



eISSN 2311-2468
Том 6, № 14. 2018
Vol. 6, no. 14. 2018

электронное периодическое издание
для студентов и аспирантов

Огарёв-онлайн Ogarev-online

<https://journal.mrsu.ru>



ФИРСОВА С. А., ТЕТИН Е. Н.

**РЕАЛИЗАЦИЯ СТАТИСТИЧЕСКОГО КОНТРОЛЯ КАЧЕСТВА
ПРИ ПРОВЕДЕНИИ ИСПЫТАНИЙ ОБРАЗЦОВ ТЕХНИКИ**

Аннотация. В статье описывается разработанная авторами программная система, предназначенная для автоматизации контроля качества работы климатической камеры. Приведен прогноз поведения качественных характеристик камеры с целью успешного прохождения ее периодической аттестации.

Ключевые слова: климатическая камера, методы аттестации оборудования, программная система, прогнозирование изменения характеристик.

FIRSOVA S. A., TETIN E. N.

**IMPLEMENTATION OF STATISTICAL QUALITY CONTROL
FOR EQUIPMENT CERTIFICATION**

Abstract. The article describes the software system developed by the authors and designed to automate the quality control of the climatic chamber. The prediction of the changes in the chamber qualitative characteristics, aimed at successful passing of its scheduled certification, is provided.

Keywords: climatic chamber, methods of equipment certification, software system, prediction of changes in characteristics.

Введение. Возможность проведения испытания в климатической камере напрямую зависит от прохождения аттестации. Аттестация климатической камеры проводится ежегодно, в результате чего формируются значительные объемы данных, которые подлежат дальнейшей обработке. В настоящее время такая обработка проводится аттестующей организацией вручную. Специалист снимает через определенные промежутки времени показания с датчиков климатической камеры, установленных в соответствии с ГОСТ Р 53618-2009 [1]. Эти показания фиксируются в соответствующем журнале. Затем проводятся расчеты характеристик работы климатической камеры по формулам, указанным в ГОСТе. Безусловно, такая процедура требует достаточно больших временных затрат, поэтому разработка приложения, обеспечивающего автоматизацию обработки данных, является актуальной задачей [2].

Новизна разработанной авторами программной системы состоит в том, что в настоящее время не существует свободно распространяемых приложений для расчета параметров, характеризующих качество работы камеры. Создание такого приложения позволит решить проблему зависимости от закрытых приложений компаний, производящих

климатические камеры. Кроме того, помимо упрощения обработки данных, можно получить хронологию изменений рассчитанных параметров и спрогнозировать последующие их изменения для принятия своевременных мер руководством организации, проводящей климатические испытания, позволяющих успешно пройти аттестационные испытания.

Исходные данные. В качестве исходных данных выступали показания датчиков, фиксирующих работу климатической камеры, устанавливаемых в период аттестации камеры в соответствии с требованиями ГОСТов.

Для описания порядка и проведения аттестации климатической камеры разработаны соответствующие ГОСТы: ГОСТ Р 53618-2009 (МЭК 60068-3-5:2001) «Требования к характеристикам камер для испытаний технических изделий на стойкость к внешним воздействующим факторам. Методы аттестации камер (без загрузки) для испытаний на стойкость к воздействию температуры» [1] и ГОСТ Р 54082-2010 (МЭК 60068-3-11:2007) «Требования к характеристикам камер для испытаний технических изделий на стойкость к внешним воздействующим факторам. Методы обработки результатов аттестации камер» [3].

Перечень операций аттестации включает: 1) выбор средств измерений: скорости циркуляции воздуха, температуры, влажности; 2) подготовку к измерению: внешний осмотр, опробование, установка датчиков; 3) измерения и вычисления характеристик климатической камеры; 4) обработку результатов произведенных измерений; 5) составление протокола по результатам измерений [4].

Перечень конкретных операций, выполняемых в соответствии с приведенным перечнем при проведении аттестации, устанавливаются в программе аттестации. Результаты первичной аттестации оформляются в соответствии с ГОСТ Р 8.568 «Аттестация испытательного оборудования. Основные положения (с Изменением N 1)», раздел 5, приложения А, Б. Результаты периодической аттестации оформляются в соответствии с ГОСТ Р 8.568, раздел 6, приложение В. Обработка результатов измерений проводится в соответствии с требованиями ГОСТ Р 54082-2010. В ходе изучения данного ГОСТа были выделены формулы для определения значений характеристик температуры и относительной влажности.

За значение температуры (относительной влажности) в точке полезного объема камеры, полученное с помощью средств аттестации, принять среднее арифметическое значение результатов измерений в этой точке, определяемое по формуле:

$$\bar{t}_j = \frac{1}{k} \sum_{i=1}^k t_{ij}, \quad \bar{\eta}_j = \frac{1}{k} \sum_{i=1}^k \eta_{ij},$$

где j – номер точки полезного объема камеры, в которой проводились измерения ($j = 1, \dots, n$); n – число точек измерений; k – число измерений; i – номер измерения ($i = 1, \dots, k$); t_{ij} –

значение температуры, определяемое в j -й точке при i -ом измерении; η_{ij} – значение относительной влажности, определяемое в j -й точке при i -ом измерении.

Дисперсию определить по формуле:

$$S_{t_j} = \frac{\sum_{i=1}^k (\bar{t}_j - t_i)^2}{k}, D_j = \frac{\sum_{i=1}^k (\bar{\eta}_j - \eta_i)^2}{k}.$$

Стандартное отклонение определить по формуле:

$$\delta_{t_j} = \sqrt{S_{t_j}}, \delta_j = \sqrt{D_j}.$$

За градиент температуры (относительной влажности) (неравномерность распределения) в полезном объеме камеры принять значение, рассчитанное по формуле:

$$t_{zp} = \bar{t}_{j_{\max}} - \bar{t}_{j_{\min}}, \eta_{zp} = \bar{\eta}_{j_{\max}} - \bar{\eta}_{j_{\min}},$$

где $\bar{t}_{j_{\max}}, \bar{t}_{j_{\min}}$ – наибольшее и наименьшее из средних значений температуры, вычисленные по пункту (а); $\bar{\eta}_{j_{\max}}, \bar{\eta}_{j_{\min}}$ – наибольшее и наименьшее из средних значений относительной влажности, вычисленные по пункту (а).

Остальные формулы для определения значений характеристик температуры и относительной влажности приведены, например, в [2].

Установка датчиков проводится в соответствии с требованиями ГОСТ Р 53618-2009. Пример схемы расположения датчиков температуры воздуха для камер с объемом до 2000 л представлен на рисунке 1.

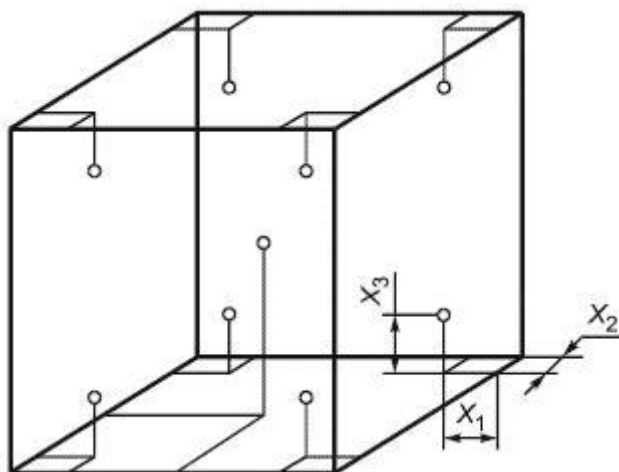


Рис. 1. Размещение датчиков температуры для камер объемом до 2000 л.

Определение вариантов использования программной системы. Так как основной целью разработанной программной системы является автоматизация проведения

статистического контроля качества работы климатической камеры по вышеприведенным формулам и прогнозирование ухудшения показателей, определяющих качество работы, то в качестве пользователя такой программной системы целесообразнее назначить одного сотрудника, отвечающего за проведение испытаний. Описание его поведения представлено на диаграмме вариантов использования (рис. 2).

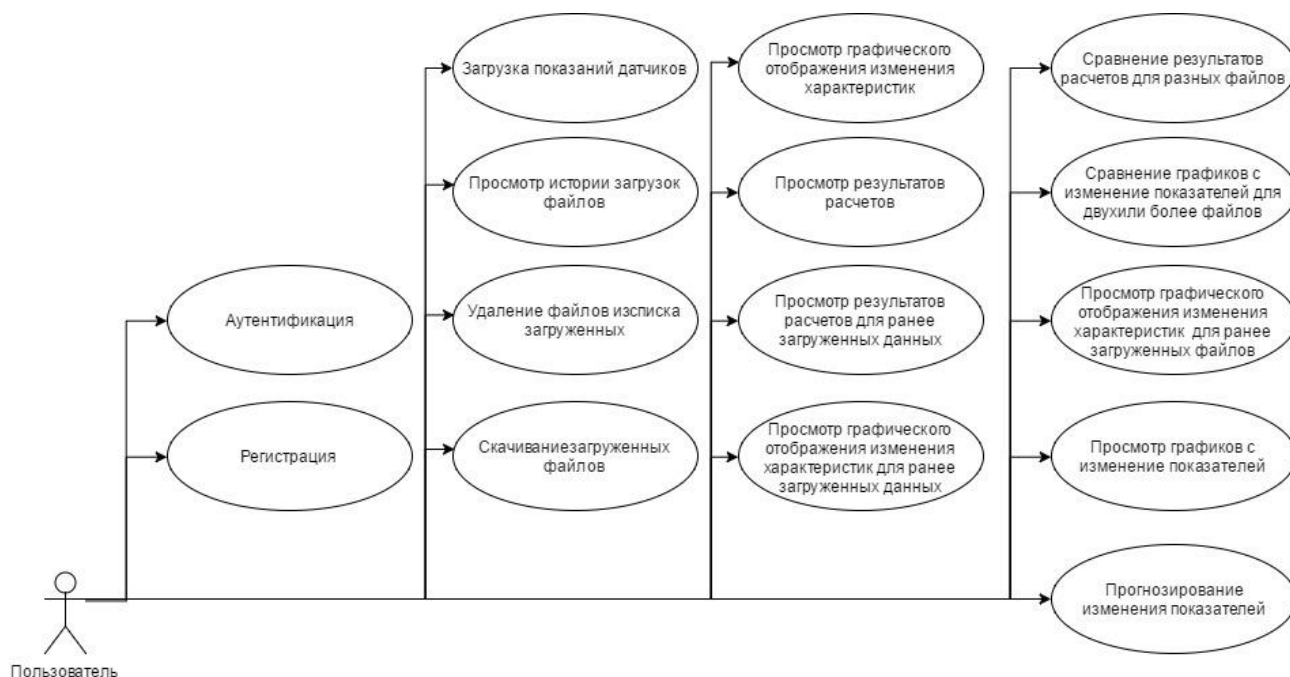


Рис. 2. Диаграмма вариантов использования.

Выбор платформы и стека технологий. Основываясь на приведенном выше наборе вариантов использования приложения, было принято решение реализовать приложение в форме веб-приложения, которое обладает следующими преимуществами по сравнению с desktop-платформой:

- кроссплатформенность – работоспособность приложения в наименьшей степени зависит от пользователя;
- гибкость – большой спектр технологий при разработке;
- мобильность – пользователю не нужно устанавливать приложение и доступ к приложению зависит только от доступа к Интернету;
- обновление приложения – исходный код приложения находится в одном месте, что позволит обновлять приложение лишь в одном месте и освободит пользователей от этой необходимости;
- хранение данных – хранение пользовательских файлов привязано к серверу.

В качестве стека технологий был выбран стек LEMP. Linux был выбран в качестве серверной операционной системы, Nginx – в качестве веб-сервера. MySQL был использован как СУБД, PHP – как скриптовый язык программирования.

Разработка клиентской части системы. Загружаемый файл, в котором по строкам будут находиться показания датчиков, должен иметь формат «.csv». Данный формат удобен тем, что его можно создать как текстовый файл или изменить расширение файла с «.xls» на необходимое расширение «.csv». Пример загружаемого файла с данными приведен на рисунке 3.

	A	B	C	D	E	F	G	H	I	J	K
1		Показание 1	Показание 2	Показание 3	Показание 4	Показание 5	Показание 6	Показание 7	Показание 8	Показание 9	Показание 10
2	Датчик 1	10	13	17	20	25	25	30	34	31	55
3	Датчик 2	11	12	16	21	27	24	31	36	32	54
4	Датчик 3	10	14	16	22	26	24	31	34	30	53
5	Датчик 4	11	13	16	21	25	23	30	35	31	56
6	Датчик 5	10	12	15	20	25	26	32	37	31	57
7	Датчик 6	19	12	17	20	27	27	32	37	30	55
8	Датчик 7	12	15	16	29	24	25	30	35	32	56
9	Датчик 8	10	13	17	21	26	26	31	36	32	56
10	Датчик 9	19	13	16	20	24	26	32	38	30	55

Рис. 3. Пример загружаемого файла с показаниями датчиков.

При запуске приложения открывается вкладка «Расчеты» (рис. 4), с которой можно произвести расчеты характеристик температуры, относительной влажности, скорости циркуляции воздуха, предварительно загрузив файл с данными датчиков.

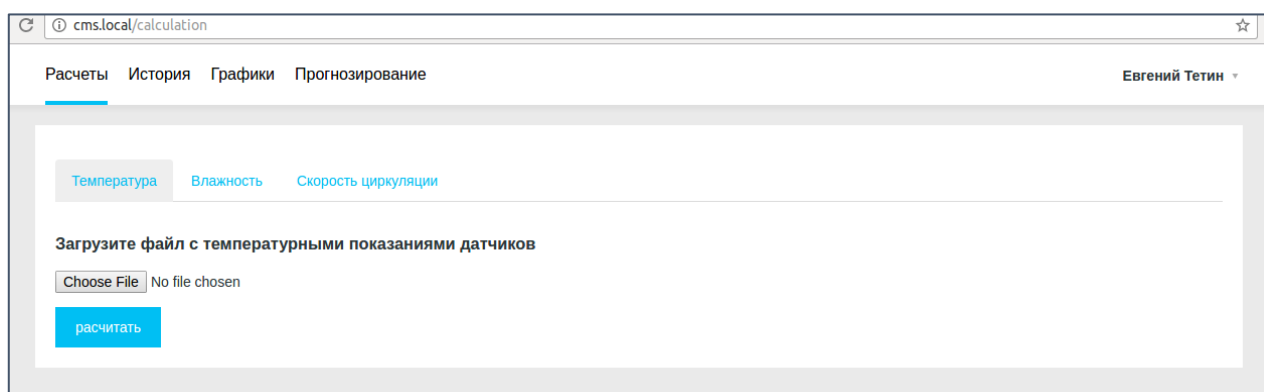


Рис. 4. Начальная страница приложения.

Выбранный файл загружается на сервер, где и происходит расчет, после чего на текущей странице становятся доступными вкладки «Результаты расчетов» (рис. 5) и «Графическое отображение» (рис. 6), при выборе которых можно просмотреть изменение параметров для датчиков в текстовом или графическом представлении.

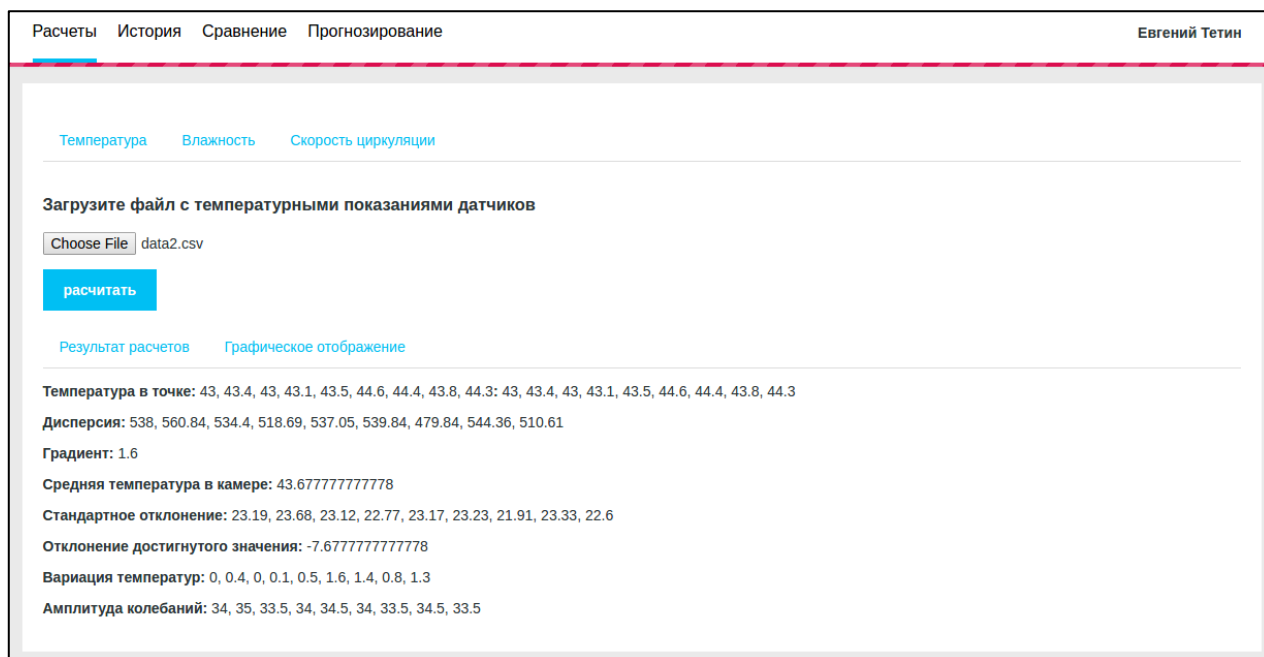


Рис. 5. Отображение результатов расчета характеристик.

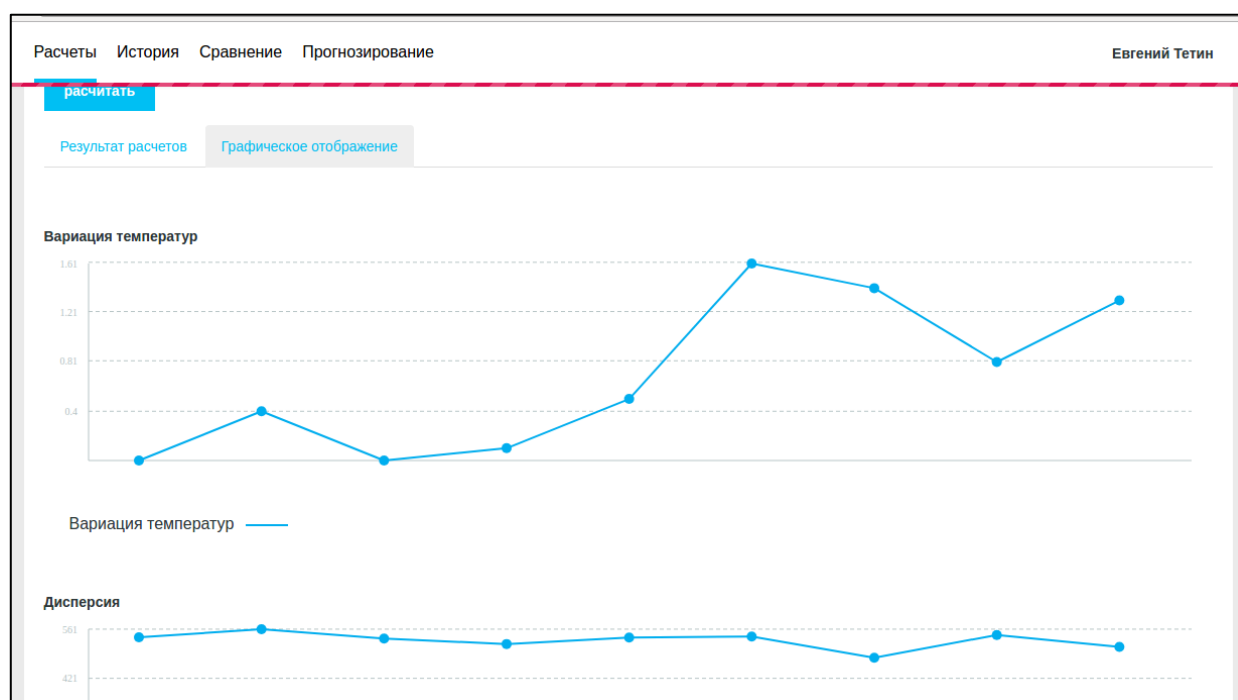


Рис. 6. Отображение результатов в графическом представлении.

На вкладке «История» можно просмотреть список загруженных ранее файлов с указанием их названия и даты загрузки, предоставляется возможность скачивания или удаления этих файлов.

Для сравнения рассчитанных характеристик из нескольких файлов на вкладке «Сравнение» предложено два способа: в форме таблицы (рис. 7) и в виде графиков (рис. 8).

Расчеты	История	Сравнение	Прогнозирование	Евгений Тетин
---------	---------	-----------	-----------------	---------------

Результат расчетов	Графическое отображение
--------------------	-------------------------

Характеристики	data1.csv 2017-06-18 10:01:07	data2.csv 2017-06-18 10:01:12
Температура в точке	26,26.4,26,26.1,26.5,27.6,27.4,26.8,27.3	43,43.4,43,43.1,43.5,44.6,44.4,43.8,44.3
Дисперсия	149,149.44,135.4,155.09,175.05,135.24,144.44,158.56,137.81	538,560.84,534.4,518.69,537.05,539.84,479.84,544.36,510.61
Градиент	1.6	1.6
Средняя температура в камере	26.6778	43.6778
Стандартное отклонение	12.21,12.22,11.64,12.45,13.23,11.63,12.02,12.59,11.74	23.19,23.68,23.12,22.77,23.17,23.23,21.91,23.33,22.6
Отклонение достигнутого значения	9.32222	-7.67778
Вариация температур	0,0.4,0,0.1,0.5,1.6,1.4,0.8,1.3	0,0.4,0,0.1,0.5,1.6,1.4,0.8,1.3
Амплитуда колебаний	22.5,21.5,21.5,22.5,23.5,21.5,22,23,21	34,35,33.5,34,34.5,34,33.5,34.5,33.5

Рис. 7. Табличная форма сравнения характеристик.

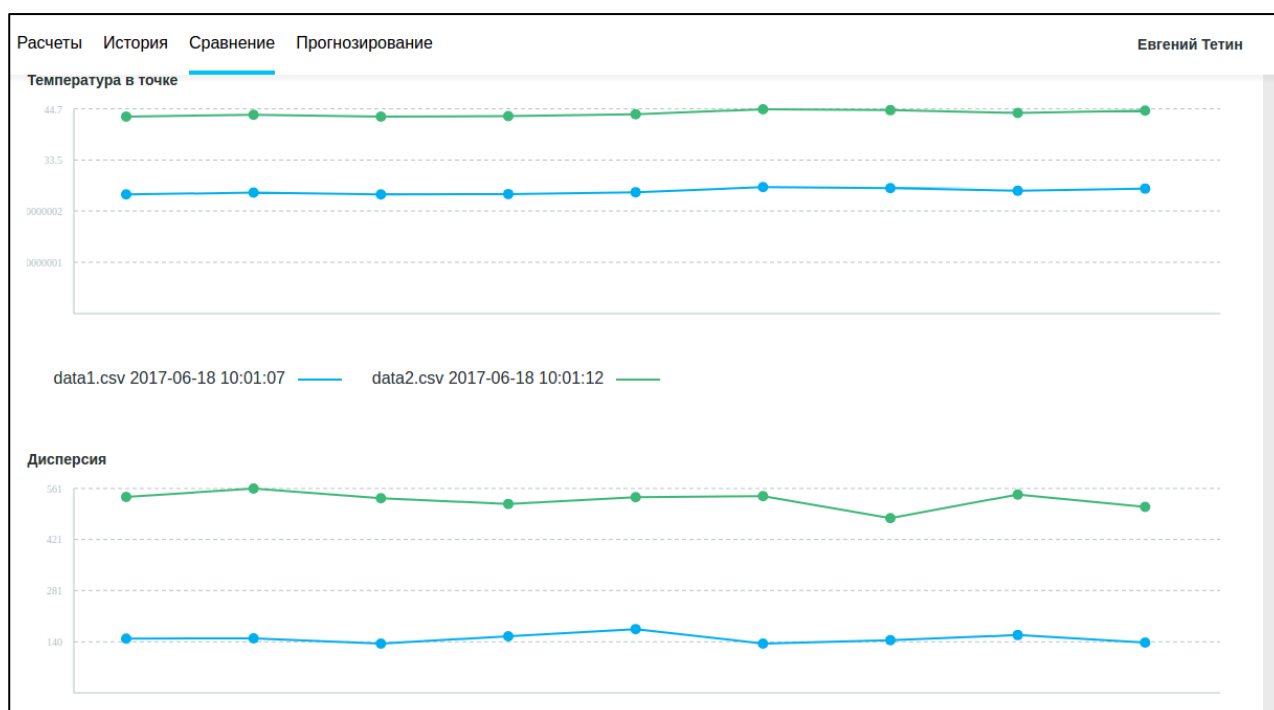


Рис. 8. Графики сравниваемых характеристик.

Для прогнозирования поведения характеристик, учитываемых при аттестации камеры, на вкладке «История» необходимо выбрать файлы, основываясь на которых будет сделан прогноз (рис. 9). При нажатии на кнопку «Перейти к выбору параметров прогнозирования» пользователь попадает на вкладку «Прогнозирование», где необходимо выбрать метод прогнозирования, нужную характеристику и ее критическое значение, при достижении которого камера считается не прошедшей аттестацию. Результатом прогнозирования является момент времени (дата), когда характеристика достигнет своего критического значения (рис. 10) [5].

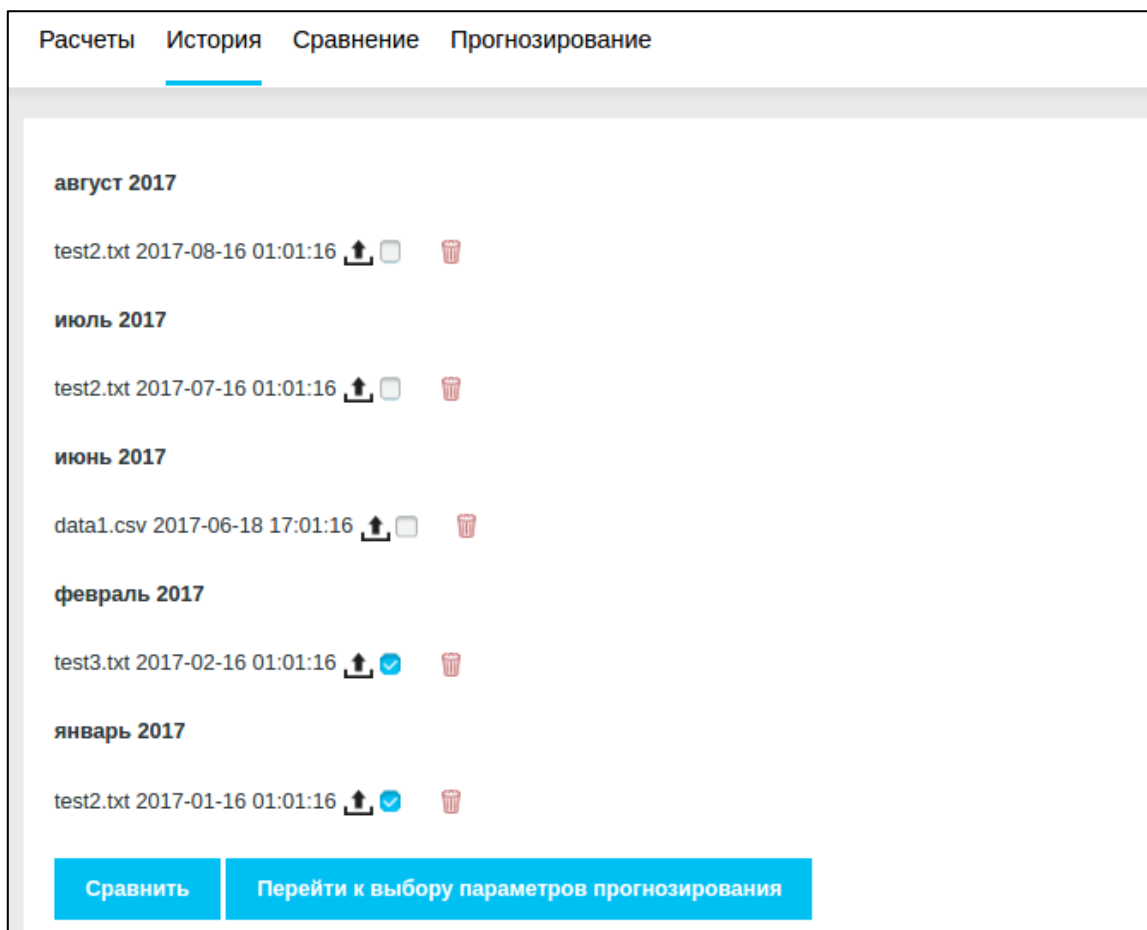


Рис. 9. Выбор файлов с данными для выполнения прогнозирования.

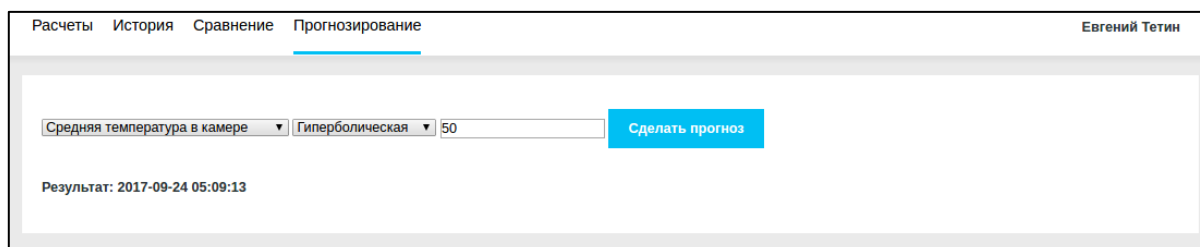


Рис. 10. Результат прогнозирования.

Заключение. Результатом работы является создание программной системы для автоматизации контроля качества работы климатической камеры и прогнозирование ее качественных характеристик для успешного прохождения аттестации. За счет универсальности платформы и выбранных технологий поддержка и расширение функционала системы не вызовет затруднений в дальнейшем, что делает данную систему актуальной и перспективной. Кроме того, достоинством разработанной системы является возможность ее доработки с целью использования для решения задач контроля качества работы не только климатической камеры, но и другого оборудования Центра проектирования инноваций АУ «Технопарк-Мордовия».

ЛИТЕРАТУРА

1. ГОСТ Р 53618-2009 [Электронный ресурс]. – Режим доступа: <http://docs.cntd.ru/document/1200080743>.
2. Чупракова А. М. Обеспечение качества при реализации внутрилабораторного контроля в исследовательском лабораторном центре // Вестник Южно-Уральского государственного университета. Серия: Экономика и менеджмент. – 2015. – Т. 9. – № 1. – С. 199–205.
3. ГОСТ Р 54082-2010 [Электронный ресурс]. – Режим доступа: <http://docs.cntd.ru/document/1200080743>.
4. Писарев В. Н. О стандартизации в области аттестации климатических камер // Труды международного симпозиума «Надежность и качество». – 2013. – Т. 1. – С. 91–93.
5. Лазаренко М. П. Математическое моделирование процессов теплообмена в климатических камерах // Международный научный журнал. – 2013. – № 6. – С. 48–52.

ТРИФОНОВ К. Е.

**ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММНО-ИНФОРМАЦИОННОЙ
СИСТЕМЫ АВТОМАТИЗАЦИИ РАБОТЫ АВТОСЕРВИСНОЙ СЛУЖБЫ**

Аннотация. В статье представлен анализ существующего программного обеспечения в области автоматизации работы предприятий по обслуживанию автомобилей. Автором приведены основные этапы моделирования, проектирования и реализации собственной программно-информационной системы, предназначенной для автоматизации работы автосервисной службы.

Ключевые слова: проектирование, моделирование, унифицированный язык моделирования, программно-информационная система, автоматизация автосервисной службы.

TRIFONOV K. E.

**DESIGNING AND IMPLEMENTING SOFTWARE
AND INFORMATION SYSTEM FOR AUTOMATION OF CAR SERVICE CENTER**

Abstract. The article provides an analysis of the software for automation of car service companies. The author describes the main stages of modeling, designing and implementation of the software and information system developed by him and designed to automate the work of a car service center.

Keywords: design, modeling, unified modeling language, software and information system, automation of car service center.

Введение. В настоящее время правительство России уделяет особое внимание развитию дорожно-транспортной инфраструктуры как обязательной меры для дальнейшего развития всех отраслей экономики и повышения комфорта жизни граждан [1]. Государственные программы льготного автокредитования «Первый автомобиль» и «Семейный автомобиль», расширяющееся производство современных отечественных и иностранных автомобилей, постепенное развитие дорожной сети при одновременном улучшении самих дорог – все это будет способствовать увеличению прибыли предприятий по обслуживанию автотранспорта.

Данная тенденция вызывает жесткую конкуренцию среди автосервисов в борьбе за клиента, который обращается в ремонтные службы для того, чтобы получить уважительное, быстрое и качественное обслуживание. С точки зрения автосервиса, рынок не просто расширяется, но также растет и спектр услуг, поэтому только высокая квалификация

персонала и применение современных средств управления и координации деятельности предприятия могут обеспечить устойчивое положение на рынке [1].

Объем необходимых информационных сведений, которые необходимо быстро и эффективно обработать для повышения эффективности управления транспортным предприятием, достиг таких размеров, что нельзя обойтись без специальных информационных систем. Во всем мире миллионы подобных предприятий уже закончили создание корпоративных сетей, подключили и используют технологии электронного документооборота, автоматизации складского и финансового учета [2].

Анализ программного обеспечения для автоматизации работы автосервисов.

Согласно сервису «Google карты», на территории города Саранска действует 119 предприятий по обслуживанию автомобилей и продаже автозапчастей, причем только 54 из них обладают веб-сайтами. Однако, стоит заметить, что подавляющее большинство такого рода сайтов выполняет лишь информационные задачи, не предоставляя возможности записи автомобиля на обслуживание или онлайн-покупки необходимой запчастей.

Чтобы выяснить причины такой относительно низкой информатизации проанализируем существующее программное обеспечение для предприятий автобизнеса.

Отраслевое решение «1С: Предприятие 8. Автосервис» фирмы «1С-Рарус» предназначено для автоматизации работы небольших предприятий автобизнеса, оказывающих услуги по ремонту и обслуживанию автомобилей: автосервисы, автомойки, станции технического обслуживания. Основные возможности данной программы представлены в [3]. Разработка «Control365» фирмы «Control365» является системой управления автомойкой, шиномонтажом или автосервисом, облегчающей контроль персонала и выполнение заказов путем автоматизации части производственных процессов [4]. Программный продукт «AllCarService» от производителя «ALLYOURPROG» может использоваться автосервисами, занимающимися ремонтом и обслуживанием транспортных средств любого масштаба. Интеграция станции технического обслуживания с данным программным продуктом позволяет значительно ускорить и упростить оформление заказов, упорядочить финансовую отчетность и прием денежных средств [5].

Проанализировав данные решения (табл. 1), можно сделать вывод о том, что они либо мало функциональны и не соответствуют современным требованиям, либо являются очень дорогими для приобретения небольшими предприятиями. Поэтому было принято решение создать программно-информационную систему «iMechanic», которая будет сочетать в себе всю необходимую функциональность с выгодной ценой, и позволит значительно сократить время на оказание услуг и расширить круг потенциальных потребителей, не потребовав для этого увеличения штата сотрудников.

Сравнение функциональности существующих решений

Функциональность	1С:Предприя- тие Автосервис	Control365	AllCarService	iMechanic
База клиентов	+	+	+	+
История заказов	+	+	+	+
Справочники услуг и материалов	+	+	+	+
Управление складскими запасами	+	-	+	+
Оформление документов	+	+	+	+
Формирование отчетов	+	+	+	+
Связь с клиентом	+	+	-	+
Удаленный доступ	+	+	-	+
Предварительная запись на ремонт	+	-	-	+
Быстрое внедрение	-	+	+	+
Интеграция с ПО бухгалтерии	+	+	-	-
Подключаемое торговое оборудование	+	+	+	-
Техническая поддержка и обновления	+	+	-	-
Цена	56 300 руб.	1000 руб./месяц	2900 руб.	3500 руб.

Проектирование программной системы (ПС). На начальном этапе проектирования необходимо определить отношения между основными типами пользователей и функциями, которые предоставляет ПС. Для этого строится диаграмма вариантов использования, описывающая разрабатываемую систему на концептуальном уровне.



Рис. 1. Диаграмма вариантов использования.

На рисунке 1 показано, что пользователями ПС являются клиент, менеджер и механик, причем к основным вариантам использования ПС для клиента относится возможность записи автомобиля на обслуживание и просмотр истории заказов; для менеджера – управление данными сотрудников и параметрами оказываемых услуг, формирование отчетов, заказ автозапчастей; для механика – выполнение действий, связанных с исполнением заказа: формирование различных актов, просмотр имеющихся на складе автозапчастей и истории обслуживания автомобилей.

На следующем этапе разработки ПС необходимо перейти от концептуального уровня системы к программному описанию определенных сущностей посредством классов, интерфейсов, структур. Для этой цели служит диаграмма классов, которая определяет типы классов системы и различного рода статические связи, которые существуют между ними.

На рисунке 2 показан пакет (совокупность взаимосвязанных классов), «Оформленные услуги», который необходим для организации хранения, обработки и управления услугами, заказанными клиентами. Класс «Work», являющийся центральным в данном пакете, имеет ассоциации с классами клиент («Client»), сотрудник («Staff»), автомобили («Cars») и услуга («Service»). Это позволяет быстро и удобно получать ответы на следующие вопросы: Какой автомобиль обслуживается? Сколько стоит услуга? Кто из сотрудников должен выполнить обслуживание? и т.д.

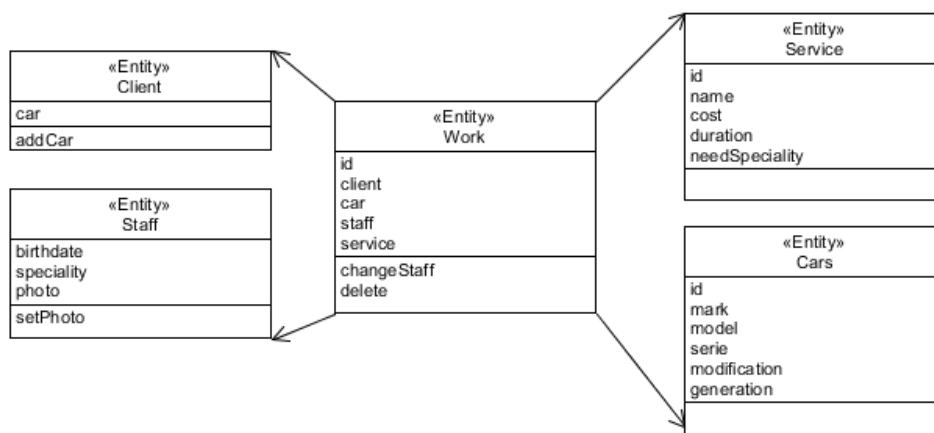


Рис. 2. Пакет «Оформленные услуги».

На завершающей стадии проектирования ПС выполняется разработка диаграммы компонентов. Диаграмма компонентов – статическая структурная диаграмма, показывающая разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами. В качестве физических компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и т. п. Разрабатываемая ПС имеет клиент-серверную архитектуру, поэтому было необходимо организовать межкомпонентное взаимодействие как

сервера с клиентом, так и программного обеспечения сервера с базовыми средствами операционной системы с помощью различных протоколов и типов связи. Запрос пользователя обрабатывается сервером и, в зависимости от его типа, могут быть задействованы самые разные компоненты. Например, компонент PHP Word выполняет формирование документов по заранее созданным шаблонам, а база данных и файловое хранилище предоставляют необходимые данные: изображения, документы, иная информация. Созданная структура взаимосвязи компонентов представлена на рисунке 3.

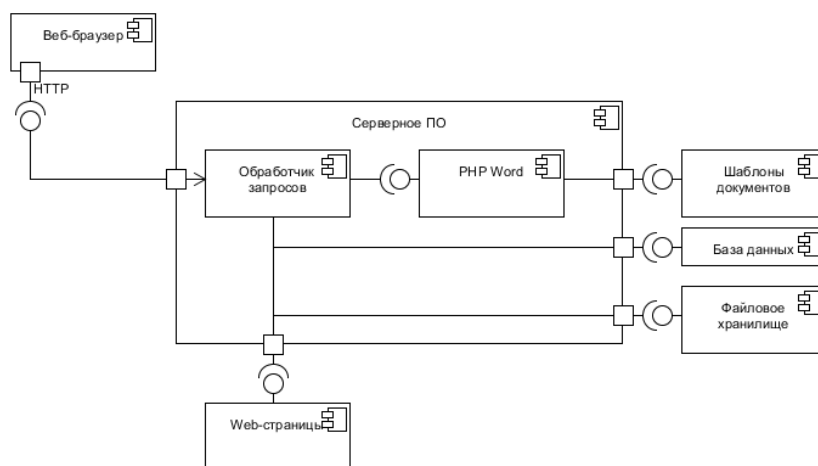


Рис. 3. Общая диаграмма компонентов.

Как было отмечено выше, ПС «iMechanic» имеет клиент-серверную архитектуру. Согласно этому, пользователь взаимодействует с сервером системы, на котором в отдельных исполнительных средах хранятся данные и производятся основные вычисления, посредством каналов связи сети Интернет. Эти особенности отражены на рисунке 4 в диаграмме развертывания.

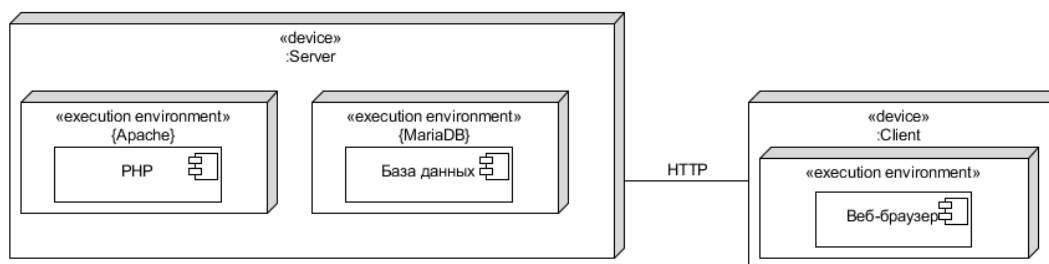


Рис. 4. Диаграмма развертывания.

Технологии реализации программной системы. Для реализации ПС использовались следующие технологии и языки программирования:

- для создания и оформления страниц сайта – HTML, JavaScript, CSS, библиотека jQuery;

- для работы с сессиями пользователей и взаимодействия с базами данных – язык программирования PHP, для генерации отчетов из заранее созданных шаблонов документов формата *.docx – библиотека PHPWord, для отправки email уведомлений пользователям – библиотека PHPMailer, которая предоставляет функциональность для отправки сообщений в формате HTML;
- в качестве системы управления базами данных – MariaDB, которая является ответвлением от СУБД MySQL, разрабатываемым Open Source сообществом под лицензией GNU General Public License;
- в качестве веб-сервера используется XAMPP – кроссплатформенная сборка, содержащая Apache, MariaDB, интерпретатор скриптов PHP, язык программирования Perl и большое количество дополнительных библиотек, позволяющих сильно сэкономить время разработки.

Обзор возможностей реализованной программно-информационной системы.

Рассмотрим основные возможности ПС «iMechanic».

Запись

Volkswagen Scirocco

Диагностика кузова

Асеев Александр

\$ 500 руб.

🕒 02:00 ч.

◀ Сегодня ▶ декабрь 25 — 31, 2017

Неделя Месяц agenda

Весь день	Пн 25	Вт 26	Ср 27	Чт 28	Пт 29	Сб 30	Вс 31
10:00	10:00–12:00 Свободно		10:00–12:00 Свободно	10:00–12:00 Свободно	10:00–12:00 Свободно	10:00–12:00 Свободно	10:00–12:00 Свободно
11:00		11:00–12:00 Занято					
12:00	12:00–14:00 Свободно	12:00–13:00 Занято	12:00–14:00 Свободно	12:00–14:00 Свободно	12:00–14:00 Свободно	12:00–14:00 Свободно	12:00–14:00 Свободно
13:00							
14:00	14:00–16:00 Свободно	14:00–16:00 Свободно	14:00–16:00 Свободно	14:00–16:00 Свободно	14:00–16:00 Свободно	14:00–16:00 Свободно	14:00–16:00 Свободно
15:00							
16:00	16:00–18:00 Свободно	16:00–18:00 Свободно	16:00–18:00 Свободно	16:00–18:00 Свободно	16:00–18:00 Свободно	16:00–17:00 Занято	16:00–18:00 Свободно
17:00							

Рис. 5. Раздел «Запись автомобиля на обслуживание».

Одной из главных функций клиента автосервиса является возможность удаленной записи своего автомобиля на обслуживание (рис. 5), при этом необходимо выбрать автомобиль, который нуждается в обслуживании, услугу, специалиста и удобное время, после чего подтвердить запись. Список специалистов и возможное время записи формируется автоматически в зависимости от их свободного времени и типа услуги.

Еще одной важной функцией для клиента является предоставление истории обслуживания его автомобилей, причем возможен просмотр как запланированных ремонтов, так и завершенных, что позволяет оценить финансовые затраты и выполнить экспорт данных в различные форматы: txt, xls и т.д.

История обслуживания

Volkswagen Scirocco

Предстоящие
Находится на обслуживании
Предстоящие
Завершенные

	Название	Цена	Дата	Сотрудник	Действия
+	Диагностика кузова	500.00	2018-03-12 10:00:00	Асяев Александр	
+	Замена тормозных колодок	1000.00	2018-03-12 12:00:00	Асяев Александр	
+	Диагностика подвески	1000.00	2018-03-12 10:00:00	Демидкин Сергей	
	Итого	2500			

Рис. 6. Раздел «История обслуживания».

Для менеджера ПС «iMechanic» предоставляет следующие возможности: управление списком сотрудников (рис. 7), просмотр списка клиентов, формирование финансовых отчетов, изменение официальных реквизитов автосервиса, заказ автозапчастей, управление складскими запасами, отправление напоминаний пользователям об обслуживании, управление списком поставщиков, управление параметрами оказываемых услуг.

Персонал

+ Добавить

Search

↺
📄
☰
🖨

Сотрудник	Специальность	Дата рождения	Действия
+ Асяев Александр	Автомеханик широкого профиля	1967-07-26	✕
+ Демидкин Сергей	Автомеханик широкого профиля	1974-09-18	✕
+ Иван Иванов		1969-07-15	✕
+ Козюков Юрий		1990-10-18	✕
+ Лунькин Иван		1967-07-18	✕
+ Миронов Дмитрий		1989-07-06	✕
+ Синичкин Виктор		1968-02-27	✕
+ Трифонов Кирилл		1996-11-12	✕

Новый сотрудник

Полное имя

Логин

Пароль

Email

Телефон

ДД . ММ . ГГГГ

Специальность

Photo Browse...

Зарегистрировать

Showing 1 to 8 of 8 rows

Рис. 7. Раздел «Персонал».

В финансовом разделе системы предоставлена информация об оказанных услугах за выбранный период, что позволяет руководству следить за финансовой эффективностью каждого из сотрудников и автосервиса в целом (рис. 8). Также данный раздел предоставляет возможность экспортировать информацию в формат xls, что позволит организовать цифровой или бумажный документооборот.

Финансы

ДД . ММ . ГГГГ

ДД . ММ . ГГГГ

Сотрудник

Search

↺
📄
☰
🖨

Сотрудник	Усл	Детали	Всего
+ Асяев Александр	Зам	10050.00	14050.00
+ Демидкин Сергей	Рег	5800.00	9800.00
+ Асяев Александр	Диа	5650.00	7650.00
+ Асяев Александр	Замена тормозных колодок	1000.00	5000.00
+ Асяев Александр	Замена тормозных колодок	1000.00	5000.00
+ Асяев Александр	Замена тормозных колодок	1000.00	5000.00
+ Лунькин Иван	Диагностика двигателя	2000.00	-
+ Демидкин Сергей	Регулировка подвески	2000.00	-
+ Козюков Юрий	Диагностика электроники	1200.00	-
Итого		18200	36500

Showing 1 to 10 of 11 rows 10 rows per page

1
2

Рис. 8. Раздел «Финансы».

Механик при использовании ПС «iMechanic» может просматривать информацию о своих клиентах и их автомобилях, свои финансовые показатели и содержимое склада автосервиса.

Выполнение заказа № 6

Название	Диагностика подвески
Клиент	Белозёров Иван Петрович
Автомобиль	Асига NSX
Начало	2017-12-30 10:00:00
Конец	2017-12-30 12:00:00
Цена	1000.00

+ Добавить

✕ Удалить

Search

☐	ID	Наименование	Цена	Количество
☐	13	Стабилизатор поперечной устойчивости	3650.00	1
☐	14	Кулаки поворотные и ступицы	2000.00	1

Showing 1 to 2 of 2 rows

Комментарий клиента

Комментарий специалиста

Заказ выполнен

Рис. 9. Раздел «Выполнение заказа».

Но наиболее важной функцией данного типа пользователя является возможность управления заказами. Автомеханик может просматривать подробную информацию о предстоящих, текущих или уже завершённых заказах: время оказания, цена, используемые запчасти, ФИО клиента (рис. 9).

По мере фактического выполнения заказа механик обязан изменять его статус. При этом система будет автоматически формировать необходимые документы, такие как «Акт приёмки автомобиля» и «Акт оказания услуг». Также механику предоставляется возможность указывать информацию об использованных при обслуживании автозапчастях. При добавлении автозапчастей, которых в данный момент нет на складе, системой будет предложено сформировать заказ на их покупку, которая будет рассмотрена менеджером.

Заключение. Сравнение функциональности существующих программных продуктов и ПС «iMechanic» было представлено выше. Очевидно, что созданная программная система не уступает по функциональности рассмотренным аналогам и, кроме того, сочетает в себе

высокую производительность, простоту использования, удобный и лаконичный интерфейс. Эти особенности позволят повысить эффективность работы автосервисного предприятия при минимальных затратах. Для повышения конкурентоспособности ПС «iMechanic» необходимо реализовать ее интеграцию с профессиональными бухгалтерскими решениями и торговым оборудованием, что позволит избежать ручной работы по переносу данных из одной системы в другую.

ЛИТЕРАТУРА

1. Усуфов М. М., Маковецкая-Абрамова О. В. Современные особенности развития автосервиса // Техничко-технологические проблемы сервиса. – 2012. – № 2 (20). – С. 55–60.
2. Тарабукина Н. А. Применение информационных технологий на автотранспортных предприятиях // Проблемы развития транспортной инфраструктуры Европейского Севера России: Материалы межрегиональной научно-практической конференции. – Вып. 2. – Котлас: СПГУВК, 2005. – С. 25–31.
3. Основные возможности «1С: Предприятие 8.Автосервис» [Электронный ресурс]. – Режим доступа: <https://rarus.ru/1c-auto/1c8-avtoservis>.
4. Программа для автоматизации автомойки, шиномонтажа и автосервиса (СТО) Control365 [Электронный ресурс]. – Режим доступа: <http://control365.ru>.
5. Страница продукта AllCarService [Электронный ресурс]. – Режим доступа: <http://www.allyourprog.far.ru/index.php?show=page&idd=1446893400>.

ЦАРЕГОРОДЦЕВ А. В., ЛОГИНОВА А. О., БЛОХИНА О. В.
МЕТОДИКА КОЛИЧЕСТВЕННОЙ ОЦЕНКИ РИСКА ИНФОРМАЦИОННОЙ
БЕЗОПАСНОСТИ ДЛЯ ОБЛАЧНОЙ ИНФРАСТРУКТУРЫ ОРГАНИЗАЦИИ

Аннотация. В связи с тем, что облачные вычисления несут новые вызовы в области информационной безопасности, для организации крайне важно контролировать процесс управления информационными рисками в облачной среде. В статье предложен подход к оценке рисков, используемый при выборе наиболее приемлемого варианта конфигурации среды облачных вычислений с точки зрения требований безопасности.

Ключевые слова: облачные вычисления, угрозы информационной безопасности, анализ информационных рисков, методы управления информационной безопасностью, требования информационной безопасности.

TSAREGORODTSEV A. V., LOGINOVA A. O., BLOKHINA O. V.
DATA SECURITY RISK ASSESSMENT
FOR CLOUD INFRASTRUCTURE OF ORGANIZATION

Abstract. Due to the fact that cloud computing brings about new challenges for information security, it is imperative for organizations to control the process of information risk management in a cloud. This paper proposes a risk assessment approach used for the selection of the most appropriate configuration options of cloud computing environment from the point of view of security requirements.

Keywords: cloud computing, information security threats, information risk analysis, information security management tools, information security requirements.

Одной из серьезных угроз информационной безопасности (ИБ) облачных сред является использование со стороны злоумышленников известных, но не исправленных уязвимостей [4]. Успешная реализация эксплойта потенциально может привести к значительному финансовому ущербу для клиента, потере репутации облачного провайдера и компрометации используемых механизмов защиты [2].

Принимая во внимание тот факт, что поддержка конфигурации облачной среды, управление уязвимостями и обновлениями зависит от модели предоставления облачного сервиса, ответственность за своевременное выполнение этих задач частично или полностью лежит на стороне облачного провайдера [7].

В этом случае одной из основных задач обеспечения ИБ информационных систем (ИС) становится рассмотрение и классификация уязвимостей облачных сред и возможность использования этой информации при проведении количественной оценки риска.

Предлагаемая методика позволит принять решение при выборе систем защиты информации, программного обеспечения для компонентов ИС, функционирующей на основе технологии облачных вычислений.

Основные этапы методики количественной оценки риска.

Для возможности проведения количественной оценки и построения риск-модели облачной среды необходимо решить следующие задачи.

1. Определить и описать возможные технические риски использования облачных сред в их взаимосвязи с уязвимостями и активами организации.
2. Сформировать перечень уязвимостей для каждого риска, построить для них базовые векторы системы общего учета уязвимостей (CVSS).
3. Разработать методику по оценке уровня риска. Показатели частоты и урона будут рассчитываться на основе показателей CVSS метрик: базовой, временной и инфраструктурной.
4. Определить риск-модель на основе полученных уровней влияния рассматриваемых уязвимостей. Группировка уязвимостей по принципу принадлежности одному уровню влияния позволит ввести новый показатель – сервисный уровень. Совокупное представление возможных сервисных уровней и интенсивности переходов между ними позволит прогнозировать уровни риска в определенный момент времени.

Представим методику оценки рисков в виде следующих связанных процессов/действий: идентификация контекста оценки риска, идентификация риска, анализ риска, оценивание риска, обработка риска (рис. 1).

На верхнем уровне предлагаемая методика по оценке рисков включает два основных этапа. Первый этап описывает управляемый риском анализ, который включает оценивание набора злонамеренных использований и связанных с ними уровней риска, которые являются результатом шагов 2, 3, 4 описанной методики с последующим сравнением полученных значений с критериями принятия риска, которые определены на шаге 1. Результатом этой фазы является набор рисков, требующих обработки [9; 10].

Набор рисков для обработки, набор альтернативных решений и других компромиссных параметров, соответствующих разработке, проекту и финансовому состоянию являются входными данными для второго этапа методики.



Рис. 1. Методика оценки риска.

Действия, описанные в рамках первого этапа, включают ключевые элементы анализа: набор угроз, уязвимости; злонамеренное использование, его частота и влияние; риск ИБ, критерии принятия риска. При этом риск рассчитывается для каждого злонамеренного использования путем комбинации его частоты с одним из влияний. Это означает, что злонамеренное использование приводит к появлению одного или нескольких рисков ИБ, зависящих от количества связанных влияний [8].

Частота злонамеренного использования и его влияние могут быть представлены в виде количественных показателей: определенное количество проявлений в течение временного интервала или вероятность появления злонамеренного использования в

определенный период времени. Влияние может быть представлено в виде финансовых потерь, потери репутации и т. д. [1; 3]. Статистические ожидаемые потери =

$$= (I_1 \times F_1) \times L_1 + (I_i \times F_i) \times L_i + \dots + (I_n \times F_n) \times L_n \quad (1)$$

Используя полученные выражения как базу для оценки, рассмотрим основные положения общей системы учета уязвимостей (CVSS), предоставим их характеристику и интерпретацию показателей применительно к среде облачных вычислений.

Основные положения общей системы учета уязвимостей. Общая система учета уязвимостей (CVSS) [6] в настоящее время достаточно широко применяется и все больше принимает вид стандарта для определения и оценки уязвимостей. Основная задача системы состоит в оценке уровня серьезности уязвимостей и предоставлении рекомендаций по смягчению последствий проявления угроз.

Показатели *базовой группы* описывают характеристики уязвимости, которые являются постоянными и не зависят ни от времени, ни от инфраструктуры. Основными показателями данной группы являются:

а) вектор доступа. Этот показатель отражает то, каким доступом должен обладать злоумышленник для эксплуатации уязвимости. Значения показателя и описание приведены в таблице 1;

б) сложность доступа. Этот показатель описывает сложность атаки, необходимой для эксплуатации уязвимости. Чем ниже уровень сложности, тем выше показатель уязвимости;

в) аутентификация. Показатель «Аутентификация» описывает количество необходимых сеансов аутентификации цели при эксплуатации уязвимости. Показатель не учитывает сложности этого процесса, а лишь характеризует саму необходимость аутентификации для использования уязвимости. Аутентификация происходит только в том случае, если доступ к ресурсу уже получен.

Расчет базовой метрики происходит следующим образом:

$$BaseScore =$$

$$= round_to_1_decimal\{[(0,6 * Impact) + (0,4 * Exploitability) - 1,5] + f(Impact)\}, \quad (3)$$

где *BaseScore* – базовая метрика, *Impact* – общее влияние (урон), *Exploitability* – доступность использования эксплойта;

$$Impact = 10,41 * (1 - (1 - ConfImpact) * (1 - IntegImpact) * (1 - AvailImpact)), \quad (4)$$

где *ConfImpact* – урон конфиденциальности, *IntegImpact* – урон целостности, *AvailImpact* – урон доступности;

$$Exploitability = 20 * AccessVector * AccessComplexity * Authentication, \quad (5)$$

где *AccessVector* – вектор доступа, *AccessComplexity* – вектор сложности, *Authentication* – аутентификация; $f(impact) = 0$, если $Impact = 0$, в других случаях $f = 1,176$.

Временная метрика описывает показатели угрозы, реализующей рассматриваемую уязвимость. Метрика включает 3 показателя. Приведем краткое описание каждого из них.

А) Доступность кода и техники эксплойта. Этот показатель характеризует доступность и технику (код) эксплойта. Доступность в общедоступном доступе рабочего эксплойта (открытого кода) резко повышает количество потенциальных злоумышленников.

Б) Степень готовности решения для ликвидации последствий уязвимости. В общем случае уязвимость после ее появления в течение определенного времени не имеет исправлений в виде патча или официального обновления.

В) Степень достоверности информации об уязвимости. Этот показатель отражает степень достоверности источников о существования самой уязвимости, а также возможность использования технических деталей эксплойта.

Расчет временной метрики включает веса временных показателей с комбинацией с базовой оценкой, при этом результат находится в диапазоне от 0 до 10. Итоговая оценка временной метрики не превышает базовую оценку, но должна быть не меньше 33% от нее.

$$TemporalScore = round_to_1_decimal (BaseScore * Exploitability * RemediationLevel * ReportConfidence), \quad (6)$$

где *TemporalScore* – временная метрика, *BaseScore* – базовая метрика, *Exploitability* – доступность кода и техники эксплойта *RemediationLevel* – степень готовности решения, *ReportConfidence* – достоверность информации.

Метрики среды эксплуатации (инфраструктуры).

А) Сопутствующий потенциальный ущерб. Показатель описывает возможные потери (финансовые) в результате успешной эксплуатации уязвимости.

Б) Распределение целевых систем. Показатель описывает, какая часть компонентов облачной ИТКС подвержена уязвимости.

В) Требования к безопасности. Показатель позволяет специалисту ИБ определить приоритет и важность ключевых требований безопасности: конфиденциальность, целостность, доступность. Общий эффект инфраструктурного показателя зависит от соответствующих значений частных показателей из базовой метрики – урон для конфиденциальности, целостности и доступности. Расчет инфраструктурной метрики происходит следующим образом:

$$EnvironmentalScore = round_to_1_decimal((AdjustedTemporal + (10 - AdjustedTemporal) * CollateralDamagePotential) * TargetDistribution), \quad (7)$$

где *EnvironmentalScore* – инфраструктурная метрика, *TargetDistribution* – распределение целевых систем, *CollateralDamagePotential* – сопутствующий потенциальный ущерб, *AdjustedTemporal* – скорректированная оценка временной метрики, пересчитанная с учетом требований безопасности и урона из базовой метрики (*AdjustedImpact*).

$$AdjustedImpact = \min(10, 10, 41 * (1 - (1 - ConfImpact * ConfReq) * (1 - IntegImpact * IntegReq) * (1 - AvailImpact * AvailReq))), \quad (8)$$

где *ConfReq*, *IntegReq*, *AvailReq* – требования к конфиденциальности, целостности, доступности.

Таким образом, получается базовый, временной и инфраструктурный векторы, которые приведены в таблице 1.

Таблица 1

Показатели требований к безопасности

Группа метрик	Вектор
Базовая	AV:[L,A,N]/AC:[H,M,L]/Au:[M,S,N]/C:[N,P,C]/I:[N,P,C]/A:[N,P,C]
Временная	E:[U,POC,F,H,ND]/RL:[OF,TF,W,U,ND]/RC:[UC,UR,C,ND]
Инфраструктурная	CDP:[N,L,LM,MH,H,ND]/TD:[N,L,M,H,ND]/CR:[L,M,H,ND]/ IR:[L,M,H,ND]/AR:[L,M,H,ND]

Используя базовый, временной и инфраструктурный векторы определяются два основных интегральных показателя, влияющих на оценку риска. Чем выше уровень подверженности уязвимости применению эксплойта, тем больше шансов у злоумышленника провести успешную атаку и тем больше показатель частоты злонамеренного использования (*F*). Потенциальное влияние определяется как урон (*влияние*), зависящий от показателей уязвимости в базовой метрике и, в то же время, может быть увеличено или уменьшено в зависимости от требований к конфиденциальности, доступности и целостности, определенных в инфраструктурной метрике.

Анализ возможных угроз и анализ рисков служит основой для обоснования выбора мер по обеспечению ИБИС облачных вычислений, которые должны быть осуществлены для снижения риска до приемлемого уровня.

На базе общей системы оценки уязвимостей (CVSS), позволяющей определить качественный показатель подверженности уязвимостям с учетом факторов окружающей среды, была разработана методика количественной оценки потенциальных уязвимостей для различных типов развертывания облачных сред.

Предложенный подход к анализу и управлению рисками позволяет провести оценку защищенности облачной среды, функционирующей в условиях воздействия рассматриваемого класса угроз, а также эффективности комплекса мер и средств противодействия этим угрозам. На основе полученной оценки появляется возможность сделать выбор между различными вариантами конфигурации среды облачных вычислений и выбрать наиболее приемлемый вариант с точки зрения требований безопасности.

ЛИТЕРАТУРА

1. Accorsi R., Wonnemann C. Auditing Workflow Execution against Dataflow Policies. – In Proc. BIS, 2010. – pp. 207–217.
2. Bell D. E., LaPadula L. J. Secure Computer System: Unified Exposition and Multics Interpretation. – Tech report ESD-TR-75-306, Mitre Corp, Bedford, Ma, 1976.
3. Bishop M. Computer Security: Art and Science. – Boston: Addison-Wesley Professional. 2002. – 1084 p.
4. Вдовин И. COBIT 4.1. – М.: Аудит и контроль информационных систем, 2008. – 240 с.
5. Mell P., Tim Grance T. The NIST Definition of Cloud Computing [Электронный ресурс]. – Режим доступа: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>.
6. NVD Common Vulnerability Scoring System Support v.2. [Электронный ресурс]. – Режим доступа: <http://nvd.nist.gov/cvss.cfm?calculator&version=2>.
7. Trope R. L. [and other] A Coherent Strategy for Data Security through Data Governance // Data Governance. – 2007. – Vol. 5. – No. 3. – pp. 32–39.
8. Волков С. Д., Царегородцев А. В., Цацкина Е. П. Особенности построения систем обнаружения компьютерных атак для информационно-телекоммуникационных систем, функционирующих на основе технологии облачных вычислений [Электронный ресурс] // Огарев-online. – 2017. – №13. – Режим доступа: <http://journal.mrsu.ru/arts/osobennosti-postroeniya-sistem-obnaruzheniya-kompyuternyx-atak-dlya-informacionno-telekommunikacionnyx-sistem-funkcioniruyushhix-na-osnove-texnologii-oblachnyx-vychislenij>.
9. Царегородцев А. В., Качко А. К. Обеспечение информационной безопасности на облачной архитектуре организации // Национальная безопасность. – 2011. – № 5. – С. 25–34.
10. Царегородцев А. В., Качко А. К. Один из подходов к управлению информационной безопасностью при разработке информационной инфраструктуры организации // Национальная безопасность. – 2012. – № 1(18). – С. 46–59.

ЗАРОЧЕНЦЕВ Н. С.

**РАЗРАБОТКА АРХИТЕКТУРЫ «INTERNET OF ENERGY»
НА БАЗЕ НАКОПИТЕЛЕЙ ЭЛЕКТРИЧЕСКОЙ ЭНЕРГИИ**

Аннотация. В статье представлена разработка архитектуры программного обеспечения энергоинформационной системы «Internet of Energy» на базе накопителей электрической энергии. Дано описание динамической сетевой инфраструктуры, которая связывает энергетическую сеть с сетью Интернет и обеспечивает эффективное использование энергетических ресурсов. Проектирование архитектуры велось при помощи методологии RUP.

Ключевые слова: интеллектуальная энергосистема, архитектура программного обеспечения, накопитель электрической энергии, методология RUP.

ZAROCHENTSEV N. S.

**DESIGNING INTERNET OF ENERGY ARCHITECTURE
BASED ON ELECTRIC ENERGY ACCUMULATOR**

Abstract. The article presents the designing of the software architecture of the energy-information system "Internet of Energy" based on electric energy accumulator. The author gives a description of the dynamic network infrastructure that connects the power network to the Internet and ensures the efficient use of energy resources. The architecture was designed using the rational unified process.

Keywords: smart grid, software architecture, electric energy accumulator, rational unified process.

Введение. В настоящее время во всем мире наблюдается повышение спроса на электроэнергию. Потребители ожидают от поставщиков снижения тарифов на электроэнергию с увеличением показателей надежности услуг, прозрачности схем поставок и наличия возможности выбора поставщика в условиях конкурентного рынка. Разработка систем, удовлетворяющих данным требованиям, соответствует одному из основных направлений энергетической стратегии России до 2030 года [1], стимулирующему создание высоко интегрированных интеллектуальных системообразующих и распределительных электрических сетей нового поколения в Единой энергетической системе России и концепции построения интеллектуальной электроэнергетической системы с активно-адаптивной сетью.

В статье разрабатывается архитектура программного обеспечения энергоинформационной системы, которая призвана решить проблему дефицита

энергоресурсов и электроэнергии с помощью повышения энергоэффективности и энергосбережения. Планируется, что энергия в новой системе будет передаваться только в случае запроса со стороны пользователя, тем самым энергия тратится только при необходимости. Также архитектура системы предусматривает использование распределенных источников энергии, с помощью чего возможно снижение угрозы благополучию окружающей среды вследствие техногенного воздействия объектов энергетики.

1. Определение требований к системе. Требования к системе делятся на функциональные и специальные.

Функциональные требования представляют собой описание вариантов использования. Пользователь при использовании системы должен иметь возможность:

- зарегистрироваться и авторизоваться в системе;
- сформировать запрос на предоставление энергопакетов;
- уведомить систему о наличии избытка электроэнергии;
- при помощи специальных команд управлять работой накопителя электроэнергии и подключенных к нему потребителей и генераторов электроэнергии;
- получить информацию о состоянии накопителя и альтернативных источников энергии;
- получить отчет о работе системы;

Администратору сети должны быть предоставлены следующие действия:

- получить информацию о состоянии накопителя и альтернативных источников энергии;
- получить отчет о работе системы;
- получить информацию о состоянии системы.

К специальным требованиям относятся следующие требования:

- программно-технические средства компонентов системы должны соответствовать стандартам интерфейса Ethernet с использованием протокола TCP/IP;
- система должна функционировать непрерывно, за исключением периодов проведения профилактических и других работ, предусмотренных регламентом, а также устранения возникших нештатных ситуаций;
- при возникновении сбоев в аппаратном обеспечении, включая аварийное отключение электропитания, энергоинформационная система должна автоматически восстанавливать свою работоспособность после устранения сбоев и корректного перезапуска аппаратного

обеспечения (за исключением случаев повреждения рабочих носителей информации с исполняемым программным кодом);

– в системе также должны быть предусмотрены средства контроля корректности входных данных и оповещения пользователя в случае ошибочных действий.

2. Разработка архитектурной модели системы. Разработанная архитектурная модель представляет собой иерархичную систему, состоящую из четырех уровней (подсистем): Pico Grid, Nano Grid, Micro Grid и Smart Grid.

На нижнем уровне системы находится подсистема Pico Grid, которая представляет собой сложносоставное конечное устройство потребления, накопления и генерации электроэнергии. На базе данной подсистемы должны быть построены все электроустройства, находящиеся в управлении у пользователя. Устройства потребления электроэнергии будут отличаться от традиционных тем, что они будут способны оповещать о необходимом для работы количестве электроэнергии, тем самым исключая передачу излишней энергии.

На следующем уровне иерархии находится подсистема Nano Grid. Данная система включает в себя энергокоммутатор – устройство, которое координирует работу систем Pico Grid. Именно через это устройство пользователь будет взаимодействовать с рынком электроэнергии. Энергокоммутатор позволит направлять электроэнергию на электроустройства, принимать электроэнергию от генераторов электроэнергии и направлять ее в накопитель электроэнергии.

Подсистемы Nano Grid объединяются в систему Micro Grid. Данная система является минимальной единицей энергетической самодостаточной сети, то есть даже при отключении центрального источника энергии система продолжит функционировать. Для синхронизации входящих в систему устройств используется энергорутер. Все операции по передаче электроэнергии между энергокоммутаторами осуществляются через него. Именно энергорутер обеспечивает двунаправленную передачу энергии. Каждый энергорутер охватывает район города, поселок или торговый центр и при запросе с контролируемого энергокоммутатора выбирает наиболее выгодный источник энергии. Также к каждому энергорутеру подключен накопитель энергии, с помощью которого производится оперативное сглаживание пиков потребления внутри системы.

Верхним уровнем иерархии является система Smart Grid. Эта система контролирует энергорутеры целого города. Также на этом уровне осуществляется интеграция традиционных сетей и Micro Grid. Также цифровая подстанция координирует передачу электроэнергии между энергорутерами.

Контроль за всеми операциями по передаче электроэнергии производится с помощью серверов. К каждому энергокоммутатору, энергорутеру и цифровой подстанции

подключено по одному серверу. Кроме того, сервер имеет связь со всеми серверами, находящимися на уровень ниже и с сервером, объединяющим сервера его уровня. С помощью этого становится возможен контроль энергетических и финансовых транзакций.

Пример реализации вышеописанной архитектуры приведен на рисунке 1.

Стоит отметить, что на каждом уровне для объединения устройств может применяться не только топология «Звезда». В каждом случае топология должна определяться в соответствии с конкретными условиями реализации.

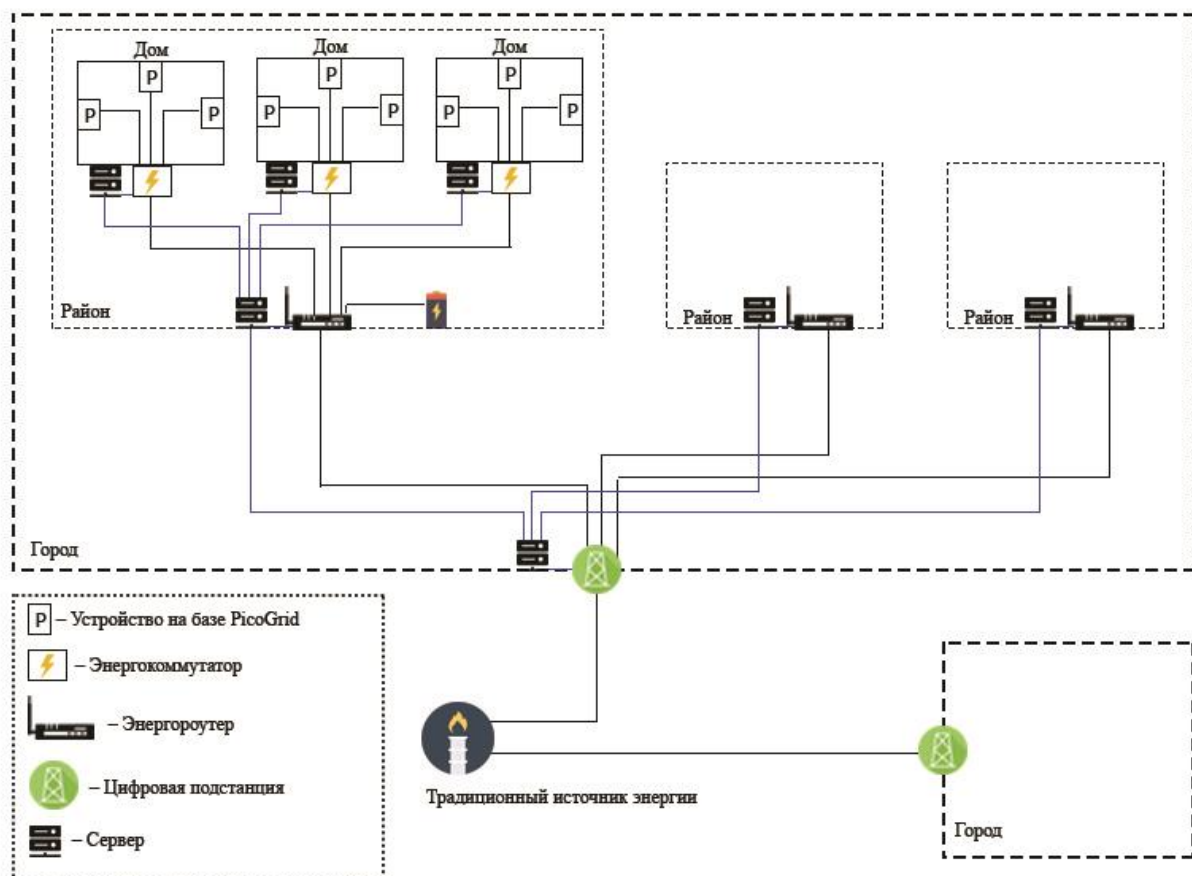


Рис. 1. Архитектурная модель системы.

3. Разработка концептуальной модели системы. Концептуальная модель описывает участников системы (актеров) и основные сценарии и варианты работы системы (прецеденты). Взаимоотношения актеров и прецедентов представлены на диаграмме вариантов использования (рис. 2).

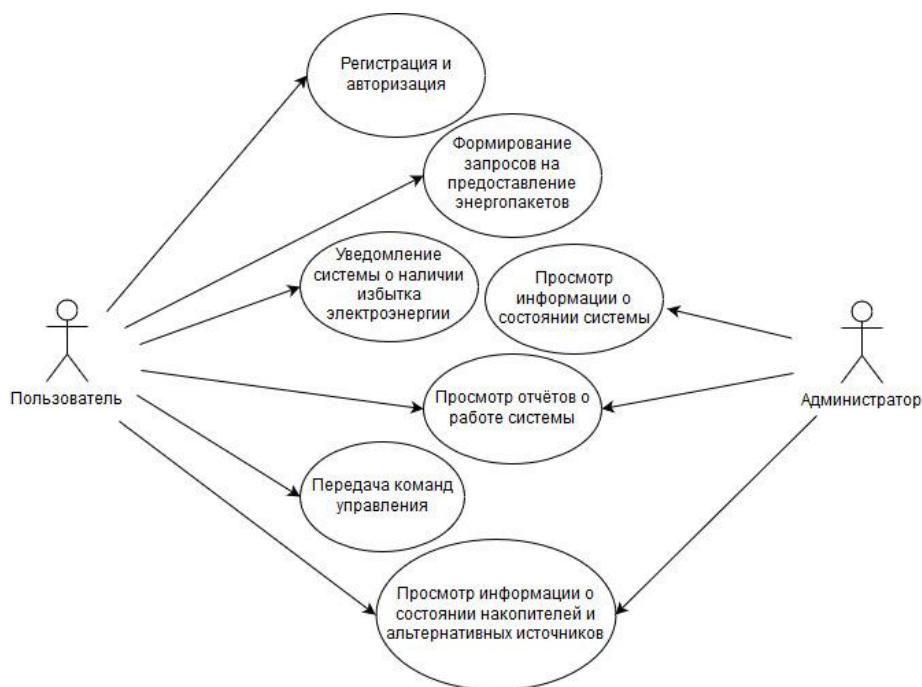


Рис. 2. Диаграмма вариантов использования.

4. Разработка логической модели системы. Логическая модель включает описание и диаграммы наиболее важных модулей системы, описания значимых классов, диаграммы последовательностей выполнения типовых задач и процессов приложения. На рисунке 3 представлена диаграмма классов системы.

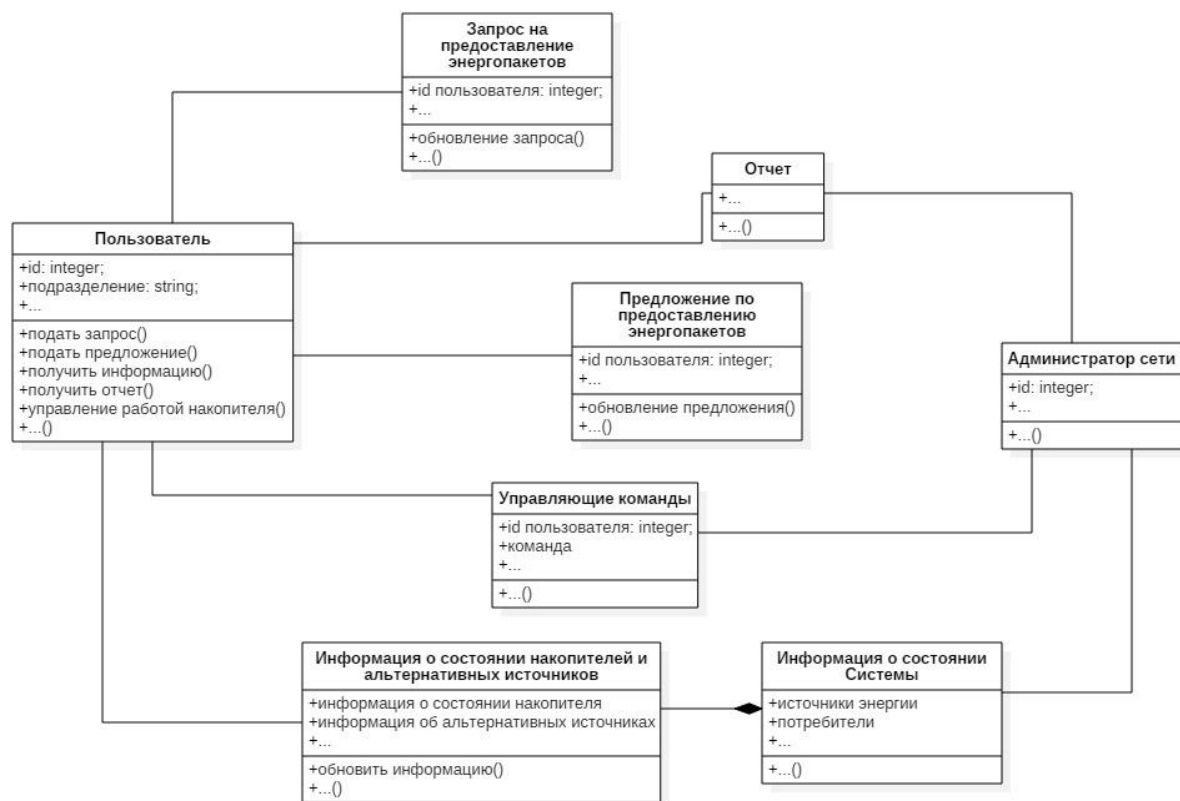


Рис. 3. Диаграмма классов.

Для демонстрации логики взаимодействия объектов системы использована диаграмма последовательностей. На рисунке 4 проиллюстрирован один из возможных сценариев работы системы при отсутствии электроэнергии на накопителе энергии пользователя.

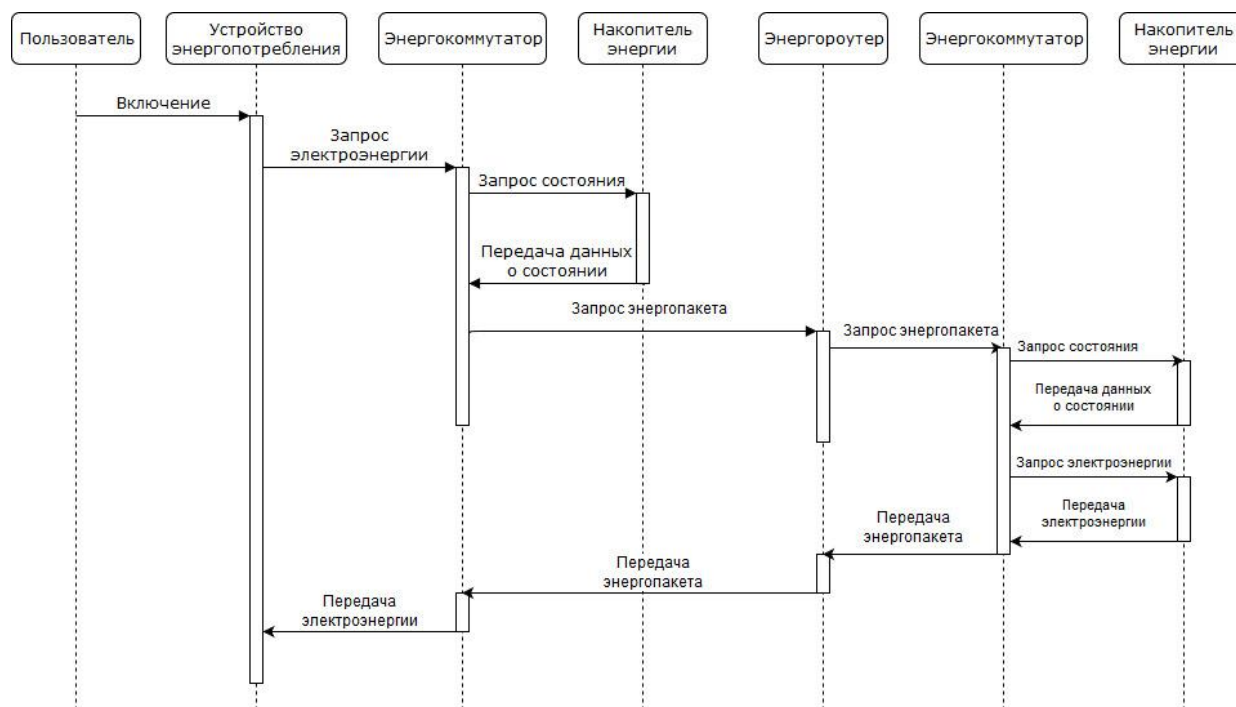


Рис. 4. Диаграмма последовательностей.

При этом стоит отметить, что, если бы в накопителе энергии пользователя имелось достаточное количество энергии, то количество произведенных операций бы значительно сократилось.

5. Разработка модели размещения системы. Модель размещения описывает варианты физического размещения элементов системы, требования к оборудованию.

Для отображения варианта физического размещения элементов энергоинформационной системы использована диаграмма развертывания, представленная на рисунке 5.

Устройства энергопотребления соединяются с энергокоммутаторами напрямую без соединения между собой. Энергокоммутаторы же объединены в некую топологию, в которой также присутствует энергорутер. Аналогично энергорутеры объединены между собой и соединены с цифровой подстанцией топологией, которая определяется индивидуально при каждой реализации.

Энергокоммутаторы кроме координации работы устройств потребления, накопления и генерации электроэнергии и передачи запросов от устройств энергопотребления к

энергорутерам осуществляют «распаковку» энергопакетов, в результате которой вместо энергии с информацией в устройства приходит лишь электроэнергия. При этом происходит понижение мощности с 100-1000 кВт до 10-50 кВт.

Энергорутеры, кроме организации передачи энергопакетов между энергокоммутаторами, принимают энергопакеты от других энергорутеров и от цифровой подстанции, понижая при этом мощность до 100-1000 кВт и напряжение до 220 В. В распоряжении каждого энергорутера есть накопитель энергии, с помощью которого энергорутер может запастись энергией для использования ее во время дефицита и выравнивать скачки напряжения в сети во время интенсивного использования. Стоит отметить, что все городские устройства энергопотребления вроде уличных фонарей подключаются к энергорутеру через свои энергокоммутаторы.

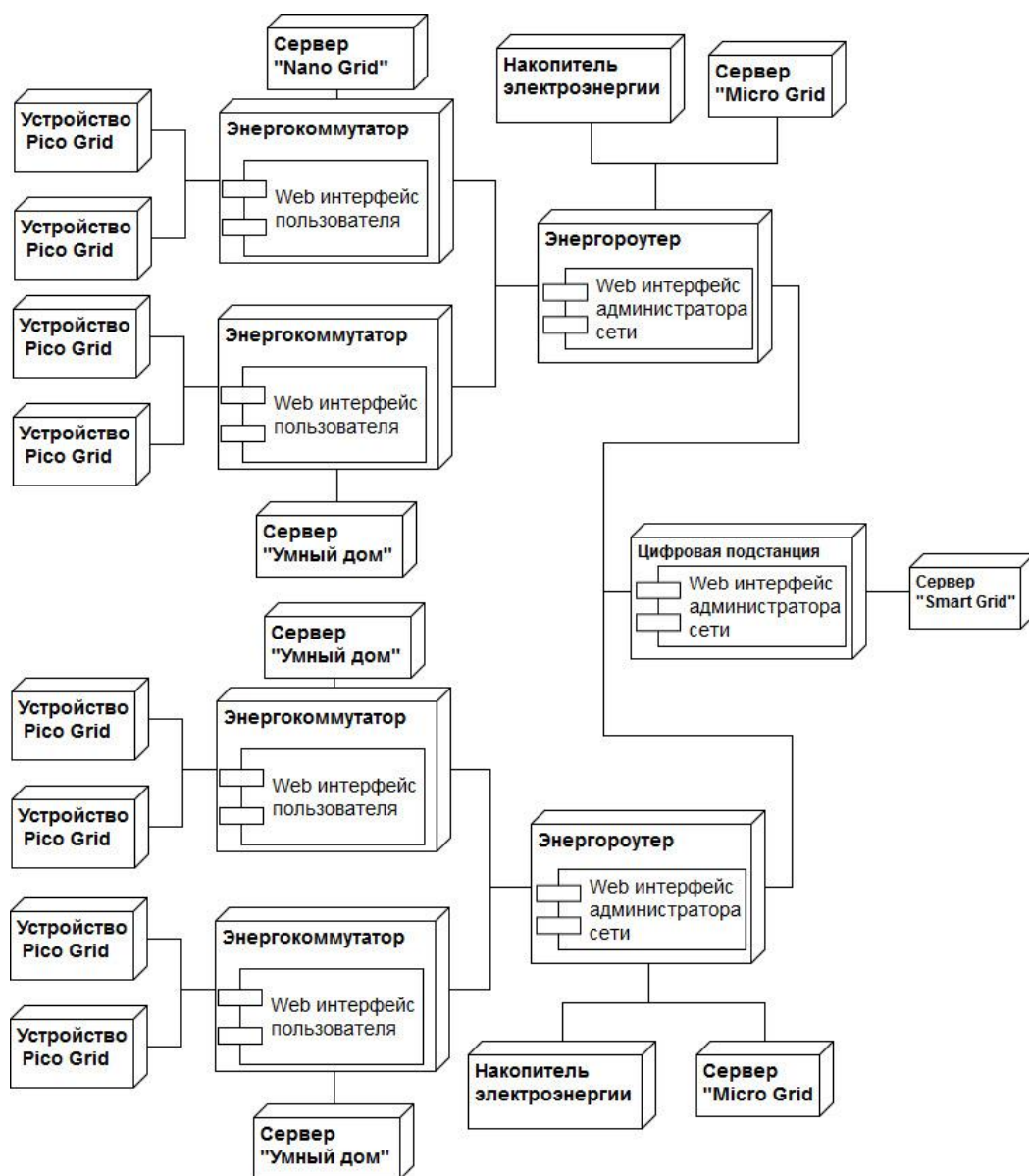


Рис. 5. Диаграмма развертывания.

Основная задача цифровых подстанций, помимо координации работы энергоустройств, – интеграция данной системы с существующей сетью электроэнергетики. Цифровая подстанция работает с токами мощностью выше 1000 кВт и напряжением от 10 до 35 кВ. Принимая электроэнергию, цифровая подстанция понижает напряжение до 10-35 кВ и формирует энергопакеты, с которыми работают энергоустройства.

Для передачи информации используется технология PLC (англ. Power Line Communications), которая позволяет использовать линии электропередач для передачи информации. Таким образом, отпадает необходимость создания сети для соединения серверов – все они будут подключены к устройствам, которые способны кроме электроэнергии передавать и информацию.

6. Разработка модели реализации. Описывает разделение системы на отдельные компоненты, независимые задачи, подпрограммы, информационные и управляющие потоки и связи между элементами системы. Для представления модели реализации используется диаграмма компонентов, представленная на рисунке 6.

Архитектура программного обеспечения энергоинформационной системы делится на три уровня: уровень представления информации, уровень логики, уровень данных.

На уровне представления информации осуществляется взаимодействие системы с пользователем. Через интерфейс пользователя производится ввод данных в систему и вывод запрашиваемой пользователем информации.

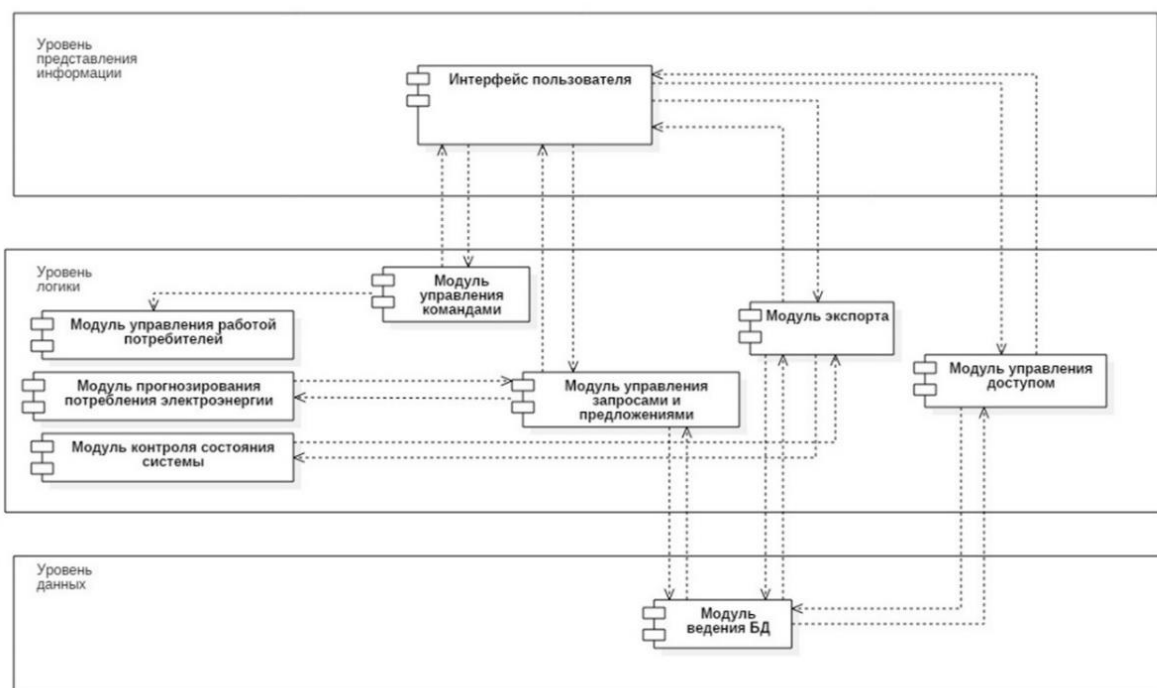


Рис. 6. Диаграмма компонентов.

Все вычисления системы производятся на уровне логики. Здесь обрабатываются команды и запросы, принимаемые от пользователя, формируются отчеты, осуществляется управление распределением энергии.

Уровень данных содержит базу данных, в которой находятся все данные о пользователях, тарифах и других данных, получаемых от пользователя или вводимых администратором сети.

Заключение. Среди проблем энергетической сферы выделяют следующие три, которые в наибольшей степени влияют на все аспекты человеческой жизни и затрагивают сами основы устойчивого развития цивилизации. Эту триаду составляют: 1) дефицит энергоресурсов и электроэнергии; 2) угроза благополучию окружающей среды вследствие техногенного воздействия объектов энергетики; 3) геополитические и социальные угрозы.

Разработанная в статье архитектура описывает систему, которая может решить первые две из указанных проблем.

ЛИТЕРАТУРА

1. Энергетическая стратегия России на период до 2030 года, утв. распоряжением Правительства Российской Федерации от 13 ноября 2009 г. № 1715 [Электронный ресурс]. – Режим доступа: <https://minenergo.gov.ru/node/1026>.

ГОРЮНОВ М. В., ПЕСКОВА Е. Е.
РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ ДЛЯ РЕШЕНИЯ
УРАВНЕНИЙ ХИМИЧЕСКОЙ КИНЕТИКИ¹

Аннотация. Работа посвящена разработке программного модуля для решения уравнений химической кинетики с использованием специализированной явной схемы второго порядка. Были решены системы уравнений, описывающие механизм пиролиза этана. Показано, что реализованный алгоритм дает заметный выигрыш по времени выполнения.

Ключевые слова: химическая кинетика, жесткие системы дифференциальных уравнений, пиролиз этана.

GORYUNOV M. V., PESKOVA E. E.
DEVELOPMENT OF SOFTWARE MODULE
FOR SOLVING CHEMICAL KINETICS EQUATIONS

Abstract. The study deals with the development of a specific software module for solving chemical kinetics equations using a specialized explicit second-order scheme. The equations systems describing the ethane pyrolysis mechanism have been solved. It is shown that the implemented algorithm gives a noticeable gain on the execution time.

Keyword: chemical kinetics, hard systems of differential equations, ethane pyrolysis.

Введение. Химия – экспериментальная наука, в которой перед практическим использованием результатов, исследования веществ и реакций должны проверяться на опыте с последующими рекомендациями. На сегодняшний день актуальным подходом является математическое моделирование химических процессов. Оно зарекомендовало себя как эффективный и безопасный метод исследования, не требующий больших материальных затрат. Основные причины этого – технический прогресс, который определяется, прежде всего, фантастическими достижениями компьютерных технологий и успехами развития теоретических представлений о строении веществ.

Моделирование химических реакций в газах имеет огромную область практических приложений, таких как работа ТЭЦ, бензиновых и дизельных моторов, газофазные химические реакторы.

При численном решении задач химической кинетики возникают трудности, связанные с жесткостью решаемой системы дифференциальных уравнений. Жесткими называют задачи, решение которых содержит компоненты с резко различными характерными

¹ Работа выполнена при поддержке Минобрнауки РФ (№ 1.6958.2017/8.9) и Российского фонда фундаментальных исследований (проект № 18-31-00102).

временами изменения [1]. При решении жестких систем ОДУ используются многошаговые методы Гира и схемы Розенброка. Однако данные схемы обладают высокой трудоемкостью.

Для численного решения задач химической кинетики в работе [2] предложен метод, основанный на специфической форме правых частей, которыми обладают дифференциальные уравнения, построенные на основе кинетических схем. Это явный алгоритм, имеющий малую трудоемкость и существенно превосходящий известные методы по простоте и надежности.

Численный метод. Предположим, что реагирующая смесь состоит из M компонент. Химический процесс можно описать совокупностью N элементарных стадий:

$$\sum_{i=1}^M v'_{in} A_i \leftrightarrow \sum_{i=1}^M v''_{in} A_i, n = \overline{1, N}.$$

Здесь A_i – химические элементы, v'_{in} , v''_{in} – стехиометрические коэффициенты компонента A_i в стадии n .

Система уравнений химической кинетики записывается следующим образом:

$$\frac{dc_i}{dt} = \sum_{n=1}^N (v''_{in} - v'_{in}) w_n, \quad i = 1, \dots, M, \quad (1)$$

где c_i – концентрация компонента A_i , w_n – скорость элементарной стадии реакции:

$$w_n = k_n \prod_{i=1}^M (c_i)^{v'_{in}} - k_{-n} \prod_{i=1}^M (c_i)^{v''_{in}},$$

k_n , k_{-n} – константы скоростей прямой и обратной стадии реакции, которые определяются из выражений Аррениуса [3]:

$$k_n = A_n \cdot e^{\left(-\frac{E_n}{RT}\right)}.$$

Здесь A_n – предэкспоненциальный множитель, E_n – энергия активации.

Согласно [2], систему (1) можно представить в следующем виде:

$$\frac{dc_i}{dt} = -c_i \varphi_i(\mathbf{c}) + \psi_i(\mathbf{c}), \quad \mathbf{c} = (c_1, c_2, \dots, c_M), \quad i = 1, \dots, M.$$

Здесь $c_i \geq 0$, $\varphi_i(\mathbf{c}) \geq 0$, $\psi_i(\mathbf{c}) \geq 0$.

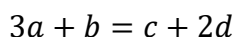
Решение данной системы находится простыми итерациями [2]:

$$\hat{c}_i^{s+1} = \frac{c_i + \tau\psi_i(\bar{c}^s)(1 + \tau\varphi_i(\bar{c}^s)/2)}{1 + \tau\varphi_i(\bar{c}^s) + (\tau\varphi_i(\bar{c}^s))^2/2}, \bar{c}^s = \frac{(c + \hat{c})}{2}, \hat{c}^0 = c.$$

Здесь c_i – решение в исходный момент времени, $\hat{c}_i$ – решение в новый момент времени.

При численном решении системы выполняем две итерации, последующие итерации не повышают порядок точности и ухудшают надежность схемы [2].

Программная реализация. Рассмотрим элементарную стадию реакции следующего вида:



Занумеруем компоненты: $a - 1, b - 2, c - 3, d - 4$. Для каждой реакции указываются номера компонент a, b, c, d в том порядке, в котором они записаны в реакции. В случае отсутствия некоторой компоненты записываем 0. Также будем полагать, что 0 не могут быть равны одновременно a и b, c и d .

Следующий этап – это запись коэффициентов, стоящих перед каждой компонентой. Коэффициент перед компонентой $a - \bar{a}, b - \bar{b}, c - \bar{c}, d - \bar{d}$.

Таким образом, каждая реакция определяется набором параметров $a, b, c, d, \bar{a}, \bar{b}, \bar{c}, \bar{d}$, а также предэкспоненциальным множителем A и энергией активации E . Эти 10 величин определяют структуру элементарной стадии реакции:

$$S = (a, b, c, d, \bar{a}, \bar{b}, \bar{c}, \bar{d}, A, E) \quad (2)$$

Схема реакции представляется в виде таблицы, число строк в которой равно числу элементарных стадий. Добавление новых стадий сводится к добавлению строк этой таблицы.

Реальные химические задачи описываются большим количеством элементарных стадий с сотнями реагирующих компонент. В данной работе реализован алгоритм, который определяет φ_i и ψ_i автоматически. Этот алгоритм, реализованный в функциях *solve_phi* и *solve_psi*, сводится к цепочке последовательных проверок, которые находятся внутри двух циклов: внешнего и внутреннего. Внешний *0...count_substance* – позволяет выбрать конкретное вещество. Внутренний *0...count_reaction* – позволяет пройти по всем реакциям и найти совпадения, которые впоследствии будут обработаны исходя из условий.

Если φ_i – это φ для i -того вещества; ψ_i – это ψ для i -того вещества; k_j – константа скорости для j -той стадии реакции; c_a – концентрация вещества, с номером a ; c_b – концентрация вещества, с номером b .

Для *phi* совпадения ищем среди a и b .

Если исходные вещества реакции одинаковые и оба совпадают с i -тым веществом ($a = b$), то $\varphi_i = \varphi_i + (\bar{a} + \bar{b}) \cdot k_j \cdot c_a^{\bar{a} + \bar{b} - 1}$.

Если a равно i -тому веществу и $b = 0$, то $\varphi_i = \varphi_i + \bar{a} \cdot k_j \cdot c_a^{\bar{a} - 1}$, если b любое другое вещество, то $\varphi_i = \varphi_i + \bar{a} \cdot k_j \cdot c_a^{\bar{a} - 1} c_b$.

Если b равно i -тому веществу и $a = 0$, то $\varphi_i = \varphi_i + \bar{b} \cdot k_j \cdot c_b^{\bar{b} - 1}$, если a любое другое вещество, то $\varphi_i = \varphi_i + \bar{b} \cdot k_j \cdot c_b^{\bar{b} - 1} c_a$.

Для psi ищем совпадения в c и d .

Если c равно i -тому веществу, возникает три случая. Если $a = 0$, то $\psi_i = \psi_i + \bar{c} \cdot k_j \cdot c_b^{\bar{b}}$; если $b = 0$, то $\psi_i = \psi_i + \bar{c} \cdot k_j \cdot c_a^{\bar{a}}$; если a и b разные вещества, то $\psi_i = \psi_i + \bar{c} \cdot k_j \cdot c_a^{\bar{a}} \cdot c_b^{\bar{b}}$.

Если d равно i -тому веществу, возникают также три случая. Если $a = 0$, то $\psi_i = \psi_i + \bar{d} \cdot k_j \cdot c_b^{\bar{b}}$; если $b = 0$, то $\psi_i = \psi_i + \bar{d} \cdot k_j \cdot c_a^{\bar{a}}$; если a и b разные вещества, тогда $\psi_i = \psi_i + \bar{d} \cdot k_j \cdot c_a^{\bar{a}} \cdot c_b^{\bar{b}}$.

Вычислительный эксперимент. Описанный алгоритм был реализован на языке программирования C++. Для проверки работы программного модуля была решена система уравнений, описывающая процесс пиролиза этана. В качестве кинетической схемы принята брутто-реакция пиролиза этана, состоящая из двух элементарных стадий [4; 5].

Таблица 1

Схема и кинетические параметры механизма брутто-реакции пиролиза этана

№	Стадия	$A_i, 1/c$ или $л/(моль \cdot с)$	$E_i, Дж/моль$
1	$C_2H_6 \rightarrow C_2H_4 + H_2$	$1.08E + 16$	$2.5E + 5$
2	$2C_2H_6 \rightarrow C_2H_4 + 2CH_4$	$3.16E + 16$	$2.7E + 5$

Для записи данных о каждой элементарной стадии реакции пронумеруем компоненты смеси: $B_1 = [C_2H_6]$, $B_2 = [C_2H_4]$, $B_3 = [H_2]$, $B_4 = [CH_4]$.

Соответственно, имеем следующую запись для двух стадий реакции:

$$(1, 0, 2, 3, 1, 0, 1, 1, 1.08 \cdot 10^{16}, 2.5 \cdot 10^5)$$

$$(1, 0, 2, 4, 2, 0, 1, 2, 3.16 \cdot 10^{16}, 2.7 \cdot 10^5)$$

Задачу решаем со следующими начальными условиями:

$$T = 800K, c_1(0) = 1, c_2(0) = 0, c_3(0) = 0, c_4(0) = 0$$

На рисунке 1 представлены результаты работы программы в момент установления. Полученные результаты хорошо согласуются с результатами, полученными другими

авторами [6; 7]. В таблице 2 представлены значения концентраций в установившемся течении.

Таблица 2

Сравнение результатов работы программы с полученными ранее

Концентрации	Результаты текущей работы	Результаты, полученные аналитически в [6]
c_1	0	0
c_2	0.939841	0.94
c_3	0.878950	0.88
c_4	0.121782	0.12

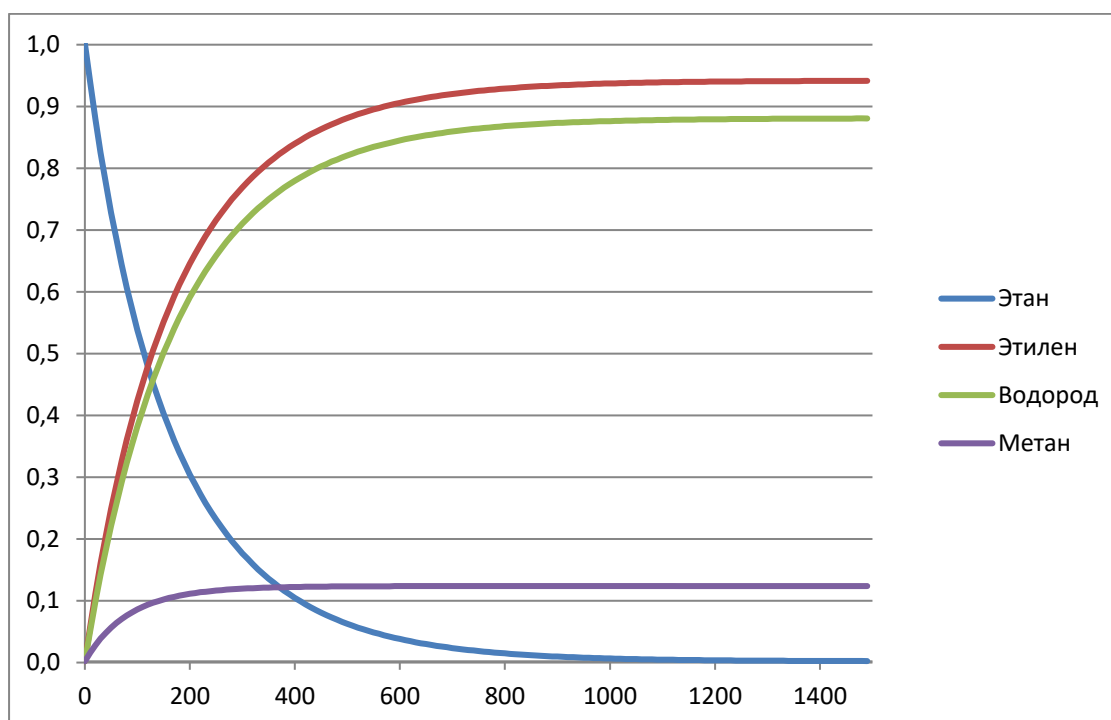


Рис. 1. Изменение концентраций веществ при температуре 800 К, брутто-схема.

Для более точного описания химических превращений в вычислительной практике, как правило, используются схемы с большим количеством элементарных стадий реакции. Покажем, что разработанный программный код дает выигрыш по времени вычислений в сравнении с явным методом Эйлера и методом Розенброка второго порядка, реализованного в системе MATLAB.

Для построения математической модели пиролиза этана принята 15-ти стадийная схема реакции, включающая 12 компонент смеси, представленная в таблице 3 [8].

Схема и кинетические параметры радикального механизма пиролиза этана

№	Стадия	$\lg A_i, 1/c$ или $l/(моль \cdot c)$	$E_i, \text{кДж/моль}$
1	$C_2H_6 \rightarrow CH_3^\bullet + CH_3^\bullet$	16.0	360.0
2	$CH_3^\bullet + C_2H_6 \rightarrow CH_4 + C_2H_5^\bullet$	10.0	50.0
3	$C_2H_5^\bullet \rightarrow C_2H_4 + H^\bullet$	13.5	170.0
4	$H^\bullet + C_2H_6 \rightarrow H_2 + C_2H_5^\bullet$	11.0	40.0
5	$H^\bullet + C_2H_4 \rightarrow C_2H_5^\bullet$	10.4	8.4
6	$CH_3^\bullet + C_2H_4 \rightarrow n - C_3H_7^\bullet$	10.9	33.0
7	$n - C_3H_7^\bullet \rightarrow CH_3^\bullet + C_2H_4$	13.9	137.0
8	$C_2H_5^\bullet + C_2H_5^\bullet \rightarrow C_2H_4 + C_2H_6$	10.0	8.4
9	$n - C_3H_7^\bullet + C_2H_4 \rightarrow C_2H_5^\bullet + C_3H_6$	7.4	27.6
10	$CH_3^\bullet + C_2H_4 \rightarrow CH_4 + C_2H_3^\bullet$	8.6	35.0
11	$CH_3^\bullet + C_2H_3^\bullet \rightarrow CH_4 + C_2H_2$	9.95	3.2
12	$C_2H_3^\bullet + H^\bullet \rightarrow C_2H_2 + H_2$	10.0	0.0
13	$C_2H_4 \rightarrow \cdot C_2H_4$	15.8	253.0
14	$\cdot C_2H_4 + C_2H_6 \rightarrow CH_3^\bullet + n - C_3H_7^\bullet$	14.7	216.0
15	$\cdot C_2H_4 \rightarrow C_2H_4$	5.38	0.0

Введем следующие обозначения: $B_1 = [C_2H_6]$, $B_2 = [C_2H_4]$, $B_3 = [H_2]$, $B_4 = [CH_4]$, $B_5 = [CH_3^\bullet]$, $B_6 = [C_2H_5^\bullet]$, $B_7 = [H^\bullet]$, $B_8 = [n - C_3H_7^\bullet]$, $B_9 = [C_3H_6]$, $B_{10} = [C_2H_3^\bullet]$, $B_{11} = [C_2H_2]$, $B_{12} = [\cdot C_2H_4]$.

Формируем записи элементарных стадий реакции согласно таблице 3 (табл. 4).

Запись стадий реакции для дальнейших расчетов

№	Запись стадий реакции
1	(1,0,5,0,1,0,2,0,16,360)
2	(5,1,4,6,1,1,1,1,10,10,50)
3	(6,0,2,7,1,0,1,1,13,170)
4	(7,1,3,6,1,1,1,1,11,40)
5	(7,2,6,0,1,1,1,0,10,4,8.4)
6	(5,2,8,0,1,1,1,0,10,9,33)
7	(8,0,5,2,1,0,1,1,13,9,137)
8	(6,6,2,1,1,1,1,1,10,8.4)
9	(8,2,6,9,1,1,1,1,7,4,27.6)
10	(5,2,4,10,1,1,1,1,8,6,35)
11	(5,10,4,11,1,1,1,1,9,35,3.2)
12	(10,7,11,3,1,1,1,1,15,10,0)
13	(2,0,12,0,1,0,1,0,15,8,253)
14	(12,1,5,8,1,1,1,1,14,7,216)
15	(12,0,2,0,1,0,1,0,5,38,0)

Задачу решаем со следующими начальными условиями:

$$T = 900 \text{ K}, c_1(0) = 1, c_2(0) = 0, \dots, c_{12}(0) = 0$$

Были проведены расчеты с использованием трех вышеупомянутых методов, получены хорошо согласованные друг с другом результаты. На рис. 2 представлен результат работы разработанного программного модуля.

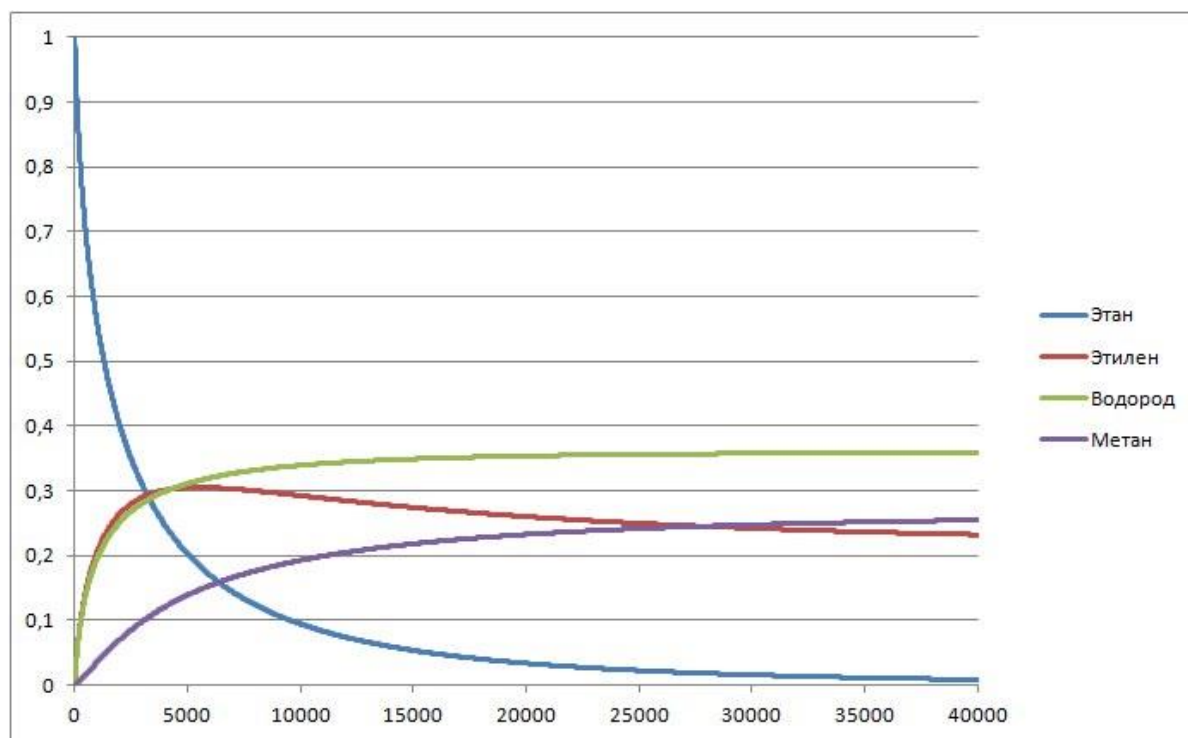


Рис. 2. Изменение концентраций веществ при температуре 800 К.

В таблице 5 представлено время выполнения каждого метода, расчеты велись до времени $t = 1$ сек.

Таблица 5

Время расчета методов

	Разработанный код на основе метода [2]	Разработанный код на основе явного метода Эйлера	Метод Розенброка второго порядка в системе MATLAB
Время	3 мс	29109 мс	138 мс

Таким образом, разработанный программный код имеет преимущество по времени выполнения по отношению к явному методу Эйлера и широко известному методу Розенброка второго порядка для решения жестких ОДУ, реализованному в системе MATLAB.

Заключение. С помощью разработанного программного модуля можно проводить моделирование задач химической кинетики, включающих любое количество элементарных стадий. В дальнейшем планируется включение данного модуля в разрабатываемый пакет программ [9; 10] для расчета динамики многокомпонентного ламинарного газового потока с учетом химических реакций.

ЛИТЕРАТУРА

1. Галанин М. П., Ходжаева С. Р. Методы решения жестких обыкновенных дифференциальных уравнений. Результаты тестовых расчетов // Препринт ИПМ им. М. В. Келдыша. – 2013. – № 98. – 29 с.
2. Белов А. А., Калиткин Н. Н., Кузьмина Л. В. Моделирование химической кинетики в газах // Математическое моделирование. – 2016. – Т. 28, № 8. – С. 46–64.
3. Штиллер В. Уравнение Аррениуса и неравновесная кинетика. – М.: Мир, 2000. – 176 с.
4. Мухина Т. Н., Барабанов Н. Л., Бабаш С. Е. и др. Пиролиз углеводородного сырья. – М.: Химия, 1987. – 240 с.
5. Губайдуллин И. М., Пескова Е. Е., Язовцева О. С. Математическая модель динамики многокомпонентного газа на примере брутто-реакции пиролиза этана [Электронный ресурс] // Огарев-online. – 2016. – № 20. – Режим доступа: <http://journal.mrsu.ru/arts/matematiceskaya-model-dinamiki-mnogokomponentnogo-gaza-na-primere-brutto-reakcii-piroliza-etana>.
6. Язовцева О. С. Локальная покомпонентная асимптотическая эквивалентность и ее применение к исследованию устойчивости по части переменных [Электронный ресурс] // Огарев-online. – 2017. – № 13. – Режим доступа: <http://journal.mrsu.ru/arts/lokalnaya-pokomponentnaya-asimptoticheskaya-ekvivalentnost-i-ee-primenenie-k-issledovaniyu-ustojchivosti-po-chasti-peremennykh>.
7. Назаров В. И., Пескова Е. Е., Язовцева О. С. Численное моделирование жестких систем с использованием (4,2)-метода [Электронный ресурс] // Огарев-online. – 2017. – № 13. – Режим доступа: <http://journal.mrsu.ru/arts/chislennoe-modelirovanie-zhestkix-sistem-s-ispolzovaniem-42-metoda>.
8. Nurislamova L. F, Stoyanovskaya O. P., Stadnichenko O. A., Gubaidullin I. M., Snytnikov V. N., Novichkova A. V. Few-Step Kinetic Model of Gaseous Autocatalytic Ethane Pyrolysis and Its Evaluation by Means of Uncertainty and Sensitivity Analysis // Chemical Product and Process Modeling. – 2014. – 9 (2). – pp. 143–154.

9. Жалнин Р. В., Пескова Е. Е., Стадниченко О. А., Тишкин В. Ф. Математическое моделирование динамики многокомпонентного газа с использованием WENO схем на примере пиролиза этана // Журнал Средневолжского математического общества. – 2016. – Т. 18, № 3. – С. 98–106.
10. Жалнин Р. В., Пескова Е. Е., Стадниченко О. А., Тишкин В. Ф. Моделирование течения многокомпонентного реагирующего газа с использованием алгоритмов высокого порядка точности // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. – 2017. – Т. 27, № 4. – С. 608–617.

ТИМОВКИН С. Н., ЕГОРОВА Д. К.
РЕАЛИЗАЦИЯ МЕТОДА ГЛАВНЫХ КОМПОНЕНТ
В ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ R

Аннотация. В работе рассмотрена реализация метода главных компонент в вычислительной среде R. В качестве исходных данных использовались рейтинговые показатели вузов.

Ключевые слова: вычислительная среда R, анализ, визуализация, многомерные данные, снижение размерности, метод главных компонент.

TIMOVKIN S. N., EGOROVA D. K.
IMPLEMENTATION OF PRINCIPLE COMPONENT ANALYSIS
IN COMPUTING ENVIRONMENT R

Abstract. The article considers the principal component analysis in the computing environment R. The initial data are the ratings of the universities.

Keywords: computing environment R, analysis, visualization, multidimensional data, dimensional reduction, principal component analysis.

Среда R является программным средством, предназначенным для анализа многомерных данных и их визуализации, с мощными графическими возможностями. Среди множества других видов программного обеспечения ее выделяет то, что это среда с открытым исходным кодом, развиваемая в рамках проекта по разработке свободного программного обеспечения (GNU), причем функциональность ее базовой версии может быть значительно увеличена путем применения расширенной системы команд или пакетов, которые так же находятся в свободном доступе. В данной среде используется интерфейс командной строки, но для удобства работы созданы и интегрированные среды разработки (IDE), например, RStudio, JGR, RKWard, SciViews-R, R Commander. В данной работе рассмотрена реализация метода главных компонент в RStudio, которая написана на C++ и использует фреймворк Qt для графического интерфейса пользователя [1].

Рассмотрим следующую задачу: требуется провести анализ и визуализацию рейтинговых показателей вузов. В качестве исходных данных были выбраны двадцать семь вузов Приволжского федерального округа, городов Москва и Санкт-Петербург. Данные взяты с интернет-сайта Рейтинг вузов России [2]. Перечислим их: КФУ, КГМУ, КНИТУ, КНИТУ-КАИ им. А.Н. Туполева, СамГТУ, СамГМУ, СамГУ, САУ им. ак. С.П. Королёва, МГУ им. Н.П. Огарёва, НГТУ им. Р. Е. Алексеева, ННГУ им. Н.И. Лобачевского, ИжГТУ им. М. Т. Калашникова, СГУ им. Н. Г. Чернышевского, СГМУ им. В. И. Разумовского,

ПНИПУ, ПГНИУ, МИ «РЕАВИЗ», НИЯУ, НИУ ВШЭ, ТПУ, ТГУ, НГУ, ИТМО, МИФИ, НИТУ «МИСиС», НИУ «МЭИ», МАИ, НИУ «МИЭТ». Каждый из вузов оценивается по девяти рейтинговым показателям, подразделенными на три основные группы и итоговым рейтинговым функционалом.

Условия для получения качественного образования

1. Международная интеграция.
2. Востребованность среди абитуриентов.
3. Уровень преподавания.
4. Ресурсное обеспечение.

Востребованность у работодателя

5. Сотрудничество с работодателями.
6. Качество карьеры выпускников.

Научно исследовательская активность

7. Инновационная активность.
8. Научные достижения.
9. Инновационная инфраструктура.

Визуализация рейтинговых показателей возможна, если будет произведено снижение размерности исходного признакового пространства с девяти показателей к двум или трем, тогда каждый вуз будет характеризоваться двумерной или трехмерной точкой в декартовой прямоугольной системе координат. В качестве метода снижения размерности был выбран метод главных компонент [3]. Согласно данному методу, переход к новому набору показателей осуществляется таким образом, чтобы выполнялось условие $I_{p'}(\tilde{Z}(X)) = \max_{Z \in F} \{I_{p'}(Z(X))\}$, где $I_{p'}$ – мера информативности, $Z(X)$ – p -мерная вектор-функция исходных переменных, $\tilde{Z}(X)$ – p' -мерная вектор-функция новых переменных и $p' \ll p$. В качестве класса допустимых преобразований F берут всевозможные линейные ортогональные нормированные комбинации исходных показателей и переходят к p' первым главным компонентам по формуле

$$z^{(1)}(X) = c_{j1}(x^{(1)} - \mu^{(1)}) + \dots + c_{jp}(x^{(p)} - \mu^{(p)}).$$

Здесь на коэффициенты накладываются условия нормировки $\sum_{v=1}^p c_{jv}^2 = 1, j = 1, 2, \dots, p'$, $\sum_{v=1}^p c_{jv} * c_{kv} = 0, k = 1, 2, \dots, p, j \neq k$, где $\mu^{(v)}$ – математическое ожидание $x^{(v)}$. Критерием оптимальности p' -мерной системы новых показателей является выражение

$$I_{p'}(Z(X)) = \frac{Dz^{(1)} + \dots + Dz^{(p')}}{Dx^{(1)} + \dots + Dx^{(p')}}, \text{ где } D - \text{ дисперсия.}$$

Для реализации метода главных компонент в среде R к ее базовой версии могут быть подключены пакеты *ade4* и *vegan* (далее в работе будут приведены отрывки кода R с соответствующими комментариями) [4]. Следующая команда R ставит в соответствие каждому из двадцати семи вузов вектора с рейтинговыми показателями, затем все они записываются в массив:

```
VUZ<-rbind(KazanPrFedUn,SamGosAeroUn,KazGosMedUnZdr,NizhGosUnLob,SamGosMedUnZdr,  
PermNazIslPolUn,SamGosTechUn,KazNazIslTechUn,KazNazIslUnTupol,SamGosUn,  
MordGosOGAREVA,NizhGosTechAleks,IzhGosTechKalash,SargosUnChernish,  
SargosMedRazumovsk,PermGosNazIslUn,MedInstReaviz,MIFI,VShEco,TomSPolTec,  
Novosib,TomGos,MISiS,MEI,SPBInfTech,MoskAvia,MIET)
```

Далее каждому рейтинговому показателю (компоненте) присвоены имена:

```
colnames(VUZ) <- c("MezhdIntegr", "VostrebAbitur", "UrovenPrepod", "ResursObesp",  
"SotrudnSRabotodmi", "Kachestvokarerivip", "InnovatzAktiv", "NauchnDostig",  
"InnovationInfrastr")
```

Затем была произведена оценка взаимосвязи исходных данных путем составления корреляционной таблицы, из которой можно отметить достаточно высокую взаимосвязь:

- «Международной интеграции» с «Уровнем преподавания», «Ресурсным обеспечением», «Востребованностью абитуриентами» (входят в группу «Условия для получения качественного образования»);
- «Востребованности абитуриентами» с «Ресурсным обеспечением»;
- «Уровня преподавания» с «Ресурсным обеспечением» и «Качеством карьеры выпускников»;
- «Ресурсного обеспечения» и «Качества карьеры выпускников»;
- «Инновационной активности» и «Инновационной инфраструктуры».

Полное отсутствие корреляции наблюдается у «Сотрудничества с работодателями» со всеми другими критериями.

Следующие команды R создают объект VUZ.pca и формируют главные компоненты:

```
VUZ.pca <-prcomp(VUZ)  
pc <-VUZ.pca$x  
round(pc,4)
```

Результат выполнения команд приведен на рисунке 1.

##	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
## KazanPrFedUn	-3.1334	0.0279	-0.6446	-1.0625	-0.4141	0.0439	-0.1118	0.3969	0.2303
## SamGosAeroUn	-0.2790	0.3641	-0.3254	1.9837	-0.1741	0.1249	-0.1059	-0.0433	0.1072
## KazGosMedUnZdr	1.1261	-1.1972	1.8275	0.1365	0.5045	-0.7698	0.2700	-0.4384	0.3910
## NizhGosUnLob	-0.1768	-0.5181	-1.0181	0.4141	0.8561	0.1161	-0.1419	0.3750	0.0242
## SamGosMedUnZdr	2.0767	0.3743	1.1172	0.1166	0.0784	0.0760	0.1759	0.7126	0.2544
## PermNazIslPolUn	1.4958	1.4356	-0.9270	0.5311	0.0104	-0.0639	-0.2668	0.0643	-0.0757
## SamGosTechUn	1.9513	0.9483	-0.0364	0.5204	-0.7608	0.1829	0.9649	0.2349	-0.1226
## KazNazIslTechUn	1.7175	-0.0582	-0.6398	-0.4909	0.3439	-0.0977	0.6258	0.0529	0.2390
## KazNazIslUnTupoI	1.5016	-0.3023	-0.6120	-0.5417	-0.8341	0.1751	-0.7153	-0.1706	0.0314
## SamGosUn	1.6339	-1.6237	-1.1333	0.3075	-0.9972	-0.7228	0.7535	-0.4096	-0.1440
## MorGosOGAREVA	2.0398	1.2830	-0.5847	-0.5343	0.3181	0.7254	0.2646	-0.5878	0.1331
## NizhGosTechAleks	2.6496	0.9065	0.3103	0.4377	-0.0255	-0.0771	0.0672	-0.0828	-0.0775
## IzhGosTechKalash	2.3457	1.0751	0.1649	-0.4581	1.3018	-0.1682	0.1481	0.2080	-0.7001
## SarGosUnChernish	1.7140	-0.3955	-2.1058	-0.2580	0.2547	-0.7889	-0.5482	-0.1475	0.2451
## SarGosMedRazumovsk	3.1626	-0.7705	1.4617	-0.2735	-0.1353	0.1855	-0.1671	0.0274	0.0741
## PermGosNazIslUn	2.4908	-0.8453	-0.6263	-0.0153	-0.6259	0.4602	-0.0974	0.3952	-0.2118
## MedInstReaviz	3.5688	-0.5861	1.2731	-1.0084	-0.2158	-0.6027	-0.2380	-0.0183	-0.2238
## MIFI	-4.6318	0.5311	0.2319	0.4380	0.4475	-1.1256	0.0346	0.0502	0.0590
## VShEco	-4.1826	0.0411	0.8583	0.9165	-0.5500	0.3969	0.1132	-0.2666	-0.1983
## TomsPolTec	-4.5735	-0.0704	-0.3439	-0.6626	-0.0798	-0.2433	0.1234	0.0173	-0.2283
## Novosib	-3.9651	-0.2649	0.5397	-0.0130	-0.7503	-0.5753	-0.3352	0.4370	-0.2086
## TomsGos	-3.0387	0.4365	-0.2848	-0.3757	0.6477	0.2017	0.5780	-0.1619	0.2007
## MISis	-2.9984	-1.2168	0.0758	-0.6255	0.1696	1.0558	0.3805	0.3010	0.2040
## MEI	-2.3315	0.9768	0.5406	-0.7264	-0.4947	0.5540	-0.2490	-0.7005	-0.1793
## SPBInfTech	-1.7680	-1.4494	-0.0191	0.4729	1.1425	0.2637	-0.4130	-0.2738	-0.2362
## MoskAvia	0.1359	2.2645	0.6430	0.0243	-0.2412	-0.1620	-0.5934	0.0372	0.3617
## MIET	1.4687	-1.3663	0.2573	0.7468	0.2237	0.8353	-0.5168	-0.0088	0.0510

Рис. 1. Таблица главных компонент.

Выясним, как первая главная компонента связана с рейтинговым функционалом в таблице. Следующие команды создают вектор с рейтинговыми данными и вычисляют коэффициент корреляции главной компоненты и рейтингового функционала:

```

M<-c(3.4193,2.7737,2.5191,2.5113,2.1431,2.0928,2.0859,1.9914,1.9839,1.96,1.8972,
1.8894,1.8868, 1.7874,1.7649,1.6854,1.5942,4.1514,4.0377,3.9215,3.883,3.5104,
3.4247, 3.3442,3.0739, 2.7339, 2.1233)
cor(M,pc[,1])
## [1] -0.9870379

```

Высокое значение коэффициента корреляции свидетельствует о том, что рейтинговый функционал и первая главная компонента сильно взаимосвязаны. Первая главная компонента вбирает в себя максимум отличий между вузами. Таким образом, не зная способа подсчета рейтингового функционала (представленного в [2]), были восстановлены позиции вузов в таблице рейтинга.

Следующая команда визуализирует корреляцию первой компоненты с итоговым местом вуза в рейтинге (рис.2):

```
plot(Mesto,pc[,1])
```

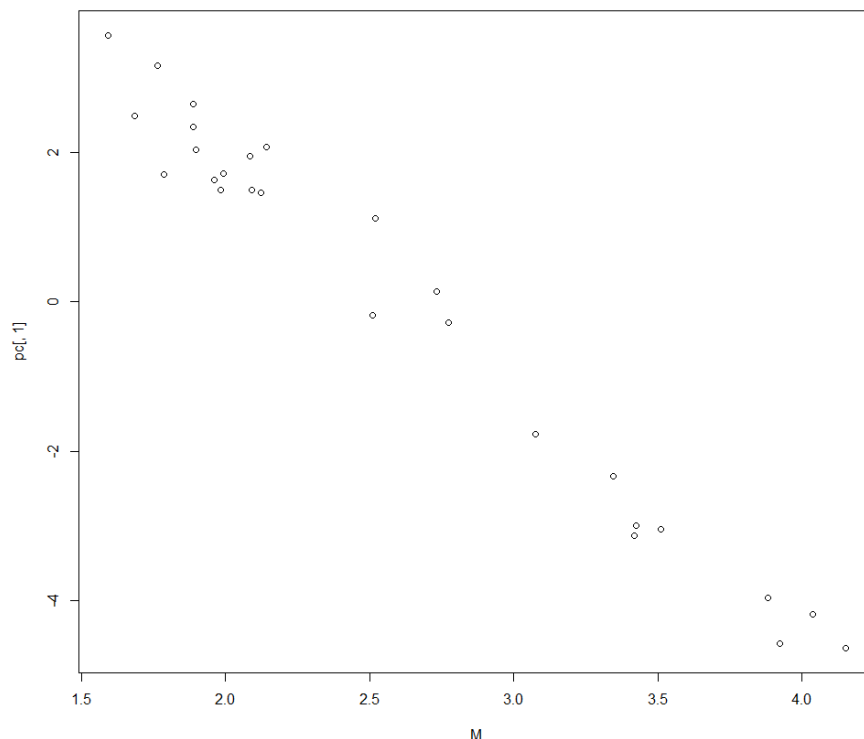


Рис. 2. Корреляционное поле.

На графике корреляционного поля между первой главной компонентой и значением рейтингового функционала наблюдается сильно выраженная прямая отрицательная корреляционная зависимость. Также можно отметить, что с увеличением рейтингового функционала значение главной компоненты убывает.

Следующая команда R позволяет найти основные статистические характеристики компонент. Результат приведен на рисунке 3.

```
summary(VUZ.pca)
```

```
## Importance of components:
##          PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8      PC9
## Standard deviation  2.65 0.98407 0.88107 0.66817 0.59094 0.52228 0.41989 0.32713 0.24007
## Proportion of Variance 0.69 0.09518 0.07629 0.04388 0.03432 0.02681 0.01733 0.01052 0.00566
## Cumulative Proportion 0.69 0.78519 0.86148 0.90536 0.93968 0.96649 0.98382 0.99434 1.00000
```

Рис. 3. Основные статистические характеристики компонент.

Первая строка «Standard deviation» показывает среднеквадратичное отклонение главных компонент. С большим числом компонент отклонение уменьшается, что и должно быть. Вторая строка «Proportion of Variance» объясняет долю отклонений компонент. Из строки «Cumulative Proportion» следует, что первая главная компонента PC1 описывает 69%

дисперсии исходного набора данных, первая и вторая главные компоненты PC1 и PC2 объясняют приблизительно 78,5% исходных данных. Далее видно, что, например, пять главных компонент PC1-PC5 описывают около 94% набора исходных данных и т.д.

С помощью следующих команд извлечем в матрицу *v1* веса, с которыми старые переменные, результаты отдельных критериев вузов, входят в новые переменные (рис.4):

```
v1<-VUZ.pca$rotation
round(v1,4)
```

##	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
## MezhdIntegr	-0.3935	-0.1671	0.3863	-0.4027	0.4748	-0.0001	0.2137	-0.0932	-0.4754
## VostrebAbitur	-0.2892	-0.1989	0.2497	0.6056	0.1579	0.2747	-0.4568	-0.3716	0.0244
## UrovenPrepod	-0.3487	-0.0331	0.1664	-0.3790	-0.2558	-0.5431	-0.5002	-0.0928	0.2945
## ResursObesp	-0.4139	-0.0866	0.1452	-0.1745	-0.2174	0.5977	0.0513	0.4909	0.3485
## SotrudnSRabotodmi	-0.0166	0.4682	0.5161	0.2358	0.2324	-0.2223	0.3704	0.0032	0.4668
## KachestvoKareriVip	-0.3685	0.2241	0.1065	0.3046	-0.6358	-0.1483	0.2561	0.0094	-0.4669
## InnovatzAktiv	-0.2884	0.3055	-0.3192	0.2394	0.4051	-0.2187	-0.2844	0.5856	-0.1688
## NauchnDostig	-0.3960	-0.4685	-0.4369	0.1881	0.0995	-0.2629	0.4573	-0.1221	0.3071
## InnovationInfrastr	-0.3041	0.5845	-0.4113	-0.2364	0.0684	0.2898	0.0052	-0.4957	0.0802

Рис. 4. Матрица весов.

Матрица *v1* показывает, как новые оси смещены относительно старых. Далее каждая компонента вычисляется как сумма произведений значений весов на соответствующий критерий. Например, первая главная компонента будет вычислена так:

$$PC1 = \text{MezhdIntegr} \times (-0.3935) + \text{VostrebAbitur} \times (-0.2892) + \text{UrovenPrepod} \times (-0.3487) + \\ + \text{ResursObesp} \times (-0.4139) + \text{SotrudnSRabotodmi} \times (-0.0166) + \text{KachestvoKareriVip} \times (-0.3685) + \\ + \text{InnovatzAktiv} \times (-0.2884) + \text{NauchnDostig} \times (-0.3960) + \text{InnovationInfrastr} \times (-0.3041).$$

В первом столбце матрицы *v1* показано, что критерий «Международная интеграция» входит в главную компоненту PC1 с весом -0.3935, «Востребованность абитуриентами» с весом -0.2892 и так далее. По таблице можно сказать, что наиболее существенные переменные – это «Международная интеграция», «Ресурсное обеспечение» и «Научные достижения» с весами -0.3935, -0.4139 и -0.3960 соответственно; критерий «Сотрудничество с работодателями» входит в первую главную компоненту с очень маленьким весом -0.0166.

Перед построением графика главных компонент визуализируем долю объясняемой дисперсии разными компонентами. Для этого используем следующую команду:

```
plot(vuz.pca)
```

Результат приведен на рисунке 5.

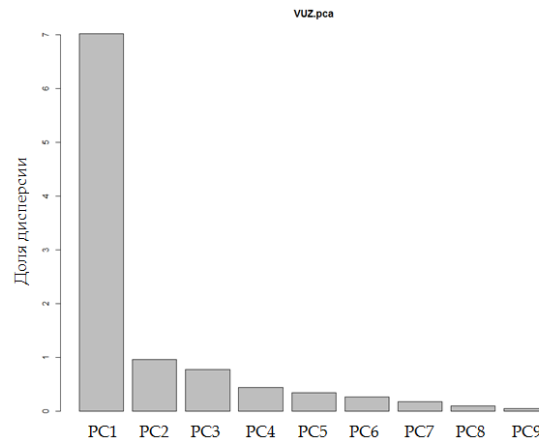


Рис. 5. График относительных вкладов компонент.

График показывает относительные вклады каждой компоненты в общий разброс данных. Хорошо видно, что компонент девять, как и признаков, но, в отличие от первоначальных признаков, наибольший вклад вносит первая главная компонента. Аналогичные результаты, но в текстовом виде, получили в «Результатах оценивания».

Визуализируем наблюдения в осях первой и второй главных компонент командой:

```
biplot(vuz.pca, xlim = c(-0.8, 0.8), ylim = c(-0.6, 0.6))
```

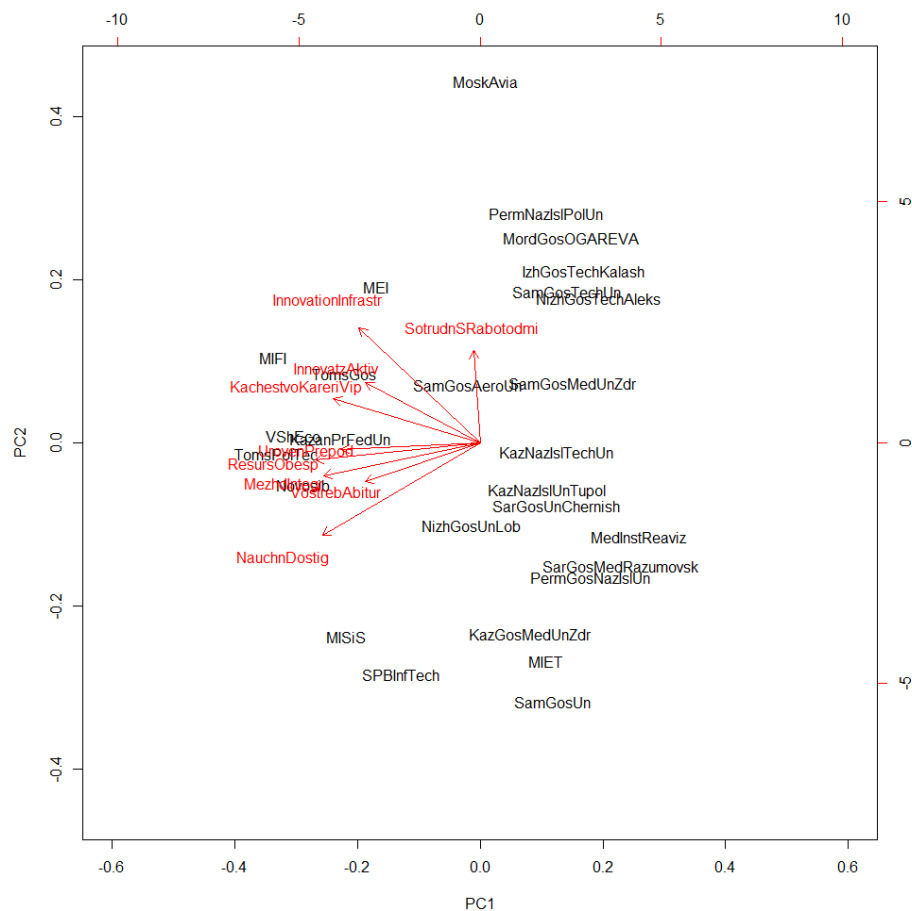


Рис. 6. Визуализация первой и второй главных компонент.

На рисунке 6 приведен график величины вклада каждого из девяти исходных признаков в первые две компоненты.

Красные векторы – это проекции единичных векторов исходных координат на плоскость первых двух главных компонент. Векторы, соответствующие исходным координатам, направлены влево. Это подтверждает то, что коэффициенты в первом столбце матрицы вращения отрицательные. При этом стрелки *SotrudnSRabotodmi*, *KachestvoKareriVip*, *InnovatzAktiv*, *InnovationInfrastr* направлены вверх и соответствующий коэффициент (вторая главная компонента) во втором столбце положительный, остальные – вниз и соответственно коэффициент отрицательный. Проекция вектора *SotrudnSRabotodmi* на ось первой главной компоненты почти нулевая, а на ось второй компоненты – положительная и заметно больше нуля, что также подтверждается таблицей.

Признаки *UrovenPrepod*, *ResursObesp*, *MezhdIntegr* и *VostrebAbitur* вносят очень маленький вклад во вторую главную компоненту, они входят в группу «Условия для получения качественного образования». Их корреляционная зависимость была найдена ранее. Можно предположить, что они дают информацию об общем статусе вуза: высшие учебные заведения занявшие самые высокие позиции в итоговой таблице (МИФИ, НИУ ВШЭ, НГУ, ТПУ) расположились именно в этой области, но мало информации о том, как это влияет на «Научно-исследовательскую активность» и «Востребованность работодателями».

ЛИТЕРАТУРА

1. Кабаков Р. И. R в действии. Анализ и визуализация данных в программе R. – М.: ДМК Пресс, 2016. – 588 с.
2. Рейтинг вузов России [Электронный ресурс]. – Режим доступа: <https://top100universities.ru>.
3. Айвазян А.С., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: Классификация и снижение размерности. – М.: Финансы и статистика, 1989. – 609 с.
4. Шипунов А. Б., Балдин Е. М., Волкова П. А. и др. Наглядная статистика. Используем R [Электронный ресурс]. – Режим доступа: <https://cran.r-project.org/doc/contrib/Shipunov-rbook.pdf>.

ИГОНИН К. К., ЯРОСЛАВЦЕВ А. В.
О СТАБИЛИЗАЦИИ ПЛАСТИНЫ, ЗАКРЕПЛЕННОЙ ШАРНИРАМИ
С ДВУХ СТОРОН, ПРИ СВЕРХЗВУКОВОМ ОБТЕКАНИИ

Аннотация. Рассматривается задача стабилизации динамики упругого элемента конструкции в виде пластины-полосы при одностороннем обтекании ее сверхзвуковым потоком идеального газа. Получена область асимптотической устойчивости пластины. Для случая неустойчивости предложено стабилизирующее управление. Приведены результаты численного моделирования.

Ключевые слова: уравнение в частных производных, упругий элемент, асимптотическая устойчивость, стабилизирующее управление, метод Галеркина.

IGONIN K. K., YAROSLAVTSEV A.V.
ON STABILIZATION OF PLATE FIXED BY HINGES
FROM TWO SIDES IN SUPERSONIC FLOW

Abstract. The problem of stabilizing the dynamics of an elastic element of construction in the shape of a plate-strip with its flowing by supersonic flow of ideal gas is considered. The domain of asymptotic stability of the plate is obtained. Stabilizing control is proposed for the case of instability. The results of the numerical simulation are presented.

Keywords: partial differential equation, elastic element, stability, stabilizing control, Galerkin method.

При проектировании конструкций с упругими элементами в случае, когда они могут подвергаться воздействию сверхзвукового потока газа, важное значение имеет поведение упругих элементов. Воздействие потока на составные части конструкций может приводить к возникновению неустойчивых колебаний и деформации упругих элементов, нарушению требуемых функциональных свойств и к их полному разрушению. В связи с этим возникает задача стабилизации конструкций с целью обеспечения надежности их эксплуатации и удовлетворения требуемых функциональных свойств даже во время экстремальных ситуаций.

Рассмотрим задачу управления динамикой упругого элемента конструкции в виде пластины-полосы при одностороннем обтекании ее сверхзвуковым потоком идеального газа в случае ее шарнирного закрепления на обоих концах. В работе [1] были получены достаточные условия устойчивости упругого элемента конструкции при обтекании его сверхзвуковым потоком газа. В работе [2] в виде неравенств сформулированы достаточные

условия асимптотической устойчивости и показано, что в случае двустороннего обтекания можно добиться выполнения этих условий с помощью подходящего выбора сжимающего продольного усилия, приложенного к пластине.

Исследуем, при какой длине пластины, в случае одностороннего обтекания ее сверхзвуковым потоком идеального газа, будет наблюдаться ее асимптотическая устойчивость. А для случая неустойчивости построим стабилизирующее управление.

Механическая система, описывающая динамику упругой стенки (защитного экрана) резервуара, заполненного жидкостью, при взаимодействии стенки со сверхзвуковым потоком газа представляет собой краевую задачу для дифференциального уравнения с частными производными [2]

$$m\ddot{w} + Dw'''' + Nw'' + \alpha\ddot{w}''' - \beta\ddot{w}'' + \gamma(\dot{w} + Vw') + f(x, t, w, \dot{w}) = 0, \quad x \in (0; l), \quad t \geq 0. \quad (1)$$

Здесь $w(x, t)$ – прогиб пластины; x – продольная координата, t – время, l – длина пластины; V – скорость потока; m – погонная масса пластины; D – изгибная жесткость пластины; N – сжимающее ($N > 0$) или растягивающее ($N < 0$) продольное усилие; α – коэффициент внутреннего демпфирования; β – коэффициент, учитывающий инерцию вращения сечений; $\gamma(\dot{w} + Vw')$ – аэродинамическая нагрузка, $\gamma = \alpha_0 \rho_0 a_0 = \text{const}$ ($\alpha_0 = 1$ при одностороннем обтекании, ρ_0 – плотность газа, a_0 – скорость звука); $f(x, t, w, \dot{w})$ – некоторое управляющее воздействие; штрих обозначает производную по координате x , точка – производную по времени t .

В качестве управляющего воздействия выберем

$$f(x, t, w, \dot{w}) = u_1(t) + u_2(t) \cos \frac{\pi x}{l}, \quad (2)$$

где $u_1(t)$, $u_2(t)$ – некоторые кусочно-непрерывные функции (управления).

Шарнирное закрепление на обоих концах пластины описывается краевыми условиями:

$$w(0, t) = w(l, t) = w''(0, t) = w''(l, t) = 0. \quad (3)$$

Будем искать решение задачи (1), (2), (3) методом Галеркина в виде [3]

$$w(x, t) = v_1(t) \sin\left(\frac{\pi x}{l}\right) + v_2(t) \sin\left(\frac{2\pi x}{l}\right).$$

В результате, после замены $y_1 = v_1(t)$, $y_2 = v_2(t)$, $y_3 = \dot{v}_1(t)$, $y_4 = \dot{v}_2(t)$ получим систему линейных дифференциальных уравнений с постоянными коэффициентами и приведем ее к нормальному виду:

$$\begin{cases} \dot{y}_1 = y_3 \\ \dot{y}_2 = y_4 \\ \dot{y}_3 = \frac{1}{(\pi^2 \beta + l^2 m)} \left(\frac{\pi^2 (Nl^2 - \pi^2 D)}{l^2} y_1 + \frac{8l\gamma V}{3} y_2 - \frac{\pi^2 \alpha + l^4 \gamma}{l^2} y_3 - \frac{4l^2}{\pi} u_1(t) \right) \\ \dot{y}_4 = \frac{1}{(4\pi^2 \beta + l^2 m)} \left(-\frac{8l\gamma V}{3} y_1 + \frac{4\pi^2 (Nl^2 - 4\pi^2 D)}{l^2} y_2 - \frac{16\pi^2 \alpha + l^4 \gamma}{l^2} y_4 - \frac{8l^2}{3\pi} u_2(t) \right) \end{cases} \quad (4)$$

Найдем область асимптотической устойчивости системы (4) в зависимости от длины пластины l при нулевых управляющих воздействиях $u_1(t)=0$, $u_2(t)=0$ путем пошагового просчета. Для этого, выбирая длину пластины от 1 до 30 метров с шагом 0.125 метра, находим собственные числа матрицы, составленной из коэффициентов системы (4), ищем решение уравнения (1) при краевых условиях (3) методом Галеркина – функцию прогиба $w(x,t)$ и строим ее график. При этом в одних случаях мы получаем, что вещественные части собственных чисел матрицы системы отрицательны, а значит решение системы асимптотически устойчиво. На графике функции прогиба мы можем наблюдать затухающие колебания пластины (рис. 1). В других случаях среди собственных чисел матрицы системы имеются собственные числа с положительными вещественными частями и, следовательно, решение неустойчиво. Тогда амплитуда колебаний на графике функции прогиба с течением времени неограниченно возрастает (рис. 2). В результате вычислений получена область асимптотической устойчивости $l \in [1; 5.125]$.

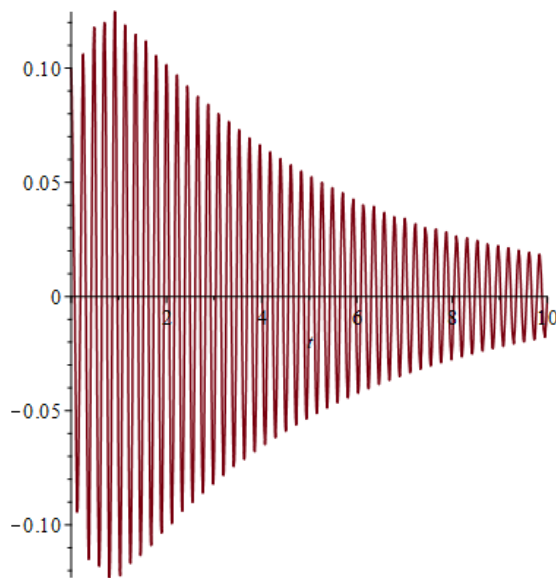


Рис. 1. Функция $w\left(\frac{l}{2}, t\right)$ при $l = 5.125$.

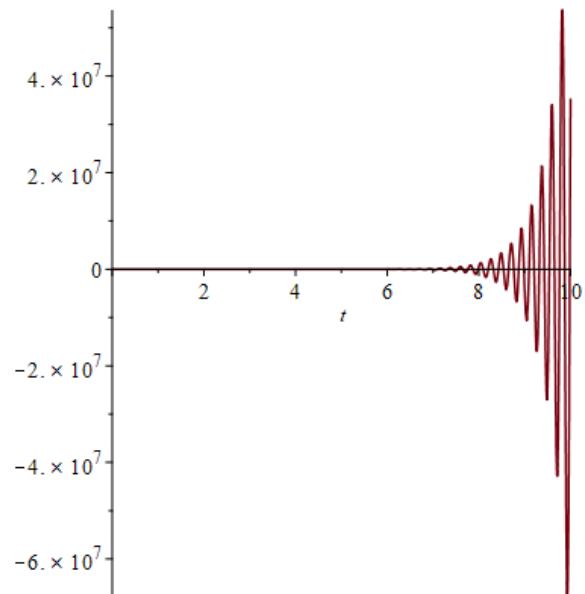


Рис. 2. Функция $w\left(\frac{l}{2}, t\right)$ при $l = 5.25$.

В случае, когда система (4) неустойчива, найдем управляющие воздействия, обеспечивающие ее стабилизацию [4]. По теореме об устойчивости решений линейной системы уравнений с постоянными коэффициентами [5], решения системы

$$\begin{cases} \dot{z}_1 = z_2(t) \\ \dot{z}_2 = p z_1(t) - z_2(t) \end{cases} \quad (5)$$

устойчивы при $p = 0$, асимптотически устойчивы при $p < 0$, неустойчивы при $p > 0$.

Приведем систему уравнений (4) к виду (5) дважды путем такого выбора управлений $u_1(t)$, $u_2(t)$, что при подстановке их в систему (4) эта система распадается на две системы, каждая из которых принимает вид (5). Для этого выберем управления в виде:

$$u_1(t) = \frac{\pi}{2l} \left(\left(\frac{\pi^2 N}{2l} - \frac{\pi^4 D}{2l^3} - p \right) y_1 + \frac{4}{3} \mathcal{W} y_2 \right), \quad u_2(t) = \frac{3\pi}{4l} \left(-\frac{4}{3} \mathcal{W} y_1 + \left(\frac{2\pi^2 N}{l} - \frac{8\pi^4 D}{l^3} - p \right) y_2 \right).$$

Выбирая величину параметра p по модулю больше или меньше, можем получить более быструю или более медленную стабилизацию решения краевой задачи (1), (2), (3). На рисунках 3 и 4 показаны графики функции прогиба для значений $p = -10$ и $p = -10000$ соответственно.

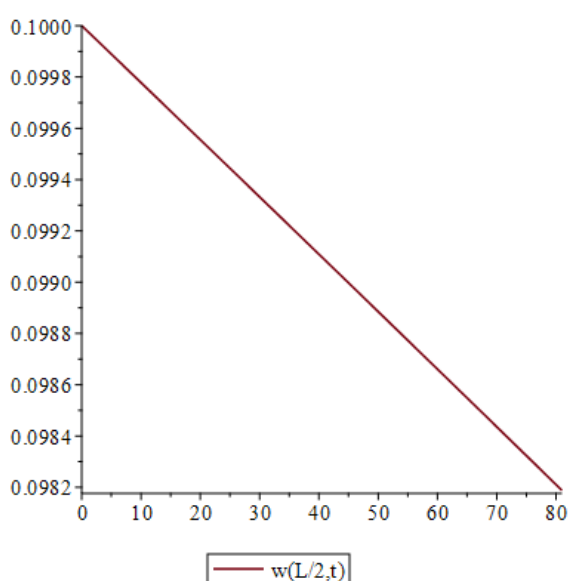


Рис. 3. Стабилизация при $p = -10$.

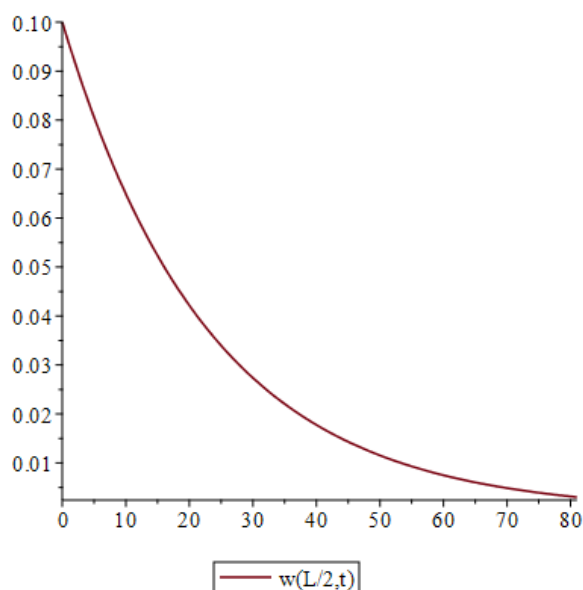


Рис. 4. Стабилизация при $p = -10000$.

При решении краевой задачи методом Галеркина с помощью математического пакета были выбраны следующие параметры системы:

$$m = 117.75 \text{ кг}; \quad \gamma = 420.175 \frac{\text{кг} \cdot \text{с}}{\text{м}^2}; \quad \alpha = 0.2; \quad \beta = 0.1; \quad D = 64086.9141 \text{ Па}; \quad N = 1000 \text{ Н}; \quad V = 400 \text{ м/с}.$$

ЛИТЕРАТУРА

1. Анкилов А. В., Вельмисов П. А., Судаков В. А. Об устойчивости решений начально-краевой задачи о динамике защитного экрана при взаимодействии со сверхзвуковым потоком газа. // Вестник Ульяновского государственного технического университета. – 2013. – № 3(63). – С. 45–52.
2. Анкилов А. В., Вельмисов П. А. Исследование динамики и устойчивости упругого элемента конструкции при сверхзвуковом обтекании. // Вестник Саратовского государственного технического университета. – 2011. – № 3(57), Вып. 1. – С. 59–67.
3. Флетчер К. Численные методы на основе метода Галеркина. – М.: Мир, 1988. – 352 с.
4. Вельмисов П. А., Гладун А. В. Об управлении динамикой трубопровода // Журнал Средневолжского математического общества. – 2016. – Т. 18, № 4. – С. 89–97.
5. Еругин Н. П., Штокайло И. З. и др. Курс обыкновенных дифференциальных уравнений: учебное пособие. – Киев: Вища школа, 1974. – 472 с.

НАЗАРОВ В. И., ЯЗОВЦЕВА О. С.
ЧИСЛЕННЫЙ МЕТОД УСТАНОВЛЕНИЯ СООТВЕТСТВИЯ
МЕЖДУ НАЧАЛЬНЫМИ ТОЧКАМИ

ЛОКАЛЬНО АСИМПТОТИЧЕСКИ ЭКВИВАЛЕНТНЫХ СИСТЕМ

Аннотация. В статье приводится методика расчета начальных точек системы обыкновенных дифференциальных уравнений через начальные точки локально асимптотически эквивалентной системы. Численное решение нелинейной системы было найдено с использованием (4,2)-метода. Численный расчет интегрального отображения, устанавливающего соответствие между начальными значениями, был проведен при помощи метода Симпсона. Рассчитанное отображение обеспечивает локальную асимптотическую эквивалентность систем, что дает возможность исследования решения линейного приближения вместо нелинейной системы.

Ключевые слова: нелинейные обыкновенные дифференциальные уравнения, локальная покомпонентная асимптотическая эквивалентность, пиролиз этана, химическая кинетика, (4,2)-метод, метод Симпсона.

NAZAROV V. I., YAZOVTSEVA O. S.
NUMERICAL METHOD OF MAPPING BETWEEN INITIAL POINTS
OF LOCALLY ASYMPTOTIC EQUIVALENT SYSTEMS

Abstract. The article presents the method of calculating the initial points of a system of ordinary differential equations by the initial points of a locally asymptotic equivalent system. The numerical solution of the nonlinear system was found using the (4,2)-method. The numerical calculation of the integral mapping between the initial values was carried out using the Simpson method. The calculated mapping provides locally asymptotic equivalence of the systems, which makes it possible to seek the solutions of the linear approximation instead of the nonlinear system.

Keywords: nonlinear ordinary differential equations, locally component-wise asymptotic equivalence, ethane pyrolysis, chemical kinetics, (4,2)-method, Simpson method.

При проведении численного моделирования химических процессов зачастую приходится сталкиваться с проблемой решения жестких по части компонент нелинейных систем обыкновенных дифференциальных уравнений. Решение задачи Коши для таких систем требует применения особых методов, но даже использование специализированных методик не дает приемлемого сокращения временных затрат при проведении численного эксперимента. В данном случае удобно использовать систему, локально асимптотически эквивалентную исследуемой. В качестве такой системы можно использовать линейное

приближение. Такой подход изложен в работах [1-4], где приведены достаточные условия асимптотической эквивалентности для систем обыкновенных дифференциальных уравнений.

Рассмотрим нелинейную систему обыкновенных дифференциальных уравнений

$$\begin{cases} \dot{x}_1 = -k_1 x_1 - 2k_2 x_1^2 \\ \dot{x}_2 = k_1 x_1 + k_2 x_1^2 \\ \dot{x}_3 = k_1 x_1 \\ \dot{x}_4 = 2k_2 x_1^2 \end{cases} \quad (1)$$

Здесь $t \geq 0$; x_i – концентрации веществ C_2H_6 , C_2H_4 , H_2 , CH_4 соответственно, $x_i \geq 0, (i = \overline{1,4})$; $k_1 > 0, k_2 > 0$ – константы скоростей химических реакций. Система (1) соответствует кинетической модели брутто-реакции пиролиза этана [3, 5, 6]. Рассмотрим задачу Коши с начальными данными $x_1(0) = 1, x_2(0) = 0, x_3(0) = 0, x_4(0) = 0$ для системы (1).

В векторной форме данная система имеет вид:

$$\dot{x} = Ax + P(x),$$

где $x = colon(x_1, x_2, x_3, x_4)$.

$$A = \begin{pmatrix} -k_1 & 0 & 0 & 0 \\ k_1 & 0 & 0 & 0 \\ k_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad P(x) = \begin{pmatrix} -2k_2 x_1^2 \\ k_2 x_1^2 \\ 0 \\ 2k_2 x_1^2 \end{pmatrix}.$$

Ее линейное приближение имеет вид:

$$\begin{cases} \dot{y}_1 = -k_1 y_1 \\ \dot{y}_2 = k_1 y_1 \\ \dot{y}_3 = k_1 y_1 \\ \dot{y}_4 = 0 \end{cases} \quad (2)$$

Локальная покомпонентная асимптотическая эквивалентность для систем (1) и (2) показана в [3].

Ставится задача о нахождении начальных данных для системы (2) через известные начальные данные системы (1).

В работе [4] построено отображение, устанавливающее соотношение между начальными точками исследуемой системы и ее линейного приближения:

$$y^{(0)} = x^{(0)} + \int_0^{+\infty} Y(-s)P(x(s))ds, \quad (3)$$

где $x(t) = x(t; 0, x^{(0)})$, $y(t) = Y(t)y^{(0)}$, $Y(t)$ – фундаментальная матрица линейного приближения.

Для системы (2) она имеет вид:

$$Y(t) = \begin{pmatrix} e^{-k_1 t} & 0 & 0 & 0 \\ 1 - e^{-k_1 t} & 1 & 0 & 0 \\ 1 - e^{-k_1 t} & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Отображение (3) в координатной форме имеет следующий вид:

$$y_i^{(0)} = x_i^{(0)} + \sum_{j=1}^n \int_0^{+\infty} y_{ij}(-s) P_j(x(s)) ds, i = \overline{1,4}, \quad (4)$$

где $y_{ij}(t)$ – элементы фундаментальной матрицы $Y(t)$ системы (2).

Представим отображение (3) в виде

$$y_i^{(0)} = x_i^{(0)} + \sum_{j=1}^n \int_0^b y_{ij}(-s) P_j(x(s)) ds + \sum_{j=1}^n \int_b^{+\infty} y_{ij}(-s) P_j(x(s)) ds,$$

где b определяется из условия [4]:

$$b > \frac{1}{k_1} \ln \left(\frac{2k_2 c_0}{k_1} \right),$$

при этом c_0 из формулы (3.8) из работы [4].

Таким образом,

$$\tilde{y}_i^{(0)} = x_i^{(0)} + \sum_{j=1}^n \int_0^b y_{ij}(-s) P_j(x(s)) ds. \quad (5)$$

Для численного решения системы (1) применялся (4,2)-метод [7; 8], для вычисления интеграла из выражения (4) использовался метод Симпсона [9; 10].

Расчет начальных точек для системы линейного приближения проведен для различных значений констант скоростей химических реакций k_1 и k_2 , которые соответствуют протеканию реакции пиролиза этана при различных температурах и рассчитаны из уравнения Аррениуса [5]. Результаты, рассчитанные с точностью $\varepsilon = 0.001$, представлены в таблице.

Таблица

Расчет начальных точек для решений системы линейного приближения

T, K	800	900	1000	1200
k_1	0.503	32.832	928.979	139822
k_2	0.073	6.625	244.927	55056.8
$\tilde{y}_1^0$	0.777	0.875	0.790	0.989
$\tilde{y}_2^0$	0.169	0.091	0.150	0.005
$\tilde{y}_3^0$	0.115	0.058	0.090	0.000
$\tilde{y}_4^0$	0.121	0.069	0.118	0.010

На рисунке приведены графики решений систем (1) и (2), между начальными значениями которых установлено соотношение в соответствии с формулой (4), при значениях k_1 и k_2 , соответствующих температуре $T = 1000$ К.

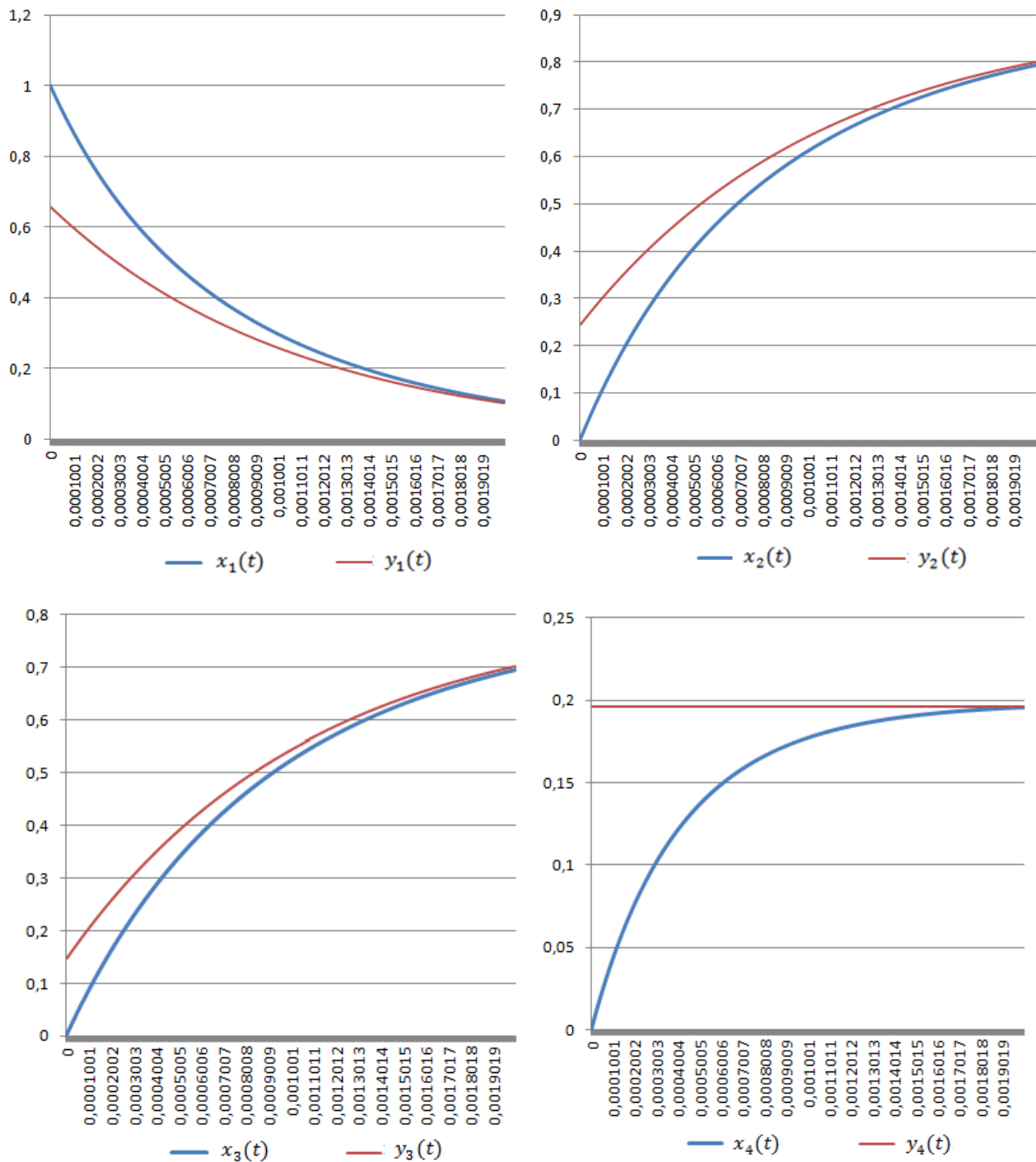


Рис. Графики решений $x_i(t)$ и $y_i(t)$, между начальными значениями которых установлено взаимно-однозначное соответствие.

Таким образом, по начальным точкам нелинейной системы обыкновенных дифференциальных уравнений были рассчитаны начальные точки ее линейного приближения, которые обеспечивают локальную покомпонентную асимптотическую эквивалентность систем согласно формуле (4).

Из численного эксперимента видно, что поведение решений системы (1), начиная с момента времени равного 0,002 с, мало отличается от поведения решений системы (2). Это обуславливается достаточно большими значениями констант скоростей химических реакций при температуре 1000 К, а, следовательно, и высокой скоростью протекания стадий химического превращения.

Данная методика позволяет при проведении расчетов использовать вместо численного решения жесткой системы аналитическое решение ее линейного приближения, что значительно сокращает требования к вычислительным мощностям и повышает точность расчетов.

ЛИТЕРАТУРА

1. Воскресенский Е. В. Методы сравнения в нелинейном анализе. – Саратов: Изд-во Саратовского ун-та, 1990. – 224 с.
2. Воскресенский Е. В. Асимптотические методы: теория и приложения: монография. – Саранск, 2000. – 300 с.
3. Язовцева О. С. Локальная покомпонентная асимптотическая эквивалентность и ее применение к исследованию устойчивости по части переменных [Электронный ресурс] // Огарев-online. – 2017. – № 13. – Режим доступа: <http://journal.mrsu.ru/arts/lokalnaya-pokomponentnaya-asimptoticheskaya-ekvivalentnost-i-ee-primenenie-k-issledovaniyu-ustojchivosti-po-chasti-peremennykh>.
4. Шаманаев П. А., Язовцева О. С. Достаточные условия локальной покомпонентной асимптотической эквивалентности нелинейных систем обыкновенных дифференциальных уравнений и ее приложение к устойчивости по части переменных // Журнал Средневолжского математического общества. – 2017. – Т. 19, № 1. – С. 102–115.
5. Губайдуллин И. М., Пескова Е. Е., Язовцева О. С. Математическая модель динамики многокомпонентного газа на примере брутто-реакции пиролиза этана [Электронный ресурс] // Огарев-online. – 2016. – № 20. – Режим доступа: <http://journal.mrsu.ru/arts/matematiceskaya-model-dinamiki-mnogokomponentnogo-gaza-na-primere-brutto-reakcii-piroliza-etana>.

6. Жалнин Р. В., Пескова Е. Е., Стадниченко О. А., Тишкин В. Ф. Математическое моделирование динамики многокомпонентного газа с использованием WENO схем на примере пиролиза этана // Журнал Средневолжского математического общества. – 2016. – Т. 18, № 3. – С. 98–106.
7. Галанин М. П., Ходжаева С. Р. Методы решения жестких обыкновенных дифференциальных уравнений. Результаты тестовых расчетов // Препринты ИПМ им. М. В. Келдыша. – 2013. – № 98. – 29 с.
8. Назаров В. И., Пескова Е. Е., Язовцева О. С. Численное моделирование жестких систем с использованием (4,2)-метода [Электронный ресурс] // Огарев-online. – 2017. – № 13. – Режим доступа: <http://journal.mrsu.ru/arts/chislennoe-modelirovanie-zhestkix-sistem-s-ispolzovaniem-42-metoda>.
9. Самарский А. А., Гулин А. В. Численные методы: учеб. пособие для вузов. – М.: Наука, 1989. – 432 с.
10. Шаманаев П. А. Ляпуновские преобразования и устойчивость движения: автореф. дис. ... канд. физ.-мат. наук. – Саранск, 1997. – 16 с.

ЛЕБЕДЕВ М. А.

**МИНИМИЗАЦИЯ ЭНЕРГОПОТРЕБЛЕНИЯ, СВЯЗАННОГО С ОПРЕДЕЛЕНИЕМ
МЕСТОПОЛОЖЕНИЯ В МОБИЛЬНОМ ПРИЛОЖЕНИИ
ДЛЯ УДАЛЕННОГО МОНИТОРИНГА СОБЫТИЙ УСТРОЙСТВ
НА ОПЕРАЦИОННОЙ СИСТЕМЕ ANDROID**

Аннотация. В статье представлен анализ мобильных приложений для удаленного мониторинга событий устройств на операционной системе Android. Выявляется их главный недостаток, связанный с энергопотреблением. Предлагается решение, позволяющее минимизировать энергопотребление, связанное с определением местоположения в мобильном приложении для удаленного мониторинга событий устройств на операционной системе Android.

Ключевые слова: Android, мобильное приложение, местоположение, энергопотребление, мониторинг событий.

LEBEDEV M. A.

**MINIMIZATION OF POWER CONSUMPTION RELATED TO LOCATION TRACKING
IN MOBILE APPLICATION FOR REMOTE MONITORING
OF DEVICE EVENTS ON OPERATING SYSTEM ANDROID**

Abstract. The article presents an analysis of mobile applications for remote monitoring of device events on the operating system Android. Their main drawback associated with power consumption is revealed. A solution is proposed that minimizes the power consumption related to location tracking in a mobile application for remote monitoring of device events on the operating system Android.

Keywords: Android, mobile application, location, power consumption, event monitoring.

Одна из полезных функций мобильного устройства – это возможность определять местоположение его владельца. Приложения, которые используют эту функцию, нередко используются для контроля сотрудников служб, связанных с доставкой. Также существует много программ нацеленных на так называемый «родительский контроль», который позволяет членам семьи поддерживать контакт, узнавать местоположение друг друга и осуществлять контроль за своими детьми.

В сравнении с другими системами слежения, приложения мобильного мониторинга имеют ряд заметных преимуществ:

- низкая стоимость внедрения по сравнению с другими GPS-системами;
- высокая информативность;

- использование большего числа источников для определения местоположения (вышки сотовой связи, Wi-Fi точки, навигационные спутники);
- широкий спектр дополнительных возможностей.

В настоящее время существует немало мобильных приложений, которые дают возможность получать информацию о положении целевого устройства в заданный момент времени и решают задачи, связанные с контролем местоположения сотрудников. В ходе исследования приложений для операционной системы Android, реализующих такой функционал, были выявлены схожие проблемы. Главная проблема – серьезный расход заряда батареи смартфона в процессе работы приложения.

Проблема с сильным расходом батареи обусловлена, в первую очередь, непрерывным использованием соответствующих датчиков, нужных для определения геопозиции устройства, а также перманентной отправкой этих данных на сервер.

В данной статье будет разработан и описан один из возможных алгоритмов, позволяющих минимизировать энергопотребление, связанное с определением местоположения в мобильном приложении для удаленного мониторинга событий устройств на операционной системе Android.

Для начала определимся с возможностями системы и составим некоторые требования, предъявляемые к ней. Система будет ориентирована на работу с выездными сотрудниками какого-либо предприятия, т.е. взаимодействие будет происходить между диспетчером и рабочим данного предприятия. Несмотря на это, пользоваться обе взаимодействующие стороны будут одним приложением, функционал которого будет меняться в зависимости от типа аккаунта пользователя. Аккаунт диспетчера будет выступать в роли агрегатора аккаунтов сотрудников, т.е. существование сотрудника в системе независимо от диспетчера невозможно. Таким образом, система должна состоять из следующих частей: клиентское приложение с автоматической сменой возможностей в зависимости от типа аккаунта пользователя (диспетчер или сотрудник), бекенд системы.

На основе проведенного анализа и предполагаемого сценария использования приложения были составлены следующие требования к разрабатываемой системе.

1. Мобильное приложение должно позволять получать координаты, с помощью GPS, GSM-сети и Wi-Fi.
2. Мобильное приложение должно иметь режим экономии энергии.
3. Система должна позволять синхронизировать данные между мобильным приложением и сервером.
4. Система должна иметь функционал для работы оффлайн.

5. Система должна поддерживать возможность обмена сообщениями между сотрудником и диспетчером.
6. В системе должна быть поддержка PUSH-уведомлений на различные действия.
7. В системе должен присутствовать функционал создания и просмотра заданий.
8. В системе должен присутствовать функционал смены статуса сотрудника.
9. Мобильное приложение должно автоматически запускаться после старта системы.
10. Функционал мобильного приложения должен зависеть от типа авторизованного аккаунта пользователя (сотрудник или диспетчер)
11. Мобильное приложение должно иметь адаптированный для смартфонов интерфейс.

Для решения проблемы энергосбережения стоит разобраться, какие способы получения местоположения доступны в операционной системе Android.

В Android есть 3 провайдера (поставщика, источника) данных о местоположении.

GPS (GPS, AGPS): провайдер местоположения посредством GPS. Этот провайдер определяет местоположение с использованием одноименных спутников. В зависимости от условий ему может потребоваться некоторое время, чтобы вернуть обновленное местоположение. Оказывает сильное влияние на заряд батареи устройства.

Network (AGPS, CellID, WiFi MACID): поставщик местоположения, который использует сеть. Он определяет местоположение на основе информации, полученной от доступных точек сотовой связи и точек доступа Wi-Fi. Имеет среднее воздействие на заряд батареи устройства.

Passive (CellID, WiFi MACID): специальный провайдер местоположения для приема местоположений без фактического инициирования его обновления. Этот провайдер может использоваться для пассивного получения обновлений данных о локации. То есть информация, которую он использует, будет обновляться в момент, когда другие приложения или службы запрашивают их. Таким образом с помощью него мы получаем только закешированные системой обращения к GEO-данным от других приложений. Это могут быть как обращения Network (Wi-Fi) в активном и пассивном режиме, мониторинг вышек сотовой связи, так и использование GPS. Оказывает минимальное воздействие на заряд батареи устройства [2].

Почти все приложения, использующие в своей работе данные о локации, используют один из описанных выше провайдеров локации для определения местоположения, другие используют свои закрытые реализации. В доступных режимах экономии энергии, которые предоставляют исследованные приложения, суть заключается в выборе одного из

поставщиков локаций, что позволяет добиться либо максимальной точности, либо минимально возможной нагрузки на батарею. В последнем случае экономия энергии происходит за счет снижения точности. Поэтому решение проблемы с высоким энергопотреблением кроется не в выборе оптимального поставщика местоположения, а в том, как с ним построено взаимодействие нашего приложения. Но, несмотря на это, в нашем приложении будет присутствовать и стандартная настройка выбора поставщика локаций.

Заметим, что обновление информации о географических координатах в базе данных будет происходить в моменты получения этих данных от датчиков устройства. Поэтому далее мы будем говорить только об обновлении данных поставщиками локаций, а отправка этих данных на сервер будет подразумеваться.

Первый способ уменьшения энергозатрат, который напрашивается для реализации, это установка некоторого интервала обновления данных о локациях устройства. Скорее всего, в проанализированных приложениях такой интервал равен некой константе. Соответственно, чем длиннее будет такой интервал, тем меньшее потребление заряда аккумулятора мы получим. Точность при любых интервалах останется на прежнем уровне и будет зависеть от выбранного поставщика. Минус заключается в том, что, исходя из специфики нашего приложения, диспетчеру может потребоваться увидеть местоположение сотрудника в случайный момент времени, а не дожидаться, когда данные придут на сервер.

Отсюда вытекает идея сделать такой режим работы приложения, при котором бы диспетчер получал информацию о местонахождении сотрудника только в моменты, когда такая информация ему потребуется, а именно, когда он будет наблюдать на экране своего смартфона за передвижением конкретного сотрудника. Соответственно, данные о местоположении сотрудника будут обновляться только в такие моменты времени, что позволит минимизировать энергозатраты.

Для того, чтобы диспетчер в режиме реального времени получал информацию о том, что сотрудник прибыл на точку выполнения задания, будем в этот момент времени обновлять географические координаты сотрудника, а также посылать уведомление диспетчеру.

Эти особенности будут реализованы только для режима энергосбережения, всегда будет возможность переключиться в обычный режим работы приложения, в котором обновление информации о местоположении будет происходить непрерывно. К тому же, логично, что данные будут обновляться и те моменты, когда сотрудник наблюдает свое местоположение на экране своего смартфона.

Исходя из сказанного, в приложении будет реализовано два режима работы.

1. Обычный режим, в котором данные о местоположении будут обновляться непрерывно.

2. Режим экономии энергии, который будет реализовывать описанные возможности.

Первым шагом реализации алгоритма, решающего проблемы энергосбережения, будет подбор подходящей под наши нужды базы данных, которая бы могла принимать множественные запросы на запись и чтение геоданных. Также требуется реализовать бекенд системы, который бы связывал мобильные клиенты диспетчера и сотрудника с базой данных. Системой, решающей обе задачи, была выбрана NoSQL база данных Firebase от компании Google. Вместо обычных HTTP-запросов она использует синхронизацию данных. Каждый раз, когда данные изменяются, любое подключенное устройство получает это обновление в течении миллисекунд. Данные в базе хранятся в формате JSON. Общение между базой данных Firebase и мобильным клиентом производится путем подключения пользователей к узлам баз данных. Таким образом, любое изменение в некотором узле базы, связанным с данными о местоположении сотрудника, будет немедленно отправлено на все устройства, которые в этот момент держат с ним (узлом) соединение.

Для того, чтобы мобильное приложение всегда оставалось отзывчивым и смогло бы получать и отправлять данные о локации требуется создание некоторой службы, которая будет существовать независимо от того, запущено приложение или нет. Для этого идеально подойдет Service, один из стандартных компонентов, предоставляемый SDK Android. Service – компонент приложения, который может выполнять длительные операции в фоновом режиме и не содержит пользовательского интерфейса. Именно в нем и будут реализованы методы взаимодействия с нашей базой данных.

Для оповещения об изменении данных в базе Firebase использует механизм обратных вызовов. Большие объемы данных и возможные тяжелые операции, связанные с ними, не рекомендуется выполнять в основном потоке Android-приложения, так как этот поток отвечает в том числе за отрисовку пользовательского интерфейса, который может зависнуть в ходе долгих операций на нем. Для корректной работы с потоками приложения будем использовать библиотеку RxJava 2, которая реализует принципы реактивного программирования для JVM. События Firebase, поступающие в методы обратных вызовов, будем транслировать в потоки данных и обрабатывать с помощью реализации паттерна «Наблюдатель». В нем есть объект Observable, который генерирует события. В нашем случае база данных будет генерировать события изменения своих полей, и Observer, который подписывается и получает эти события, вызывая методы для дальнейшей обработки данных, в том числе для вывода в реальном времени геоположения сотрудника на экране устройства диспетчера.

Подписка на события обновления локации сотрудника будет полностью связана с жизненным циклом приложения для операционной системы Android и активна только в тот период времени, когда диспетчер наблюдает за этим сотрудником на экране своего смартфона. Соответственно, получение информации от поставщиков местоположения на устройстве сотрудника будет происходить только в эти моменты, что и будет являться главным фактором уменьшения времени интенсивной нагрузки на батарею устройства-источника локации. В остальные моменты времени подписка будет неактивна, и сервис, запущенный в приложении сотрудника, будет находиться в режиме ожидания. Для определения моментов наблюдения диспетчером конкретных сотрудников, в узле базы данных, связанным с учетной записью диспетчера будет существовать специальное поле, хранящее уникальный идентификатор этого сотрудника. Сигналом к подписке на обновления определенного устройства будет являться изменение именно этого поля.

Таким образом, мы уменьшили расход энергии мобильного устройства, сведя к минимуму временной интервал взаимодействия нашего приложения с поставщиками локации, необходимый для нормального функционирования системы для удаленного мониторинга событий устройств на операционной системе Android.

ЛИТЕРАТУРА

1. Харди Б., Филлипс Б. Android. Программирование для профессионалов. – 2-е изд. – СПб.: Питер, 2016. – 640 с.
2. Android Location Providers – gps, network, passive – Tutorial [Электронный ресурс]. – Режим доступа: <https://developerlife.com/2010/10/20/gps>.

СТЕНИН И. В., ШАМАНАЕВ П. А.

ИДЕНТИФИКАЦИЯ ПАРАМЕТРОВ ДИНАМИЧЕСКОЙ СИСТЕМЫ
ВТОРОГО ПОРЯДКА ПО ЭКСПЕРИМЕНТАЛЬНЫМ ДАННЫМ

Аннотация. В статье описан подход к решению задачи идентификации параметров динамических систем второго порядка по экспериментальным данным. Поставленная задача решается путем сведения ее к задаче минимизации квадратичного функционала с ограничениями в виде нелинейных алгебраических уравнений.

Ключевые слова: идентификация параметров, обыкновенные дифференциальные уравнения, условная минимизация.

STENIN I. V., SHAMANAEV P. A.

IDENTIFICATION OF SECOND-ORDER DYNAMIC SYSTEM
PARAMETERS ON EXPERIMENTAL DATA

Abstract. The approach to solving the problem of identifying the parameters of second-order dynamical systems on experimental data is described by reducing it to the problem of minimizing a quadratic functional with constraints in the form of nonlinear algebraic equations.

Keywords: parameter identification, ordinary differential equations, constraint minimization.

Рассмотрим систему линейных обыкновенных дифференциальных уравнений:

$$\begin{cases} \frac{dx_1}{dt} = \theta_1 x_1 + \theta_2 x_2 \\ \frac{dx_2}{dt} = \theta_3 x_1 + \theta_4 x_2 \end{cases} \quad (1)$$

где $x_i \in \mathbb{R}, i = 1, 2$ – зависимые переменные, $t \in [0, b]$ – независимая переменная, $b > 0$, $\theta_k \in \mathbb{R}, k = 1, \dots, 4$ – неизвестные параметры.

Обозначим через $x_j(t, \theta)$ j -компоненту решения системы (1), $j = 1, 2$, зависящую от векторного параметра $\theta = column(\theta_1, \theta_2, \theta_3, \theta_4)$.

Пусть при некоторых фиксированных значениях θ_k решение системы (1) удовлетворяет задаче Коши с начальными условиями:

$$x_1^{(1)} = x_1(0), x_2^{(1)} = x_2(0). \quad (2)$$

Пусть на равномерной сетке по переменной x с шагом $\tau = \frac{b}{N}$

$$t_1 = 0, \dots, t_{i+1} = t_i + \tau, \dots, t_N = b. \quad (3)$$

для экспериментальных данных справедливы соотношения

$$\tilde{x}^{(i)} = x^{(i)} + \varepsilon^{(i)}, \quad i = 1, \dots, N, \quad (4)$$

где $\tilde{x}^{(i)} = \text{column}(\tilde{x}_1^{(i)}, \tilde{x}_2^{(i)})$, $x^{(i)} = \text{column}(x_1^{(i)}, x_2^{(i)})$, $x_j^{(i)} = x_j(t_i, \theta)$ – значение компоненты решения системы (1) в точке t_i при фиксированном значении векторного параметра θ , $\varepsilon^{(i)} = \text{column}(\varepsilon_{i1}, \dots, \varepsilon_{iN})$ – вектор, элементы которого являются случайными величинами, имеющими стандартное нормальное распределение с параметрами 0 и 1, то есть $\varepsilon_{ij} \in N(0, 1)$, $i = \overline{1, N}$, $j = 1, 2$.

Ставится задача идентификации параметров системы линейных обыкновенных дифференциальных уравнений вида (1), заключающаяся в нахождении таких оценок параметров θ_k , $k = 1, \dots, 4$, при которых решение задачи (1)-(2) приближается к экспериментальным данным $\{\tilde{x}^{(i)}, i = \overline{1, N}\}$ в смысле метода наименьших квадратов.

Согласно [4], эта задача может быть сформулирована следующим образом

$$\min_{\theta} \sum_{i=1}^N [\tilde{x}^{(i)} - x^{(i)}]^T [\tilde{x}^{(i)} - x^{(i)}], \quad (5)$$

$$T_1 x^{(1)} = h,$$

при условии, что $x^{(i)} = x(t_i, \theta)$ – значение решения системы (1) в точке t_i . Здесь $T_1 = I_2$ – единичная (2×2) – матрица, $h = \text{column}(h_1, h_2)$, $h_1 = x_1^{(1)}$, $h_2 = x_2^{(1)}$.

Вводя обозначения [3]

$$z = \text{column}(x^{(N)}, \theta), \quad x^{(N)} = \text{column}(x^{(1)}, \dots, x^{(N)}),$$

$$\tilde{z} = \text{column}(\tilde{x}^{(N)}, o_4), \quad \tilde{x}^{(N)} = \text{column}(\tilde{x}^{(1)}, \dots, \tilde{x}^{(N)}),$$

$$\theta = \text{column}(\theta_1, \theta_2, \theta_3, \theta_4),$$

$$o_4 \text{ – нулевой вектор размерности } 4,$$

получим

$$x^{(N)} = H_1 z, \quad \tilde{x}^{(N)} = H_1 \tilde{z}, \quad (6)$$

где

$$H_1 = [I_{2N} : O_{2N \times 4}], \quad H_2 = \frac{1}{N} H_1^T H_1, \quad (7)$$

здесь I_{2N} – единичная $(2N \times 2N)$ -матрица, $O_{2N \times 4}$ – нулевая $(2N \times 4)$ -матрица.

Учитывая (6) и (7), целевая функция в соотношении (5) может быть представлена в виде

$$\sum_{i=1}^N [\tilde{x}^{(i)} - x^{(i)}]^T [\tilde{x}^{(i)} - x^{(i)}] = \frac{1}{2} [(H_2 z, z) - 2(H_2 \tilde{z}, z) + (H_2 \tilde{z}, \tilde{z})]. \quad (8)$$

Заменим уравнение (1) симметричной разностной схемой [1] на сетке (3)

$$\frac{x_1^{(i+1)} - x_1^{(i)}}{\tau} = \frac{1}{2} (\theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \theta_1 x_1^{(i+1)} + \theta_2 x_2^{(i+1)}),$$

$$\frac{x_2^{(i+1)} - x_2^{(i)}}{\tau} = \frac{1}{2} \left(\theta_3 x_1^{(i)} + \theta_4 x_2^{(i)} + \theta_3 x_1^{(i+1)} + \theta_4 x_2^{(i+1)} \right), \quad (9)$$

$$i = \overline{1, N}.$$

С учетом обозначений

$$g_{i1}(z) \equiv g_{i1}(x^{(N)}, \theta) = \left(1 + \frac{\tau}{2} \theta_1\right) x_1^{(i)} + \frac{\tau}{2} \theta_2 x_2^{(i)} - \left(1 - \frac{\tau}{2} \theta_1\right) x_1^{(i+1)} + \frac{\tau}{2} \theta_2 x_2^{(i+1)},$$

$$g_{i2}(z) \equiv g_{i2}(x^{(N)}, \theta) = \frac{\tau}{2} \theta_3 x_1^{(i)} + \left(1 + \frac{\tau}{2} \theta_4\right) x_2^{(i)} + \frac{\tau}{2} \theta_3 x_1^{(i+1)} - \left(1 - \frac{\tau}{2} \theta_4\right) x_2^{(i+1)},$$

$$g_i(z) = \text{column}(g_{i1}(z), g_{i2}(z)), i = \overline{1, N},$$

$$g(z) = \text{column}(g_1(z), \dots, g_N(z))$$

разностная схема (9) принимает вид

$$g(z) = 0. \quad (10)$$

Тогда задача (5) может быть сформулирована как задача минимизации квадратичного функционала с ограничениями в виде нелинейных алгебраических уравнений [3]

$$\min_z m(z), \quad m(z) = \frac{1}{2} (H_2 z, z) - (H_2 \tilde{z}, z),$$

$$T(z) = h, \quad (11)$$

$$g(z) = 0,$$

где $T = [T_1 \ T_2, \dots, T_N, T_\theta] - (2 \times (2N + 4))$ -матрица, $T_i, i = \overline{2, N}$ – нулевые (2×2) -матрицы, $T_\theta - (2 \times 4)$ -матрица.

Задачу (11) аппроксимируем последовательностью квадратичных задач минимизации с линейными ограничениями [3]. Алгоритм решения задачи (1) имеет вид:

1. Задаем точность вычислений $\delta > 0$.
2. Задаем h, b, N и τ .
3. Формируем экспериментальные данные $\tilde{x}^{(i)}, i = \overline{1, N}$.
4. Полагаем $r = 0$ и задаем начальные данные $z^{(0)} = \text{column}(x^{(N)}, \theta^{(0)}), \lambda^{(0)}, \mu^{(0)}$.
5. Вычисляем

$$M^{(r)} = \begin{pmatrix} H & T^T & A^T \\ T & O & O \\ A & O & O \end{pmatrix}, \quad d^{(r)} = \begin{pmatrix} -H_2(z^{(r)} - \tilde{z}) \\ h - Tz^{(r)} \\ -g(z^{(r)}) \end{pmatrix}.$$

6. Решаем систему линейных алгебраических уравнений

$$M^{(r)} y^{(r)} = d^{(r)}, \quad y^{(r)} = \text{column}(s^{(r)}, \mu^{(r)}, \lambda^{(r)}). \quad (12)$$

7. $z^{(r+1)} = z^{(r)} + s^{(r)}$.

8. Если $\|H_2(z^{(r)} - \tilde{z}) + A^T \lambda^{(r)} + T^T \mu^{(r)}\| \leq \delta$, то алгоритм завершен, иначе $r = r + 1$ и переход к пункту 5.

Замечание 1. Для решения системы вида (12) использовался подход, основанный на методе сопряженных градиентов и описанный в работе [2]. В этой же работе приведены структуры матриц A и H .

Вычисления проводились на трех наборах экспериментальных данных $\tilde{x}^{(i)}$, $i = \overline{1, N}$, $\varepsilon_{ij} \in N(0, 1)$, $j = 1, 2$, которые соответствовали трем типам положения равновесия системы (1): седло, узел и центр. Точность вычислений δ полагалась равной 0.001.

Приведем графики компонент $x_1(t, \hat{\theta})$, $x_2(t, \hat{\theta})$ решения с начальными данными $x_1(0) = 1$, $x_2(0) = 1$ системы (1), оценки параметров которой найдены согласно вышеизложенному алгоритму. Здесь $\hat{\theta} = column(\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \hat{\theta}_4)$ – оценка векторного параметра θ (см. рис.1 – 9).

Случай 1. Положение равновесия системы (1) – седло.

Получены следующие оценки параметров $\theta_1, \theta_2, \theta_3, \theta_4$:

$$\hat{\theta}_1 = 1.00007, \hat{\theta}_2 = -0.00028, \hat{\theta}_3 = 0.00039, \hat{\theta}_4 = -1.99924.$$

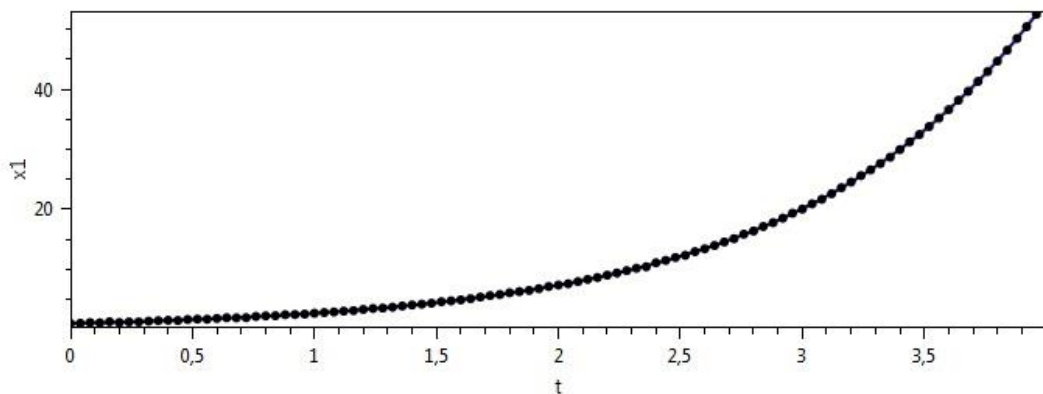


Рис. 1. Экспериментальные данные
и график компоненты $x_1(t, \hat{\theta})$ решения системы (1).

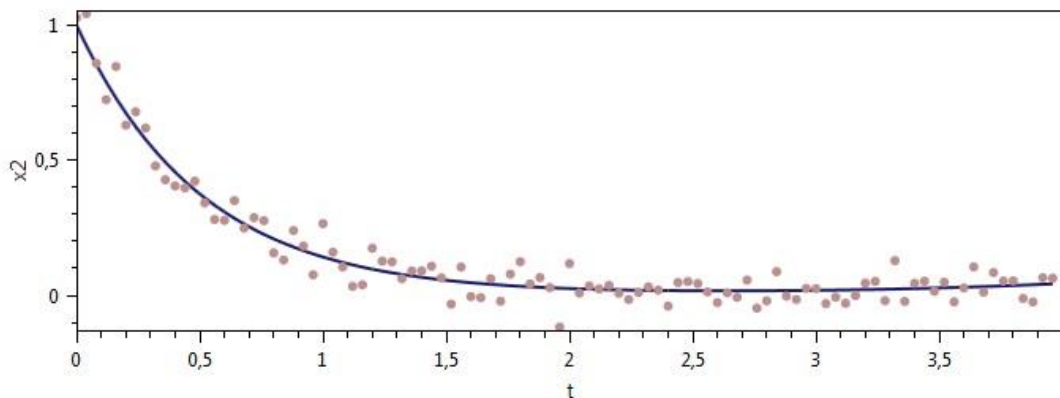


Рис. 2. Экспериментальные данные
и график компоненты $x_2(t, \hat{\theta})$ решения системы (1).

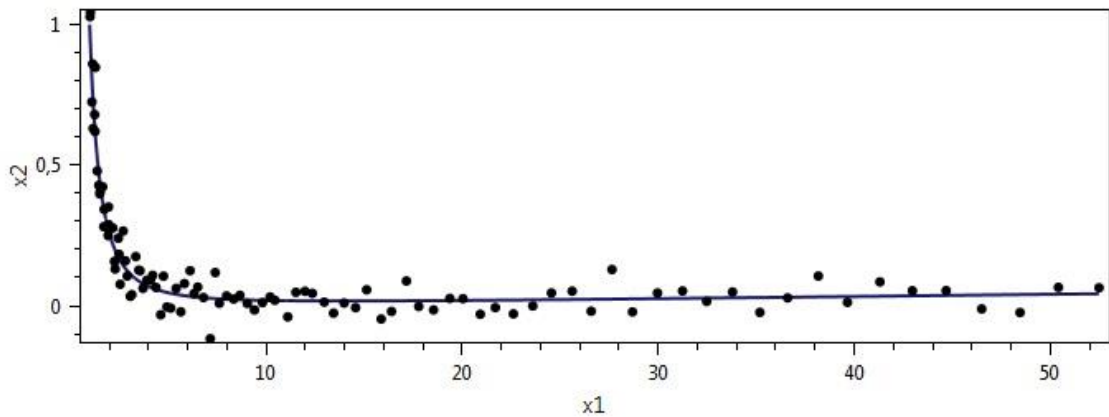


Рис. 3. Экспериментальные данные и фазовая траектория системы (1) с начальными данными $x_1(0) = 1$, $x_2(0) = 1$.

Случай 2. Положение равновесия системы (1) – узел.

Получены следующие оценки параметров $\theta_1, \theta_2, \theta_3, \theta_4$:

$$\hat{\theta}_1 = -1.00015, \hat{\theta}_2 = -0.00081, \hat{\theta}_3 = 2.00096, \hat{\theta}_4 = 0.00058.$$

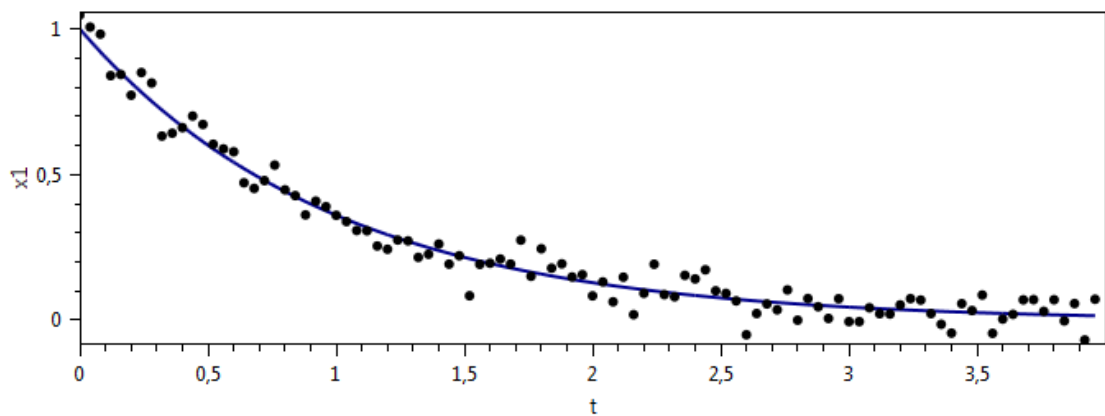


Рис. 4. Экспериментальные данные и график компоненты $x_1(t, \hat{\theta})$ решения системы (1).

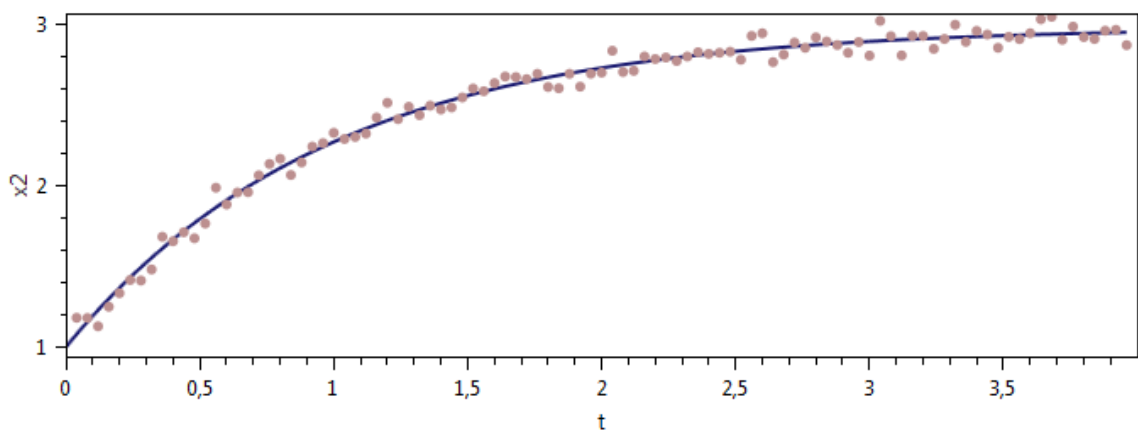


Рис. 5. Экспериментальные данные и график компоненты $x_2(t, \hat{\theta})$ решения системы (1).

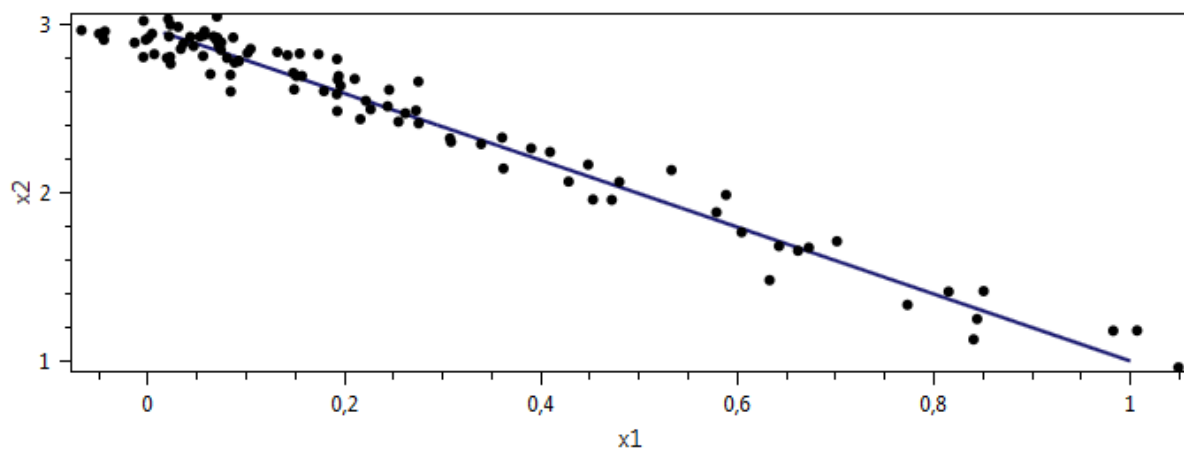


Рис. 6. Экспериментальные данные и фазовая траектория системы (1)
с начальными данными $x_1(0) = 1$, $x_2(0) = 1$.

Случай 3. Положение равновесия системы (1) – центр.

Получены следующие оценки параметров $\theta_1, \theta_2, \theta_3, \theta_4$:

$$\hat{\theta}_1 = 0.00088, \quad \hat{\theta}_2 = 2.99919, \quad \hat{\theta}_3 = -2.99914, \quad \hat{\theta}_4 = -0.00058.$$

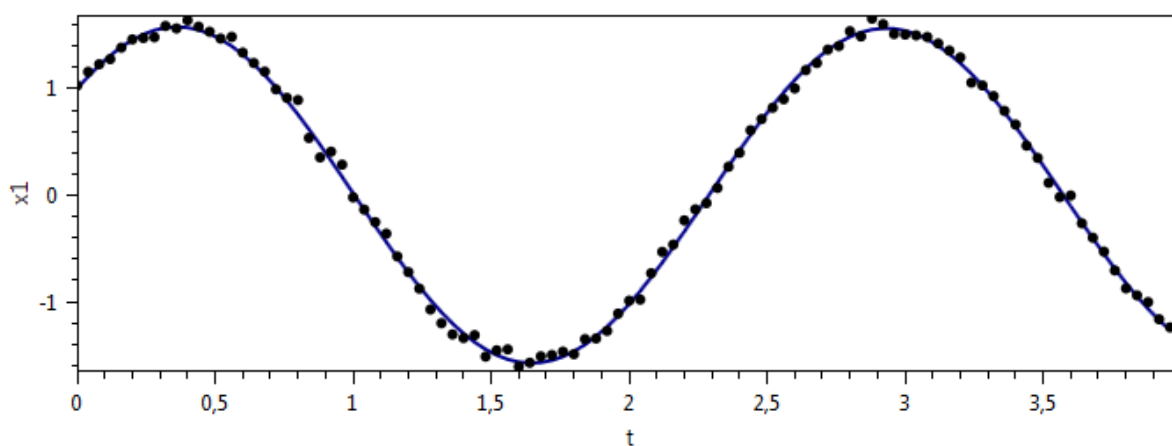


Рис. 7. Экспериментальные данные
и график компоненты $x_1(t, \hat{\theta})$ решения системы (1).

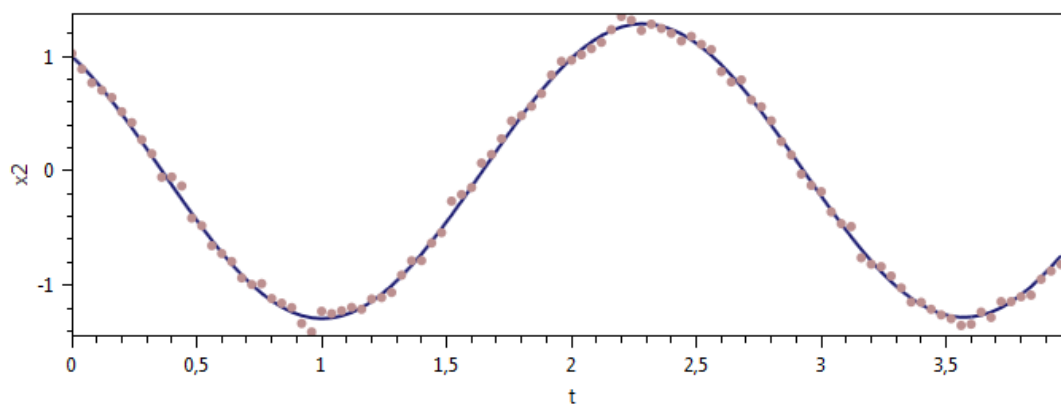


Рис. 8. Экспериментальные данные
и график компоненты $x_2(t, \hat{\theta})$ решения системы (1).

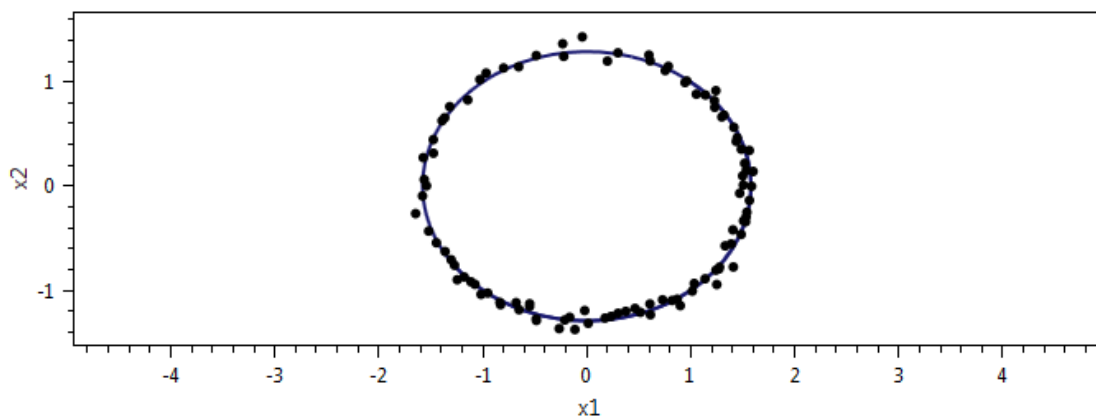


Рис. 9. Экспериментальные данные и фазовая траектория системы (1)
с начальными данными $x_1(0) = 1$, $x_2(0) = 1$.

ЛИТЕРАТУРА

1. Самарский А. А., Гулин А. В. Численные методы. – М.: Наука, 1989. – 432 с.
2. Стенин И. В., Шаманаев П. А. Алгоритм решения разреженной системы линейных алгебраических уравнений большой размерности с использованием метода сопряженных градиентов [Электронный ресурс] // Огарев-online. – 2017. – № 13. – Режим доступа: <http://journal.mrsu.ru/arts/algorithm-resheniya-razrezhennoj-sistemy-linejnyx-algebraicheskix-uravnenij-bolshoj-razmernosti-s-ispolzovaniem-metoda-sopryazhennyx-gradientov>.
3. Челышов М. С., Шаманаев П. А. Решение задачи идентификации параметров динамических систем с использованием метода ортогональной циклической редукции // Прикладная математика и механика: сб. научных трудов. – № 11. – Ульяновск: УлГТУ, 2017. – С. 264–271.
4. Zhengfeng Li, Michael R. Osborne, Tania Prvan. Parameter estimation of ordinary differential equations // IMA Journal of Numerical Analysis. – 2005. – No. 25. – pp. 264–285.