Кирова  
nemsha@mail.ru

## **О ХРАНЕНИИ ПРОСТРАНСТВЕННЫХ ДАННЫХ**

*Аннотация.* Рассмотрены вопросы выбора организации и программных средств, используемых для хранения пространственных данных, в том числе космических данных.

*Ключевые слова:* пространственные данные (ПД, Spatial data, Geographical data, Geospatial data, Georeferenced data), системы управления данными (СУБД), хранилища данных (ХД), PostGIS.

### **Введение.**

Вопросы организации хранения данных вообще и пространственных данных в особенности становятся с течением времени все более актуальными. Развиваются уже испытанные и появляются новые программные средства для этой цели. Необходимо рассмотреть их новые возможности. В первую очередь это касается пространственных данных, получаемых с космических аппаратов (КА), применяемых для дистанционного зондирования Земли (ДЗЗ). Это нашло отражение в принятии в конце декабря 2021 года государственной программы «Национальная система пространственных данных» (НСПД), в том числе, в изменениях в законодательстве, создании новых цифровых сервисов для граждан и бизнеса, трансформации контрольно-надзорной деятельности.

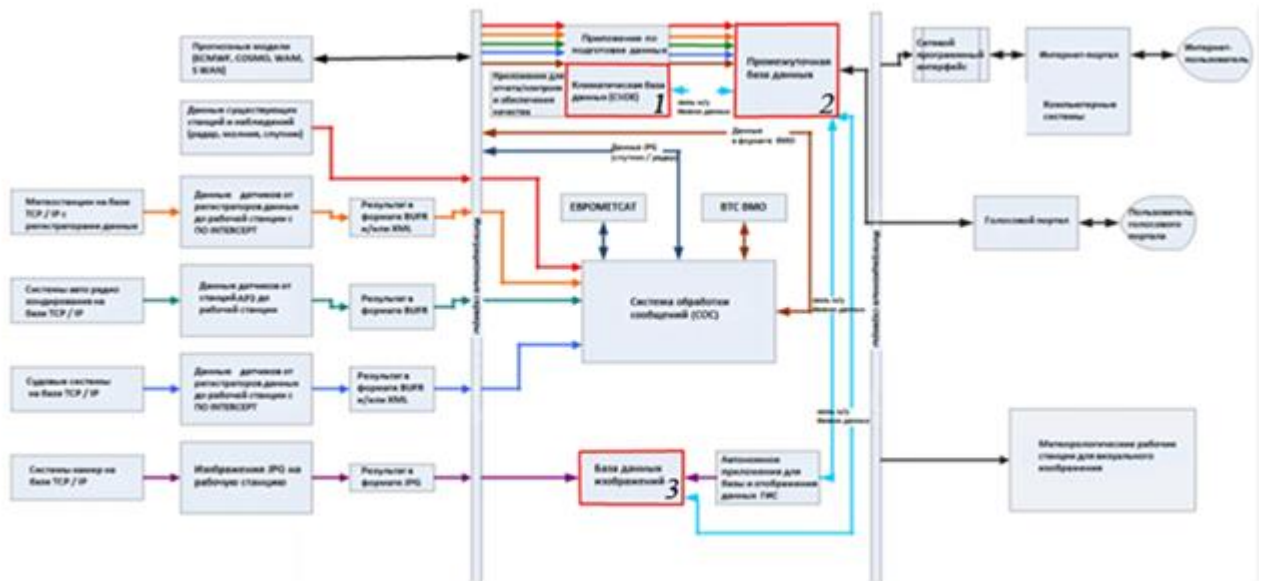
К настоящему времени разработана и постоянно развивается теория построения баз данных, которая существенно зависит от цели создания, типа исходных данных, типа связей между данными, количественных характеристик, выбранного способа представления данных на разных этапах работы с ними. С самого начала возникли разные подходы к хранению данных.

Пространственные данные (ПД) - цифровые данные о пространственных объектах, включают сведения об их местоположении и свойствах, пространственных и непространственных атрибутах, представленные в координатно-временной системе», отличаются многообразием форматов, необходимостью преобразования форматов в процессе хранения, визуализации в виде карт, планов и другими особенностями.

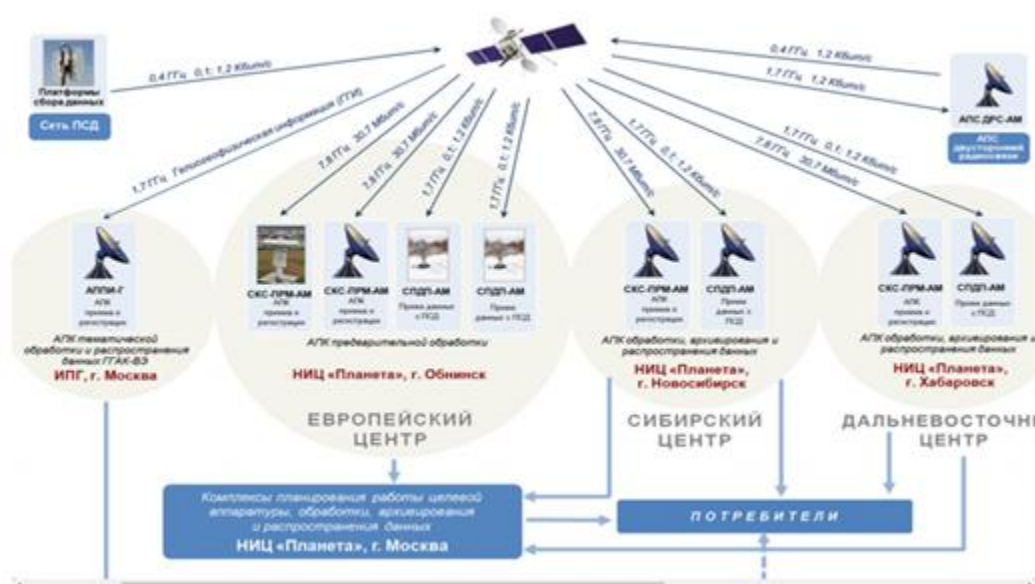
Примерами систем сбора, хранения и распространения пространственных данных могут служить системы климатических наблюдений компаний BARON и НИЦ «Планета» (рис. 1 – рис.4).

Архив НИЦ Планета ведется с 1979 года, включает данные, полученные с российских и зарубежных космических аппаратов и информаци-

онную продукцию в виде фотоматериалов, в цифровом формате – более 6 млн Тб.



**Рис. 1.** Пример организации системы накопления и представления пользователям данных климатических наблюдений компании Varon [ 6 ]. Красным цветом выделены климатическая БД (1), база промежуточных данных (2) и база данных изображений (3).



**Рис. 2.** Организация сбора данных и представления пользователям данных климатических наблюдений НИЦ «Планета» [ 5 ]. Слева сверху рисунка – платформа сбора данных.

НИЦ Планета совместно с ИКИ РАН, РГРТУ, ИВМиМГ СО РАН, СПбГУ, МГУ, ИВТ СО РАН, ЮНИИ ИТ, ВЦ ДВО РАН создано и внедрено более 40 специализированных программных комплексов (ПК) обработки спутниковых данных для решения задач гидрометеорологии и монито-



ринга окружающей среды: многофункциональные ПК для обработки спутниковых данных и построения тематических карт разного назначения.



Рис. 3. Схема взаимодействия баз данных и распространения данных в НИЦ Планета [5]

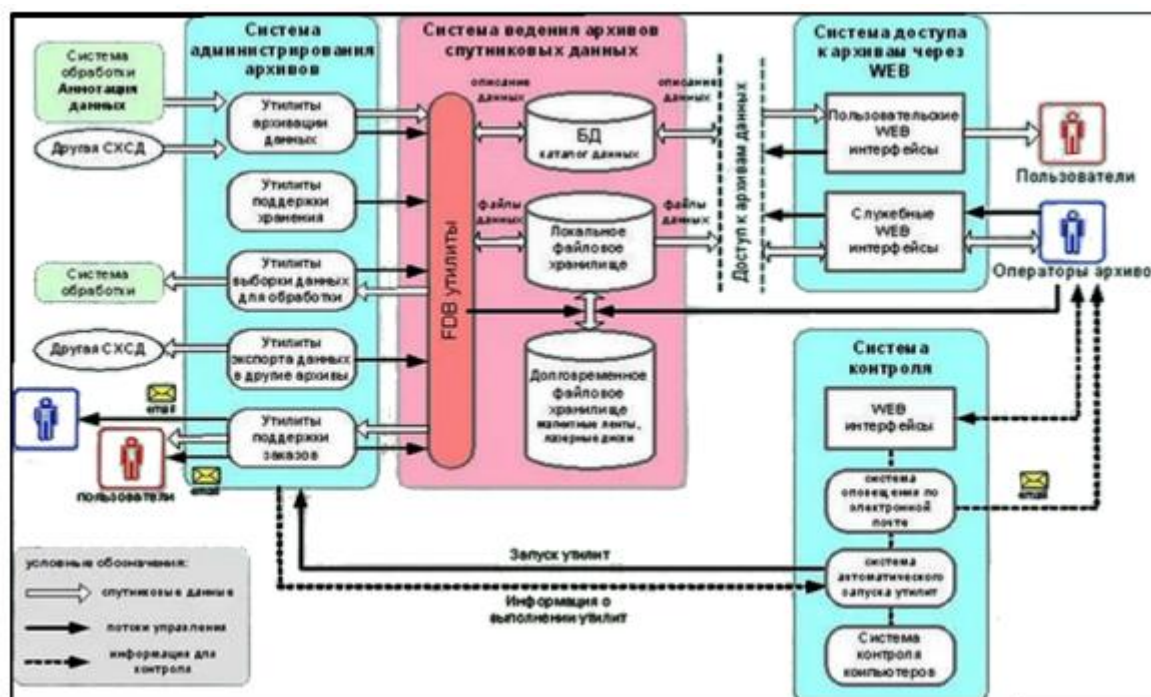


Рис. 4. Система хранения спутниковых данных ИКИ.

Основные модели баз данных, их классификация, теория и разработанные инструменты для работы с данными, ряд СУБД используются и применительно к пространственным данным, но привязка данных к местоположению, увеличение количественных характеристик и представления данных потребовала создать новые программно-аппаратные средства, ча-

сто сочетающие хранение, обработку, распространение, и визуализацию данных. Значительные изменения связаны с использованием облачных инфраструктур. Так, было заключено соглашение о сотрудничестве Google Cloud и компании Planet с целью объединения данных высокочастотных съемок Земли.

В настоящее время запускают спутники несколько десятков стран, перечень даже основных позволяет понять сложность организации взаимодействия операторов, ответственных за хранение данных. Естественно, каждый разработчик использует собственные форматы данных, сервисы, программы обработки данных, схемы хранения, хотя сами данные используются совместно, поэтому задачи консолидации данных в профессиональных областях представляют собой сложную задачу.

Из числа основных операторов получения данных ДЗЗ можно назвать следующие.

Компания Maxar Technologies объединяет четыре ведущих бренда коммерческих космических технологий — SSL, MDA, DigitalGlobe и Radiant, один из мировых лидеров в отрасли ДЗЗ сверхвысокого разрешения. Компания Planet (США) начала разворачивать космическую группировку наноспутников дистанционного зондирования Земли PlanetScore (около 150 спутников). Компания Airbus Defence and Space оператор оптических спутников высокого и сверхвысокого разрешения SPOT и Pleiades, радарных спутников нового поколения TerraSAR-X и TanDEM-X. Основными сферами деятельности китайской компании BeijingSpaceEyeInnovationTechnologyCo., Ltd. (BSEI) являются дистанционное зондирование Земли, геоинформационные и навигационные технологии. SpaceWill Info. Co., Ltd. (быв. Beijing Space View Technology Co., Ltd.) является ведущим поставщиком спутниковых данных и геоинформационных сервисов в Китае. Компания является оператором группировки спутников сверхвысокого разрешения SuperView, а также дистрибьютором данных с китайских спутников ДЗЗ GF-4, GF-3, GF-2, GF-1, ZY-3, ZY-3 02, HJ-1A,B и др.

ОАО «Российские космические системы» — ведущее предприятие российской ракетно-космической отрасли, специализирующееся на разработке, изготовлении, осуществляет прием, регистрацию, обработку, архивацию, каталогизацию и распространение космической информации с отечественных и зарубежных космических аппаратов ДЗЗ. НЦ ОМЗ — оператор спутников «Ресурс-ДК1», «Ресурс-П», «Канопус-В». Одним из основных источников материалов ДЗЗ, в том числе и бесплатных, является архив Геологической службы (ГС) США (USGS). Институт космических исследований РАН — головной академический институт по исследованию и использованию космического пространства в интересах фундаментальных наук (рис. 4). В составе спутникового мониторинга выделяются: разработка методов, технологий и систем дистанционного мониторинга состояния окружающей среды, исследования различных природных процессов и анализа воздействий, связанных с различными антропогенными факторами; создание и ведение архивов спутниковых данных и результатов их обра-

ботки, необходимых для различных научных и прикладных задач. Отделом «Технологии спутникового мониторинга» разработаны Информационная система дистанционного мониторинга Федерального агентства лесного хозяйства (ИСДМ-Рослесхоз) для мониторинга лесных пожаров, система дистанционного мониторинга земель агропромышленного комплекса (СДМЗ АПК) и др.

Для представления пространственных данных используется две основных модели данных – векторная (объектно-ориентированная) и растровая (пространственно-ориентированная) [1].

Используемые архитектуры баз пространственных данных, файл-серверная и клиент-серверная. Форматы хранения пространственных данных можно разделить на две категории: файловые форматы и хранение данных в СУБД — системах управления базами данных. Наиболее распространенные СУБД – реляционные. Реляционная СУБД - IBM System R, созданная в середине 1970-х знаменательна тем, что стала первой реализацией Structured Query Language (SQL), который и сейчас является стандартом языка запросов для реляционных баз данных.

Также для пространственных данных используются известные системы хранения данных: прямое подключение к серверу (DirectAttachedStorage, DAS), сетевые подсистемы хранения данных (Network Attached Storage, NAS), специальная выделенная сеть, объединяющая устройства хранения данных с серверами приложений (Storage Area Network, SAN).

Специфика обработки и хранения пространственных данных была учтена в создании отдельного класса геоинформационных систем (ГИС).

Для работы с пространственными данными, получаемыми с космических аппаратов (КА) используются программные продукты, отличающиеся большим разнообразием: SAGAGIS (SystemforAutomatedGeoscientificAnalyses, Геттинген), GRASSGIS (GeographicResourcesAnalysisSupportSystem, 1995-2013, разработчик USA - CERL), QGIS (QuantumGIS, 2004-2014, разработчик FreeSoftwareFoundation, USA на основе ОС GNU), ARCGIS (2005-2015, разработчик ESRI, США), PHOTOMOD (1994-2014, разработчик МФТИ, Россия), ENVI (EnvironmentforVisualizingImages, разработчик Exelis, бывшая ITTVisualInformationSolutions, США), раньше всего появившиеся на рынке – ERDAS (ERDASIMAGINE, 1978-2014, разработчики ERDAS, LeicaGeosystem, ныне Intergraph, США), ERMapper (EarthResourceMapping, разработчик ERDAS), MapInfo, GlobalMapper и др.

Среди компаний, разрабатывающих программные продукты для работы с данными, следует отметить следующие [4].

Компания ESRI — один из ведущих мировых разработчиков и поставщиков геоинформационного программного обеспечения. Программные продукты ArcGIS, создаваемые на основе самых современных компьютерных технологий, поддерживают все открытые (OGC) стандарты, что позволяет использовать их во многих прикладных сферах и на разных уровнях организации работы (настольные, серверные и мобильные ГИС).

В мае 2008 г. в Москве была учреждена Компания Esri CIS, ESRI CIS получила статус эксклюзивного международного дистрибьютора ESRI в России и странах СНГ.

Последняя версия ГИС ESRI 10.8.1 2021 г. Написана на Visual C, Python для ОС Windows, Linux включает читаемые форматы: Intergraph Standard File Formats, DWG, DXF, ADRG Overview Image File, ADRG Legend Image File, Maptech BSB documentation file, Compressed ARC Digitized Raster Graphics, Controlled Image Base, Digital Geographic Exchange Standard, Digital Terrain Elevation Data, Enhanced Compression Wavelet, Band Interleaved by Line Image File, Band Interleaved by Pixel Image File, GIF, Hierarchical Data Format, Intergraph Raster, JPEG File Interchange Format, version 1.0, JPEG 2000, MrSID, National Imagery Transmission Format, version 2.0, National Imagery Transmission Format, version 2.1, MAP, PNG, TIFF<sup>[1]</sup>, USGS DEM, BMP, X Pixmap, ESRI File Geodatabase, BigTIFF, Shapefile header index, Shapefile projections definitions file и ArcGIS Package Info, создавая форматы: Shapefile header index, Shapefile projections definitions file и ArcGIS Package Info.

База геоданных представляет собой хранилище для пространственных данных внутри РСУБД. Это хранилище содержит векторные данные, растровые данные, таблицы и другие объекты ГИС.

Для обеспечения эффективного доступа и хранения растровых изображений большого объема в базе геоданных, растровые данные автоматически режутся на отдельные фрагменты, располагаемые каскадом, и при этом сжимаются. При загрузке растровых данных можно сшивать в мозаику необходимое число таких фрагментов, чтобы обеспечить покрытие требуемой площади. Когда идет работа с большой растровой базой данных, для минимизации времени ожидания желательно определять отображаемый экстенд растрового изображения. Для ускорения процесса вывода и отображения растровых изображений используются пирамидные слои.

Carto — это веб-платформа для создания пользовательских динамических карт, отображающих в привязке к местности нужные вам данные, разработанная компанией Vizzuality. Также существует редактор Carto — приложение на базе платформы, инструмент для самостоятельного управления данными и обработки карт, есть возможность выполнять SQL-запросы данных, использовать CartoCSS — специальные каскадные таблицы стилей для карт.

Компания «Совзонд» является официальным дистрибьютором ведущих мировых компаний-операторов спутников ДЗЗ — ОАО «Российские космические системы» (Россия), DigitalGlobe (США), Planet (США), Airbus Defence and Space (Франция-Германия), MDA (Канада), e-GEOS (Италия), RESTEC (Япония), DMCii (Великобритания), Deimos (Испания), BSEI (Китай) и др., для заказчиков из России. Компания «Совзонд» поставщик решений в области ГИС технологий, ГИС интеграции и космического мониторинга [2]. Сервис АО «Терра Тех», Terra Cloud предоставляет как архивные

снимки всех видов космической съёмки с необходимыми характеристиками, начиная с 2000 г., так и заказы на новую съёмку.

Наряду с задачей организовать хранение данных, получаемых в процессе съёмки, не менее сложная задача встает перед пользователями, занимающимися их использованием для решения конкретных профессиональных задач, где к данным ДЗЗ добавляются другие необходимые данные.

В настоящее время активно развиваются помимо известных хорошо зарекомендовавших себя СУБД свободно распространяемые программные средства. Среди недавно появившихся программных средств для работы с пространственными данными можно указать, например, пакеты на основе языка R, возникшего, как ответвление языка S. В R для моделирования пространственных данных используются пакеты программ: `library(sf)`, `library(stars)`, `library(mapview)`, `library(dplyr)`, `library(readr)`.

Пакет SF появился относительно недавно (в 2016 году), и активно развивается в настоящее время. Все функции начинаются с префикса `st_`, что означает “spatiotemporal”[3]. Данные префиксы были выбраны для унификации с аналогичными названиями функций, используемых в широко распространенной СУБД PostgreSQL для оперирования объектами *Simple Features*.

- Векторные данные

1. Simple Features (официально *Simple Features Access*) — это стандарт OGC 06-103, разработанный Open Geospatial Consortium (OGC) и реализованный также в виде международного стандарта [ISO 19125](#), который определяет общую модель хранения и доступа к векторным объектам (точка, линия, многоугольник, мульти точечные и т. д.), в географических информационных системах.

Геометрическое представление пространственных объектов базируется на следующих принципах: а) все геометрии состоят из точек; б) точки являются координатами в 2-, 3- или 4-мерном пространстве в) все точки в геометрии имеют одинаковую размерность. Для описания используются координаты: X-широта, Y-долгота, Z- высота и M- мера, т.е. возможны описания сочетаниями XY, XYZ, XYZM, XYM.

Существует два официально закрепленных формата представления пакета SF: *Well-Known Text (WKT)* и *Well-Known Binary (WKB)*, которые необходимы для чтения таких данных.

*Well-Known Text (WKT)* — стандарт представления геометрии в виде множества списков координат, в которых координаты вершин разделены пробелами, вершины разделены запятыми, а компоненты полигонов и мультигеометрий заключены в круглые скобки и также разделены запятыми.

*Well-Known Binary (WKB)*— бинарный формат хранения координат. Именно этот формат фактически используется в базах данных, поскольку он обеспечивает высокую скорость чтения и записи данных (в отличие от текстового). Однако внешний вид данных в формате WKB мало о чем говорит человеку, поскольку он предназначен для чтения компьютером.



Благодаря библиотеке GDAL пакет **sf** имеет возможность читать и записывать более 90 форматов векторных данных.

Множество пространственных объектов представляется в виде таблицы атрибутов, где каждая строка соответствует объекту, а каждый столбец — атрибуту объекта. С каждой строкой таблицы должна быть ассоциирована информация о геометрии объекта, которая, в зависимости от формата данных, может либо храниться непосредственно в таблице (в специальном столбце), либо быть вынесена в отдельную структуру данных, которая связана с таблицей атрибутов посредством ключа.

Если информация о геометрии хранится в специальном столбце таблицы, каждая ячейка этого столбца соответствует геометрическому объекту. Представление геометрических объектов реализовано такими средствами, как списки, матрицы и векторы. Поскольку геометрический столбец хранит не обычные переменные, а структуры данных, он реализуется в виде так называемого *списка-колонки (list-column)*, каждый элемент которой соответствует отдельному объекту. Всего стандарт Simple Features включает в себя 17 типов геометрий. Пакет **sf** базируется на библиотеках PROJ, GDAL, CEOS, S2, которые устанавливаются вместе с ним. GDAL (C++) обеспечивает работу с 150 растровыми форматами, 90 векторными, утилитами для обработки данных, PROJ (C) – более 140 проекций для карт, алгоритмы проецирования и трансформирования координат; GEOS (C++) – операции вычислительной геометрии для двумерных объектов на плоскости; S2 (C++) – операции вычислительной геометрии для двумерных объектов на сфере.

Кроме указанных средств наиболее часто выбирают СУБД СУБД PostgreSQL с расширением PostGIS, позволяющую хранить в базе географические данные (геометрию объектов). Основным достоинством PostGIS является возможность использования языка SQL совместно с пространственными операторами и функциями. Кроме хранения данных, PostGIS позволяет осуществлять любые виды операций над ними.

PostGIS был выпущен в 2001 году компанией Refrations Research и является свободным программным продуктом с открытым исходным кодом [2]. В этой СУБД пространственные данные хранятся в таблицах, имеется таблица с числовыми идентификаторами и текстовыми систем координат. Как и в других SQL- базах можно задавать права доступа, привилегии. Каждая таблица представляет собой картографический слой, а каждая запись — геометрический объект (точка, линия, полигон). Запись в таблице описывает сам объект: координаты, форма, атрибутика и так далее. А сама таблица может содержать только один тип объектов: точка, линия или полигон. Таблица — это слой, а запись — объект слоя. Естественно, что слой может иметь только один тип. Таблица — это слой, а запись — объект слоя. Естественно, что слой может иметь только один тип.

PostGIS включает поддержку пространственных индексов R-Tree/GiST и функции обработки геоданных. Позволяет работать с геометрическими объектами: *point*, *line*, *polygon*, *multipart*, *multiline*,

multipolygon, and geometrycollections (точка, линия, полигон, мультиточка, мультилиния, мультиполигон и геометрическая коллекция). Основные стандартные способы определения пространственных объектов: уже упоминавшиеся Well-Known Text (WKT) и Well-Known Binary (WKB), включающие информацию о типе объекта и координат, составляющих объект. Расширения JDBC предоставляют соответствующие объекты Java внутренним типам PostGIS. Эти объекты могут быть использованы для написания Java-клиентов, совершающих запросы к базе данных PostGIS, получающих или обрабатывающих данные ГИС.

Наличие индексов GIST, возможности кластеризации геометрических индексов позволяют увеличить производительность работы. Кроме того, включает функции управления, геометрической связи, обработки геометрии, геометрические способы доступа, геометрические конструкторы, геометрические редакторы, функции измерения, функции ArcSDE и др.

### **Заключение.**

Таким образом, при выборе решений по хранению пространственных данных наблюдается тенденция использования сравнительно новых открытых программных средств в сочетании с привычными методами их хранения в виде слоев привязанных изображений и карт. Дальнейшее развитие систем хранения пространственных данных связано с развитием алгоритмов анализа данных, сжатия и агрегирования, а также стандартизацией решений.

### **Биографический список**

1. Кузнецов С.Д. Базы данных.- Учебник. М.: Бином.: 2012. - 496 с.
2. Руководство пользователя PostGIS. <https://gis-lab.info/docs/postgis/manual/postgis-manual-ru-1.3.4.pdf>
3. Самсонов Т.Е. Визуализация и анализ географических данных на языке R. М.: Географический факультет МГУ, 2021.
4. Шубина, М.А. Новые программные продукты ГИС-технологий.- Сборник научных трудов. Информационные системы и технологии: Теория и практика — СПб.: СПбГЛТУ, 2016. - С. 118-127.
5. <https://sovzond.ru>
6. <https://www.baronweather.com>

## ОГЛАВЛЕНИЕ

<b>Горбачеву Виктору Александровичу</b> посвящается...	3
<b>А.М. Заяц</b> Анатолий Михайлович Половко – надёжность в науке и жизни	6
<b>А.К. Бойцов, А.А. Мокринский</b> ИОТ для системы автоматического управления аппаратно-программного комплекса с использованием датчика углекислого газа в лесу	14
<b>А.О. Бутусова, А.Ю. Ничипорук, С.С. Колмогорова</b> Программная обработка «умного браслета» для людей с ограниченными возможностями	21
<b>А.Ю. Ничипорук, А.О. Бутусова, С.С. Колмогорова</b> Разработка интерфейса пользователя «умного браслета» для лиц с ограниченными возможностями	25
<b>Н.А. Карманова, А.Г. Карманов</b> Повышение надёжности систем дистанционного зондирования земли	30
<b>М.Е. Кутузов, С.С. Колмогорова</b> Программная обработка лабораторных данных с использованием искусственног интеллекта	37
<b>Е.В. Тетюев, С.П. Хабаров</b> Разработка сетевого приложения с контролем доставки и синхронизацией данных с помощью WebSocket на основе Python3	43
<b>К.Н. Черник, Д.В. Черник</b> Исследование параметров манипулятора лесной машины с применением Solidworks и Mathcad	49
<b>М.Р. Вагизов, А.М.Заяц</b> Современные проблемы цифровизации лесного хозяйства	57
<b>Р.А. Анохин, Б.М. Каримов, С.А. Кныш, К.А. Анохин, В.Я. Шапиро</b> Автоматизация расчетов технологических параметров групповой окорки лесных балансов	62
<b>Р.А. Анохин, Б.М. Каримов, С.А. Кныш, К.А. Анохин, В.Я. Шапиро</b> Автоматизация расчетов технологических параметров роторной окорки лесных балансов	67
<b>Н.П. Васильев, М.Р. Вагизов</b> WebGL для наглядного представления геополей	72
<b>М.Р. Вагизов, А.А. Гаврилюк</b> Управление лесными землями на основе геоинформационных моделей лесных экосистем	94
<b>А.А. Гаврилюк, А.О. Бутусова, А.Ю. Ничипорук, С.С. Колмогорова</b> Разработка программы управления и оценка возможности движения в одномерной проекции квадрокоптера в среде Matlab	100
<b>А.Д. Григорьев</b> Концепция Web 3.0 и её современная реализация, отличия от Web1.0 и Web 2.0	113

<b>А.М. Заяц</b>	
Графовые модели в анализе и синтезе беспроводных сенсорных сетей	118
<b>А.М. Заяц</b>	
Обзор применения технологий виртуальной и дополнительной реальности при проектировании и исследовании систем мониторинга лесных территорий	127
<b>Т.К. Екшикеев, И.А. Обухова</b>	
Использование бально-рейтинговой системы при оценке знаний студентов, формируемых в электронной информационно-образовательной среде	136
<b>С.А. Иванов</b>	
Обзор инструментов интеллектуального анализа данных современных ИТ-платформ для решения задач	141
<b>П.А. Кашо, Р.А. Яковлев, С.С. Колмогорова</b>	
Система визуального проектирования Node-RED компонентов интернета вещей в лесной отрасли	144
<b>А.П. Кичук</b>	
Взаимодействие микросервисов через API на примере ERP системы	151
<b>С.С. Колмогорова, А.С. Колмогоров, Д.С. Баранов, С.В. Бирюков</b>	
<b>О.Н. Колбина, А.Д. Романовская</b>	
Программно-технический комплекс контроля и прогнозирования опасных ситуаций	157
<b>Н.В. Меламед, А.А. Мамонтова</b>	
Использование методов многокритериального принятия решений в задачах транспортной логистики	168
<b>Д.Е. Поваров, Н.В. Меламед</b>	
Разработка программного комплекса для анализа и моделирования хода роста древостоев на базе БД ПИК ГИЛ	181
<b>Р.А. Рухлов, С.П. Хабаров</b>	
Исследование технологии VPN с использованием сети из виртуальных машин на базе Windows	201
<b>И. Б. Соловьев, С. П. Хабаров</b>	
Основы реализации многопоточной клиент-серверной архитектуры для БПЛА	209
<b>С.П. Хабаров</b>	
Использование Notepad++ для тестирования простейших сетевых Java-приложений	214
<b>М.Л.Шилкина</b>	
Адаптация мультиагентного приложения JADE по обслуживанию лесоперевозок для работы на Web-сервере	226
<b>М.А.Шубина</b>	
О хранении пространственных данных	240



*Научное издание*

Отв. редактор  
**Вагизов Марсель Равильевич**

**ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ:  
ТЕОРИЯ И ПРАКТИКА**

Сборник научных трудов

В ы п у с к 14

*В авторской редакции с готового оригинал-макета*

---

Подписано с оригинал-макета 30.05.22.

Усл.-печ. л. 15,75. Заказ № 46. С 13.

---

Санкт-Петербургский государственный  
лесотехнический университет имени С. М. Кирова  
Издательско-полиграфический отдел  
194021, Санкт-Петербург, Институтский пер., 3