



eISSN 2311-2468
Том 7, № 10. 2019
Vol. 7, no. 10. 2019

электронное периодическое издание
для студентов и аспирантов

Огарёв-онлайн

Ogarev-online

<https://journal.mrsu.ru>



**КАПУСТИНА Е. С., АЛЬБИКОВ Р. Р.,
КУЛЯГИН А. И., ФРОЛОВ Т. Е., ЧАЛДАЕВ А. И.
РЕШЕНИЕ ЗАДАЧ ГАЗОВОЙ ДИНАМИКИ РАЗРЫВНЫМ
МЕТОДОМ ГАЛЁРКИНА В КРИВОЛИНЕЙНОЙ СИСТЕМЕ КООРДИНАТ¹**

Аннотация. Рассматривается разработка методики для решения одномерного уравнения газовой динамики разрывным методом Галёркина. Выполнены тестовые расчеты для задачи о сильном точечном взрыве.

Ключевые слова: разрывный метод Галёркина, газовая динамика, задача Седова.

**KAPUSTINA E. S., ALBIKOV R. R.,
KULYAGIN A. I., FROLOV T. E., CHALDAEV A. I.
SOLUTION FOR THE FLUID DYNAMICS PROBLEMS
BY THE DISCONTINUOUS GALERKIN METHOD
IN THE CURVILINEAR COORDINATE SYSTEM**

Abstract. The development of a method for solving the one-dimensional fluid dynamics equation by the discontinuous Galerkin method is considered. Test calculations have been done for the problem of a strong point explosion.

Keywords: discontinuous Galerkin method, fluid dynamics, Sedov problem.

Введение. Численные методы повышенного порядка точности для решения задач механики сплошной среды на сегодняшний день развиваются многими научными коллективами [1–4]. Наиболее перспективным методом является метод Галёркина с разрывными базисными функциями [5]. Он сочетает в себе качества конечно-объемных и конечно-элементных методов, что позволяет получать с достаточной точностью решать задачи, решения которых характеризуются высокими значениями градиентов. В данной работе построена методика решения одномерных задач газовой динамики, обладающих сферической симметрией. Выполнены тестовые расчеты для задачи о сильном точечном взрыве [6; 7].

Разрывный метод Галёркина. Определим разрывный метод Галёркина [5] для одномерной неоднородной системы уравнений Эйлера.

$$\frac{\partial q}{\partial t} + \frac{1}{r^n} \frac{\partial(r^n f)}{\partial r} + h = 0, \quad (1)$$

¹ Работа выполнена при финансовой поддержке РФФИ и Правительства Республики Мордовия (проект № 18-41-130001 p_a).

где

$$q = \begin{pmatrix} \rho \\ \rho u \\ \rho E \end{pmatrix}; \quad (2)$$

$$f = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u H \end{pmatrix}; \quad (3)$$

$$h = \frac{n}{r} \begin{pmatrix} 0 \\ -p \\ 0 \end{pmatrix}. \quad (4)$$

В приведенных выше уравнениях ρ – плотность жидкости, u – скорость жидкости, p – статическое давление, E и H – полная энергия и полная энтальпия на единицу массы.

$$E = \frac{p}{\rho(\gamma - 1)} + \frac{u^2}{2}, \quad (5)$$

$$H = E + \frac{p}{\rho}. \quad (6)$$

Для нахождения давления используем уравнение состояния идеального газа

$$p = (\gamma - 1)\rho\varepsilon, \quad (7)$$

где γ – показатель адиабаты.

В уравнениях (1) – (4) $n = 0$ для декартовых координат, $n = 1$ для цилиндрических координат и $n = 2$ для сферических координат.

Для применения разрывного метода Галеркина введем равномерную сетку

$$0 < r_{\frac{1}{2}} < \dots < r_{i-\frac{1}{2}} < r_{i+\frac{1}{2}} < \dots < r_{N-\frac{1}{2}} < 1. \quad (8)$$

На каждом отрезке определим базис

$$\varphi_{0k}(t) = 1, \quad (9)$$

$$\varphi_{1k}(t) = \frac{r - r_c}{\Delta r}, \quad (10)$$

$$\varphi_{2k}(t) = \left(\frac{r - r_c}{\Delta r}\right)^2, \quad (11)$$

$$\Delta r = r_{i+\frac{1}{2}} - r_{i-\frac{1}{2}}. \quad (12)$$

Приближенное решение q_h в каждой ячейке $[r_{i-\frac{1}{2}}, r_{i+\frac{1}{2}}]$ представляется в виде разложения по базисным функциям (9) – (12).

$$q_h(r, t) = \sum_{i=0}^2 q_{ik}(t)\varphi_{ik}(r), \quad (13)$$

Разрывный метод Галеркина получается путем умножения уравнения Эйлера (1) на базисную функцию φ_k и интегрирование по некоторому объему (K).

$$\int_K \frac{\partial q}{\partial t} \varphi_k(r) dK + \int_K \frac{1}{r^n} \frac{\partial(r^n f)}{\partial r} \varphi_k(r) dK + \int_K h \varphi_k(r) dK = 0, \quad (14)$$

где $dK = r^n dr$ – для одномерного потока.

Для одного пространственного измерения уравнение (14) принимает вид

$$\begin{aligned} & \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \frac{\partial q}{\partial t} \varphi_k(r) r^n dr + [f r^n \varphi_k(r)]_{\frac{i+1}{2}} - [f r^n \varphi_k(r)]_{\frac{i-1}{2}} - \\ & - \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} f \frac{d\varphi_k(r)}{dr} r^n dr + \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} h \varphi_k(r) r^n dr = 0. \end{aligned} \quad (15)$$

Функция f не определена в точках $r_{i-\frac{1}{2}}$ и $r_{i+\frac{1}{2}}$ – решение может быть разрывным на гранях элементов, что приведет к неоднозначности. Эта проблема преодолевается с помощью соответствующего выбора численных потоков на гранях, зависящих от предельных значений функции q_h слева и справа от точек $r_{i-\frac{1}{2}}$ и $r_{i+\frac{1}{2}}$.

Будем использовать дискретные потоки Русанова-Лакса-Фридрикса.

$$\hat{f}_{i+\frac{1}{2}} = \hat{f}(q_{i+\frac{1}{2}}^L, q_{i+\frac{1}{2}}^R) = \frac{1}{2} \left(f(q_{i+\frac{1}{2}}^R) + f(q_{i+\frac{1}{2}}^L) - \alpha (q_{i+\frac{1}{2}}^R - q_{i+\frac{1}{2}}^L) \right), \quad (16)$$

$$\alpha = \max \left\{ |q_{i+\frac{1}{2}}^L| + c^L, |q_{i+\frac{1}{2}}^R| + c^R \right\}, \quad c = \sqrt{\frac{\gamma p}{\rho}}. \quad (17)$$

Используя приближение для функции q_h , получим систему дифференциальных уравнений

$$\frac{dq}{dt} = -M^{-1} \left[\left(\hat{f} r^n \begin{pmatrix} \varphi_0 \\ \varphi_1 \\ \varphi_2 \end{pmatrix} \right) \Big|_{r_{i+\frac{1}{2}}}^{r_{i-\frac{1}{2}}} - \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} f r^n \frac{\partial}{\partial r} \begin{pmatrix} \varphi_0 \\ \varphi_1 \\ \varphi_2 \end{pmatrix} dr + \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} h r^n \begin{pmatrix} \varphi_0 \\ \varphi_1 \\ \varphi_2 \end{pmatrix} dr \right]. \quad (18)$$

Ограничители наклона. Для многих конечно-разностных или конечно-объемных вычислений характерны разрывные решения и возникновение осцилляции. Так же и в нашем случае, вблизи разрывов потока могут возникать ложные колебания, поэтому необходимо использование лимитеров. Одним из общих подходов к решению данной проблемы служат специальные ограничители.

Согласно [5] будем обозначать действие оператора лимитирования на функцию q следующим образом: $\Lambda \Pi_h q$. Одним из простых подходов ограничения служит лимитер Кокбурна.

Запишем разложение решения по линейному базису в ячейке:

$$q = q_0 + q_1 \frac{r - r_c}{\Delta r}, \quad \Delta r = r_{i+\frac{1}{2}} - r_{i-\frac{1}{2}}. \quad (19)$$

Для функции (19) действие оператора $\Lambda \Pi_h q$ запишем как

$$\Lambda \Pi_h q(r, t) = q_0 + \widetilde{q}_1 \frac{r - r_c}{\Delta r}, \quad \Delta r = r_{i+\frac{1}{2}} - r_{i-\frac{1}{2}}. \quad (20)$$

Значение функции \widetilde{q}_1 вычисляется как

$$\widetilde{q}_1 = 2 * \min \text{mod} \left[q \left(r_{i+\frac{1}{2}} \right) - q_{0i}, \alpha \left(\bar{q}_{i+\frac{1}{2}} - q_{0i} \right), \alpha \left(q_{0i} - \bar{q}_{i-\frac{1}{2}} \right) \right], \quad (21)$$

где q_{0i} - среднее значение на интервале $r_{i-\frac{1}{2}}, r_{i+\frac{1}{2}}$, а $\bar{q}_{i+\frac{1}{2}}$ и $\bar{q}_{i-\frac{1}{2}}$ равны

$$\bar{q}_{i+\frac{1}{2}} = \frac{q_{0i+1} + q_{0i}}{2}, \quad \bar{q}_{i-\frac{1}{2}} = \frac{q_{0i-1} + q_{0i}}{2}. \quad (22)$$

Если порядок полинома больше 1, то

$$q^l(r, t) = q_0^l + q_1^l \frac{r - r_c}{\Delta r}, \quad \Delta r = r_{i+\frac{1}{2}} - r_{i-\frac{1}{2}}, \quad (23)$$

где $q_0^l = q_0 + \frac{q_2}{12}$ - среднее интегральное значение на интервале $r_{i-\frac{1}{2}}, r_{i+\frac{1}{2}}$ для полинома степени равной 2.

После применения лимитера

$$\Lambda \Pi_h q^l(r, t) = \widetilde{q}_0^l + \widetilde{q}_1^l \frac{r - r_c}{\Delta r}, \quad \Delta r = r_{i+\frac{1}{2}} - r_{i-\frac{1}{2}}. \quad (24)$$

Постановка задачи Седова о сильном точечном взрыве [6]. Поместим небольшое количество безразмерной энергии $\varepsilon = 1$ в небольшую область радиуса dr в центре сетки, заполненной неподвижной средой плотности ρ_0 , с давлением p_0 . Безразмерное давление внутри этого объема определяется:

$$p_0 = \frac{3(\gamma - 1)\varepsilon}{(\nu + 1)\pi dr^\nu}, \quad x < x_c, \quad (25)$$

$$p_0 = 10^{-5}, \quad x > x_c, \quad (26)$$

где $\nu = 2$ для цилиндрических координат, $\nu = 3$ для сферических координат.

Выбираем dr в 3,5 раза больше, чем шаг сетки. Плотность устанавливается равной $\rho_0 = 1$ по всей сетке, скорость изначально в покое $u_0 = 0$.

На рисунке 1 представлены результаты расчетов, масштабированные относительно значений на фронте ударной волны. Результаты расчетов хорошо согласуются с данными из классических источников [6; 7].

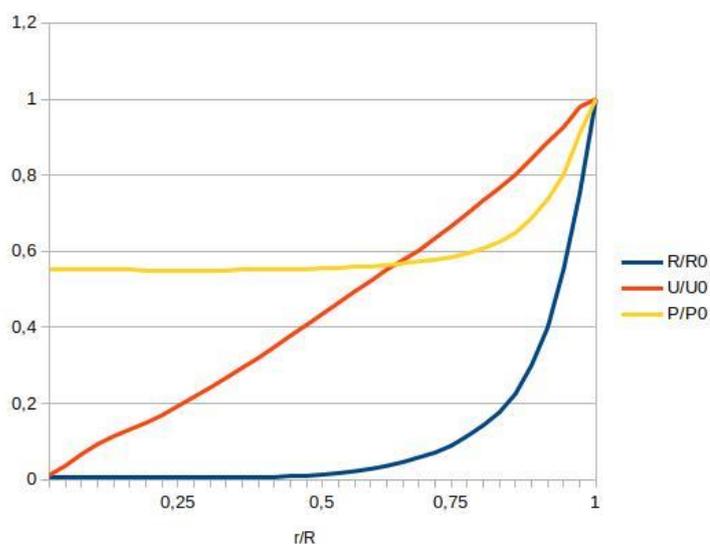


Рис. 1. Распределения плотности, скорости и давления, нормированные относительно значений на фронте ударной волны.

ЛИТЕРАТУРА

1. Вязников К. В., Тишкин В. Ф., Фаворский А. П. Построение монотонных разностных схем повышенного порядка аппроксимации для систем уравнений гиперболического типа // Математическое моделирование. – 1989. – Т. 1. № 5. – С. 95–120.
2. Jiang G.-S., Shu C.-W. Efficient implementation of weighted ENO schemes // Journal of computational physics. – 1996. – Vol. 126. – P. 202–228.
3. Shu C.-W. Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws. ICASE Report No. 97-65. – Institute for Computer Applications in Science and Engineering (ICASE), 1997. – 78 p.
4. Жалнин Р. В. О построении параллельного вычислительного алгоритма высокого порядка точности для гиперболических систем уравнений // Труды Средневолжского математического общества. – 2007. – Т. 9. № 1. – С. 145–153.
5. Cockburn B. An Introduction to the Discontinuous Galerkin Method for Convection-Dominated Problems // Advanced Numerical Approximation of Nonlinear Hyperbolic Equations (Lecture Notes in Mathematics). – 1998. – Vol. 1697. – P. 151–268.
6. Седов Л. И. Механика сплошной среды: в 2 т. – Т. 1. – М.: Наука, 1970. – 492 с.
7. Зельдович Я. Б., Райзер Ю. П. Физика ударных волн и высокотемпературных гидродинамических явлений. – М.: ФИЗМАТЛИТ, 2008. – 656 с.

ИЛЬЧЕНКО П. В., ШАМАЕВ А. В.

SCADA-СИСТЕМА АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ КЛИМАТОМ

Аннотация. В статье дано описание разрабатываемой авторами SCADA-системы автоматизированного управления климатом. Приведены основные контролируемые параметры физических сред, описан алгоритм построения системы.

Ключевые слова: автоматизация управления, центральный диспетчерский пункт, SCADA-система.

ILCHENKO P. V., SHAMAEV A.V.

SCADA SYSTEM FOR AUTOMATED CLIMATE CONTROL

Abstract. The article describes the SCADA system for automated climate control. The main controlled parameters and implementation of the system construction are given.

Keywords: control automation, central control room, SCADA system.

В течение последних 20 лет и по настоящее время наблюдается активное внедрение систем автоматизированного управления технологическими процессами, что приводит к уменьшению доли человеческого труда во многих сферах современного производства: машиностроении, перерабатывающей, электротехнической, химической, пищевой, текстильной промышленности, производстве электронных компонентов, производстве лекарственных средств, электроэнергетике. В результате участие человека сводится к контролю и обслуживанию состояния оборудования, контролю качества готового продукта и действий, направленных на устранение сбоев, возникающих в работе оборудования.

Автоматизированные системы управления технологическими процессами применяются не только в промышленности. Поддержание определенных значений параметров климата, в частности, температуры и влажности, требуется во многих промышленных и общегражданских помещениях, таких как лаборатории, производственные помещения, используемые для выпуска прецизионной продукции, медицинские учреждения, архивы, музейные комнаты, торговые залы и т. д. В настоящее время для организации управления этими параметрами все чаще используются системы диспетчеризации, построенные на основе специализированных программ (SCADA-программ) [1].

Процесс разработки таких систем можно условно разделить на два этапа: разработка аппаратной части и разработка программной части. При этом на этапе разработки аппаратной части решаются задачи разработки схемы вентиляции и кондиционирования,

подбора необходимых датчиков и исполнительных механизмов, согласования уровней сигналов и импедансов, разработки электрических схем.

В данной статье основное внимание уделено этапу разработки программной составляющей и архитектуры SCADA-системы автоматизированного управления климатом для промышленного здания (далее – Система).

Реализация данного этапа включает решение задач разработки:

- функциональных требований к программе;
- мнемосхемы работы автоматики;
- программной архитектуры;
- человеко-машинного интерфейса;
- программного кода.

Опишем архитектуру и процесс функционирования разрабатываемой Системы.

Разрабатываемая Система (см. рис. 1) включает следующее оборудование: персональный компьютер с установленным программным обеспечением, сервер баз данных, маршрутизатор, датчики и исполнительные механизмы [2].

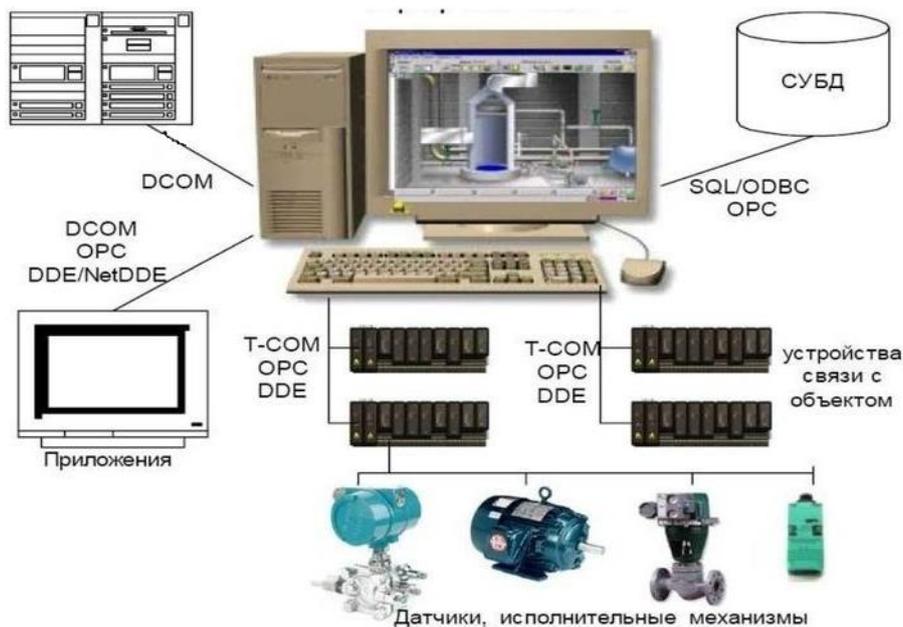


Рис. 1. Структура разрабатываемой SCADA-системы.

На персональном компьютере должно быть установлено разрабатываемое программное обеспечение, в состав которого должен входить сервер баз данных, предназначенный для архивирования и хранения данных, передаваемых с датчиков.

Маршрутизатор организует связь персонального компьютера с контроллером.

Контроллер принимает информацию от датчиков, а также выдает управляющие сигналы исполнительным устройствам. Он должен быть связан с персональным компьютером по интерфейсу RS-485 и на нем должно быть установлено специализированное встроенное программное обеспечение.

Датчики формируют информационные сигналы, включающие информацию о значении измеряемой величины и состоянии оборудования, которые через маршрутизатор поступают в контроллер.

Исполнительные устройства предназначены для оказания воздействия на значения контролируемых параметров и/или состояние управляемого объекта в целом. По своему назначению исполнительные устройства могут быть как регулирующие, так и защитные.

Информация от удаленных объектов по каналам связи поступает на управляющий компьютер, который [3]:

- интерпретирует все данные, поступающие от датчиков;
- принимает решения в соответствии с алгоритмом работы программы;
- посылает управляющие сигналы исполнительным устройствам;
- обменивается информацией с человеком-оператором и реагирует на его команды.

Проектируемое программное обеспечение должно выполнять следующие функции:

- динамическая визуализация состояния контролируемых объектов и процессов;
- мониторинг и управление контролируемыми параметрами помещений;
- дистанционный контроль данных;
- авторизованный доступ и управление пользователями;
- регистрация и управление событиями;
- отслеживание состояния исполнительных механизмов и аварийных ситуаций;
- создание отчетов об авариях и неисправностях;
- подтверждение, блокировка и разблокировка аварийных сообщений;
- построение графиков изменения контролируемых параметров в реальном времени;
- управление хранением и архивированием данных;
- визуальное и звуковое сопровождение аварийных сообщений;
- поддержка многооконного интерфейса, всплывающих подсказок;
- взаимодействие с системой управления базами данных;
- сетевая коммуникация, реализованная по технологии «клиент-сервер»;
- поддержка автоматического перехода на зимнее и летнее время.

На рисунке 2 представлена мнемосхема работы автоматики, обеспечивающей контроль параметров воздуха, который поступает из окружающей среды. На данной схеме представлены основные контролируемые параметры, к которым относятся: температуры теплоносителя, наружного воздуха, воздуха на выходе водяного калорифера, а также положение заслонок и жалюзи (открыто/закрыто), состояния вентиляторов (включено/выключено) и водяных насосов (включено/выключено) [4].

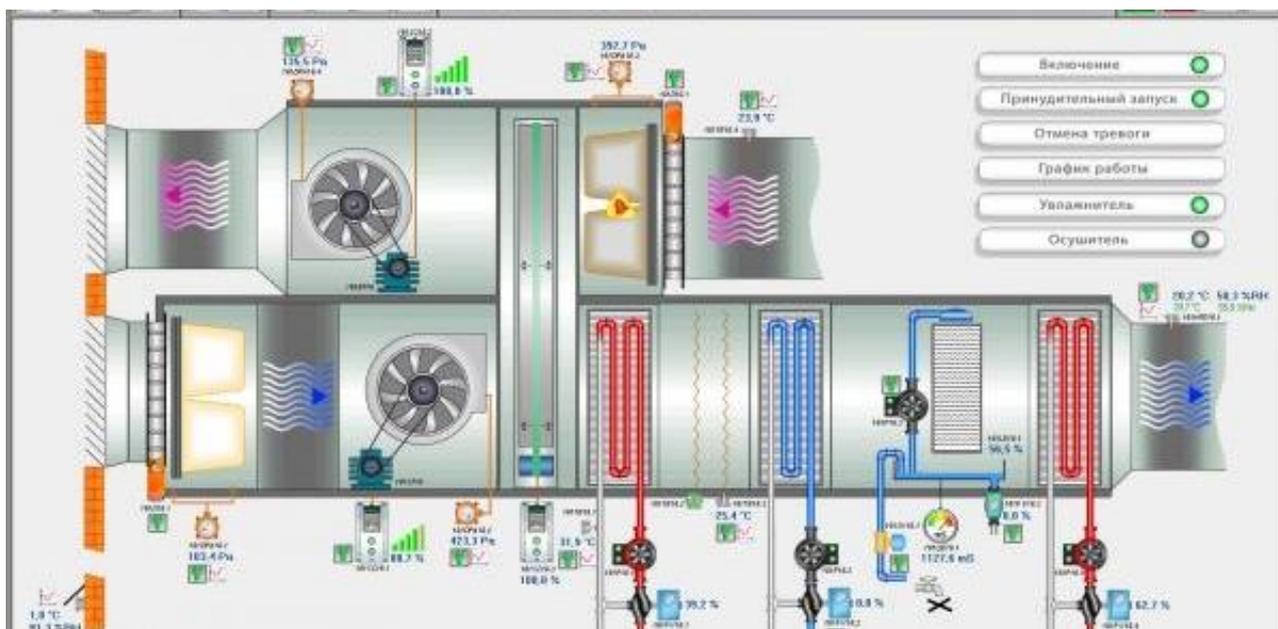


Рис. 2. Мнемосхема включения автоматики.

Опишем функционирование разрабатываемой Системы. Все помещения должны быть оборудованы датчиками – измерителями контролируемых параметров. Для каждого помещения устанавливаются требуемые значения контролируемых параметров, а именно: температуры, влажности, класса чистоты воздуха, которые должны поддерживаться. Данные, формируемые датчиками, передаются по линии связи на персональный компьютер. Программа сравнивает фактические значения контролируемых параметров, полученные от датчиков, с их требуемыми значениями и формирует соответствующие управляющие воздействия на исполнительные механизмы. Например, в зависимости от того, является ли измеренное значение температуры в помещении больше или меньше требуемого, система увеличивает производительность кондиционера или калорифера соответственно. Система позволяет оператору дистанционно управлять исполнительными устройствами, отраженными на мнемосхеме. В частности, он может изменять значения параметров

исполнительных устройств, оказывающих влияние на контролируемые параметры помещения, например, путем изменения температуры теплоносителя, производительности кондиционера, парогенератора и т. д., а также изменять состояние данных устройств («включено/выключено»). Программа в режиме реального времени должна отображать на экране оператора графики изменения заданного и фактического значения температуры в каждом контролируемом помещении. Программа генерирует цифровые сигналы, которые поступают на контроллер, а тот, в свою очередь, выдает сигналы на исполнительные устройства. Кроме того, при возникновении аварийной ситуации программа должна выдавать оператору сообщение о месте обнаружения аварийной ситуации для облегчения процесса принятия решения и выполнения действий по ее устранению. Также Система должна вести журнал событий и сохранять накопленные данные в специализированной СУБД [5].

ЛИТЕРАТУРА

1. Группа компаний «Мосрегионвент» [Электронный ресурс]. – Режим доступа: <https://mosregionvent.ru/> (дата обращения 20.04.2018).
2. Данилушкин И. А. Аппаратные средства и программное обеспечение систем промышленной автоматизации: учебное пособие. – Самара: Самар. гос. техн. ун-т, 2007. – 168 с.
3. Обзор SCADA TRACE MODE 6 [Электронный ресурс]. – Режим доступа: <http://www.adastra.ru/products/overview/> (дата обращения 18.04.2018).
4. SCADA система MasterSCADA [Электронный ресурс]. – Режим доступа: <http://insat.ru> (дата обращения 22.04.2018).
5. Система диспетчеризации DesigoInsight [Электронный ресурс]. – Режим доступа: <http://buildingtechnologies.siemens.ru/products/cps/desigo/di/> (дата обращения 22.04.2018).

КУРБАТОВ Д. И., ЕГОРОВА Д. К.

**ПРИМЕНЕНИЕ НЕЧЕТКОГО АЛГОРИТМА КЛАСТЕРИЗАЦИИ C-MEANS
ДЛЯ СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ**

Аннотация. В статье рассматривается применение нечеткого алгоритма кластеризации c-means для анализа изображений. Приведена программная реализация алгоритма на языке C#.

Ключевые слова: кластеризация, нечеткий метод c-means, итерации, цвет, пиксели.

KURBATOV D. I., EGOROVA D. K.

**APPLICATION OF FUZZY C-MEANS CLUSTERING
ALGORITHM FOR IMAGE SEGMENTATION**

Abstract. Application of fuzzy clustering algorithm with c-means for image analysis is considered in this article. A software implementation of the algorithm in C# is provided.

Keywords: clustering, fuzzy c-means method, iteration, color, pixels.

Введение. Разбиение изображения на однородные области является ключевым элементом компьютерного (технического) зрения. Для этих целей могут быть использованы методы кластерного анализа, так как они позволяют осуществить сегментацию пикселей на основе сходства по цвету или уровня интенсивности серого.

В работе приведена программная реализация сегментации цветного изображения методом c-means, относящегося к методам «мягкой» кластеризации и позволяющего более точно вычислить принадлежность элемента кластеру. Этот алгоритм успешно используется для кластеризации изображений и неконтролируемой сегментации медицинских, геологических, спутниковых изображений и т.п. [1].

Описание алгоритма. Нечеткий алгоритм кластеризации c-means был разработан (для случая $m=2$) J. C. Dunn в 1973 г. и усовершенствован (для случая $m>2$) J. C. Bezdek в 1981 г. Данный метод предполагает, что входные данные могут принадлежать более чем одному кластеру одновременно, то есть первоначально для каждого вектора, описывающего исследуемый объект, случайным образом определяется вероятность принадлежности заданным кластерам. Алгоритм получает на входе набор кластеризуемых векторов, количество кластеров, коэффициент неопределенности m и коэффициент $\varepsilon > 0$, определяющий точность алгоритма. Затем запускается итерационный процесс, состоящий в выполнении следующей последовательности действий: 1) расчет центров кластеров; 2) расчет расстояния от каждого вектора до центра каждого кластера; 3) расчет и нормализация коэффициентов принадлежности векторов кластерам; 4) расчет значения матрицы нечеткого

разбиения и сравнение этого значения со значением матрицы нечеткого разбиения на предшествующей итерации. На выходе получают матрицу вероятностей принадлежности каждого входного вектора каждому кластеру.

Таким образом, нечеткий алгоритм c-means минимизирует величину

$$\sum_{i=1}^{|X|} \sum_{j=1}^C u_{i,j}^m \|x_i - c_j\|^2, \quad 1 \leq m \leq \infty, \quad (1)$$

где $m \in R$, $u_{i,j}$ – коэффициент принадлежности вектора x_i кластеру c_j , x_i – i -ый компонент $|X|$ -мерного вектора X , C – количество кластеров, c_j – центр j -ого кластера, а $\| * \|$ – норма, определяющая расстояние от вектора до центра кластера. Нечеткое разбиение входных данных на кластеры производится итеративной оптимизацией функции (1), пересчетом коэффициентов принадлежности $u_{i,j}$ и переопределением центров кластеров c_j . [2]

Вычисляемые величины. На каждой итерации вычисляются следующие величины.

1. Центры кластеров:

$$c_j = \frac{\sum_{i=1}^n u_{i,j}^m x_i}{\sum_{i=1}^n u_{i,j}^m}, \quad (2)$$

где $u_{i,j}$ – коэффициент принадлежности x_i вектора к кластеру c_j .

2. Коэффициент принадлежности:

$$u_{i,j} = \frac{1}{\sum_{k=1}^n \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}. \quad (3)$$

3. Решающая функция:

$$\max(|u_{i,j}^{(k)} - u_{i,j}^{(k-1)}|), \quad (4)$$

где k – номер итерации алгоритма.

4. Проверка завершения итерационного процесса.

Реализация алгоритма. Далее приведена программная реализация алгоритма c-means для сегментации цветных изображений на языке C# в программной среде VisualStudio 2017 [3].

Графический интерфейс представлен на рисунке 1. Для загрузки сегментируемого изображения предусмотрено меню «File».

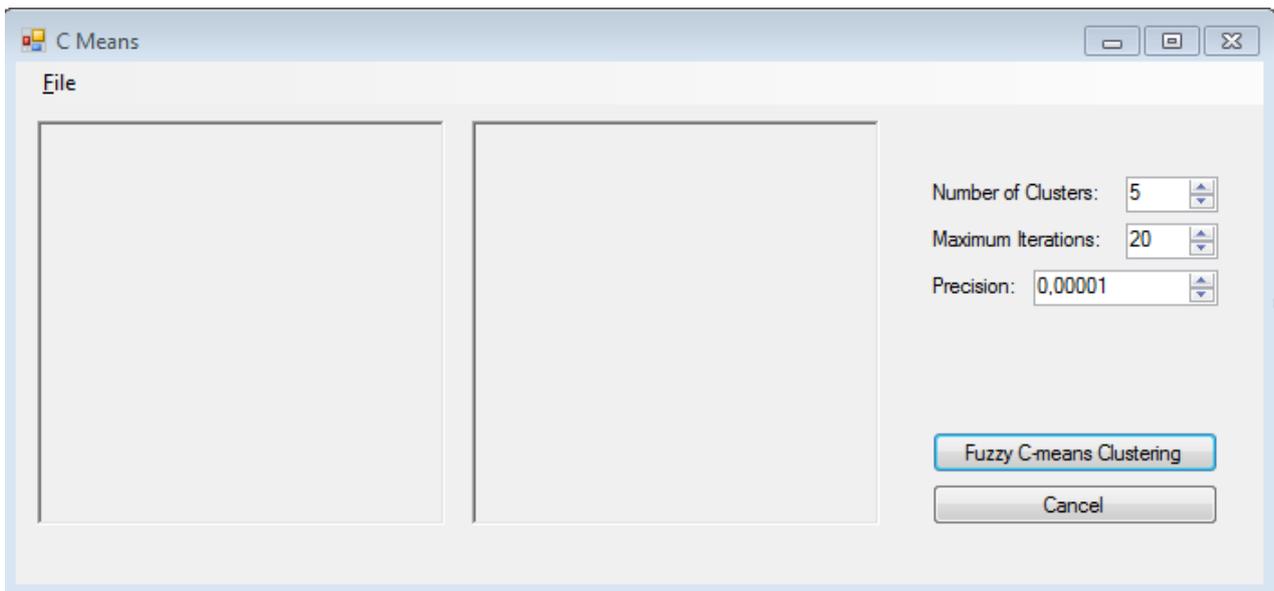


Рис. 1. Графический интерфейс.

Параметры, доступные для изменения, приведены в правой верхней части интерфейса программы – это количество кластеров, максимальное количество итераций и точность.

При активации «Fuzzy C-means Clustering» запускается процесс создания точечных объектов кластера для каждого пикселя изображения (см. код, приведенный ниже).

```
List<ClusterPoint> points = new List<ClusterPoint>();
for (int row = 0; row < originalImage.Width; ++row)
{
    for (int col = 0; col < originalImage.Height; ++col)
    {
        Color c2 = originalImage.GetPixel(row, col);
        points.Add(new ClusterPoint(row, col, c2));
    }
}
```

Затем создается заданное количество «ClusterCentroid» объектов кластера. Центры кластеров (или центроиды) на первой итерации выбираются случайным образом и корректируются алгоритмом (см. код).

```
List<ClusterCentroid> centroids = new List<ClusterCentroid>();
//Create random points to use as the cluster centroids
Random random = new Random();
for (int i = 0; i < numClusters; i++)
```

```

{
    int randomNumber1 = random.Next(sourceImage.Width);
    int randomNumber2 = random.Next(sourceImage.Height);
    centroids.Add(new ClusterCentroid(randomNumber1, randomNumber2,
        filteredImage.GetPixel(randomNumber1, randomNumber2)));
}

```

Далее создается FCM-объект и запускаются итерации (см. код).

```

FCM alg = new FCM(points, centroids, 2, filteredImage, (int)numericUpDown2.Value);
k++;
alg.J = alg.CalculateObjectiveFunction();
alg.CalculateClusterCentroids();
alg.Step();
double Jnew = alg.CalculateObjectiveFunction();
Console.WriteLine("Run method i={0} accuracy = {1} delta={2}",
    k, alg.J, Math.Abs(alg.J - Jnew));
backgroundWorker.ReportProgress((100 * k) / maxIterations, "Iteration " + k);
if (Math.Abs(alg.J - Jnew) < accuracy) break;

```

После завершения итераций алгоритм выполняет процесс построения результирующего изображения, назначая каждый пиксель кластеру, для которого он имеет наибольший коэффициент принадлежности.

Для тестового примера было выбрано изображение, представленное на рисунке 2. Сегментация осуществлялась на 2 кластера.



Рис. 2. Исходное изображение.

Результирующее изображение представлено на рисунке 3.



Рис.3. Результат выполнения алгоритма.

ЛИТЕРАТУРА

1. Jain A. K. Data clustering: 50 years beyond K-means // Pattern Recognition Letters. – 2010. – Vol. 31. – P. 651–666.
2. Барсегян А. А., Куприянов М. С., Холод И. И., Тесс М. Д., Елизаров С. И. Анализ данных и процессов: учеб. пособие / под ред. А. А. Барсегян. – 3-е изд., перераб. и доп. – СПб.: БХВ–Петербург, 2009. – 512 с.
3. Gauge С. Cluster analysis [Электронный ресурс]. – Режим доступа: <https://ru.scribd.com> (дата обращения 08.05.2019).

СИДОРЕНКО Д. С., МУРЮМИН С. М.
АСИМПТОТИЧЕСКАЯ ЭКВИВАЛЕНТНОСТЬ
ДВУХСЛОЙНЫХ РАЗНОСТНЫХ СХЕМ

Аннотация. Получены оценки возмущения для постоянных матричных операторов, используемых для описания операторно-разностных схем, при которых гарантируется асимптотическая эквивалентность оных. Проведены подтверждающие численные эксперименты.

Ключевые слова: асимптотическая эквивалентность, разностная схема, операторно-разностная схема, порядок аппроксимации, оценка возмущения, численный эксперимент.

SIDORENKO D. S., MURYUMIN S. M.
ASYMPTOTIC EQUIVALENCE OF TWO-STAGE DIFFERENCE SCHEMES

Abstract. Value estimations for fluctuations of difference schemes' matrix operators, which provide asymptotic equivalence of these schemes, are obtained. Numeric experiments for the confirmation of theoretic results are described.

Keywords: asymptotic equivalence, difference scheme, operator-difference scheme, approximation order, estimation of fluctuation, numeric experiment.

Краевые задачи для дифференциальных уравнений типа

$$Lu + f(x) = 0$$

можно трактовать как операторные уравнения:

$$Au = f.$$

Это справедливо в частности для уравнений параболического типа:

$$\frac{\partial u}{\partial t} = Lu + f(x, t),$$

исследованию асимптотического поведения разностных схем для которых и посвящена эта работа.

Рассмотрим искомую функцию $u = u(x, t)$ как некоторую абстрактную функцию $u(t)$ со значениями в некотором банаховом пространстве. Если ввести для t равномерную сетку с шагом τ , то можно получить выражение для r -слойной операторно-разностной схемы:

$$B_0(t_n)u(t_{n+1}) = \sum_{s=1}^{r-1} C_s(t_n)u(t_{n+1-s}) + f(t_n), \quad n \geq r - 1,$$

где $B_0(t), C_s(t)$ – линейные операторы, действующие из некоторого линейного нормированного пространства \mathcal{H} в \mathcal{H} , которое в свою очередь зависит от некоторого параметра h . Заметим, что эти линейные операторы также зависят от τ и h .

Тогда для двухслойной схемы имеем:

$$B_0 u_{n+1} + B_1 u_n = \tau \tilde{f}_n, \quad n \geq 0.$$

Выполняя замену, предложенную в [1], получим каноническую форму двухслойных схем:

$$B(t_n) \frac{u_{n+1} - u_n}{\tau} + A(t_n) u_n = \tilde{f}_n, \quad n \geq 0;$$

$$B u_t + A u = \tilde{f}(t). \quad (1)$$

Рассмотрим первую краевую задачу для одномерного однородного уравнения теплопроводности:

$$u_t - u_{xx} = 0,$$

$$u(x, 0) = \phi(x), \quad 0 \leq x \leq L,$$

$$u(0, t) = \psi_1(t), \quad 0 \leq t \leq T, \quad (2)$$

$$u(L, t) = \psi_2(t), \quad 0 \leq t \leq T.$$

Пусть τ – шаг сетки по t , h – шаг сетки по x . Тогда для этой задачи (1) примет вид

$$B u_t + A u = 0, \quad B = E + \tau A, \quad A = -\Lambda, \quad (3)$$

где Λ – оператор вторых конечных разностей:

$$\Lambda = \begin{pmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \dots & 0 \\ 0 & 1 & -2 & \dots & 0 \\ & & \dots & \dots & \\ 0 & \dots & 0 & 1 & -2 \end{pmatrix} \frac{1}{h^2}.$$

Разрешая (3) через u_{n+1} – значение на верхнем слое, – получим

$$u_{n+1} = R u_n, \quad R = E - \tau B^{-1} A,$$

где оператор R называется *оператором перехода*, а B^{-1} – обратный к B оператор.

Рассмотрим два разностные схемы:

$$u_{n+1} = R_1 u_n, \quad u_{n+1} = R_2 u_n, \quad R_1 = R_h^{(1)} \neq R_h^{(2)} = R_2.$$

Согласно [2], условие асимптотической эквивалентности разностных схем при условии согласования норм можно записать как

$$\|R_h^{(1)n} - R_h^{(2)n} P_2\| \leq C h^l,$$

$$\|R_h^{(1)n} P_1 - R_h^{(2)n}\| \leq C h^l,$$

где P_1, P_2 – некоторые операторы, воздействующие на начальные условия, соответствующие первой и второй разностной схеме, l – порядок аппроксимации.

Не теряя общности, можно положить

$$P_1 \equiv P_2 \equiv E.$$

Тогда получим иную форму условия асимптотической эквивалентности:

$$\|R_1^n - R_2^n\| \leq Ch^l. \quad (4)$$

Получим некоторые оценки для нормы выше, при которых (4) справедливо.

Для явной схемы $B \equiv E$, поэтому оператор перехода имеет вид

$$R = E - \tau A.$$

Рассмотрим два различных оператора A_1 и A_2 такие, что:

$$A_2 = A_1 + Q, \quad (5)$$

где Q – оператор возмущения. В этой работе будем полагать, что возмущение постоянно и одинаково для каждого узла сетки:

$$Q = \begin{pmatrix} \varepsilon & \varepsilon & 0 & \dots & 0 \\ \varepsilon & \varepsilon & \varepsilon & \dots & 0 \\ 0 & \varepsilon & \varepsilon & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \varepsilon & \varepsilon \end{pmatrix}. \quad (6)$$

Тогда имеем два оператора перехода R_1 и R_2 :

$$R_1 = E - \tau A_1, \quad R_2 = E - \tau A_2.$$

Получим оценку для возмущения ε , при котором схемы, порожденные операторами R_1 и R_2 , являются асимптотически эквивалентными.

Подставляя операторы в левую часть условия (4) и используя формулу разности степеней:

$$\|R_1^n - R_2^n\| = \|(R_1 - R_2)(R_1^{n-1} + R_1^{n-2}R_2 + \dots + R_1R_2^{n-2} + R_2^{n-1})\|. \quad (7)$$

Здесь и далее будем рассматривать m -норму:

$$\|A\| = \|A\|_1 = \max_i \sum_j |a_{ij}|.$$

Обозначим множители в правой части (7) как D_1 и D_2 , то есть:

$$\|R_1^n - R_2^n\| = \|D_1 D_2\|, \quad (8)$$

$$D_1 = R_1 - R_2, \quad D_2 = R_1^{n-1} + R_1^{n-2}R_2 + \dots + R_1R_2^{n-2} + R_2^{n-1}.$$

По свойству нормы:

$$\|D_1 D_2\| \leq \|D_1\| \cdot \|D_2\|. \quad (9)$$

Сначала оценим вторую норму, потому что далее эта оценка будет справедлива для всех рассматриваемых случаев. Имеем:

$$\|D_2\| = \|R_1^{n-1} + R_1^{n-2}R_2 + \dots + R_1R_2^{n-2} + R_2^{n-1}\|.$$

По теореме о достаточных условиях устойчивости двухслойных схем в линейных нормированных пространствах [1] можно положить:

$$\|R_1^{n-1}\| \leq M_1, \quad \|R_2^{n-1}\| \leq M_2,$$

$$\|R_1^m R_2^{n-1-m}\| \leq \max(M_1, M_2, M_1 M_2) = M, \quad 0 \leq m \leq n-1.$$

Получим

$$\|D_2\| = \|R_1^{n-1} + R_1^{n-2}R_2 + \dots + R_1R_2^{n-2} + R_2^{n-1}\| \leq nM. \quad (10)$$

Теперь оценим первую норму

$$\|D_1\| = \|E - \tau A_1 - E + \tau A_2\| = \|\tau(A_2 - A_1)\| \leq |\tau| \cdot \|A_2 - A_1\|$$

Учитывая, что $\tau > 0$:

$$\|D_1\| \leq |\tau| \cdot \|A_2 - A_1\| = \tau\|A_1 + Q - A_1\| = \tau\|Q\|. \quad (11)$$

Легко вычислить норму Q :

$$\|Q\| = \max_i \sum_j |q_{ij}| = \max_i (2\varepsilon, 3\varepsilon) = 3\varepsilon. \quad (12)$$

Подставляя (12) в (11):

$$\|D_1\| \leq \tau\|Q\| = 3\tau\varepsilon. \quad (13)$$

Объединяя (8), (9), (10) и (13), получим:

$$\|R_1^n - R_2^n\| \leq 3\tau\varepsilon nM.$$

Выполняя замену $\tau = rh^2$, $n = h^{-1}$, имеем:

$$\|R_1^n - R_2^n\| \leq 3Mr\varepsilon h.$$

Используя правую часть условия (4):

$$3Mr\varepsilon h \leq Ch^l,$$

или

$$\varepsilon \leq \tilde{C}h^{l-1}, \quad \tilde{C} = \frac{C}{3Mr}. \quad (14)$$

Теперь получим аналогичную оценку для *неявной* схемы, когда возмущениям подвержен только оператор A .

Рассмотрим два различных оператора A_1 и A_2 аналогично (5). Получим два оператора перехода R_1 и R_2 :

$$R_1 = E - \tau B^{-1}A_1, \quad R_2 = E - \tau B^{-1}A_2.$$

Как было сказано ранее, оценка для D_2 из (8) остаётся прежней, поэтому нас интересует оценка D_1 :

$$\begin{aligned} \|D_1\| &= \|E - \tau B^{-1}A_1 - E + \tau B^{-1}A_2\| = \\ &= \|\tau B^{-1}(A_2 - A_1)\| \leq \tau\|B^{-1}\| \cdot \|Q\| = 3\tau\varepsilon\|B^{-1}\|. \end{aligned} \quad (15)$$

Для оценки $\|B^{-1}\|$ воспользуемся тем, что B – монотонная матрица. Действительно, используя (3):

$$\begin{aligned}
B = E - \tau A &= \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \dots & 0 \\ 0 & -1 & 2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix} \frac{\tau}{h^2} = \\
&= \begin{pmatrix} 1-2r & r & 0 & \dots & 0 \\ r & 1-2r & r & \dots & 0 \\ 0 & r & 1-2r & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & r & 1-2r \end{pmatrix}.
\end{aligned}$$

В работе [3] представлена теорема, согласно которой для монотонной матрицы G

$$\|G^{-1}\|_{\infty} \leq \frac{1}{R_*(G)},$$

где $R_*(G)$ – минимальная величина диагонального преобладания матрицы G . Заметим, что здесь используется ∞ -норма, но для симметричной матрицы они взаимозаменяемы:

$$\|A\|_1 = \max_i \sum_j |a_{ij}| = \max_j \sum_i |a_{ij}| = \|A\|_{\infty}.$$

Вычислим $R_*(B)$:

$$R_*(B) = 1 + 2r - r - r = 1.$$

Таким образом

$$\|B^{-1}\| \leq \frac{1}{R_*(B)} = \frac{1}{1} = 1. \quad (16)$$

Подставляя (16) в (15):

$$\|D_1\| \leq 3\tau\varepsilon\|B^{-1}\| \leq 3\tau\varepsilon. \quad (17)$$

Так как оценка (17) совпадает с оценкой (13), а оценка (10) одинакова для обоих случаев, то делаем вывод, что для неявной схемы в случае $A_1 \neq A_2$ оценка возмущения одинакова и выражается (14).

Наконец рассмотрим случай, когда для неявной схемы $A_1 \equiv A_2, B_1 \neq B_2$. Имеем:

$$\begin{aligned}
R_1 &= E - \tau B_1^{-1}A, & R_2 &= E - \tau B_2^{-1}A, \\
B_2 &= B_1 + Q,
\end{aligned}$$

причём Q по-прежнему имеет вид (6).

Рассмотрим оценку D_1 :

$$\begin{aligned}
\|D_1\| &= \|E - \tau B_1^{-1}A - E + \tau B_2^{-1}A\| \leq \\
&\leq \|\tau A(B_2^{-1} - B_1^{-1})\| \leq \tau\|A\| \cdot \|B_2^{-1} - B_1^{-1}\|.
\end{aligned} \quad (18)$$

Легко вычислить норму A :

$$\|A\| = \max_i \sum_j |q_{ij}| = \max_i (2h^{-2} + h^{-2}, 2h^{-2} + h^{-2} + h^{-2}) = 4h^{-2}.$$

Подставляя её в (18):

$$\|D_1\| \leq \tau \cdot 4h^{-2} \|B_2^{-1} - B_1^{-1}\|.$$

Для оценки $\|B_2^{-1} - B_1^{-1}\|$ используем теорему о матрице, обратной сумме матриц [4], в частности её формой как тождества Хуа [5]:

$$(A + B)^{-1} = A^{-1} - (A + AB^{-1}A)^{-1}.$$

Оценим

$$\begin{aligned} \|B_2^{-1} - B_1^{-1}\| &= \|(B_1 + Q)^{-1} - B_1^{-1}\| = \|B_1^{-1} - (B_1 + B_1Q^{-1}B_1)^{-1} - B_1^{-1}\| = \\ &= \|(B_1 + B_1Q^{-1}B_1)^{-1}\| = \|(B_1(E + Q^{-1}B_1))^{-1}\|. \end{aligned}$$

По свойству обратной матрицы:

$$\|(B_1(E + Q^{-1}B_1))^{-1}\| = \|(E + Q^{-1}B_1)^{-1}B_1^{-1}\| \leq \|(E + Q^{-1}B_1)^{-1}\| \cdot \|B_1^{-1}\|$$

Оценка нормы невозмущенной матрицы B представлена выражением (16). Подставим её вместо $\|B^{-1}\|$:

$$\begin{aligned} \|(E + Q^{-1}B_1)^{-1}\| \cdot \|B_1^{-1}\| &\leq \|(E + Q^{-1}B_1)^{-1}\| = \|(Q^{-1}(B_1 + Q))^{-1}\| = \\ &= \|(B_1 + Q)^{-1}Q\| \leq \|(B_1 + Q)^{-1}\| \cdot \|Q\|. \end{aligned}$$

Используя оценку нормы оператора Q , представленной (12), имеем

$$\|(B_1 + Q)^{-1}\| \cdot \|Q\| \leq 3\varepsilon \|(B_1 + Q)^{-1}\|. \quad (19)$$

Для оценки $\|(B_1 + Q)^{-1}\|$ снова воспользуемся теоремой из [3]. Чтобы однозначно вычислить $R_*(B_1 + Q)$, введём условие

$$r > \varepsilon. \quad (20)$$

Тогда

$$\begin{aligned} R_*(B_1 + Q) &= 1 + 2r + \varepsilon - (r - \varepsilon) - (r - \varepsilon) = \\ &= 1 + 2r + \varepsilon - 2r + 2\varepsilon = 1 + 3\varepsilon. \end{aligned}$$

Таким образом

$$\|(B_1 + Q)^{-1}\| \leq \frac{1}{R_*(B_1 + Q)} = \frac{1}{1 + 3\varepsilon}.$$

Подставляя в (19), получаем

$$\|(B_1 + Q)^{-1}\| \cdot \|Q\| \leq 3\varepsilon \|(B_1 + Q)^{-1}\| \leq 3\varepsilon \cdot \frac{1}{1 + 3\varepsilon} \leq 3\varepsilon.$$

Объединяя рассуждения выше:

$$\|D_1\| \leq \tau \cdot 4h^{-2} \|B_2^{-1} - B_1^{-1}\| \leq \tau \cdot 4h^{-2} \cdot 3\varepsilon = 12\tau\varepsilon h^{-2}.$$

Выполняя замену $\tau = rh^2$ и объединяя с оценкой нормы D_2 :

$$\|R_1^n - R_2^n\| = \|D_1 D_2\| \leq 12Mr\varepsilon h^{-1} \leq Ch^l,$$

или

$$\varepsilon \leq \tilde{C}h^{l+1}, \quad \tilde{C} = \frac{c}{12Mr}. \quad (21)$$

Проверим корректность полученных оценок в рамках численного эксперимента. Рассмотрим краевую задачу (2), для которой

$$\begin{aligned} \phi(x) = \sin \pi x, \quad \psi_1(t) \equiv 0, \quad \psi_2(t) \equiv 0, \quad L = 1, \\ h = 0.01, \quad r = 0.4. \end{aligned}$$

Эта задача имеет точное решение:

$$u(x, t) = e^{-\pi^2 t} \sin \pi x.$$

Для решения с помощью неявной схемы будем использовать обычный метод прогонки. Явная схема имеет первый порядок аппроксимации ($l = 1$), поэтому (14) для явной схемы имеет вид

$$\varepsilon \leq \tilde{C}h^0 = \tilde{C}.$$

Неявная схема имеет второй порядок аппроксимации ($l = 2$), поэтому (14) для неявной схемы ($A_1 \neq A_2$) имеет вид

$$\varepsilon \leq \tilde{C}h,$$

а (21) для неявной схемы ($B_1 \neq B_2$) имеет вид

$$\varepsilon \leq \tilde{C}h^3.$$

Таблица 1

Результаты численного эксперимента

Схема	ε	$\ R_1^n - R_2^n\ $	Итог
явная	0.000000	0.000004	эквивалентны
явная	0.000001	0.000004	эквивалентны
явная	0.000100	0.000006	эквивалентны
явная	0.010000	0.000120	эквивалентны
явная	0.100000	0.001157	эквивалентны
явная	1.000000	0.011472	эквивалентны
неявная ($A_1 \neq A_2$)	0.000000	0.000011	эквивалентны
неявная ($A_1 \neq A_2$)	0.000001	0.000011	эквивалентны
неявная ($A_1 \neq A_2$)	0.000100	0.000010	эквивалентны
неявная ($A_1 \neq A_2$)	0.010000	0.000028	эквивалентны
неявная ($A_1 \neq A_2$)	0.100000	0.000374	эквивалентны
неявная ($A_1 \neq A_2$)	1.000000	0.003827	неэквивалентны
неявная ($B_1 \neq B_2$)	0.000000	0.000011	эквивалентны
неявная ($B_1 \neq B_2$)	0.000001	0.000181	эквивалентны
неявная ($B_1 \neq B_2$)	0.000100	0.019000	неэквивалентны
неявная ($B_1 \neq B_2$)	0.010000	0.825713	неэквивалентны
неявная ($B_1 \neq B_2$)	0.100000	0.983464	неэквивалентны

В рамках эксперимента будем рассматривать возмущения

$$\varepsilon = [0, h^3, h^2, h, h^{0.5}, h^0] = [0, 0.000001, 0.0001, 0.01, 0.1, 1.0],$$

но заметим, что в виду (20) для неявной схемы ($B_1 \neq B_2$) мы не можем полагать $\varepsilon = 1.0$.

Результаты эксперимента представлены в таблице 1. Как можно видеть, все полученные оценки корректны.

ЛИТЕРАТУРА

1. Самарский А. А. Теория разностных схем. – М.: Наука, 1977. – 656 с.
2. Мурюмин С. М., Язовцева О. С. Асимптотическая эквивалентность разностных схем для решения задачи Коши [Электронный ресурс] // Дифференциальные уравнения и их приложения в математическом моделировании: материалы XIII Международной научной конференции. (Саранск, 12-16 июля 2017 г.). – Саранск: СВМО, 2017. – С. 491–497. – Режим доступа: <http://conf.svmo.ru/files/deamm2017/papers/paper67.pdf> (дата обращения 12.04.2019).
3. Волков Ю. С., Мирошниченко В. Л. Оценки норм матриц, обратных к матрицам монотонного вида и вполне неотрицательным матрицам // Сиб. матем. журн. – 2009. – Т.50, № 6. – С. 1248–1254.
4. Henderson H. V., Searle S. R. On Deriving the Inverse of a Sum of Matrices // SIAM Review. – 1981. – Vol. 23, No. 1. – P. 53–60.
5. Cohn P. M. Further Algebra and Applications. – London: Springer-Verlag, 2003. – 451 p.

КОКНАЕВ Н. А., ФИРСОВА С. А.

**РАЗРАБОТКА КОМПЛЕКСА ПРОГРАММ АВТОМАТИЗАЦИИ ОБРАБОТКИ
РЕЗУЛЬТАТОВ ИСПЫТАНИЙ В ЛАБОРАТОРИИ МЕХАНИЧЕСКИХ ИСПЫТАНИЙ**

Аннотация. В статье рассмотрен созданный авторами комплекс программ, который выполняет автоматизированную обработку результатов механических испытаний, проведенных на вибрационном стенде: вычисление оптимальных весовых коэффициентов для приближения фактического ускорения, зафиксированного на устройстве, к целевому ускорению; нахождение резонансных частот, зафиксированных при испытании тестируемого изделия.

Ключевые слова: механические испытания, вибрационный стенд, Econ Technologies, Vibration Control System, весовые коэффициенты, резонансные частоты.

KOKNAEV N. A., FIRSOVA S. A.

DEVELOPMENT OF SOFTWARE SYSTEM

FOR AUTOMATED PROCESSING OF MECHANICAL TEST RESULTS

Abstract. The article considers a software system developed by the authors, which performs automated processing of the results of mechanical tests carried out on the vibration stand: calculation of the optimal weight coefficients to approximate the actual acceleration recorded on the device to the target acceleration; finding the resonant frequencies recorded during the test of the tested product.

Keywords: mechanical tests, vibration stand, Econ Technologies, Vibration Control System, weight coefficients, resonant frequencies.

Введение. В настоящее время Центр проектирования инноваций (АУ «Технопарк – Мордовия», Республика Мордовия) осуществляет проведение испытаний для целей сертификации. В состав научно-испытательного центра ЦПИ входят 3 лаборатории: электромагнитной совместимости, климатических испытаний, механических испытаний [1].

Лаборатория механических испытаний оснащена вибрационными тестовыми системами ETS-Solutions MPA714/M748/VT1000M, в состав которых входит вибрационный стенд, предназначенный для испытания образцов на устойчивость к вибрациям в вертикальной плоскости (для моделирования горизонтальных вибраций применяются горизонтальные столы).

Работа вибрационного стенда заключается в проверке образца на устойчивость к вибрациям в заданном режиме. Для этого на стенде закреплен один датчик, фиксирующий значение ускорения на самой установке и требуемое количество датчиков на образце,

фиксирующих значения ускорений на каждом из участков образца. На основе показаний датчиков вычисляется результирующее контрольное ускорение изделия. Для этого к вибрационному стенду подключен контроллер, который считывает контрольные локальные ускорения с датчиков, подключенных к тестируемому изделию, и передает считываемые ускорения в программную систему, в которой производится первоначальный расчет результирующего контрольного ускорения. Чем ближе результирующее контрольное ускорение образца к целевому ускорению, тем точнее будет результат.

Если известно, что на какой-то из участков тестируемого изделия оказывается малое вибрационное воздействие, то весовой коэффициент увеличивается, если вибрационное воздействие слишком велико, то наоборот – уменьшается. Вибрационное воздействие должно быть равномерным, иначе пострадает точность проводимого испытания. Поэтому актуальной задачей становится разработка комплекса программ для автоматизированного расчета, который заключается в нахождении оптимальных весовых коэффициентов.

Программное обеспечение, взаимодействующее с вибрационным стендом. Для взаимодействия с вибрационным стендом компанией Ecom Technologies была разработана программная система Vibration Control System [2]. При запуске программной системы предлагается выбрать тип испытания (см. рис. 1).

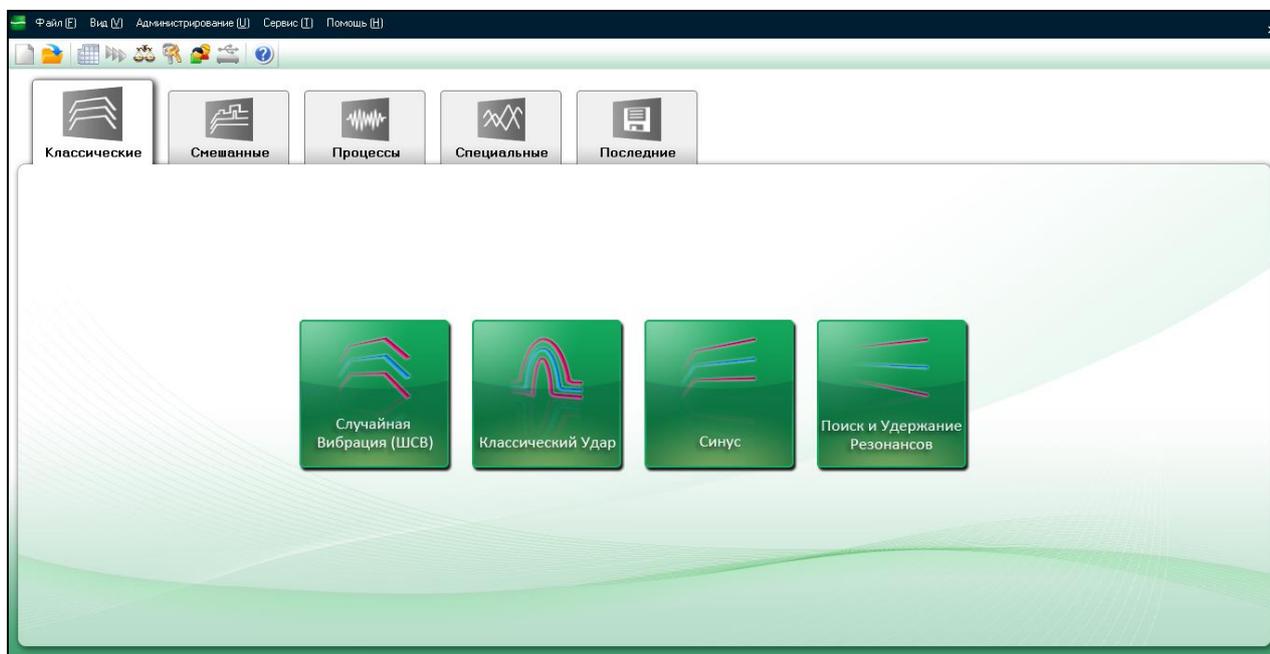


Рис. 1. Начальное окно программы Vibration Control System.

После выбора типа испытания, например, «Синус», открывается окно, в котором показан график зависимости ускорения от частоты. Например, в конфигурации, соответствующей рисунку 2, показаны значения ускорений, зафиксированные на четырех датчиках, закрепленных на тестируемом образце и значение целевого ускорения.

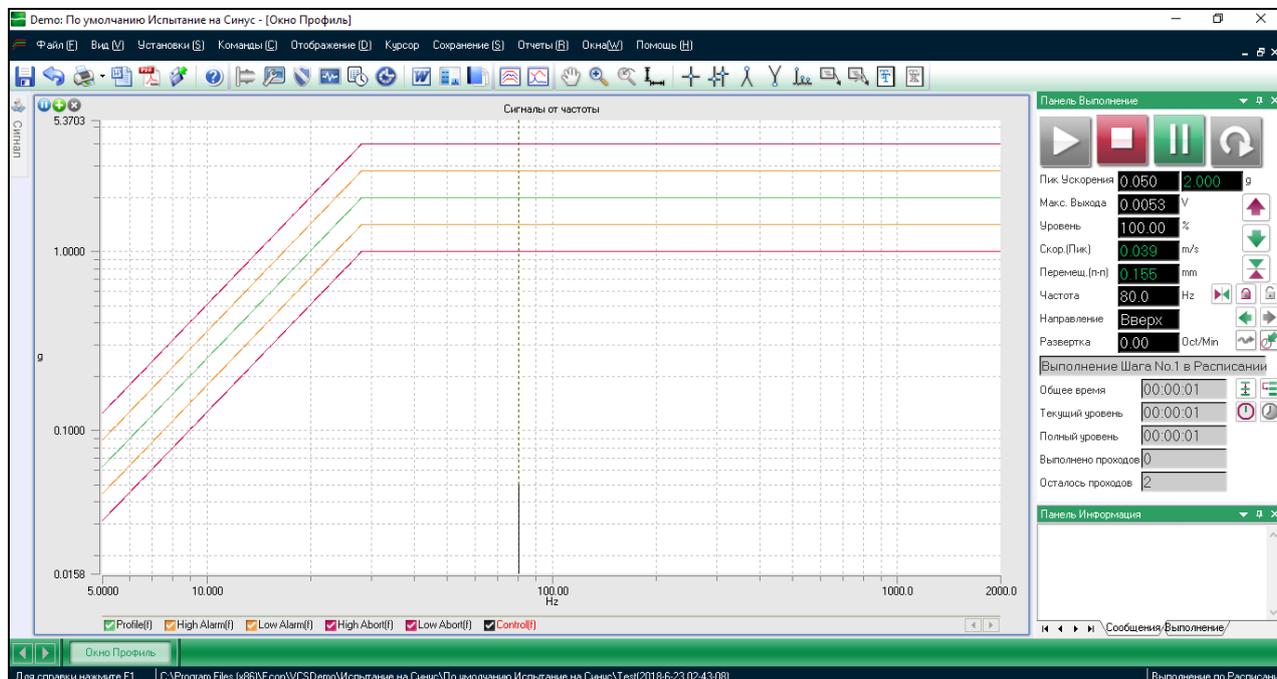


Рис. 2. График испытания устройства на синус.

В результате испытания формируется отчет о работе станда, фрагменты которого продемонстрированы на рисунках 3–6. Так, в таблице, представленной на рисунке 3, показаны максимальные значения ускорений, скоростей и смещений. Целевые значения вышеперечисленных критериев показаны сверху, значения, зафиксированные на станде, – снизу.

Profile Parameters	
Profile	
Profile Acceleration's Peak	1 g
Profile Velocity's Peak	0.0699803 m/s
Profile Displacement's Peak-Peak	1 mm
Shaker acceleration(peak)	34 g
Shaker velocity (peak)	1.8 m/s
Shaker displacement (peak-peak)	51 mm

Рис. 3. Отчет о проведенном испытании (начало).

Таблица на рисунке 4 отображает целевые значения частоты, ускорения, скорости и смещения. Все вышеперечисленные значения зафиксированы в начале испытания, после преломления ускорения (при достижении максимального ускорения) и в конце испытания.

ECON									
Profile table parameters									
Frequency(Hz)	Acceleration (g)	Velocity (m/s)	Displacement (Peak-Peak)(mm)	Left slope(dB/Oct)	Right slope(dB/Oct)	High Alarm(dB)	Low Alarm(dB)	High Abort(dB)	Low Abort(dB)
10	0.201421	0.0314159	1		Constant Disp.	3	-3	6	-6
22.2817	1	0.07	1	Constant Disp.	Constant Acc.	3	-3	6	-6
100	1	0.0155972	0.0496474	Constant Acc.		3	-3	6	-6

Рис. 4. Отчет о проведенном испытании (продолжение).

На рисунке 5 представлен график, отражающий результирующее целевое значение ускорения и целевые значения ускорений, зафиксированные на образце.

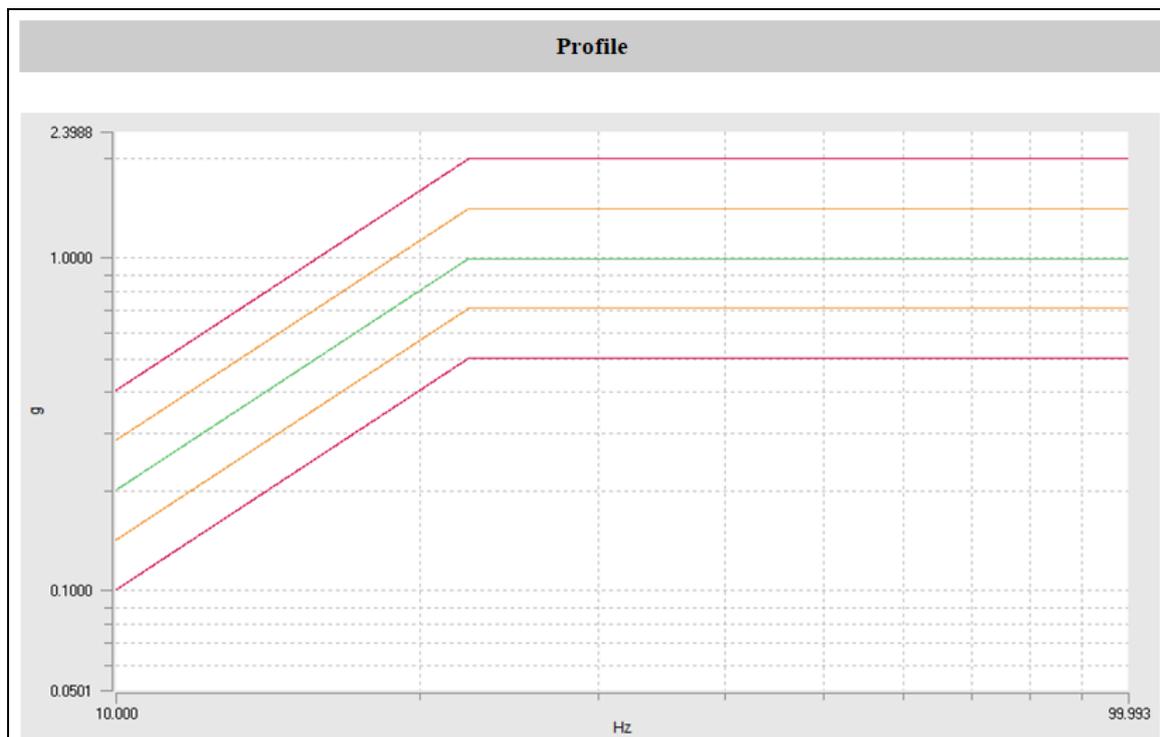


Рис. 5. Отчет о проведенном испытании (продолжение).

На рисунке 6 показаны контрольные значения ускорений, полученные в ходе испытания, а также значение целевого ускорения.

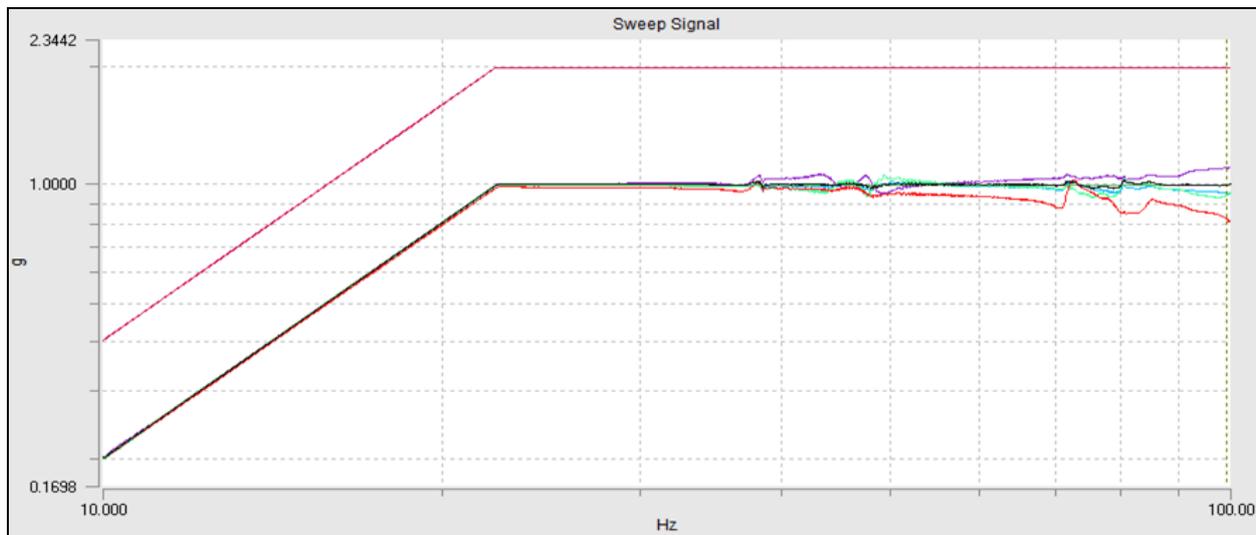


Рис. 6. Отчет о проведенном испытании (окончание).

Результат проведенного испытания сохраняется в файл формата .xls. Данные из этого файла будут использоваться в качестве исходных для работы программного комплекса. Фрагмент файла продемонстрирован на рисунке 7.

	A	B	C	D	E	F	G
1	Frequency	Control1	Control2	Control3	Stand	ResultantControl	Target
2	X (Hz)	Y (g)	Y (g)				
3	10	0,201638	0,203157	0,200102	0,200814	0,201632	0,201421
4	10,0113	0,20157	0,203231	0,200096	0,200807	0,201632	0,201874
5	10,0225	0,201575	0,203243	0,200106	0,200795	0,201641	0,202329
6	10,0338	0,201839	0,203353	0,200103	0,200887	0,201765	0,202785
7	10,0451	0,202142	0,203804	0,200583	0,2013	0,202177	0,203241
8	10,0564	0,202553	0,204367	0,201253	0,201864	0,202725	0,203699
9	10,0677	0,203103	0,204952	0,201835	0,202438	0,203297	0,204158
10	10,0791	0,20371	0,205562	0,202403	0,202976	0,203892	0,204618
11	10,0904	0,204364	0,206245	0,203138	0,203706	0,204582	0,205078
12	10,1018	0,204904	0,206809	0,203633	0,204252	0,205115	0,20554
13	10,1131	0,205488	0,207315	0,204027	0,204749	0,20561	0,206003
14	10,1245	0,206006	0,207786	0,20434	0,205127	0,206044	0,206467

Рис.7. Результат проведенного испытания.

В первом столбце таблицы хранятся частоты, воздействующие на образец, во втором, третьем и четвертом столбцах показаны ускорения, зафиксированные на первом, втором и третьем датчиках, закрепленных на образце, в пятом столбце показаны ускорения, зафиксированные на датчике, закрепленном на стенде, в шестом столбце показано результирующее контрольное ускорение, а в седьмом – целевое ускорение.

Разработка диаграммы вариантов использования. Комплекс программ, реализующий автоматизированную обработку результатов испытаний в лаборатории механических испытаний ЦПИ, представляет собой программное обеспечение для расчета весовых коэффициентов и резонансных частот и систему с архитектурой клиент-сервер, которая объединяет аппаратную часть системы, представляющую собой компьютер, фиксирующий и форматирующий результаты испытаний, компьютер-сервер, получающий форматированную информацию, и компьютеры-клиенты, на которых производятся расчеты.

Взаимоотношения актеров и прецедентов представлены на диаграмме вариантов использования, продемонстрированной на рисунке 8.

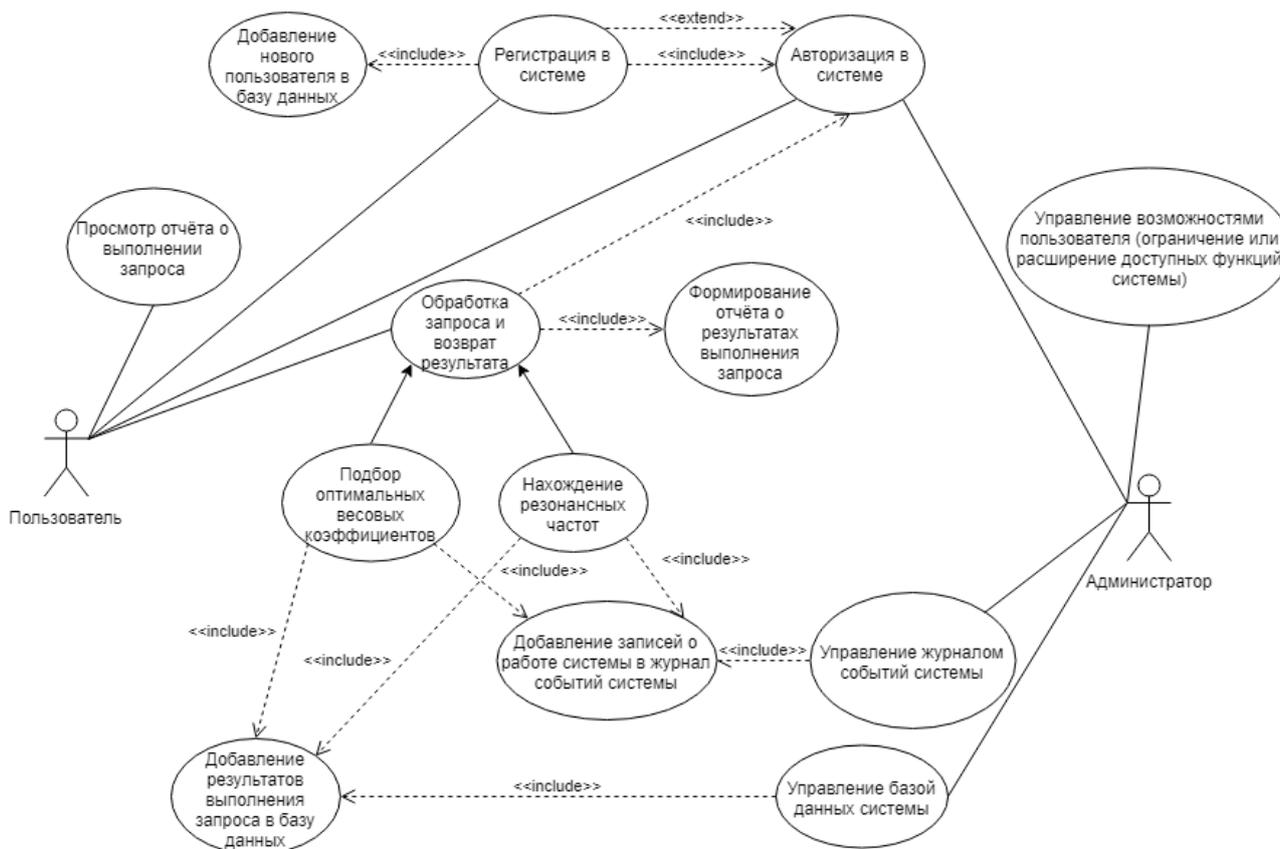


Рис. 8. Диаграмма вариантов использования.

Разработка логической модели. Логическая модель включает описание и диаграммы наиболее важных модулей системы, описания значимых классов [3].

На рисунке 9 представлена диаграмма классов системы.

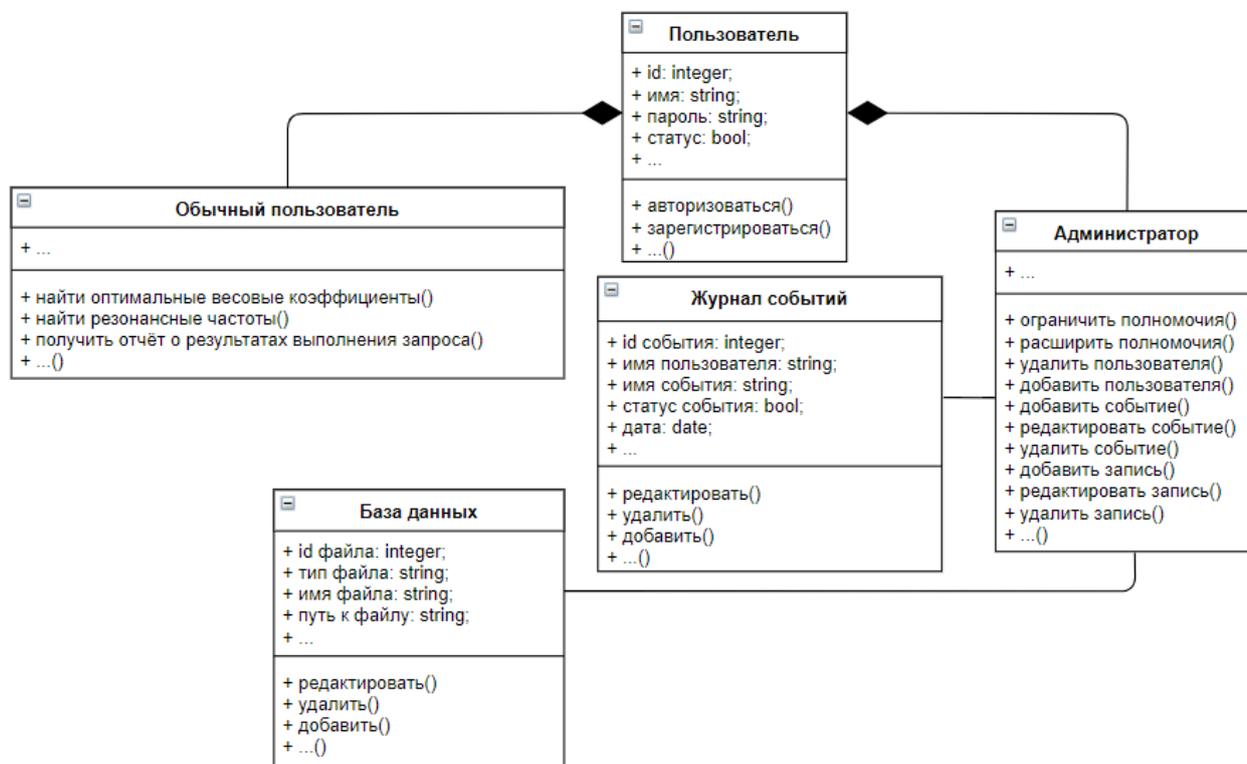


Рис. 9. Диаграмма классов.

Разработка модели размещения. Модель размещения описывает варианты физического размещения элементов системы, требования к оборудованию [3]. Для отображения варианта физического размещения элементов комплекса программ автоматизации обработки результатов испытаний использована диаграмма развертывания, представленная на рисунке 10.

Как видно из диаграммы развертывания, результаты испытаний изначально сохраняются на компьютере, взаимодействующим с вибрационным стендом. На этом компьютере установлено специализированное ПО для получения результатов испытаний в нужном формате. Далее результаты испытаний передаются на компьютер-сервер, который сохраняет все результаты испытаний в базу данных. База данных создана и в настоящий момент управляется СУБД MS SQL Server 2017. Программное обеспечение, установленное на компьютере-сервере, ориентировано на работу с базой данных, а ПО на компьютерах-клиентах ориентировано непосредственно на выполнение вычислений, а также на работу с базой данных. База данных едина и для клиента, и для сервера, что позволяет обращаться к

результатам испытаний и к результатам выполнения запросов с любого компьютера в рамках данной сети.

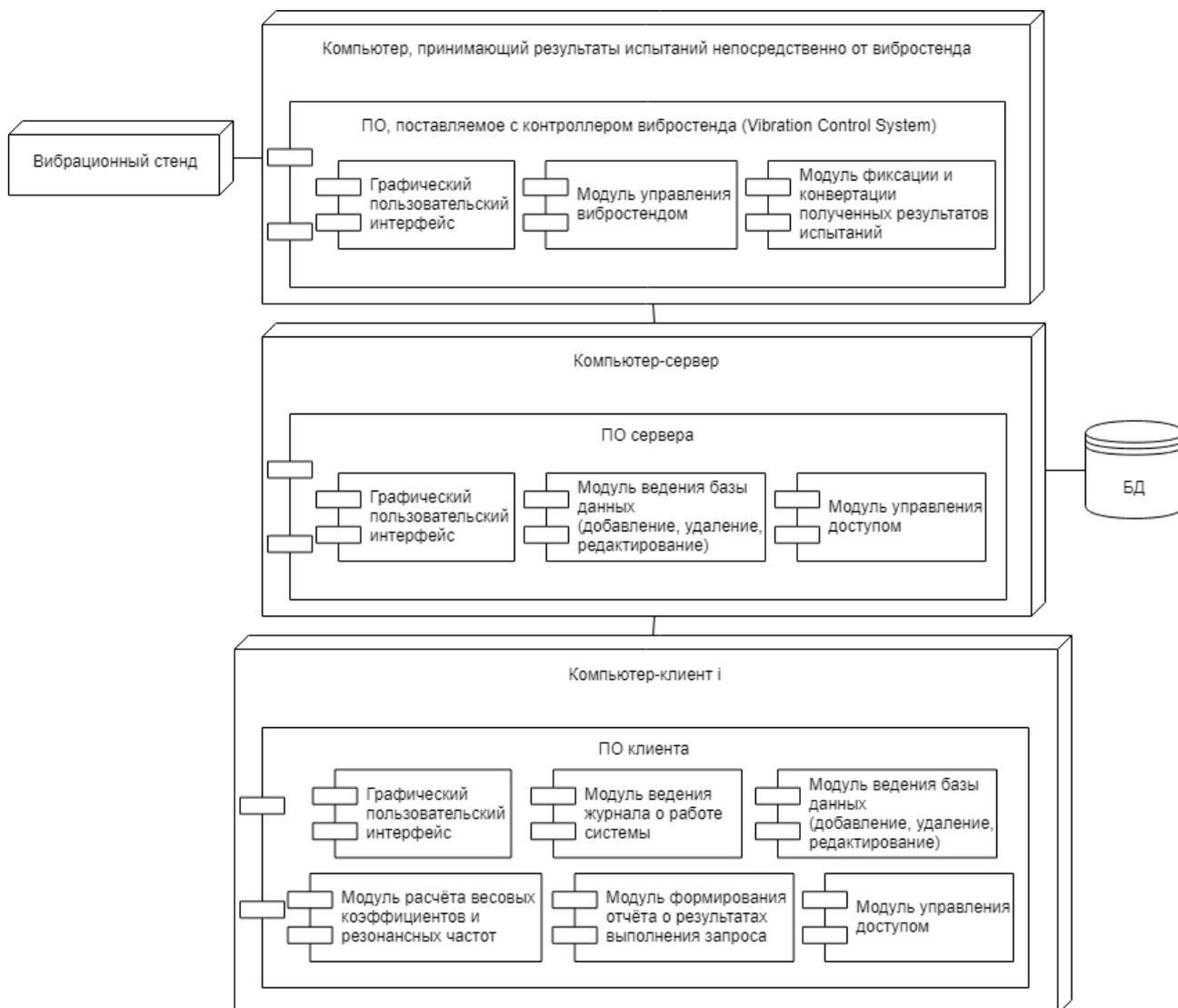


Рис. 10. Диаграмма развертывания.

Разработка модели реализации. Модель реализации описывает разделение системы на отдельные компоненты, независимые задачи, подпрограммы, информационные и управляющие потоки и связи между элементами системы [4].

Для представления модели реализации используется диаграмма компонентов, представленная на рисунке 11.

Архитектура комплекса программ автоматизации обработки результатов испытаний делится на три уровня:

- уровень представления информации;
- уровень логики;
- уровень данных.

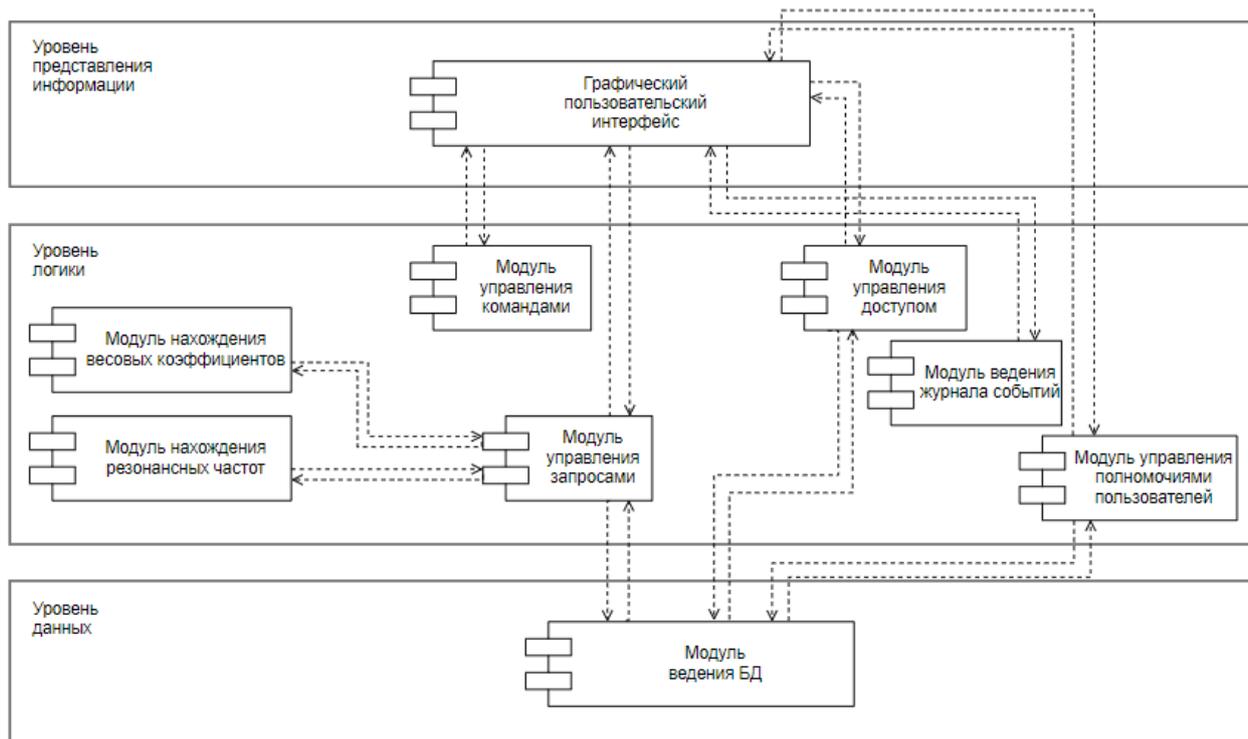


Рис. 11. Диаграмма компонентов.

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

Все вычисления системы производятся на уровне логики. Здесь обрабатываются команды и запросы, принимаемые от пользователя, формируются отчеты, осуществляются расчеты весовых коэффициентов и резонансных частот.

Уровень данных содержит базу данных, в которой находятся все данные о пользователях, результатах испытаний и других данных, получаемых в результате выполнения запросов или вводимых администратором системы.

Заключение. В лаборатории механических испытаний Центра проектирования инноваций (АУ «Технопарк – Мордовия», Республика Мордовия) проводятся испытания на вибрационном стенде МРА714/М748/ВТ1000М, в ходе которых требуется осуществление непосредственного контроля оператора над работой программной системы Vibration Control System, взаимодействующей с вибростендом. Например, оператор должен производить вычисление и подстановку оптимальных начальных значений (в данном случае весовых коэффициентов) вручную с целью достижения целевого ускорения. Разрабатываемый комплекс программ предполагает автоматизированный расчет и подстановку оптимальных значений, что существенно повысит точность и скорость проведения испытаний. Кроме того, выполняется нахождение резонансных частот устройства, если таковые имеются.

ЛИТЕРАТУРА

1. Центр проектирования инноваций (ЦПИ). Лаборатория механических испытаний [Электронный ресурс]. – Режим доступа: <http://www.technopark-mordovia.ru/design-center-of-innovation/#dci-09> (дата обращения: 04.06.2019).
2. ECON Technologies (Hangzhou) Co.,LTD Products [Электронный ресурс]. – Режим доступа: https://econtechology.en.ec21.com/Multi_axis_Vibration_Control_System-6877959_6877993.html (дата обращения: 04.06.2019).
3. ISO/IEC 19505-1:2012 – Information technology – Object Management Group Unified Modeling Language (OMG UML). – Part 1: Infrastructure [Электронный ресурс]. – Режим доступа: <https://www.iso.org/standard/32624.html> (дата обращения: 11.02.2018).
4. ISO/IEC 19505-1:2012 – Information technology – Object Management Group Unified Modeling Language (OMG UML). – Part 2: Superstructure [Электронный ресурс]. — Режим доступа: <https://www.iso.org/standard/52854.html> (дата обращения: 11.02.2018).

ПОЛЯЕВА И. А., МАМЕДОВА Т. Ф.

**АВТОМАТИЗАЦИЯ ПРОЦЕССА РАСПРЕДЕЛЕНИЯ И ПРОИЗВОДСТВА
ПРОДУКЦИИ ДЛЯ ПИЩЕВОЙ ПРОМЫШЛЕННОСТИ**

Аннотация. Рассматривается создание и автоматизация эффективного (оптимального) распределения производства продукции и управление этим процессом на основе программы «1С: Управление производственным предприятием».

Ключевые слова: пищевая промышленность, алгоритм, квадрат Пирсона, управление производством.

POLYAEVA I. A., MAMEDOVA T. F.

**AUTOMATION OF DISTRIBUTION AND PRODUCTION
OF PRODUCTS FOR FOOD INDUSTRY**

Abstract. The study deals with the development and automation of the effective (optimal) distribution of production and production management based on the software "1С: Manufacturing Enterprise Management".

Keywords: food industry, algorithm, Pearson square, production management.

Автоматизация различных технологических процессов в последние годы все глубже проникает в производственную деятельность. В этой связи актуальным вопросом является разработка нового программного обеспечения для пищевой промышленности. Очевидно, что предприятия пищевой промышленности должны работать эффективно и безопасно. Эффективность работы предприятия зависит от правильной организации труда, оптимизации всех расходов, четкого управления и моделирования всем процессом производства [1; 2].

Одна из главных задач, которая стоит перед технологами пищевой промышленности – это уметь правильно распределять поступающее на предприятие сырье, а также рассчитывать количество получаемых при этом продуктов. В настоящее время прикладных программ, в которых автоматизирован данный расчет, не существует [3].

Рассмотрим алгоритм решения таких задач, в основе которого лежит математическое моделирование с помощью квадрата Пирсона [4–6].

Введем обозначения:

m_1 – масса первого раствора,

m_2 – масса второго раствора,

x_1 – массовая доля растворенного вещества в первом растворе,

x_2 – массовая доля растворенного вещества во втором растворе,

x – массовая доля растворенного вещества в смеси первого и второго раствора.

Тогда будут справедливы следующие формулы:

$$m_1 x_1 + m_2 x_2 = x(m_1 + m_2),$$

$$m_1(x_1 - x) = m_2(x - x_2).$$

Рассмотрим геометрическую интерпретацию полученной модели (рис.1).

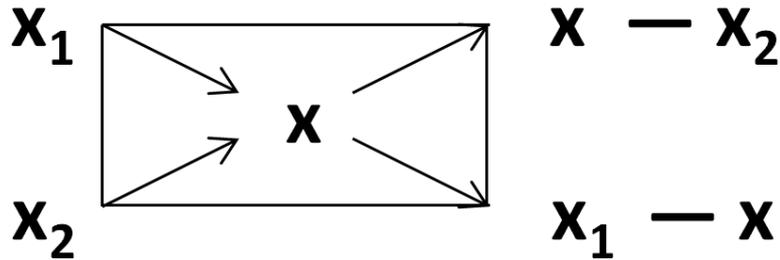


Рис. 1. Квадрат Пирсона.

Для применения моделирования по алгоритму эффективного (оптимального) распределения сырья по методу Пирсона рассмотрим молочную промышленность.

Наиболее общая схема технологических направлений переработки молока на городском молочном заводе представлена на рис. 2.

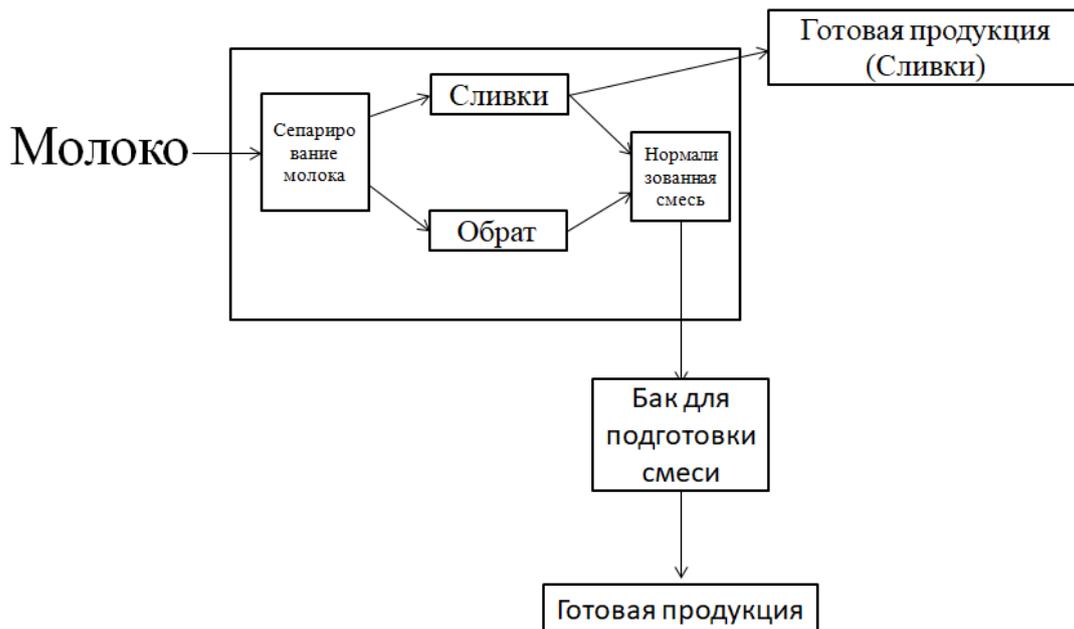


Рис. 2. Схема переработки молока на городском молочном заводе.

Молоко проходит сепарирование и из него получают сливки и обрат. Часть сливок выпускается уже как готовая продукция, часть идет дальше в производство и смешивается с обратом. Это соединение дает нам нормализованную смесь, которую дальше обрабатывают, и получается еще один готовый продукт.

Известно количество сливок как готовой продукции, которое получается в конце смены и известно какая их часть идет в производство, но необходимо определить какое количество молока использовалось для производства продукции за смену.

Программная реализации разрабатываемой системы была выполнена на базе программы «1С: Управление производственным предприятием». Программа является комплексным прикладным решением, охватывающим основные контуры управления и учета на производственном предприятии.

Конфигурация программы до внесения изменений представлена на рисунках 3 и 4.

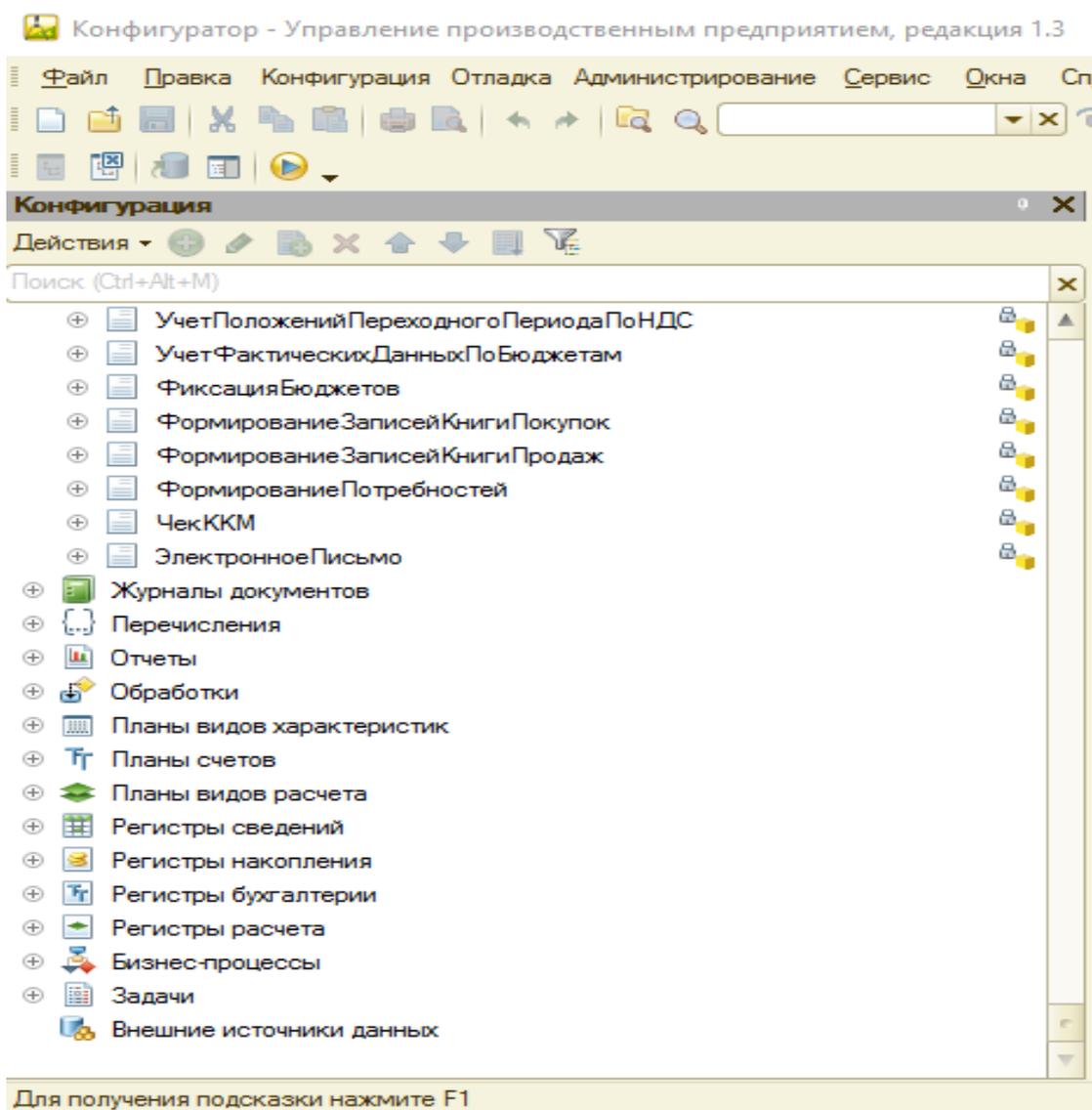


Рис. 3. Конфигурация «1С: Управление производственным предприятием».

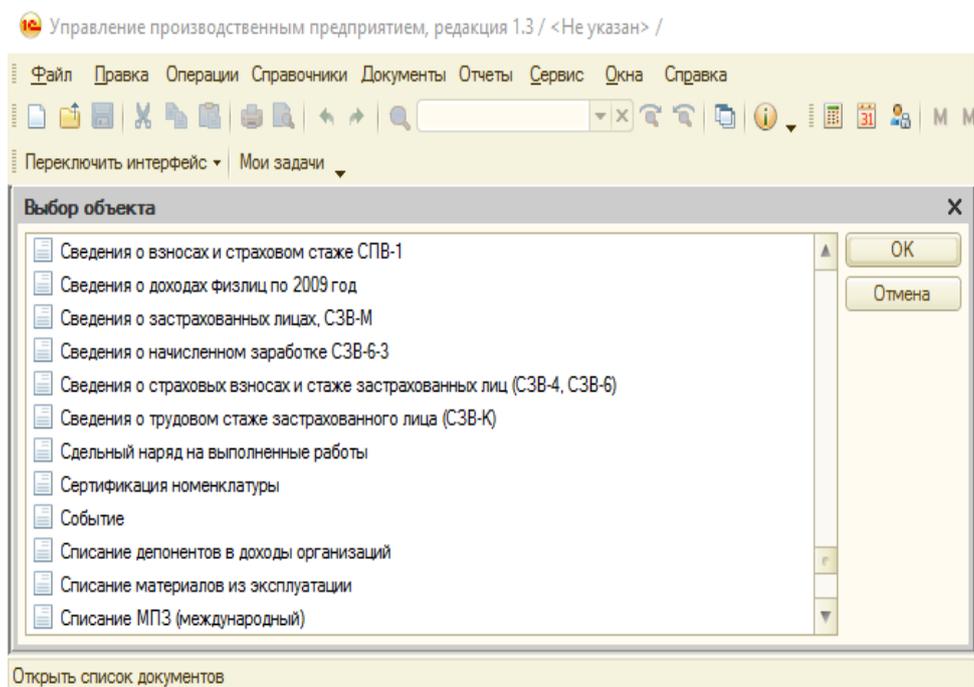


Рис. 4. Выбор объекта в «1С: Управление производственным предприятием».

Для автоматизации управления математической моделью эффективного (оптимального) распределения, получаемого для производства продукции сырья, в программу был добавлен документ «Сепарирование молока» (см. рис.5–6).

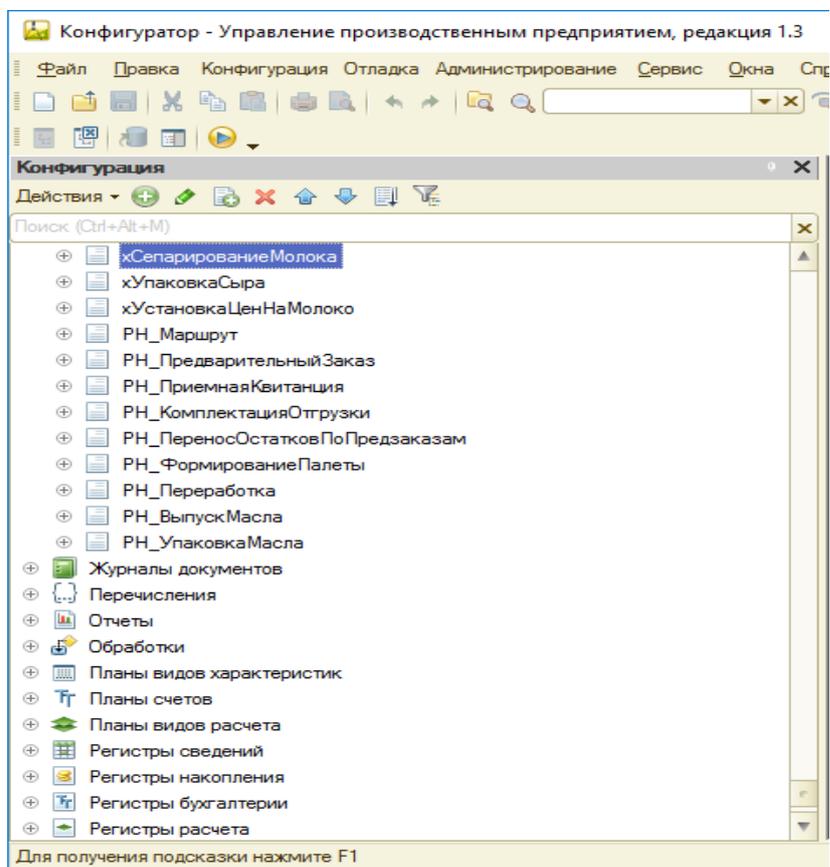


Рис. 5. Конфигурация. Документ «Сепарирование молока».

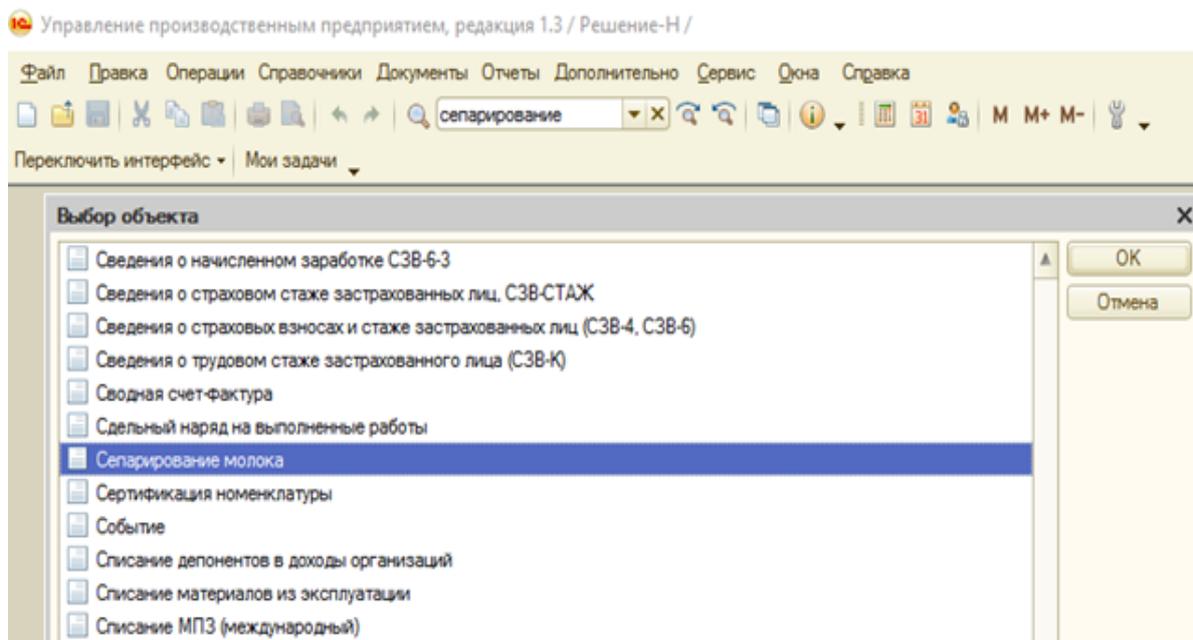


Рис. 6. Документ «Сепарирование молока».

Таким образом, представленное в статье программное обеспечение может быть применено для автоматизации управления процессом получения необходимого состава сырья для производства молочной продукции.

ЛИТЕРАТУРА

1. Минюк С. А., Ровба Е. А., Кузьмич К. К. Математические методы и модели в экономике. – Мн.: ТетраСистемс, 2002. – 432 с.
2. Бережной Е. В. Математические методы моделирования экономических систем. – 2-е изд., перераб. и доп. – М.: Финансы и статистика, 2006. – 432 с.
3. АКАДЕМИК [Электронный ресурс]. – Режим доступа: <https://dic.academic.ru/dic.nsf/ruwiki/1028588> (дата обращения 18.05.2019).
4. Мамедова Т. Ф., Шабанова В. Г., Шабанов Г. И. Принципы разработки модуля математико-экономической диагностики и прогноза предприятий АПК // Энергоэффективные и ресурсосберегающие технологии и системы. Межвузовский сборник научных трудов. – Саранск, 2017. – С. 218–222.
5. Мамедова Т. Ф., Шабанова В. Г. Управление производственным процессом по оптимальному критерию качества // Математическое моделирование, численные методы и комплексы программ: материалы VII Всероссийской научной

молодежной школы-семинара имени Е. В. Воскресенского с международным участием. – Саранск, 2016. – С. 102–104.

6. Мамедова Т. Ф., Каледин О. Е., Шабанова В. Г., Кирейчева Е. Ю. Математическая модель оптимизации управления хозяйственной деятельностью одного производственного предприятия // Математическое и компьютерное моделирование естественнонаучных и социальных проблем: материалы X Международной научно-технической конференции молодых специалистов, аспирантов и студентов / под ред. И. В. Бойкова. – Пенза, 2016. – С. 125–130.

КИСЕЛЕВ В. К., ФИРСОВА С. А.

ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

ПРОГРАММНО-ИНФОРМАЦИОННОЙ СИСТЕМЫ ПОИСКА ПОПУТЧИКОВ

Аннотация. В статье представлен анализ существующего программного обеспечения в области автоматизации поиска попутчиков. Приведены основные этапы моделирования, проектирования и реализации собственной программно-информационной системы, предназначенной для автоматизации взаимодействия между водителем и попутчиками.

Ключевые слова: проектирование, моделирование, унифицированный язык моделирования, программно-информационная система, автоматизация поиска попутчиков, карпулинг.

KISELEV V. K., FIRSOVA S. A.

DESIGNING AND IMPLEMENTING SOFTWARE

AND INFORMATION SYSTEM FOR RIDESHARING

Abstract. The article presents an overview of ridesharing software. The authors consider the main stages of modeling, designing and implementing their own software and information system designed to automate the interaction between the car driver and the travelling companion.

Keywords: designing, modeling, unified modeling language, software and information system, automation of travel companion search, carpooling.

Введение. Райдшеринг и карпулинг получили недавнее распространение в России, хотя эти термины пришли к нам с Запада – в семантике и по смыслу они означают «совместные поездки» (англ. *ride* – поездка, *share* – делиться, *car* – машина, *pool* – объединение) [1].

Идея этого движения состоит в том, чтобы для экономии топлива и денежных средств люди могли объединяться для дальних поездок, складываясь при этом на бензин. Благодаря всемирной паутине и GPS-технологии найти попутчиков сейчас несложно, однако не стоит забывать о мерах предосторожности и нюансах данного сервиса.

Совершение совместных поездок происходит по следующей схеме:

- водитель на специализированных сервисах заполняет анкету и выкладывает объявление о своей поездке;
- пассажиры находят эту поездку, связываются с водителем и обсуждают детали;
- попутчики вместе едут по определенному маршруту, при этом водитель экономит на топливе, пассажиры получают больший комфорт, чем в автобусе или поезде.

Если в Европе карпулинг широко распространен и поддерживается как современный и перспективный способ передвижения, то в России сервис только начинает внедряться. Конечно, есть ряд особенностей и нюансов, связанных именно с российским райдшарингом:

- пользователи сервиса не всегда ответственно и серьезно относятся к заполнению анкеты на сайте;
- отсутствие пунктуальности: попутчики могут в последний момент передумать и отказаться от поездки, даже не предупредив об этом заранее остальных участников.

Анализ информационных систем поиска попутчиков. Согласно сервису «Google» на территории России действует всего три сервиса для поиска автомобильных попутчиков, что несоизмеримо мало по сравнению с количеством подобных сервисов в Европе и США. Также стоит отметить, что на территориях большинства Европейских стран действуют локальные сервисы для поиска автомобильных попутчиков, которые не предназначены для использования по всей стране и рассчитаны только на маленькие города. Среди причин низкой распространенности и малого количества ресурсов для поиска попутчиков в России следует указать банальное незнание об их существовании; отсутствие знаний, необходимых для использования такого вида онлайн-сервисов, а также определенную степень недоверия к ним как со стороны водителя, так и со стороны пассажиров.

Проанализируем существующие решения в сфере сервисов по поиску попутчиков, представленных на территории Российской Федерации.

В первую очередь следует указать «VlaVlaCar», который является крупнейшим в мире международным онлайн-сервисом поиска автомобильных попутчиков, работающим в 21 стране мира [2]. Среди решений российских разработчиков первым стал «WeerCar», разработанный Mail.Ru Group в феврале 2017 [3]. Также существует сервис «Яндекс.Попутка», запущенный в 2018 году, но все еще не вышедший для всеобщего использования.

Проанализировав данные решения (см. таблицу 1), можно сделать вывод о том, что они либо мало функциональны и не соответствуют современным требованиям, либо являются очень сложными для использования простыми пользователями. Поэтому было принято решение создать программно-информационную систему «Карпул», которая будет сочетать в себе всю необходимую функциональность с адаптацией под среднестатистического водителя как будущего пользователя данным сервисом.

Сравнение функциональности существующих решений

Функциональность	BlaBlaCar	BeepCar	Яндекс.Попутка	Карпул
Веб-сайт	+	+	-	+
Приложение для телефона	+	-	+	-
Свободная регистрация	+	+	-	+
Перевозка грузов	-	-	-	+
Авторизация при помощи социальных сетей	+	-	-	+
Регистрация в качестве водителя или пассажира	+	-	-	+
Регистрация при помощи социальных сетей	+	-	-	+
Авторизация через социальные сети	+	-	-	+
Добавление информации о себе	+	+	+	+
Добавление фото	+	+	+	+
Добавление аккаунтов социальных сетей	+	+	-	+
Выбора места бронирования в авто	-	-	-	+
Добавление авто с фото и основной информацией	+	+	+	+
Добавление автобусов и газелей	-	-	-	+
Добавление промежуточных городов	+	+	+	+
Сортировка по цене	+	+	+	+
Сортировка по статусу авто	+	+	-	+
Сортировка по стажу водителя	+	+	-	+
Опция курящего салона	+	+	-	+
Опция перевозки животных	-	-	-	+
Опция разговорчивого водителя	+	+	-	+
Увеличения радиуса поиска доступных поездок	+	+	+	+
Новости сервиса	+	+	+	+
Блог сервиса	+	+	-	+
Наличие страницы помощи новым пользователем	+	+	-	+

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. Так, на рисунке 1 показано, что пользователями ПС являются водитель, попутчик, администратор, причем к основным вариантам использования ПС для водителя является добавление новых поездок,

подтверждение и отмена бронирований, для попутчика – поиск поездок и их бронирование, а для администратора – это основные административные функции. Стоит отметить, что изначально каждый пользователь системы является равноправным участником. Иными словами, каждый пользователь, включая администратора может не только создавать поездки, но и искать и бронировать их, т.е. каждый пользователь системы может одновременно быть и попутчиком, и водителем.



Рис. 1. Диаграмма вариантов использования.

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 показаны все существующие в данной автоматизированной системе связи, представленные в виде, соответствующему стандарту ISO/IEC 19505-1, 19505-2.

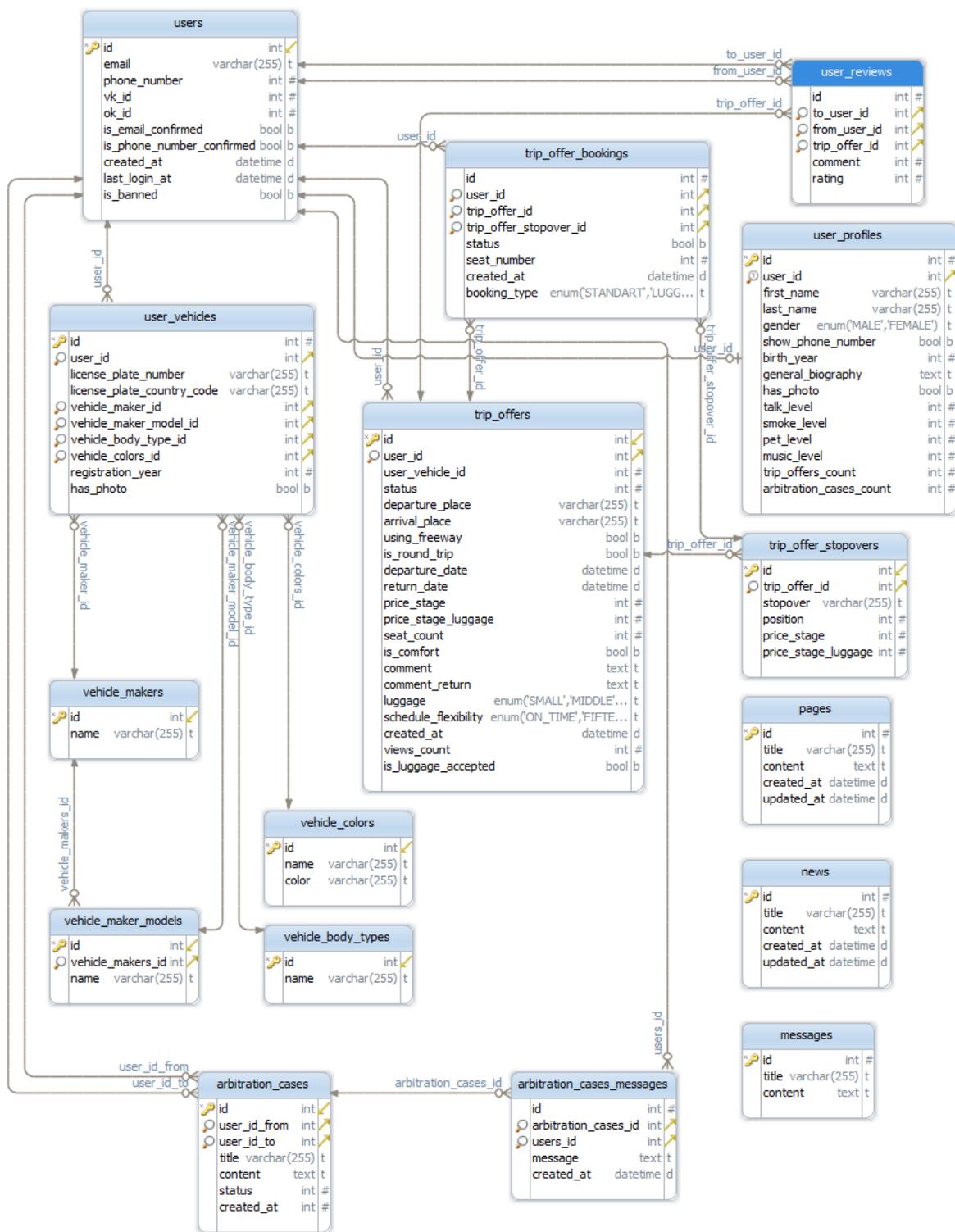


Рис. 2. Диаграмма классов.

На завершающей стадии проектирования ПС выполняется разработка диаграммы компонентов. Диаграмма компонентов – статическая структурная диаграмма, показывает 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. Так компонент PHP Word выполняет формирование документов по заранее созданным шаблонам для отчетности, а база данных и файловое хранилище предоставляют необходимые данные, будь то изображения, документы или иная информация. Созданная структура взаимосвязи компонентов представлена на рисунке 3.

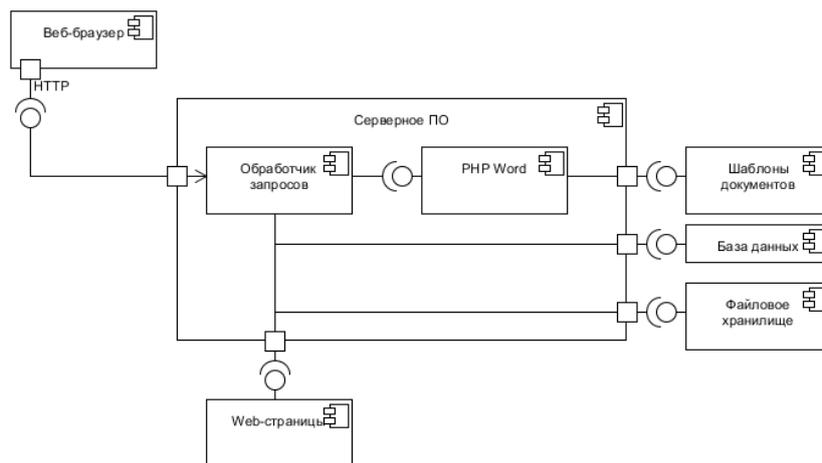


Рис. 3. Общая диаграмма компонентов.

Как было отмечено выше, ПС «Карупл» имеет клиент-серверную архитектуру, согласно которой 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 5 в диаграмме развертывания.

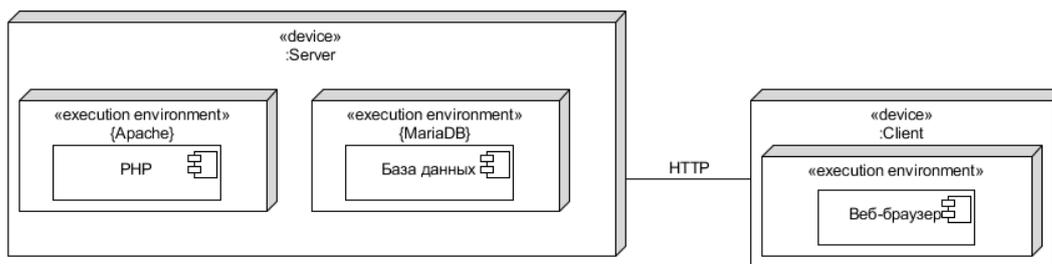


Рис. 4. Диаграмма развертывания.

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

- для создания и оформления страниц сайта – HTML, JavaScript, CSS, библиотеки jQuery, Twitter Bootstrap 4, jQuery-Color, Yandex.Maps.API;
- для работы с сессиями пользователей, взаимодействия с базами данных – язык программирования PHP-7 с применением веб-фреймворка Laravel;
- в качестве системы управления базами данных – MySQL;
- в качестве веб-сервера используется Apache с проксирующим Nginx 1.10.

Рассмотрим основные моменты работы с программно-информационной системой.

Изначально, когда пользователь попадает на сайт, его нельзя перегружать какой-либо информацией и обилием всевозможного функционала, поэтому в качестве главной страницы было сделано решение реализовать на ней посадочную страницу (так называемый landing page), показанную на рисунке 5.

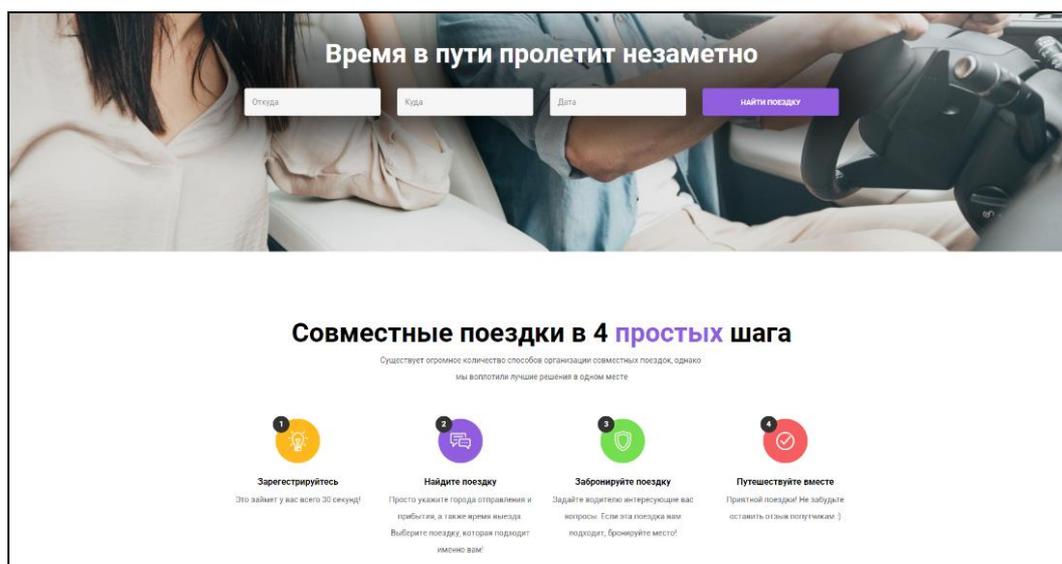


Рис. 5. Главная страница.

На сайте реализована возможность регистрации и авторизации как при помощи email, так и при помощи социальной сети ВКонтакте.

После авторизации пользователь попадает на свою публичную страницу, которая так же видна всем остальным пользователям сервиса. У пользователя существует возможность редактирования своей персональной информации, в том числе смены пароля на специальных страницах сервиса.

Каждый пользователь может предложить поездку. На странице создания поездки пользователю предлагается ввести пункт выезда и пункт приезда, а также добавить промежуточные пункты. Автоматически рассчитывается стоимость поездки и время прибытия. При вводе пользователем данных выезда, приезда и промежуточных пунктов автоматически не только подсказываются города и места, но и строится карта с полной информацией о предстоящей поездке (рис. 6).

откуда	куда	расстояние пути	время в пробках	время в пути без пробок
Россия, Республика Мордовия, Саранск	Россия, Санкт-Петербург	1400 км	23 ч 17 мин	18 ч 35 мин
Россия, Республика Мордовия, Саранск	Россия, Нижний Новгород	290 км	5 ч 46 мин	4 ч 1 мин
Россия, Нижний Новгород	Россия, Санкт-Петербург	1100 км	17 ч 31 мин	14 ч 34 мин

Рис. 6. Страница добавления поездки.

На странице поиска поездок пользователь может выбрать пункт выезда и приезда, а необходимые опции поиска поездки в случае необходимости (рис. 7).

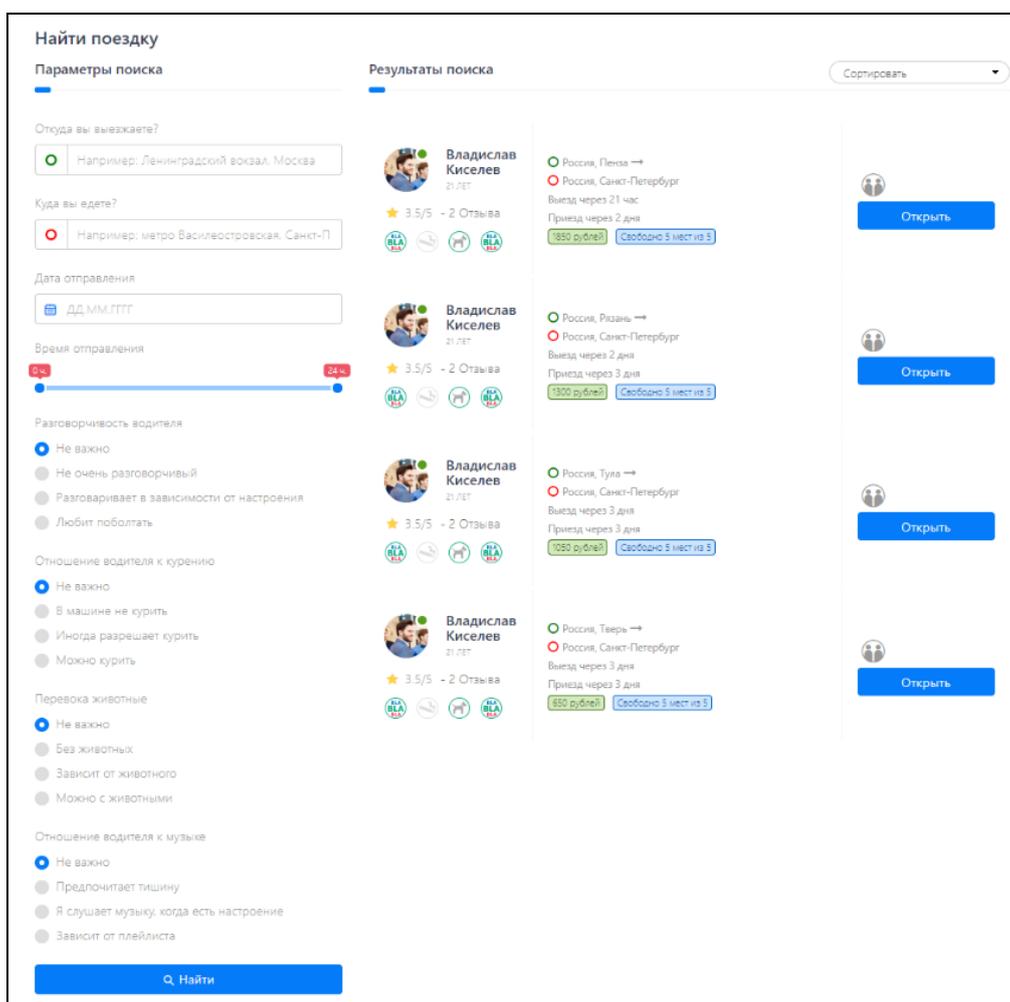


Рис. 7. Страница поиска поездки.

Заключение. Сравнение функциональности существующих программных продуктов и ПС «Карпул» представлено выше в таблице 1. Очевидно, что созданная программная система не уступает по функциональности рассмотренным аналогам и, кроме того, сочетает в себе высокую производительность, простоту использования, удобный лаконичный интерфейс. Для повышения конкурентоспособности ПС «Карпул» необходимо провести качественную рекламную кампанию, которая привлечет новую аудиторию.

ЛИТЕРАТУРА

1. Райдшеринг и карпулинг – что это такое? [Электронный ресурс]. – Режим доступа: <https://www.rudorogi.ru/articles/advice/ridesharing-carpulling.html> (дата обращения: 11.02.2018).
2. В дорогу с попутчиками без хлопот [Электронный ресурс]. – Режим доступа: <https://www.blablacar.ru/> (дата обращения: 11.02.2018).

3. Бипкар – поиск попутчиков, карпулинг, сервис райдшеринга Веерсар [Электронный ресурс]. – Режим доступа: <https://www.beersar.ru/> (дата обращения: 11.02.2018).
4. Яндекс.Попутка для водителей – сервис, который ищет попутчиков [Электронный ресурс]. – Режим доступа: <https://yandex.ru/promo/poputka> (дата обращения: 11.02.2018).
5. ISO/IEC 19505-1:2012 – Information technology – Object Management Group Unified Modeling Language (OMG UML). – Part 1: Infrastructure [Электронный ресурс]. – Режим доступа: <https://www.iso.org/standard/32624.html> (дата обращения: 11.02.2018).
6. ISO/IEC 19505-1:2012 – Information technology – Object Management Group Unified Modeling Language (OMG UML). – Part 2: Superstructure [Электронный ресурс]. – Режим доступа: <https://www.iso.org/standard/52854.html> (дата обращения: 11.02.2018).

БЕЛОВ В. Ф., ЕВДОКИМОВ А. С., КУМАКШЕВ А. С.

**ИНТЕГРАЦИЯ ОБОРУДОВАНИЯ РАЗЛИЧНЫХ ПРОИЗВОДИТЕЛЕЙ ДЛЯ
ПРОВЕДЕНИЯ ИСПЫТАНИЙ НА ЭЛЕКТРОМАГНИТНУЮ СОВМЕСТИМОСТЬ**

Аннотация. В статье описано применение программной оболочки EMC32, позволяющей интегрировать оборудование различных производителей для проведения испытаний на электромагнитную совместимость. Определена роль данной программной оболочки в системной архитектуре испытательного комплекса.

Ключевые слова: программное обеспечение, электромагнитная совместимость, программный комплекс, испытательный центр.

BELOV V. F., EVDOKIMOV A. S., KUMAKSHEV A. S.

**INTEGRATION OF EQUIPMENT OF DIFFERENT MANUFACTURERS
FOR ELECTROMAGNETIC COMPATIBILITY TESTING**

Abstract. The article describes the use of the software EMC32, which allows integrating equipment produced by different manufacturers for conducting tests of electromagnetic compatibility. The role of the test complex in the system architecture is defined.

Keywords: software, electromagnetic compatibility, software complex, test center.

На современном конкурентно-ориентированном рынке невозможно обойтись без средств автоматизации технологического процесса проведения испытаний. Автоматизация позволяет уменьшить время и повысить точность испытаний, что, несомненно, дает экономический эффект. Большинство производителей испытательного и измерительного оборудования разрабатывают собственное программное обеспечение (далее – ПО). Как правило, оно не способно конфигурировать аппаратные комплексы аналогичных конкурентных компаний. Однако среди широкого спектра программных комплексов выделяется ПО EMC32 от компании Rohde&Schwarz (Германия), способное интегрировать оборудование различных производителей.

ПО EMC32 представляет собой системное программное обеспечение для проведения испытаний на ЭМС, предназначенное для работы в операционных системах компании Microsoft. ПО EMC32 предоставляет стандартный пользовательский интерфейс для проведения испытаний на помехоэмиссию и помехоустойчивость, работа в котором подобна управлению виртуальным прибором. Он сочетает удобство интуитивно понятного графического пользовательского интерфейса, соответствующего действующим стандартам

Windows, с высочайшим уровнем гибкости, как в контексте поддержки измерительных приборов, так и в рамках проведения испытаний на ЭМС [1].

ПО EMC32 не только обеспечивает возможность выполнения сканирования по частоте в соответствии с требованиями соответствующих стандартов, но и позволяет проводить интерактивный анализ испытуемого устройства для оптимизации измерений, а также компоновать оборудование различных производителей исходя из требований заказчика (см. рис. 1).

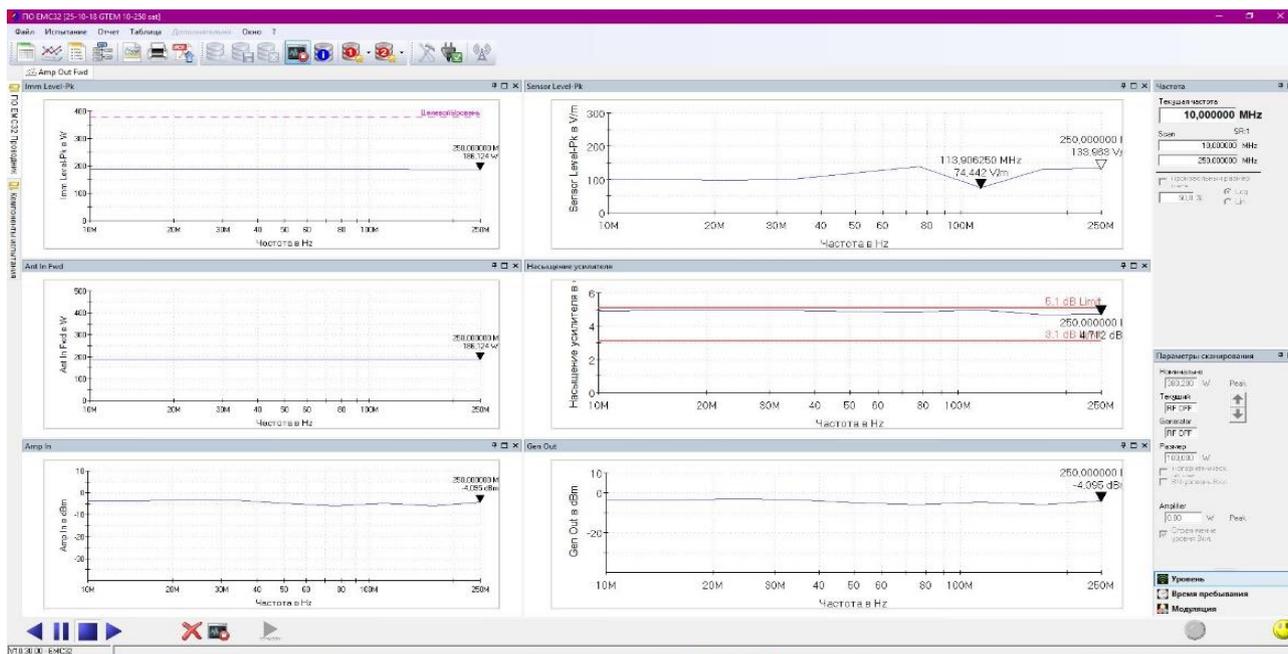


Рис. 1. Экран проведения испытания программного обеспечения EMC32.

Современные испытательные системы конфигурируются с применением аппаратно-программных платформ различных производителей. Исключительными свойствами ПО EMC32 являются возможности реализации на его основе платформенной независимости, взаимозаменяемости, масштабируемости и модульности [2]. EMC32 – достаточно гибкая программная среда, позволяющая интегрировать различные типы испытательного и измерительного оборудования для решения функциональных задач испытательного процесса. Можно утверждать, что EMC32 является ядром для конфигурирования испытательных комплексов по ЭМС с открытой архитектурой. Пример архитектуры такого испытательного комплекса приведен на рисунке 2. Эта архитектура реализуется в Центре проектирования инноваций (ЦПИ) АУ «Технопарк-Мордовия».

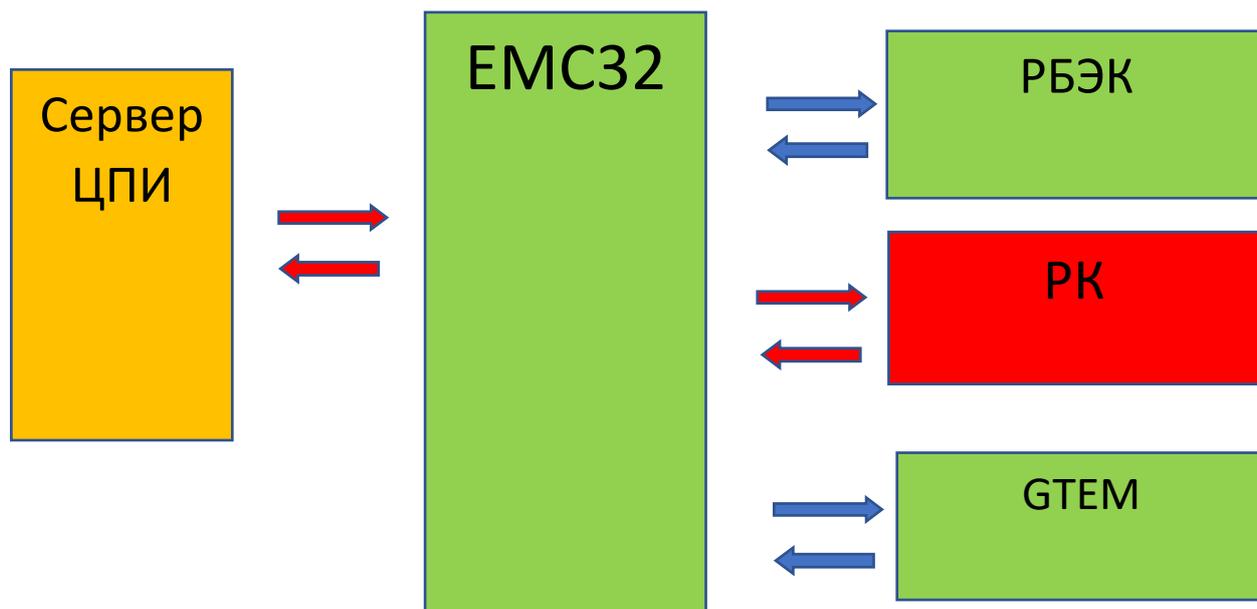


Рис. 2. Системная архитектура комплекса для испытаний на ЭМС.

На данном этапе проведен ряд работ, а именно налажено взаимодействие между элементами радиочастотной безэховой камеры (РБЭК), элементами высокочастотной камеры поперечной электромагнитной волны (ГТЕМ) и программным обеспечением EMC32 (рис. 3). На следующем этапе планируется подключение к комплексу реверберационной камеры (РК) и создание сети удаленного доступа с возможностью администрирования и мониторинга испытаний в режиме реального времени.

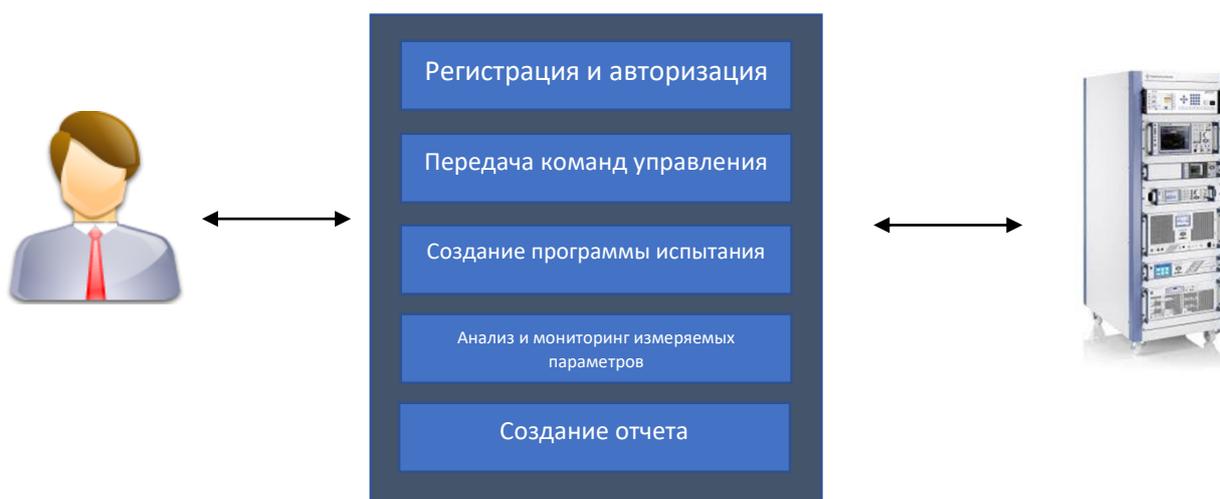


Рис. 3. Схема взаимодействия с испытательным комплексом.

Выводы.

1. Интеграция оборудования различных производителей в комплексы средств испытаний на ЭМС существенно расширит возможности вариантного анализа при выборе фирм-производителей оборудования и программного обеспечения, а также будет способствовать повышению конкурентоспособности ЦПИ.

2. Передача данных по испытательному процессу на сервер ЦПИ создаст условия для подключения комплекса средств испытаний на ЭМС к системе научных и испытательных центров.

ЛИТЕРАТУРА

1. Rohde & Schwarz GmbH & Co KG – Программное обеспечение для проведения испытаний на ЭМС EMC32, версия 9.0. – 2012. – 107 с.
2. Белов В., Петров В. Конфигурирование испытательной лаборатории ЭМС с открытой архитектурой // Электромагнитная совместимость в электронике: сборник статей. – 2018 – С. 148–150.

ЧЕКУРОВА А. А.
ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ
ПРОГРАММНО-ИНФОРМАЦИОННОЙ СИСТЕМЫ
АВТОМАТИЗАЦИИ РАБОТЫ ВЕТЕРИНАРНОЙ АПТЕКИ

Аннотация. В статье представлен анализ функциональности существующих информационных систем, служащих для автоматизации работы ветеринарных аптек и их сетей. На примере автоматизации работы ветеринарной аптеки рассмотрены базовые этапы разработки программно-информационной системы: моделирование, проектирование и непосредственная программная реализация компонентов.

Ключевые слова: проектирование, моделирование, автоматизированная информационная система, унифицированный язык моделирования, автоматизация работы ветеринарной аптеки.

CHEKUROVA A. A.
DESIGNING AND IMPLEMENTING SOFTWARE AND INFORMATION SYSTEM
FOR AUTOMATION OF VETERINARY PHARMACY

Abstract. The article provides an analysis of the functionality of automated information systems for veterinary pharmacies and their chains. Doing a project of the automation of a veterinary pharmacy, the author describes the basic stages of software development for the information system: modeling, designing and implementing of the system components.

Keywords: design, modeling, automated information system, unified modeling language, automation of veterinary pharmacy.

Введение. По данным новостного ресурса «Интерфакс», численность домашних животных в России за последние три года выросла на 14% или на 6,3 миллиона особей. Причем отношение к питомцам становится все менее утилитарным, а более ответственным [1]. В связи с этим возрастает потребность в ветеринарных товарах и услугах.

Стремление упростить и ускорить свою работу, а также сэкономить время подталкивает людей к поиску новых эффективных решений взаимодействия, как при потреблении, так и при предоставлении услуг, что приводит к необходимости автоматизации деятельности ветеринарных организаций средствами сети Интернет.

Не является исключением и сфера ветеринарной фармацевтики. Ассортимент товаров и потребности людей растут так быстро, что все больше ветеринарных компаний организуют онлайн-продажу медикаментов для домашних животных и сопутствующих товаров.

Анализ информационных систем для автоматизации работы ветеринарных аптек. Согласно сервису «2GIS», на территории города Саранска действует 5 организаций, осуществляющих оборот ветеринарных и сопутствующих товаров, причем только одна из них, принадлежащая к пензенской сети, имеет собственный веб-сайт.

Для выяснения причин столь низкой информатизации в данной сфере были рассмотрены несколько существующих информационных систем ветеринарной фармацевтики. Основной функционал ресурсов описан в таблице 1.

Таблица 1

Сравнение функциональности существующих аналогичных решений

Функциональность	farmmed.ru	vetapteka1.ru	vetapteka.ru	PharmaVet.ru
Возможность регистрации клиента в системе	+	+	+	+
Возможность заказа товаров	+	+	+	+
Возможность ранжирования товаров по типу и/или назначению	+	+	+	+
Возможность задания критериев поиска	+	+	+	–
Возможность просмотра информации об организации	+	+	+	+
Возможность просмотра информации о доставке	+	+	+	+
Возможность просмотра информации о специальных предложениях и скидках	+	+	–	–
Возможность просмотра информации об оплате	+	+	+	+
Возможность просмотра информации о возврате товаров	–	+	+	–
Возможность заказа звонка по «горячей линии»	+	+	+	–
Возможность просмотра количества товаров в корзине	+	+	+	+
Возможность просмотра стоимости товаров в корзине	+	–	–	+
Возможность просмотра нормативной базы организации	+	+	–	+
Возможность задавать вопросы специалистам организации	+	+	–	–
Возможность читать статьи	+	+	–	–
Возможность просмотра новостей организации	+	+	+	–
Возможность оставления отзывов о ресурсе	+	+	–	–
Ссылки на группы в социальных сетях	+	+	–	–
Возможность просмотра контактной информации	+	+	+	+

На основе анализа приведенных выше решений можно сделать вывод об избыточности и перегруженности функционала существующих систем, которые могут привести к возникновению неудобств у пользователей при работе с системой [2]. Поэтому было принято решение реализовать программно-информационную систему «PharmaVET», функционал которой не будет перегружен. Это позволит значительно упростить работу с системой, ускорить процесс оборота товаров и расширить клиентскую базу без значительного увеличения штата сотрудников.

Проектирование программной системы (ПС). На начальном этапе проектирования система воспринимается не как набор функциональных блоков, а как единое целое. Для более подробного описания создаваемого продукта применяется метод декомпозиции. Суть метода – в замене решения исходной, более крупной задачи решением серии нескольких более простых взаимосвязанных задач [3].

Для декомпозиции первого уровня применяется методология DFD (англ. Data Flow Diagrams – диаграммы потоков данных). Это методология графического структурного анализа системы, которая описывает внешние по отношению к системе сущности (источники и приемники данных), процессы, логические функции, потоки и хранилища данных, к которым производится доступ. Такая нотация моделирует создаваемую систему с точки зрения хранения, обработки и передачи информации.

Рассмотрим основные составляющие DFD:

- Процесс – это последовательность действий, которые необходимо предпринять для успешной обработки данных.
- Внешние сущности – любые объекты, которые не входят в состав системы, но являются источниками либо приемниками информации, используемой в системе. К внешним сущностям могут относиться: человек, сторонняя система, носитель информации или хранилище данных.
- Хранилище данных – способ организации и хранения данных для процессов в системе (база данных, таблица и др.).
- Поток данных определяет информацию, передаваемую посредством некоторого соединения от источника к приемнику, а также направление ее передачи [4].

Диаграммы, моделирующие основные ситуации, возникающие при работе с системой, представлены на рис. 1 – 3.

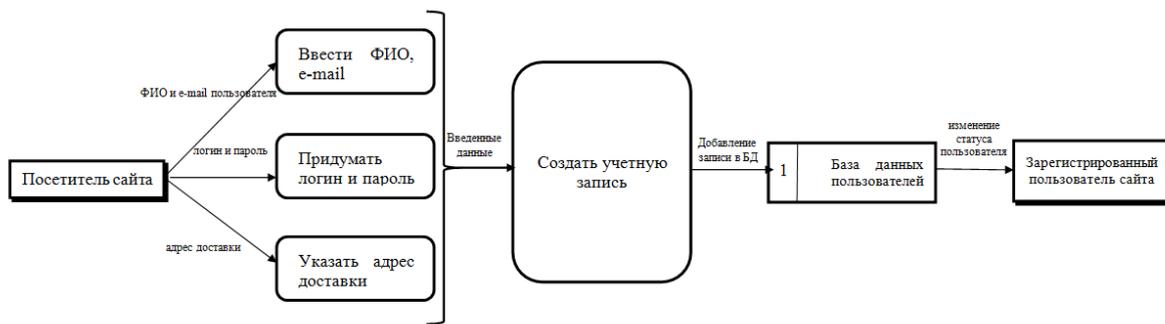


Рис. 1. Диаграмма регистрации пользователя в системе.

Процессы обработки: запросить данные пользователя, создать новую учетную запись, добавить пользователя в БД;

Потоки данных: ФИО и e-mail пользователя, логин и пароль, введенные данные, добавление записи в БД, изменение статуса пользователя;

Хранилища данных: база данных пользователей;

Внешние объекты: посетитель сайта, зарегистрированный пользователь сайта.



Рис. 2. Диаграмма авторизации пользователя в системе.

Процессы обработки: авторизовать пользователя, проверить введенные логин и пароль;

Потоки данных: логин/пароль, введенные данные, информация о логине и пароле, информация об учетной записи найдена, информация об учетной записи не найдена, подтвержденные логин и пароль, ошибка авторизации;

Хранилища данных: база данных пользователей;

Внешние объекты: посетитель сайта, зарегистрированный пользователь сайта.

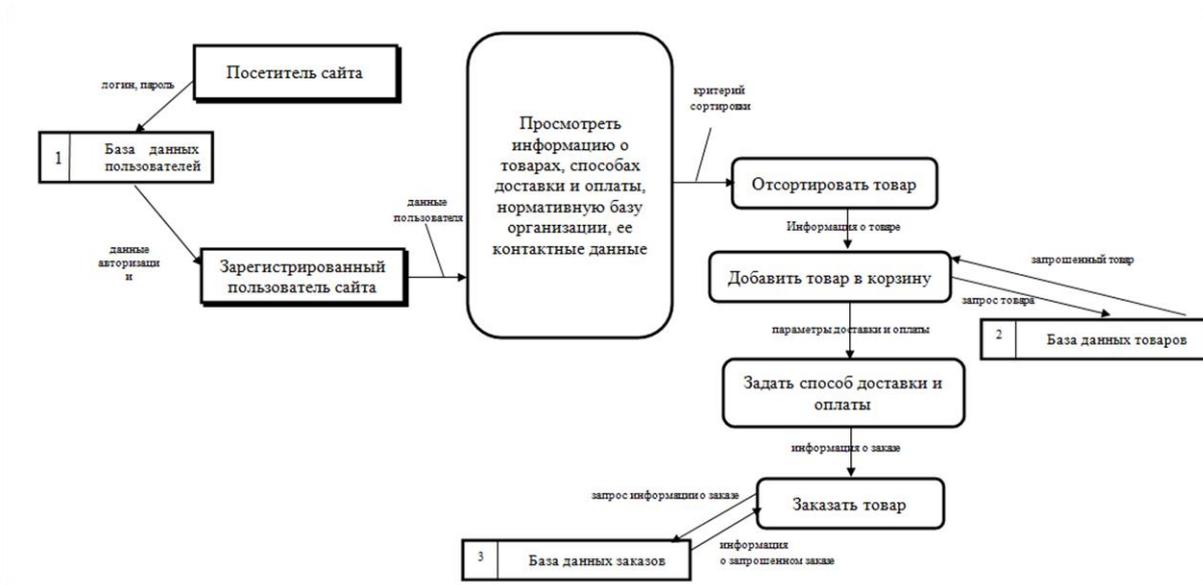


Рис. 3. Диаграмма заказа товара.

Процессы обработки: просмотреть информацию, отсортировать товар, добавить товар в корзину, задать способ доставки и оплаты, заказать товар;

Потоки данных: логин/пароль, данные авторизации, данные пользователя, данные об активности пользователя, параметры сортировки, запрос товара, запрошенный товар, информация о заказе, параметры доставки и оплаты, запрос информации о заказе, информация о запрошенном заказе;

Хранилища данных: база данных пользователей, база данных товаров, база данных заказов;

Внешние объекты: посетитель сайта, зарегистрированный пользователь сайта.

На следующем этапе проектирования необходимо определить основные группы пользователей, систему отношений между ними и функциями ресурса [5]. Для описания системы на концептуальном уровне построим диаграмму вариантов ее использования (рис. 4).

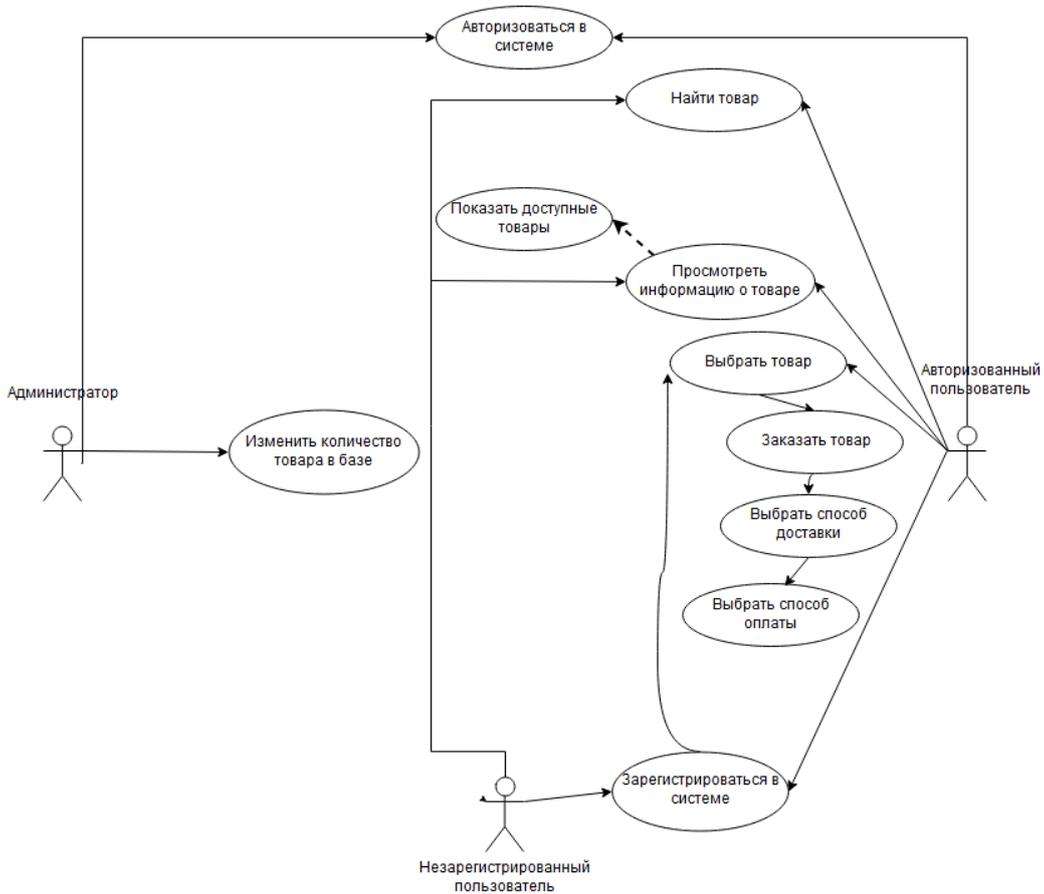


Рис. 4. Диаграмма вариантов использования.

Далее перейдем от концептуального уровня к описанию отдельных сущностей системы. Для этого определим основные типы используемых данных, типы объектов системы и различные статические связи между ними и построим схему базы данных (рис. 5). Краткое описание структуры базы данных приведено в таблице 2.

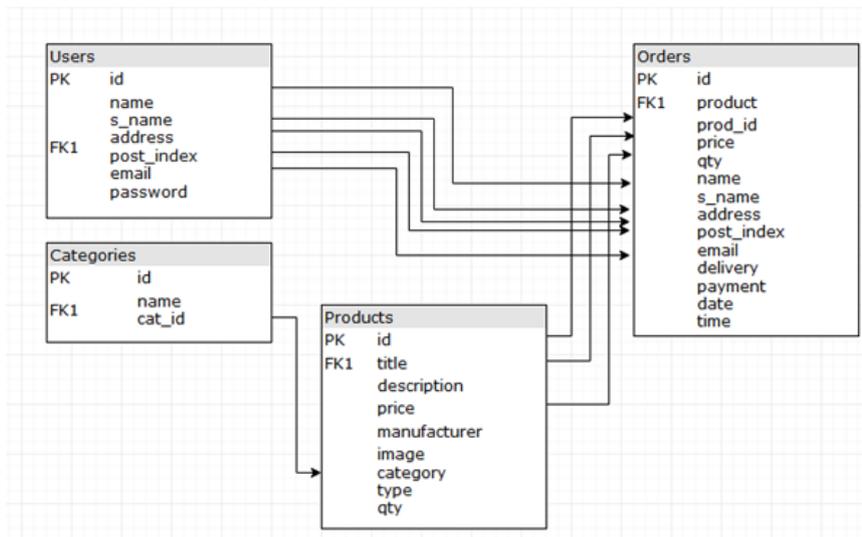


Рис. 5. Схема базы данных системы.

Обзор структуры базы данных

Название оригинальное	Название	Описание
Products	Товары	Содержит список доступных товаров с указанием названия, типа, цены и краткого описания.
Orders	Заказы	Содержит информацию о заказе: информацию о заказанном товаре, количество заказанных позиций, данные заказчика, дату и время заказа, сумму заказа, тип оплаты.
Categories	Категории товаров	Отражает имеющиеся категории товаров.
Users	Пользователи	Содержит данные пользователей: фамилию, имя, адрес с индексом, e-mail и пароль.

Технологии реализации программной системы (ПС). Для реализации описанной программной системы (ПС) были использованы следующие технологии и языки программирования:

- создание и оформление страниц сайта – HTML, CSS, JavaScript;
- работа с сессиями пользователей и взаимодействие с базами данных – язык программирования PHP;
- система управления базами данных (СУБД) MySQL;
- веб-сервер – локальный сервер Denwer.

Обзор возможностей реализованной программной системы (ПС). Рассмотрим основные возможности системы «PharmaVET».

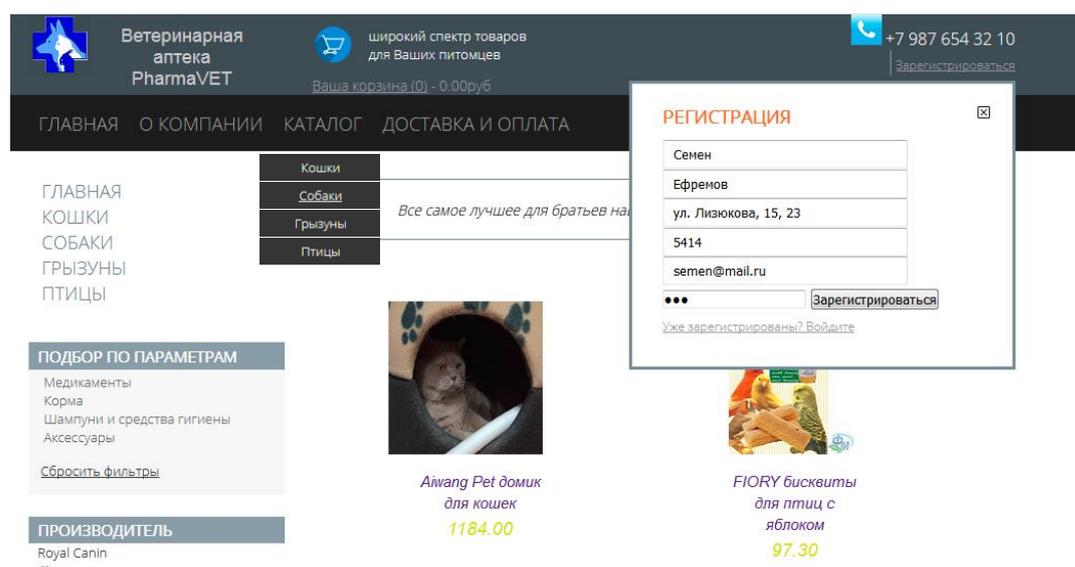


Рис. 6. Главная страница ресурса, выпадающая панель навигации и форма регистрации.

С любой страницы ресурса доступны переходы между страницами посредством горизонтального и вертикального (выпадающего) меню, строка поиска, выпадающее окно регистрации, форма авторизации, ссылки на главную страницу и корзину, задание критериев фильтрации товаров в каталоге. Внешний вид главной страницы с активными выпадающими элементами представлен на рисунке 6.

Для авторизации в системе пользователь вводит оригинальные логин и пароль (рис. 7). В случае некорректного ввода данных либо попытке авторизации незарегистрированного пользователя выводятся сообщения об ошибках.

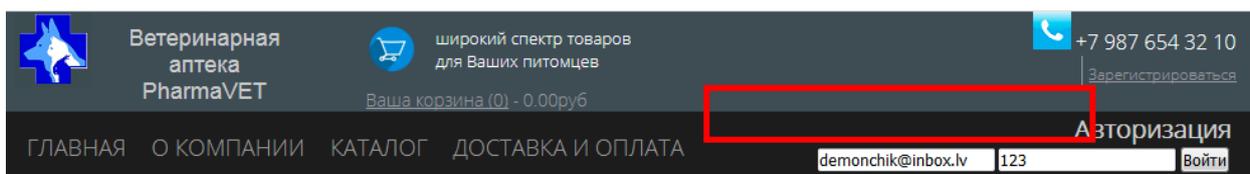


Рис. 7. Форма авторизации.

После успешного входа пользователь попадает на страницу своего профиля, где видит историю заказов в ветеринарной аптеке (см. рис. 8).

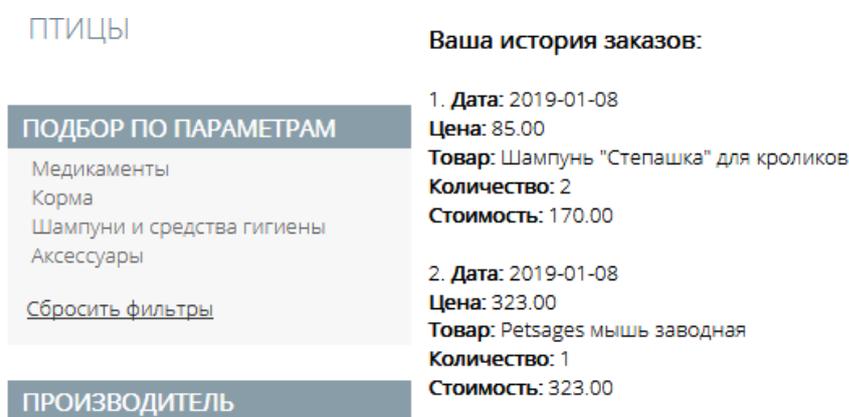


Рис.8. Страница профиля пользователя – история заказов.

Подробную информацию о товаре можно получить, щелкнув на его название в каталоге. Внешний вид страницы описания товара с изменением индикатора состояния корзины при добавлении товара представлен на рисунке 9.

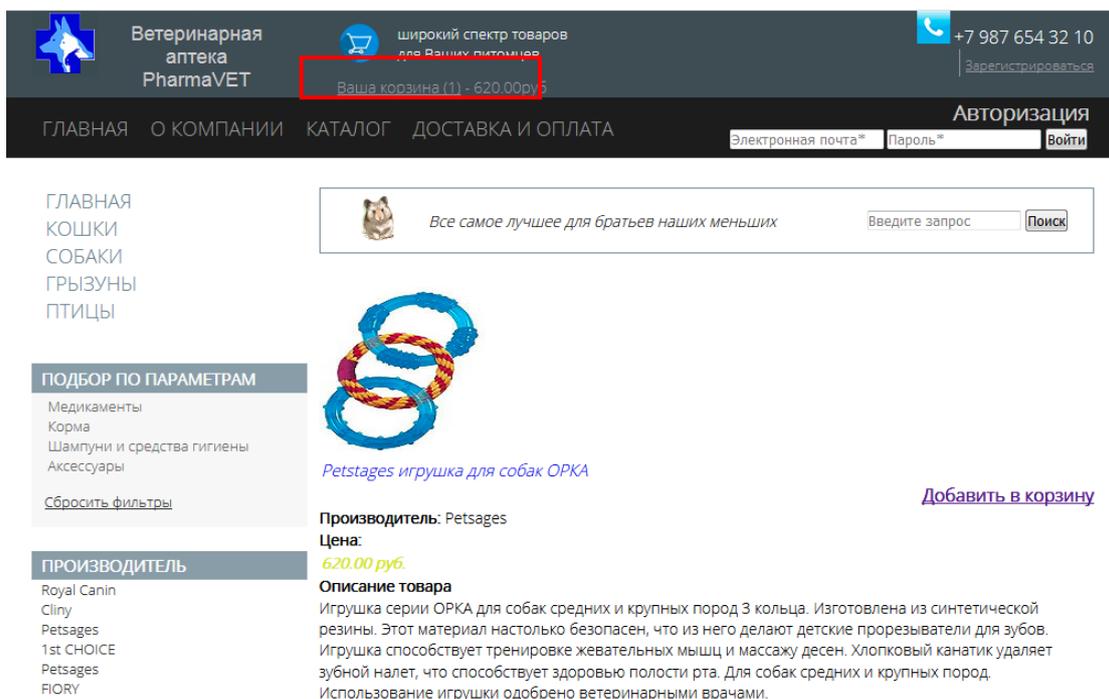


Рис. 9. Страница описания продукта и изменение индикатора состояния корзины после добавления товара.

На странице корзины реализована возможность изменения количества единиц приобретаемого товара. Для оформления заказа необходимо ввести данные авторизации (логин и пароль), а также выбрать способы доставки и оплаты. Внешний вид страницы оформления заказов изображен на рисунке 10.

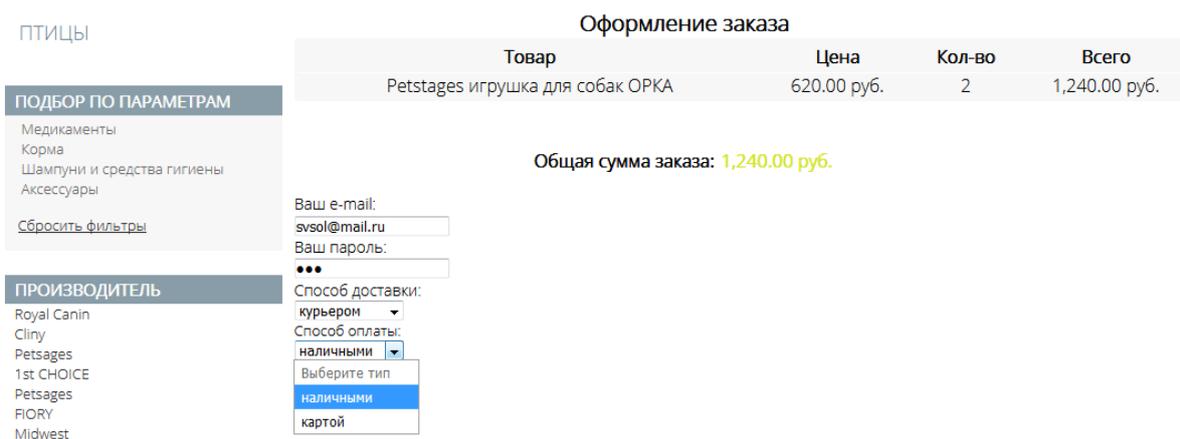


Рис. 10. Страница корзины товаров и параметры оформления заказа.

Одной из основных функций системы является функция поиска товаров в каталоге. Страница результатов поиска представлена на рисунке 11.



Рис. 11. Вывод результатов поиска.

Заключение. Сравнительное описание функциональности существующих систем и программной системы (ПС) «PharmaVET» было рассмотрено выше. Разработанная система не уступает в основной функциональности рассмотренным аналогам, но при этом не содержит функций, которые могут показаться пользователю избыточными и, как следствие, затруднить процесс его взаимодействия с системой. Кроме того, лаконичный и дружелюбный интерфейс сочетается с удобством использования и высокой производительностью. Описанные свойства позволят сделать работу ветеринарной аптеки более удобной и эффективной при минимальных затратах. Для повышения конкурентоспособности ПС «PharmaVET» на рынке необходимо интегрировать систему с профессиональным торговым оборудованием, что позволит повысить надежность системы и избежать ручного переноса данных между различными системами.

ЛИТЕРАТУРА

1. Число домашних животных в РФ выросло на 14% за три года [Электронный ресурс]. – Режим доступа: <https://www.interfax.ru/russia/631927> (дата обращения: 27.06.2019).
2. Симаков И.В., Анчихрова Н.А., Курицин А.И., Вольников М.И. Обзор программного обеспечения для автоматизации процесса оказания ветеринарных услуг // Вклад молодых ученых в инновационное развитие АПК России: сборник статей Международной научно-практической конференции молодых ученых, посвященной 65-летию ФГБОУ ВО Пензенская ГСХА. – Пенза, 2016. – С. 3–4.
3. Хорошев А. Н. Введение в управление проектированием механических систем: учебное пособие. – Белгород, 1999. – 372 с.
4. Что такое DFD (диаграммы потоков данных) [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/trinion/blog/340064/> (дата обращения: 27.06.2019).
5. Розенберг Д., Скотт К. Применение объектного моделирования с использованием UML и анализ прецедентов / пер. с англ. – М.: ДМК Пресс, 2002. – 484 с.

СТЕНИН И. В., ШАМАНАЕВ П. А., ГОРШУНОВА Т. А.

АЛГОРИТМ ИДЕНТИФИКАЦИИ ПАРАМЕТРОВ ДИНАМИЧЕСКОЙ СИСТЕМЫ
ВТОРОГО ПОРЯДКА С МАЛЫМИ ВОЗМУЩЕНИЯМИ
ПО ЭКСПЕРИМЕНТАЛЬНЫМ ДАННЫМ

Аннотация. Задача идентификации параметров линейных динамических систем второго порядка с малыми возмущениями по экспериментальным данным решается путем сведения ее к задаче минимизации квадратичного функционала с ограничениями в виде нелинейных алгебраических уравнений. Проведены вычисления на трех экспериментальных наборах данных, соответствующих трем типам особой точки линейной системы: седло, узел, центр.

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

STENIN I. V., SHAMANAEV P. A., GORSHUNOVA T. A.

ALGORITHM OF PARAMETER IDENTIFICATION OF SECOND-ORDER
DYNAMICAL SYSTEM WITH SMALL PERTURBATIONS
ACCORDING TO EXPERIMENTAL DATA

Abstract. The problem of identifying the parameters of the second-order linear dynamical systems with small perturbations according to experimental data is solved by reducing it to the problem of minimizing a quadratic functional with constraints in the form of nonlinear algebraic equations. The calculations were carried out on three experimental data sets corresponding to the three types of singular points of a linear system: the saddle, the node, and the center.

Keywords: identification of parameters, ordinary differential equations, minimization.

Рассмотрим линейную систему обыкновенных дифференциальных уравнений с малыми возмущениями вида

$$\begin{cases} \frac{dx_1}{dt} = \theta_1 x_1 + \theta_2 x_2 + \varepsilon_1 x_1 \\ \frac{dx_2}{dt} = \theta_3 x_1 + \theta_4 x_2 + \varepsilon_2 x_2 \end{cases} \quad (1)$$

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, ε_j ($j = 1, 2$) – достаточно малые вещественные параметры.

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

Пусть при некоторых фиксированных значениях $\theta_k, k = 1, \dots, 4$, решение системы (1) удовлетворяет задаче Коши с начальными условиями:

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

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

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

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, то есть $\varepsilon_{ij} \in N(0, 1)$, $i = \overline{1, N}, j = 1, 2$.

Ставится задача идентификации параметров системы вида (1), заключающаяся в нахождении таких оценок $\hat{\theta}_k$ параметров $\theta_k, k = 1, \dots, 4$, при которых решение задачи (1) приближается экспериментальными данными $\{\tilde{x}^{(i)}, i = \overline{1, N}\}$ в смысле метода наименьших квадратов [1].

Заменим уравнение (1) симметричной разностной схемой [2] на сетке (3)

$$\begin{aligned} \frac{x_1^{(i+1)} - x_1^{(i)}}{\tau} &= \frac{1}{2}(f_{i,1} + f_{i+1,1}), \\ \frac{x_2^{(i+1)} - x_2^{(i)}}{\tau} &= \frac{1}{2}(f_{i,2} + f_{i+1,2}), \\ i &= \overline{1, N-1}, \end{aligned} \quad (5)$$

где

$$\begin{aligned} f_{i,1} &= f_1(x^{(i)}, \theta) \equiv \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \varepsilon_1 x_1^{(i)}, \\ f_{i,2} &= f_2(x^{(i)}, \theta) \equiv \theta_3 x_1^{(i)} + \theta_4 x_2^{(i)} + \varepsilon_2 x_2^{(i)}. \end{aligned}$$

Вводя обозначения [3]

$$\begin{aligned} z &= \text{column}(x^{(N)}, \theta), \quad x^{(N)} = \text{column}(x^{(1)}, \dots, x^{(N)}), \\ \tilde{z} &= \text{column}(\tilde{x}^{(N)}, o_4), \quad \tilde{x}^{(N)} = \text{column}(\tilde{x}^{(1)}, \dots, \tilde{x}^{(N)}), \\ o_4 & \text{ – нулевой вектор размерности } 4, \end{aligned}$$

получим

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

где

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

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

Тогда, согласно [3–5], задача идентификации параметров может быть сформулирована как задача минимизации квадратичного функционала с ограничениями в виде нелинейных алгебраических уравнений

$$\begin{cases} \min_z m(z), & m(z) = \frac{1}{2} (H_2 z, z) - (H_2 \tilde{z}, z) \\ T(z) = h \\ g(z) = 0 \end{cases}. \quad (6)$$

Здесь введены следующие обозначения: $T = [T_1 T_2, \dots, T_N, T_\theta]$ – $(2 \times (2N + 4))$ – матрица, T_i , $i = \overline{2, N}$ – нулевые (2×2) – матрицы, T_θ – (2×4) – матрица;

$$\begin{aligned} g(z) &= \text{column}(g_1(z), \dots, g_N(z)), \quad g_i(z) = \text{column}(g_{i1}(z), g_{i2}(z)), \quad i = \overline{1, N}, \\ g_{i1}(z) &= g_{i1}(x^{(N)}, \theta) \equiv \\ &\equiv \left(1 + \frac{\tau}{2}(\theta_1 + \varepsilon_1)\right) x_1^{(i)} + \frac{\tau}{2} \theta_2 x_2^{(i)} - \left(1 - \frac{\tau}{2}(\theta_1 + \varepsilon_1)\right) x_1^{(i+1)} + \frac{\tau}{2} \theta_2 x_2^{(i+1)}, \end{aligned} \quad (7)$$

$$\begin{aligned} g_{i2}(z) &= g_{i2}(x^{(N)}, \theta) \equiv \\ &\equiv \frac{\tau}{2} \theta_3 x_1^{(i)} + \left(1 + \frac{\tau}{2}(\theta_4 + \varepsilon_2)\right) x_2^{(i)} + \frac{\tau}{2} \theta_3 x_1^{(i+1)} - \left(1 - \frac{\tau}{2}(\theta_4 + \varepsilon_2)\right) x_2^{(i+1)}. \end{aligned} \quad (8)$$

Заметим, что разностная схема (5) с учетом обозначений (7)-(8) может быть записана в виде

$$g(z) = 0.$$

Для решения задачи (6) воспользуемся алгоритмом [3], основанным на аппроксимации исходной задачи последовательностью квадратичных задач минимизации с линейными ограничениями. На каждом шаге разреженная система линейных алгебраических уравнений большой размерности решалась с использованием метода сопряженных градиентов [4].

Вычисления проводились на трех наборах экспериментальных данных $\tilde{x}^{(i)}$, $i = \overline{1, N}$, отличающихся от приближенных решений с начальными данными $x_1(0) = 1$, $x_2(0) = 1$ системы (1) на случайные величины $\varepsilon_{ij} \in N(0, 1)$ ($j = 1, 2$), имеющие стандартное нормальное распределение. Каждое приближенное решение системы (1) получено при фиксированном наборе параметров θ_k , $k = 1, \dots, 4$, соответствующих трем типам особой точки системы (1): седло, узел, центр. Точность вычислений δ полагалась равной 0.001.

В результате вычислений для трех различных наборов экспериментальных данных получены оценки $\hat{\theta}_k$ для параметров θ_k , $k = 1, \dots, 4$, а также компоненты $x_1(t, \hat{\theta})$, $x_2(t, \hat{\theta})$ приближенных решений системы (1).

Приведем графики компонент $x_1(t, \hat{\theta})$, $x_2(t, \hat{\theta})$ приближенных решений системы (1) и соответствующие им экспериментальные данные.

Случай 1. Нулевое положение равновесия системы (1) – седло.

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

$$\hat{\theta}_1 = 2.02298, \hat{\theta}_2 = 0.0001, \hat{\theta}_3 = 0.00002, \hat{\theta}_4 = -2.90002..$$

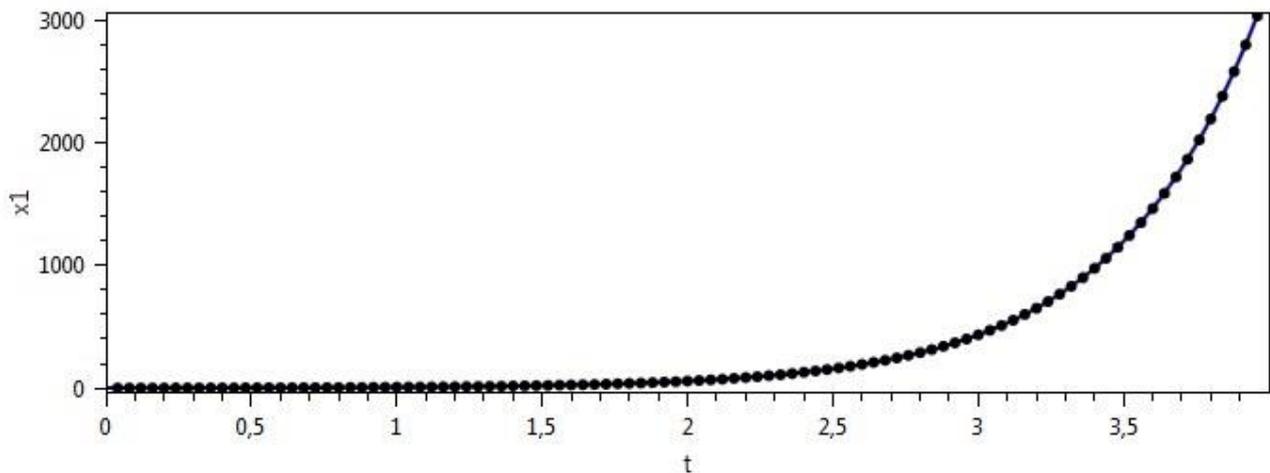


Рис. 1. Экспериментальные данные и график компоненты $x_1(t, \hat{\theta})$ приближенного решения системы (1).

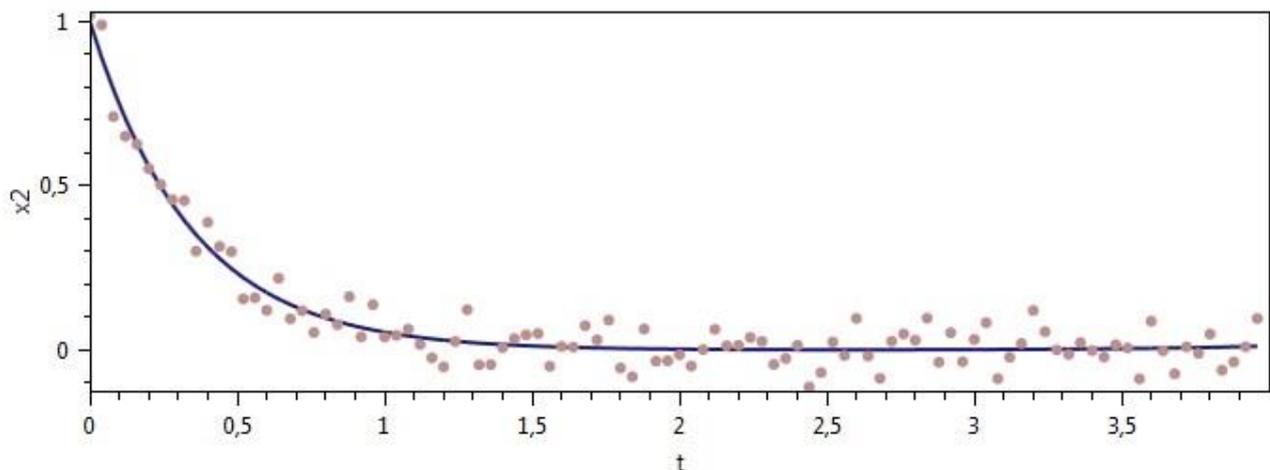


Рис. 2. Экспериментальные данные и график компоненты $x_2(t, \hat{\theta})$ приближенного решения системы (1).

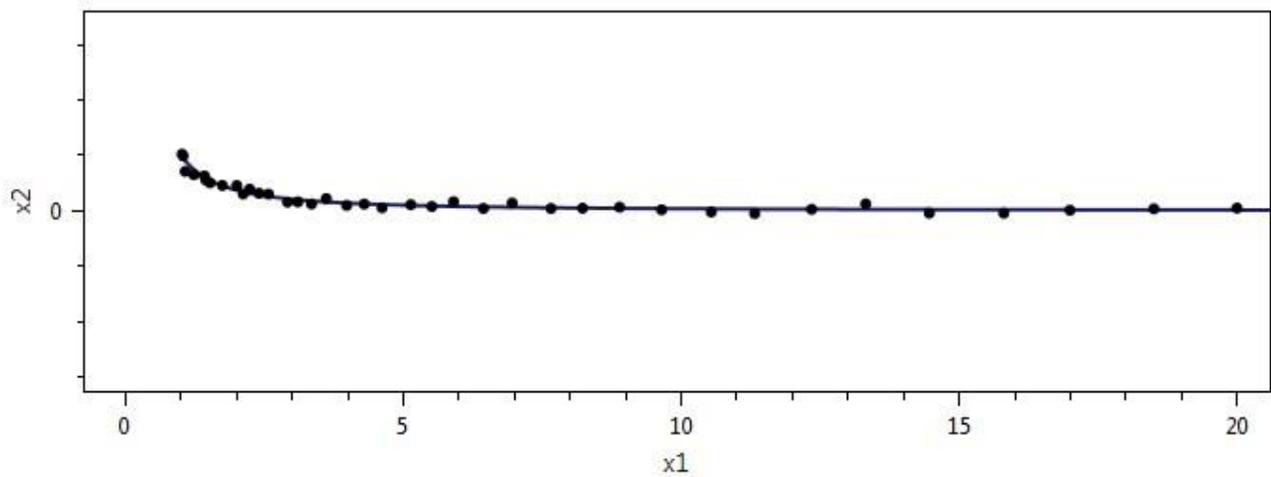


Рис. 3. Экспериментальные данные и приближение фазовой траектории системы (1) с начальными данными $x_1(0) = 1$, $x_2(0) = 1$.

Случай 2. Нулевое положение равновесия системы (1) – узел.

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

$$\hat{\theta}_1 = -3.14132, \hat{\theta}_2 = 0.01895, \hat{\theta}_3 = 1.03559, \hat{\theta}_4 = 0.02129.$$

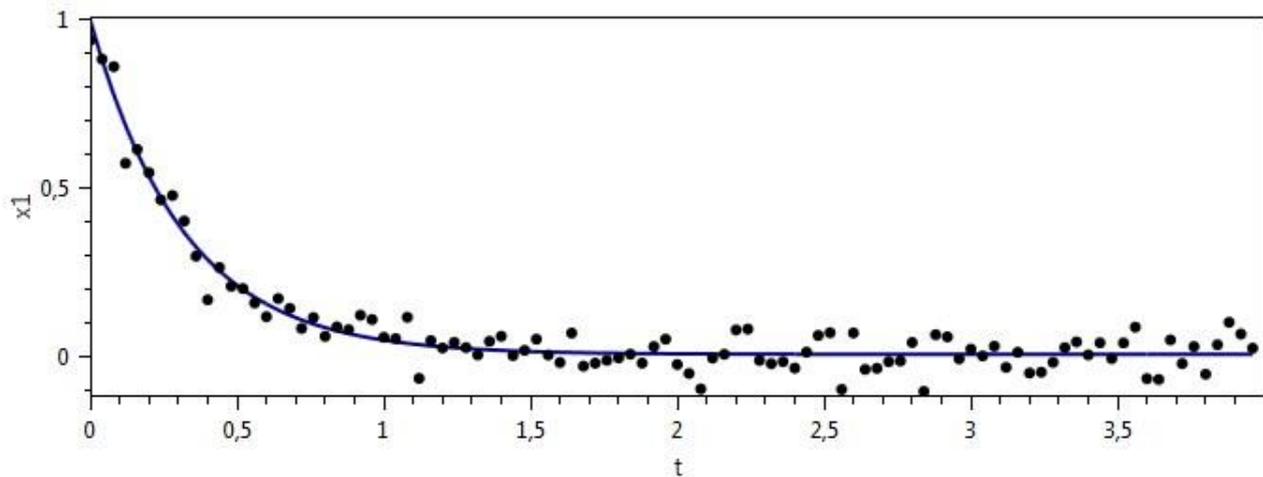


Рис. 4. Экспериментальные данные и график компоненты $x_1(t, \hat{\theta})$ приближенного решения системы (1).

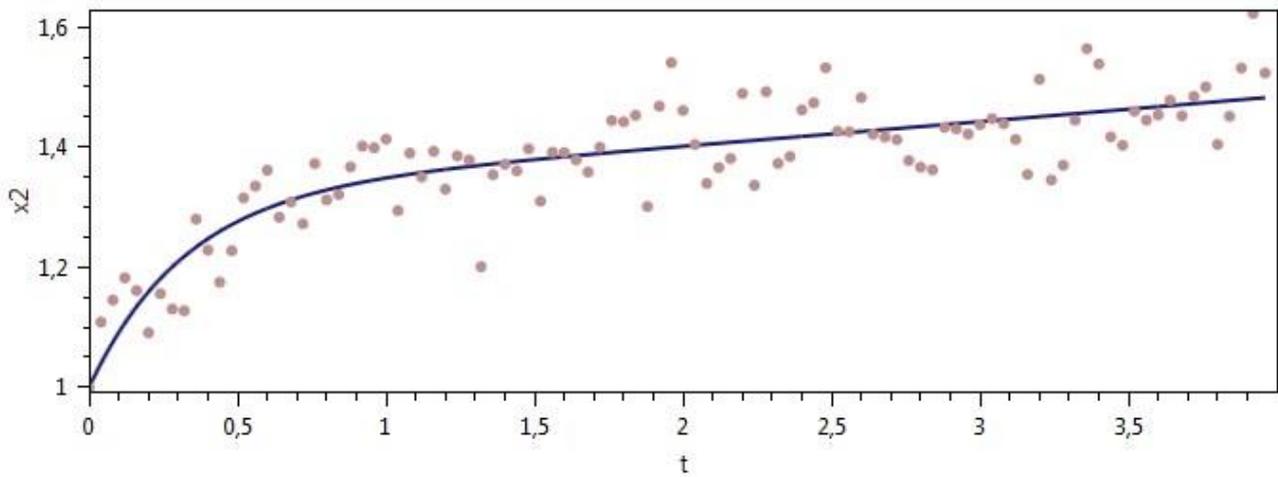


Рис. 5. Экспериментальные данные и график компоненты $x_2(t, \hat{\theta})$ приближенного решения системы (1).

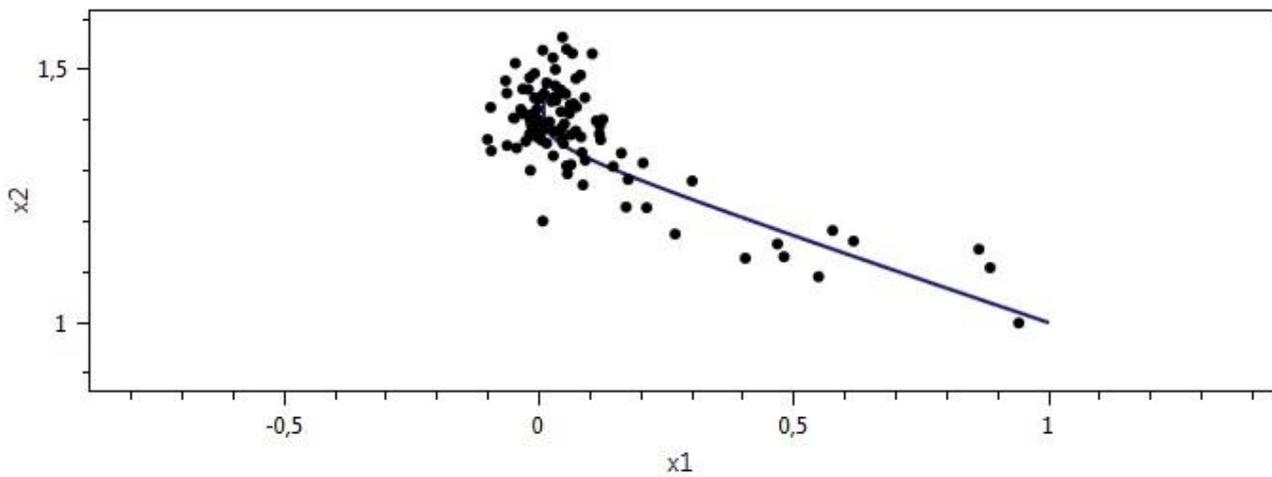


Рис. 6. Экспериментальные данные и приближение фазовой траектория системы (1) с начальными данными $x_1(0) = 1, x_2(0) = 1$.

Случай 3. Нулевое положение равновесия системы (1) – центр.

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

$$\hat{\theta}_1 = 0.07274, \hat{\theta}_2 = 4.01077, \hat{\theta}_3 = -3.98702, \hat{\theta}_4 = -0.02925.$$

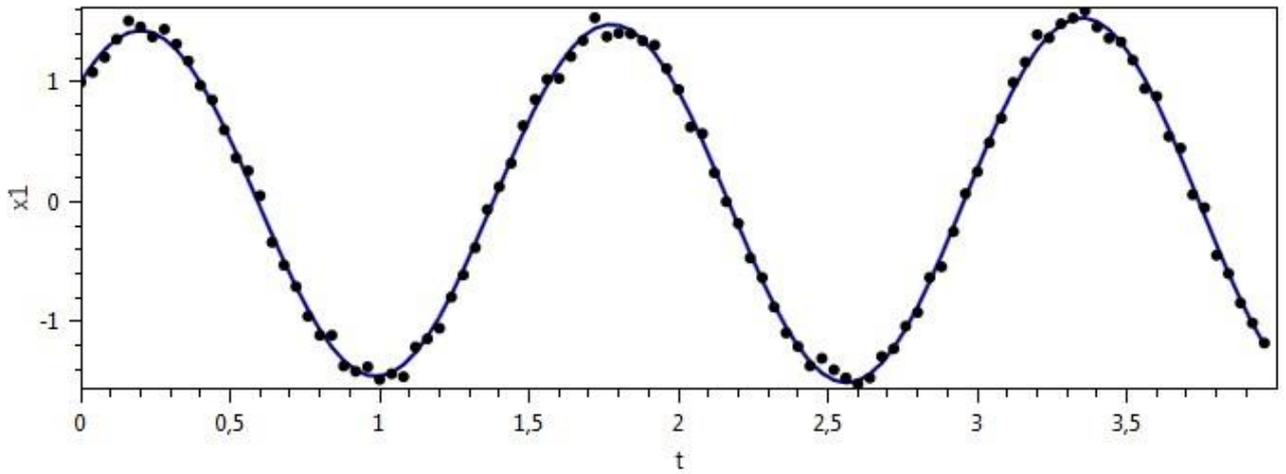


Рис. 7. Экспериментальные данные и график компоненты $x_1(t, \hat{\theta})$ приближенного решения системы (1).

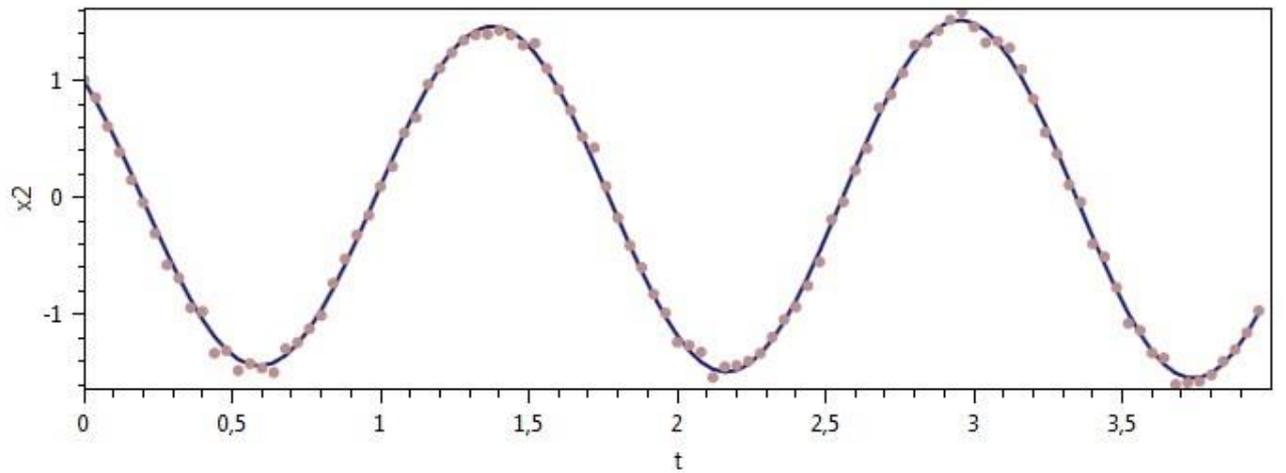


Рис. 8. Экспериментальные данные и график компоненты $x_2(t, \hat{\theta})$ приближенного решения системы (1).

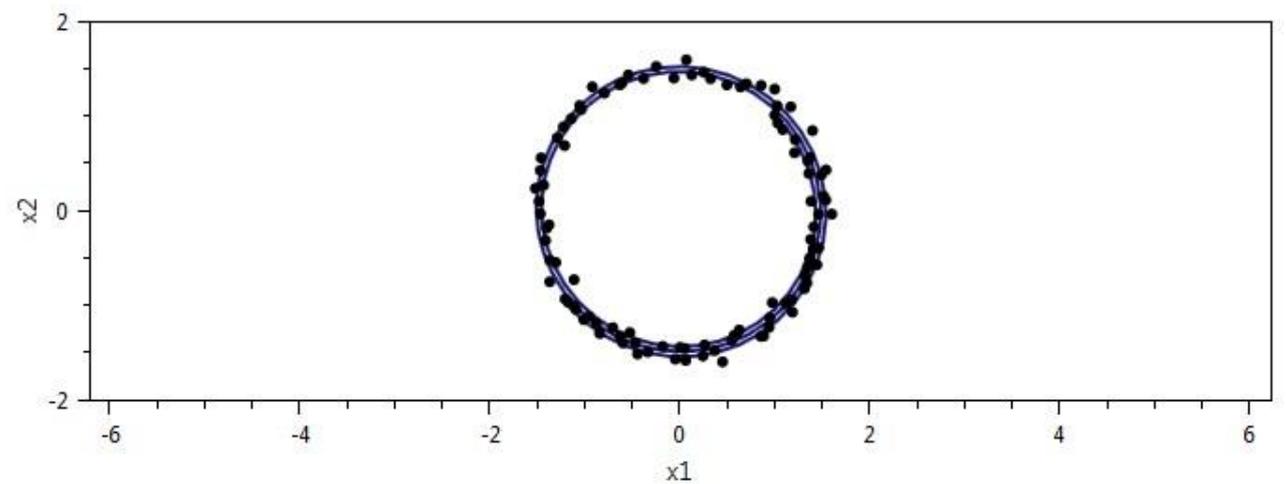


Рис. 9. Экспериментальные данные и приближение фазовой траектории системы (1).

ЛИТЕРАТУРА

1. Zhengfeng Li, Michael R. Osborne, Tania Prvan. Parameter estimation of ordinary differential equations // IMA Journal of Numerical Analysis. – 2005. – No. 25. – P. 264–285.
2. Самарский А. А., Гулин А. В. Численные методы. – М.: Наука, 1989. – 432 с.
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. Стенин И. В., Шаманаев П. А. Алгоритм решения разреженной системы линейных алгебраических уравнений большой размерности с использованием метода сопряженных градиентов // Огарев-online. – 2017. – № 13 [Электронный ресурс]. – Режим доступа: <http://journal.mrsu.ru/arts/algorithm-resheniya-razrezhennoj-sistemy-linejnyx-algebraicheskix-uravnenij-bolshoj-razmernosti-s-ispolzovaniem-metoda-sopryazhennyx-gradientov> (дата обращения 27.06.2019).
5. Стенин И. В., Шаманаев П. А. Идентификация параметров динамической системы второго порядка по экспериментальным данным [Электронный ресурс] // Огарев-online. – 2018. – № 14. – Режим доступа: <http://journal.mrsu.ru/arts/identifikaciya-parametrov-dinamicheskoy-sistemy-vtorogo-poryadka-po-eksperimentalnym-dannym> (дата обращения 27.06.2019).