

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

---

# ШЕСТЬДЕСЯТ ВОСЬМАЯ МЕЖДУНАРОДНАЯ СТУДЕНЧЕСКАЯ НАУЧНАЯ КОНФЕРЕНЦИЯ ГУАП

Часть I

ТЕХНИЧЕСКИЕ НАУКИ

Сборник докладов

13–17 апреля 2015 г.



Санкт-Петербург  
2015

УДК 001  
ББК 72  
Ш51

Ш51           Шестьдесят восьмая международная студенческая научная конференция ГУАП:  
Сб. докл.: в 2 ч. Ч. I. Технические науки / СПб.: ГУАП., 2015. 335 с.: ил.  
ISBN 978-5-8088-1032-7 (Ч. I)  
ISBN 978-5-8088-1031-0

Представлены доклады Шестьдесят восьмой международной студенческой научной конференции ГУАП, прошедшей в апреле 2015 г.

Научные работы студентов ГУАП посвящены актуальным проблемам авиационного и аэрокосмического приборостроения, проблемам развития новых информационных технологий, радиотехники, электроники и связи, современным проблемам экономики, философии и права.

Под общей редакцией  
доктора юридических наук, профессора  
*В. М. Боева*

Редакционная коллегия:

*А. Р. Бестугин*, доктор технических наук, доцент,  
*А. С. Будагов*, доктор экономических наук, доцент,  
*Е. А. Крук*, доктор технических наук, профессор,  
*К. В. Лосев*, доктор экономических наук, доцент,  
*И. А. Павлов*, кандидат военных наук, доцент,  
*Е. Г. Семенова*, доктор технических наук, профессор,  
*М. Б. Сергеев*, доктор технических наук, профессор,  
*В. А. Фетисов*, доктор технических наук, профессор,  
*Н. А. Чернова*, кандидат экономических наук,  
*Л. И. Чубраева*, доктор технических наук, профессор

Ответственный секретарь редакционной коллегии *В. Н. Соколова*

ISBN 978-5-8088-1032-7 (Ч. I)  
ISBN 978-5-8088-1031-0

© Санкт-Петербургский государственный  
университет аэрокосмического  
приборостроения, 2015

# ТЕХНИЧЕСКИЕ НАУКИ

## Аэрокосмические приборы и системы

УДК 004.942

**А. Ю. Асенчик**

студент кафедры системного анализа и логистики

**Н. Н. Майоров**

канд. техн. наук, доц. – научный руководитель

### ИССЛЕДОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ТРАНСПОРТНЫХ СИСТЕМ

Транспортная система в общем случае – это совокупность работников, транспортных средств и оборудования, элементов транспортной инфраструктуры и инфраструктуры субъектов перевозки, включая систему управления, направленная на эффективное перемещение грузов и пассажиров [1].

При прогнозе эффективности перевозок, проектировании движения транспортных средств решаются задачи моделирования. Для решения транспортных задач может быть использовано физическое, математическое и имитационное моделирование.

При проектировании и анализе работы транспортных систем особенно эффективно именно имитационное моделирование. Приведем несколько примеров решаемых задач с помощью имитационного моделирования на различных видах транспорта:

- планирование и составление расписаний работы транспортной системы;
- управление парком транспортных средств и перевозками: стратегическое и оперативное управление парком транспортных средств, расчет объема заявок, который может обслужить фирма по перевозкам при заданном уровне обслуживания, оптимизация и планирование перевозок, минимизация затрат на управление перевозками и содержание парка, оценка рисков принимаемых решений;
- управление работой складов, терминалов, портов;
- определение суточной пропускной способности аэропорта, ж/д вокзала, метрополитена, морского порта и пр., времени их работы в режиме перегрузки;
- исследование влияния надежности работы оборудования на пропускную способность системы, анализ эффекта оргмероприятий: (к примеру, расстановки ограждений, турникетов, введения (отмены) льготного проезда) и др. [2].

Особенными характеристиками программного обеспечения имитационного моделирования являются:

- способность моделирования и отслеживание в общем модельном времени различных потоков (материальных, информационных, денежных и пр.);
- возможность уточнения параметров и ведения экстремального эксперимента.

В настоящее время существуют десятки пакетов для реализации имитационного моделирования. В данной работе мной выбраны следующие две программные среды: AnyLogic и LiteSMO.

LiteSMO – это простая и удобная в использовании среда для моделирования систем массового обслуживания, в которой модель может быть представлена в виде графической схемы, состоящей из генераторов заявок, каналов и накопителей.

AnyLogic – программное обеспечение для имитационного моделирования, обладающее современным графическим интерфейсом и позволяющее использовать язык Java для разработки моделей; единственный инструмент имитационного моделирования, который поддерживает все подходы к созданию имитационных моделей: процессно-ориентированный (дискретно-событийный), системно динамический и агентный, а также любую их комбинацию.

Приведем сравнительную характеристику основных возможностей и преимуществ двух этих систем (табл. 1).

Таблица 1

	LiteSMO	Anylogic
Основные возможности	<ul style="list-style-type: none"> <li>– определение максимальных и среднестатистических значений основных показателей эффективности системы;</li> <li>– создание стохастических систем (построение гистограмм распределений различных характеристик модели, проверка гипотезы о предполагаемом законе распределения, поддержка 4 видов закона распределения: детерминированного, равномерного, показательного и нормального);</li> <li>– создание графиков, отражающих зависимость показателей эффективности системы от параметров модели</li> </ul>	<ul style="list-style-type: none"> <li>– поддержка 3-х подходов к созданию имитационных моделей;</li> <li>возможность расширения модели за счет программирования на Java, создания пользовательских библиотек;</li> <li>– возможность работы с базами данных;</li> <li>– создание сложных стохастических моделей;</li> <li>– возможность создания интерактивной анимации для улучшения наглядности моделей;</li> <li>– возможность использования модели AnyLogic как отдельного Java приложения полностью независимого от среды разработки</li> </ul>
Преимущества над сравниваемой средой	<ul style="list-style-type: none"> <li>– простота использования;</li> <li>– свободное распространение программы</li> </ul>	<ul style="list-style-type: none"> <li>– возможность создания визуализации модели (как двумерной, так и трехмерной);</li> <li>– возможность программирования собственных библиотек</li> </ul>

Приведем оконные формы этих сред моделирования на примере задачи работы морского порта на рис. 1 и рис. 2.

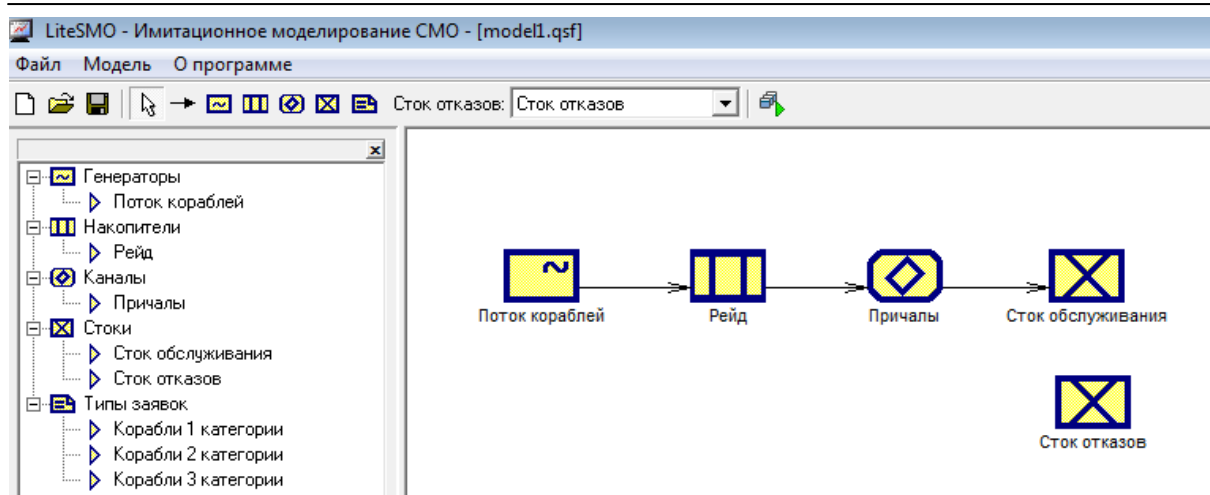


Рис. 1. Оконная форма среды моделирования LiteSMO

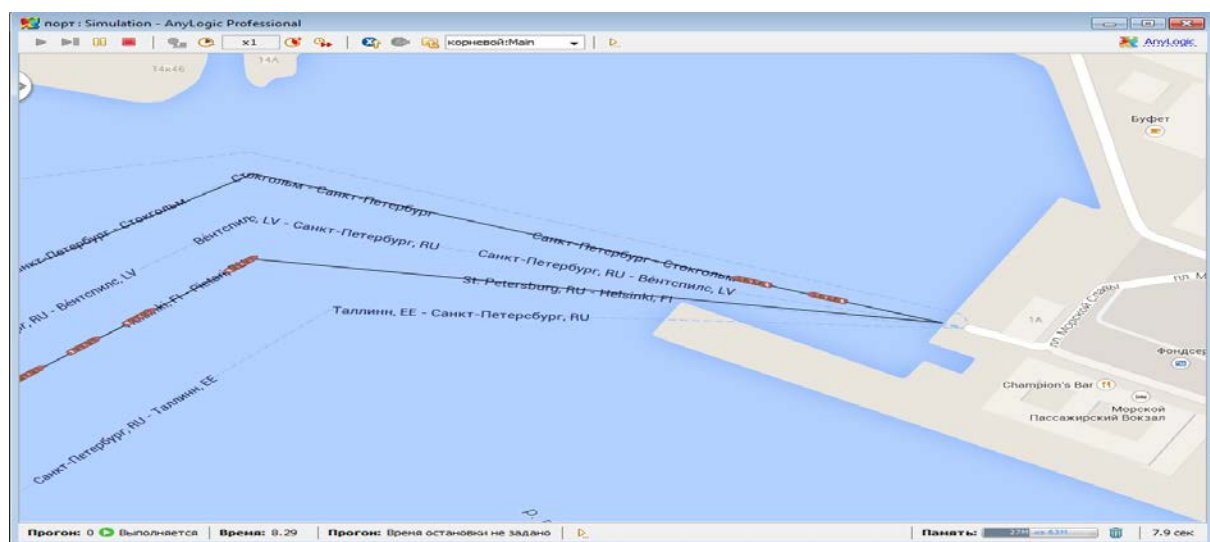


Рис. 2. Оконная форма работы модели в среде AnyLogic

Таким образом, исходя из проведенного мною исследования, можно сделать следующие выводы:

Программная среда LiteSMO является простым средством для исследования необходимой модели транспортной системы на начальном этапе изучения систем массового обслуживания. LiteSMO позволяет легко определить основные статистические показатели работы системы, а также наглядно предоставить результаты этих расчетов в виде графиков и гистограмм. Данная среда хорошо подходит для задач, в которых не нужна визуализация модели, а требуются лишь основные статистические показатели.

Программная среда AnyLogic, в свою очередь, является универсальным средством моделирования транспортных систем. AnyLogic требует большего изучения и вовлечения в среду, но взамен предоставляет пользователю большие возможности в визуализации модели, математических расчетов, а также создании собственных библиотек на языке среды Java.

### Библиографический список

1. Моделирование транспортных систем. Майоров Н. Н., Фетисов В. А. // СПб.: ГУАП, 2011. 165 с.
2. Рыжиков Ю. И. Имитационное моделирование: теория и технологии. М.: Альтекс, 2004.

УДК 681.51

**Л. И. Барбашова**

студент кафедры аэрокосмических измерительно-вычислительных комплексов

**Ю. П. Иванов**

канд. техн. наук, доц. – научный руководитель

## ИССЛЕДОВАНИЕ СИНГУЛЯРНО-СПЕКТРАЛЬНЫХ АЛГОРИТМОВ ФИЛЬТРАЦИИ СИГНАЛОВ

Методы оптимального оценивания сигналов в условиях полной априорной определенности по мере поступления информации широко применяются в различных областях техники. Оптимальные алгоритмы оценивания являются результатом решения соответствующих модельных задач при наличии полной априорной определенности относительно моделей сигналов и помех. На практике столь полное описание моделей полезных сигналов и погрешностей встречается относительно редко. Чаще всего задачи обработки информации и принятия решения сопровождаются большей или меньшей априорной неопределенностью, которая ограничивает полноту статистического описания. Распространенность подобных ситуаций и их большая практическая значимость делают особо важными разработку и использование методов обработки информации и принятия решений в условиях априорной неопределенности [1].

В данной работе исследуются характеристики адаптивных сингулярно-спектрального и оптимального сингулярно-спектрального алгоритмов фильтрации сигналов (SSA) в зависимости от изменения параметров алгоритмов фильтрации и помех и проводится сравнительный анализ данных алгоритмов фильтрации с фильтрацией Калмана по показателю качества – дисперсии ошибки оценки. Модель измерения линейная. Погрешность аддитивная и некоррелирована с полезным сигналом. Априорная информация о входных сигналах системы отсутствует.

Алгоритм сингулярно-спектрального метода фильтрации состоит из четырех шагов (рис. 1) [2].

1. Вложение.

Процедура вложения переводит исходный временной ряд длины  $N$  в последовательность многомерных векторов.

Пусть  $R$  – длина окна,  $1 < R < N$ . Процедура вложения образует  $C = N - R + 1$  векторов вложения, имеющих размерность  $R$ , которые можно представить в виде траекторной матрицы  $X$ :

$$X = \begin{matrix} Y_1 & Y_2 & \cdot & \cdot & Y_C \\ Y_2 & Y_3 & \cdot & \cdot & Y_{C+1} \\ Y_3 & Y_4 & \cdot & \cdot & Y_{C+2} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ Y_R & Y_{R+1} & \cdot & \cdot & Y_N \end{matrix}$$

2. Сингулярное разложение.

На этом этапе находятся собственные числа  $(\lambda_1, \dots, \lambda_i)$ , левые  $(U_1, \dots, U_i)$  и правые  $(V_1, \dots, V_i)$  сингулярные вектора матрицы  $X$ . Набор  $(\lambda_i, U_i, V_i)$  называется  $i$ -ой собственной тройкой сингулярного разложения. Таким образом, происходит сингулярное разложение траекторной матрицы.

3. Группировка.

На данном шаге происходит группировка собственных троек и формирование матрицы, состоящей из собственных троек сингулярного разложения.

4. Диагональное усреднение

Каждая матрица сгруппированного разложения переводится в новый ряд длины  $N$  по формуле диагонального усреднения.

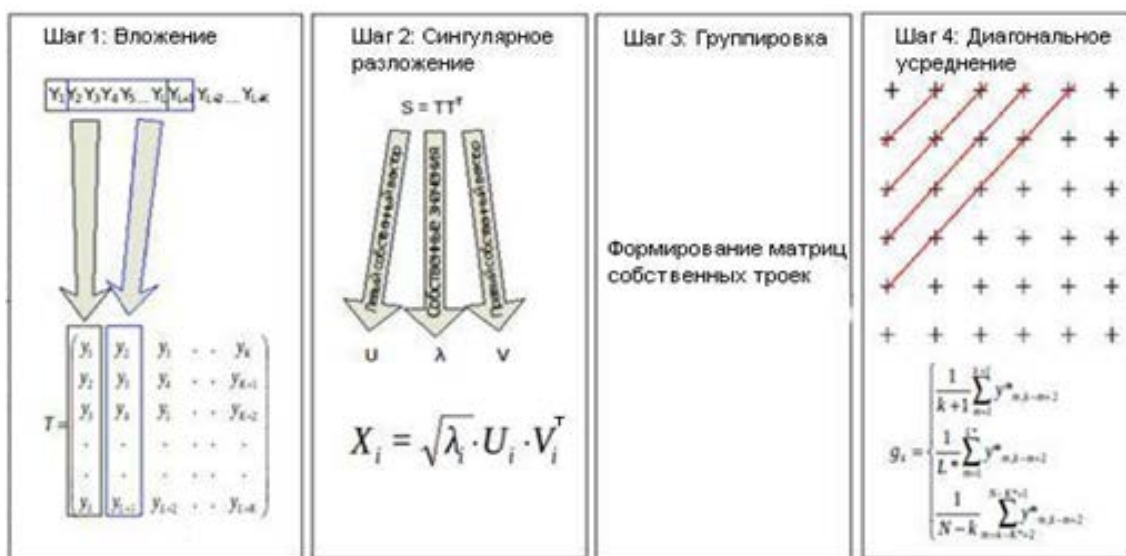


Рис. 1. Алгоритм сингулярно-спектрального метода фильтрации сигналов

В основе сингулярно-спектрального метода фильтрации с оптимальной матрицей  $A$  (SSA\_opt) лежит матрица вида:

$$A = P^T (S - Z) P B^{-1},$$

где  $P$  – матрица собственных векторов;  $S = X X^T$  – матрица измерений;  $Z$  – диагональная матрица погрешностей;  $B$  – диагональная матрица собственных чисел.

Исходные данные моделирования.

Модели полезного сигнала и погрешностей измерения – гауссовские, стационарные, эргодические случайные процессы.

Модель полезного сигнала при моделировании имеет корреляционную функцию

$$K_X(\tau) = \sigma_S^2 e^{-\alpha|\tau|},$$

где  $\sigma_S$  – среднеквадратическое отклонение полезного сигнала;  $\alpha$  – параметр корреляционной функции полезного сигнала.

В работе рассматриваются варианты моделей погрешности в виде белого дискретного гауссовского шума и в виде коррелированного случайного процесса.

Корреляционная функция погрешности в виде белого шума имеет вид:

$$K_V = \sigma_n^2 \delta_{ij},$$

где  $\sigma_n$  – среднеквадратическое отклонение помехи;  $\delta_{ij}$  – символ Кронекера.

Корреляционная функция погрешности в виде коррелированного случайного процесса имеет вид:

$$K_V(\tau) = \sigma_n^2 e^{-\beta|\tau|},$$

где  $\sigma_n$  – среднеквадратическое отклонение помехи;  $\beta$  – параметр корреляционной функции помехи.

Для моделирования алгоритмов сингулярно-спектральной фильтрации сигналов выбраны:

- полезный сигнал – стационарный эргодический случайный процесс (рис. 2, 3):

$$x_i = x_{i-1}e^{-\alpha\Delta} + \delta_s\sqrt{1-e^{-2\alpha\Delta}}v_i$$

$$v_i = \sqrt{-2\ln\xi_1[j]}\sin(2\pi\xi_2[j]),$$

где  $\xi_1[j]$ ,  $\xi_2[j]$  – независимые равномерно распределенные в интервале  $[0..1]$  случайные величины;  $\Delta$  – интервал дискретизации;  $\alpha$  – параметр корреляционной функции полезного сигнала;  $\sigma_s$  – среднеквадратическое отклонение полезного сигнала;

- модель погрешности в виде белого дискретного гауссовского шума (см. рис. 2):

$$v_i = \sigma_n\omega_i,$$

где  $\sigma_n$  – СКО помехи;  $\omega$  – нормированная дискретная гауссовская последовательность;

- модель погрешности в виде коррелированного случайного процесса (рис. 3):

$$v_i = v_{i-1}e^{-\beta\tau} + \delta_n\sqrt{1-e^{-2\beta\tau}}w_i;$$

- модель измерения линейная (см. рис. 2, 3):

$$y_i = x_i + v_i,$$

где  $y$  – наблюдаемый сигнал;  $x$  – полезный сигнал;  $v$  – погрешность измерения.

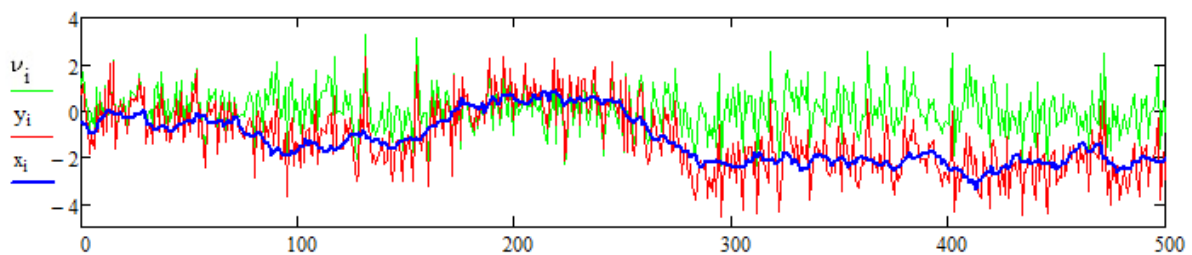


Рис. 2. Моделирование полезного сигнала ( $x$ ), наблюдаемого сигнала ( $y$ ) и погрешности измерения в виде белого шума ( $v$ )

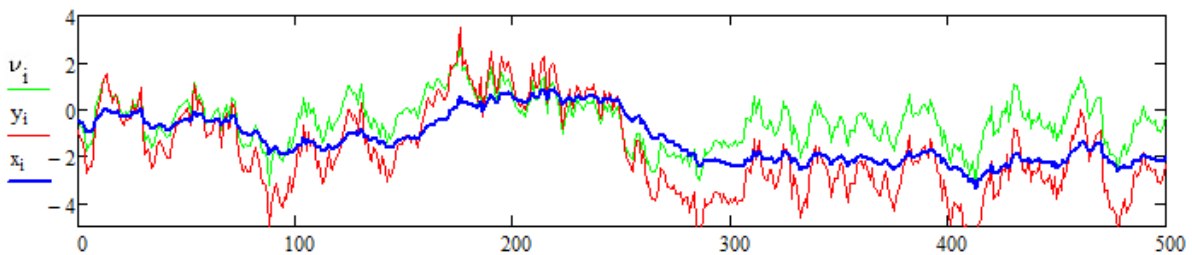


Рис. 3. Моделирование полезного сигнала ( $x$ ), наблюдаемого сигнала ( $y$ ) и погрешности измерения в виде коррелированного случайного процесса ( $v$ )

Результаты моделирования.

1. Зависимость показателя качества оценки алгоритма от длины окна  $R$  (рис. 4, 5).

Длина окна – один из основных параметров сингулярно-спектрального метода фильтрации сигналов, так как она определяет количество спектральных компонент, на которые раскладывается исходный ряд.



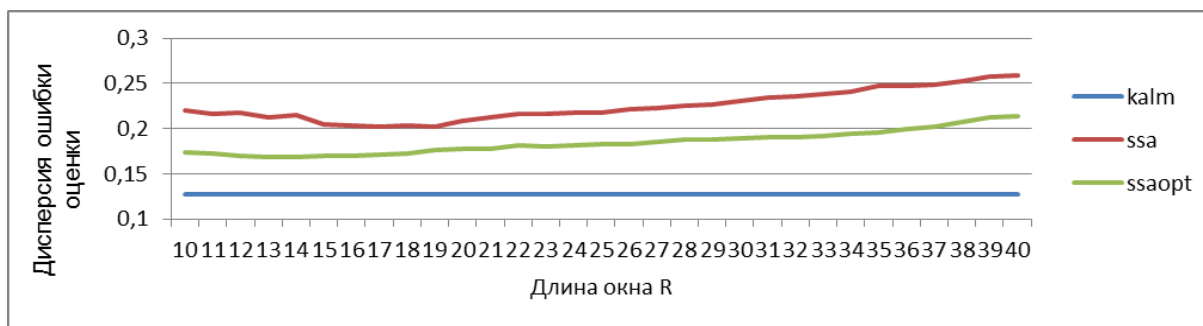


Рис. 4. График зависимости дисперсии ошибки оценки от длины окна при помехе – белый шум

Из графика, представленного на рис. 4, видно, что имеется оптимальное значение длины окна, при котором дисперсия ошибки оценки минимальна. Дальнейшее увеличение значений дисперсии ошибки оценки объясняется тем, что при увеличении длины окна  $R$  увеличивается количество элементов в траекторной матрице, что приводит к худшему усреднению. При уменьшении длины окна уменьшается учитываемая память системы, что также приводит к увеличению дисперсии ошибки оценки.

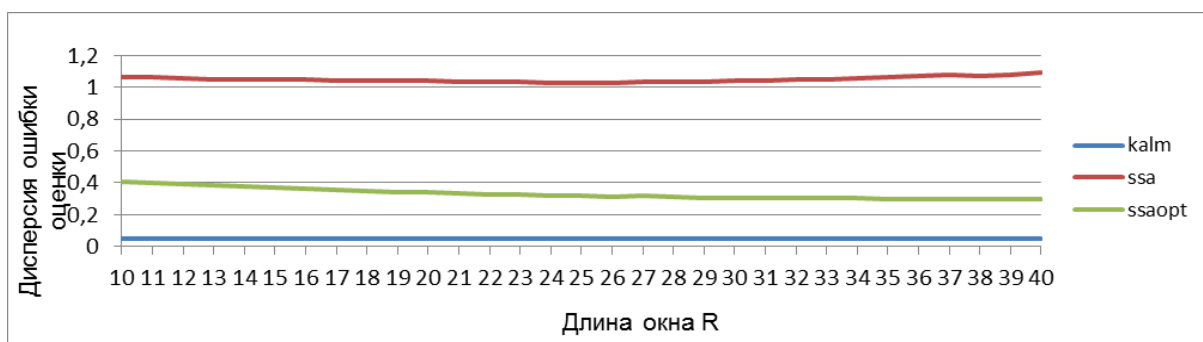


Рис. 5. График зависимости дисперсии ошибки оценки от длины окна при коррелированной помехе

Из графика, представленного на рис. 5, видно, что при коррелированной помехе необходимо использовать сингулярно-спектральный фильтр на основе оптимальной матрицы  $A$ .

2. Зависимость показателя качества оценки алгоритма от длины обрабатываемой последовательности  $N$  (рис. 6, 7).

Длина обрабатываемой последовательности  $N$  – это объем выборки («кусочек») всей последовательности, обрабатываемый на текущем шаге.

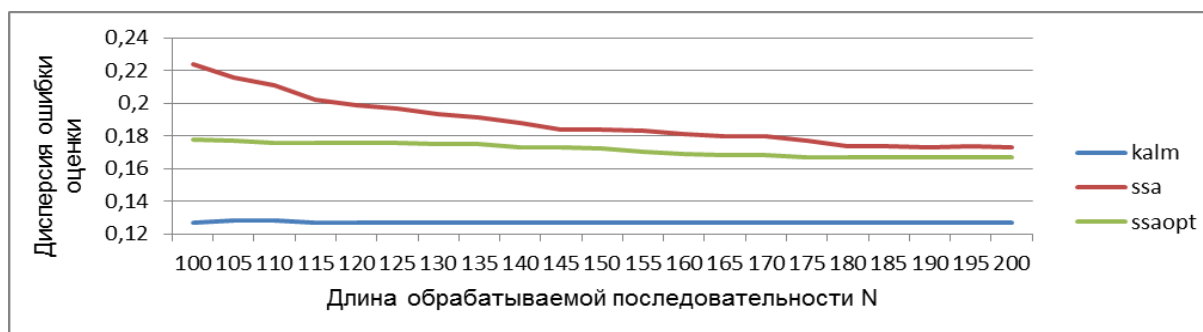


Рис. 6. График зависимости дисперсии ошибки оценки от  $N$  при помехе – белый шум

При увеличении длины обрабатываемой последовательности корреляционная матрица в алгоритме сингулярно-спектральной фильтрации становится точнее, поэтому становится точнее оценка полезного сигнала. Данная зависимость видна на графике, представленном на рис. 6.

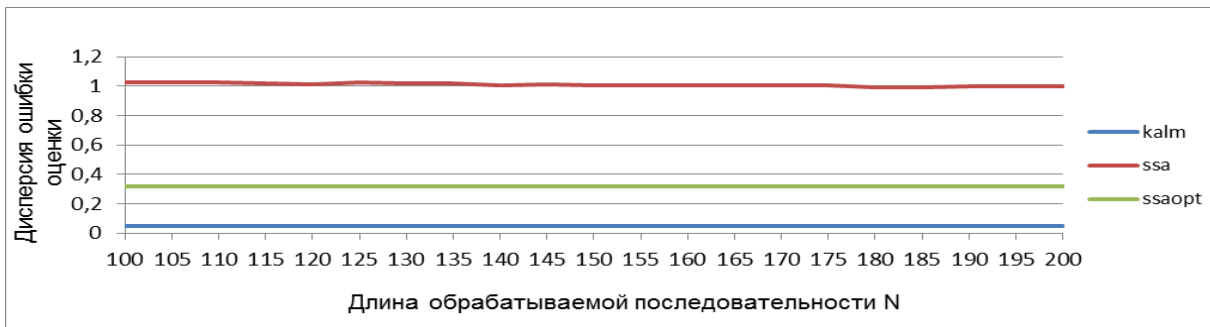


Рис. 7. График зависимости дисперсии ошибки оценки от  $N$  при коррелированной помехе

Из графика, представленного на рис. 7, видно, что при коррелированной помехе необходимо использовать сингулярно-спектральный фильтр на основе оптимальной матрицы  $A$ . Изменение значений длины обрабатываемой последовательности  $N$  не влияет на точность оценки полезного сигнала.

3. Зависимость показателя качества оценки алгоритма от отношения СКО помехи к СКО полезного сигнала (рис. 8, 9).

Отношение среднеквадратического отклонения помехи к среднеквадратическому отклонению полезного сигнала показывает, насколько сильно зашумлен сигнал.

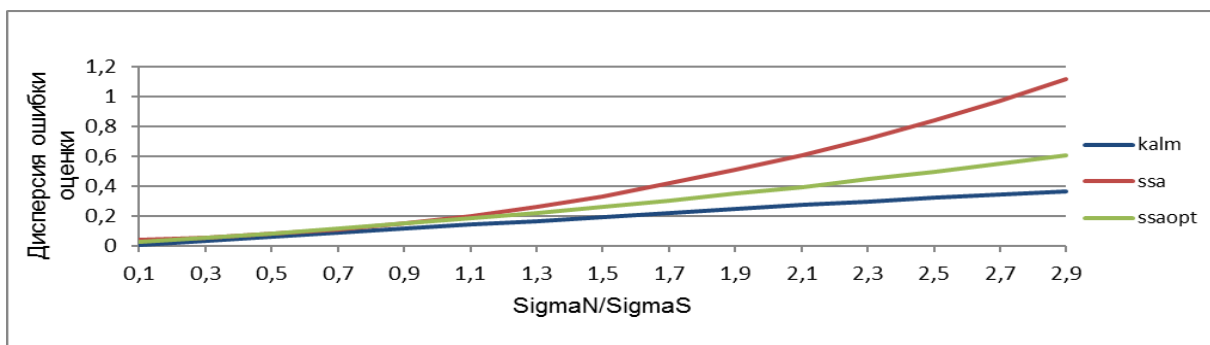


Рис. 8. График зависимости дисперсии ошибки оценки от  $\sigma_n/\sigma_s$  при помехе – белый шум

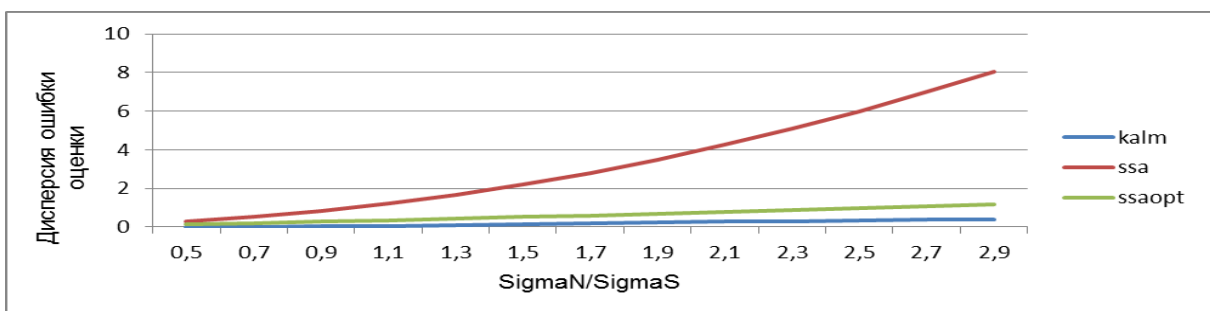


Рис. 9. График зависимости дисперсии ошибки оценки от  $\sigma_n/\sigma_s$  при коррелированной помехе

По графикам, представленным на рис. 8 и 9, видно, что при большей зашумленности входного сигнала труднее выделить полезный сигнал на фоне помехи.

4. Зависимость показателя качества оценки алгоритма от количества собственных троек сингулярного разложения (рис. 10, 11).

Количество собственных троек, используемых в алгоритме сингулярно-спектральной фильтрации сигналов, влияет на качество оценки полезного сигнала.

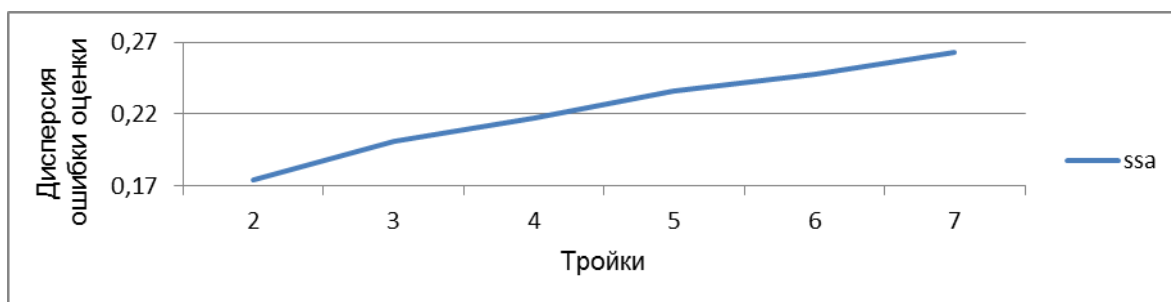


Рис. 10. Зависимость дисперсии ошибки оценки количества троек при помехе – белый шум

Из графика, представленного на рис. 10, видно, что при помехе в виде белого дискретного гауссовского шума в алгоритме сингулярно-спектральной фильтрации достаточно использовать две собственные тройки сингулярного разложения для получения лучшей оценки полезного сигнала.

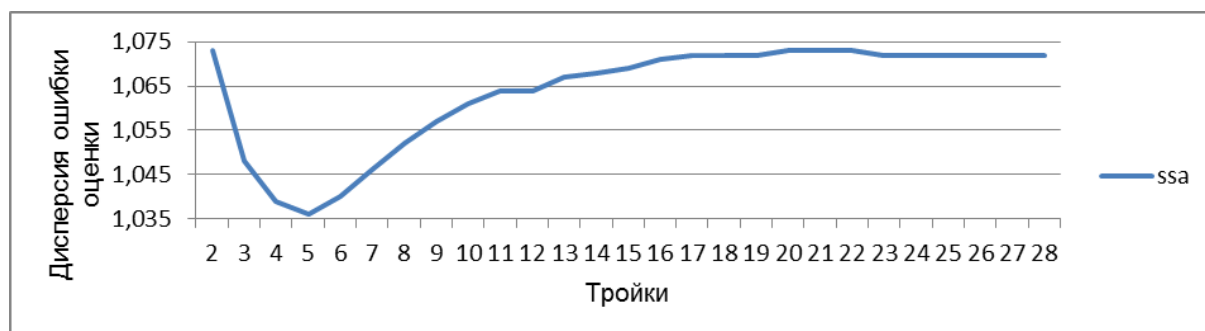


Рис. 11. Зависимость дисперсии ошибки оценки количества троек при коррелированной помехе

Из графика, представленного на рис. 11, видно, что при помехе в виде коррелированного случайного процесса в алгоритме сингулярно-спектральной фильтрации нужно использовать пять собственных троек сингулярного разложения для получения лучшей оценки полезного сигнала.

#### Выводы

Сингулярно-спектральный алгоритм фильтрации на основе собственных троек по дисперсии ошибки оценки уступает алгоритму, в основе которого лежит оптимальная матрица  $A$ . При определенном выборе параметров сингулярно-спектральный алгоритм на основе оптимальной матрицы  $A$  по точности приближается к фильтру Калмана. Данный метод фильтрации не требует представления сигнала в пространстве состояний, достаточно прост в реализации, применяется в условиях априорной неопределенности. Сингулярно-спектральный алгоритм фильтрации сигналов не требует стационарности и марковости ряда, знаний о корреляционной функции сиг-

нала. Для работы с методом нужно знать только, что модель измерения линейная, погрешность аддитивная, некоррелированная с полезным сигналом.

### **Библиографический список**

1. Огарков М. А. Методы статистического оценивания параметров случайных процессов. М.: Энергоатомиздат, 1990. 208 с.
2. Голяндина Н. Э. Метод «Гусеница»-SSA: анализ временных рядов: учеб. пособие. СПб., 2004. 76 с.

УДК 681.51

**Д. В. Васильев**

студент кафедры аэрокосмических измерительно-вычислительных комплексов

**А. П. Козионов**

научный руководитель

## **ИССЛЕДОВАНИЕ АЛГОРИТМОВ ОДНОКЛАССОВОЙ КЛАССИФИКАЦИИ ДЛЯ МОНИТОРИНГА СОСТОЯНИЯ ТЕХНИЧЕСКИХ ОБЪЕКТОВ**

Контроль состояния объектов является важной и сложной задачей, актуальной для промышленности, городского хозяйства, медицины и экономики. Это необходимо для обеспечения оперативного определения состояния сложных систем и прогнозирования критических ситуаций.

Алгоритмы контроля должны в режиме онлайн по сигналам с аппаратуры контроля определять аномалии в поведении объектов. Аномалии в поведении сигналов могут соответствовать как сбоям измерительной аппаратуры, так и критическому или близкому к критическому состоянию объекта. Зачастую алгоритмы контроля лишь сравнивают текущее значение измеряемого параметра с пороговым и выдают оператору сигнал в случае критических отклонений. В большинстве случаев данные алгоритмы не позволяют обнаружить критическое или близкое к критическому состоянию объекта, что может вызвать серьезные последствия.

В данной работе требуется исследовать методы одноклассовой классификации для задачи обнаружения аномалий (аномального состояния) и сравнительного анализа данных алгоритмов. Нередко данные аномальное поведение объектов скрыто в частотных свойствах сигналов, поэтому необходимо исследовать алгоритм мониторинга аномального состояния технических объектов на основе метода частотно-временного анализа и одноклассовой классификации сигналов.

Одним из наиболее перспективных направлений в области обнаружения аномального поведения сигналов является схема анализа данных, при которой из сигнала выделяется необходимая полезная информация, которая затем используется классификатором для определения аномалии в поведении сигнала.

В качестве метода выделения полезной информации из сигнала рассматривается частотно-временной анализ, результаты которого используется классификатором для определения вероятности аномалии для данного момента времени (рис. 1).

В данной работе будут рассмотрены следующие алгоритмы одноклассовой классификации [2, 3]: алгоритм «нейронные облака» (NC), алгоритм «машина опорных векторов» (SVM) и алгоритм «гауссовские смеси» (GMM). Для обучения классификатора используются лишь данные о нормальном поведении сигнала, данные о видах аномального поведения недоступны.



Рис. 1. Схема обнаружения аномалий в поведении сигналов

Сигналы, используемые в алгоритме, одномерные, нестационарные в узком смысле, не стационарность по дисперсии и математическому ожиданию. Возможные состояния нормального поведения сигнала должны присутствовать в обучающем множестве. Сигнал может быть зашумленным.

Алгоритм не должен определять, вызвана ли аномалия в поведении сигнала критическим состоянием объекта или сбоем/отказом измерителя.

Каждый из алгоритмов одноклассовой классификации имеет ряд параметров, исходя из выбора которых, повышается точность работы данных алгоритмов. В данной работе в качестве метода обучения алгоритмов одноклассовой классификации используется алгоритм перекрестной проверки для оценки параметров алгоритмов (рис. 2).

Перекрестная проверка – метод оценки аналитической модели и ее поведения на независимых данных [1]. При оценке модели имеющиеся в наличии данные разбиваются на  $k$  частей. Затем на  $k - 1$  частях данных производится обучение модели, а оставшаяся часть данных используется для тестирования. Процедура повторяется  $k$  раз; в итоге каждая из  $k$  частей данных используется для тестирования. В результате получается оценка эффективности выбранной модели с наиболее равномерным использованием имеющихся данных. В качестве показателя качества используется риск изготовителя.

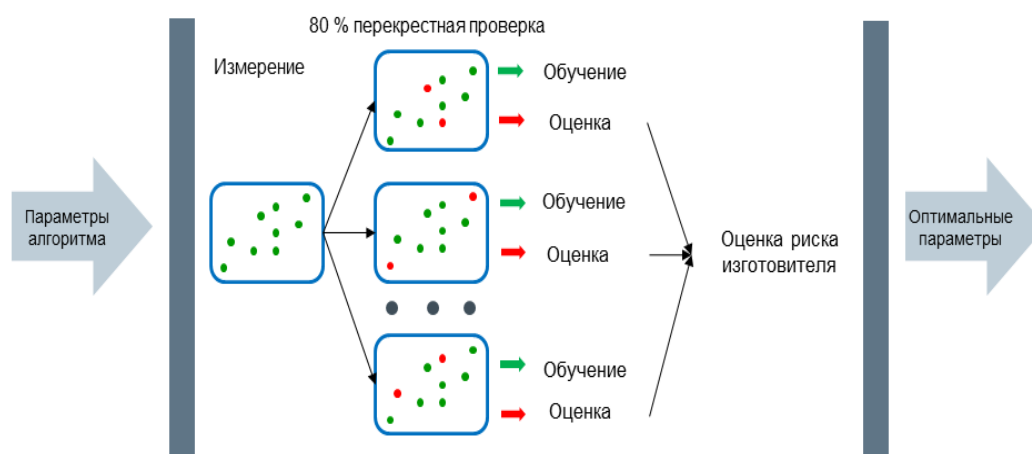


Рис. 2. Алгоритм перекрестной проверки для оценки параметров алгоритмов

Таким образом, на выходе каждого из исследуемых алгоритмов для заданного входного вектора выдается вероятность его принадлежности к области нормального поведения.

Моделирование аномалий

В данной работе для проведения сравнительного анализа точности обнаружения аномального состояния алгоритмами одноклассовой классификации были смоделированы следующие аномалии: выброс (рис. 3), ступенька (рис. 4), наложение шума на некотором участке сигнала (рис. 5).

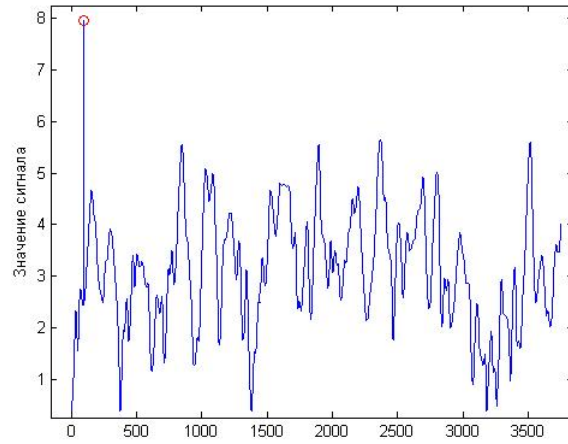


Рис. 3. Вид аномалии – выброс

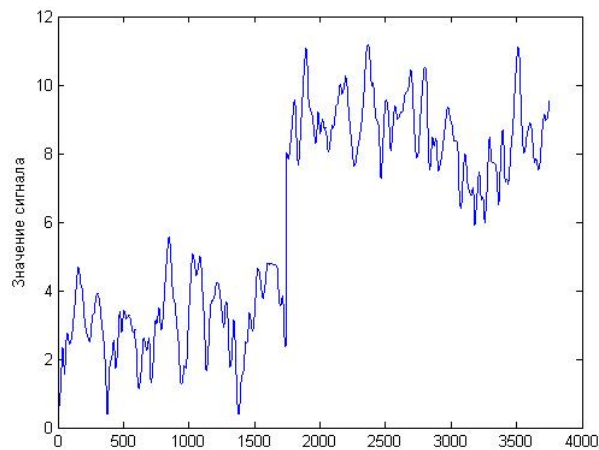


Рис. 4. Вид аномалии – ступенька

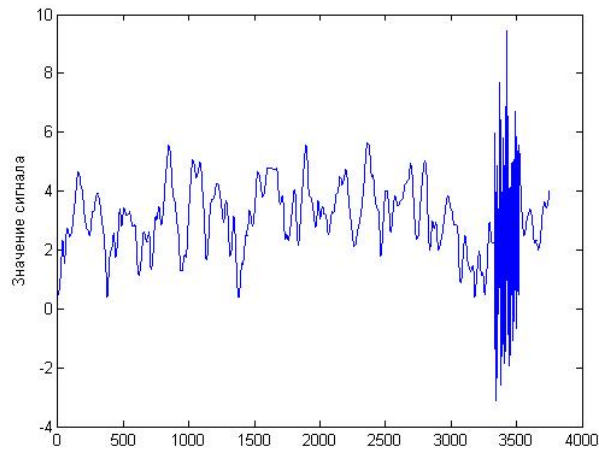


Рис. 5. Вид аномалии – наложение шума

Анализ результатов моделирования

В качестве показателя качества используется риск заказчика. Чем меньше риск заказчика, тем эффективнее работает алгоритм одноклассовой классификации.

Зависимость показателя качества оценки алгоритмов от величины дисперсии аномалии относительно сигнала для рассматриваемых видов аномалии приведены на рис. 6–8.

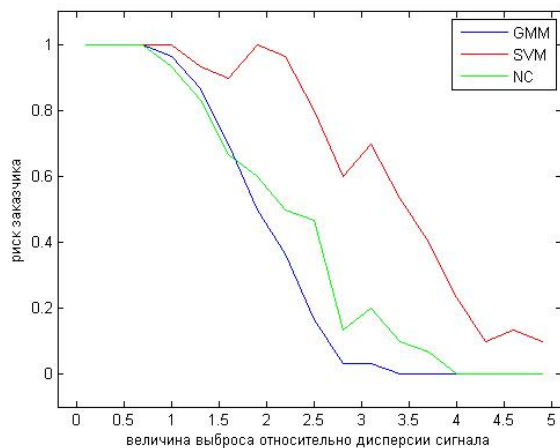


Рис. 6. График зависимости показателя качества оценки алгоритмов от величины дисперсии аномалии относительно сигнала. Вид аномалии – выброс

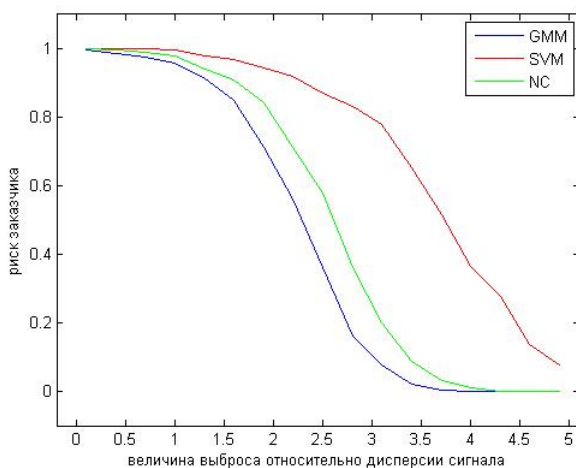


Рис. 7. График зависимости показателя качества оценки алгоритмов от величины дисперсии аномалии относительно сигнала. Вид аномалии – ступенька

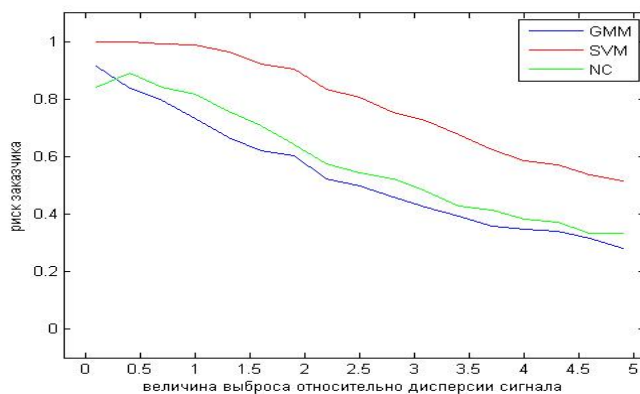


Рис. 8. График зависимости показателя качества оценки алгоритмов от величины дисперсии аномалии относительно сигнала. Вид аномалии – наложение шума

Пример обнаружения аномального поведения сигнала с помощью алгоритма «нейронные облака» (рис. 9).

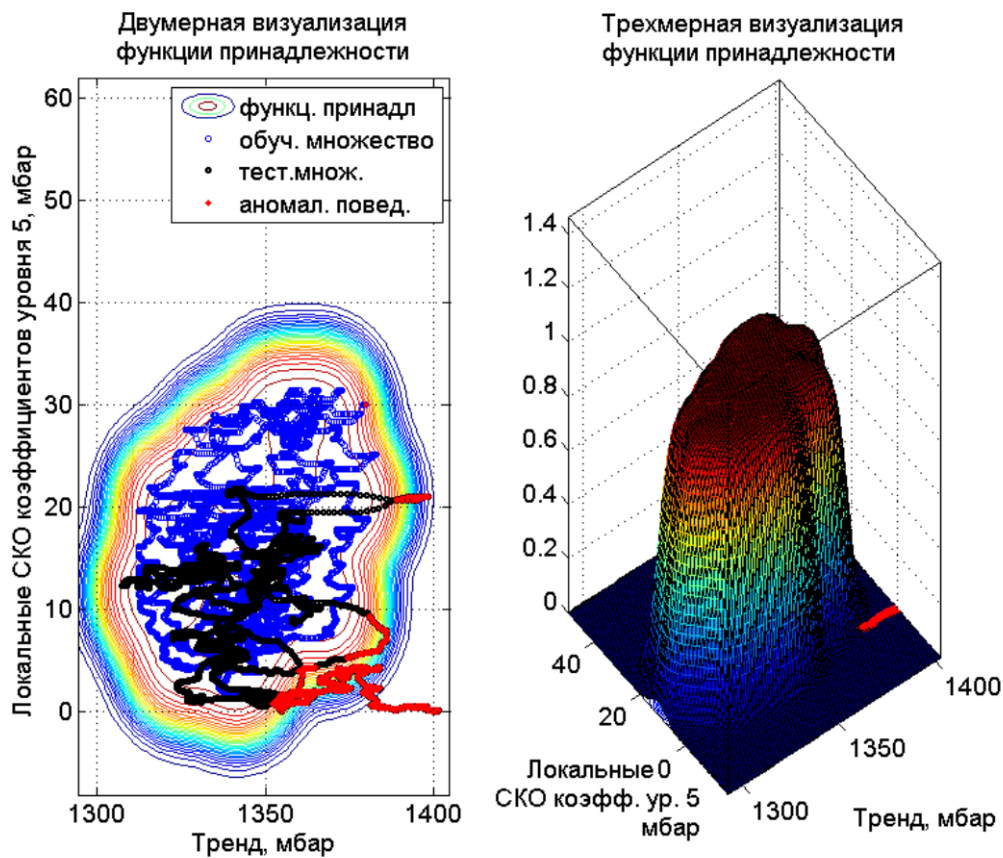


Рис. 9. Обнаружение аномалии с помощью алгоритма «нейронные облака»

#### Выводы

Для аномалии вида выброс все алгоритмы показывают высокую эффективность при дисперсии больше трех. Но алгоритмы «нейронные облака» и «машина опорных векторов» уступают алгоритму «гауссовские смеси».

Такие же результаты наблюдается для аномалии вида ступенька.

Для аномалии вида наложение шума на некотором участке сигнала все алгоритмы показывают менее высокую эффективность, даже при дисперсии равной пяти. Но алгоритмы «нейронные облака» и «машина опорных векторов» уступают алгоритму «гауссовские смеси».

#### Библиографический список

1. Nason G. P. Wavelet shrinkage using cross-validation/ Nason G. P. // Journal of the Royal Statistical Society. Series B (Methodological) 1996.
2. Lang B. Application of artificial intelligence in steel processing, Automatisierung in der Metallurgie / Lang B., Poppe T., Runkler T. // Heft 89 der Schriftenreihe der GDMB, 2001.
3. Lang B. Neural Clouds for Monitoring of Complex Systems / Lang B., Poppe T., Minin A., Mokhov I., Kuperin Y., Mekler A., Liapakina I. // Optical Memory and Neural Networks, 3, 2008.



УДК 62-83(075.8)

**А. Ю. Виноградов**

студент кафедры аэрокосмических компьютерных технологий

**Е. Н. Яблоков**

научный руководитель

### АНАЛИЗ ХАРАКТЕРИСТИК МОСТА ETHERNET–SPACEWIRE

При проектировании любого устройства необходимо выяснить его примерные характеристики и предпринять методы для их улучшения. Это позволит выбрать наиболее оптимальные способы решения задачи, позволит сэкономить ресурсы.

Мост Ethernet–SpaceWire необходим для передачи пакетов из сети SpaceWire [1] в сеть Ethernet [2] и обратно. Это позволят связывать устройства удаленные друг от друга на расстояние более 10 м, так же многие устройства используют сеть Ethernet.

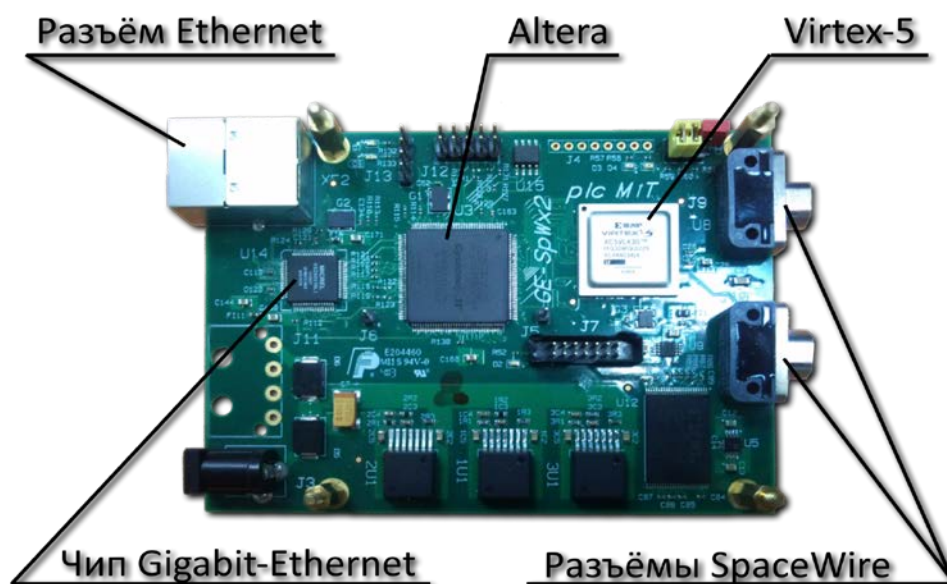


Рис. 1. Прототип моста Ethernet–SpaceWire

На рис. 1 представлен прототип устройства, имеющий на своем борту две ПЛИС – Virtex и Altera, 2 разъема SpaceWire, разъем Ethernet и чип Gigabit–Ethernet, работающий на скорости 10, 100 и 1000 мегабит [3].

В качестве характеристик рассмотрим мост как систему массового обслуживания с ограниченной очередью [4], для определения необходимого числа буферов. Рассчитаем максимальную скорость передачи и соотношение полезных байт к отправленным в зависимости от длины пакета.

Блок асинхронных буферов FIFO используется при приеме данных из сети Ethernet. Он позволяет раскладывать каждый приходящий фрейм в свой буфер.

На рис. 2 показаны типы систем массового обслуживания.

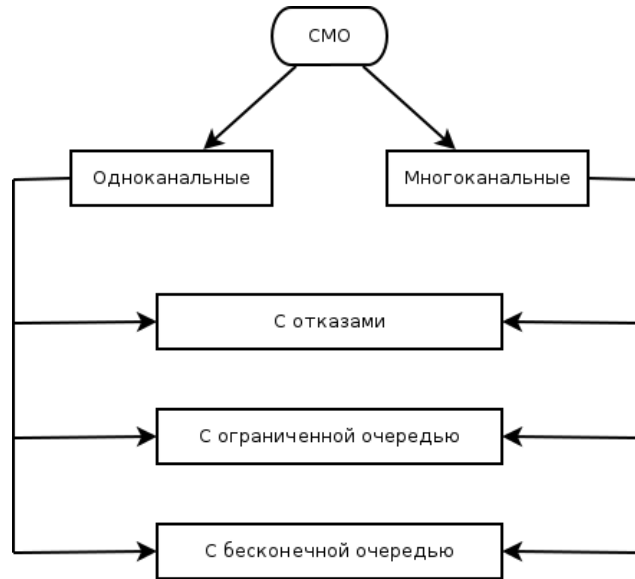


Рис. 2. Системы массового обслуживания

СМО – система, производящая обслуживание поступающих в нее заявок. СМО может содержать от одного до бесконечного числа обслуживающих приборов. СМО делятся на:

- системы с потерями, в которых входящие заявки, не нашедшие в момент поступления ни одного свободного прибора, теряются;
- системы с ожиданием, в которых имеется бесконечная очередь;
- системы с ограниченной очередью.

Для расчета вероятности отказа, в пакете Matlab был написан скрипт, который позволяет показать вероятность отказа для разного количества буферов и разной загруженности сети. Результат моделирования можно увидеть на рис. 3.

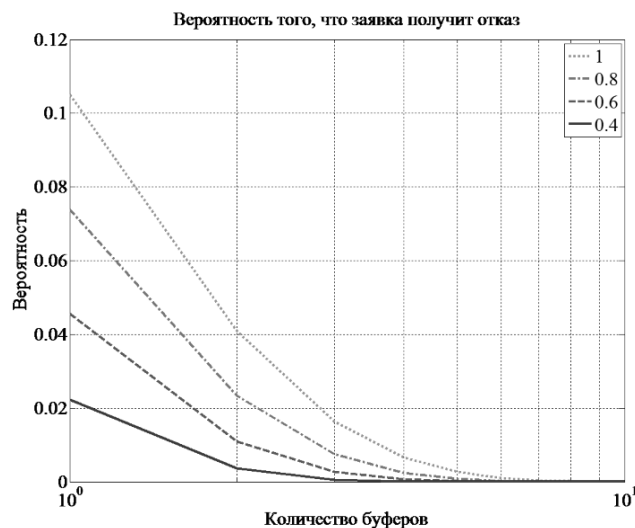


Рис. 3. Вероятность отказа

По графикам на рис. 3 видно, что уже при четырех буферах вероятность отказа очень мала.

В среде Matlab так же была рассчитана максимальная скорость передачи данных в зависимости от размера фрейма для сети Ethernet (рис. 4) и от размера пакета для сети SpaceWire (рис. 5).

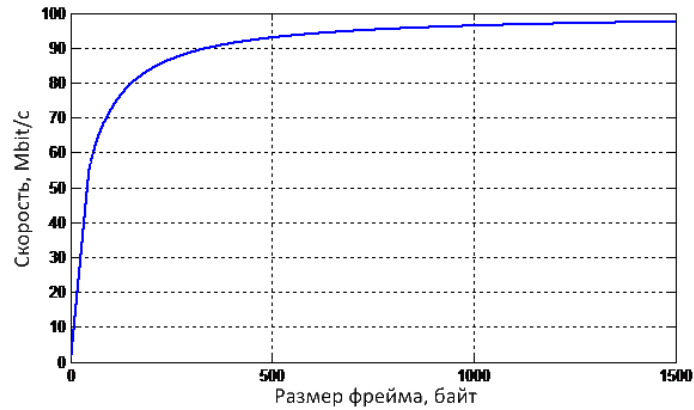


Рис. 4. Зависимость скорости от размера фрейма Ethernet

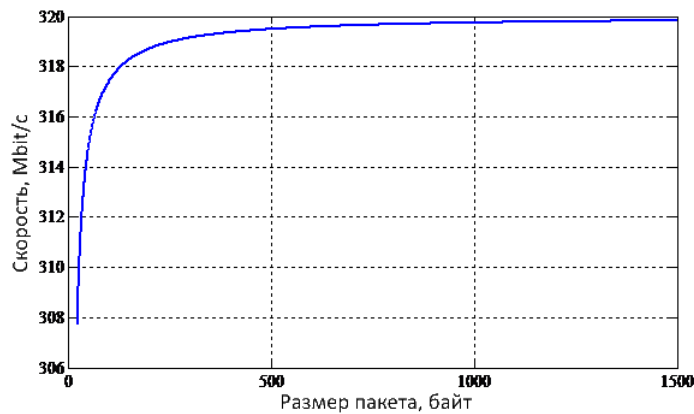


Рис. 5. Зависимость скорости от размера пакета SpaceWire

Из полученных графиков можно сделать вывод, что максимальную скорость будет определять сеть Ethernet. Для получения более высокой скорости, необходимо использовать гигабитный Ethernet.

На рис. 6 представлены графики, показывающие процентное соотношение полезных и отправленных данных в зависимости от длины поля данных.

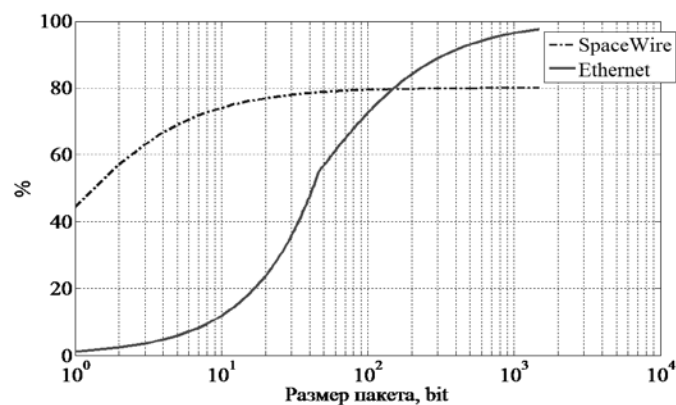


Рис. 6. Отношение полезных данных к отправленным

По графику видно, что максимальное количество передаваемой полезной информации в сети SpaceWire наступает уже при длине пакета в 70–80 байт, а в сети Ethernet при этих же значениях всего 35%.

Из этого следует, что при передаче маленьких пакетов из сети SpaceWire в сеть Ethernet необходимо группировать мелкие пакеты SpaceWire в один большой.

В результате исследования был проведен расчет вероятности отказа для разного количества буферов и с разной загруженности сети. В результате расчета выявлено, что в прототипе моста достаточно использовать четыре буфера. Размер каждого буфера равен 2048 байт, т. к. размер должен быть степенью двойки и при этом должен влезать максимальный пакет размером 1500 байт. Из графиков изображенных на рис. 4 и 5, можно сделать вывод, что Fast Ethernet не дает необходимой производительности, поэтому лучше использовать Gigabit Ethernet. По графикам на рис. 6 видно, что максимальное количество передаваемой полезной информации в сети SpaceWire наступает при длине пакета 70 байт, а в сети Ethernet только при размере пакета близком к максимальному.

### **Библиографический список**

1. ECSS Standard ECSS-E-ST-50-12C, "SpaceWire, Links, Nodes, Routers and Networks", Issue 1, European Cooperation for Space Data Standardization, July 2008.
2. Ethernet Standard IEEE 802.3
3. Micrel Inc, Gigabit Ethernet Transceiver with RGMII Support, Revision 1.2, February, 2014.
4. Гнеденко Б. В., Коваленко И. Н. Введение в теорию массового обслуживания. – М.: КомКнига, 2005.

УДК 658.512

**К. В. Гоголев**

студент кафедры системного анализа и логистики

**Н. Н. Майоров**

канд. техн. наук, доц. – научный руководитель

## **СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРИКЛАДНЫХ ПАКЕТОВ ПРОГРАММ ДЛЯ ПЛАНИРОВАНИЯ ГРУЗОВЫХ ПЕРЕВОЗОК НА АВТОМОБИЛЬНОМ ТРАНСПОРТЕ**

Одним из главных процессов, без которого невозможно обойтись при организации автомобильных перевозок, является планирование. Планирование автомобильных перевозок позволяет обеспечить эффективность коммерческих операций, уменьшить время их выполнения, своевременно отреагировать на уменьшение или увеличение потребительского спроса на услуги грузоперевозок.

Таким образом, в период высокой конкуренции, оптимальное планирование автомобильных перевозок выступает как эффективный метод привлечения клиентов.

Планирование автомобильных перевозок решает вопросы, связанные с разработкой маршрутов и выбора марки машины в соответствии с характеристиками предъявляемого к транспортировке груза, снижением затрат на автомобильную перевозку, расчетом времени на грузоперевозку с целью доставки груза в срок и т.д.

Планирование автомобильных перевозок базируется на анализе поступивших от клиентов заявок, проведении исследования основных маршрутов грузопотоков и составления в соответствии с полученными данными оптимального графика перемещения автомобилей.

В зависимости от решаемой задачи в практике планирования перевозок для получения оптимальных решений применяются различные математические методы. В связи с тем, что в качестве критерия оптимальности, как правило, используют экономические показатели, часто такие методы носят название экономико-математических [1].

Важным показателем эффективности планирования в автомобильных перевозках выступает непрерывность грузоперевозок (передвижения автомобилей, перевозящих груз) по имею-

щимся маршрутам. Непрерывность движения обеспечивается выходом в рейс автомобилей в согласно графику отправки, без задержек, и своевременное их прибытие на контрольные станции и в пункт назначения груза. Допускаются небольшие отклонения во времени движения и прибытия автомашин, в процессе автомобильной перевозки, в соответствии с предписанными нормативами.

Помимо планирования текущих грузоперевозок необходимо создавать планы на их корректировку в случае возникновения форс мажорных ситуаций и изменения обстановки на рынке данных услуг.

В данной статье, проведем сравнительный анализ программного обеспечения PTV «SmarTour», TopPlan «TopLogistic», 1С «БИТ.Управление транспортной логистикой».

PTVLogistics.SmarTour – программное обеспечение для автоматического логистического планирования распространения товаров и оптимизации грузоперевозок [2].

TopLogistic – это система управления перевозками позволяющая оптимально выполнять составление маршрутов доставки. Причем, составление маршрутов доставки выполняется наиболее эффективным образом, с учетом адресов точек доставки, временных интервалов доставки и многих других параметров. Фактически, эта система управления перевозками позволяет в масштабах больших, средних и малых предприятий организовать работу транспорта по доставке грузов наиболее оптимальным образом [3].

Система «БИТ.Управление транспортной логистикой» предназначена для автоматизации работы компаний, занимающихся внутригородской и региональной доставкой товара. Решение разработано по современным технологиям управления транспортной логистикой на базе платформы «1С:Предприятие 8». Программный продукт состоит из трех модулей: местная доставка (входит в основную поставку программы); автоматическое планирование доставки; GPS/ГЛОНАСС-мониторинг [4].

Рассмотрим внутреннюю среду программного обеспечения, то есть интерфейс каждого из продуктов.

Таблица 1

Внутренняя среда программного обеспечения

Критерий	SmarTour	TopLogistic	БИТ. Управление транспортной логистикой
Сложность работы	адаптивный интерфейс с заложенным в него математическим аппаратом		
Локализация	англоязычная	русифицированная	русифицированная
Доступность	– руководство пользователя; – дистрибутив CD; – ключ защиты USB.	– руководство пользователя; – дистрибутив CD; – ключ защиты USB. – возможность работы через браузер	– руководство пользователя; – дистрибутив CD; – регистрационная анкета; – ключ защиты USB.
Совместимость	– MS Office	– MSOffice – 1С Предприятие – Ахapta – Navision – MS SQL2000 - MS – SQL2008	– 1С:Предприятие 8. Управление торговлей; – 1С:Предприятие 8. Управление производственным предприятием; – 1С:Комплексная автоматизация 8. – БИТ. Экспедирование – БИТ. Автотранспорт (подробная информация о транспортных средствах, путевые листы, штафы, ремонт, топливо)

## ТЕХНИЧЕСКИЕ НАУКИ

Цена	не поставляется на российский рынок	от 25 000 до 160 000 руб. + доп. ПО и оборудование	от 49 000 до 500 000 руб. В зависимости от модуля ПО + доп. ПО и оборудование
------	-------------------------------------	--	--

Рассмотрим внутреннюю среду программного обеспечения, то есть возможности программы при решении поставленной задачи.

Таблица 2

### Внешняя среда программного обеспечения

Критерий	SmarTour	TopLogistic	БИТ.Управление транспортной логистикой
Визуализация	визуализация адресов и маршрутов доставки на электронной карте		
Транспортные средства	задание параметров транспортных средств(вес, объем, количество мест в партии)		
	возможность создания не только грузового автомобиля, но и прицепов, полуприцепов	задание государственного регистрационного знака транспортного средства	– задание государственного регистрационного знака транспортного средства; – выбор кузова: закрытый, рефрижератор
Количество водителей на одном транспортном средстве	два	один	возможность создания водителя и экспедитора
Точки отправления и потребления	учет времени открытия и закрытия точек	нельзя использовать одну точку доставки на нескольких маршрутах	учет времени открытия и закрытия точек
Критерий оптимизации	выбор критерия оптимизации маршрута (самый короткий, быстрый, экономичный)		
Маршрутизация	автоматическая маршрутизация позволяет распределить все заказы между автомобилями быстро и экономично с учетом всех заданных ограничений		автоматическое планирование маршрутов (при активации лицензии «Автоматическое планирование доставки»)

Программное обеспечение PTV «SmarTour», TopPlan «TopLogistic», 1С «БИТ.Управление транспортной логистикой» позволяет существенно повысить эффективность использования транспорта и экономию затрат на транспорт; оптимизировать грузовые перевозки, помогая правильно организовать управление этими перевозками в крупном городе, регионе за счет рациональной прокладки маршрутов доставки.

### Библиографический список

1. Майоров Н. Н., Фетисов В. А. Моделирование транспортных процессов: учеб. пособие. – СПб. ГУАП, 2011. – 165 с.
2. Программное обеспечение «SmarTour»[Электронный ресурс] // PTVPartner–Режим доступа:<http://www.ptv-vision.ru/>
3. Программное обеспечение «TopLogistic»[Электронный ресурс] // TopPlan Транспортная логистика – Режим доступа: <http://www.toplogistic.ru/>
4. Программное обеспечение «БИТ.Управление транспортной логистикой»[Электронный ресурс] // 1С.Первый БИТ–Режим доступа:<http://spb.1cbit.ru/>

УДК 629.7.018.3

**А. П. Григорьев**

аспирант кафедры аэрокосмических измерительно-вычислительных комплексов

**А. О. Чернелевский**

студент кафедры аэрокосмических измерительно-вычислительных комплексов

**В. Я. Мамеев**

канд. техн. наук, доц. – научный руководитель

### **СЦЕНАРНЫЙ МЕТОД КОНТРОЛЯ ЗНАНИЙ ПРИ РЕАЛИЗАЦИИ КОМПЕТЕНТНОСТНОГО ПОДХОДА НА ТЕОРЕТИЧЕСКОМ ЭТАПЕ ОБУЧЕНИЯ АВИАЦИОННЫХ СПЕЦИАЛИСТОВ**

Актуальные вопросы, связанные с реализацией компетентностного подхода при обучении авиационных специалистов, были подробно рассмотрены в [1, 3, 4, 12], что позволяет в рамках данной статьи проанализировать проблемные аспекты его реализации и рассмотреть сценарный метод контроля знаний при внедрении компетентностного подхода в обучающий процесс (ОП) высшей технической школы.

Рядом авторов [1, 3, 4] отмечается, что современные требования федерального государственного образовательного стандарта (ФГОС) с одной стороны направлены на интенсификацию ОП за счет снижения времени, отводимого на аудиторную лекционную и практическую подготовку [3], активизацию самостоятельной работы (СР) обучаемых, как правило, внеаудиторную [5, 12], развитие элементов собственной креативной и познавательной деятельности [4], а с другой направлены на сохранение и/или повышение текущего уровня качества обучения (успеваемость).

В настоящее время, реализация требований ФГОС, на современном этапе развития системы образования России, осуществляется за счет повсеместного и многоуровневого внедрения компетентностного подхода, подробно описанного в [1, 3, 4, 12]. Тем не менее, не смотря на актуальность и высокую эффективность [1, 3, 4] данного подхода при его непосредственной реализации возникает ряд характерных трудностей. Перечислим лишь наиболее важные из них:

- организация эффективной внеаудиторной СР без надлежащего контроля и с учетом снижения управляющей роли субъекта управления (преподаватель) затруднительна;
- организация эффективной процедуры аудиторной подготовки невозможна без внедрения инновационных методов организации теоретического обучения (модернизация методических и дидактических баз обеспечения ОП), использования электронных учебников (ЭУ), интеллектуальных ЭУ (ИЭУ) [7], средств мультимедиа (СМ) [11], а также использования принципиально новых подходов к процедуре контроля, диагностики и восстановления знаний. При этом особое внимание должно уделяться именно восстановлению утраченных и/или неполученных знаний [8].

Рассмотрим подробнее проблемные аспекты организации компетентностного подхода при обучении авиационных специалистов.

Во-первых, проблема эффективности внеаудиторной подготовки, заключающаяся в том, что ввиду сокращения аудиторных часов, сокращается общее время на подготовку, что неизбежно приводит к снижению контроля со стороны преподавателя с одновременным повышением значимости СР [5]. Таким образом, необходимо разрабатывать мотивационные сценарии обучения [9] и эффективные процедуры самоорганизации и самоконтроля обучаемых. Мотивационные сценарии в рамках аудиторной работы разрабатываются и внедряются непосредственно преподавателем-экспертом, а в рамках внеаудиторной подготовки – за счет внедрения мотивационных сценариев в состав дистанционной адаптивной обучающей системы (ДАОС) [9]. Аспекты, связанные с мотивацией обучаемых в рамках аудиторной подготовки подробно рассмотрены в [4], вопросы организации внеаудиторной теоретической подготовки в рамках СР за счет использования

обучаемыми ЭУ, ИЭУ и СМ в [5], а проблемные аспекты организации процедуры восстановления знаний в [8]. Резюмируя все выше сказанное, отметим, что заинтересованность обучаемых в контексте компетентного подхода формируется за счет разнородных дидактических, методических и алгоритмически-программных средств сопровождения ОП на этапе аудиторной подготовки (лекция) и поддерживается в рамках внеаудиторной работы, за счет эффективного использования средств ДАОС.

Во-вторых, необходимость повышать эффективность методов организации теоретической подготовки в рамках аудиторной подготовки, что немыслимо, в свою очередь, без внедрения ЭУ, ИЭУ и СМ, а также разработки инновационных средств контроля знаний. Таким образом, по мнению авторов данной статьи необходимо особое внимание уделить процессу реорганизации лекционных занятий в части перехода от «классно-урочных» методов к мультимедийным (лекции-дискуссии, лекции-презентации и т. д.), также должное внимание следует уделить вопросам, связанным с повышением уровня адаптации и интеллектуализации ЭУ [7]. Так как вопросы, связанные с модернизацией методической и дидактической баз сопровождения теоретического этапа ОП, были подробно рассмотрены в [3, 4], то целесообразно в рамках данной статьи акцентировать внимание на вопросах, связанных с контролем качества усвоения учебного материала на теоретическом этапе обучения.

Целым рядом авторов [2, 10] отмечено, что задачи контроля на разных этапах теоретического обучения существенно разнятся, что предполагает необходимость использования различных моделей контроля [2], следовательно, целесообразно провести модернизацию методов и алгоритмов контроля усвоения теоретического материала, исходя из требований конкретного этапа (входной, промежуточный, рубежный, итоговый).

Таким образом, рационально на разных этапах контроля использовать различные по своим характеристикам тесты (сценарии).

Под «Сценарием» или «Сценарным методом» будем понимать некоторый набор моделей и алгоритмов адаптивного тестирования, с непосредственными программными реализациями процедуры контроля знаний. Характерные особенности и описания всех, предлагаемых авторами статьи, сценариев приведены в [6, 13].

Обобщив теоретический материал из [6], проанализируем и соотнесем каждый из этапов контроля знаний с соответствующим сценарием, исходя из специфики задач ОП (таблица).

Таблица

Сценарии контроля знаний на основных этапах ОП

Этапы контроля знаний	Цели этапа контроля знаний	Сценарий контроля знаний, рекомендуемый авторами статьи
Входной	Оценка уровня знаний в начале обучения	Пизанская башня, Сизифов камень
Промежуточный	Оценка усвоения знаний в течение обучения	ГАШИГ, Путеводная нить Ариадны
Рубежный (Тематический)	Оценка усвоения знаний, после изучения темы, раздела и т. д.	Неоклассицизм, Маятник
Итоговый (предэкзаменационный и экзаменационный)	Применяется для систематизации пройденного материала в конце обучения	Маятник, Загадка сфинкса, Ахиллесова пята

Характерные особенности и описания всех рассмотренных сценариев приведены в [6, 13].

Пройти дистанционный адаптивный контроль знаний по специальным, гуманитарным и естественно-научным дисциплинам можно на сайте [14].



Таким образом, используя сценарный метод контроля знаний при реализации компетентностного подхода на теоретическом этапе обучения авиационных специалистов, нетрудно организовать сквозной непрерывный контроль знаний на разных этапах в течение всего процесса теоретической подготовки, что в свою очередь существенно повысит эффективность обучения и интенсифицирует его темпы.

В заключение стоит отметить, что для эффективной реализации ФГОС за счет внедрения компетентностного подхода необходимо решить весь спектр, описанных ранее задач, причем на одинаково высоком уровне, что, в свою очередь, требует кардинального переосмысления ОП и существенной модернизации методической и дидактической баз.

### Библиографический список

1. Андрухина Т. Н. Проектирование и реализация компетентностной модели профессиональной подготовки специалистов автомобильного транспорта // Вестник Самар. гос. техн. ун-та. Сер. Психолого-педагогические науки. – Вып. 1(9). – Самара: Самар. гос. техн. ун-т, 2008. – С. 3–7.
2. Глова В. И., Дуплик С. В. Модели педагогического тестирования обучаемых // Вестник Казан, гос. техн. ун-та им. А. Н. Туполева. 2003 г. № 2. С. 74–79.
3. Григорьев А. П. Реализация магистерских программ подготовки специалистов аэрокосмического профиля на базе компетентностного подхода. Педагогическая наука и современное образование. Сборник статей Всероссийской научно-практической конференции. Санкт-Петербург Издательство РГПУ им. А. И. Герцена, 2014 г. С. 330–335.
4. Григорьев А. П. Сопровождение и поддержка учебно-творческой деятельности специалистов аэрокосмического профиля на базе комбинированного аксиологокомпетентностного подхода. Перспективные формы, методы и приемы организации учебно-воспитательного процесса: из опыта работы. Сборник материалов Международной заочной научно-практической конференции. – Чебоксары: ЧГПУ им. Яковлева, 2014. – С. 19–22.
5. Григорьев А. П., Елчуева Т. Д. Организация самостоятельной работы при подготовке авиационных специалистов. 66-ая студенческая научно-техническая конференция ГУАП: Сб. докл. В 2 ч. Ч. I Технические науки / СПб.: ГУАП, 2013. 411 с.
6. Григорьев А. П., Демьянов А. А., Ивахива Т. А. Разработка алгоритмов дистанционного адаптивного тестового контроля знаний закрытого типа. Педагогическая наука и современное образование. Сборник статей II Международной научно-практической конференции, посвященной Дню российской науки. Санкт-Петербург. Издательство РГПУ им. А. И. Герцена, 2015 г. С. 235–239.
7. Григорьев А. П., Долгий А. И. Интеллектуальный электронный учебник для теоретической подготовки авиационных специалистов. Педагогическая наука и современное образование. Сборник статей II Международной научно-практической конференции, посвященной Дню российской науки. Санкт-Петербург. Издательство РГПУ им. А. И. Герцена, 2015 г. С. 239–242.
8. Григорьев А. П., Долгий А. И. Контроль, диагностика и восстановление знаний в дистанционных адаптивных обучающих системах. Педагогическая наука и современное образование. Сборник статей III Международной научно-практической конференции, посвященной Дню российской науки. Санкт-Петербург. Издательство РГПУ им. А. И. Герцена, 2015 г. С. 242–245.
9. Григорьев А. П., Чернелевский А. О. Разработка сценария мотивационного тестирования для системы дистанционного адаптивного контроля знаний. Педагогическая наука и современное образование. Сборник статей II Международной научно-практической конференции, посвященной Дню российской науки. Санкт-Петербург. Издательство РГПУ им. А. И. Герцена, 2015 г. С. 254–257.
10. Зайцева Л. В. Модели и методы адаптации к учащимся в системах компьютерного обучения. *Educational Technology & Society*. 6(4) 2003.
11. Кузнецова И. В. Профессиональная подготовка специалиста и информационные технологии // Высшее образование сегодня. – 2007. – № 12. – С. 51–54.
12. Пичкова Л. С. Организация самостоятельной работы студентов как фактор формирования профессионально значимых компетенций // Пути повышения конкурентоспособности экономики России в условиях глобализации, Мат. Конф. МГИМО (У) МИД РФ. – М.: МГИМО (У), 2008 г.
13. [http://cpmssmedia.dlinkddns.com/other\\_web/test2/index.php?route=proc](http://cpmssmedia.dlinkddns.com/other_web/test2/index.php?route=proc)
14. [http://cpmssmedia.dlinkddns.com/other\\_web/test2/index.php?route=ch\\_test](http://cpmssmedia.dlinkddns.com/other_web/test2/index.php?route=ch_test)

**Донхан Ким**

магистрант кафедры аэрокосмических измерительно-вычислительных комплексов

**А. В. Небылов**

д-р техн. наук, проф. – научный руководитель

**МЕТОДЫ ПОСТРОЕНИЯ ТРЕНАЖЕРОВ ДЛЯ ПИЛОТОВ ВЕРТОЛЕТОВ<sup>1</sup>**

Многочисленные вертолеты были разработаны в процессе развития авиационной промышленности, и требования к подготовке пилотов для них постоянно возрастают. Для снижения очень больших затрат и риска при обучении используют моделирование виртуального полета на тренажере, и это является одним из способов минимизации проблем, связанных с подготовкой пилотов вертолетов.

Авиационные тренажеры являются средством летной подготовки пилотов, которая может быть осуществлена на земле с имитацией ситуаций и действий, как на реальном вертолете.

Для оценки фактических затрат на летную подготовку важны такие факторы, как уровень аварийности полетов, возможные отказы на вертолете, насыщенность воздушного пространства воздушными судами, количество обслуживающего персонала и оборудования, развитость сети аэродромов, объем потребляемого топлива и т. д. Возникает необходимость в большом количестве человеческих и материальных ресурсов.

По сравнению с этим, в случае тренажера обучение проходит без каких-либо эксплуатационных расходов за пределами первоначальной установки тренажера и разработки методики обучения, стоимость обучения при этом существенно снижается.

**Классификация и применение тренажеров**

*Тактические тренажеры (Full Mission Simulator)* предназначены для отработки групповых действий. Они могут быть объединены в единую сеть с помощью интерфейса HLA (High Level Architecture), который позволяет совместно работать с разнородными тренажерами – авиационными, танковыми, артиллерийскими и др.

*Процедурные тренажеры (Flight Procedures Training Device)* предназначены для отработки экипажем процедур подготовки и выполнения полета.

В тренажерах такого назначения пульты, приборы и органы управления обычно имитируются с помощью сенсорных мониторов. Для удобства отдельные пульты и органы управления могут быть представлены в виде полноразмерных макетов.

Под *комплексными тренажерами (Full Flight Simulator)* понимают авиационные тренажеры, обеспечивающие подготовку экипажей в полном объеме их функциональных обязанностей по летной эксплуатации воздушного судна конкретного типа. Комплексные тренажеры – это тренажеры самого высокого уровня. Как правило, они имеют систему подвижности. Кабина комплексного тренажера выполняется в виде копии реальной кабины воздушного судна.

**Элементы необходимой эффективной тренажерной подготовки**

В реальной подготовке пилотов есть много ситуаций, в которых количество воздушных судов, эксплуатируемых одновременно, велико, и это создает большие трудности, чем полет в одиночку. При этом нужна независимая виртуальная симуляция в реальном времени взаимодействия между воздушными судами.

При выполнении тренинга в режиме реального времени взаимодействие между этими элементами миссии тактической подготовки через имитацию полета несколько учебно-тренировочных летательных аппаратов должно быть сделано одновременно. В этом случае эф-

<sup>1</sup> Автор имеет лицензию и опыт пилотирования вертолета в Южной Корее

фактивен в использовании перестраиваемый тренажер для поддержки множества различных типов воздушных судов с ограниченными ресурсами и расходами.

Один из таких реконфигурируемых тренажеров-симуляторов AVCATT (Aviation Combined Arms Tactical Trainer) разработан в США фирмой «L-3 Communications Holdings» (рис. 1) и может симулировать пять видов вертолетов (UH-60, OH-58D, AH-64H, AH-64D, CH-47).

В этой статье исследуется тренажер с изменяемой конфигурацией, чтобы быть подходящим для эффективной летной подготовки пилота вертолета.

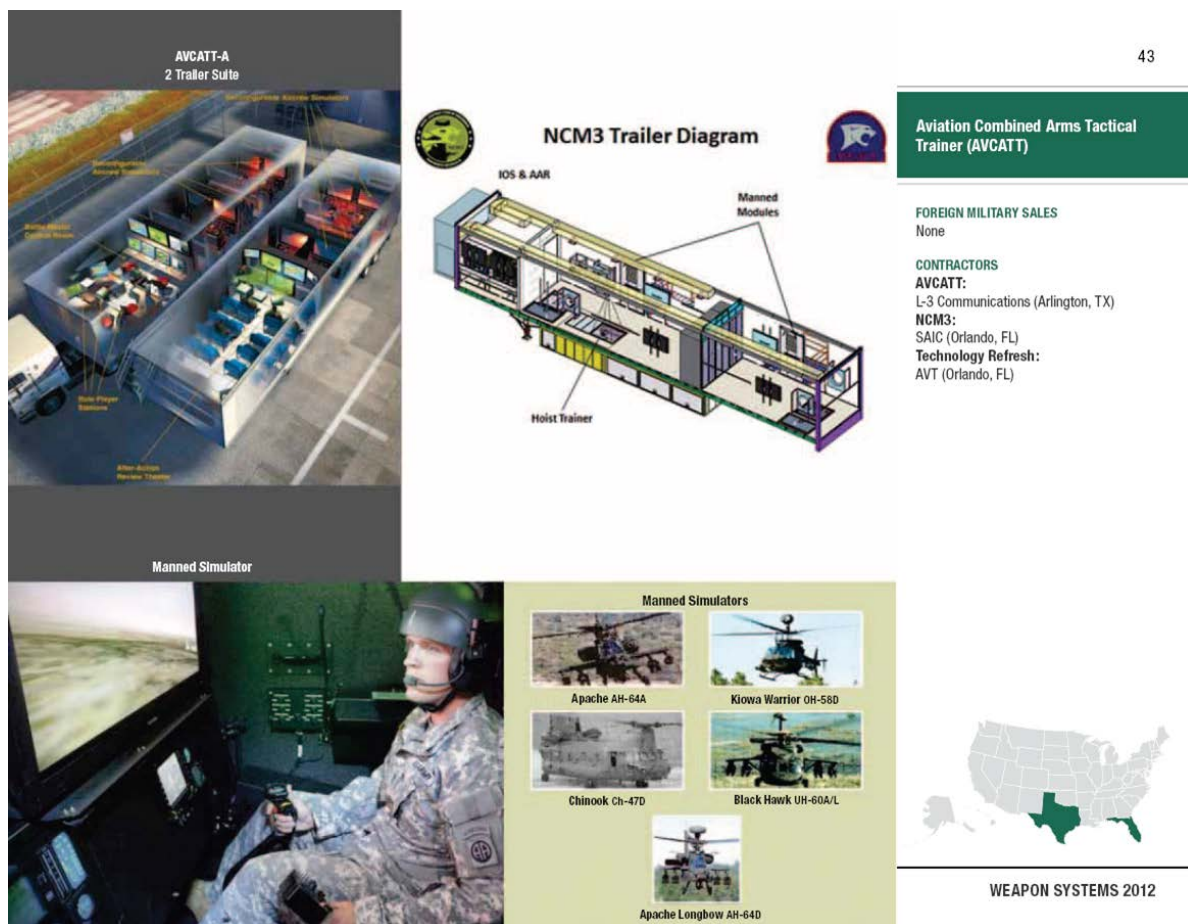


Рис. 1. Элементы тактического тренажера AVCATT [1]

### Конфигурация тренажера

Есть два базовых условия для тренажеров, используемых для обучения пилотов вертолетов навыкам полета:

- 1) должны быть реализованы разнообразные модели вертолетов, включая фактические формы бортовых приборов и органов управления;
- 2) подготовка должна быть доступна для выполнения различных миссий и для различных ситуаций полета.

Также различные тренажеры должны быть связаны между собой.

Реконфигурируемые тренажеры (RFS) являются одной из возможных реализаций различных моделей в симуляторе, и должны отображать различные ситуации для возможных тактических тренажеров при выполнении различных миссий.

Для этого реконфигурируемая кабина должна быть приспособлена для того, чтобы моделировать все типы вертолетов, работающих в стране или конкретной транспортной организации.

И если появляющаяся новая модель будет добавлена в будущем, она должна быть реализована в модульной конфигурации, или же используемый ЖК-дисплей должен быть приспособлен для реализации различных инструментов в кабине для конкретной модели вертолета (рис. 2). Для того, чтобы обеспечить описание авиационной аппаратуры для реального оборудования вертолета, модульная конфигурация будет иметь преимущества, но нужно учесть время, необходимое для изменения модели и замены модуля, и стоимость развитие аппаратных средств будет расти для каждого модуля.

При реализации инструментов с помощью ЖК-графического дисплея не реализуется полное совпадение с кабиной вертолета в реальной среде, но важны низкая стоимость и меньшее время, затрачиваемое для с изменения модели.

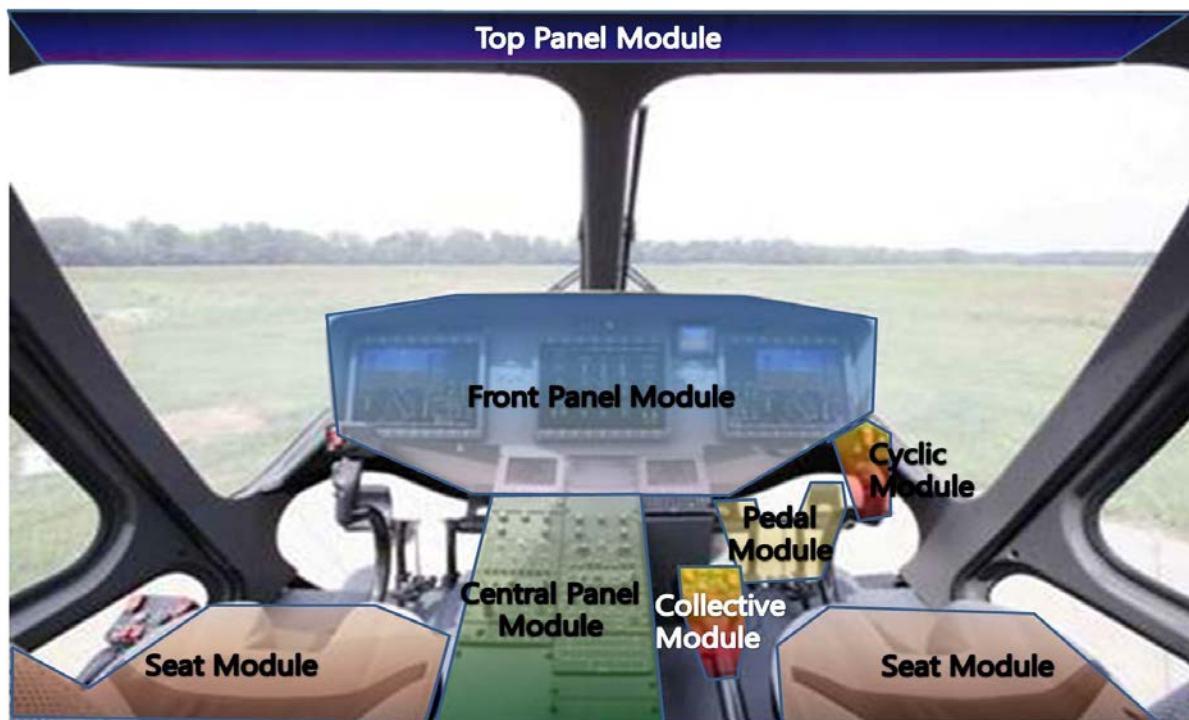


Рис. 2. Рулевые органы в кабине [2]

Основные компоненты кабины вертолета есть тип кабины и ее форма, размеры до и после перестройки, регулировка хода педалей и положение и форма органов управления.

И рабочее расстояние до передней панели, и положение органов управления могут быть определены путем контроля продольного расстояния до сиденья. В случае педалей форма и функции отличается для каждого типа педали и эти элементы должны быть разработаны, чтобы применить модульный принцип замены.

Органы управления, их форма и функциональность могут варьироваться в зависимости от типа воздушного судна, таким образом делая возможной замену.

В случае сиденья, форма каждого зависит от типа вертолета, но нет большой разницы между большинством форм.

В случае передней приборной панели, формы ее конструкции изменяются в процессе подготовки элементов модульного моделирования на сенсорном ЖК-экране.

В случае модульного конструирования размер и форма приборной панели различаются для каждого воздушного судна. Таким образом, существует способ легко заменить всю приборную панель, сделанную модульной, и варьировать только положение инструмента с помощью встроенных модулей каждого инструмента. В последнем случае, текущая форма панели прибо-

ров вертолета и симулятора могут быть разными, и есть достаточно времени для их замены, а стоимость снижается. В случае сенсорного типа ЖК-экрана, нужно всего лишь изменить модель в программном обеспечении, прибор может быть изменен и сделан непосредственно соответствующим фактическому типу воздушного судна.

Этот подход занимает меньше времени для изменения модуля, стоимость дешевле. Большинство вертолетов используют аналогичные инструменты. Таким образом, реализуют различные формы и функции, свойственные фактическому вертолету.



Рис. 3. ЖК-панели инструментов [3]

Для того чтобы имитировать все инструменты центральной/верхней панели различных моделей, можно использовать метод сенсорного ЖК-экрана (рис. 3) и способ создания модульных конструкций, используя реальные инструменты. Желательно использовать единый подход к сенсорного типа ЖК-дисплея или модульный подход в сочетании с передней приборной панелью. Центральная панель приборов вертолета в основном обеспечивает связь и навигационное оборудование, а верхняя панель содержит переключатели и плавкие предохранители для контроля двигателей и различных связанных с ними систем для целей подготовки пилотов. При этом было бы целесообразно использовать модульную систему, поскольку она дает подобие реального инструмента, а не искусственный образ как сенсорный тип ЖК-экрана, который отличается от реальных инструментов.

В перестраиваемом симуляторе модели конкретных изменений можно разделить на аппаратные изменения и изменения программного обеспечения. Изменения программного обеспечения обеспечивают изменение в моделировании конкретной характеристики: авионики, метода навигации/связи, миссии, двигательной системы, и других элементов системы, контроль инструментов управления и кабины. Аппаратные изменения, которые сводятся к замене одного модуля на другой, чтобы лучше соответствовать модели, имеют свои преимущества.

### Выводы

В современном обществе, где спрос на воздушные перевозки постоянно увеличивается, спрос на квалифицированных пилотов также увеличивается при возрастании стоимости летной

подготовки. Виртуальный тренажер является необходимым оборудованием для более эффективной подготовки пилотов. Необходимо решить много проблем, чтобы сделать обучение пилотов более эффективным с использованием тренажеров, они были разработаны для многих типов вертолетов и режимов полета, многие тренажеры в настоящее время в стадии разработки.

В этой статье даны рекомендации, как спроектировать тренажер для эффективной подготовки пилотов вертолетов. Автор учился на пилота вертолета и имеет опыт использования тренажеров.

Большинство тренажеров для подготовки пилотов, разработанных до сих пор, были разработаны в форме, которая может быть пригодной, чтобы имитировать только конкретный вертолет. Но постепенно все более интенсивно используют независимое виртуальное моделирование, обеспечивающее возможность перестройки тренажера под конкретный вертолет и возможность взаимодействия между тренажерами различных миссий. Такие подходы будут развиваться и высоко оценены специалистами.

### Библиографический список

1. Электронный источник: [http://semanticcommunity.info/Army\\_Weapon\\_Systems\\_Handbook\\_2012/Aviation\\_Combined\\_Arms\\_Tactical\\_Trainer\\_\(AVCATT\)](http://semanticcommunity.info/Army_Weapon_Systems_Handbook_2012/Aviation_Combined_Arms_Tactical_Trainer_(AVCATT)).
2. Original Picture: A cockpit view of the AgustaWestland AW169 helicopter.
3. Elite TH-100 Helicopter Flight Simulator, Thales New functions unveiled on ODICIS future cockpit prototype.

УДК 629.735

**В. С. Зайченко**

магистрант кафедры аэрокосмических измерительно-вычислительных комплексов

**В. Г. Никитин**

канд. техн. наук, доц. – научный руководитель

## СИСТЕМА ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ДВИЖЕНИЯ ТРАНСПОРТА НА АЭРОДРОМЕ

Обеспечение безопасности движения воздушных судов в зоне аэродрома при взлете и посадке, а также при маневрировании является актуальной задачей, требующей исследования с целью разработки более совершенных автоматизированных средств, обеспечивающих диспетчеров необходимой и актуальной для данного момента информацией о состоянии движения транспорта на аэродроме. Характерным примером отсутствия такой информации о движении специального автотранспорта в районе взлетно-посадочной полосы стала катастрофа самолета «Falcon» в аэропорту «Внуково» Москва, произошедшая 21 октября 2014 г. Во время разбега по взлетно-посадочной полосе № 1 самолет столкнулся со снегоборочной машиной. Стечение ряда обстоятельств, в том числе и человеческий фактор, привели к трагедии. Имеющиеся в настоящий момент технические средства, которыми оборудованы гражданские аэродромы Российской Федерации не полностью обеспечивают диспетчеров информацией об опасных ситуациях, связанных с передвижением транспортных средств (ТС) и возможных нарушениях. Используемые на данный момент на аэродромах Российской Федерации системы SMGCS имеют недостатки, связанные с переотражением радиолокационных сигналов, появлением так называемых «слепых» зон, вследствие перекрытия одного объекта другим, невозможностью идентификации типа и номера транспортного средства<sup>1</sup>. Предлагаемое в данной статье техническое решение позволит за счет выдачи дополнительной и достоверной информации о передвижении ТС снизить риск возникновения аварийной ситуации на территории аэродрома, путем предотвращения столкновения воздушных судов со спецтехникой.

<sup>1</sup> ИКАО doc. 9830 «Руководство по усовершенствованным системам управления наземным движением и контроля за ним (A-SMGCS)» Издание первое – 2004.

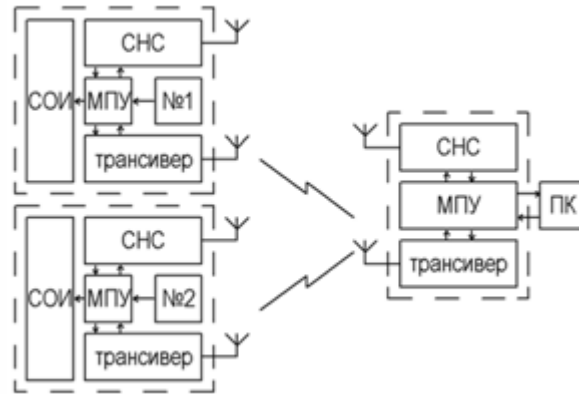


Рис. 1. Структурная схема системы

Разрабатываемая система обеспечения безопасности движения транспорта на аэродроме позволит устранить указанные выше проблемы за счет применения приемников спутниковой навигации GPS/ГЛОНАСС, позволяющих достаточно точно определять координаты ТС и тем самым их положение относительно взлетно-посадочных полос, рулежных дорожек и стоянок воздушных судов. Система может применяться как дополнение к используемым на сегодняшний день на территории Российской Федерации системам SMGCS (ИКАО doc. 9476).

Система включает в себя (рис. 1):

- пульт диспетчера (ПД), состоящий из приемника спутниковой навигационной системы (СНС), микропроцессорного устройства, трансивера и персонального компьютера;
- автономный модуль (АМ), установленный на борту каждого ТС, находящегося на территории аэродрома. АМ, в свою очередь, включает в себя приемник СНС, микропроцессорное устройство и трансивер для передачи диспетчеру информации о своем местоположении, а также для получения и выдачи оператору ТС предупреждений о возможности возникновения ОС непосредственно с пульта диспетчера.

Все АМ принимают сигналы одной и той же группы спутников, в результате чего, ошибки, связанные с неоднородностью тропосферного и ионосферного слоя, а также с погрешностью часов спутников и приемников, будут взаимно компенсироваться с применением дифференциального режима работы СНС. Для реализации работы СНС в дифференциальном режиме необходимо в каждый момент времени иметь данные о координатах как ТС (АМ), так и наблюдателя (ПД). Вычитая координаты АМ из координат ПД, можно получить относительные координаты объекта на территории аэродрома.

Разрабатываемая система оперирует полудуплексной линией связи для сужения канала передачи данных. Интервал между посылкой ПД и гарантированным ответом АМ определяется скоростью передачи данных и, как следствие, частотой несущей. Интервал между приемом посылки от АМ и посылкой нового запроса ПД должно быть как можно короче, поэтому обработка полученных данных должна производиться в интервалах передачи последующей посылки. В связи с этим, работа на уровне протокола осуществляется исключительно в прерываниях МПУ. Согласно данным требованиям, обмен данными производится в следующем порядке:

- ПД посылает команду-запрос с адресом очередного АМ (код состояния и номер угрожающего ТС) и переходит в режим приема. Код состояния служит для информирования оператора ТС о возможности движения в текущей зоне и предупреждении возникновения опасной ситуации;
- все АМ получают запрос, но обрабатывает и отвечает тот, номер которого указан в посылке ПД. В посылку АМ включает только информацию о своем местоположении;

- ПД, получив посылку от выбранного АМ, производит определение его относительных координат, скорости и направления перемещения, используя информацию о предыдущем местоположении и времени опроса. На основании полученных данных производится наложение на карту доступных положений установленного для конкретного ТС шаблона, геометрические размеры которого являются функцией скорости его перемещения. В случае «вхождения» шаблона в «запрещенную» зону, программа контроля изменяет содержимое кода состояния и передает его при следующем запросе. Если посылка не приходит в отведенное время, запрос повторяется, после чего, при повторной неудаче, производится запрос по следующему (согласно установленному для штата ТС списку) адресу, а оператору выводится информация о недоступности АМ;
- в случае корректного ответа АМ, по полученным данным ПД определяет, находится ли АМ в допустимой зоне как по положению, определенному режимом работы аэродрома, так и по скорости, предел которой для каждого АМ определен заранее;
- на основе полученных данных формируется информация для диспетчера, а также код состояния, отправляемый текущему АМ в следующей посылке;
- производится смена адреса АМ по установленному для штата ТС списку.

В результате работы над системой, разработана программа персонального компьютера для ПД, производящая информирование диспетчера о возникновении опасной ситуации. Внешний вид лицевой панели программы представлен на рис. 2.

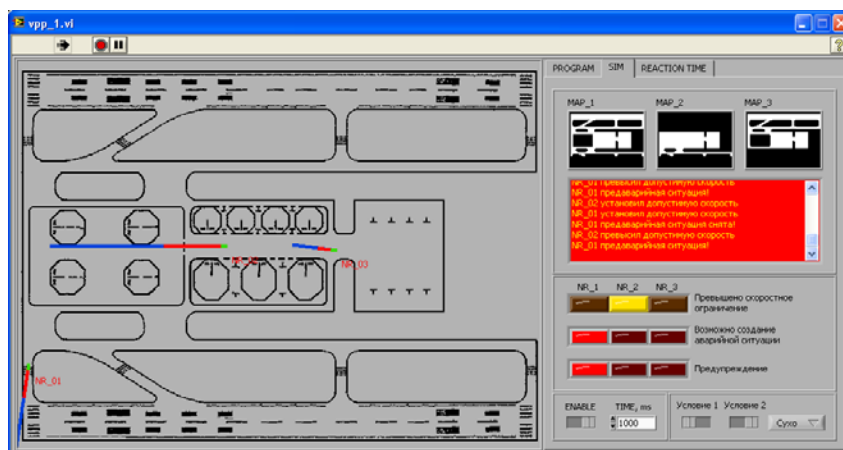


Рис. 2. Внешний вид лицевой панели программы

Рабочая область программы состоит из:

- окна вывода текущей обстановки на территории аэродрома (слева);
- селектора режима работы программы (справа). Здесь пользователю предлагается три варианта:

*PROGRAM*. Режим записи, сохранения и загрузки траектории транспортных средств для проведения моделирования;

*SIM*. Рабочая область диспетчера;

*REACTION TIME*. Режим задачи соответствия номера транспортного средства времени реакции оператора.

Режим SIM, как на наиболее интересующий с точки зрения диспетчера, состоит из:

- маски допустимых положений для каждого из ТС (изменяются в зависимости от условий работы аэродрома);
- окна вывода сообщений программы (также служит индикатором возникновения опасной ситуации, переходя в режим мерцания);



- индикаторов особых ситуаций;
- задатчика режимов работы аэропорта. Служит для смены масок допустимых положений в зависимости от режима работы аэропорта, а также погодных условий.

В окне вывода (рис. 2) выделено три объекта, траектории которых заранее заданы, а номера известны. Объекты включают в себя композицию из трех зон:

- зона, занимаемая непосредственно ТС (зеленая). Рассчитывается с учетом ошибок получения координат дифференциальной СНС;

- зона, рассчитанная как предельно возможная для остановки ТС (красная). Ширина данной зоны определяется шириной зоны ТС (зеленой). Длина данной зоны определяется по результатам измерения времени между получением координат, расстояния между точками, а также временем реакции оператора транспортного средства. При вхождении данной зоны в недопустимую область, определенную маской допустимых положений ТС, производится выдача звукового сигнала оператору соответствующего ТС, обязывающего его незамедлительно остановить движение своего ТС;

- зона предупреждения (синяя). Длина данной зоны определена как утроенная от длины красной зоны и является гарантированной для принятия решения об остановке. При пересечении данной зоны с маской допустимых положений, производится выдача предупреждения диспетчеру для незамедлительного выполнения мер по обеспечению безопасности в зоне движения угрожающего ТС.

Разрабатываемая система производит анализ текущей обстановки, выдачу рекомендательных сообщений диспетчеру и отправляет оператору ТС предупреждение при наступлении факта:

- превышения ТС скорости, установленной для конкретной зоны;
- нахождения ТС вне допустимой для него зоны (допустимость зон определяется также режимом работы аэродрома).

Как можно видеть из примера, представленного на рис. 2, программой был выделен объект NR\_01, вошедший в недопустимую зону, определенную занятой взлетно-посадочной полосой, а также объект NR\_02, превысивший установленную в зоне стоянки скорость. По результатам измерений временных и пространственных интервалов, а также контроля пересечений зон транспортных средств с масками допустимых положений изменяется состояние индикаторов особых ситуаций, что служит основой для принятия решения диспетчером.

Описанная выше система позволит в дополнение к используемой на данный момент системе SMGCS решить вопрос идентификации транспортного средства в рабочей зоне и, как следствие, наиболее оперативно провести процедуру устранения опасной ситуации.

УДК 656.025

**М. С. Колокольцова**

студент кафедры системного анализа и логистики

**Н. Н. Майоров**

канд. техн. наук, доц. – научный руководитель

### ПРИНЦИПЫ ФОРМИРОВАНИЯ ЛОГИСТИЧЕСКОЙ ЦЕПИ ПЕРЕДВИЖЕНИЯ В МЕГАПОЛИСЕ

Мегаполис – это высокая концентрация людей, машин, организаций в ограниченном пространстве. Транспорт в мегаполисе играет «связующую» роль и имеет огромное влияние на жизнь большого города. Поэтому транспортная система мегаполиса должна быть хорошо управляемой, сбалансированной, быстро реагирующей на все события.

Транспортные потребности и транспортные связи людей многообразны. Возможность или невозможность воспользоваться транспортной системой определяется показателем доступности. Доступность – состояние совокупности элементов транспортной системы (объектов транспортной инфраструктуры, транспортных средств, транспортных и сопутствующих услуг), при котором любой человек может воспользоваться ими для реализации своих транспортных потребностей.

Доступность подразделяется на следующие виды [1]:

- физическая (определяется формой и размерами объектов, а также расстоянием до места их расположения);
- информационная (определяется легкостью понимания человеком окружающей его ситуации в пространстве и во времени);
- организационно-технологическая (определяется режимом работы транспорта, стоимостью проезда; уровнем обслуживания, нормативными требованиями и др.).

Необходимым условием доступности транспортной системы является неразрывность ее элементов на всех этапах передвижения человека между пунктами отправления и назначения. Учитывая приоритетность этого условия, наиболее удобным способом формализации маршрутов передвижения людей является их представление в виде логистической цепи, состоящей из элементов. В случае недоступности хотя бы одного элемента логистическая цепь передвижения становится разорванной.

Можно формализовать схему транспортного обслуживания населения [2]:

- элементы транспортной системы макроуровня (вид или подвид транспорта, вид сообщения, объекты транспортной инфраструктуры);
- элементы мезоуровня (особенности конструкции транспортных средств, объектов транспортной инфраструктуры; транспортные и сопутствующие услуги);
- элементы микроуровня (технические характеристики элементов транспортной системы мезоуровня, например: уклон пандуса, высота ступеней, расстояние от поручня до стены, высота окна кассы и др.).

Такое деление по трем основным классификационным признакам удобно при планировании системы транспортного обслуживания населения. Для обоснования и реализации стратегических целей может быть достаточно использования элементов макроуровня. Для разработки нормативных правовых документов и программ развития использование этих элементов является необходимым, но недостаточным условием: требуется использование элементов мезоуровня. Элементы микроуровня целесообразно использовать при разработке документов технического регулирования.

Формализация логистических цепочек возможна на основе введения классификатора типовых элементов транспортной системы. Для разработки такого классификатора установлены следующие основополагающие элементы транспортного процесса:

- пешее передвижение;
- пешее передвижение на объектах инфраструктуры пассажирского транспорта, которое может быть связано или не связано с поездкой;
- передвижение в транспортном средстве.

Всех этих передвижений может вовсе не быть по причине отсутствия объектов транспортной инфраструктуры, транспортных средств, ценовой недоступности поездки, и по другим причинам, поэтому важно выделить дополнительный элемент, представляющий собой организационно-технологические условия и ограничения.

Указанные элементы транспортного процесса позволяют выделить следующие типовые макроэлементы транспортной системы:

- «П» – пешеходные пути;
- «И» – объекты инфраструктуры пассажирского транспорта;
- «Т» – транспортные средства;
- «У» – организационно-технологические условия и ограничения.

Используя эти элементы, можно составить макрологистическую цепь передвижения человека.

*Пункт отправления* → *пешее передвижение по пешеходным путям* → *пешее передвижение и размещение на объекте инфраструктуры пассажирского транспорта* → *посадка, передвижение в транспортном средстве и высадка из него* → *пешее передвижение на объекте инфраструктуры пассажирского транспорта* → *пешее передвижение по пешеходным путям* → *Пункт назначения*.

В формализованном виде эта же макрологистическая цепь будет иметь вид:  $A \rightarrow П \rightarrow И \rightarrow Т \rightarrow И \rightarrow П \rightarrow В$ .

А для каждого макроэлемента сформированы классификационные таблицы, содержащие типовой состав мезоуровня и их формализованное буквенно-числовое обозначение. Разработанная классификация и в целом методологический подход к определению доступности элементов транспортной системы позволяют создавать модели большинства логистических цепей передвижения людей.

В качестве примера моделирования логистической цепи на мезоуровне рассмотрим предложенный ранее пример передвижения человека. В неформализованном виде логистическая цепь передвижения будет иметь вид:

*Выход из дома* → *пешее передвижение к остановочному пункту (тротуар – переход через проезжую часть по нерегулируемому пешеходному переходу – тротуар)* → *остановочный пункт для ожидания автомобильного транспорта (место ожидания – информация о работе транспорта – оборудованное место для посадки в транспортное средство)* → *передвижение на автобусе общего пользования (передвижение в салоне – места для сидения – средства информационного обеспечения – пункты контроля оплаты проезда)* → *остановочный пункт (оборудованное место для высадки из транспортного средства)* → *пешее передвижение от остановочного пункта до метро (тротуар)* → *станция метро (зона входа на территорию – пешеходные пути на территории объекта – пункт продажи проездных билетов – место ожидания – место посадки в транспортное средство)* → *электрический транспорт (передвижение в салоне – места для сидения – комфортность поездки – средства информационного обеспечения)* → *станция метро (оборудованный остановочный пункт – пешеходные пути на территории объекта)* → *пешее передвижение к месту назначения (тротуар – переход через проезжую часть по регулируемому пешеходному переходу – тротуар)* → *пункт назначения*.

В формализованном виде логистическая цепь будет иметь вид:  $(A) \rightarrow П (1-21-1) \rightarrow И_0^a (14-20-15) \rightarrow T_{20}^a (04-05-09-10) \rightarrow И_0^a (15) \rightarrow П1 \rightarrow И_m^a (03-04-05-14-15) \rightarrow T_{20}^a (04-05-07-09) \rightarrow И_0^a (02-04) \rightarrow П (1-22-1) \rightarrow (B)$ , где (А) – пункт отправления; (В) – пункт назначения.

Формирование таких моделей может быть полезным при решении различных задач в сфере транспорта. Например: оценка транспортных потребностей населения и проектирование транспортной системы с учетом этих потребностей; проведение мероприятий по мониторингу и контролю доступности маршрутов передвижения для всех слоев населения по каждому элементу транспортной системы.

### Библиографический список

1. Логистика: общественный пассажирский транспорт. учебник для студентов экономических вузов/Под общ.ред. Л. Б. Миротина. – М.: Издательство «Экзамен», 2003. – 224 с.
2. Моделирование транспортных систем. Майоров Н. Н., Фетисов В. А. // СПб.: ГУАП, 2011. 165 с.

**А. В. Кононенко, Е. К. Мухина**

студенты кафедры аэрокосмических измерительно-вычислительных комплексов

**В. Г. Никитин**

канд. техн. наук, доц. – научный руководитель

## ИССЛЕДОВАНИЕ ЦИФРОВОЙ ФИЛЬТРАЦИИ СИГНАЛОВ МИКРОМЕХАНИЧЕСКИХ ДАТЧИКОВ

Микромеханические датчики находят самое разнообразное применение во многих областях человеческой деятельности, и в том числе в пилотажно-навигационных комплексах летательных аппаратов [1, 2]. Единство материала и технологий микроэлектроники позволяет создавать миниатюрные конструкции на одном кристалле, объединяющем в себе чувствительные элементы, преобразующие и электронные компоненты, вычислительные устройства. Для повышения точности необходимо устранить влияние помех с помощью фильтрации сигналов микромеханических датчиков [2]. В данной работе исследуется цифровая фильтрация сигналов датчиков путем компьютерного моделирования цифровых КИХ (с конечной импульсной характеристикой) и БИХ (с бесконечной импульсной характеристикой) фильтров, проводится сравнение и анализ их характеристик.

КИХ-фильтр – фильтр с конечной импульсной характеристикой (нерекурсивный фильтр) – один из видов электронных фильтров, характерной особенностью которого является ограниченность по времени его импульсной характеристики (с какого-то момента времени она становится точно равной нулю). Знаменатель передаточной функции такого фильтра – некая константа. БИХ-фильтр – фильтр с бесконечной импульсной характеристикой (рекурсивный фильтр) – электронный фильтр, использующий один и более своих выходов в качестве входа, то есть образует обратную связь. Основным свойством таких фильтров является то, что их импульсная переходная характеристика имеет бесконечную длину во временной области, а передаточная функция имеет дробно-рациональный вид. Как и любой другой алгоритм измерительно-вычислительной системы, алгоритм цифровой фильтрации имеет математическое описание, а именно разностное уравнение во временной области:

$$y(n\Delta t) = - \sum_{j=1}^{M-1} b(j)y(n\Delta t - j\Delta t) + y(n\Delta t) = \sum_{i=1}^{N-1} a(i)x(n\Delta t - i\Delta t)$$

где  $x(n\Delta t)$  и  $y(n\Delta t)$  –  $n$ -ые отсчеты входного и выходного сигналов фильтра, взятые через интервал  $\Delta t = \Delta t_n$ ;  $a(i)$  и  $b(j)$  – постоянные коэффициенты цифрового фильтра [3]. Это описание является наиболее общим и объединяет оба вида цифровых фильтров.

Основное свойство любого фильтра – это его частотная и фазовая характеристики. Они показывают, какое влияние фильтр оказывает на амплитуду и фазу различных гармоник обрабатываемого сигнала. Аналоговые фильтры строятся на RC цепочках, так что, чтобы внести изменения в фильтр, нужно менять фильтр аппаратно, в то время как преимущества цифровых фильтров заключается в том, что они являются программируемыми. Для того чтобы изменить частоту среза, скорость спада или ФЧХ, все, что необходимо сделать – это изменить небольшое число коэффициентов. Можно довольно легко произвести радикальные изменения, например, превращение низкочастотного фильтра в высокочастотный. Идея изменения характеристик фильтра с помощью изменения небольшого числа коэффициентов открывает и более широкие перспективы его использования.

В работе в качестве шума рассматривается высокочастотная электромагнитная помеха, источниками которой могут быть как функциональными, так и нефункциональными. Данная помеха может лежать в частотах от сотен килогерц до единиц мегагерц. Для моделирования была

взята помеха частотой 1 МГц. Ее вид в общем случае – вид суммы всех воздействующих помех, но для конкретной задачи проверки цифровых фильтров была выбрана синусоида с амплитудой, много меньшей амплитуды полезного сигнала. Для фильтрации помехи используется фильтр низких частот. Фильтры низких частот (НЧ-фильтры, low-passfilters) пропускают без изменения все частоты ниже заданной и удаляют из сигнала все частоты выше заданной. Для фильтрации выходного сигнала мы взяли микромеханический (MEMS) датчик давления, измеряющий давление с помощью упругой мембраны, на плоскость которой с одной из сторон действует измеряемое давление, вызывая прогиб, который далее регистрируется механическим или электрическим способом.

Моделирование цифрового фильтра осуществлялось в среде MATLAB. Пакет «SignalProcessingToolbox» MATLAB имеет интерактивную среду для проектирования и анализа фильтров «FilterDesignandAnalysisTool»(FDATool) с графическим интерфейсом пользователя «GUI». FDATool может использоваться:

- для проектирования фильтров, в том числе адаптивных и многоскоростных;
- для анализа цифровых фильтров;
- для модификации существующих фильтров;
- для квантования фильтров и анализа результатов квантования;
- для выполнения частотных преобразований цифровых фильтров [4].

Разработаны следующие модели цифрового фильтра:

- с конечной импульсной характеристикой (КИХ) (рис. 1,а);
- с бесконечной импульсной характеристикой (БИХ) (рис. 1,б).

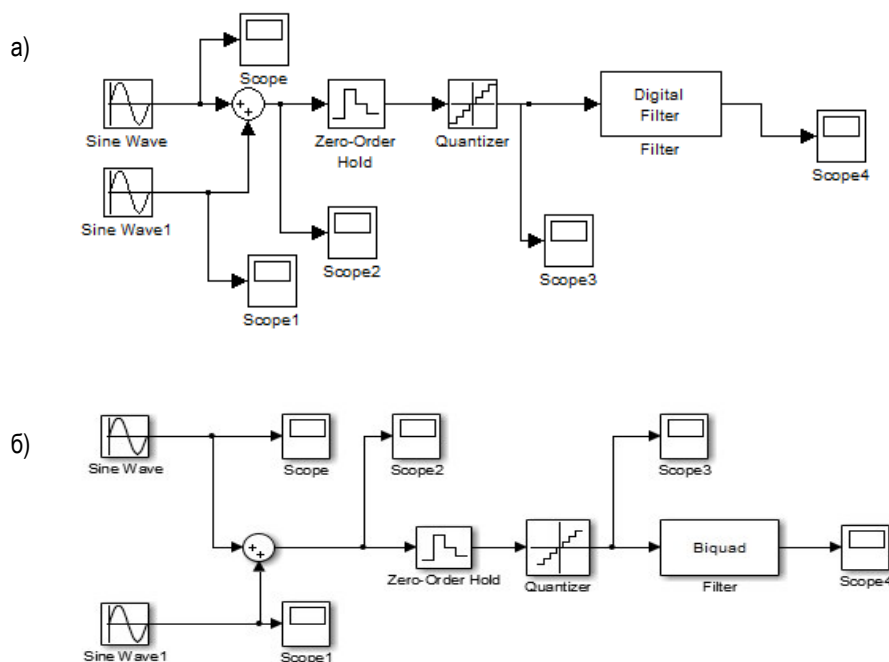


Рис. 1. Моделирование цифровых фильтров

В данных моделях: SineWave – полезный сигнал, SineWave1 – помеха, сумматор двух сигналов, идущих с генераторов, дискретизатор Zero-OrderHold, преобразующий непрерывный сигнал, поступающий с сумматора, в дискретный, квантователь Quantizer, преобразующий дискретный сигнал в цифровой; пять осциллографов: Scope (показывает полезный сигнал), Scope1 (показывает помеху), Scope2 (показывает непрерывный сигнал с выходного сумматора), Scope3 (показывает цифровой сигнал перед фильтрацией) Scope4 (показывает спектр цифрового сигнала после фильтрации).

Для фильтрации сигнала был выбран фильтр Чебышева по причине простоты проектирования. Фильтр Чебышева – один из типов линейных аналоговых или цифровых фильтров, отличительной особенностью которого является более крутой спад амплитудно-частотной характеристики (АЧХ) и существенные пульсации амплитудно-частотной характеристики на частотах полос пропускания (фильтр Чебышева I рода) и подавления (фильтр Чебышева II рода), чем у фильтров других типов. Фильтры Чебышева обычно используются там, где требуется с помощью фильтра небольшого порядка обеспечить требуемые характеристики АЧХ, в частности, хорошее подавление частот из полосы подавления, и при этом гладкость АЧХ на частотах полос пропускания и подавления не столь важна.

Для проектирования были выбраны следующие типы фильтров:

равноволновой КИХ-фильтр (Equiripple) – фильтр, минимизирующий максимальную ошибку в полосах задерживания и пропускания (то есть оптимальный в минимаксном смысле). Известно, что необходимым и достаточным условием этого является равенство всплесков частотной характеристики;

БИХ-фильтр Чебышева I-го порядка. У него отсутствуют нули частотной характеристики, пульсации в полосе пропускания, также нет пульсаций в полосе задержки, более короткая область перехода, чем у КИХ-фильтров.

В ходе моделирования были получены некоторые результаты, которые отобразились в графиках на выходах виртуальных осциллографов. Для большей наглядности все полученные графики представлены без масштабирования и в увеличенном масштабе (рис. 2–9).

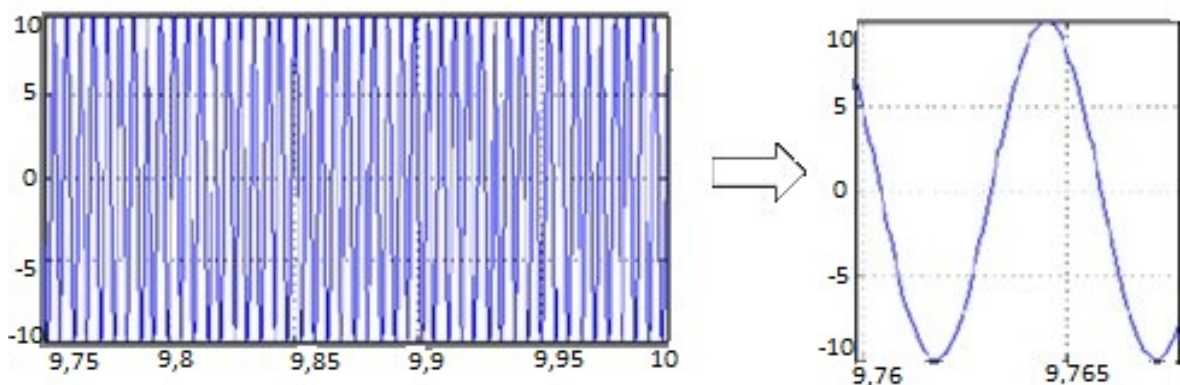


Рис. 2. Полезный сигнал

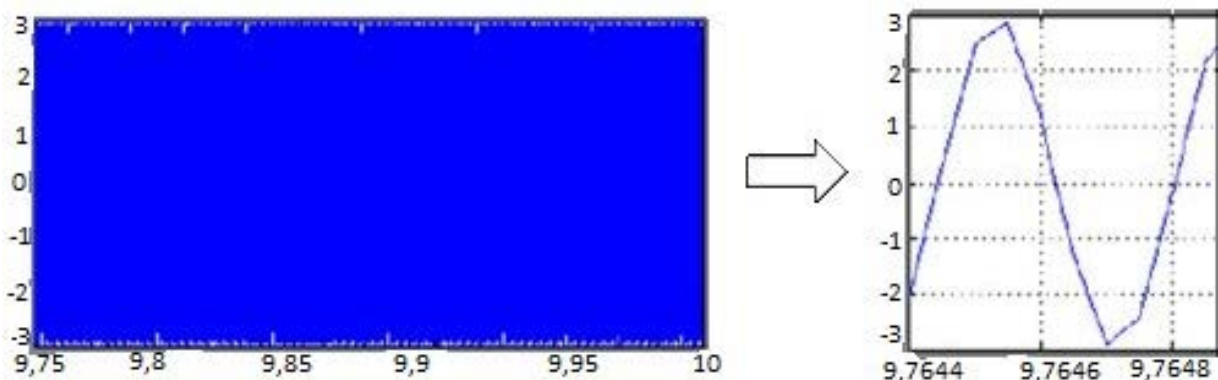


Рис. 3. Помеха

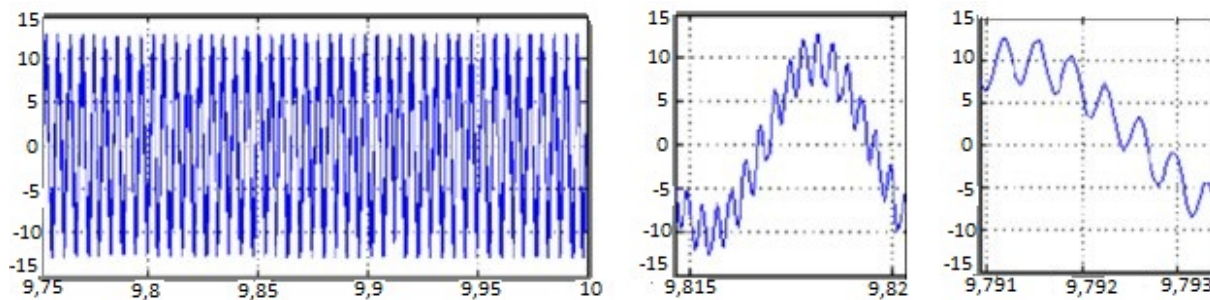


Рис. 4. Сигнал на выходе сумматора

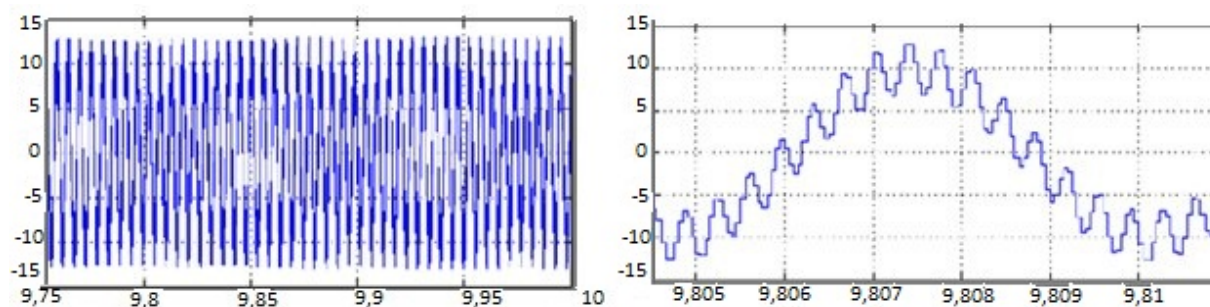


Рис. 5. Сигнал, поступающий на фильтр

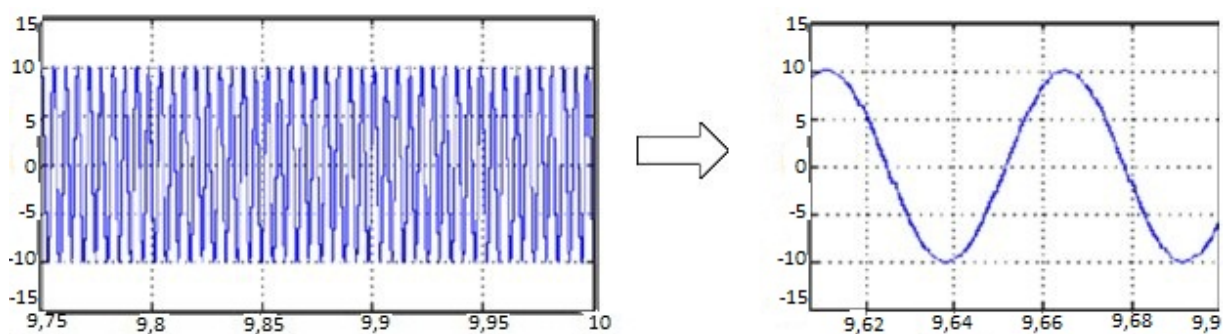


Рис. 6. Выходной сигнал КИХ-фильтра

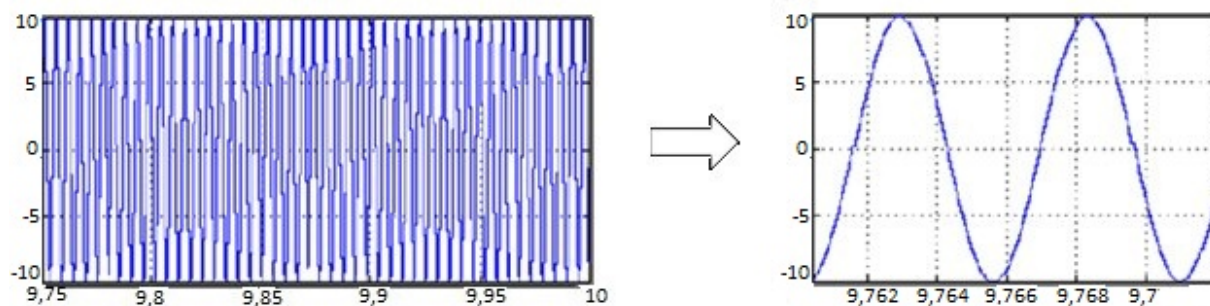


Рис. 7. Выходной сигнал БИХ-фильтра

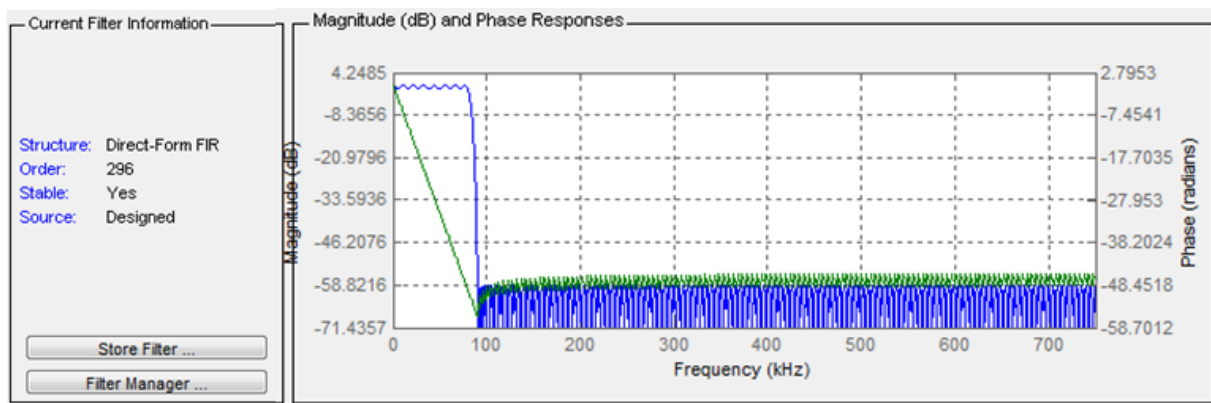


Рис. 8. АЧХ и ФЧХ КИХ-фильтра

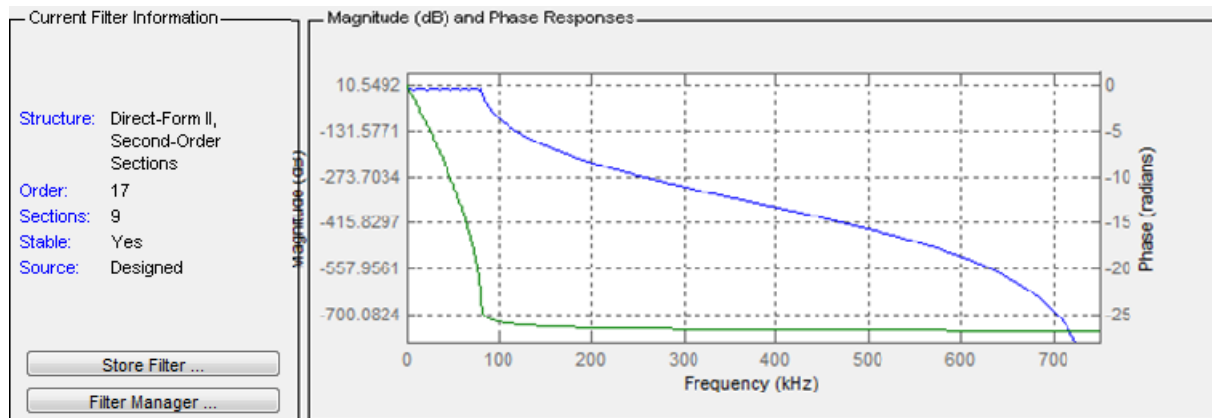


Рис. 9. АЧХ и ФЧХ БИХ-фильтра

Из последних графиков (рис. 8, 9) видно, что БИХ-фильтр более эффективен, так имеет меньший порядок реализации, из чего следует, что он требует меньшего количества памяти и меньшего количества операций умножения с накоплением.

С другой стороны, КИХ-фильтр требует большего количества звеньев и, соответственно, операций умножения с накоплением для реализации частотной характеристики с заданной частотой среза, но при этом имеют линейную фазовую характеристику.

В итоге можно сказать, что если необходимы фильтры с крутым спадом, и испытывается дефицит времени, отведенного для обработки, хорошим выбором являются БИХ-фильтры. Если число операций умножения с накоплением не является чрезмерным, и требуется линейная фаза, то должен быть выбран КИХ-фильтр. Фильтр с линейной ФЧХ задерживает все частоты на одно и то же время. Следовательно, выходной сигнал не искажается, а лишь задерживается на соответствующее время. Поскольку реальный сигнал содержит много частот, задержка каждой частоты на разное время приводит к искажению выходного сигнала. Линейная ФЧХ исключительно важна в полосе пропускания, поскольку тот диапазон, в котором происходит выделение интересующих нас частотных составляющих сигнала. Обычно имеет место компромисс между линейной ФЧХ в полосе пропускания и другими требованиями к фильтру, такими как крутизна спада и затухание в полосе задерживания.

### Библиографический список

1. Распопов В. Я. Микромеханические приборы. М.: Книга, 2007, 400 с.
2. Кулик А. С. и др. Обработка сигналов датчиков первичной информации бесплатформенной инерциальной навигационной системы цифровой фильтрацией // Радиоэлектронные компьютерные системы. 2010. № 3. С. 120–124.
3. Куприянов М. С., Матюшин Б. Д. Цифровая обработка сигналов.
4. Дьяконов В. П. Абраменкова И. В. MATLAB-обработка сигналов и изображений. СПб.: Книга, 2002. 608 с.



**Л. И. Курбанов**

студент кафедры аэрокосмических компьютерных и программных систем

**Н. А. Матвеева**

научный руководитель

### ПРОЕКТИРОВАНИЕ БУФЕРНОГО ПРОСТРАНСТВА ДЛЯ ВСТРОЕННЫХ СЕТЕЙ-НА-КРИСТАЛЛЕ И СИСТЕМ-В-КОРПУСЕ

В большинстве сетей-на-кристалле (СенК) в маршрутизаторах используются буфера (рис. 1). Они могут использоваться как при маршрутизации/коммутации с буферизацией, так и при маршрутизации «на лету». В последнем случае они используются для устранения возможности взаимоблокировок потоков данных, а также позволяют существенно снизить эффект от локальных перегрузок отдельных участков сети. Это происходит за счет того, что пакеты могут накапливаться в буферах маршрутизаторов, оказавшихся в зоне такой локальной перегрузки. Размер перегруженной зоны при этом не увеличивается или увеличивается гораздо медленнее, чем в системах без буферов. При этом, чем больше размер буферов, тем большая защита от распространения по сети перегрузки может быть обеспечена. Размеры буферов должны выбираться таким образом, чтобы среднее время передачи пакетов данных удовлетворяло пользовательским требованиям [1].

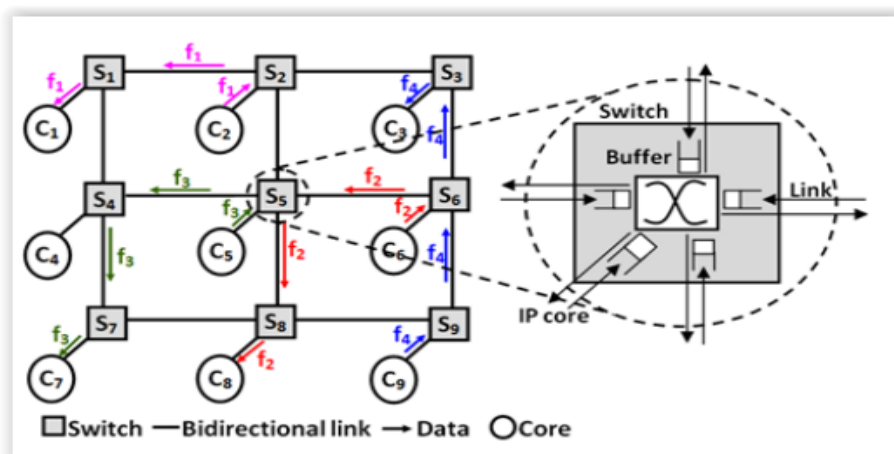


Рис. 1. Использование БП в СенК

( $C_i$  –  $i$ -й IP-блок (терминальный узел);  $S_j$  –  $j$ -й коммутатор (switch-узел);  $f_k$  –  $k$ -й поток)

Однако с увеличением размеров буферов очень существенно увеличиваются аппаратные затраты на реализацию маршрутизаторов и их энергопотребление. Эти параметры являются очень критичным при реализации современных СенК. Как правило, существуют достаточно жесткие ограничения на предельно допустимый размер буферов в каждом из маршрутизаторов, определяемые технологическими ограничениями, а также ограничения на суммарную площадь компонентов, входящих в состав коммуникационной системы. В соответствии с этими параметрами определяется ограничение на суммарное количество буферов в маршрутизаторах.

Определение размеров буферов является сложной, многокритериальной задачей, которая должна решаться в сочетании с задачей выбора топологии сети и распределения потоков данных в ней.

Для проектирования БП в СенК используется аналитическое и имитационное моделирование.

Аналитическое моделирование – это математический прием исследования логистических систем, позволяющий получать точные решения. Аналитическое моделирование осуществляется в следующей последовательности:

- формулировка математических законов, связывающих объекты системы. Эти законы записываются в виде некоторых функциональных соотношений (алгебраических, дифференциальных и т. п.);
- решение уравнений, получение теоретических результатов;
- сопоставление полученных теоретических результатов с практикой (проверка на адекватность).

Наиболее полное исследование процесса функционирования системы можно провести, если известны явные зависимости, связывающие искомые характеристики с начальными условиями, параметрами и переменными системы. Однако такие зависимости удается получить только для сравнительно простых систем.

*Достоинство* аналитических методов заключается в возможности получения решения в явной аналитической форме, позволяющей проводить детальный анализ процессов, протекающих в исследуемой системе, в широком диапазоне изменения параметров системы. Результаты в аналитической форме являются основой для выбора оптимальных вариантов структурно-функциональной организации системы на этапе синтеза.

*Недостаток* аналитических методов – использование целого ряда допущений и предположений в процессе построения математических моделей и невозможность, в некоторых случаях, получить решение в явном виде из-за неразрешимости уравнений в аналитической форме, отсутствия первообразных для подынтегральных функций и т.п.

Среди существующих аналитических методов проектирования БП в СенК применяются детерминированный, стохастический, физический и системный [2].

Детерминированный метод проектирования основан главным образом на теории графов. Модель, разработанную этим методом, можно представить в виде цикло-статического графа потоков данных. С помощью этой модели можно определить размер буферного пространства в узлах СенК.

*Достоинством* является достаточно точный результат анализа модели.

*Недостатком* – требование к глубокому знанию о структуре связей между IP блоками и коммутаторами.

Стохастический метод проектирования основан на теории массового обслуживания. Обычно для расчетов систем массового обслуживания используются простейшие потоки событий. Берется простейший поток, так как он создает наиболее тяжелый режим функционирования системы по сравнению с другими потоками и используется для определения предельных значений характеристик качества обслуживания. Если реальный поток отличен от простейшего, то система будет работать не хуже рассчитанной [3]. Таким образом, полученная модель будет рассчитана на наихудший случай.

*Достоинством* является хорошо исследованный математический аппарат.

*Недостатком* – отличие реальных потоков от моделируемого. Модель рассчитана на наихудший случай.

Физический метод проектирования основан на теории информации и теории статистической физики. Статистическая физика может моделировать взаимодействия между различными компонентами, учитывая долгосрочные эффекты памяти. Главной идеей этой модели является то, что пакеты передаются в сети из одного узла в другой как частицы, двигающиеся в бозе-газе и переходящие между различными энергетическими уровнями от изменения температуры. Другими словами, данный подход заключается в виртуальной случайно растущей сети, которая описывает эволюцию буферов СенК как функцию интенсивности входного потока. Ана-

лиз эффективности предложенной модели показывает, что заполнение буфера происходит по степенному закону распределения.

*Достоинством* является достаточно точный результат анализа модели.

*Недостатком* – большие временные затраты.

Системный метод проектирования – основан на теории систем, которая успешно применяется для разработки электронных схем. Сетевые калькуляторы с помощью этой теории моделируют и оценивают размеры буферов для каждого коммутатора. В зависимости от вида передачи потока, проектировщики способны наблюдать некоторые динамические характеристики сети. Также сетевой калькулятор может использоваться для анализа и оценки показателей производительности сети на кристалле. Тем не менее, данный метод рассчитан для худшего случая поведения сети.

*Достоинством* является способность наблюдения динамических характеристик сети.

*Недостатком* – то, что модель рассчитана на наихудший случай поведения сети.

На сегодняшний день существует достаточно много аналитических решений проектирования БП в СенК. В табл. 1 приведены достоинства и недостатки некоторых из них.

Таблица 1

Существующие аналитические решения для проектирования БП в СенК

Название	TDMA [4]	Flow Regulation [5]	D-buffer[6]	Bufferallocation [7]
Достоинства	Снижение потребляемой мощности и занимаемой площади БП	Минимизирует общее количество буферов	Устраняет задержку изображения	Увеличивает производительность сети
Недостатки	Только для управления потоком точка-точка с кредитованием	Низкая производительность сети	Требует больших временных ресурсов	Требует больших затрат на памяти на БП

Рассмотрим в качестве примера решение «BufferAllocation» [7]. Его можно отнести к стохастическому методу, так как использует теорию массового обслуживания для распределения БП в СенК. Этот метод ориентирован на сети с топологией типа «решетка», XY-маршрутизацией и коммутацией «на лету». Его главная идея заключается в том, чтобы выявить «слабые»/«узкие» места в сети, где вероятность перегрузки узла наибольшая, и расширить в этом узле БП из предоставляемого проектировщику ресурса памяти.

Этот метод значительно увеличивает производительность сети и уменьшает вероятности перегрузок, однако требует больших затрат памяти на БП. Этот недостаток может привести к увеличению площади СенК.

Помимо аналитического моделирования используется и имитационное моделирование для проектирования БП в СенК.

Имитационное моделирование – метод исследования объекта (сети), основанный на том, что изучаемый объект заменяется имитирующим объектом. С имитирующим объектом проводят эксперименты (не прибегая к экспериментам на реальном объекте) и в результате получают информацию об изучаемом объекте.

Имитационное моделирование применяется в случае, если:

- идет процесс познания объекта моделирования;
- аналитические методы исследования имеются, но составляющие их математические процедуры очень сложны и трудоемки;
- необходимо осуществить наблюдение за поведением компонент системы в течение определенного времени;

- необходимо контролировать протекание процессов в системе путем замедления или ускорения явлений в ходе имитации;
- особое значение имеет последовательность событий в проектируемых системах и модель используется для предсказания так называемых «узких» мест;
- при подготовке специалистов для приобретения необходимых навыков в эксплуатации новой техники;
- имитационное моделирование оказывается единственным способом исследований из-за невозможности проведения реальных экспериментов.

Имитационное моделирование состоит из следующих этапов:

- создание модели (программирование работы сети; учитываются статическая и динамическая составляющие сети);
- проведение экспериментов (запуск программы с различными входными данными)
- обработка полученных данных.

На основе полученных данных можно вычислить зависимость потребляемой энергии, скорости передачи данных от размера БП.

В случае имитационного моделирования существуют уже готовые инструменты для создания таких моделей (табл. 2).

Таблица 2

Инструменты для имитационного моделирования БП в СенК

Название	Фреймворк	Параллелизм	Топология	Доступность
SICOSYS	C++	–	Limited	+
Noxim	SystemC	–	Mesh	+
NNSE	SystemC	–	Mesh/Torus	+
Nirgam	SystemC	–	All	+
gpNoCsim	Java	–	All	+
DARSIM	C++	+	All	+
OMNeT++	C++/NED	+	All	+

Как видно из табл. 2, инструменты для имитационного моделирования отличаются по:

- используемому программному языку;
- поддержке параллелизма;
- способности моделирования определенных топологий сетей.

Из рассмотренных инструментов можно выделить DARSIM и OMNET++ как обладающих самым широким функционалом из бесплатных продуктов.

В качестве выводов можно сказать, что при проектировании БП для СенК с простыми топологиями и небольшим количеством портов в коммутаторах, можно применить аналитический метод BufferAllocation. Недостаток этого метода (требует больших затрат по объему памяти для БП) можно убрать введением понятия допустимой вероятности перегрузки (ДВП) порта в узле. ДВП задается проектировщиком, и если максимальная вероятность перегрузки порта в узле будет меньше либо равна ДВП, распределение БП остановится. Это позволит сэкономить затраты памяти для БП. При проектировании БП для СенК со сложной структурой передачи данных и большим количеством портов в коммутаторах следует выбрать один из инструментов имитационного моделирования, опираясь при выборе на поддерживаемый ими функционал и язык программирования.

**Библиографический список**

1. Kurbanov L. "The Network Calculator for NoC Buffer Space Evaluation Nadezhda Matveeva", Lev Kurbanov, Elena Suvorova, Yuriy Sheynin Saint-Petersburg State University of Aerospace Instrumentation Saint-Petersburg, Russian Federation.
2. Bakhouya M. "A Buffer-space Allocation Approach for Application-specific Network-on-Chip" / M. Bakhouya, A. Chariete, J. Gaber, M. WackUniversite de Technologie de Belfort-Montbéliard 90010 Belfort, France.
3. Майорова, С. А. Основы теории вычислительных систем. М.:Высшаяшкола, 1978. 408 с.
4. Coenen M. "A buffer-sizing algorithm for networks on chip using TDMA and credit-based end-to-end flow control" / M. Coenen, Eindhoven Philips Res., K. Goossens, G. De Micheli, S. Murali, in Proceedings of the 4th International Conference Hardware/Software Codesign and System Synthesis, Oct. 2006, pp. 130–135.
5. FahimehJafari, "Buffer Optimization in Network-on-Chip Through Flow Regulation"/ FahimehJafari, Zhonghai Lu, Axel Jantsch, Mohammad Hossein Yaghmaee, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, Dec. 2010,pp. 1973–1986.
6. Leonel Tedesco, "Buffer Sizing for QoS Flows in Wormhole Packet Switching NoCs" / Leonel Tedesco, Ney Calazans, Fernando Moraes, Proceedings of the 20th Annual Symposium on Integrated Circuits and Systems Design, 2007.
7. Wang L. "Application specific buffer allocation for wormhole routing networks-on-chip"/ L. Wang, Y. Cao, X. Li, and X. Zhu, NoCarc08, MICRO-41, 2008.

УДК 004.7

**В. И. Миненко**

студент кафедры аэрокосмических компьютерных и программных систем

**Е. А. Суворова**

канд. техн. наук, доц. – научный руководитель

**ОБЕСПЕЧЕНИЕ ОТКАЗОУСТОЙЧИВОСТИ  
БОРТОВЫХ ИНФОКОММУНИКАЦИОННЫХ СЕТЕЙ**

В работе рассматриваются актуальность и методы повышения надежности бортовых инфокоммуникационных сетей. Надежность как обеспечение сетью требуемых характеристик сервиса (гарантированная пропускная способность, время доставки). Понятие структурной избыточности.

Важной особенностью современных бортовых информационно-измерительных комплексов (авионики) является постоянная модернизация существующего оборудования. При этом количество функций выполняемых электроникой возрастает, что приводит увеличению количества обрабатываемых данных, поступающих от различных датчиков. Это в свою очередь приводит к необходимости увеличения пропускной способности и надежности бортовых вычислительных сетей.

Обеспечение отказоустойчивости бортовых инфокоммуникационных сетей во время полета, обуславливает необходимость разработки комплексного подхода к оценке количественных показателей их безотказности. В связи с этим предметом исследования является обеспечение отказоустойчивости бортовых инфокоммуникационных сетей.

**Понятие отказа системы**

Под отказом системы принято понимать событие, заключающееся в нарушении ее работоспособного состояния. В авионике возможны два типа отказов: структурный и параметрический. Под структурным отказом системы [1] понимается отказ, связанный с неработоспособностью ее компонента и, как следствие, приводящий к нарушению взаимодействия этого компонента с другими компонентами авионики. Под параметрическим отказом понимается отказ, связанный с отклонением рабочего параметра одного из компонентов за пределы допуска. Согласно авиационным правилам параметрический отказ любого компонента авионики должен быть выявлен с помощью встроенных диагностических средств, в результате чего такой отказавший компонент должен быть исключен из рабочего процесса. При этом независимо от типа отказа система вы-

ключается из общей работы авионики. Поэтому оба типа отказов имеют равноценные последствия для изменения структуры авионики, и их следует рассматривать в целом.

Отказы системы могут происходить как из-за влияния внешних факторов, так и по естественным причинам, связанным с ее структурой и конструкцией составляющих. Основными причинами сбоев и отказов в рассматриваемых системах являются помехи создаваемые внутренними электромагнитными полями, воздействие внешних факторов и перепады нагрузки в сети. При эксплуатации самолета возникновение отказов компонентов авионики не должно влиять на процесс выполнения авионикой требуемых функций. Этот вопрос на практике решается с помощью специальных методов и способов, обеспечивающих гарантированную отказоустойчивость сети, например обеспечение гарантированной пропускной способности (минимальная пропускная способность сети), гарантированное время доставки пакета.

#### Методы повешения надежности

Методы повышения надежности бортовых систем, связанные с понятиями надежности, отказоустойчивости и живучести аппаратуры, можно разделить на три группы: 1) конструктивно-технологические методы, направленные на улучшение качества отдельных элементов и облегчение режимов работы систем [2]; 2) методы технической диагностики, связанные с применением аппаратно-программных средств автоматического или полуавтоматического контроля исправности аппаратуры и программного обеспечения в процессе эксплуатации системы; 3) структурные методы, с помощью которых может быть выбрана наиболее рациональная структура системы и ее компонентов, по возможности ослабляющая влияние отказов отдельных элементов на погрешности и качество ее работы. Как правило, все эти методы не исключают, а взаимно дополняют друг друга, что ведет к созданию наиболее технически эффективных систем.

Для достижения цели настоящего исследования рассмотрим существующие структурные методы, более подробно.

Современные системы связи используют большое количество элементов, что обуславливает необходимость использования резервирования и обходных маршрутов для повышения коэффициента готовности системы связи в целом.

1. Методы, основанные на оптимизации структурной схемы без применения функциональной избыточности. Здесь изменение в широких пределах входных параметров или данных наиболее нестабильных модулей или элементов не должно влиять или должно мало влиять на стабильность решения поставленных задач. При этом проводится оптимизация параметров систем.

Оптимизация параметров системы позволяет существенно снизить вероятность параметрических отказов. Однако при возникновении структурного отказа – полной утрате работоспособности отказавшего элемента – восстановление работоспособности системы без применения функциональной избыточности невозможно.

2. Методы, основанные на введении функциональной избыточности. При этом основными способами являются резервирование и комплексирование по мультимодальному принципу.

Резервирование может осуществляться на различных уровнях. Могут резервироваться либо отдельные элементы системы, либо блоки, входящие в систему – раздельное резервирование, либо вся система в целом – общее резервирование. Чем большая часть системы резервируется как единое целое, тем крупнее масштаб резервирования.

Если в системе имеются группы одинаковых элементов, то в этом случае резервные элементы могут объединяться, создавая тем самым скользящий резерв, применяемый для совокупности одинаковых элементов.

Комплексирование по мультимодальному принципу заключается в том, что одна и та же задача решается с помощью нескольких средств или процессов, основанных на различных методах, алгоритмах и средствах программирования и реализуемых на универсальном или специализированном вычислительном устройстве или устройствах, обеспечивающих достижение поставленной цели.

По способу реализации методы повышения надежности аппаратуры систем подразделяются на активные и пассивные. При этом применение активной отказоустойчивости характеризуется более экономным расходом аппаратных средств, чем применение пассивной отказоустойчивости. Однако оно связано с некоторыми потерями времени при восстановлении работы системы после отказа, а также, возможно, потерями некоторой части данных. Активная отказоустойчивость вычислителя системы реализуема только в многопроцессорных системах.

Методы повышения надежности на примере сети AFDX

Рассмотрим реализацию данных методов подробнее на примере существующей сети AFDX.

AFDX (Avionics Full Duplex Switched Ethernet) [3] – авиационная сеть передачи данных базирующаяся на основе 10/100MbitEthernet. AFDX использует специальный протокол для обеспечения безопасной и надежной передачи данных, с гарантированным временем их передачи.

Основной целью стандарта AFDX является создание детерминистской сети передачи данных, которая может быть использована для использования необходимыми для управления полетом системами. Эта цель достигается при помощи предоставления выделенных полос пропускания трафика для каждого маршрута информации в сети и обеспечения доступности спецификации качества обслуживания (QoS) на каждом узле системы.

Сеть AFDX состоит из самолетных компьютеров и коммутаторов. В этой сети самолетные компьютеры являются потребителями сервисов передачи в сети. Уровень Конечных Систем (КС) на содержащем его компьютере предоставляет этот сервис. Конечные Системы позволяют приложениям, исполняющимся на компьютере, посылать и получать данные достоверным и безопасным способом.

КС коммутируются на основе виртуальных каналов с приоритезацией трафика (Traffic Shaping), с помощью назначения пропускной способности каждого виртуального канала. Непревышение пропускной способности виртуальных каналов достигается благодаря согласованным действиям отправителей и коммутаторов. Цель функции формирования трафика заключается в ограничении мгновенной скорости кадров в виртуальных каналах путем введения интервалов между кадрами.

КС связываются друг с другом при помощи коммутирующих пакеты компьютеров, называемых «коммутаторами AFDX». Основной задачей коммутаторов является обмен кадрами между КС, подключенными к их портам. В дополнение к этому коммутаторы реализуют правила контроля трафика на заранее сконфигурированных виртуальных каналах, статически определенных пользователем. Виртуальный Канал (ВК) это связь, используемая КС для передачи данных через сеть. ВК представляет собой однонаправленный «один ко многим» коммуникационный маршрут.

Гарантированное обслуживание обеспечивает четкую, математически обоснованную задержку, ограниченную сквозной передачей кадра. В результате этого, для того, чтобы гарантировать ограниченную задержку необходимо гарантировать определенную пропускную способность на уровне канала передачи.

Поэтому гарантированное обслуживание обеспечивает и ограниченную по верхнему пределу задержку и постоянную пропускную способность, что ведет к логически открытому соединению между одним передающим узлом и одним или несколькими принимающими узлами. Кадры, относящиеся к одному соединению, образуют поток.

Основные концепции сети AFDX

### 1. Избыточность

Это свойство достигается путем использования двух независимых маршрутов между каждой конечной системой. Для каждого ВК передающая конечная система посылает по копии данных в обе сети. Принимающая КС принимает первый прибывший экземпляр данных и отбрасывает вторую копию. При диагностировании ошибки (например, несовпадение контрольной суммы) в одной сети данные берутся из другой сети.

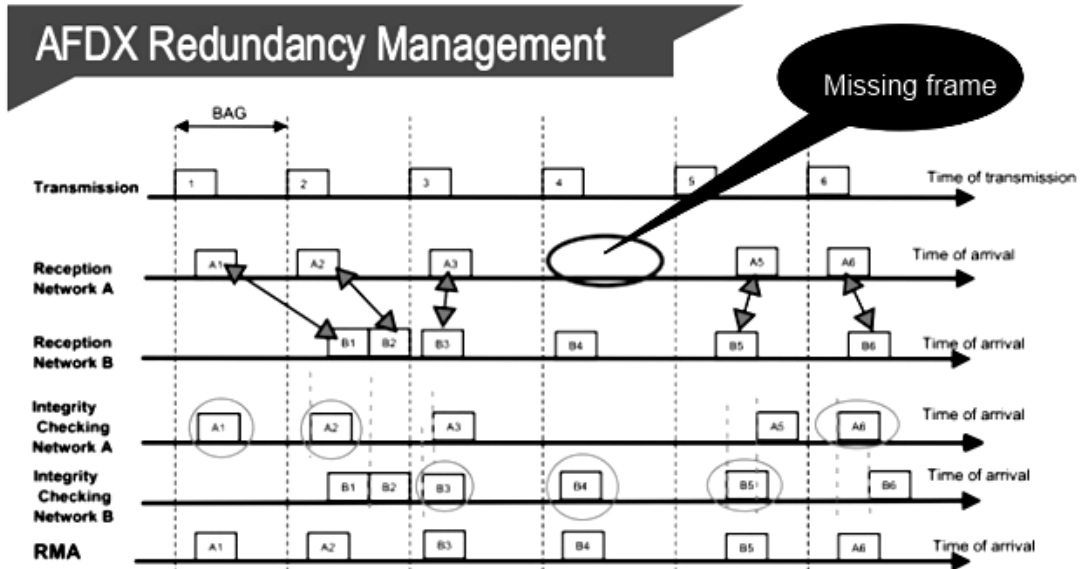


Рис. 1. Пример потери пакета в сети AFDX

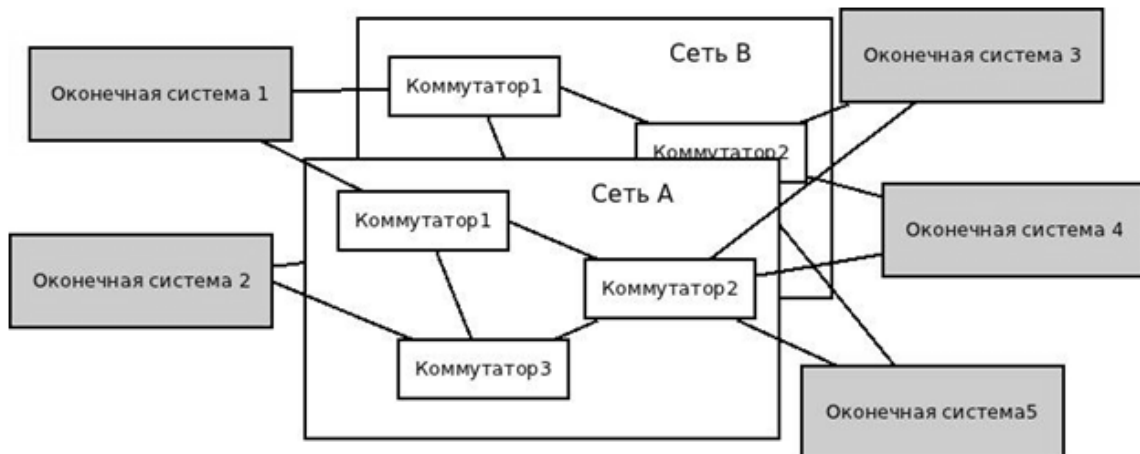


Рис. 2. Пример передачи данных

## 2. Подгонка трафика

Обработка ВК осуществляется посредством механизма управления потоком, который регулирует потоки данных из разных источников, относящихся к данной конечной системе, причем этот механизм обеспечивает разделение на уровне сети.

Каждый ВК имеет выделенную максимальную полосу пропускания. Эта полоса определяет системный интегратор при помощи нескольких конфигурируемых параметров.

На выходе конечной системы поток кадров, относящийся к конкретному ВК, характеризуется тремя параметрами: интервалом в распределенной пропускной способности (BAG) и житемом [4] и максимальным размером кадра.

BAG – BandwidthAllocationGap – минимальный интервал времени между началами выдачи последовательных кадров на одном ВК (1–128 мс, является степенью двойки);

$L_{max}$  – максимальный размер кадра ( $\leq 1518$ байт);

$J_{max}$  – максимально допустимое отклонение между кадрами от BAG.



Размер полосы пропускания, выделенной для ВК, является функцией значения BAG в герцах и  $L_{max}$  в байтах. Например, в простой сети, представленной на рис. 1, максимальная полоса пропускания, доступная ВК номер 1, вычисляется как  $(1000/2)*420 = 210$  Кбайт/с.

Если кадры не испытывают дрожания от планировщика, BAG представляет минимальный интервал времени между первыми битами двух последовательных кадров из одного ВК.

В передающей оконечной системе с несколькими VL планировщик мультиплексирует различные потоки, исходящие от регуляторов, как показано на рис. 2.

### Выводы

Помимо таких параметров, как потери пакетов, задержка пакетов, вариация задержки, составляющими качества бортовых вычислительных сетей являются параметры надежности и отказоустойчивости. Каждый запрос о передаче через сеть должен быть обслужен независимо от типа данных, и максимальная задержка передачи по сети должна быть гарантирована. Поэтому, единственным классом обслуживания в бортовой сети является гарантированное обслуживание.

### Библиографический список

1. Пупков А. Н. Модель анализа надежности распределенных вычислительных систем / Царев Р. Ю., Пупков А. Н. Вестник СибГАУ № 1 (47) 2013. – 86–90 с.
2. Конструкторско-технологическое проектирование электронной аппаратуры: учебник для вузов. – М.: Изд. МГТУ им. Н. Э. Баумана, 2002. – 528 с. URL
3. [http://www.afdx.com/pdf/AFDX\\_Training\\_October\\_2010\\_Full.pdf](http://www.afdx.com/pdf/AFDX_Training_October_2010_Full.pdf)
4. Ion Stoica, Providing guaranteed services without per flow management / Ion Stoica, Hui Zhang

УДК 004.7

### **М. В. Паламарчук**

студент кафедры аэрокосмических компьютерных и программных систем

### **Е. А. Суворова**

канд. техн. наук, доц. – научный руководитель

## ОБЗОР ОПРЕДЕЛЕНИЙ ДЖИТТЕРА ПРИ ПЕРЕДАЧЕ АУДИОИНФОРМАЦИИ И МЕТОДОВ БОРЬБЫ С НИМ

### Понятие качества обслуживания

В современном мире идет быстрое усиление влияния сетевых технологий на повседневную жизнь. Это проявляется в обилии телекоммуникационных средств – телефонов, планшетов, нетбуков и других, и, как следствие, их использование для коммуникаций людей с помощью этих новых технических средств. Как раз для обеспечения хорошего качества связи, которая не вызвала бы нареканий у пользователя, нужно рассмотреть джиттер – нежелательные задержки и колебания при передаче данных.

Начиная разговор о джиттере, стоит упомянуть о том, откуда появилось это определение, а именно вспомнить о Quality of Service (QoS), частью чего он является. Quality of Service – это качество обслуживания, оно имеет множество определений, но приведем лишь одно из них, высказанное компанией Cisco: QoS – способность сети обеспечить необходимый сервис заданному трафику в определенных технологических рамках. Данный необходимый сервис описывается многими параметрами, но среди них можно выделить некоторые: 1) Bandwidth (BW) – описывает номинальную пропускную способность среды передачи информации, определяет ширину канала, 2) Delay – задержка при передаче пакета, 3) Jitter – колебание задержки при передаче пакета, 4) Packet Loss – пределяет количество пакетов, отбрасываемых сетью во время передачи.

Понятие джиттера

Джиттер (jitter) – в переводе с английского – колебание или дрожание.

Можно выделить несколько вариантов определения для этого понятия. Оно используется в различных технических сферах: от соединений линий связи на электрических схемах до высокоскоростной передачи данных.

– Джиттер – нежелательные фазовые или частотные случайные отклонения передаваемого сигнала, которые возникают из-за нестабильности, изменений параметров линии передачи во времени и различной скорости распространения частотных составляющих одного и того же сигнала. Определение джиттера используется в цифровых системах. Он проявляется в виде случайных быстрых изменений передачи цифрового сигнала во времени, что приводит к рассинхронизации и искажению передаваемой информации [1].

– Джиттер – это разница между максимальным и минимальным значением задержки передачи данных. Вычисляется по формуле:

$$J = d_{max} - d_{min},$$

где  $J$  – джиттер,  $d_{max}$  – максимальное значение задержки,  $d_{min}$  – минимальное значение задержки [2].

– Джиттер – это среднее отклонение каждой отдельной задержки от среднего значения задержки. Вычисляется по формуле:

$$J = \sqrt{\frac{\sum (d_i - D)^2}{N - 1}},$$

где  $D$  – это среднее значение задержки,  $d_i$  – значение  $i$ -й задержки,  $N$  – количество задержек. Среднее значение задержки вычисляется по формуле:

$$D = \sum \frac{d_i}{N}.$$

Значение задержки и среднее значение задержки, джиттер могут измеряться в секундах или ее производных единицах измерений (мс, мкс, нс и т. д.) [3].

– Jitter (джиттер или дребезг) – это неравномерность периодов времени, отведенных на доставку пакета. Джиттер присущ только сетям с пакетной коммутацией. Задержка в доставке пакета и джиттер обычно исчисляются в миллисекундах, и на первый взгляд, не являются столь критичными и едва ли могут быть замечены человеком. Это так, до тех пор, пока это не касается передачи голоса или видео, другими словами, до тех пор, пока мы не столкнемся с передачей контента в реальном режиме времени [4].

Проблемы возникновения джиттера

Тестирование джиттера, его измерение, поддержка и минимизация – важная задача, так как хочется увеличения скорости работ в цифровых сетях для повышения производительности устройств. При увеличении временной частоты уменьшается окно пропуска данных и, как следствие, налагаются жесткие ограничения на размер джиттера.

Данный вопрос очень актуален для систем реального времени, например в телефонии, IP-сетях, VoIP-технологиях и других. Например, задержка передачи речевого сигнала оказывает психологическое воздействие на пользователей, которые удивительно точно чувствуют ритм разговора и, не услышав ответа собеседника в течение определенного времени, повторяют вопрос. При телефонном разговоре не существует визуального контакта, который позволил бы одному из его участников увидеть, что задержка с ответом вызвана, например, тем, что другой абонент задумался. При общении по телефону абонент, задавший вопрос, полагается на внутреннее восприятие разговора и, не получив своевременного ответа, повторяет вопрос или уточняет, слышит ли его другая сторона. Если канал передачи приводит к чрезмерной задержке сигнала, повтор-

ный или уточняющий вопрос одного из участников разговора совпадает с ответом другого абонента, и возникает путаница. Так было выяснено, что задержка, равная 70–100 мс остается незамеченной, а вот, если ее размер превышает 150 мс, то уже многие абоненты начинают жаловаться на задержку [5].

Существует ряд способов, предназначенных для борьбы с джиттером. Рассмотрим некоторые из них, допустимые при передаче аудио информации:

- джиттер буфера;
- виртуальные часы;
- Jitter Virtual Clock.

Способ решения проблемы: джиттер буфера.

В сети пакеты могут приходиться с разной скоростью и в различном порядке, что приводит к появлению джиттера. Наиболее простой метод избавиться от него – это метод джиттер буфера. Он основывается на том, что используется определенный размер буфера для получения пакетов и передачи его приемнику с наименьшей задержкой. В буфере по мере поступления пакетов из сети, восстанавливается их порядок следования. Если некоторые пакеты не находятся в буфере (он потерян или еще не пришел) тогда их необходимо проигнорировать. Иллюстрация работы метода показана на рис. 1 [6].

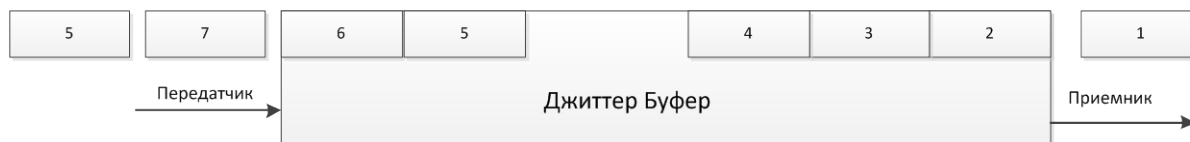


Рис. 1. Джиттер буфера

Способ решения проблемы: виртуальные часы

Дисциплина обслуживания, которая повторяет технологию Time division multiplexing – мультиплексирование с разделением по времени. Отличительной особенностью этого метода является то, что для каждого пакета выделяется виртуальное время передачи, в которое он должен быть отправлен, так же создается две переменные, использование которых показано далее. Одна из них – VirtualClock (VC) и вторая, являющаяся дополнительной, переменная VirtualClock (auxVC).

Каждый коммутатор сети выполняет две основные функции: передача данных и наблюдение за потоком данных. Функция передачи данных включает в себя следующие операции:

- после получения первого пакета потока данных в переменные VC и auxVC заносится значение счетчика реального времени (обозначим RealTime);
- после принятия каждого последующего пакета в переменную заносится значение  $\max(\text{RealTime}, \text{auxVC})$ . В переменную VC заносится значение  $VC + V_{\text{tick}}$ , значение auxVC так же возрастает на величину  $V_{\text{tick}}$ , где  $V_{\text{tick}} = \frac{1}{AR}$ . Пакету назначается метка со значением равным значению переменной auxVC;
- пакет передается в очередь на отправку. Из очереди пакеты передаются в порядке возрастания значения метки.

Для мониторинга потока данных выполняется проверка значений переменных VC и auxVC. Допустим, если значение VC не превосходит значения текущего счетчика времени RealTime, то переменной VC присваивается значение RealTime [6].

Способ решения проблемы: Jitter Virtual Clock

Этот алгоритм – это версия алгоритма Virtual Clock, которая представляет комбинацию механизмов Delay-jitter rate-controller и Virtual Clock. Он работает следующим образом: каждому

пакету в момент его прибытия назначается планируемое время отправки (aligible time) и дедлайн. Пакет удерживается контроллером передачи данных до тех пор, пока системное время не превысит значение планируемого времени отправки, назначенное пакету. После этого планировщик отправляет пакеты в порядке их дедлайнов. Для пакета с номером  $k$  потока  $i$  планируемое время отправки обозначается  $e_{i,j}^k$ , дедлайн  $d_{i,j}^k$ . В узле с номером  $j$  эти параметры вычисляются следующим образом:

$$e_{i,j}^1 = a_{i,j}^1$$

$$e_{i,j}^k = \max(a_{i,j}^k + g_{i,j-1}^k, d_{i,j}^{k-1}), \quad i, j \geq 1, k > 1$$

$$d_{i,j}^k = e_{i,j}^k + \frac{l_i^k}{r_i}, \quad i, j, k > 1$$

где  $l_i^k$  – длина пакета,  $r_i$  – зарезервированная скорость для потока,  $a_{i,j}^k$  – время прибытия пакета в узел  $j$ ,  $g_{i,j}^k$  – метка, хранящаяся в заголовке пакета, которая устанавливается предыдущим узлом. Ее значение равно разнице между дедлайном пакета и его фактическим временем отправления из узла  $j - 1$  [7].

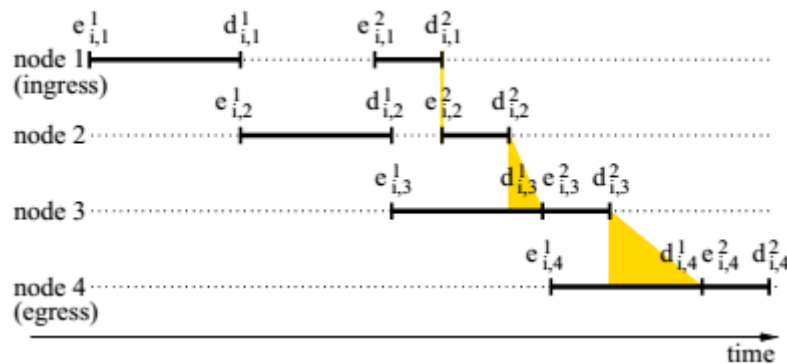


Рис. 2. Jitter Virtual Clock

Выводы

Приведем сравнение этих методов (таблица).

Таблица

Сравнение методов

Джиттер буфера	Virtual Clock	Jitter Virtual Clock
Использование буфера фиксированного размера. Может возникать переполнение буфера, в следствие чего могут появляться шумы и потеря информации	Для каждого пакета выделяется свое время передачи. Производится наблюдение и управление потоком данных (ограничения задаются средней скоростью и средним интервалом, задающимися спецификацией)	Каждому пакету назначается планируемое время отправки и дедлайн. С помощью этого алгоритма получается компенсировать вариацию задержки, зависящую от загрузки предыдущего узла. Повышается сложность работы планировщика

В итоге можно сказать, что методов минимизации джиттера достаточно много, например: Fair Queueing, Core-Jitter Virtual Clock (CJVC), Jitter-EDD, Delay Earliest-Due-Date, Stop-and-Go, Hierarchical Round Robin и другие. Однозначно сказать, какой из методов лучше нельзя, так как каждый из них может быть применим к определенной задаче.

В данной статье рассматриваются способы поддержки передачи данных в сети, рассмотрен джиттер и несколько его способов минимизации.

### Библиографический список

1. IP Packet Delay Variation Metric for IP Performance Metrics (IPPM), 2002.
2. Real-Time systems. Design Principles of Distributed Embedded Applications. 2011.
3. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. СПб.: Питер, 2006.
4. <http://it.aleksandr.ru/asterisk-jitterbuffer.html>
5. Вопросы качественной передачи голоса Финнеран часть 2 / Михаил Финнеран.
6. Lixia Zhang. VirtualClock: A New Traffic Control Algorithm for Packet-Switched Networks.
7. Ion Stoica, Providing guaranteed services without per flow management / Ion Stoica, Hui Zhang.

УДК 62-83(075.8)

**С. М. Пахарев**

студент кафедры аэрокосмических компьютерных технологий

**Г. С. Бритов**

канд. техн. наук – научный руководитель

### ДИНАМИЧЕСКИЕ ХАРАКТЕРИСТИКИ СЕТЕЦЕНТРИЧЕСКИХ СИСТЕМ

В настоящее время достаточно актуальной проблемой является разработка и проектирование сетевых систем (СЦС). «Сетевизм» стал результатом прорыва в информационных технологиях в конце прошлого века, что позволило организовать взаимодействие между компьютерами, несмотря на различные операционные системы [1]. Согласно этому принципу, управление, а также каждая единица техники и пользователи будут объединены в единую информационную сеть. При этом все «узлы» такой сети смогут получать и отправлять необходимую информацию даже находясь на большом расстоянии друг от друга, а также если один или несколько узлов такой сети вышли из строя. Важными аспектами при построении сети является то, что в сетевых системах высокий трафик передаваемой информации, а также строгая иерархия, что требует больших затрат на проектирование и тестирование таких систем.

Целью выполненной работы было исследовать динамические характеристики СЦС на основе функций математического пакета MatLab.

#### Исходные данные

Будем рассматривать СЦС, которые включают каналы связи, объединяющие в систему устройства обработки информации. Для выполнения исследования СЦС необходимы следующие исходные данные:

- вид объединения устройств обработки информации в СЦС;
- число устройств обработки информации;
- математическая модель обработки информации;
- требуемые динамические характеристики СЦС.

По виду объединения устройств обработки информации в сеть можно рассматривать два варианта:

- однородные сетевые структуры (ОСС);
- центрические сетевые структуры (ЦСС).

Пример ОСС, содержащей четыре устройства обработки информации УОИ 1 – УОИ 4, соединенных восемью каналами связи, с учетом входного и выходного каналов, приведен на рис. 1. Из приведенной схемы следует, что ОСС состоит из двух сетевых единиц (СЕ) УОИ 1, УОИ 2

и УОИ 3, УОИ 4. Они имеют характерный вид прямых и обратных каналов, входы и выходы. Из таких СЕ можно построить достаточно сложную однородную сеть.

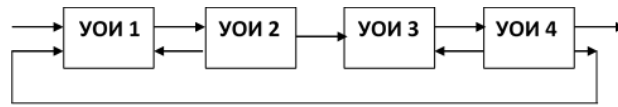


Рис. 1. Схема ОСС

Схема ЦСС, содержащей четыре устройства обработки информации УОИ 1 – УОИ 4, соединенных четырьмя каналами связи, приведена на рис. 2.

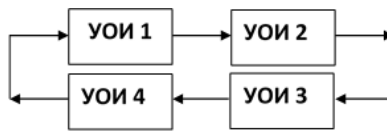


Рис. 2. Схема ЦСС

Из рис. 2 следует, что все устройства обработки информации объединены каналами связи в круговую, сетцентрическую структуру. Это упрощает исследование такой регулярной структуры. Однако она может быть усложнена хордовыми каналами связи.

Динамика устройства обработки информации опирается на следующие положения:

- каждое устройство имеет вход и выход;
- входами и выходами служат потоки данных, объемы которых могут быть измерены;
- обработка информации осуществляется в дискретном времени;
- требуется получить динамические характеристики потоков данных.

Динамические характеристики:

- устойчивость СЦС как линейной дискретной динамической системы;
- управляемость и наблюдаемость СЦС;
- весовая характеристика и переходная характеристика;
- реакция на произвольное входное воздействие.

Задачей исследования СЦС является построение передаточной функции системы или ее уравнений динамики. С помощью таких математических моделей можно получить динамические характеристики системы. Вместе с тем, можно выполнить функциональное и тестовое диагностирование системы по модели [2].

#### Принципы исследования СЦС

В основе исследования динамических характеристик ОСС были положены следующие принципы:

- математическая модель системы представляет собой вложение сетевых единиц;
- вход и выход ОСС доступны для измерения;
- динамические параметры СЕ заданы.

В результате исследования должна быть получена полностью описанная система, состоящая из  $n$  вложенных СЕ. Анализ такой системы начинается с одной СЕ и заканчивается всей системой. Таким образом, получается, что вычисления осуществляются циклическим образом.

Это является очень важным моментом, поскольку, исходя из циклического принципа, можно построить циклический алгоритм анализа ОСС.

На рис. 3 показана схема математической модели СЕ.

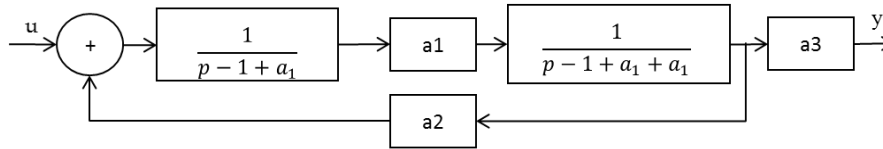


Рис. 3. Схема математической модели СЕ

Условные обозначения, принятые на рис. 3:  $u$ ,  $y$  – вход и выход СЕ;  $a_1$ ,  $a_2$ ,  $a_3$  – коэффициенты передачи;  $p$  – оператор временного сдвига.

Алгоритм исследования ОСС имеет  $n$  шагов:

1-ый шаг. Расчет передаточной функции 1-ой СЕ:

$$W_{CE}(p) = \frac{a_1 \cdot a_3}{(p-1+a_1) \cdot (p-1+a_2+a_3) - a_1 \cdot a_3}.$$

2-ой шаг. Расчет передаточной функции системы из 2-х СЕ:

$$W_2(p) = \frac{W_{CE}(p) \cdot W_{CE}(p) \cdot a_1 \cdot a_3}{1 - a_1 \cdot a_2 \cdot W_{CE}(p) \cdot W_{CE}(p)}.$$

$n$ -ый шаг. Расчет передаточной функции системы из  $n$  СЕ:

$$W_n(p) = \frac{W_{n-1}(p) \cdot W_{CE}(p) \cdot a_1 \cdot a_3}{1 - a_1 \cdot a_2 \cdot W_{n-1}(p) \cdot W_{CE}(p)}.$$

Данный алгоритм был реализован с помощью математического пакета Matlab для получения быстрого и точного результата проектирования ОСС. Для работы был организован визуально понятный интерфейс для взаимодействия с пользователем. Кроме того, учитывая особое значение вектора коэффициентов передачи целесообразно осуществить интерактивную процедуру выбора его по критерию, например, величины максимального стационарного коэффициента передачи системы.

В результате исследования должна быть получена полностью описанная система, состоящая из  $n$  УОИ, выходное значение системы может быть измерено. Анализ такой системы начинается с построения матрицы уравнений динамики.

Результаты исследования систем с четырьмя устройствами обработки информации

В результате исследования пользователь получает сведения о наблюдаемости и управляемости системы. Кроме того, будут построены переходная и весовая характеристики системы (рис. 4), реакция ее на случайной вход (рис. 5).

На рис. 6 показано собственное движение данных от начальных значений до стационарного режима в СЦС.

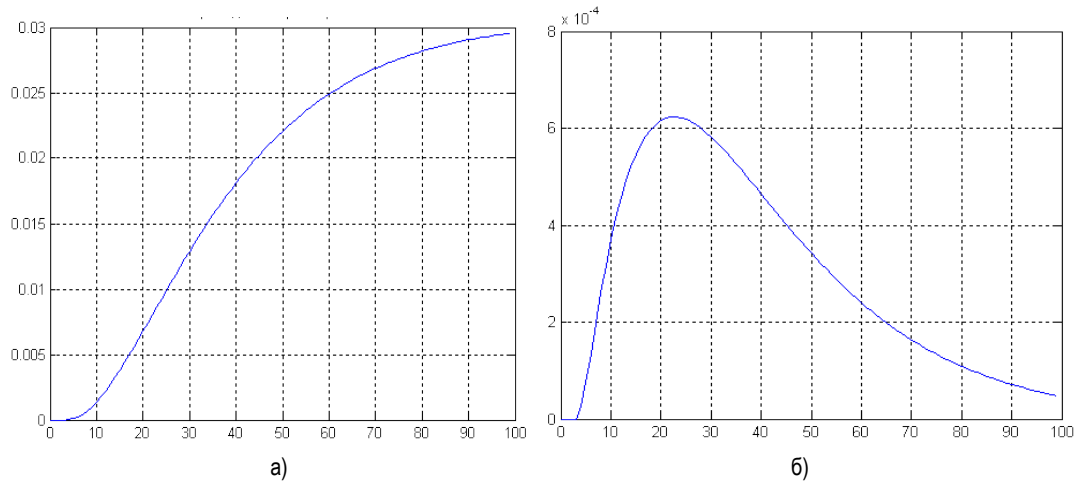


Рис. 4. Переходная (а) и весовая (б) характеристики ОСС

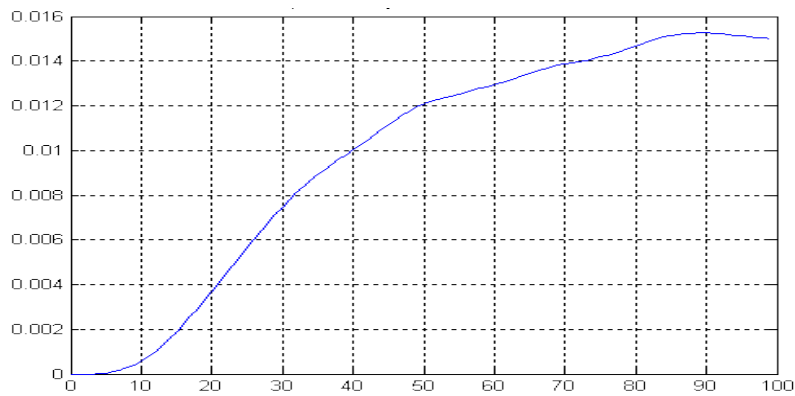


Рис. 5. Реакция ОСС на случайное воздействие

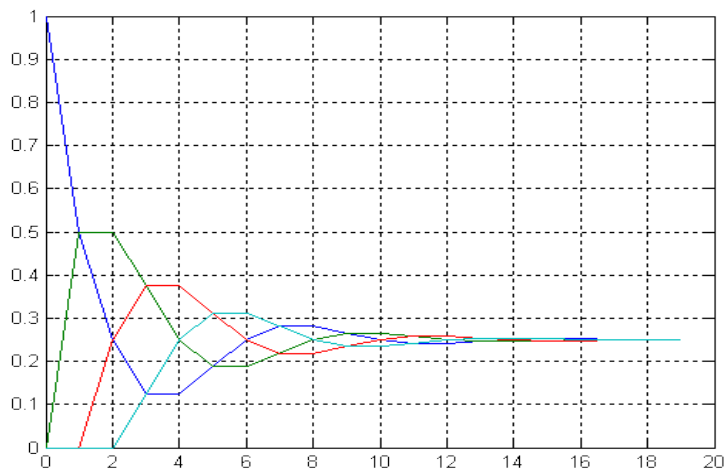


Рис. 6. Собственное движение данных в СЦС

### Заключение

Разработка и изучение сетецентрических систем на сегодняшний день является одним из основных направлений в области ВПК практически любой страны. Так, например, в России



активно развивается сеть ОАЦСС, являющаяся интегрированной информационной структурой ВС РФ. Однако анализ таких систем вручную практически невозможен, что требует проектирования автоматизированных систем расчета. Для анализа необходимо знание о динамических характеристиках таких систем, что позволяет оценивать реакцию систем на заранее известные изменения, на основе чего возможно моделирование реальных условий работы сетевых систем.

### Библиографический список

1. Кондратьев А. Сетевые фронт // Национальная оборона. 2009. № 45.
2. Атанов В. А., Бритов А. Г. Автоматизация расчета систем автоматического регулирования электроприводов // Завалишинские чтения'12, СПб., 2012. С. 12–19.

УДК 629.734.336

**Г. А. Подшивалов**

магистрант кафедры аэрокосмических измерительно-вычислительных комплексов

**В. Я. Мамаев**

канд. техн. наук, доц. – научный руководитель

## ИССЛЕДОВАНИЕ АЛГОРИТМОВ ЦЕНТРИРОВАНИЯ ЛЕГКИХ МОТОПЛАНЕРОВ В ВОЗДУШНЫХ ВОСХОДЯЩИХ ПОТОКАХ

Тенденция развития современной авиации показывает нарастающую роль применения автономных летательных аппаратов (АЛА). Это объясняется такими преимуществами АЛА как повышенные маневренные и скоростные характеристики, относительно малая стоимость, малая заметность и отсутствие потерь среди личного состава. Последними тремя достоинствами в полной мере обладают малогабаритные АЛА (МАЛА). Как правило, такие аппараты имеют массу до 5 кг, снабжены электродвигателем и имеют продолжительность полета в среднем 1 час и не более 2,5 часов. Малая продолжительность полета является основным недостатком МАЛА. Существует два глобальных направления увеличения продолжительности полета АЛА: увеличение емкости аккумуляторной батареи, и использование энергии окружающей среды. Последнее направление также можно разделить на несколько направлений, главные из которых: использование солнечной энергии и использование восходящих воздушных потоков (ВВП), частным случаем которых являются особо популярные среди планеристов, тепловые конвективные потоки (ТКП). Если использование солнечных батарей применительно, скорее, к крупным летательным аппаратам (ЛА), то использование восходящих потоков как раз более характерно для малых ЛА. Отметим, что данная тематика начала активно развиваться в середине прошлого десятилетия. Очевидно, что для эффективного использования ВВП, необходимо выполнение ряда требований к ЛА и управлению его движением, одним из ключевых элементов которого является определение алгоритма центрирования МАЛА в ВВП.

Чтобы определить, какая тактика парения автономных летательных аппаратов будет эффективной, был проведен сравнительный анализ некоторых существующих алгоритмов центрирования.

Рассматривались алгоритм Аллена и алгоритм вытягивания спирали. Алгоритм Аллена был разработан под руководством Майкла Аллена в середине прошлого десятилетия в NASA Dryden Flight Research Center. Он описывается следующим набором правил [5]:

- при увеличении вертикальной скорости ЛА, уменьшить радиус кривизны траектории полета (задать крен приблизительно  $15^{\circ}$ );
- при уменьшении вертикальной скорости ЛА, увеличить радиус кривизны траектории полета (задать крен приблизительно  $50^{\circ}$ );

– при постоянной вертикальной скорости ЛА, держать постоянный радиус кривизны траектории полета (задать крен приблизительно  $25^{\circ}$ ).

В отечественном планеризме уже давно известен такой алгоритм центрирования как алгоритм втягивания спирали [4]. Его суть заключается в том, что ЛА движется в потоке до достижения максимальной вертикальной скорости, после чего совершает разворот на  $180^{\circ}$ . Если после разворота скорость не уменьшилась, то движение по окружности продолжается. Если нет – совершается доворот на  $90^{\circ}$ . Таким образом, как правило, ЛА окажется в поле действия положительной вертикальной скорости в течение длительного времени.

В качестве МАЛА, способного совершать парящий полет и использовать для длительного пребывания в воздухе ВВП, рассматривался класс мотопланеров. Эти ЛА совмещают в себе аэродинамические характеристики планеров и возможность лететь при помощи винтового двигателя при отсутствии ВВП и для выхода на необходимую высоту.

Перечислим основные параметры мотопланера, моделирование полета которого было проведено: масса  $m = 0,5$  кг, размах крыла  $l = 1,05$  м, площадь крыла  $S_{кр} = 0,152$  м<sup>2</sup>, коэффициент лобового сопротивления  $C_x = 0,1$ , коэффициент подъемной силы  $C_y = 0,65$ . Для каждого ЛА можно заранее определить собственную скорость планирования, которая представляется уравнением [2]:

$$V_{пл} = \sqrt{\frac{2mg}{C_y \rho S} \cos \theta},$$

где  $m$  – масса ЛА;  $S$  – площадь несущей поверхности;  $C_y$  – коэффициент подъемной силы;  $\rho$  – плотность воздуха;  $\theta$  – угол планирования, который определяется из выражения:

$$\theta = \arctg\left(\frac{C_x}{C_y}\right).$$

Принималась следующая математическая модель движения мотопланера:

$$\begin{cases} m \frac{dV}{dt} = T - D - mg \sin \theta - m \frac{dW_{xy}}{dt} \cos \theta \sin \psi, \frac{dh}{dt} = V \sin \theta \\ mV \cos \theta \frac{d\psi}{dt} = L \sin \gamma - m \frac{dW_{xy}}{dt} \cos \psi, \frac{dx}{dt} = V \cos \theta \sin \psi + W_{xy}, \\ mV \frac{d\theta}{dt} = L \cos \gamma - mg \cos \theta + m \frac{dW_{xy}}{dt} \sin \theta \sin \psi, \frac{dy}{dt} = V \cos \theta \cos \psi. \end{cases}$$

где  $D = C_x \frac{\rho V^2}{2} S$  – сила лобового сопротивления,  $L = C_y \frac{\rho V^2}{2} S$  – подъемная сила,  $V$  – истинная воздушная скорость;  $T$  – сила тяги ( $T = 0$ ),  $D$  – сила сопротивления воздуха;  $x$  и  $y$  – координаты МАЛА в прямоугольной системе координат,  $h$  – истинная высота полета МАЛА,  $C_y$  – коэффициент подъемной силы,  $C_x$  – коэффициент лобового сопротивления,  $\rho$  – плотность воздуха,  $\psi$  – угол курса,  $\gamma$  – угол крена,  $W_{xy}$  – скорость горизонтального ветра.

Очевидно, что каждый ТКП существует какое-то определенное время. Считается, что продолжительность существования ТКП находится в диапазоне от нескольких до 20 минут [3]. Кроме того, следует учитывать, какое количество ТКП приходится на единичный участок площади. Согласно проведенным исследованиям [1] ВВП с помощью планеров установлено, что в летнее время года в среднем на 1 км<sup>2</sup>, приходится около 3–4 крупных ТКП диаметром не менее 100 м. Вертикальная скорость в потоке зависит от высоты на которой рассматривается движение воздуха, от текущей высоты конденсации облаков (ВКО) и от некоторых других параметров, характерных для той или иной местности. Деятельность тепловых конвективных потоков приводит к обра-

зованию конвективных облаков. К этому типу относятся кучевые и кучево-дождевые облака, для которых характерна ВКО от 800 м до 1500 м. Под ними чаще всего образуются ТКП.

При моделировании полета скорость ветра принималась равная нулю. Другими атмосферными возмущениями также пренебрегли. Тяга также принималась равная нулю, т.е. моделировался именно парящий полет. Принималась математическая модель ТКП подробно описанная в [6].

На первом этапе моделирования было произведено по 20 запусков мотопланера с высоты 500 м и ВКО 1000 м для каждого алгоритма. Оказалось, что при номинальной продолжительности полета в 366 секунд алгоритм Аллена в среднем позволяет пролететь 1137 секунд, а алгоритм втягивания спирали 3036 секунд. При маневрировании по алгоритму втягивания спирали угол крена при развороте принимался равным  $25^\circ$ . На рис. 1 представлена диаграмма результатов моделирования, в которой представлены продолжительности каждого полета.

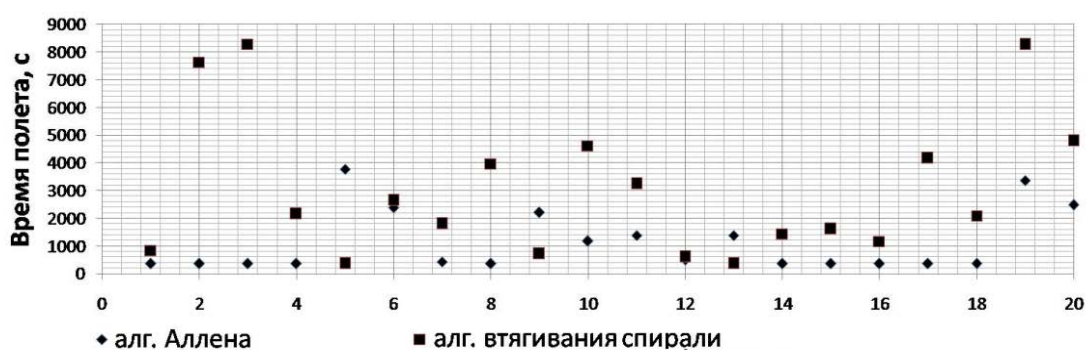


Рис. 1. Диаграмма продолжительности парящего полета при использовании алгоритмов центрирования

На рис. 2 представлены по одной визуализации траектории парящего полета для каждого алгоритма.

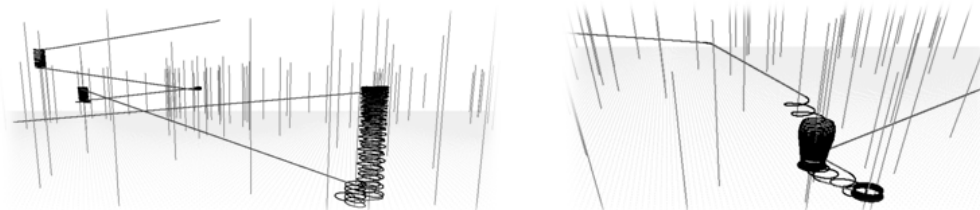


Рис. 2. Визуализация траектории парящего полета для алгоритма втягивания спирали (слева) и алгоритма Аллена (справа)

Таким образом, в невозмущенной атмосфере, алгоритм втягивания спирали показал значительно лучший результат. Далее он рассматривался отдельно.

Было проведено три серии моделирования полетов на квадратном участке площадью  $12 \text{ км}^2$  с ВКО 800 м, 1100 м и 1400 м соответственно. На этом участке было размещено 532 ТКП таким образом, чтобы их центральные оси располагались не ближе 200 м друг от друга. Запуск мотопланера производился из центра участка с различной начальной высоты. В каждой серии

было проведено по 100 запусков с высоты от 100 м до ВКО с интервалом 100 м. На рис. 3 представлена зависимость продолжительности парящего полета мотопланера от высоты запуска при различных ВКО:

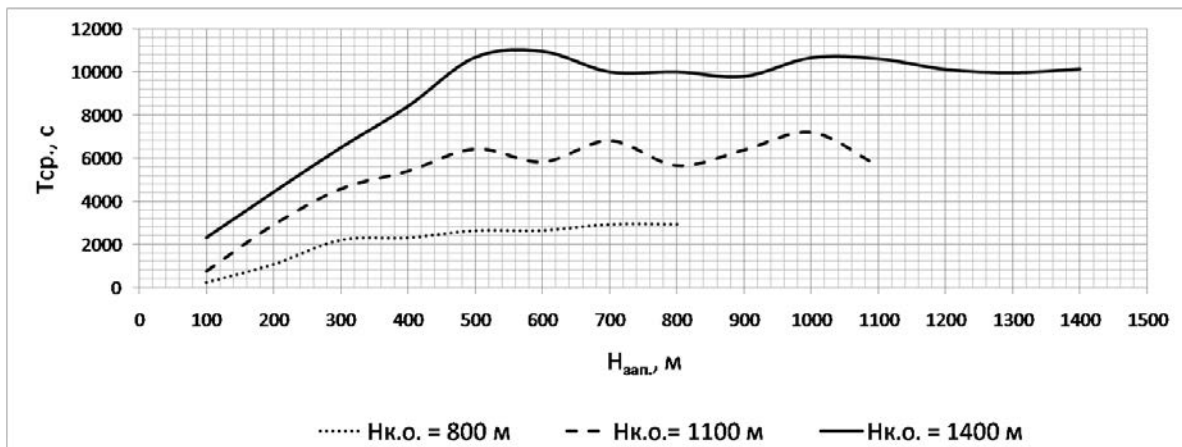


Рис. 3. Зависимость продолжительности парящего полета мотопланера от высоты запуска при различных ВКО

На рис. 3 видно, что продолжительность полета зависит и от высоты, с которой мотопланер переходит в парящий полет и от ВКО. При этом можно констатировать, что, начиная с некоторой высоты запуска мотопланера, среднее время парящего полета будет приблизительно одинаковым. Таким образом, не имеет смысла выводить мотопланер на слишком большие высоты, для перехода в парящий полет.

#### Заключение

Сравнительный анализ алгоритмов втягивания в спираль и алгоритма Аллена показал большую эффективность алгоритма втягивания в спираль в невозмущенной атмосфере.

Для идеализированного случая невозмущенной атмосферы определены такое условие эффективности использования алгоритма вытягивания в спираль, как высота перехода в парящий полет.

По результатам моделирования парящих полетов установлено, что в невозмущенной атмосфере применение алгоритма вытягивания в спираль позволяет увеличить продолжительность полета до трех часов при благоприятных погодных условиях.

#### Библиографический список

1. Воронцов П. А.. Турбулентность и вертикальные токи в пограничном слое атмосферы, Главная геофизическая обсерватория им. А. И. Воейкова. Л., Гидрометеиздат, 1966. 296 с.
2. Двоеносов Д., Замятин В., Снешко Ю. Нагрузки, действующие на планер в полете. М.: Изд-во ДОСААФ, 1963. 138 с.
3. Довгалою Ю. А., Ивлев Л. С. Физика атмосферных аэрозольных систем. СПб: Изд-во СПбГУ, 1999. 188 с.
4. Руденский Е. Г. Полет на планере. Пособие для планеристов / Е. Г. Руденский. – М.: Изд-во ДОСААФ, 1977. – 72 с.
5. Allen M. J. Autonomous soaring for improved endurance of a small uninhabited air vehicle // AIAA 2005-1025, 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 10–13 January, 2005.
6. Allen M. J. Updraft model for development of autonomous soaring uninhabited air vehicles // AIAA 2006-1510, 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 9–12 January, 2006.

УДК 62-83(075.8)

**Д. М. Рагинис**

студент кафедры вычислительных систем и сетей

**Г. С. Бритов**

канд. техн. наук, доц. – научный руководитель

## ДИАГНОСТИРОВАНИЕ ДЕФЕКТОВ В СИСТЕМАХ УПРАВЛЕНИЯ

В настоящее время достаточно актуальной проблемой является разработка и проектирование систем управления различного назначения. Практически любая система измерения топлива на сегодняшний момент использует именно систему управления как наиболее правильное техническое решение. Систему управления можно использовать не только в системе измерения топлива, но и в любой автоматической системе. Поэтому важной задачей является обеспечение диагностирования дефектов в системах управления.

Целью выполненной работы было проанализировать современные методы обнаружения дефектов в системах управления, рассматривая их как линейные динамические системы [1–5].

### Параметрическая диагностика

Объект диагностирования – линейная система управления в виде последовательного соединения звеньев первого порядка. Для выполнения параметрического диагностирования необходимо выполнение следующих требований:

- на вход системы можно подавать тестовые сигналы;
- при диагностировании для измерения доступны все выходы звеньев;
- известны передаточные функции звеньев;
- известны номинальные значения коэффициентов передачи и постоянные времени звеньев.

Параметрическая диагностика требует определения в специальном режиме работы системы истинных значений коэффициентов передачи и постоянных времени звеньев. Сравнение их с соответствующими номинальными значениями позволяет принять обоснованное решение о наличии дефектов в системе. В дальнейшем принимается гипотеза об однократности дефекта в системе.

Постановка задачи параметрической диагностики: определить истинные параметры передаточных функций звеньев системы. Передаточные функции звеньев:

$$W_i(s) = \frac{K_i}{T_i s + 1}, \quad i = 1, \dots, n$$

где  $K_i$ ,  $T_i$  – коэффициент передачи и постоянная времени звена.

Алгоритм параметрической диагностики:

1. Определение коэффициентов передачи:
  - а) на вход системы подать единичный скачок;
  - б) измерить выходные сигналы звеньев в стационарном режиме  $y_{ci}$ ,  $i = 1 \dots n$ ;
  - в) рассчитать коэффициенты передачи звеньев:

$$K_1 = y_{c1}, \quad K_i = \frac{y_{ci}}{K_1 \dots K_{i-1}}$$

Если  $K_i \approx K_{iном}$ , то дефекта нет, иначе ошибочен коэффициент передачи, для которого не выполняется приблизительное равенство.

2. Определение постоянных времени звеньев:

- а) задать время проверки  $t_1 = T_1$ ;

- b) на вход системы подать единичный скачок;
- c) измерить выходной сигнал звена  $y_1(t_1)$ ;
- d) рассчитать номинальное значение  $y_{1\text{ nom}}(t_1)$  с помощью программы rst.m;
- e) если  $y_1(t_1) \approx y_{1\text{ nom}}(t_1)$ , то дефекта нет, иначе ошибочна постоянная времени  $T_1$ ;
- f) повторить п.п. 2a–2e для остальных звеньев системы.

Задача расчета номинальных выходных сигналов звеньев решается в программе rst.m. С ее помощью в математическом пакете MatLab можно получить все необходимые данные.

Рассмотрим результаты компьютерного эксперимента. Объект диагностирования – система управления в виде последовательного соединения трех звеньев первого порядка. Схема модели в пакете SIMULINK показана на рис. 1.

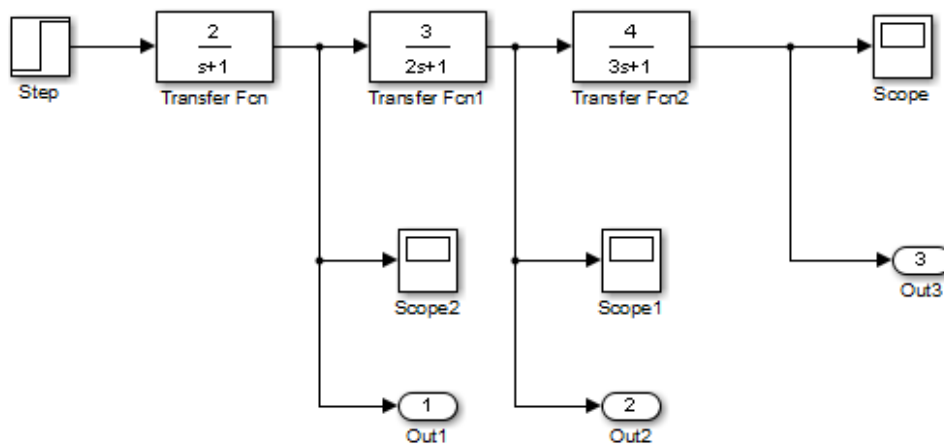


Рис. 1. Схема моделирования

Результаты работы программы rst.m при определении коэффициентов передачи:

```
>> rst1
Начало
Конец тестирования? yes
Результат:
    2.0000    3.0000    3.9992
```

Готово

Результаты работы программы rst.m при определении постоянных времени:

```
>> rst1
Начало
Конец тестирования? yes
Номер звена =
    1
Числитель ПФ = 2
Знаменатель ПФ = [1 1]
Результаты:
    1.2642
    1.2642
Результаты:
    2.3975
    2.3975
Готово
Начало
Номер звена = 3
Постоянная времени = 3
Готово
Начало
Конец тестирования? yes
```

Готово	Номер звена =	3
Начало		
Номер звена = 2	Числитель ПФ = 4	
Постоянная времени = 2	Знаменатель ПФ = [3 1]	
Готово	Результаты:	
Начало		5.0921
Конец тестирования? yes		
Номер звена =		5.0921
2		
	Готово	
Числитель ПФ = 3	Конец	
Знаменатель ПФ = [2 1]	>>	

Таким образом, параметрическая диагностика систем управления позволяет просто и однозначно определять работоспособность системы и осуществлять локализацию дефекта до уровня звена.

#### Тестовая диагностика

При всем разнообразии методов тестового диагностирования, в них можно выделить генератор тестовых воздействий, подаваемых на проверяемый объект, и устройство анализа реакции объекта на эти воздействия. В ряде методов используются простые генераторы, но сложные анализаторы, в других методах, наоборот, стараются упростить процедуру анализа за счет усложнения входных сигналов [3]. На рис. 2 показано дерево узлов моделирования системы тестового диагностирования по IDEF3-технологии. Глубина дерева невелика, всего три уровня. Тем не менее, модель дает первое, общее представление о процессе тестирования любого объекта диагностирования (ОД).



Рис. 2. Дерево узлов IDEF3-модели системы тестового диагностирования

Функциональная диагностика

Простейшее функциональное диагностирование может осуществляться методом контроля по модели, в котором диагностические признаки получают как отклонения выходных сигналов объекта диагностирования от соответствующих сигналов модели объекта. Другой метод связан с модальным диагностированием динамических систем, когда моделируется не весь объект, а только одна его мода. Возможен также параметрический контроль, предполагающий идентификацию одного или нескольких параметров объекта по входным и выходным сигналам [2]. На рис. 3 показано дерево узлов моделирования системы функционального диагностирования по IDEF3-технологии [3].

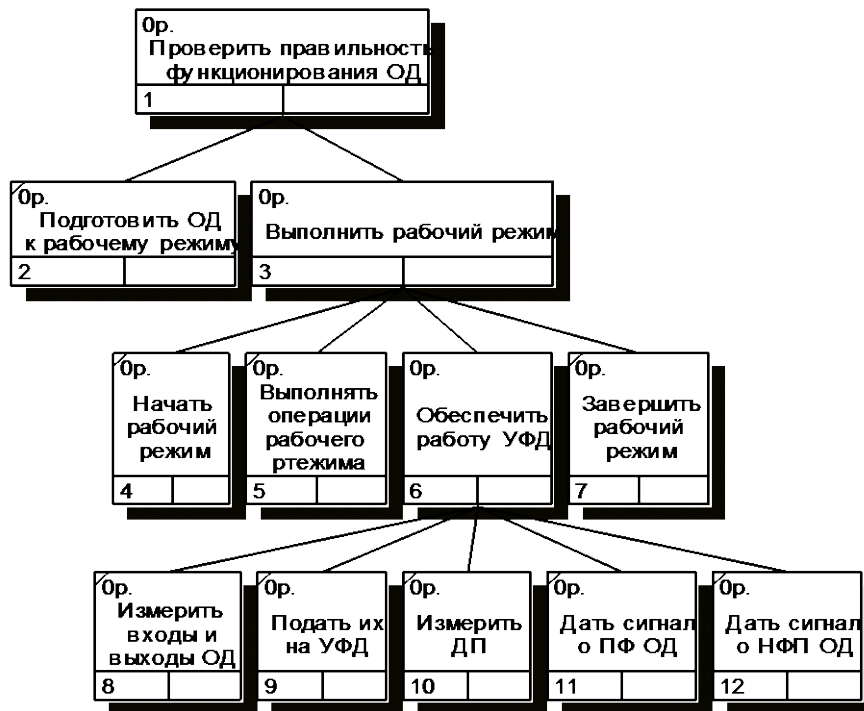


Рис. 3. Дерево узлов модели системы функционального диагностирования

Заключение

В статье были рассмотрены три основные группы методов диагностирования: параметрическое, тестовое и функциональное. Выполнено моделирование параметрической диагностики на примере простой динамической системы. Исходя из результатов моделирования, можно заключить, что даже для столь малой системы требуется точное выполнение ряда требований. На нынешнем этапе развития даже самые современные, сверхмощные системы диагностики могут представлять не полную картину диагностируемых параметров, и как следствие, не дадут оператору системы измерения топлива возможности сделать прогноз о возможных или уже имеющихся дефектах в системе.

Совершенствование известных и разработка новых методов диагностирования и их исследование позволит повысить эффективность контроля и диагностики систем автоматического управления и уменьшить затраты на организацию диагностирования.

Библиографический список

1. Мироновский Л. А. Функциональное диагностирование динамических систем: Научное издание / СПб., 1998. 256 с.



2. Шалобанов С. В., Шалобанов С. В. Способ поиска неисправностей блоков в непрерывной динамической системе / Патент на изобретение № 2473105. 2013.
3. Дедученко Ф. М. Научно-технические аспекты создания объектно-ориентированных систем контроля, диагностирования и аварийной защиты / [www.mrks.ru](http://www.mrks.ru) [Электронный ресурс].
4. Бритов Г. С., Мироновский Л. А. Проблемы тестового диагностирования систем управления // Известия Международной академии высшей школы. 2011. № 2. С. 41–49.
5. Бритов Г. С. Метод тестового диагностирования линейных динамических систем // Информационно-управляющие системы. 2015. № 1. С. 77–84.

УДК 004.057.4

**Н. И. Синёв**

студент кафедры аэрокосмических компьютерных и программных систем

**И. Л. Коробков**

аспирант кафедры аэрокосмических компьютерных и программных систем

**В. Л. Оленев**

канд. техн. наук – научный руководитель

### ТЕСТИРОВАНИЕ ПРОТОКОЛА СТП-ИСС НА МОДЕЛИ БОРТОВОЙ СЕТИ КОСМИЧЕСКОГО АППАРАТА

Современная тенденция по усложнению архитектуры сетей, взаимодействию нескольких разнородных сетей и созданию мультисервисных сетей привела к усложнению их внутренней структуры. Поэтому для обеспечения совместимости и правильной работы многоуровневых сетей, примером которых являются сети SpaceWire [1], необходимо проводить тестирование.

Существует несколько видов тестирования:

- *аттестационное тестирование* – проверка спецификации протокола путем выполнения комплектов тестовых сценариев, содержащихся в спецификации протокола;
- *тестирование производительности* – тестирование системы под нагрузкой и сверка полученных результатов с допустимым диапазоном значений в спецификации;
- *тестирование совместного функционирования* – тестирование разных реализаций одного и того же протокола на совместную работу;
- *тестирование взаимодействия* – тестирование взаимодействия разных протоколов и систем сигнализации;
- *тестирование функциональности* – тестирование для проверки реализуемых функциональных требований;
- *мониторинг* – прием данных из каналов передачи и их проверка на соответствие данным спецификации протокола (пример, кодирование) [2].

Нами разработана имитационная модель транспортного протокола СТП-ИСС и проведено тестирование протокола СТП-ИСС, соответствующее аттестационному тестированию и в некоторой степени тестированию производительности, заключающееся в проведении тестов по проверке выполнения требований по допустимым задержкам передачи пакетов СТП-ИСС по сети SpaceWire. Данная статья посвящена тестированию протокола транспортного уровня СТП-ИСС для бортовых сетей космических аппаратов.

Протокол транспортного уровня СТП-ИСС

СТП-ИСС – Сетевой Транспортный Протокол для космических аппаратов (КА) АО «ИСС» имени академика М. Ф. Решетнева. Данный протокол регламентирует форматы передаваемых данных и правила передачи сообщений между абонентами бортовой сети SpaceWire.

Особенности протокола СТП-ИСС [3]:

- определяет информационно-логическое взаимодействие;

- регламентирует форматы передаваемых данных и правила передачи сообщений между абонентами бортовой сети SpaceWire;
- обеспечивает транспортировку данных между удаленными узлами сети с предоставлением требуемого качества сервиса в соответствии с приоритетами потоков данных [4];
- предоставляет возможность повторной передачи данных и обнаружения ошибок в принимаемых данных, таким образом, обеспечивая надежность доставки данных;
- предоставляет возможность гибкой настройки параметров через конфигурационный интерфейс, который позволяет настраивать протокол в зависимости от требований к качеству предоставляемого сервиса и типа используемой аппаратуры.

Бортовое программное обеспечение космического аппарата может выполнять функции прикладного, представительского и сеансового уровней в соответствии с моделью OSI.



Рис. 1. Соотношение стека протоколов SpaceWire и протокола СТП-ИСС с моделью OSI

Существуют две редакции протокола СТП-ИСС, которые отличаются друг от друга поддерживаемыми в них механизмами [4]. В данной работе проводилось тестирование и исследование спецификации протокола СТП-ИСС ред. 2.

#### Тестирование протокола СТП-ИСС

Модель бортовой сети, функционирующей по стандарту SpaceWire [1], предоставляется средством моделирования сетей SpaceWire – библиотекой NetSim [5, 6] и состоит из следующих элементов:

- модель узла (терминала) SpaceWire;
- модель коммутатора SpaceWire;
- модель канала SpaceWire с помехами.

Поверх каждого узла SpaceWire функционирует имитационная модель транспортного протокола СТП-ИСС. Модель СТП-ИСС состоит из трех блоков (рис. 2), которые представляют собой отдельные C++ классы с наборами SystemC портов и интерфейсов [7, 8]:

- *Trafgen* – модель прикладного процесса, осуществляет необходимое взаимодействие с моделью транспортного протокола STP\_ISS;
- *STP\_ISS* – модель протокола транспортного уровня СТП-ИСС;
- *Sw\_Wrapper* – модуль приема и передачи пакетов СТП-ИСС, маркеров времени и распределенных прерываний из/в модель терминала SpaceWire.

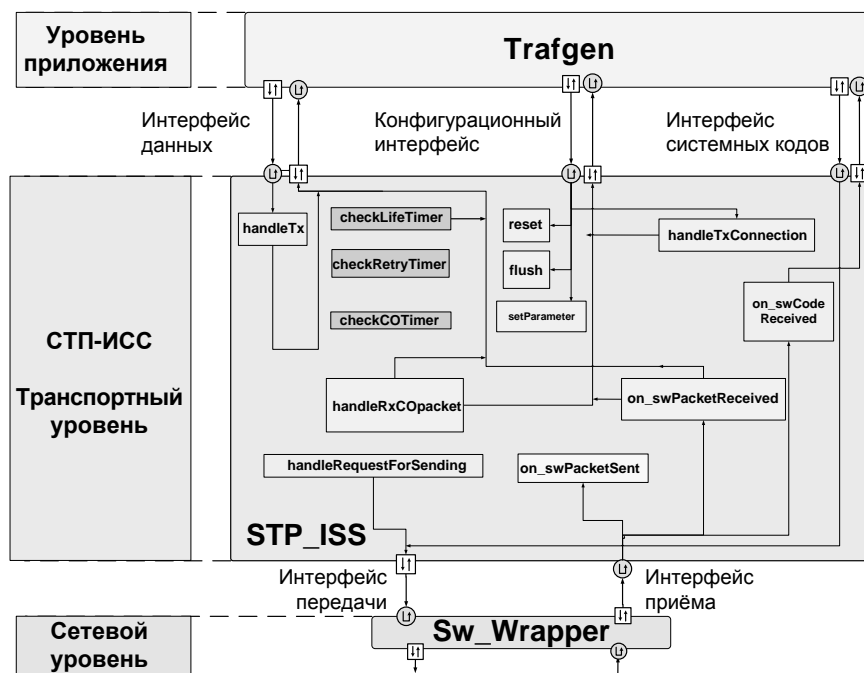


Рис. 2. Схема модели STP-ИСС

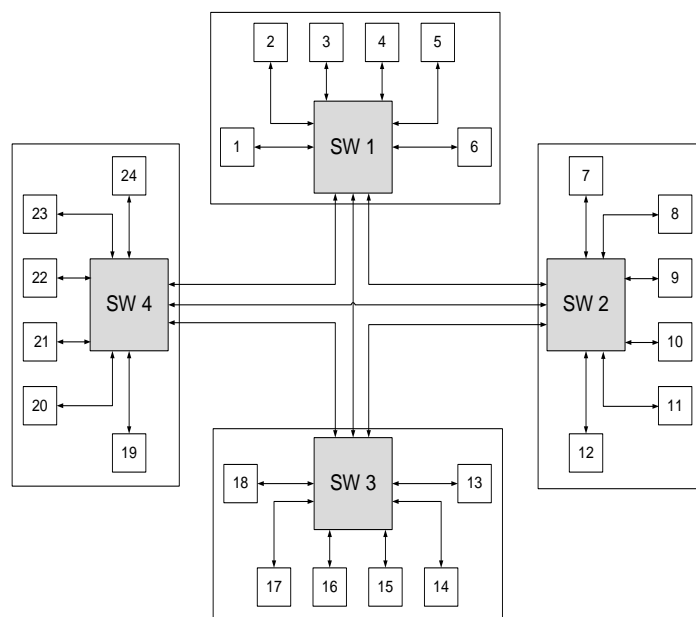


Рис. 3. Пример бортовой сети SpaceWire KA

Было проведено множество тестов (более 50) с различными параметрами для проверки корректности механизмов протокола STP-ИСС. Тестирование проводилось на примере планируемой бортовой сети KA со смешанной топологией (рис. 3), собранной при помощи библиотеки NetSim.

Некоторые из проведенных тестов приведены ниже:

- передача данных по приоритетам при гарантированной и негарантированной доставке данных;
- передача системных кодов (маркеров времени, кодов прерываний, кодов подтверждения прерываний);

- передача данных по транспортным соединениям: установка соединения, передача данных и закрытие соединения;
- передача данных при блокировках трафика в сети;
- передача данных при наличии неисправного узла в сети;
- проверка выполнения требований по допустимым задержкам передачи пакетов.

Все тесты были выполнены успешно. Это означает, что в рамках проведенного тестирования не было выявлено ситуаций, приводящих к непредвиденному функционированию протокола СТП-ИСС и нарушению его логики работы.

При этом в ходе тестирования были обнаружены ситуации, при которых ухудшается производительность бортовой сети SpaceWire:

- при передаче данных по приоритетам при гарантированной и негарантированной доставке обнаружена монополизация канала в передающем узле высокоприоритетным типом данных. Так как пакеты с наивысшим приоритетом отправляются в первую очередь, чем пакеты с меньшим приоритетом, то низкоприоритетные пакеты не были отправлены в сеть и удалялись из выходного буфера после истечения таймера времени жизни пакета.

*Возможное решение:* чередование отправки высокоприоритетных и низкоприоритетных пакетов;

- выявлена избыточная повторная отправка пакетов при гарантированной доставке данных при наличии неисправного узла в сети. Это происходит, потому что таймер повтора истекает до прихода пакета подтверждения к отправителю, так как неисправный узел своей незапланированной передачей данных на длительное время занимает порт в коммутаторе SpaceWire, из-за чего пакет подтверждения вынужден ожидать освобождения порта.

*Возможное решение:* использование выборочных подтверждений (Selective Acknowledgements) для подтверждения сразу нескольких полученных пакетов данных, что позволит сократить количество повторных отправок данных.

Также было проверено выполнение требований по допустимым задержкам передачи пакетов: данные передавались по самому длинному маршруту между регионами сети для проверки выполнения требований. Результаты проверки приведены в таблице. Все фактические задержки по доставке пакетов удовлетворяют допустимым значениям.

Таблица

Требования и результаты по задержкам доставки пакетов по сети SpaceWire

Тип пакета	Длина	Требование задержки, мс	Фактическая задержка, мс
Команды управления	16 бит	$\leq 0,5$	0,02653
Срочные сообщения с подтверждением	4 байта	$\leq 0,25$	0,02653
	1 Кбайт	$\leq 0,5$	0,23061
Обычные сообщения с подтверждения	4 байта	$\leq 1$	0,02653
	1 Кбайт	$\leq 1$	0,23061
Обычные сообщения без подтверждения	4 байта	$\leq 1$	0,01741
	1 Кбайт	$\leq 1$	0,22157
Маркеры времени	6 бит	$\leq 0,1$	0,00296
Коды прерываний	5 + 1 бит	$\leq 0,1$	0,0028
	5 + 1 бит	$\leq 0,1$	0,00272

### Заключение

В данной статье была представлена работа по тестированию транспортного протокола СТП-ИСС для бортовых сетей SpaceWire космических аппаратов нового поколения. В ходе проведенного тестирования не было выявлено несоответствий и ситуаций, приводящих к непредви-

денному функционированию протокола и нарушению его логики работы. На основании этого можно сделать вывод о корректности спецификации СТП–ИСС. При этом были обнаружены ситуации, при которых ухудшалась производительность бортовой сети SpaceWire, и предложены возможные решения для их устранения. Кроме того, было проверено выполнение требований по допустимым задержкам передачи пакетов СТП–ИСС по бортовой сети SpaceWire при помощи разработанной модели протокола СТП–ИСС и средства моделирования сетей SpaceWire – NetSim. Задержки передачи пакетов не превосходят допустимых значений. Все полученные результаты были предоставлены разработчикам протокола СТП–ИСС и могут быть учтены при выпуске следующих редакций протокола.

### Библиографический список

1. Стандарт ECSS-E-50-12C. Space engineering. SpaceWire – Links, nodes, routers and networks. European cooperation for space standardization. ESA Publications Division ESTEC, Noordwijk, The Netherlands. 2008. 129 с.
2. Гольдштейн Б. С., Ехриель И. М., Рерле Р. Д. Тестирование телекоммуникационных протоколов: проблемы и подходы // Электронное издание. 2002. Режим доступа: [http://www.ccc.ru/magazine/depot/02\\_12/read.html?0303.htm](http://www.ccc.ru/magazine/depot/02_12/read.html?0303.htm)
3. Sheynin Y., Olenev V., Lavrovskaya I., Korobkov I., Dymov D. "STP-ISS Transport Protocol for Spacecraft On-board Networks", Proceedings of 6th International Conference SpaceWire 2014 Program (pp. 26–31), Athens, Greece, 2014.
4. Sheynin Y., Olenev V., Lavrovskaya I., Korobkov I., Kochura S., Dymov D. "Second Revision of STP-ISS Transport Protocol for On-Board SpaceWire Networks", Proceedings of 17th Conference of Open Innovations Association Finnish-Russian University Cooperation in Telecommunications (FRUCT) Program (pp. 192-200), P.G. Demidov Yaroslavl State University, Yaroslavl, Russia, 2015.
5. Eganyan A., Koblyakova L., Suvorova E. "SpaceWire Network Simulator" Proceedings from the 3rd International SpaceWire Conference, St.Petersburg, Russia, 2010.
6. Eganyan A., Sheynin Y., Suvorova E., Khakulin A., Orlovsky I. SpaceWire Network Simulator" 5th International SpaceWire Conference, Gothenburg, Sweden, 2013.
7. Open SystemC Initiative (OSCI), "IEEE 1666™-2005 Standard for SystemC", 2005. 614 p.
8. Black D., Donovan J., Bunton B., Keist A. SystemC: From the Ground Up, Springer, 2010.

УДК 629.7

**К. В. Соловяненко, Р. Ю. Авагян**

студенты кафедры аэрокосмических приборов и измерительно-вычислительных комплексов

**И. С. Зегжда**

канд. техн. наук, доц. – научный руководитель

### МЕТОДИКА РАСЧЕТА ЭЛЕМЕНТОВ СИСТЕМ АВАРИЙНОГО СПАСЕНИЯ КОСМОНАВТОВ

Система аварийного спасения (САС) – бортовая система для спасения космического корабля в случае возникновения аварийной ситуации на ракете-носителе (РН). В случае аварийной ситуации его включение и работа позволяют «увести» бытовой и приборный отсеки от места старта на безопасное расстояние.

Система спасения должна функционировать в любых условиях, вплоть до неуправляемого хаотичного падения ракеты. Для этого сначала основные двигатели САС отрывают спасаемую часть от ракеты и быстро уводят ее в сторону, а затем включаются управляющие двигатели, которые формируют нужную траекторию спуска. Скоротечность многих аварийных ситуаций требует от САС высокого быстродействия, поэтому все ее двигатели – твердотопливные. По сравнению с жидкостными они проще, надежнее и быстрее набирают максимальную тягу. Перегрузку в 20 единиц, действующую в направлении «от груди к спине», человек способен выносить всего лишь около секунды. Этого времени не хватит, чтобы увести спасаемую часть корабля на безопасное расстояние от ракеты. Приходится ограничивать тягу спасательных двигателей так, чтобы перегрузка не превышала 10–15 единиц, так как такое ускорение можно поддерживать дол-

ше. Система аварийного спасения «вводится» за 15 мин до подъема РН и обеспечивает спасение экипажа в случае аварии РН как на стартовом столе, так и на любом участке полета [1].

Расчет элементов системы аварийного спасения космонавтов  
 Определение массового секундного расхода:

$$m_c = \rho^* V^* S^* , \quad (1)$$

где  $\rho^*$  – критическая плотность,  $V^*$  – критическая скорость,  $S^*$  – площадь критического сечения. Газодинамические формулы позволяют определить критические параметры через параметры

торможения. Подставляя их в формулу (1), получим:  $m_c = \frac{AP_0 S^*}{\sqrt{T_0}}$ , где  $A = \sqrt{\frac{k}{R}} \left( \frac{2}{k+1} \right)^{\frac{k+1}{2(k-1)}}$ ,

где  $P_0$  и  $T_0$  – давление и температура торможения в камере сгорания двигателя,  $R$  – газовая постоянная топлива после его сгорания,  $k$  – коэффициент адиабаты, характеризующий отношение удельных теплоемкостей газа после сгорания топлива.

Для реальных процессов вводят коэффициент расхода, равный 0,98. Таким образом формула для расчета массового секундного расхода:  $m_c = 0,98 \frac{AP_0 S^*}{\sqrt{T_0}}$ .

Время работы основного двигателя системы аварийного спасения (САС)

Время работы двигателя САС можно определить визуально по натурной видеосъемке спасения космонавтов Волкова и Стрекалова при аварийном старте. Если такой возможности нет, то, опираясь на опыт и литературные источники [3], можно задаться временем работы САС из диапазона (6–8) сек. Неудачный выбор времени работы САС может сказаться на том, что не будет достигнута необходимая высота для успешной работы парашютной системы.

Скорость горения твердотопливного заряда

Скорость горения топлива сильно зависит от характера распределения температуры в прогревом слое. При быстром повышении давления прогретый слой может оказаться толще, чем слой, соответствующий стационарному горению. Нестационарный процесс в этом случае может быть взрывным. Если прогретый слой слишком тонок, прогревание не успевает за уровнем давления, то прогретый слой, соответствующий стационарному горению, может не образоваться и произойдет гашение заряда.

Скорость горения определяется по формуле  $U = BP_0^V$ , где  $B$  – коэффициент горения топлива, а  $P$  – давление (берется в МПа).

Форма и толщина свода заряда

Для обеспечения процесса горения, в котором общая площадь горения считается постоянной, можно выбрать форму заряда в виде полого толстого цилиндра, у которого торцевые поверхности бронируются. Бронирование заряда производится, когда необходимо, чтобы некоторые поверхности заряда не участвовали в процессе горения [2]. Диаметр заряда можно определить по формуле  $d_3 = 2Ut$ , где  $U$  – скорость горения заряда,  $t$  – время работы двигателя.

Толщина стенки двигателя

Камера сгорания двигателя находится под действием высокого давления и температуры. Толщина стенки камеры сгорания должна выдерживать эти напряжения – это зависит от выбора материала. В литературных источниках есть предложения по маркам стали, которые используются в ракетостроении. Для выбранной марки стали следует найти значение допустимого напря-

жения  $[\sigma]$ . Из [1] находим  $\sigma = \frac{fKP_0d_{дв}}{2[\sigma]}$ , где  $f$  – заброс давления при пуске двигателя в первые доли секунды;  $K$  – коэффициент запаса прочности,  $d_{дв}$  – диаметр двигателя.

Масса двигателя (без сопла)

Если предположить, что цилиндрическая обечайка и эллиптические днища имеют одну и ту же толщину обшивки, то масса обечайки и днища можно выразить так:

$$m_{об+дн} = S_б \cdot \delta \cdot \rho,$$

где  $S_б$  – боковая поверхность двигателя с учетом площади днища;  $\delta$  – толщина обечайки (корпуса);  $\rho$  – плотность материала двигателя.

С учетом добавки на эллиптичность днищ, следуя рекомендациям, напишем:

$$S_б = \pi d_n^2 (\lambda_k + 0.69).$$

Толщина стенки камеры сгорания определена ранее, величина удлинения корпуса двигателя находится с учетом рекомендаций конструкторов. Зная плотность выбранного материала для корпуса двигателя, можно найти массу двигателя.

Масса заряда и суммарная площадь горения

Массу заряда можно рассчитать по формуле:

$$m_з = V_з \cdot \rho_T,$$

где  $V_з$  – объем заряда,  $\rho_T$  – плотность заряда.

Объем заряда можно найти, используя данные полученные ранее:

$$V_з = \frac{\pi}{4} (d_3^2 - d^2) \cdot l_n = \frac{\pi}{4} (d_3^2 - d^2) \cdot \lambda_k d_3 = \frac{\pi}{4} \lambda_k d_3^3 \left[ 1 - \left( \frac{d}{d_3} \right)^2 \right]$$

Таким образом:  $V_з = \frac{\pi}{4} \lambda_k d_3^3 \left[ 1 - \left( \frac{d}{d_3} \right)^2 \right] \cdot \rho_T$ .

Суммарная площадь горения находится по формуле:  $m_{об+дн} = S_б \cdot \delta \cdot \rho$ .

Масса сопла

Масса сопла определяется по формуле:

$$m_c = 2S_{б.с} \cdot \delta_c \cdot \rho,$$

где  $S_{б.с}$  – боковая площадь сопла;  $\rho$  – плотность материала двигателя. При величине угла раскрытия сопла  $\alpha = 15^\circ$  боковая поверхность сопла может быть найдена по формуле:

$$S_{б.с} = 3.02 \cdot d_c^2 (f_c - 1) \quad (2)$$

где:  $d_c$  – диаметр критического сечения сопла;  $f_c = \frac{S_a}{S^*}$  – отношение выходного и критического сечений.

Полная температура в камере сгорания

Температуру торможения  $T_0$  в камере сгорания можно определить по формуле:

$$T_0 = T_{см} + 11.42(P_0 - 3.923); \quad (3)$$

Рекомендуемая температура для топлива  $T_{ст}$  обычно задается в исходных данных. Давление  $P_0$  в этой формуле берется в МПа.

Внутренняя баллистика сгорания

Рассмотрим процесс равновесного стабилизационного давления, при котором давление и температура в камере сгорания постоянны, а поверхность горения заряда не изменяется [5].

В этом случае расход газа через сопло можно записать в виде:

$$m_c = SU(\rho_T - \rho_K) \quad (4)$$

Сравнивая формулы (1) и (4) получим соотношение площади критического сечения к суммарной:

$$\frac{S^*}{S} = \frac{U\rho_T T_0 R - U\rho_0}{0.98AP_0 \sqrt{T_0} \cdot R},$$

подставив числовые значения, можно найти суммарную площадь горения.

Скорость газов и число М на срезе сопла

Опираясь на газодинамические формулы для расчета параметров газа в каналах переменного сечения [4] и применяя уравнение Бернулли для сечений «камера сгорания» – «срез сопла», найдем скорость газов на срезе сопла:

$$V_a = \sqrt{\frac{2k}{k-1} \cdot R \cdot T_0 \left(1 - \left(\frac{P_a}{P_0}\right)^{\frac{k-1}{k}}\right)}$$

Число М

Скорость звука на срезе и температура:

$$a_a = \sqrt{kRT_a}; T_a = T_0 \left(\frac{P_a}{P_0}\right)^{\frac{k-1}{k}}.$$

Используя газодинамическую формулу:  $\frac{P_0}{P_a} = \left(1 + \frac{k-1}{2} M_a^2\right)^{\frac{k}{k-1}}$  можно рассчитать  $M_a$ .

Тяга двигателя

В расчетном режиме по уравнению Мещерского тяга двигателя может быть определена по формуле:  $P = m_c \cdot V_a$ .

Чтобы оценить правильность расчетов необходимо рассмотреть уравнение продольного движения летательного аппарата в перегрузках:

$$\frac{1}{g} \frac{dV}{dt} = n_x - \sin\theta, \text{ где } n_x = \frac{P \cdot \cos\alpha - X_a}{mg}.$$

Известно [1], что космонавты испытывают перегрузку 5–6 единиц во время работы двигателя САС. Тяга двигателя при этом будет равна:  $P = n_x \cdot mg + X_a$ .

Сила лобового сопротивления  $X_a$  блока САС будет зависеть от коэффициента лобового сопротивления и характерной площади спасаемого, определяемой диаметром двигателя.

$$S = \frac{\pi d_{ДВ}^2}{4}, S = \frac{\pi d_{САС}^2}{4}$$

Т.о. формула для расчета силы лобового сопротивления –  $X_a = C_x \frac{\rho V^2}{2} S$ , где  $C_x = 0,5$  – коэффициент лобового сопротивления,  $S$  – суммарная площадь САС.  $S = S_{ДВ} + S_{САС}$ ,  $S_{САС} = \pi \cdot r_{САС}^2$ .

Рассчитав силу лобового сопротивления, можно рассчитать тягу и сравнить результат с расчетным режимом, отклонения должно быть небольшое.



Скорость полета САС

Конечную скорость полета системы аварийного спасения можно определить по формуле Циолковского.

Формула Циолковского определяется удельным импульсом, зависящим от газодинамических параметров в двигателе и отношением топлива к массе спасаемого объекта.

$$V = \varphi_{уд}^{\infty} \ln \frac{1}{1 - \mu_T},$$

где  $\mu_T = \frac{m_T}{m_0}$

$$\varphi_{уд}^{\infty} = V_a + \frac{RT_0}{V_a} \cdot \left( \frac{P_a}{P_0} \right)^{\frac{k-1}{2}}.$$

По предложенной методике авторами выполнен расчет системы аварийного спасения космонавтов и получены следующие результаты: массовый секундный расход топлива  $m_c = 2621$  кг/ч, скорость горения  $U = 11,78 \frac{MM}{C}$ , диаметр заряда  $d_3 = 164,92$  мм, толщина стенки двигателя  $\sigma = 10$  мм, масса сопла  $m_c = 27,05$  кг, температура торможения  $T_0 = 3301$  К, тяга двигателя  $P = 6,2 \cdot 10^4$  Н.

#### Библиографический список

1. Космические аппараты / Под общей редакцией проф. К. П. Феоктисова. М.: Военное издательство, 1983. 319 с.
2. Белов Г. В. и др. Основы проектирования ракет / М.: Машиностроение, 1974. 255 с.
3. Алемасов И. И. и др. Теория ракетных двигателей. М.: Машиностроение, 1980.
4. Лойцянский Л. Г. Механика жидкостей и газа, М.: «Наука», 1950.
5. Павлюк Ю. С. Баллистическое проектирование ракет: учеб. пособие для вузов / Издательство ЧГТУ, Челябинск, 1996.

УДК 335.585

**Н. В. Стеглов**

студент кафедры логистики и системного анализа

**Н. Н. Майоров**

канд. техн. наук, доц. – научный руководитель

### СРАВНЕНИЕ ABC И XYZ МЕТОДОВ АНАЛИЗА ШИРОКОЙ НОМЕНКЛАТУРЫ ЗАПАСОВ В ЛОГИСТИКЕ

ABC и XYZ методы анализа широкой номенклатуры запасов предприятия являются широко известными методами управления ресурсами. Каждый метод имеет свою специфику и назначение, преимущества и недостатки. Это, а так же как и где рациональнее использовать каждый способ, рассматривается в настоящей статье.

ABC анализ – метод, позволяющий классифицировать ресурсы предприятия по степени их важности. Метод основывается на принципе Парето, гласящем, что 20% деятельности в любой сфере жизни дает нам 80% результата. Для управления запасами этот закон звучит так: 20% имеющихся запасов дают нам 80% оборота, будь то запасы сырья и комплектующих, зерно либо продуктовый ряд предприятия и т. п. Метод ABC анализа проводится делением запасов на три категории: А – наиболее ценные, в – промежуточные, С – наименее ценные. В зависимости от целей анализа может быть выделено произвольное количество групп. Чаще всего выделяют 3, реже 4–5 групп.

Алгоритм проведения выглядит так:

- 1) определяем цель анализа;
- 2) определяем действия по итогам анализа;
- 3) выбираем объект анализа и параметр анализа;
- 4) составляем рейтинг;
- 5) рассчитываем долю параметра от общей суммы параметров с накопительным итогом;
- 6) выделяем группы *A, B, C*. Присваиваем значение групп выбранным объектам.

								C=ЕСЛИОШИБКА(G7/\$I\$1;"") D=ЕСЛИОШИБКА(СУММ(\$C\$2:C2);"") E=ЕСЛИОШИБКА(ЕСЛИ(D2<=\$I\$4;\$H\$4;"BC");"") F=ЕСЛИОШИБКА(ЕСЛИ(D2>\$I\$5;\$H\$6;\$H\$5);"") G=ЕСЛИОШИБКА(ЕСЛИ(E2=\$H\$4;E2;F2);"") I1=СУММ(B:B) I2=СЧЕТ(B:B) J4=СЧЕТЕСЛИ(G:G;H4) J5=СЧЕТЕСЛИ(G:G;H5) J6=СЧЕТЕСЛИ(G:G;H6) K4=J4/\$I\$2 K5=J5/\$I\$2 K6=J6/\$I\$2			
	A	B	C	D	E	F	G	H	I	J	K
	код товара	выручка за период, руб.	доля	Доля нарастающий итогом	группа А	группа С	ABC	сумарный объем спроса	102 370		
1											
2	648580	18 150	18%	18%	A	B	A	количество позиций	17		
3	648233	11 000	11%	28%	A	B	A		результат	количество товаров в группе	вклад
4	642543	6 250	6%	35%	A	B	A	A	0.8	12	71%
5	648630	6 100	6%	41%	A	B	A	B	0.95	3	18%
6	415949	5 600	5%	46%	A	B	A	C		2	12%
7	648187	5 600	5%	51%	A	B	A				
8	607716	5 000	5%	56%	A	B	A				
9	629112	4 800	5%	61%	A	B	A				
10	648439	4 800	5%	66%	A	B	A				
11	143507	4 725	5%	70%	A	B	A				
12	648632	4 650	5%	75%	A	B	A				
13	631700	4 420	4%	79%	A	B	A				
14	406374	4 375	4%	83%	BC	B	B				
15	630244	4 350	4%	88%	BC	B	B				
16	647527	4 300	4%	92%	BC	B	B				
17	648194	4 250	4%	96%	BC	C	C				
18	647560	4 000	4%	100%	BC	C	C				

Рис. 1. Расчет и распределение по рейтингам метом ABC анализа

Методов выделения групп существует порядка десяти, наиболее применимые из них: эмпирический метод, метод сумм и метод касательных.

Зачем всё это надо? Для того чтобы по-разному управлять разными запасами. Выводить капитал из невыгодных сегментов предприятия и вкладывать его в более в нем нуждающиеся. Например, дорогие запасы группы *A* закупать более мелкими партиями, чтобы не омертвлять капитал, а также чаще и точнее проводить их инвентаризацию. Наоборот, запасы группы *C* закупать большими партиями, а инвентаризацию проводить «на глаз».

Преимущества метода *ABC* заключаются в простоте обращения, в его прозрачности, любой этап анализа можно проследить и, если нужно, подкорректировать. Так же *ABC* анализ замечателен своей универсальностью, им можно анализировать все что угодно, что можно разделить на соответствующие элементы: товарооборот, деньги, урожай зерна и т. д. *ABC* анализ очень доступен, его можно проводить в большом количестве программ, начиная от Excel до всевозможных приложений и макросов. *ABC* анализ при успешном использовании позволяет сократить и высвободить огромное количество временных и трудовых ресурсов. Этого можно достичь путем работы над наиболее важными и ценными ресурсами и, наоборот, экономии ресурсов на их приобретении – это его главная цель.

Однако при всех плюсах *ABC* анализ является лишь инструментом в руках аналитика и его не следует использовать всегда. Есть и недостатки. Первое, на чем следует заострить внимание, это то, что *ABC* анализ – одномерный метод. Для решения проблемы одномерности *ABC* метода существует два способа. Первый – это создание двухмерной матрицы *ABC* анализа по двум критериям (к примеру, по частоте продаж и ценности продукта).

Второй – это соотношение *ABC* и *XYZ* методов анализа запасов (общие рекомендации по работе с запасами групп *ABC* классификации и выбор подходов к управлению запасами *X, Y, Z*

объединяются для решения общих задач). В обоих случаях, для этого определяют группы сразу по нескольким переменным. К примеру, делаем двухмерную матрицу, используя два показателя. Получаем 9 возможных групп. Максимальное число групп определяется количеством возможных сочетаний групп по двум признакам. Для управления запасами можно использовать сочетание двух букв, а можно заменить эти сочетания общим рейтингом. К тому же, стоит учесть, что среди товара, который продается плохо, может быть и товар который продается в убыток или не продается вовсе. Такой товар следует отнести в группу *D*. Получаем *ABCD* анализ. Так же в группу *A* может войти до 50% ассортимента. В этой ситуации необходимо создать группу *A+*. Так же при проведении анализа может возникать проблема, связанная с качеством, достоверностью и актуальностью. Если продажи имеют закономерность и структуру, то проблем не возникнет, однако если в динамике продаж наблюдается резкое изменение, то *ABC* анализ будет нерелевантен. Необходимо следить, чтобы товар, выпадающий из продаж не вносился в общий рейтинг, так как общий объем продаж будет получаться ниже потенциально возможного.

Подытожим: *ABC* анализ обладает большими преимуществами, которые выражаются в простоте, универсальности и легкой реализации. Из отрицательных моментов нужно отметить в первую очередь то, что *ABC* группировка по одному показателю далеко не всегда корректно расставляет приоритеты. При использовании многомерного *ABC* анализа количество элементов в группе *A* может быть существенно больше 20%. *ABC* анализ не умеет распознавать качество и природу данных, это должен делать аналитик перед проведением расчетов.

*XYZ* анализ позволяет произвести классификацию ресурсов компании в зависимости от характера их потребления и точности прогнозирования изменений в их потребности в течение определенного временного цикла.

Классификации номенклатуры запасов *XYZ*, в отличие от *ABC* метода, никогда не был связан с объективными взаимодействиями качественных и количественных характеристик объектов. Метод *XYZ* основывается на здравом смысле лица, использующего классификацию, хотя и использует однозначный прием разделения номенклатуры на группы. Кроме того, метод *XYZ* не имеет универсального характера *ABC* метода. Метод классификации *XYZ*, наоборот, первоначально был разработан сугубо для предпринимательских целей и был предложен исключительно в целях классификации номенклатуры материальных ресурсов, незавершенного производства и готовой продукции.

Алгоритм проведения классификации по методу *XYZ* состоит из следующих этапов:

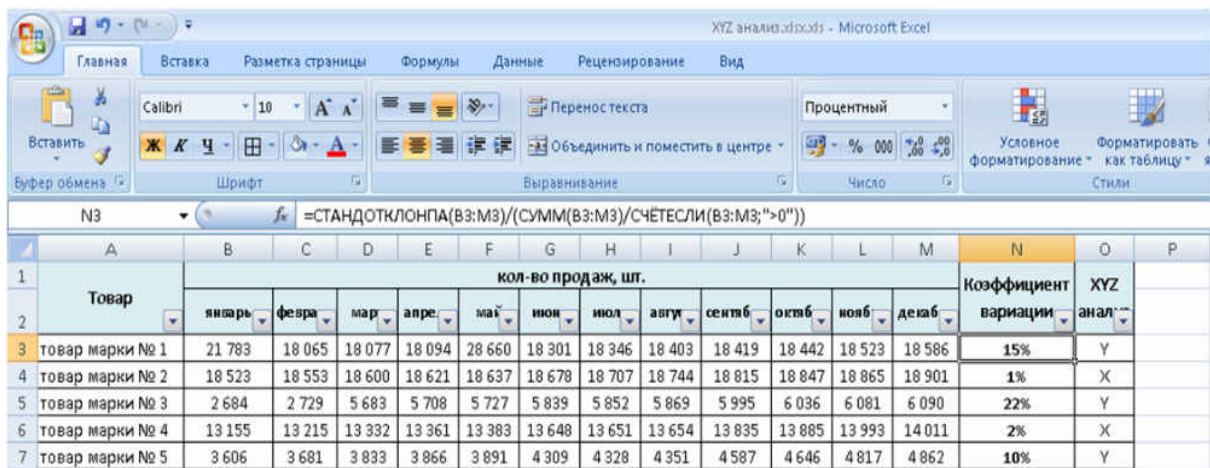
- определение объекта анализа (ассортиментная группа/подгруппа, номенклатура в целом, поставщики, клиенты). Возможна детализация направлений анализа по каналам сбыта, рыночным сегментам;
- определение коэффициентов вариации для анализируемых ресурсов; определение параметра, по которому будет проводиться анализ объекта, – средний товарный запас, руб.; объем продаж, руб.; доход, руб.; количество единиц продаж, шт.; количество заказов, шт., и т. п.;
- группировка ресурсов в соответствии с возрастанием коэффициента вариации;
- распределение по категориям *X*, *Y*, *Z*;
- графическое представление результатов анализа.

Реальное значение коэффициента вариации для разных групп может отличаться по следующим причинам: сезонность продаж, акции, дефицит и т. д.

Расчетные формулы для коэффициента вариации:

$$V = \frac{\sigma}{\bar{x}}, \quad \sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}, \quad \bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

где *V* – коэффициент вариации;  $\sigma$  – среднее квадратичное отклонение;  $\bar{x}$  – средний показатель объема продаж за весь период;  $x_i$  – объем продаж определенного товара за *i* период; *n* – количество анализируемых временных интервалов (например, количество месяцев).



Товар	кол-во продаж, шт.												Коэффициент вариации	XYZ анализ
	январь	февра	марг	апре	маг	июн	июл	авгук	сентб	октб	нояб	декаб		
товар марки № 1	21 783	18 065	18 077	18 094	28 660	18 301	18 346	18 403	18 419	18 442	18 523	18 586	15%	Y
товар марки № 2	18 523	18 553	18 600	18 621	18 637	18 678	18 707	18 744	18 815	18 847	18 865	18 901	1%	X
товар марки № 3	2 684	2 729	5 683	5 708	5 727	5 839	5 852	5 869	5 995	6 036	6 081	6 090	22%	Y
товар марки № 4	13 155	13 215	13 332	13 361	13 383	13 648	13 651	13 654	13 835	13 885	13 993	14 011	2%	X
товар марки № 5	3 606	3 681	3 833	3 866	3 891	4 309	4 328	4 351	4 587	4 646	4 817	4 862	10%	Y

Рис. 2. Расчет и распределение по рейтингам методом XYZ анализа

Итак, как XYZ классификация может быть полезна в практике управления запасами? Главное ее преимущество – возможность однозначного, т. е. объективного лишённого субъективной окраски, выбора верного подхода к управлению запасами конкретной номенклатуры запасов. XYZ анализ позволяет произвести классификацию ресурсов компании в зависимости от характера их потребления и точности прогнозирования изменений в их потребности в течение определенного временного цикла.

Оба метода являются эффективными инструментами анализа запаса предприятия. XYZ анализ дает возможность сделать выводы о том, насколько стабильны продажи, выявить те или иные отклонения. В конечном результате – классифицировать товары, клиентов в зависимости от равномерности продаж или покупок. ABC анализ позволяет нам понять, какой товар пользуется наибольшим спросом, и какой товар следует всегда держать на складе. Однако наиболее полной и эффективной картиной мира будут отличаться двухмерные анализы, приведенные выше. Если объединить результаты этих двух методик, можно получить выводы о том, какие товары (или клиенты) являются не только самыми доходными, но и насколько часто они пользуются спросом.

УДК 004.4'236

**В. Е. Степанов**

студент кафедры аэрокосмических компьютерных и программных систем

**И. Я. Лавровская**

научный руководитель

## СОСТОЯНИЕ ГОНКИ ПРИ МОДЕЛИРОВАНИИ НА SDL

SDL – формальный язык описания и спецификации, основан на математической модели расширенных взаимодействующих конечных автоматов. SDL предназначен для описания структуры и функционирования распределенных систем реального времени [1]. Описание функционирования системы на языке SDL выглядит как последовательность выходящих сигналов от последовательности входящих. Достоинством языка SDL являются заложенные в нем средства структурирования, которые облегчают описание больших и/или сложных систем. Данные средства позволяют разбить систему на отдельные единицы, которые могут быть обработаны независимо [2]. Среда разработки IBM Rational SDL Suite 6.3 предоставляет средства для проведения имитационного моделирования SDL-систем. В рамках работы симулятора IBM Rational SDL Suite

процессы в составе системы выполняют переходы и взаимодействуют посредством отправки сигналов друг другу. Данная среда разработки использует при работе симулятора такие программные механизмы как TimerQueue и ReadyQueue – очереди таймеров и процессов, готовых к выполнению, соответственно. Псевдопараллельность работы модели достигается за счет использования системного времени и дельта-циклов. Дельта-циклы – логическое разбиение момента системного времени для определения последовательности работы системы. За один дельта-цикл может выполняться только один переход для каждого из готовых процессов, другие переходы, запланированные на текущий момент системного времени, будут выполнены в последующие дельта-циклы [3].

*Определение проблемы.* При моделировании сложных и/или больших систем на SDL может возникнуть состояние гонки. Состояние гонки (race condition) – это ситуация, возникающая, когда несколько потоков одновременно обращаются к одному и тому же ресурсу, причем хотя бы один из потоков выполняет операцию записи, и порядок этих обращений точно не определен [4]. Для языка SDL состояние гонки может быть определено как ситуация, в которой нарушается ожидаемый порядок выполнения дельта-циклов при работе процессов с общим ресурсом. Рассмотрим ситуацию, при которой это может произойти. Для этого возьмем абстрактную систему (рис. 1).

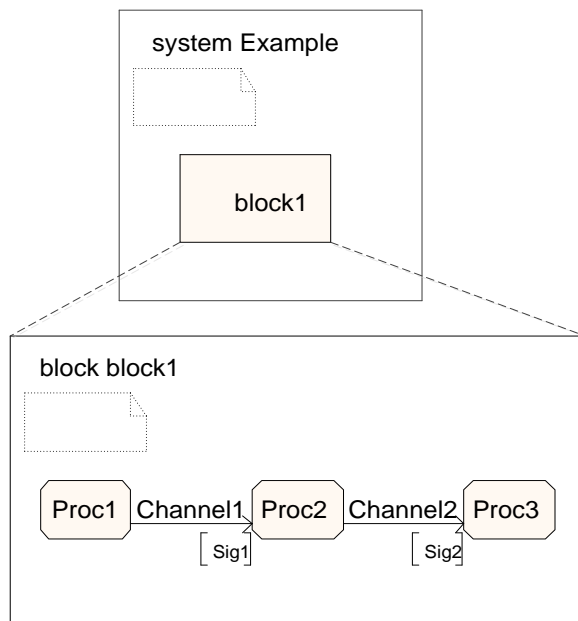


Рис. 1. Абстрактный пример – система

В данной системе в процессе Proc3 есть таймер T1 и счетчик Counter. Общее описание работы процесса Proc3 приведено ниже:

- при истечении таймера T1 увеличивается счетчик Counter и заново запускается таймер;
- при приходе сигнала Sig2 увеличивается счетчик Counter и заново запускается таймер;
- обработка сигнала Sig2 имеет больший приоритет, по сравнению с обработкой истечения таймера T1.

Ожидается, что сначала будет обработано истечение таймера T2, который находится в Proc2, затем будет передан сигнал Sig2, переустановлен таймер T1 и увеличен счетчик Counter (рис. 2).

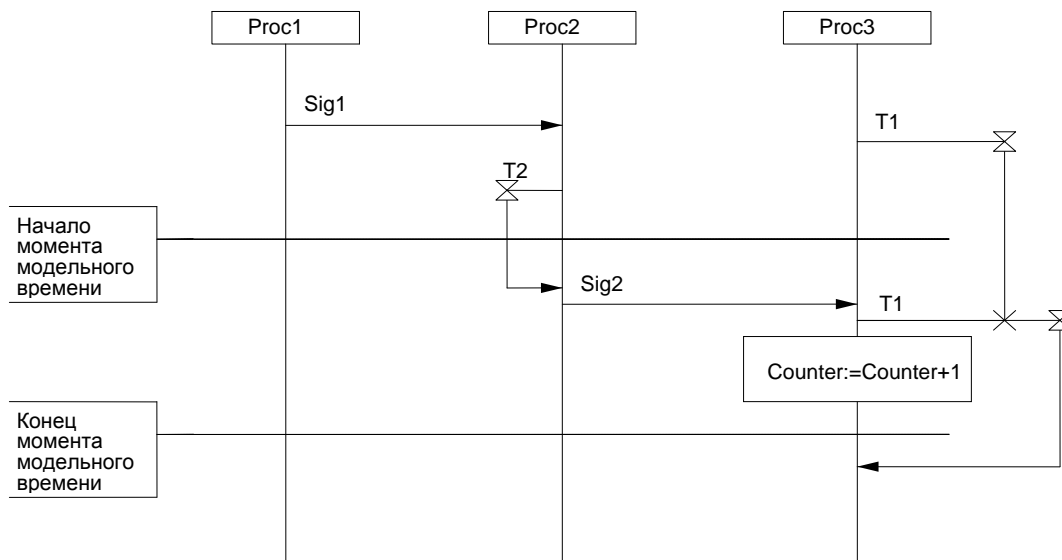


Рис. 2. Ожидаемое поведение системы Example

Однако в данной системе и на подобных фрагментах реальных систем, может возникать состояние гонки (рис. 3).

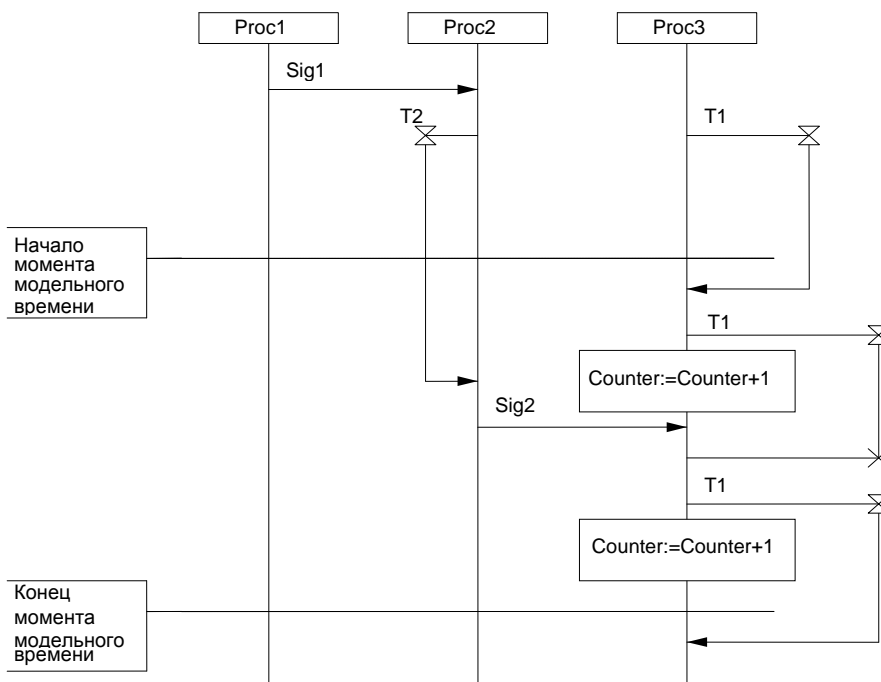


Рис. 3. Состояние гонки в системе Example

Истечение таймера  $T1$  обрабатывается раньше истечения таймера  $T2$ . После истечения таймера  $T1$  счетчик будет увеличен, таймер  $T1$  будет переустановлен, как и должно было быть при истечении таймера  $T1$ . В этот же момент модельного времени, но в другой дельта-цикл, будет обработан таймер  $T2$ , и отправлен сигнал  $Sig2$ . После приема  $Sig2$  счетчик будет увеличен, и переустановлен таймер  $T1$ , что также соответствует правильной работе системы. Однако сра-

бывание двух условий для увеличения счетчика и переустановки таймера в один момент времени не предполагалось, так как обработка сигнала *Sig2* должна была быть приоритетной, и таймер *T2* не должен был истекать.

Состояние гонки возникает из-за того, что нет четких правил, по которым будут обрабатываться таймеры, время истечения которых приходится на один момент модельного времени. Нет правил в стандарте языка SDL – Z.100 и нет правил в IBM Rational SDL Suite, в руководстве пользователя для которого говорится, что порядок обработки таймеров может быть случайным [5].

Рассмотрим некоторые варианты решения. Условно разделим их на варианты решения для тестировщика и для разработчика.

*Решения для тестировщика.* Варианты решения для тестировщика заключаются в том, что поведение модели будет меняться программными средствами во время имитационного моделирования.

Первый способ – исправление очереди готовых к выполнению процессов и/или порядка входящих сигналов в порт процесса (ReadyQueue/InputPort) с помощью средств симулятора в ручном режиме. Суть метода заключается в том, чтобы заменить нежелательную последовательность обработки таймеров/сигналов на желаемую. В этом варианте наиболее вероятно придется проходить места появления возможных проблем по шагам, чтобы не пропустить момент, когда необходимо внести изменения. Данный способ дает наибольший контроль над ситуацией, но также наиболее сложен, в том числе по причине человеческого фактора.

Второй способ – исправление очереди готовых к выполнению процессов и/или порядка входящих сигналов в порт процесса (ReadyQueue/InputPort) с помощью команд точек останова. Встроенные средства симулятора IBM Rational SDL Suite 6.3 позволяют ставить точки останова в процессах и присваивать им команды, в том числе команды, которые могут менять последовательность работы. Этот способ отличается от первого большей автоматизацией, но также требуется большая по сравнению с первым методом подготовительная работа – необходимо определить момент внесения изменений заранее. К дополнительным сложностям относится и то, что при изменении модели старый набор точек останова с командами может стать нерабочим.

Третий способ – решение с помощью скриптов, поддерживаемых симулятором. Скрипт – это последовательность команд, которые в том числе могут:

- подавать сигналы;
- устанавливать таймеры;
- менять последовательность действий;
- выводить значения переменных и прочую информацию.

Скрипты являются следующим витком автоматизации. Результаты, полученные тестами, основанными на скриптах, при неизменности модели, всегда будут одинаковыми. Скрипты имеют большие возможности, чем команды точек останова из второго способа, однако, все проблемы, характерные для второго способа, сохраняются.

*Решения для разработчика.* Варианты решения для разработчика заключаются в изменении модели. Один из вариантов такого изменения – внесение в описание процессов команды #PRIO, которая определена в IBM Rational SDL Suite 6.3 и компиляторе Cmicro [5]. Эта команда не является базовым средством языка SDL, но она позволяет поменять приоритет обработки процессов. Таким образом, можно добиться желаемого поведения системы. Этот вариант решения снимает возникшую неоднозначность. Однако данный способ сложен тем, что во время проектирования не всегда очевидна расстановка приоритетов. Это может повлечь за собой появление состояний гонки в других частях сложной модели, а, следовательно, систему придется заново тестировать.

*Заключение.* В данной статье было рассмотрено появление состояния гонки при моделировании на SDL, и были выявлены предпосылки появления данного состояния, что может помочь избежать появления состояния гонки в будущем. Были предложены варианты решения неоднозначности поведения системы, которые были условно разбиты на две группы: варианты для тестировщика и для разработчика. Варианты для тестировщика могут использоваться как времен-

ное средство от появления состояния гонки – это может быть полезно, когда важно протестировать другие механизмы работы системы, например, работу с памятью или при попытке отладить уровни приоритетов обработки процессов (результаты применения #PRIO). Варианты для разработчика нацелены на постоянное избавление от состояния гонки, и являются единственным долгосрочным решением данной проблемы.

### Библиографический список

1. International Telecommunication Union. Recommendation Z.100. Specification and Description Language (SDL). 2002.
2. Карабегов А. В., Тер-Микаэлян Т. М. Введение в язык SDL. М.: Радио и связь, 1993. 184 с.
3. Olenev V., Lavrovskaya I., Morozkin P., Rabin A., Balandin S., Gillet M. Co-Modeling of Embedded Networks Using SystemC and SDL. From Theory to Practice, Chapter 9, Advancing Embedded Systems and Real-Time Communications with Emerging Technologies, A volume in the Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) Book Series, IGI Global. С. 206–233.
4. Кудрин М. Ю., Прокопенко А. С., Тормасов А. Г. Метод нахождения состояний гонки в потоках, работающих на разделяемой памяти // ТРУДЫ МФТИ. – 2009. – Том 1, № 4. С. 182.
5. IBM Corporation. IBM Rational SDL and TTCN Suite 6.3, IBM Rational SDL and TTCN Suite, 2009.

УДК 004.021

**Е. Ю. Фортышев**

студент кафедры аэрокосмических компьютерных технологий

**К. Н. Храменкова**

научный руководитель

## АЛГОРИТМ ПОДДЕРЖКИ СЕТИ SPACEWIRE ПРИ ДЕЦЕНТРАЛИЗОВАННОМ АЛГОРИТМЕ PLUG-AND-PLAY

Одним из основных направлений при построении и эксплуатации распределенных бортовых систем является автоматизация. Алгоритмы Plug-and-Play позволяют автоматизировать процесс конфигурирования и администрирования бортовых систем в соответствии со стандартом SpaceWire. Одной из задач при разработке децентрализованного алгоритма Plug-and-Play является создание алгоритма поддержки менеджеров сети SpaceWire, при котором выход из строя одного или нескольких сетевых менеджеров не вызовет падения работы бортовой сети. Такая проблема возникает в силу того, что каждый сетевой менеджер администрирует только свою подсеть, и в случае сбоя или поломки менеджера часть сети станет неконтролируемой, что может привести к сбою всей сети и прочим негативным последствиям. На данном этапе работы рассматриваются три варианта решения поставленной задачи, подробное описание которых приводится в данной статье. Это укрупнение регионов, при котором один из соседних менеджеров должен взять на себя поддержку еще одной подсети, дублирование менеджеров, когда в работу включается запасной менеджер на данной подсети, и комбинированный метод, который объединяет первые два варианта.

От выбора метода конфигурирования и администрирования зависит эффективность и удобство работы с распределенной системой [1]. В данной работе администрирование и конфигурирование будет рассматриваться применительно к сети стандарта SpaceWire.

Стандарт SpaceWire – стандарт ECSS, направленный на применение в области управления, разработки и качества продуктов в аэрокосмических проектах. SpaceWire поддерживает интегрирование и тестирование сложных бортовых систем путем внедрения соответствующего оборудования непосредственно в аппаратуру обработки данных. Мониторинг и тестирование могут быть проведены без необходимости создания отдельного физического интерфейса с аппаратурой обработки данных [2].



К основным методам конфигурирования и администрирования коммутаторов можно отнести ручной и автоматизированный. При ручном методе конфигурирование и администрирование выполняется человеком – администратором сети. При автоматизированном методе конфигурирования и администрирования (Plug-and-Play) при включении всех устройств системы эти устройства сами по специальному алгоритму исследуют и определяют топологию распределенной сети, выдают логические адреса узлам, настраивают таблицы маршрутизации и другие необходимые параметры системы, а также снабжают данной информацией все другие устройства сети, которые этого требуют. Во время штатной работы сети подключение и отключение новых устройств также определяется автоматически [3].

Все алгоритмы Plug-and-Play в общем случае можно разделить на два вида:

- централизованные алгоритмы. В системе есть одно устройство, которое иницирует и полностью управляет процессом Plug-and-Play;
- децентрализованные алгоритмы. В системе есть несколько центров, которые отвечают за процесс Plug-and-Play [4].

При централизованном алгоритме в сети существует один сетевой менеджер, который исследует сеть и способен автоматически восстанавливать режимы работы других устройств в сети, записать необходимые значения регистров. Данный алгоритм имеет существенный недостаток: в силу того, что сетевой менеджер только один, то при какой-либо его неисправности сеть становится неконтролируемой, что непременно приведет к негативным последствиям.

Децентрализованный алгоритм позволяет избежать изложенного недостатка, поскольку сетевых менеджеров может быть несколько. Каждый из них осуществляет конфигурирование, администрирование и мониторинг сети, то есть при выходе из строя одного или нескольких менеджеров система останется работоспособной.

В сети существуют несколько сетевых менеджеров, которые и осуществляют изучение, конфигурирование и дальнейшее администрирование сети, причем каждый менеджер администрирует только свою подсеть (рис. 1). Менеджерами могут быть как некоторые терминальные узлы, так и персональные компьютеры. В памяти каждого менеджера хранятся переменные, необходимые для его корректной работы: логические адреса узлов, пути к коммутаторам и прочее.

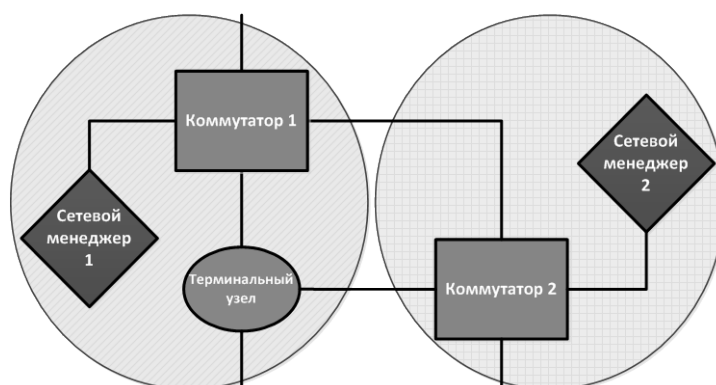


Рис. 1. Пример сети SpaceWire при децентрализованном алгоритме Plug-and-Play

Рассматриваемый в данной статье алгоритм поддержки является неотъемлемой частью разрабатываемого децентрализованного алгоритма. На данном этапе работы предполагается три варианта решения поставленной задачи: один из соседних менеджеров должен «подхватить» еще одну подсеть (укрупнение регионов), либо необходимо решить эту задачу введением в работу запасного менеджера на данной подсети (дублирование менеджеров), либо объединить первые два варианта (комбинированный метод).

Укрупнение регионов. В случае отсутствия ответного статусного сообщения от одного менеджера, соседний менеджер должен взять на себя подсеть вышедшего из строя менеджера, а также оповестить других менеджеров сети. Стоит учесть, что менеджеры обмениваются статусными сообщениями не хаотично, а последовательно и в определенном порядке. Подсеть вышедшего из строя менеджера переходит под контроль соседнему менеджеру (рис. 2), определенному через голосование.

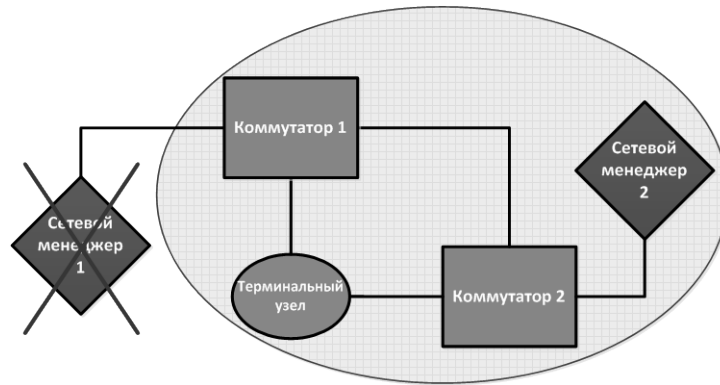


Рис. 2. Укрупнение региона при вышедшем из строя сетевом менеджере 1

Дублирование менеджеров. У каждого менеджера сети существует дублирующий менеджер, который через определенный промежуток времени получает данные от основного устройства. Если дубликат спустя заданное время не получает пакет от него, то он вступает в работу на данной подсети вместо основного менеджера. Дубликат имеет такой же логический и регионально-логический адрес, как и основной менеджер. Основной и дублирующий менеджеры параллельно подключены к маршрутизатору, который принимает решение о том, на какой менеджер переадресовывать пакет.

В теории надежности существует понятие последовательного соединения, при котором отказ в работе хотя бы одного элемента приводит к отказу аппаратуры в целом. Этот вид соединения является основным, наиболее характерным и распространенным для большинства типов радиоэлектронной аппаратуры. Параллельным или резервным соединением элементов в теории надежности считается такое соединение, при котором отказ аппаратуры наступает только в случае выхода из строя всех параллельно (резервно) включенных элементов. Следовательно, резервированием называется способ повышения надежности аппаратуры путем включения в схему ряда параллельных, резервных элементов [5].

Надежность работы двух параллельно включенных менеджеров сети, т. е. вероятность того, что, по крайней мере, один из них будет работать, равна сумме вероятностей трех возможных благоприятных исходов:

- ни основной менеджер, ни дублирующий менеджер не выйдут из строя (рис. 3, а);
- основной менеджер выйдет из строя, но дублирующий менеджер будет работать (рис. 3, б);
- дублирующий менеджер выйдет из строя, но основной менеджер будет работать (рис. 3, в).

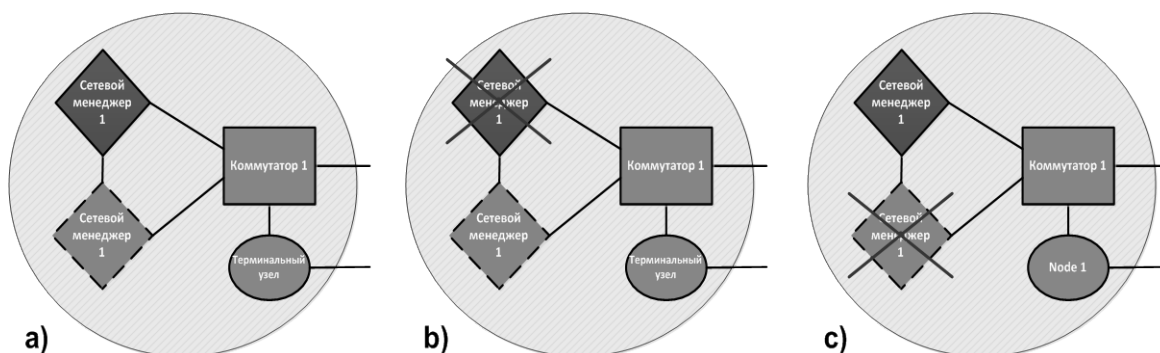


Рис. 3. Возможные ситуации, при которых подсеть менеджера 1 будет работоспособна

Данный метод позволит поддерживать работоспособность всей сети, избежать конфликтных ситуаций и арбитража между соседними менеджерами сети.

Комбинированный метод. Данный подход совмещает в себе оба предыдущих метода – метод укрупнения регионов и дублирования менеджеров. При выходе из строя основного менеджера его подсеть «подхватывает» дубликат (рис. 4, а). При выходе из строя основного и дублирующего менеджера подсеть переходит под контроль соседнему менеджеру сети (рис. 4, б), либо равномерно разделяется между несколькими соседними менеджерами.

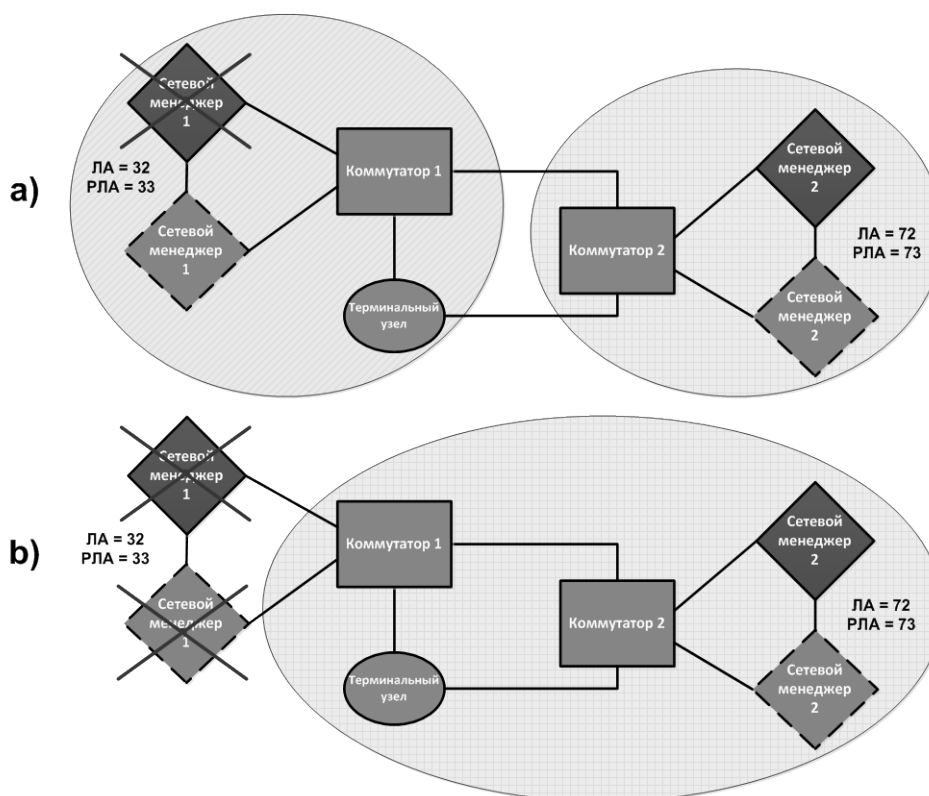


Рис. 4. Пример сети SpaceWire при комбинированном методе

Комбинированный метод имеет очевидное преимущество по сравнению с другими двумя методами, так как он сочетает в себе надежность первых двух методов и позволит сохранять управление сетью максимально долгое время.

Использование децентрализованного алгоритма повышает надежность работы сети, ее настройку и отслеживание состояния устройств. Но с другой стороны, это накладывает дополни-

тельные требования к сети и самим менеджерам. В силу своей сложности, децентрализованный алгоритм был условно разделен на задачи, которые требуют дальнейшего глубокого и независимого друг от друга изучения. Одной из таких решаемых задач является создание алгоритма поддержки менеджеров сети, на данном этапе рассматривается три варианта решения проблемы при выходе из строя одного или нескольких менеджеров. Очевидно, что из представленных в статье вариантов самым надежным вариантом реализации алгоритма поддержки является комбинированный метод. Однако он предъявляет серьезные требования к памяти, самим менеджерам и прочим затратам. Дублирование менеджеров позволяет увеличить надежность сети и избежать конфликтных ситуаций между соседними менеджерами, однако для построения такой сети необходимо в два раза больше менеджеров, что не всегда возможно реализовать на бортовой сети. Укрупнение регионов является альтернативным вариантом дублированию, однако, в силу ограниченности памяти менеджеров существуют ограничения по максимальному числу устройств, которые могут обслуживаться одним менеджером.

### Библиографический список

1. Фортыхев Е. Ю. Различные типы конфигурирования и администрирования сети SPACEWIRE, Шестьдесят седьмая студенческая научно-техническая конференция ГУАП, СПб ГУАП, 2014.
2. ESA (European Space Agency), standard ECSS-E-50-12A, Space engineering. SpaceWire – Links, nodes, routers and networks. European cooperation for space standardization, ESA Publications Division ESTEC, Noordwijk, The Netherlands, 2003.
3. Храменкова К. Н., Коблякова Л. В. Администрирование и конфигурирование распределенных систем по стандарту SPACEWIRE, Шестьдесят четвертая студенческая научно-техническая конференция ГУАП, СПб ГУАП, 2011.
4. Онищенко Л. В. Методы администрирования распределенных систем SpaceWire, Научная сессия ГУАП, 2008.
5. Сорин Я. М. Надежность радиоэлектронной аппаратуры. М.: 2012. 74 с.

УДК 004.032.26:629.734.33

**Д. О. Ягубов**

магистрант кафедры аэрокосмических измерительно-вычислительных комплексов

**В. Я. Мамаев**

канд. техн. наук, доц. – научный руководитель

## ПРИМЕНЕНИЕ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ В ЗАДАЧЕ УПРАВЛЕНИЯ ЛЕГКИМ МОТОПЛАНЕРОМ

В последнее время наблюдается все более широкое применение легких мотопланеров для решения задач мониторинга окружающей среды. Среди них можно назвать такие задачи, как наблюдение за очагами пожаров и динамики наводнений, за объектами с повышенным радиационным фоном и т. д. Однако использование легких мотопланеров ограничивается небольшой продолжительностью полета, как правило, не превышающей двух часов. Последнее связано с использованием маломощных источников энергии (аккумуляторных батарей, питающих электродвигатели). Следовательно, актуальной является задача использования на борту дополнительной энергии из окружающей среды. Так, для увеличения продолжительности полета можно использовать энергию восходящих воздушных потоков (термиков), что уже давно применяется в планерном спорте.

Для эффективного использования энергии термика необходимо, чтобы летательный аппарат (ЛА) находился ближе к его центру, где наблюдается максимальная скорость восходящего потока. При этом расход энергии источника существенно замедляется. Предпочтительной траекторией парения является движение по спирали (рис. 1).

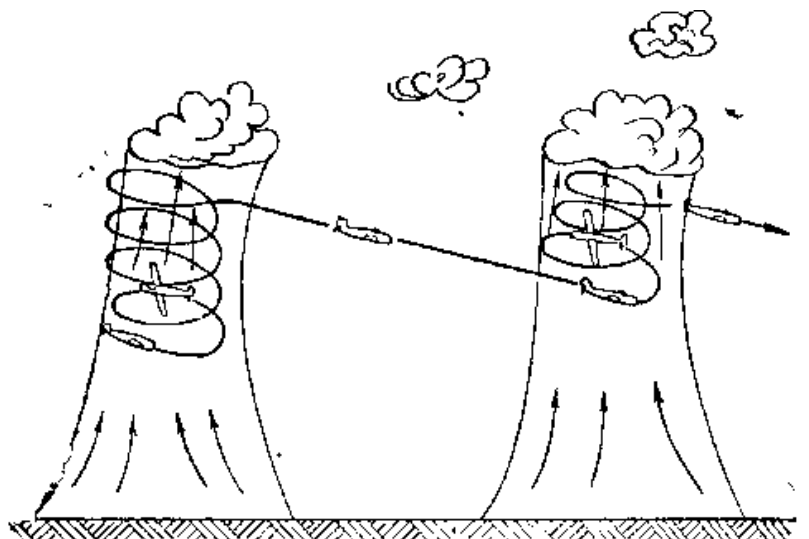


Рис. 1. Парение ЛА внутри термика

Чтобы реализовать в бортовом компьютере алгоритм движения по спирали, необходимо знать положение центра термика относительно ЛА, а так же его радиус. Исследованию возможности определения радиуса термика с помощью искусственной нейронной сети (ИНС) на борту ЛА посвящена данная статья.

Для решения данной задачи можно использовать два подхода: детерминированный и недетерминированный. Начнем рассмотрение с детерминированного подхода.

В работе [1] описана математическая модель («Updraft Model») конвективной струи, на вход которой поступают такие параметры, как высота конденсации облаков ( $Z_i$ , м), высота сечения термика ( $H$ , м) а так же некоторый безразмерный «коэффициент усиления». На выходе модели – требуемый радиус термика, а так же распределение вертикальных скоростей внутри его сечения на заданной высоте.

Модель сама по себе достаточно сложна и содержит множество математических зависимостей. Несколько основных зависимостей приведено ниже:

$$w_r = w \cdot \left( 1 - \frac{w_e}{w_{\text{ц}}} \right) + w_{\text{ц}}, \quad (1)$$

$$w_e = \frac{-\bar{w} \cdot N \cdot \pi \cdot r_{\text{в}}^2 \cdot \frac{w_{\text{н.п.}}}{K_{\text{н.п.}}}}{L_x \cdot L_y - N \cdot \pi \cdot r_{\text{в}}^2}, \quad (2)$$

$$K_{\text{н.п.}} = \begin{cases} \frac{-\pi}{6} \sin\left(\frac{\pi \cdot r}{r_{\text{в}}}\right), & \text{если } r_{\text{ц}} < r < r_{\text{в}}, \\ 0, & \text{иначе.} \end{cases} \quad (3)$$

где  $w_r$  – скорость восходящего потока на расстоянии  $r$  от центра термика с учетом движения окружающих воздушных масс;  $w_e$  – вертикальная скорость воздушной массы окружающей среды, в которой рассматривается движение;  $w_{\text{ц}}$  – скорость восходящего потока в центре термика (максимальная скорость восходящего потока);  $w$  – скорость восходящего потока на расстоянии  $r$  от центра термика;  $w_{\text{н.п.}}$  – вертикальная скорость нисходящего потока;  $\bar{w}$  – средняя вертикальная скорость термика;  $K_{\text{н.п.}}$  – весовой коэффициент нисходящего потока;  $r_{\text{в}}$  – внешний радиус восходящего потока;  $r_{\text{ц}}$  – радиус центральной части восходящего потока.

Казалось бы, эту модель можно использовать на борту и с ее помощью вычислять радиус термика, однако существует ряд проблем, препятствующих прямому ее использованию. Для вычисления радиуса модель использует массив безразмерных коэффициентов, каждый из которых содержит в себе информацию о физических условиях образования термика, таких, как химический состав атмосферы, температура воздуха, влажность, давление, количество твердых частиц в атмосфере и т. д. При этом математические закономерности, связывающие эти коэффициенты с физическими величинами, не известны и подбирались разработчиками модели эмпирически. Очевидно, что на борту ни один из этих коэффициентов известен не будет, поэтому использование данной модели затруднено.

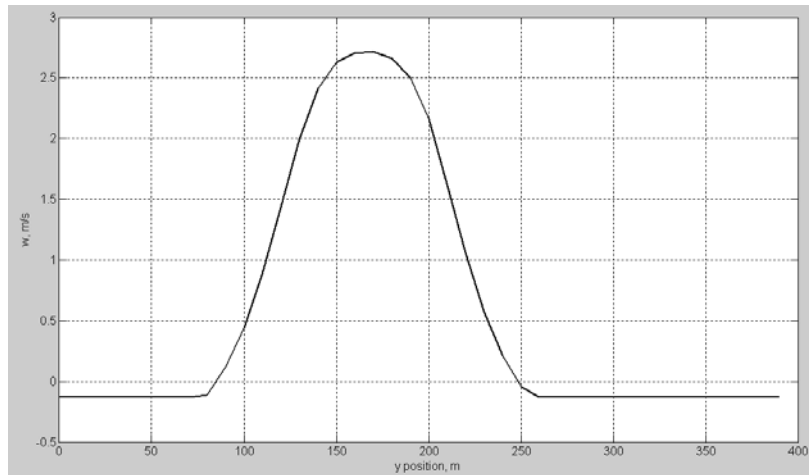


Рис. 2. Зависимость вертикальной скорости от положения относительно центра термика

На рис. 2 приведен один из вариантов распределения вертикальных скоростей в сечении термика. Следует отметить, что изменение коэффициентов, отвечающих за условия его образования, входных параметров, а так же траектории движения ЛА относительно центра, приводит к существенному изменению вида распределения.

Как видно, детерминированный подход для решения поставленной задачи не годится. Рассмотрим теперь адаптивный (недетерминированный) подход. В качестве такового рассмотрим использование ИНС [2, 3], а именно ИНС прямого распространения с активационной функцией типа «биполярный сигмоид» [4]. Данная топология выбрана исходя из следующих соображений:

- отсутствуют обратные связи. В нейронных сетях обратные связи реализуют временную память. Она используется в задачах нейроуправления, моделирования динамических структур. Для задачи распознавания образов обратные связи не требуются. Более того, в сетях прямого распространения не требуется решать задачу устойчивости выхода сети – он сходится к устойчивому состоянию по умолчанию;

- существует изученный и проверенный на практике алгоритм обучения;

- данная топология сети является классической и обладает самой большой доказательной базой, что упрощает ее использование в инженерных целях.

На рис. 3 приведена топология многослойной ИНС прямого распространения. На входной («нулевой») слой подается вектор входных параметров, далее он распространяется от входного слоя к выходному, последовательно, слой за слоем. С выходного слоя нейронов снимается результат работы сети (радиус термика). Для обучения сети используется алгоритм обратного распространения ошибки.

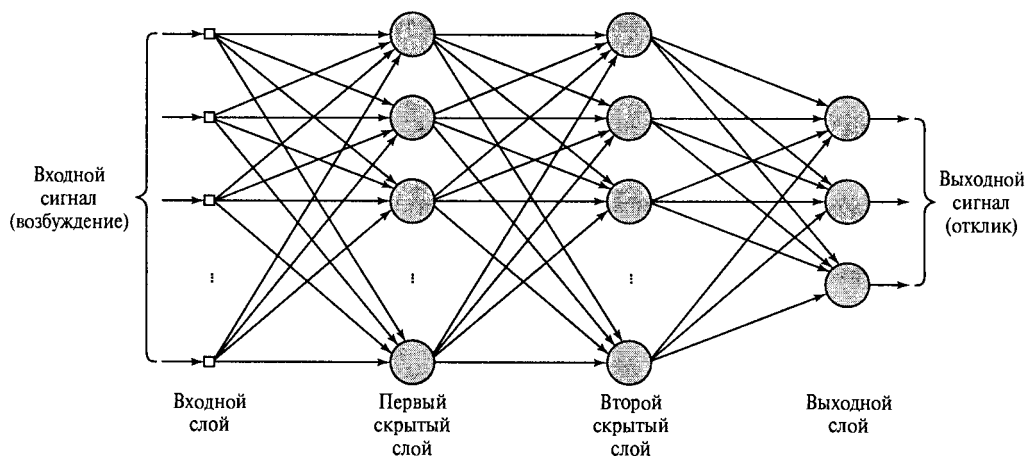


Рис. 3. Искусственная нейронная сеть прямого распространения

Эксперимент № 1.

На основе приведенной выше модели была сгенерирована обучающая выборка, состоящая из набора обучающих пар (входного вектора и желаемого выходного вектора). Входной вектор содержит следующие величины:  $Z_i$  (может быть получена из метеосводки);  $H$  – высота сечения термика (совпадает с высотой полета ЛА и может быть получена из показаний бортового высотомера); распределение вертикальных скоростей.

На последнем пункте следует остановиться подробнее. Распределение вертикальных скоростей состоит из большого массива (1000 значений) вертикальных скоростей. Такое количество входных значений нежелательно подавать на вход ИНС. Это связано не только с увеличением количества нейронов, необходимых для вычисления, но и с ограниченностью разрядной сетки ЭВМ, что затрудняет обучение сети методом градиентного спуска (обратного распространения). Поэтому, встает вопрос о сжатии данных. Для решения этой проблемы в данной работе была применена аппроксимация массива вертикальных скоростей полиномом. На вход ИНС были поданы коэффициенты аппроксимации, содержащие в себе информацию о всем распределении. При этом, самих коэффициентов значительно меньше, чем элементов первоначального вектора.

При этом траектория движения ЛА проходит только через центр термика, а коэффициенты, отвечающие за условия создания термика, остаются неизменными. Меняется только входной вектор.

Для оценки качества работы системы используются следующие формулы:

$$S_n = \sqrt{\frac{\sum_{i=1}^n (X_{ист} - X_{вых})^2}{n - 1}} \quad (4)$$

$$\varepsilon = \frac{\sum_{i=1}^n (X_{ист} - X_{вых})}{n} \quad (5)$$

Здесь:  $S_n$  – среднеквадратическая ошибка выхода ИНС;  $\varepsilon$  – усредненная ошибка выхода нейронной сети (необходима для оценки регулярной составляющей);  $X_{ист}$  – истинное значение измеряемой величины (радиуса термика);  $X_{вых}$  – значение выхода нейронной сети;  $n$  – количество элементов в тестовой выборке. Выражение (4) используется для оценки флуктуационной составляющей ошибки, а выражение (5) – для оценки регулярной составляющей.

Результаты эксперимента приведены на рис. 4.

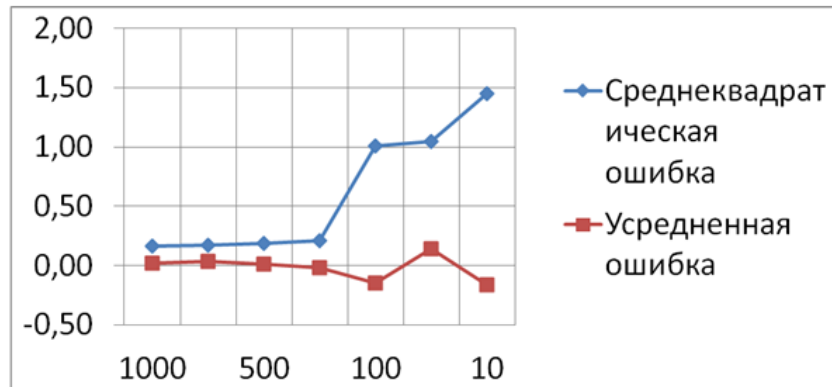


Рис. 4. Результаты эксперимента № 1 в виде зависимости ошибки от размера обучающей выборки

Из рис. 4 видно, что сеть сама пытается устранить регулярную составляющую ошибки. Флуктуационная составляющая ошибки возрастает по мере уменьшения количества элементов в обучающей выборке. При размере обучающей выборки больше 200 элементов ошибка мала и при дальнейшем увеличении размера выборки уменьшения ошибки практически не происходит.

#### Эксперимент № 2.

Условия проведения эксперимента № 2 похожи на условия проведения эксперимента № 1, отличие заключается в том, что ЛА теперь движется не только вдоль центра термика, а вдоль произвольной хорды его сечения.

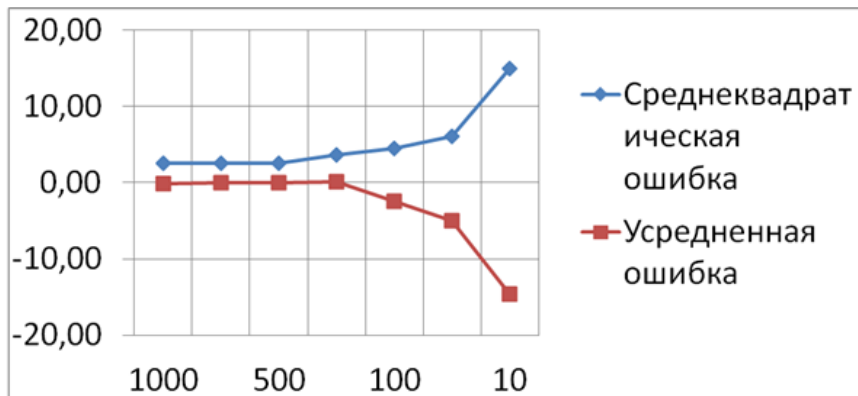


Рис. 5. Результаты эксперимента № 2 в виде зависимости ошибки от размера обучающей выборки.

Результаты эксперимента № 2 сходны с результатами эксперимента № 1, однако численное значение флуктуационной составляющей ошибки больше. Это связано с уменьшением уровня априорной информации о местоположении ЛА относительно центра термика. Следовательно, ошибка определения радиуса увеличилась.

#### Эксперимент № 3.

Условия проведения эксперимента похожи на условия проведения эксперимента № 1, отличие заключается в том, что ЛА теперь движется не только вдоль центра термика, а вдоль произвольной хорды сечения термика, а так же меняются физические условия его образования.



Результаты эксперимента (рис. 6) сходны с результатами эксперимента № 2, однако численное значение флуктуационной составляющей ошибки больше. Это связано с уменьшением уровня априорной информации о форме распределения скоростей. Следовательно, ошибка определения радиуса увеличилась.

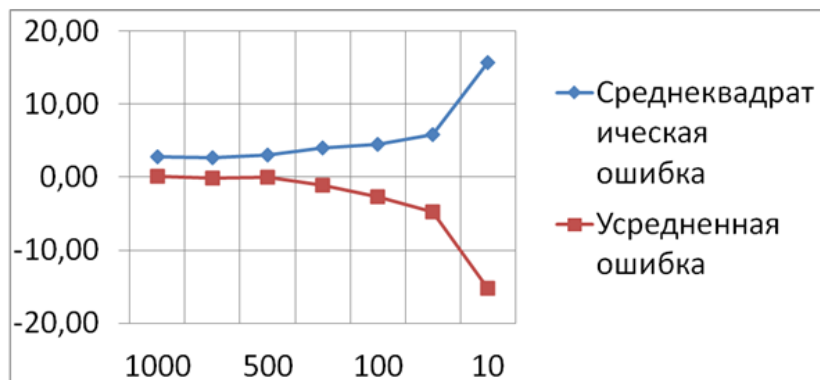


Рис. 6. Результаты эксперимента № 3 в виде зависимости ошибок от размера обучающей выборки

#### Эксперимент № 4.

В последнем эксперименте, помимо изменения всех предыдущих параметров математической модели, на вход ИНС вместо коэффициентов аппроксимации полиномом распределения вертикальных скоростей подается несколько первых значений самих скоростей. Это связано с тем, что на практике, чтобы получить коэффициенты аппроксимации, необходимо пролететь через весь термик и измерить вертикальные скорости. Но к этому моменту применять алгоритм парения будет уже поздно, т. к. ЛА уже его пролетит. Что бы этого не случилось, необходимо заблаговременно определить радиус.

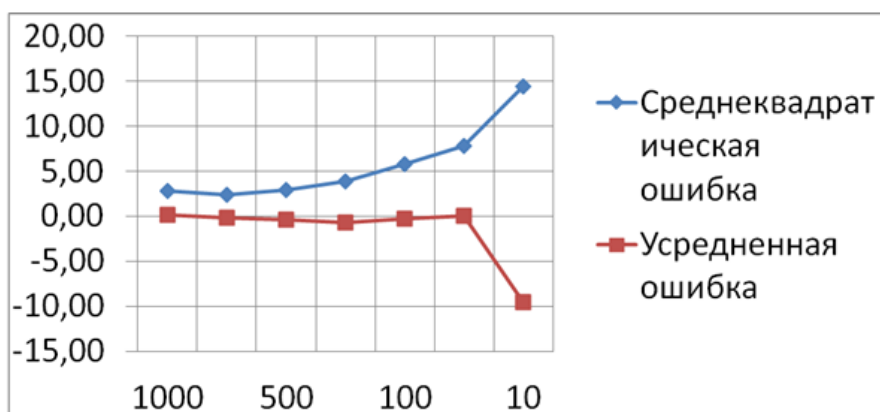


Рис. 7. Результаты эксперимента № 4 в виде зависимости ошибки от размера обучающей выборки

Результаты эксперимента сходны с результатами эксперимента № 3, однако численное значение флуктуационной составляющей ошибки больше. Это связано с уменьшением уровня априорной информации о виде распределения скоростей в целом, заложенном в коэффициентах

аппроксимации. Используется лишь часть этого распределения, что естественно увеличивает ошибку.

Выводы.

Сравнение ошибок в определении величины радиуса термика в зависимости от условий проведения эксперимента представлено на рис. 8.

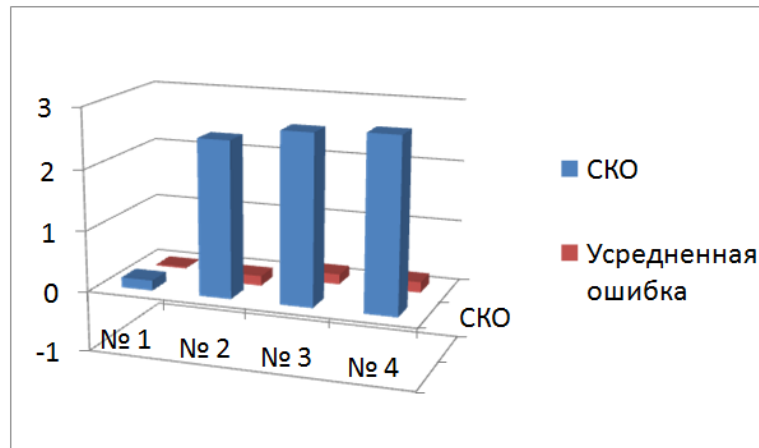


Рис. 8. Значения ошибок в зависимости от условий проведения исследований

Как видно из рис. 8, сеть пытается устранить регулярную составляющую ошибки за счет обучения. Флуктуационная составляющая ошибки увеличивается с уменьшением уровня априорной информации. Самое интенсивное скачкообразное увеличение ошибки происходит, когда ЛА движется не только через центр термика, а по произвольной хорде. В условиях, самых приближенных к реальным (последнее исследование) при достаточном размере обучающей выборки можно достичь среднеквадратической ошибки, равной 2,81 м. При этом радиусы, генерируемые моделью, лежат в диапазоне от 2 до 120 м.

### Библиографический список

1. Allen M. J., Updraft model for development of autonomous soaring uninhabited air vehicles // AIAA 2006–1510, 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 9–12 January, 2006.
2. Осипов Л. А., Яковлев С. А. Искусственный интеллект и нейронные сети: учеб. пособие. – СПб.: ГУАП, 2011. – 134 с.
3. Бураков М. В. Нейронные сети и нейроконтроллеры: учеб. пособие – СПб ГУАП, 2013. – 284 с.
4. Кричевский М. Л. Интеллектуальный анализ данных в менеджменте: учеб. пособие – СПб ГУАП, 2005. – 208 с.: ил.

УДК 621.396.6

**К. Р. Афанасьева**

студент кафедры конструирования и технологии электронных и лазерных средств

**В. П. Ларин**

д-р техн. наук, проф. – научный руководитель

## АНАЛИЗ НАПРАВЛЕНИЙ РАЗВИТИЯ MID-ТЕХНОЛОГИЙ

Что же такое 3D-MID? Это интересный вид печатной платы, представляющий объемное основание из литого высокотемпературного термопласта, на котором выполнены 3D проводники и контактные площадки. О таком изделии можно сказать много положительного. Главным образом, оно обеспечивает высокую гибкость проектирования: возможности относительно формы и размеров устройства достаточно широки; возможна интеграция электронных, механических и оптических элементов. Также оно позволяет упростить конструкцию, уменьшить материалоемкость, является более экологичным по сравнению с обычными платами, что отвечает современным требованиям. Внешний вид конструкций изделий представлен на рис. 1.

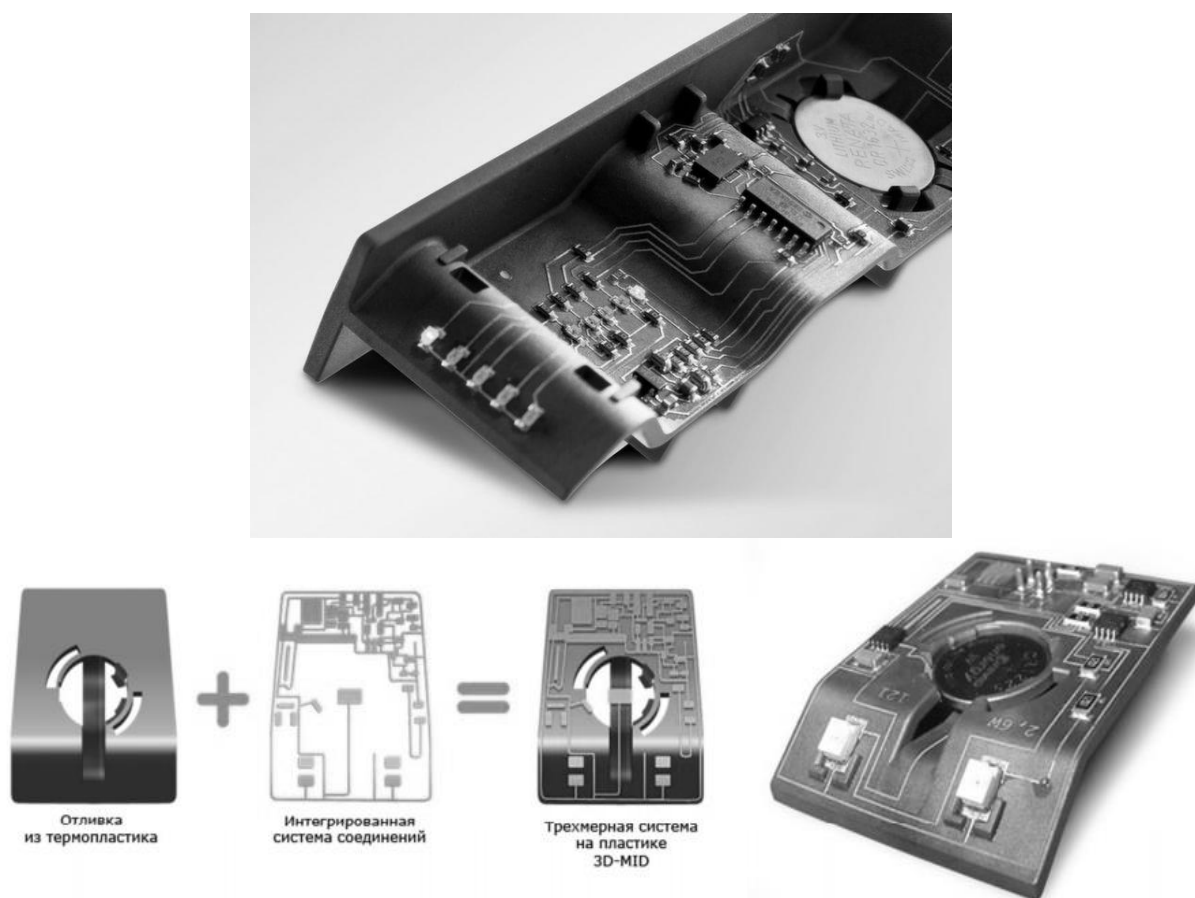











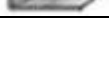
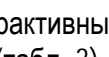
Рис. 1. Внешний вид конструкций в 3D-MID исполнении

Технологический процесс изготовления трехмерных литых электрических плат можно представить следующим образом: литье основания, активация отлитой заготовки, избирательная металлизация поверхности для получения системы проводников. Материалы для основания используются самые различные: жидкокристаллический полимер, полиамид, полибутилентерефталат, полиэтилентерефталат, полифталамид, поликарбонат. Выбор материала, конечно же, зависит от условий эксплуатации устройств, от требований к ним. Исходя из этого, рассматриваются механические и электрические свойства материала, степень влагопоглощения, паяемость, коэффициент термического расширения, который характеризует изменение объема или линейных размеров тела с увеличением температуры, обрабатываемость на разных этапах процесса, пригодность к литью и металлизации, цена.

Процесс изготовления основания конструкции с использованием метода однокомпонентного литья состоит из операций, приведенных в табл. 1.

Таблица 1



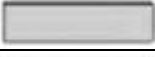
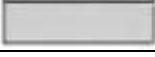
Операции технологического процесса литья основания

№ операции	Наименование операции	Схема операции
1	Литье под давлением	
2	Активация поверхности	
3	Химическое осаждение меди	
4	Нанесение фоторезиста	
5	Экспонирование через 3D-фотошаблон	
6	Проявление фоторезиста	
7	Гальваническое осаждение меди	
8	Нанесение металлорезиста (Su)	
9	Удаление фоторезиста	
10	Травление меди	
11	Нанесение финишного покрытия	

Печатные проводники на литом основании могут быть получены субтрактивным лазерным структурированием (табл. 2) или аддитивным лазерным структурированием (табл. 3).

Таблица 2

Операции получения проводниковой структуры субтрактивным методом

№ операции	Наименование операции	Схема операции
1	Литье под давлением	
2	Активация поверхности	
3	Химическое осаждение меди	
4	Гальваническое осаждение меди	








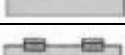

5	Гальваническое осаждение металлорезиста	
6	Лазерное структурирование	
7	Травление меди	
8	Нанесение финишного покрытия	

Таблица 3

## Операции получения проводниковой структуры аддитивным методом

№ операции	Наименование операции	Схема операции
1	Литье под давлением	
2	Активация поверхности лазером	
3	Химическое осаждение меди	
4	Гальваническое осаждение меди	
5	Нанесение финишного покрытия	

В данной технологии применяются термопласты, в состав которых входит активируемый лазером металлоорганический комплекс. Участки отливки, на которых должен быть образован проводящий рисунок, обрабатываются лазером, при этом происходит разрушение связей между атомами металла и другими атомами комплекса. Соответственно, при химической металлизации медь осаждается только на участки поверхности, активированные лазером. Процесс прямого лазерного структурирования (LDS-процесс) проиллюстрирован на рис. 2. Этот процесс используется для небольших объемов производства и большой номенклатуре изделий.

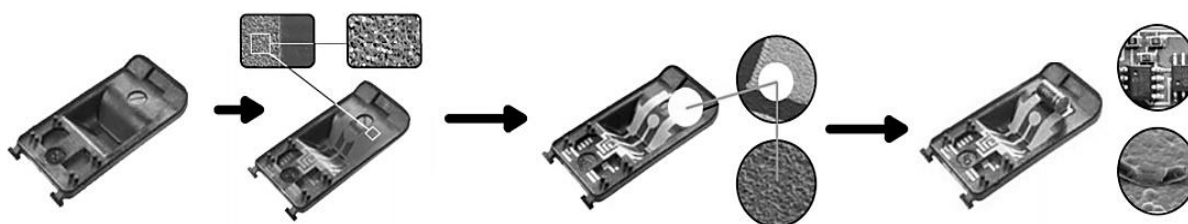







Рис. 2. Процесс получения проводников с использованием лазерного структурирования

Альтернативным процессом изготовления основания является процесс с использованием двухкомпонентного литья основания (табл. 4): Сначала создается заготовка из термопласта, затем производится активация поверхности заготовки для избирательной металлизации. Далее участки поверхности, на которых не должно быть металлизации, покрываются слоем второго термопласта, после чего на открытые участки первого термопласта производится осаждение меди.

Операции технологического процесса двухкомпонентного литья основания

№ операции	Наименование операции	Схема операции
1	Литье под давлением (первый термопласт, возможна металлизация)	
2	Активация поверхности	
3	Литье под давлением (второй термопласт, невозможна металлизация)	
4	Химическое осаждение	
5	Нанесение финишного покрытия	

Процесс двухстадийной заливки (2S-процесс) рекомендуется для массового производства (> 1 млн изделий в год), возможно нанесение покрытий на большие площади и использование сложных конструкций оснований. Этот процесс проиллюстрирован на рис. 3.



Рис. 3. Системы проводников на конструкциях изделий сложной формы

Преимущества и недостатки рассмотренных процессов отражены в табл. 5.

Сравнительная характеристика методов изготовления основания

LDS-процесс	2S-процесс
Плюсы	
Простые заливочные формы	Высокая компактность конструкции за счет сложности формы
Легкое изменение модели конструкции	Повышенная прочность проводников, образованных в углублении
Возможны «тонкие» структуры	Высокая производительность процесса
Минусы	
Процесс многоступенчатый (активация/очистка и т. д.)	Сложные заливочные формы
Возможно наличие мест, куда не может попасть лазер для активации поверхности и, соответственно, осаждения проводящего слоя	Длительность разработки процесса

Что касается сборочных операций, то они аналогичны операциям традиционного поверхностного монтажа: дозирование паяльной пасты/клея, установка компонентов, пайка оплавлением. Отличие в повышенных требованиях к процессу установки компонентов. Основная проблема заключается в необходимости установки компонентов на криволинейные поверхности либо поверхности, располагающиеся под углом, отличным от угла в  $90^\circ$  между осью Z сборочной головки и плоскостью расположения установленного компонента.

В заключение отметим, что область применения 3D-MID очень велика: автомобильная индустрия, медтехника, платежные системы, телекоммуникации. Переключатели и соединители, датчики и приводы, антенны, светотехника, модули камеры – все эти элементы изготавливаются по 3D-MID технологии и используются повсеместно. Более того, область применения данной технологии неуклонно расширяется, совершенствуются термопласты и технологические процессы, так что будущее у 3D-MID определено.

### Библиографический список

1. Нисан А. 3D-MID: области применения и технологии производства// Поверхностный монтаж. 2011. № 3. С. 10–13.
2. Бонапартов С. В. Технология 3D-MID. Процесс, возможности, области применения / ЗАО «НИИИТ» / ГК Остек, 2014.
3. Руководство по разработке трехмерных систем на пластике// Решения для трехмерных схем на пластике/ Группа компаний Остек, 2013/14.
4. Трехмерные схемы на пластике// Решения для трехмерных схем на пластике/ ЗАО «НИИИТ» / ГК Остек, 2014/15.

УДК 608.2

**Р. Р. Булякулов, С. С. Голубев**

студенты кафедры бортовой радиоэлектронной аппаратуры

## ПОРТАТИВНОЕ УСТРОЙСТВО ДЛЯ ЛЕЧЕНИЯ И ПРОФИЛАКТИКИ ЗАБОЛЕВАНИЙ ОПОРНО-ДВИГАТЕЛЬНОЙ СИСТЕМЫ ЧЕЛОВЕКА

Область прикладной медицины – одно из главных и приоритетных направлений современности. В сложившейся экологической ситуации, процент детей, рождающихся с отклонениями в развитии, неуклонно растет. Задачи, стоящие сегодня перед инженерами, прежде всего – тесное сотрудничество с врачами, для создания устройств, которые будут помогать таким детям [1].

В статье рассматривается разработка устройства, задачи которого направлены на коррекцию отклонений, вызванных заболеваниями опорно-двигательной системы. В результате долгих исследований в области медицины [2], были установлены участки тела (например, в области между лопаток), в которых кожа и подкожные вещества скапливаются, образуя рубцы. В результате у таких детей, нарушается кровообращение. Одно из самых опасных мест, куда кровь может перестать поступать, и что может оказать сильное влияние на организм – спинной мозг.

В целом, необходимый в этом случае массаж можно выполнить с помощью устройства, совершающего механические колебания по определенному алгоритму. По заказу ООО «Элок», такое устройство было разработано в портативном варианте. Устройства такого типа присутствуют на российском рынке, но имеют большие габариты и малый срок работы от батареи. По задумке, разрабатываемый аппарат встраивается в специальный кармашек ранца или рюкзака и надевается на спину рис. 1. В момент надевания автоматически активизируется устройство, в основе которого лежит алгоритм, включающий в себя несколько циклов изменения, длительности, амплитуды и частоты механических колебаний. Так как разработчики ориентируются на детей-потребителей в возрасте от 7 до 11 лет, необходимо учитывать, что самостоятельно вклю-

чать и выключать данное устройство они будут попросту забывать, в связи с чем был придуман специальный датчик, работающий по принципу емкостного реле, которое включало и включало устройство в зависимости от надевания или снятия элемента одежды, рюкзака или ранца, в которое оно встроено.

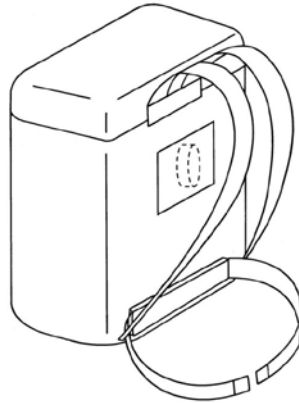


Рис. 1. Схематичное расположение устройства закрепленного на ранце (изображено пунктиром)

Внешне конструкция аппарата очень проста и представляет собой диск диаметром в 150 мм, толщиной в 15 мм и весом 250 г, на внешней части которого располагается небольшой разъем для подзарядки устройства.

Режимов работы у аппарата два. В режиме ожидания устройство потребляет 50 мкА тока. При поднесении к пластине емкостного реле к телу общая емкость системы меняется, что и детектирует микроконтроллер. Происходит это посредством измерения времени зарядки системы из конденсатора и пластины. В случае, если разница между емкостью пассивного и активного режимов превышает порог, рассчитываемый по критерию Неймана-Пирсона, запускается работа основной программы.

При этом устройство потребляет 36 мА тока. Механические колебания происходят с частотой от 50 Гц до 1 КГц. Частота колебаний задается ШИМ контроллером с шагом, выведенным из психофизического закона восприятия Веббера-Фехнера:

$$p = \frac{k \ln S}{S_0},$$

где  $S$  – значение интенсивности раздражителя,  $S_0$  – нижнее граничное значение интенсивности раздражителя, если  $S < S_0$ , раздражитель совсем не ощущается,  $k$  – константа, зависящая от субъекта ощущения. Всего в цикле 8 равных по времени шагов. Общее время работы колебаний в цикле – 15 мин, после чего устройство отключает механические колебания на 10 мин, после чего, при выполнении условия превышения порога, цикл начинается сначала.

Так как в техническом задании акцент был сделан на длительный срок работы устройства без подзарядки, к выбору элементной базы был особый подход. В качестве контроллера был выбран ATTiny13A, как один из самых энергоэффективных и дешевых на рынке на момент проектирования [3]. Пассивные элементы в прототипе были прецизионного класса точности. В качестве программного обеспечения была выбрана интегрированная среда разработки от компании Atmel – AVR Studio, как оптимальная для вышеупомянутого контроллера. В результате, получилось устройство, исправно функционирующее без подзарядки около трех дней.



К сожалению, малый объем памяти контроллера не позволил реализовать программный код на языке C/C++, поэтому часть кода была отредактирована и упрощена в машинном коде, что не повлияло существенным образом на функционал готового устройства.

На данный момент аппарат уже прошел сертификацию и готовится к массовому производству. Несомненно, этому способствует низкая цена и высокая эффективность готового устройства. Также аппарат показал достойное время работы без подзарядки, достигающее 78 часов.

### Библиографический список

1. Еремушкин М. А. Классическая техника массажа при травмах и заболеваниях опорно-двигательного аппарата. Киев: Наука и техника, 2010.
2. Васичикин В. И., Васичкина Л. Н. Методики массажа. Ростов-на-Дону: Феникс, 2006.
3. 8-bit AVR Microcontroller with 1k Bytes in-system Programmable Flash. ATtiny13A . Summary. Atmel, 2008.

УДК 535.015

**Е. В. Козырев**

магистрант кафедры радиотехнических и оптоэлектронных комплексов

**Н. А. Гладкий**

научный руководитель

## ВЫХОДНЫЕ ФАЗОКОМПЕНСИРОВАННЫЕ ЗЕРКАЛА ДЛЯ CO<sub>2</sub> ЛАЗЕРОВ СО СТУПЕНЧАТЫМ ПРОФИЛЕМ ОТРАЖЕНИЯ

В лазерах с использованием активной среды с большим коэффициентом усиления (например, технологических CO<sub>2</sub>-лазерах атмосферного давления с поперечной прокачкой), как в устойчивых, так и в неустойчивых резонаторах, возникает возможность генерации кроме основной моды, одной или нескольких поперечных мод высших порядков.

Краевые дифракционные эффекты приводят к дополнительной возможности вырождения по потерям поперечных мод, т. е. к нарушению структуры распределения основной нижней поперечной моды. В работах [1–6] рассмотрено уменьшение эффектов краевого дифракционного рассеяния в результате использования в резонаторах специальных выходных элементов – зеркал с переменным по радиусу коэффициентом отражения. Использование таких зеркал увеличивает дифракционные потери в поперечных модах и позволяет реализовать генерацию продольных мод. В настоящей работе рассмотрены зеркала со ступенчатым коэффициентом отражения и предложены различные конструкции для их реализации.

Существуют несколько путей решения данной проблемы. В одном из вариантов в качестве метода фазовой компенсации световых пучков проходящих через отражающую и просветленную области (зоны А и в на рис. 1) на центральную зону с обратной стороны наносится компенсационный слой, такой толщины, чтобы дополнить разность фаз центрального и периферийного лучей до  $2\pi$ .

Описанный метод был реализован в конструкции резонатора промышленного технологического лазера типа «Хебр» с выходной мощностью до 1,5 кВт [5].

Возможен также вариант нанесения компенсационного слоя на лицевую сторону элемента непосредственно под отражающее покрытие; при этом можно нанести вещество даже с отличным от подложки коэффициентом преломления. Но в этом случае надо учитывать, что увеличение толщины пленочной структуры на одной стороне подложки, снижает лучевую прочность и уменьшает долговечность покрытия.

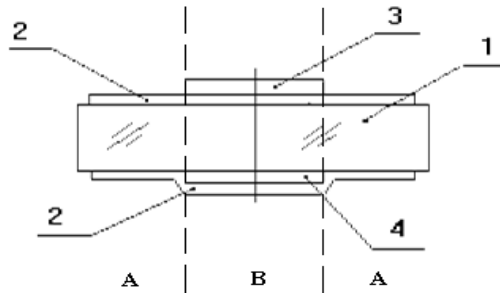


Рис. 1. Конструкция выходного элемента со ступенчатым поперечным профилем.  
 А – зона, где набег фазы равен 0. В – зона, где набег фазы не равен 0.  
 1 – подложка, 2 – просветляющее покрытие, 3 – отражающее покрытие,  
 4 – фазокомпенсирующий слой

В другом примере компенсация разности фаз пучков, проходящих через отражающую и просветленную области, достигалась путем выборки части толщины подложки с обратной стороны элемента методом ионной полировки (рис. 2). В работе [3] авторы сообщают, что ими был разработан и изготовлен подобный фазокомпенсированный элемент и успешно использован в неустойчивом резонаторе мощного 5 кВт  $\text{CO}_2$ -лазера с поперечной прокачкой активной среды.

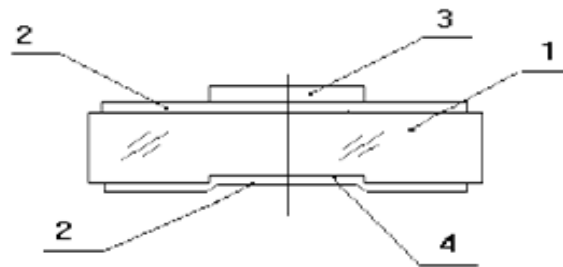


Рис. 2. Пример конструкции фазокомпенсированного выходного элемента со ступенчатым поперечным профилем коэффициента отражения.  
 1 – подложка, 2 – просветляющее покрытие, 3 – отражающее покрытие,  
 4 – выборка в подложке

Возможен и принципиально другой подход при конструировании и изготовлении фазокомпенсированного выходного элемента. Он связан с возможностью создания таких отражающих систем, которые не требуют дополнительной компенсации фазы проходящего света. Для создания такой структуры необходимо, чтобы разность набег фаз проходящей волны, между ее оптической и физической толщиной была кратна  $2\pi$ . При этом такая система должна обеспечивать требуемый коэффициент отражения. Примером такой системы может служить восьмислойное зеркало из  $\lambda/4$  – пленок с высокими (В) и низкими (Н) коэффициентами преломления, имеющее структуру типа:

$$H; B; H; B; xH; (2-x)B; (2-x)H; xB \quad (1)$$

Здесь  $x$  и  $(2-x)$  коэффициенты, задающие толщину низкопреломляющих – (Н) и высокопреломляющих – (В) пленок (в долях  $\lambda/4$ ). При использовании в таком многослойном зеркале в качестве пленкообразующих веществ  $\text{PbF}_2$  и  $\text{ZnSe}$  или  $\text{As}_2\text{S}_3$ , легко видеть, что гео-

метрическая толщина такой структуры равна  $2\pi$ , а оптическая –  $4\pi$ . Таким образом, разность набега фаз составляет  $2\pi$  при любом значении параметра  $x$ . Варьируя  $x$  в диапазоне от 1 до 1,6, мы имеем возможность получить коэффициенты отражения от 0,7 до 0,9, что обычно перекрывает область значений коэффициента отражения центральной зоны, требуемых для таких элементов. Но необходимо отметить, что в реальных условиях, при использовании пленок  $PbF_2$  и  $ZnSe$  из-за диффузии происходит усиление поглощения такой структурой [4, 5]. Поэтому сочетание этих пленок не целесообразно использовать при изготовлении зеркал технологических лазеров с высокой мощностью излучения из-за их низкой лучевой прочности.

Особенность создания фазокомпенсированного зеркала состоит в том, что при напылении необходимо использовать специальные диафрагмы, позволяющие наносить покрытие на отдельные участки подложки. А так же необходимо, чтобы коэффициент преломления компенсационного слоя не отличался от коэффициента преломления подложки.

В первом варианте оптическая толщина компенсационного слоя определялась следующим образом. Разность фаз пучков –  $\Delta\Phi$ , проходящих через отражающую и просветленную зоны, равна:

$$\Delta\Phi = \frac{2\pi}{\lambda} \left( h_B + h_H + h_K - \frac{h_B}{n_B} - \frac{h_H}{n_H} - \frac{h_K}{n_K} \right) \quad (2)$$

здесь:  $\lambda$  – длина волны,  $n_B$ ,  $n_H$  и  $n_K$  – коэффициенты преломления высокопреломляющих (В), низкопреломляющих (Н) и компенсационной (К) пленок, создающих диэлектрическое зеркало в центральной зоне элемента;  $h_B + h_H + h_K$  – суммарная оптическая толщина всех пленок.

Для фазовой компенсации необходимо, чтобы разность фаз  $\Delta\Phi$  пучков излучения, проходящих через центральную и периферийную зоны, была кратна  $2\pi$ , т.е.  $\Delta\Phi = 2\pi k$ ,  $k = 0, 1, 2, 3, \dots$ . Оптическая толщина  $h$  измеряется в долях  $\lambda/4$ . Сделав соответствующие замены, получим для толщины компенсационного слоя  $h_K$  в долях  $\lambda/4$ :

$$h_K = \left[ 4k - h_H \left( 1 - \frac{1}{n_H} \right) - h_B \left( 1 - \frac{1}{n_B} \right) \right] / \left( 1 - \frac{1}{n_K} \right) \quad (3)$$

где  $k$  – минимальное натуральное число, при котором  $h$  – положительно.

Коэффициент отражения центральной зоны элемента был выбран равным  $R = 80\%$ . Обоснование приведено в статье [1]. Для его получения было использовано шестислойное диэлектрическое зеркало с высокопреломляющими слоями из  $ZnSe$  ( $n_B = 2,42$ ) и низкопреломляющими из  $PbF_2$  ( $n_H = 1,65$ ). Формула структуры покрытия:

$$П(0,4H_1)(1,6B_1)(1H_1)(1B_1)(1H_1)(1B_1) \quad (4)$$

Коэффициент преломления компенсационного слоя из  $ZnSe$  ( $n_K = 2,42$ ) практически соответствовал коэффициенту преломления подложки. Подставляя эти данные в (3) (при  $k = 1$ ) получим толщину компенсационного слоя в долях  $\lambda/4$  равную  $h_K = 1,68$  при  $\lambda = 10,6$  мкм.

Для получения зеркала, не требующего компенсации фазы проходящего излучения, была предложена восьмислойная структура (5) из пленок с высоким  $As_2S_3$  и низким  $PbF_2$  коэффициентом преломления.

$$П(1H_1)(1B_1)(1H_1)(1B_1)(0,4H_1)(1,6B_1)(1,6H_1)(0,4B_1) \quad (5)$$

Но необходимо отметить, что такая структура, ввиду большого количества слоев, обладает большим поглощением. Поэтому для применения в мощных технологических лазерах лучше подходит первый вариант фазокомпенсированного выходного элемента.

Выводы

В работе проведен обзор конструкций фазокомпенсированных выходных элементов технологических лазеров со ступенчатым профилем отражения, которые позволяют получить равномерно заполненную моду большого объема и лишенную значительных фазовых искажений выходящего светового поля. Рассмотрена методика изготовления таких зеркал с учетом необходимых характеристик коэффициента отражения и допустимых фазовых искажений.

### Библиографический список

1. Прокашев В. Н. Разработка и исследование выходных фазокомпенсированных зеркал с профилем отражения для CO<sub>2</sub>-лазеров: Автореф. дис. на соискание ученой степени канд. техн. наук: 05.11.07 / СПб ГУАП. СПб., 2005. 21 с. 2005.
2. Аникичев С. Г., Котликов Е. Н., Прокашев В. Н. Формирование фронта лазерного излучения выходными зеркалами // Тез. докл. конф. Оптика лазеров 93. 1993. Т. 2. 258 с.
3. Yagi S., Yasui K., Takenaka Y. // Proc. SPIE. 1990. V. 1225. 357 p.
4. Котликов Е. Н., Прокашев В. Н., Хонинева Е. В. Проблема снижения оптических потерь в пленках фторидов / Оптический журнал. Июнь 2004. Т. 71, № 6. С. 84–87.
5. Kotlikov E. N., Khonineva E. V., Prokashev V. N. Output mirrors for tunable lasers in infrared spectral range. Technical Program. XI Conference on Laser Optics. St. Petersburg, June, 30-July, 4, 2003. Russia. P. 68.
6. Борисов М. Ф., Котликов Е. Н., Прокашев В. Н., Родионов А. Ю. Формирование мощного узконаправленного лазерного излучения для дальней транспортировки / Оптический журнал. Т. 66, В. 11, 1999. С. 99–103.

УДК 535.345.673: 517.51

**А. М. Комарова**

магистрант кафедры радиотехнических и оптоэлектронных комплексов

**Н. А. Гладкий**

научный руководитель

### ИССЛЕДОВАНИЕ ОПТИЧЕСКИХ КОНСТАНТ ПЛЕНОК ZNS ПО СПЕКТРАМ ПРОПУСКАНИЯ В ДИАПАЗОНЕ 0,2–1,1 МКМ

Предложена методика определения оптических констант (ОК) пленок (коэффициента преломления  $n_\lambda$  и поглощения  $k_\lambda$ ) по спектрам пропускания. В основе предложенной методики лежит способ определения вклада поглощения в спектры отражения и пропускания пленки в минимумах и максимумах спектров пропускания. Экспериментально определены оптические константы пленки сульфида цинка в диапазоне спектра 0,2–1,1 мкм.

Для определения оптических констант широко используются спектрофотометрические методы [1, 2]. В этих методах оптические константы пленок определяются по спектрам пропускания ( $T_0$ ) и отражения ( $R_0$ ). Целью данной работы являлась разработка метода определения оптических констант пленок ZnS только по спектрам пропускания. Оптические пленки изготавливались в вакууме на установке ВУ-1М путем испарения из молибденового тигля [3].

Рассмотрим предлагаемую методику. Связь между пропусканием, отражением и поглощением задается как [4, 5]:

$$1 = A + R + T \quad (1)$$

где  $A$  – полное поглощение в пленке. Поглощение определяется потерями энергии в пленке и подложке и так же, как сама энергия, является аддитивной величиной. Обозначим вклад поглощения в спектры отражения как  $A_r$ , и в спектры пропускания как  $A_t$ . Тогда мы можем разделить  $A$  на долю в пропускании  $A_t$  и в отражении  $A_r$  [6]:

$$A = A_r + A_t \quad (2)$$

Анализ спектров  $A$ ,  $A_r$  и  $A_t$  позволит определить возможность использования спектров пропускания для определения поглощения в пленке. Нахождение  $A_r$  и  $A_t$  проводилось численными методами для модели поглощающей и непоглощающей пленок. Для расчетов спектров ис-

пользовалась разработанная на кафедре физики Санкт-Петербургского государственного университета аэрокосмического приборостроения программа FilmManager [5], позволяющая анализировать спектры отражения и пропускания пленок с учетом поглощения в них и отражения от обратной стороны подложки.

В экспериментах мы использовали кюветы, имеющие 4 поверхности. Экспериментальные спектры приведены на (рис. 1). Эти данные нормировались на пропускание от одной стороны, т. е. от полубесконечной подложки.

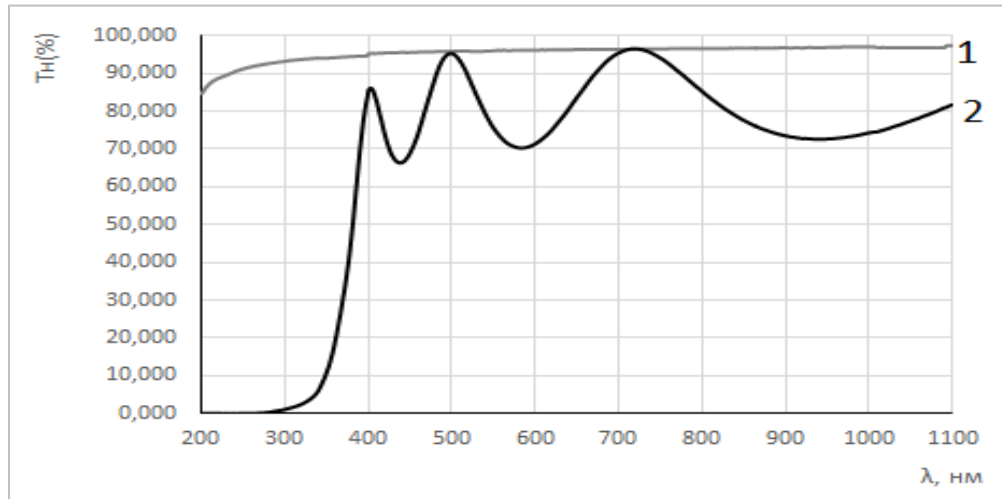


Рис. 1. Нормированные спектры пропускания кварцевой кюветы (1 –  $T$ , %) и пленки ZnS (2 –  $T$ , %)

Как видно из рис. 1, в спектрах присутствуют области, свободные от поглощения: 650–1100 нм. В этой области можно определить коэффициент преломления  $n_2$ , используя положение экстремумов для длин волн.

Длины волн, соответствующие экстремумам пропускания, связаны соотношением:

$$\lambda_{\text{экстр}k} = \frac{4n_2h_2}{2m} \quad (3)$$

где  $h_2$  – геометрическая толщина пленки,  $m = 1, 2, 3 \dots$

На основе этого выражения для соседних экстремумов можно записать систему:

$$\begin{cases} \lambda_{\text{max}k} = \frac{4n_2h_2}{m} \\ \lambda_{\text{min}k} = \frac{4n_2h_2}{(m-1)} \\ \lambda_{\text{max}k+1} = \frac{4n_2h_2}{(m+1)} \end{cases} \quad (4)$$

Решая эту систему, можно найти оптическую толщину пленки  $d = n_2 h_2$ , ее геометрическую толщину  $h_2$  и коэффициент преломления  $n_2$ . Результаты, полученные по уравнениям (3) и (4), а также с использованием программы FilmManager, следующие:  $d = 705$  (нм) и  $n_2 = 2,23$  для длины волны 942 нм.

С помощью программы FilmManager был проведен расчет спектров пропускания модели пленки ZnS на подложке из кварца: с поглощением  $T$  и без поглощения  $T_0$ . Для максимумов про-

пускания при поглощении <30% вклад поглощения в отражение является малым. Отрицательные значения поглощения в коротковолновой области спектра связаны с выбранной моделью. В экстремумах пропускания можно считать разницу между пропусканием с нулевым поглощением и пропусканием реальной пленки равной поглощению:  $A_{\text{макс}} = A = T_0 - T$ . Данные для  $T_0$  мы получали путем расчета пленки без поглощения,  $T$  – экспериментальные данные синтезированных спектров пропускания.

Результаты по оптическим константам приведены в (таблице).

Таблица

Результаты вычислений ОК для пленки ZnS на подложке из кварца

$\lambda$ , нм	$n_f$	$d$ , нм	$k$
403	2.55	806	0.011
438	2.43	768	0.001
499	2.37	749	<0.001
584	2.31	730	<0.001
718	2.27	717	<0.001
942	2.23	705	<0.001

Полученные результаты согласуются с известными литературными данными.

В настоящей работе была предложена методика определения оптических констант пленок по спектрам пропускания. В основе его лежит численно доказанное утверждение, согласно которому у пленок на определенном интервале длин волн все поглощение можно определить по спектрам пропускания. И, таким образом, оптические константы могут быть определены только из спектра пропускания. Особенностью метода является возможность определения оптических констант только для точек, соответствующих максимумам пропускания, что делает его применение невозможным при наличии в спектре исследуемого вещества узких полос поглощения.

### Библиографический список

1. Андриевский Б. В., Вахулович В. Ф., Курляк В. Ю., Романюк Н. А. Определение дисперсии показателя преломления и толщины тонких пленок по спектрам отражения или пропускания // Оптика и спектроскопия. 1988. Т. 65. № 1. С. 136–140.
2. Хевенс О. С. Измерение оптических констант тонких пленок // Сб. физика тонких пленок. В 8 т. Т. 2 / Под общ. ред. Г. Хасса и Р. Э. Тауна; перевод с англ. под ред. В. Б. Сандомирского и А. Г. Ждана. М.: Мир, 1967. С. 136–185.
3. Справочник технолога оптика. Под общей редакцией С. М. Кузнецова и М. А. Окатова. Л.: Машиностроение, 1983, – 414 с.
4. Котликов Е. Н., Варфоломеев Г. А., Лавровская Н. П., Тропин А. Н. Проектирование, изготовление и исследование интерференционных покрытий: учеб. пособие. СПб ГУАП. 2010. 185 с.
5. Котликов Е. Н., Новикова Ю. А. Программа синтеза и анализа интерференционных покрытий «Film Manager». Научный журнал Информационно-управляющие системы № 1(68) 2015. С. 15–20.
6. Котликов Е. Н., Новикова Ю. А. Исследование оптических констант пленок ВахМе1-хF2 // Опт. и спектр. 2014. Т. 117. № 3. С. 48–52.

УДК 621.396.676

**М. А. Котлова**

студент кафедры медицинской радиоэлектроники

**В. Ф. Михайлов**

д-р техн. наук, проф. – научный руководитель

## ИССЛЕДОВАНИЕ РЕАЛИЗАЦИИ СВЯЗИ В МЕДИЦИНЕ КАТАСТРОФ

Медицина катастроф – это особая область медицины, задачей которой является организация оказания медицинской помощи (вплоть до специализированной) людям, пострадавшим в результате возникновения чрезвычайной ситуации (ЧС), то есть ситуации, возникшей внезапно и создавшей угрозу для здоровья человека [1]. Важную роль в процессе оказания помощи в медицине катастроф играет организация системы связи.

Обеспечить сеанс телемедицины для разбора ситуации или оперативного медицинского вмешательства на месте ЧС можно с помощью спутниковой связи. В распоряжении современной службы медицины катастроф находятся автоматизированные самонаводящиеся антенные комплексы «iNetVu» (VSAT-комплексы, то есть терминал с небольшой антенной), разделяющиеся по способу крепления и перевозки. Наибольшую популярность в России получил автоматизированный комплекс с антенной Ku-диапазона (диапазон частот сантиметровых длин волн, от 12 до 18 ГГц или от 2,5 до 1,67 см) диаметром 1,2 м «iNetVu ®1200» для крепления на крышу автомобиля (рис. 1) [2]. Такая антенная система на мобильной платформе используется там, где требуется большая пропускная способность и хороший спутниковый канал, прежде всего, это медицинские передвижные лаборатории. Время развертывания спутникового терминала из транспортного положения в рабочее не превышает 20 мин. (с момента остановки подвижной платформы). Такие автомобили могут служить как командным центром, так и мобильной операционной (рис. 2).

Но, как и везде, информационно-коммуникационные технологии в системе Всероссийской службы медицины катастроф сталкиваются с проблемами своего применения и, соответственно, развития.

Первая проблема связана с апертурой антенны и ее техническими характеристиками. Ведь для того, чтобы обеспечить приемлемую вероятность ошибки, приходится использовать большие антенны, малозумящие элементы и сложные коды. Особенно остро эта проблема стоит в системах подвижной связи, так как в них есть ограничение на размер антенны и, как правило, на мощность передатчика.



Рис. 1. Автоматизированный комплекс «iNetVu ®1200» с антенной диаметром 1,2 м



Рис. 2. Автомобиль с автоматизированным комплексом «iNetVu @1200»

Второй существенной проблемой можно назвать дефицит спутникового ресурса на космических аппаратах, в то время как для операторов сетей VSAT, работающих в отрасли медицины катастроф, повышение эффективности спутниковых технологий является вопросом «жизни или смерти». Восполнение российской спутниковой группировки – самый очевидный и эффективный способ решения проблемы. В январе 2015 г. уже вошел в эксплуатацию спутник «Ямал-401» от ОАО «Газпром космические системы». После «Ямал-401» в ближайшие годы планируются запуски спутников «Ямал-500» и «Ямал-600», которые будут использовать совершенно новый диапазон «Ka» (диапазон частот сантиметровых и миллиметровых длин волн, от 26,5 до 40 ГГц или от 1,13 до 0,75 см), предназначенный для широкополосного доступа в Интернет. Возможность обеспечения высокоскоростной широкополосной связи на мобильных платформах как раз крайне важна для служб экстренного реагирования и аварийно-спасательных операций. Кроме того, уменьшение размера антенны – еще одно из преимуществ использования Ka-диапазона. Спутниковая антенна диаметром 0,75 м – малогабаритная и легкая. Поэтому ее сборка и установка сильно упрощены по сравнению с традиционными в России 1,2-метровыми антеннами VSAT.

Недостатком систем спутниковой связи Ka-диапазона является низкий порог помехоустойчивости (чем больше частота, тем больше пределы для возникновения помехи) и сильная зависимость от условий окружающей среды, в особенности от атмосферных осадков. Вместе с тем VSAT-системы в Ku-диапазоне проигрывают по стоимости и габаритам оборудования (минимальный диаметр антенн Ka-диапазона – 60 см). Поэтому разработчикам и производителям VSAT оборудования нужно уделять повышенное внимание вопросам кодировки сигнала и повышения уровня помехоустойчивости. [3]

Таким образом, обобщая сказанное ранее, в вопросе реализации связи в медицине катастроф открываются новые возможности его решения и появляются новые перспективы развития этого направления. По причине того, что Всероссийская Служба Медицины Катастроф обеспечивает медицинскую помощь и эвакуацию пострадавших в результате чрезвычайных ситуаций, а районы возникновения ЧС часто не имеют иных средств связи, кроме спутниковых, в настоящее время основной потребностью такой службы являются мобильные быстро разворачиваемые VSAT-станции минимального веса, которые могут перевозиться любым видом транспорта и при помощи которых будет обеспечен спутниковый широкополосный доступ, так как для передачи медицинской информации предпочтительно использовать широкополосные каналы. Реализация связи в медицине катастроф с использованием осваиваемого в настоящее время Ka-диапазона позволяет удовлетворить некоторые из вышеперечисленных требований к реализации такого вида связи, однако технические вопросы поддержания качественных параметров соединения нуждаются в более углубленном изучении данной тематики.

#### Библиографический список

1. Колб Л. И. и др. Медицина катастроф и чрезвычайных ситуаций. Минск: «Высшая школа», 2008. 465 с.



2. [www.vcmk.ru](http://www.vcmk.ru)
3. Спутниковые системы массового обслуживания Ка-диапазона – магистральное мировое направление развития VSAT или рекламная акция? // Специальный выпуск «Спутниковая связь и вещание». 2011. С. 22–31.

УДК 621.376

**П. А. Прокофьев**

магистрант кафедры инфокоммуникационных технологий

**Г. И. Худяков**

д-р техн. наук, проф. – научный руководитель

## АНАЛИЗ СКОРОСТИ ПЕРЕДАЧИ ИНФОРМАЦИИ В СИСТЕМАХ С МНОГОЧАСТОТНОЙ МОДУЛЯЦИЕЙ И МНОГОПОЗИЦИОННОЙ ЧАСТОТНОЙ МОДУЛЯЦИЕЙ

Основной проблемой теории связи является поиск методов передачи и приема сигналов, обеспечивающих малые потери информации, заложенной в сообщении, и максимально возможную скорость передачи при минимальной стоимости системы связи. На практике каждое из этих требований обеспечивается за счёт других, таких, как расширение полосы частот, увеличение средней мощности передатчика и др.

Наиболее перспективные системы передачи информации, позволяющие найти оптимальное решение совокупности указанных задач, основываются на цифровых методах передачи. [1].

### Многочастотная модуляция

В телекоммуникационной практике при передаче дискретных сообщений (ДС) по частотно-ограниченным каналам связи (КС), с целью нивелирования межсимвольных искажений (МСИ) широко применяется метод параллельной многочастотной модуляции (МЧМ), в англоязычной литературе это метод DMT (Discrete Multi None) [1, 2]. Здесь для передачи ДС используется ряд параллельных индивидуальных каналов с центральными частотами  $f_n, n = 1, N$  спектры сигналов, которые занимают полосу частот  $2F$ , много меньшую полосы частот  $\Delta F_{KS} = F_{max} - F_{min}$  канала связи (рис. 1). По сравнению с методом последовательной одночастотной передачи этот метод имеет то преимущество, что практически не чувствителен к искажениям амплитудно-частотных (АЧХ) и фазочастотных (ФЧХ) характеристик КС и, как следствие, приводит к значительному упрощению задачи выравнивания на приеме передаточной функции КС. Важное свойство модемов с МЧМ, заключается в слабой зависимости качества передачи ДС от искажений АЧХ и ФЧХ, что обеспечило им широкое применение в радиоканалах с многолучевым распространением и значительными селективными частотными искажениями. Известны различные модификации такого вида модемов, отличающиеся формой спектров сигналов переносчиков, а также частотными характеристиками соответствующих приемных фильтров [2, 3].

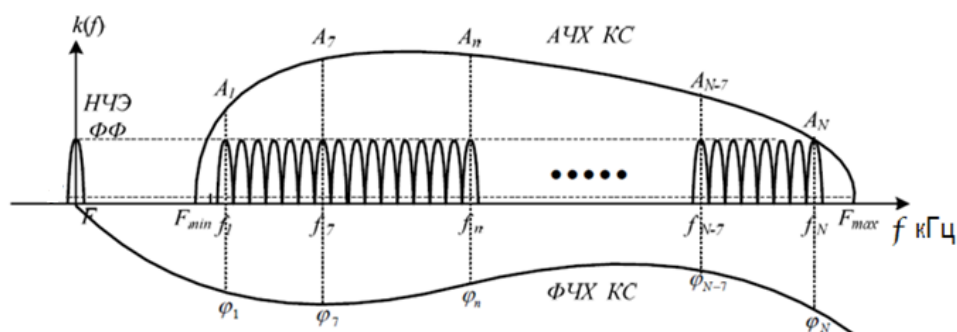


Рис. 1. Размещение индивидуальных подканалов в частотно-ограниченном КС при МЧМ. ФФ – формирующий фильтр, НЧЭ – низкочастотный эквивалент,  $\varphi$  – угловая частота

Многопозиционная частотная модуляция (MFSK)

Сигналы в многопозиционной частотной модуляции могут быть описаны в соответствии со следующим выражением:

$$S_1(t) = \cos(\omega_1 t), S_2(t) = \cos(\omega_2 t), S_N(t) = \cos(\omega_N t), \quad (1)$$

где  $S_1$  – используется для передачи первого состояния символа;  $S_2$  – для передачи второго состояния символа;  $S_N$  – для передачи  $N$ -го состояния символа;  $\omega$  – угловая частота;  $t$  – время.

Использование многопозиционной частотной модуляции позволяет реализовать высокочастотный сигнал с постоянной амплитудой. Такой сигнал позволяет строить радиопередатчики с максимальным КПД, так как при применении сигнала с постоянной амплитудой, усилитель мощности радиопередатчика работает в оптимальном режиме.

На практике получила распространение двойная частотная модуляция – ДЧТ (С4FM) использующаяся в режиме с непрерывным изменением фазы сигнала. В этом виде модуляции используется четыре значения частоты несущего колебания. Таким количеством частот можно передать два символа в течение длительности одного символа.

Дальнейшее увеличение количества частот в радиоканале не имеет смысла, так как это приводит к неоправданному расширению спектра сигнала. Ширина спектра сигнала расширяется пропорционально количеству частот, а количество одновременно передаваемых бит растет пропорционально двоичному логарифму от количества использованных частот.

Характеристика линий связи

Характеристики линий связи. Основные характеристики канала связи – пропускная способность и достоверность передачи данных. Пропускная способность канала (количество информации, передаваемое в единицу времени) оценивается предельным числом бит данных, передаваемых по каналу за единицу времени, и измеряется в бит/с. Достоверность передачи данных оценивается по интенсивности битовых ошибок (Bit Error Rate), определяемой как вероятность искажения передаваемого бита данных. Величина BER для каналов связи без дополнительной защиты от ошибок составляет  $10^{-4}$ – $10^{-6}$ . Основная причина искажений – воздействие помех на линию связи. Помехи, как правило, носят импульсный характер и имеют тенденцию к группированию – образованию пачек помех, искажающих сразу группу соседних бит в передаваемых данных.

Пропускная способность канала связи определяется полосой частот и помехоустойчивостью канала. Полоса частот, в которой амплитудно-частотная характеристика (АЧХ) линии связи имеет значение не ниже заданного (например, по уровню 0,5), называется полосой пропускания. Полоса частот  $DF = f_e - f_H$ , где  $f_H$  и  $f_e$  – нижняя и верхняя границы частот, определяет диапазон частот, эффективно передаваемых по линии. Полоса частот зависит от типа линии и ее протяженности. Проводные линии связи имеют полосу частот примерно 10 кГц, кабельные 100 кГц, коаксиальные 100 МГц, радиорелейные – 1000 МГц, волоконно-оптические – 100 ГГц.

Помехоустойчивость линии зависит от мощности помех, создаваемых в линии внешней средой или возникающих из-за шумов в самой линии. Наименее помехоустойчивыми являются радиолнии, хорошей помехоустойчивостью обладают кабельные линии, отличной – волоконно-оптические линии, не восприимчивые к электромагнитному излучению.

Пропускная способность канала зависит от ширины полосы частот линии связи и отношения мощностей сигнала и шума. Математически эта связь описывается формулами Шеннона (2) и Найквиста (3). Максимальная пропускная способность канала, построенного на основе линии с полосой частот  $DF$  и отношением сигнал-шум, составляет (бит в секунду):

$$C_{\max} DF \log_2 (1 + P_c / P_{\text{ш}}). \quad (2)$$

Значение  $1 + P_c / P_{\text{ш}}$  определяет число уровней сигнала, которое может быть воспринято приемником. Так, если отношение  $P_c / P_{\text{ш}} > 3$  то единичный сигнал может переносить четыре значения, т. е. 2 бита информации.

$$C = F \log_2(M), \quad (3)$$

где  $M$  – число различных состояний сигнала;  $C$  – число уровней сигнала.

Максимально возможная пропускная способность не зависит от способа физического кодирования, так как определяет возможности линии при гипотетическом наилучшем способе кодирования. Практическая пропускная способность канала существенно зависит от способа физического кодирования информации. Одна и та же линия связи может иметь разную пропускную способность при разных способах кодирования. При этом стремятся выбрать способ кодирования так, чтобы максимально использовать возможности линии. В свою очередь возможность применения на линии того или иного способа кодирования определяется полосой пропускания и затуханием сигнала. Затухание это отношение амплитуд входного и выходного сигналов на заданной частоте. Часто затухание выражается в децибелах и вычисляется по формуле.

$$A = 20F \log_2 10(A_{\text{вых}}/A_{\text{вх}}) \quad (4)$$

Затухание всегда задается для определенной длины линии связи. При передаче данных широко используются двоичные сигналы, принимающие значения 0 и 1. Минимальная длительность такта, с которой могут передаваться сигналы по каналу с полосой частот  $DF$ , равна  $T_{\min} = 1/(2DF)$ . Если вероятность искажения символов 0 и 1 из-за помех одинакова и равна  $p$ , то число двоичных символов, которые можно безошибочно передать по каналу в секунду:

$$C = 2DF[1 + PF \log_2 P + (1 - P) \log_2 (1 - P)]. \quad (5)$$

Это выражение определяет пропускную способность двоичного канала. Величина в квадратных скобках определяет долю двоичных символов, которые передаются по каналу с частотой  $2DF$  без искажений. Если помехи отсутствуют, вероятность искажения символа  $p = 0$  и пропускная способность  $C = 2DF$ . Если вероятность искажения  $p = 0,5$ , то пропускная способность  $C = 0$ . Если по каналу передается сообщение длиной  $n$  двоичных символов, то вероятность появления в нем  $m$  ошибок.

$$P(n, m) = C_n^m (1 - P)^{n-m}, \quad (6)$$

Наиболее часто используемый тип канала – телефонный с полосой пропускания 3,1 кГц (диапазон частот 0,3–3,4 кГц). Коммутируемый телефонный канал обеспечивает скорость передачи данных  $C = 1200$  бит/с, некоммутируемый – до 9600 бит/с.

Эффективность использования канала связи для передачи данных принято характеризовать удельной пропускной способностью  $B = C/DF$ , т. е. пропускной способностью на 1 Гц полосы частот канала. Для коммутируемых телефонных каналов удельная пропускная способность не превышает 0,4 бит/с Гц, а для некоммутируемых составляет, как правило, 3–5 бит/с Гц. Стандартизированы следующие скорости передачи данных по каналам связи: 200, 300, 600, 1200, 2400, 4800, 9600, 12 000, 24 000, 48 000 и 96 000 бит/с. Каналы с пропускной способностью до 300 бит/с называются низкоскоростными, от 600 до 4800 бит/с – среднескоростными, и с большей пропускной способностью высокоскоростными.

Пропускная способность зависит не только от физического кодирования, но и от предварительного логического кодирования, которое заключается в предварительной подготовке данных, влияющей на ширину спектра итогового сигнала.

### Заключение

Многочастотная модуляция (МЧМ) имеет ряд преимуществ. Она практически нечувствительна к искажениям амплитудно-частотных (АЧХ) и фазочастотных (ФЧХ) характеристик.

Многопозиционная частотная модуляция (MFSK) позволяет использовать высокочастотный сигнал с постоянной амплитудой. Это дает возможность работать радиопередатчикам с максимальным КПД, так как усилители мощности работают в оптимальном режиме.

**Библиографический список**

1. Худяков Г. И. Оценка пропускной способности каналов авиационной цифровой электросвязи. М.: Электросвязь. 2009. 299 с.
2. <http://www.sbi-telecom.ru/mnogochastotnaya-modulyaciya.html>
3. Котельников В. А. Теория потенциальной помехоустойчивости. М.: Госэнергоиздат, 1956. 190 с.

УДК 528.88

**Н. В. Разумова**

магистрант кафедры радиотехнических и оптоэлектронных комплексов

**А. Ф. Крячко**

д-р техн. наук, проф. – научный руководитель

**МЕТОДЫ СОВМЕЩЕНИЯ ИЗОБРАЖЕНИЙ  
В ЗАДАЧАХ ДИСТАНЦИОННОГО ЗОНДИРОВАНИЯ**

Различают три способа передачи данных со спутника на Землю: прямая передача данных на наземную станцию, находящуюся в прямой видимости спутника; сохранение данных на спутнике, а далее передача на Землю; применение системы геостационарных спутников связи, где данные передаются с одного спутника на другой, пока в зоне прямой видимости одного из них не окажется наземная станция.

Данные, которые поступают на наземную станцию исходном формате, обрабатываются: устраняются систематические ошибки, геометрические искажения и искажения, связанные с влиянием атмосферы.

Для получения данных дистанционного зондирования используют сканирующие системы, регистрирующие информацию о свойствах объекта в цифровом виде. Результатом обработки данных дистанционного зондирования является новый снимок. Существуют следующие этапы обработки цифровых снимков: коррекция и восстановление снимков; улучшение снимков; классификация данных; совмещение и объединение данных; интеграция в географические информационные системы [1].

Объекты на спутниковых изображениях иногда необходимо сопоставлять географической картой для определения географических координат объектов. Объединение изображений применяется при создании цифровых моделей рельефа по стереопарам снимков и для комбинирования изображений с различными уровнями пространственного и спектрального разрешения.

Совмещение – это процесс точного сопоставления пикселей двух наборов данных одним и тем же точками земной поверхности. Абсолютные координаты пикселей совмещенных друг относительно друга изображений могут оставаться неизвестными. Для получения более точного результата совмещения изображений требуется устранение на них искажений, которые связаны с датчиком и орбитой. Существуют различные способы объединения совмещенных изображений для упрощения получения информации. Способы, которые используются для съемки территорий с наложением снимков: повторная съемка одним и тем же датчиком с одного витка орбиты ИСЗ или с разных витков со смещением оси съемки от направления на надир; съемка с соседних витков орбиты с помощью одного и того же датчика с интервалом в несколько дней; съемка с помощью аппаратуры, установленной на разных ИСЗ; съемка с помощью космических и воздушных летательных аппаратов. Чтобы совместить изображения, необходимо выбрать преобразование координат, которое бы связывало бы координаты пикселей (строку и столбец) разных снимков между собой или с координатами общего репера. Такое преобразование можно найти с помощью нескольких наземных контрольных точек и соответствующей модели искажений. Обычно этих данных достаточно для совмещения космических снимков областей с небольшими или умеренными перепадами рельефа, но в этом случае определение и измерение наземных контрольных

точек приходится выполнять вручную. Кроме того, топографические изменения на космических и аэрофотоснимках высокого разрешения имеют пространственные частоты, которые обычно велики для точного моделирования местности с помощью глобальных полиномов. Таким образом, задача состоит в том, чтобы найти способы, которые позволили бы повысить плотность наземных контрольных точек и использовать модели более низкого порядка, например основанные на сплайн-интерполяции.

Автоматизированное определение наземных контрольных точек на двух изображениях осуществляется в два этапа. Из каждого изображения извлекается информация о пространственных объектах, далее эти объекты объединяются попарно путем сопоставления. Факторы, которые могут отрицательно повлиять на определение наземных контрольных точек – временные различия, разные углы съемки и разные характеристики датчиков.

В качестве пространственных объектов для автоматизированного совмещения можно применять небольшие участки земной поверхности – реперы. Реперы должны иметь небольшой размер, чтобы для их совмещения достаточно было простого смещения системы координат без внутренних деформационных преобразований.

Для определения наземных контрольных точек на двух снимках один из них выбирают в качестве опорного и выделяют на нем целевой репер  $T$  размером  $N \times N$ , а на втором снимке поисковый репер размером  $M \times M$ , где  $M > N$ . Затем целевой репер перемещают по центральной части поисковой области размером  $L \times L$ , и перемножают два массива значений пикселей с суммированием результатов для каждого смещения целевого репера  $(i, j)$ . Показатель пространственного сходства описывается формулой [3]:

$$\sum_{m=1}^N \sum_{n=1}^N T_{mn} S_{i+m, j+n}.$$

Совмещение двух изображений можно рассматривать как многомасштабную задачу, включающую глобальное совмещение (нахождение приближенной оценки преобразования координат с учетом глобального поворота и преобразования масштаба) и локальное совмещение (уточнение глобальной модели и устранение высокочастотных искажений).

В 1993 г. В работе Джамджи был разработан метод вейвлет-преобразования, для автоматического определения положений набора наземных контрольных точек на двух изображениях. Метод заключается в том, что в качестве пространственных элементов использовались локальные максимумы значений пикселей у высокочастотных компонентов. Данный метод позволяет автоматически выделить более 100 наземных контрольных точек.

Существует метод совмещения изображений с близкими значениями наземного интервала дискретизации, применяющийся в пространственной области. Для каждого снимка создается пирамида изображений, и на самых высоких уровнях с помощью корреляции площадных объектов находятся грубые оценки. Эти оценки учитываются на следующем более низком уровне [2, 3].

Совмещение изображений требуется в множестве решений задач дистанционного зондирования. Чтобы изображения совместить попиксельно, необходимо извлечение пространственных признаков и их согласование на обоих изображениях; для получения карт высот с высоким разрешением используют попиксельное совмещение стереопар снимков.

### Библиографический список

1. Чандра А. М., Гош С. К. Дистанционное зондирование и географические информационные системы / Чандра А. М., Гош С. К. – М., Техносфера, 2008. – 312 с.
2. Кашкин В. Б., Сухинин А. И. Дистанционное зондирование Земли из космоса. Цифровая обработка изображений: учеб. пособие. – М.: Логос, 2001. – 264с.
3. Шовенгердт Р. А. Дистанционное зондирование. Модели и методы обработки изображений/ Шовенгердт Р. А. – М., Техносфера, 2010. – 560 с.

**М. А. Семенцов**

магистрант кафедры радиотехнических и оптоэлектронных комплексов

**Н. А. Гладкий**

научный руководитель

## ОПРЕДЕЛЕНИЕ ОПТИЧЕСКИХ КОНСТАНТ ПЛЕНОК ФТОРИДА БАРИЯ ПО СПЕКТРАМ ОТРАЖЕНИЯ В ДИАПАЗОНЕ 200–1100 НМ

Для определения оптических констант (ОК) широко используются спектрофотометрические методы [1, 2]. В этих методах оптические константы пленок определяются по спектрам пропускания ( $T_0$ ) и отражения ( $R_0$ ). Однако, в некоторых случаях получение спектров отражения и пропускания одновременно технически невозможно. Для получения контрастных спектров требуется, чтобы коэффициенты преломления подложки и пленки максимально различались. Материалы, прозрачные в ультрафиолетовом (УФ) и видимом диапазонах спектра имеют значения коэффициентов преломления (1.4–1.5). Эти значения близки к значениям коэффициентов преломления пленок фторидов (1.3–1.5). Удобным материалом для получения контрастных спектров является кремний (Si), который имеет значения показателя преломления 3.5–4.0, однако, этот материал непрозрачен в УФ и видимом диапазонах спектра, поэтому получение спектров пропускания затруднительно.

В настоящей работе рассматривается методика определения ОК (коэффициентов преломления  $n_\lambda$  и поглощения  $k_\lambda$ ) с использованием только спектров отражения. В качестве подложек для напыления пленок использован кремний (Si) марки 1A2C [4]. Измерение спектров проводилось на спектрофотометре СФ-56 в диапазоне спектра 200–1100 нм, имеющего приставку зеркального отражения ПЗО-09, позволяющую проводить измерение спектров отражения.

Оптические пленки фторида бария ( $BaF_2$ ) изготавливались методом резистивного испарения в вакууме на установке ВУ-1А путем испарения материала из молибденового тигля [5]. Напыление проводилось при достижении вакуума порядка  $3 \cdot 10^{-3}$  Па. Температура подложек при напылении составляла 80 °С с точностью  $\pm 5$  °С. Контроль толщины напыляемых пленок и скоростей осаждения конденсата осуществлялся фотометрическим методом на длине  $\lambda = 0,8$  мкм.

Получение спектров отражения всегда требует наличие репера с фиксированным коэффициентом отражения. Пленки металлов, имеющие высокий коэффициент отражения не пригодны в УФ и видимом диапазонах спектра. Они не стабильны во времени и имеют малое отражение в УФ области спектра. Нами использовалась следующая методика для определения точного значения спектров отражения.

В качестве основного репера использовалась пластина кремния, коэффициент отражения которой менялся в пределах 200–1100 нм во всей области. Дополнительно в области 700–900 нм для абсолютной калибровки спектра отражения использовалось высокоотражающее 11-слойное зеркало из пленок  $As_2S_3$  и  $BaF_2$ , поглощение которого в этой области мене 0,2%. Снимался спектр пропускания такого зеркала  $T_1$ . Отражение определялось как  $R_1 = 1 - T_1$ . В дальнейшем спектр  $R_1$  использовался для нормировки полученных спектров отражения пластины кремния и пластины кремния с нанесенной пленкой фторида бария.

На рис. 1 представлены нормированные спектры пленки в диапазоне спектра 660–980 нм.

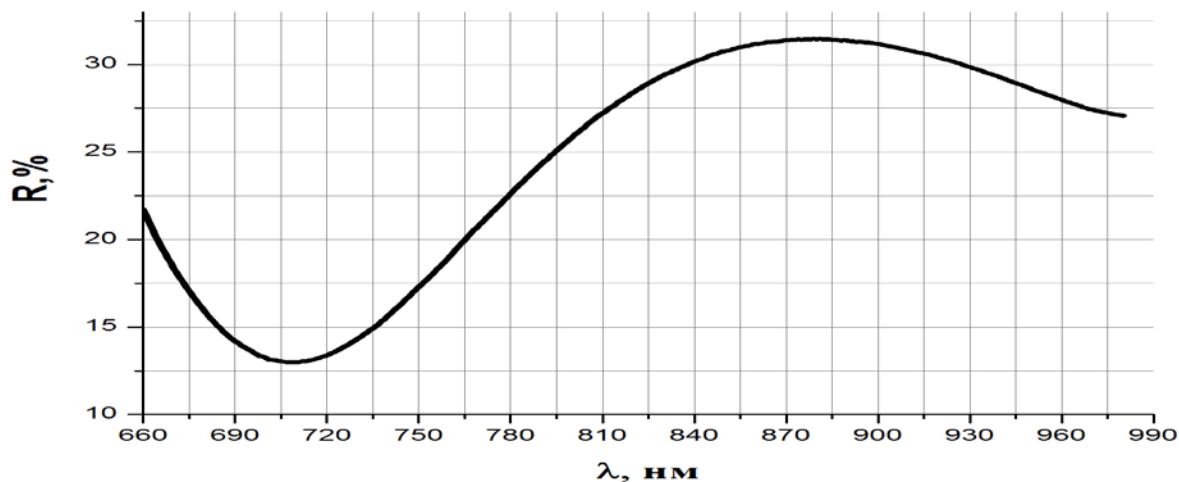
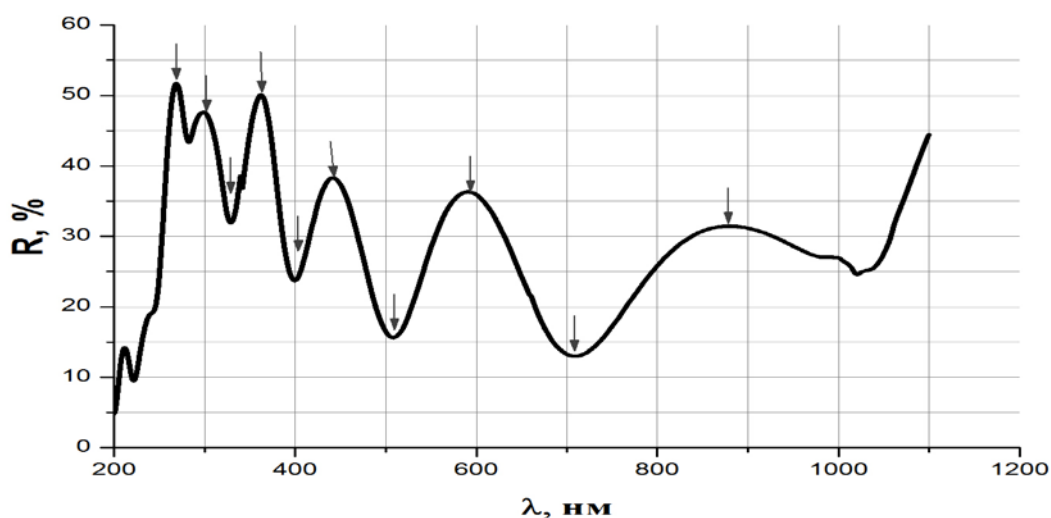


Рис. 1. Нормированные спектры отражения пленки

Этот участок спектра использовался нами для определения коэффициента преломления пленки. Для этого с помощью программы FilmManager [6] рассчитывались спектры отражения пленки с варьированием коэффициента преломления. Участок брался одинаковым для всей указанной области. Визуальное и численное совпадение с экспериментальным спектром позволило определить коэффициент преломления пленки ( $n_2$ ), который равен  $1,325 \pm 0,003$  для длины волны 880 нм.

Определение геометрической толщины пленки ( $h_2$ ), коэффициентов преломления ( $n_2$ ) и порядка интерференции проводилось по положению экстремумов. На рис. 2 изображен нормированный спектр отражения пленки  $BaF_2$  на подложке  $Si$ . Черточками указаны точки экстремумов в спектре.

Рис. 2. Спектр отражения пленки  $BaF_2$  на подложке  $Si$ 

Положение экстремумов  $R_1$  соответствует положениями оптической толщины пленки  $n_2 h_2$ , кратным падающего света  $\lambda / 4$ , т. е.  $n_2 h_2 = k \lambda / 4$  [3], где  $m=1, 2, 3, \dots$  – порядок интер-

ференции.  $k$  отсчитывается от первого экстремума и нумерация увеличивается в сторону уменьшения длины волны. Т. е.

$$\lambda_{\text{экт } m} = 4n_2h_2 / m \quad (1)$$

Используя систему уравнений (1) можно найти толщину пленки:

$$h_2 = (\lambda_{\text{max}} m) / n_2 \quad (3)$$

$$h = 0,875 \cdot 4 / 1,325 = 2,6415 \text{ мкм.}$$

Для экстремума на длине волны 0,881 мкм  $n_2(2m) = 1.325$ , что совпадает со значением, полученным с помощью расчета по программе FilmManager.

Аналитически зависимость  $n$  от  $\lambda$  можно представить по формуле Коши как [7]:

$$n = A + B / (\lambda^2 - 0,028) \quad (4)$$

Экспериментальные и рассчитанные по формуле Коши значения показателя преломления приведены на рис. 3.

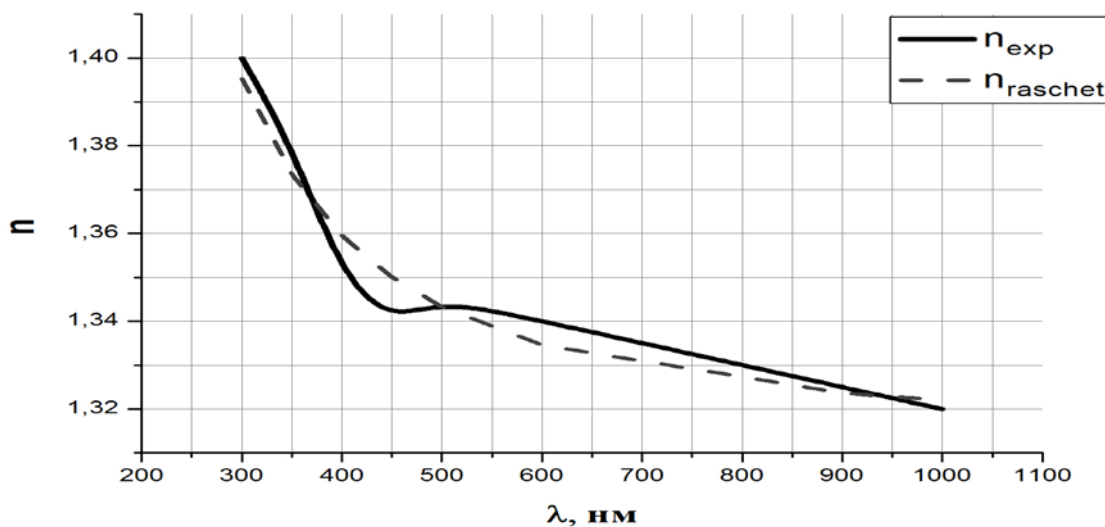


Рис. 3. Зависимость коэффициента преломления от длины волны

Результаты по определению коэффициентов преломления пленок согласуются с известными литературными данными [4, 5, 7].

#### Выводы

Предложена методика определения оптических констант пленок по спектрам отражения. В ее основе лежит численно доказанное утверждение, согласно которому у ряда пленок на определенном интервале длин волн все поглощение целиком лежит в пропускании, а значит, оптические константы могут быть определены только из спектра пропускания. Методика применена для определения оптических констант пленок BaF<sub>2</sub> в диапазоне спектра 200–1100 нм. Особенностью метода является возможность определения оптических констант только по спектрам отражения для точек, соответствующих максимумам пропускания.



**Библиографический список**

1. Андриевский Б. В., Вахулович В. Ф., Курляк В. Ю., Романюк Н. А. Определение дисперсии показателя преломления и толщины тонких пленок по спектрам отражения или пропускания // Оптика и спектроскопия. 1988. Т. 65. № 1. С. 136–140.
2. Хевенс О. С. Измерение оптических констант тонких пленок // Сб. физика тонких пленок. В 8 т. Т. / Под общ. ред. Г. Хасса и Р. Э. Тауна; перевод с англ. под ред. В. Б. Сандомирского и А. Г. Ждана. М.: Мир, 1967. С. 136–185.
3. Борн М., Вольф Э. Основы оптики. Пер. с англ. под ред. Г. П. Мотулевич М.: Наука. 1970. 856 с.
4. Технологические лазеры. Справочник Т. 2. Под общей редакцией Г. А. Абельситова. – М.: Машиностроение. 1991. – 554 с.
5. Справочник технолога оптика. Под общей редакцией С. М. Кузнецова и М. А. Окатова. – Л.: Машиностроение, 1983. – 414 С.
6. Котликов Е. Н., Новикова Ю. А. Программа синтеза и анализа интерференционных покрытий «Film Manager». Научный журнал Информационно управляющие системы № 1(68) 2014. С. 15–20.
7. Воронкова Е. М., Гречушников Б. Н., Дистлер Г. И., Петров И. П. Оптические материалы для инфракрасной техники. Наука, Москва, 336 с. 1965.

УДК 519.717

**М. Н. Соколов**

магистрант кафедры радиотехнических и оптоэлектронных комплексов

**А. Ф. Крячко**

д-р техн. наук, доц. – научный руководитель

**СРАВНИТЕЛЬНЫЙ АНАЛИЗ ДИСКРЕТНЫХ ПРЕОБРАЗОВАНИЙ  
В ЗАДАЧАХ РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ**

В цифровой обработке изображений кроме дискретных преобразований Фурье широко применяются и другие ортогональные преобразования, такие как преобразования Хаара, преобразования Уолша и др. Они находят применение в системах обработки изображений при оптической пространственной фильтрации, синтезе элементов оптоэлектроники, в изображающих системах с кодированной апертурой. Главные их применения в обработке изображений – обобщенная фильтрация для подавления помех и коррекции искаженных изображений, кодирование, выделение признаков. Общим свойством преобразований, определяющим целесообразность их применения, является простота их вычислительной реализации.

Особенности дискретного преобразования Фурье

Преобразование Фурье можно рассматривать как линейное преобразование с ядром

$$H(f, t) = \exp(-i2\pi ft), \quad (1)$$

где  $f$  – частота,  $t$  – время.

Его дискретное представление по базису

$$\{\varphi_k(t) = \text{sinc}2\pi F \left[ \left( t - \frac{k}{2F} \right) \right]\}. \quad (2)$$

где  $k$  – индекс,  $\text{sinc}2\pi F$  – отчетная базисная функция.Для сигналов с ограниченным на интервале  $(-F, F)$  спектром, справедливо представление

$$a(t) = \sum_{k=-\infty}^{\infty} a \left( \frac{k}{2F} \right) \text{sinc} \left[ 2\pi F \left( t - \frac{k}{2F} \right) \right], \quad (3)$$

где  $a$  – амплитуда сигнала.

Преобразование Фурье такого сигнала равно

$$\begin{aligned} a(f) &= \int_{-\infty}^{\infty} a(t) \exp(i2\pi ft) dt = \\ &= \sum_{k=-\infty}^{\infty} a\left(\frac{k}{2F}\right) \int_{-\infty}^{\infty} \text{sinc}\left[2\pi F\left(t - \frac{k}{2f}\right)\right] \exp(i2\pi ft) dt = \\ &= 1/2F \sum_{k=-\infty}^{\infty} a\left(\frac{k}{2F}\right) \exp\left(i2\pi f \frac{k}{2F}\right). \end{aligned} \quad (4)$$

Дискретное преобразование Фурье

$$a_s = \sqrt{\frac{2f}{T}} a\left(\frac{s}{T}\right) \exp\left(i2\pi \frac{ks}{N}\right) a_k = a\left[\frac{k-k_0}{2F}\right], \quad (5)$$

где  $N = 2TF$  (значения  $T$  и  $F$  выбраны для целого  $N$ ).

Здесь  $k_0$  выбрано так, чтобы суммирование могло производиться по  $k$  от 0 до  $N - 1$ . Тогда получим

$$a_k = 1/\sqrt{N} \sum_{k=0}^{N-1} a_k \exp\left(i2\pi \frac{k_0 s}{N}\right). \quad (6)$$

Преобразование Уолша

Подобно дискретному преобразованию Фурье можно рассматривать дискретное преобразование Уолша как дискретный аналог непрерывного преобразования сигнала по базису, составленному из функций Уолша.

Как и дискретное преобразование Фурье, дискретное преобразование Уолша строится как преобразование последовательности отсчетов сигнала по базису из отсчетов функций Уолша, взятых в дискретной последовательности точек:

$$a_s = \sum_{k=0}^{N-1} a_k \text{wal}_s(k), \quad (7)$$

где  $\text{wal}_s$  – базисная функция Уолша.

Пусть количество отсчетов сигналов  $N$  равно  $2^N$ ,  $k = 0, 1, \dots, 2^N - 1$ , количество базисных функций  $s = 0, 1, \dots, 2^N - 1$ . Значение базисной функции Уолша  $\text{wal}_s(k)$  можно записать как

$$\text{wal}_s(k) = 1/\sqrt{N} (-1)^{\sum_{j=0}^{n-1} s^j k_j}, \quad (8)$$

где  $s^j$  – разряды кода Грэя номера функции  $s$ , взятые в инверсном порядке (т. е. читаемые слева направо);  $k_j$  – разряды двоичного кода номера отсчета.

Преобразование Хаара

Дискретное преобразование Хаара строится как преобразование последовательности отсчетов сигнала  $\{a_k\}$ ,  $k = 0, 1, \dots, N - 1$ , количество которых выбирается равным целой степени 2, по базису  $\text{har}_s(k)$ , составленному из отсчетов функций Хаара:

$$a_s = \sum_{k=0}^{N-1} a_k \text{har}_s^{(n)}(k), \quad (9)$$

где  $\text{har}_s^{(n)}$  – базисная функция Хаара.

Пусть  $N = 2^N$ . Тогда

$$\text{har}_s^{(n)}(k) = 2^{\frac{s_c-n}{2}} (-1)^{K_{n-s_c-1}} \delta([k]_{n-s_c} \oplus S_{\text{mod } 2^{s_c}}), \quad (10)$$

где  $s_c$  – номер старшего равного единице разряда в двоичном представлении числа  $s$ ;  $\{K\}_{n-s_c}$  – двоичное число, составленное из  $s_c$  старших разрядов двоичного числа  $k$ ;  $\oplus$  – знак поразрядного сложения по модулю 2;  $S_{\text{mod } 2^{s_c}}$  – двоичное число, составленное из  $s_c$  младших двоичных разрядов  $s$ ;  $\delta$  – дельта-функция (символ) Кронекера.

В матричной форме преобразование Хаара описывается матрицей Хаара, строки которой составлены из функций  $\text{har}_s^{(n)}(k)$ :

$$\text{HAR}_{2^n} = \left\{ \text{har}_s^{(n)}(k) \right\}. \quad (11)$$

Вывод

Преобразования Фурье, Уолша, Хаара имеют исключительное применение в обработке изображений. Непосредственное применение этих преобразований и применение приборов, действие которых основано на их использовании, является важным для развития многих отраслей науки и техники.

### Библиографический список

1. Цифровая обработка сигналов в оптике и голографии: введение в цифровую оптику. Ярославский Л. П.
2. Введение в цифровую обработку изображений. Ярославский Л. П.
3. Преобразование Фурье, Уолша, Хаара и их применение в управлении, связи и других областях. Научное издание. – М.: Наука. Гл. ред. физ.-мат. лит., 1989. – 496 с.

УДК 004.932.2

**А. С. Суркин**

магистрант кафедры бортовой радиоэлектронной аппаратуры

**Ю. А. Корнеев**

канд. техн. наук, доц. – научный руководитель

### ОБЗОР И АНАЛИЗ МЕТОДОВ ОБНАРУЖЕНИЯ НОМЕРНЫХ ЗНАКОВ

Одним из важных этапов в задаче распознавания номерных знаков автомобилей является поиск на изображении точных координат пластины номерного знака. Такой этап обработки изображения называется локализацией номерного знака. Очевидно, что от надежности работы алгоритма локализации во многом зависит и качество всей системы распознавания номера целиком.

Алгоритмы автоматического обнаружения номерного знака можно разделить на группы, основанные на:

- анализе контуров;
- гистограммном анализе регионов;
- статистическом анализе с использованием классификаторов;
- методах сравнения с шаблонами;
- применении морфологических операторов.

В данной статье приводится обзор и анализ наиболее распространенных на сегодняшний день методов локализации пластины номерного знака, а именно: гистограммный и статистический анализ.

Гистограммный анализ основан на предположении, что пластина номерного знака является областью, содержащей внутри себя скачки яркости от черного к белому и от белого к черному [1, 3, 4, 7]. Это наглядно продемонстрировано на (рис. 1,а), где построена функция интенсивности для строки изображения соответствующей области номерного знака. Алгоритм работы данного метода заключается в том, что изображение сканируется построчно с интервалом равным половине минимальной высоты пластины знака. Для каждой строки строится функция вида:

$$f(x) = \sum_{i=0}^{x-2} |I(i+1) - I(i)| \quad (1)$$

где  $I(i)$  – значение яркости в соответствующем пикселе. Вычисляется вторая производная, максимумы которой могут совпасть с горизонтальным положением номерного знака (рис. 1,б).

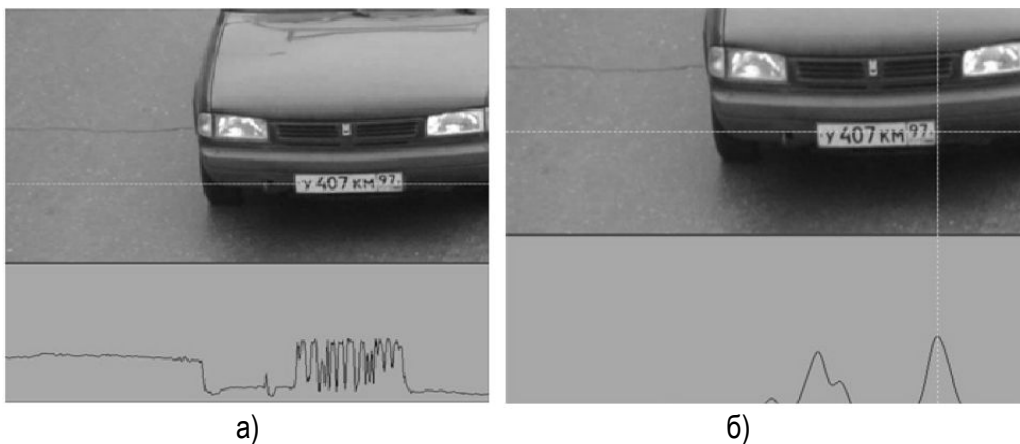


Рис. 1. Формирование пиков гистограмм горизонтальных сечений

Далее вычисляется вторая производная от функции  $f(x)$ , ее максимумы могут соответствовать горизонтальным краям пластины номерного знака (рис. 1,б).

Стоит отметить, что такой подход позволяет находить только горизонтальные координаты пластины номерного знака. Достоинством метода является высокая скорость работы. Однако данный алгоритм будет работать плохо в условиях, когда на автомобиле присутствует четко выраженная радиаторная решетка (рис. 2).

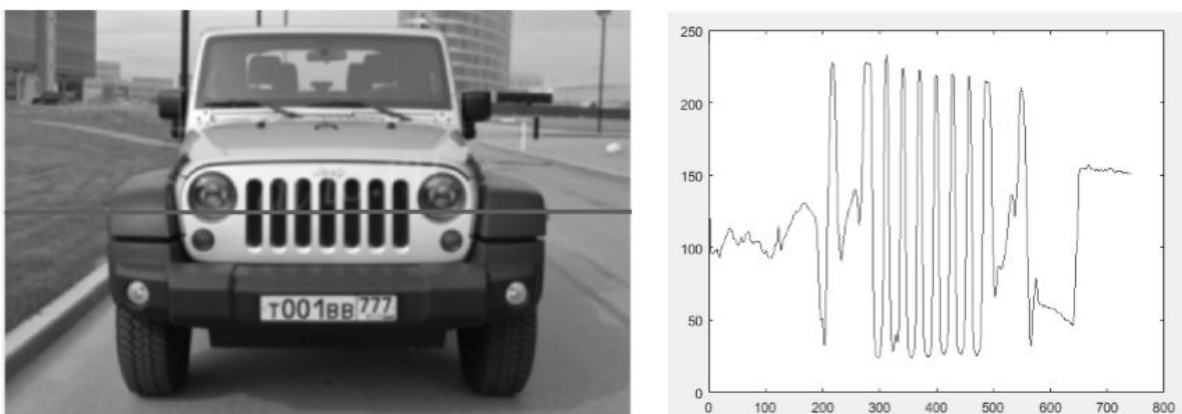


Рис. 2. Реализация сигнала, сформированного по радиаторной решетке

Существует метод [5], в рамках которого анализируется распределение яркости не отдельных строк, а всего изображения. На изображении находятся вертикальные и горизонтальные контуры, после чего строятся проекции изображения на ось  $x$  и  $y$ , которые отображают распределение нормы градиента относительно выбранной координаты (рис. 3).

Процесс выделения пластины номерного знака состоит из двух этапов:

- выделение возможной области-кандидата, в которой может находиться номер;
- нахождение координат номерного знака.

Для выделения областей-кандидатов используется проекция изображения на ось  $y$  (рис. 3). Далее, для полученной области изображения строится проекция на ось  $x$ , находится первая производная и на основе анализа максимумов и минимумов полученной функции, делается вывод о горизонтальных границах номерной пластины (рис. 4).

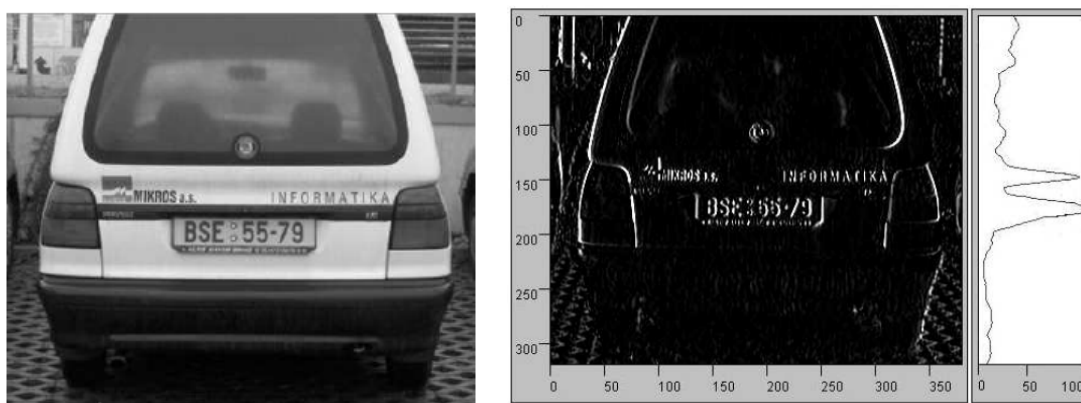


Рис. 3. Формирование гистограммы по результатам выделения границ

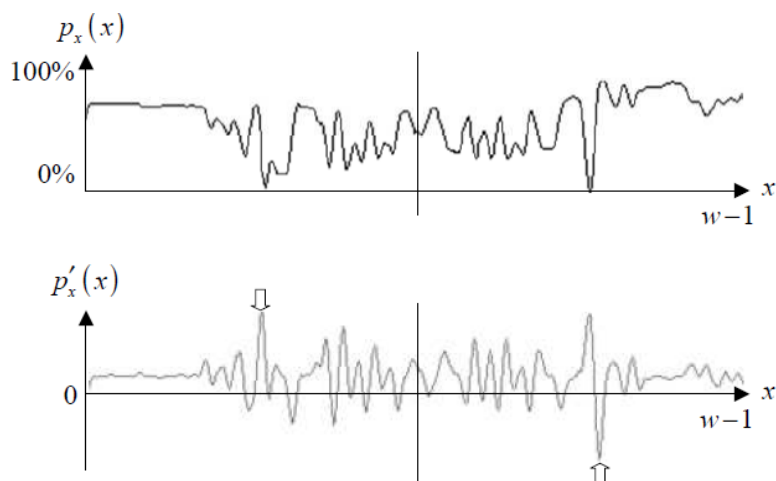


Рис. 4. Результат оценивания границ пластины номерного знака

Методы, основанные на гистограммном анализе, имеют ряд недостатков. В методе предполагается, что автомобиль по размеру должен быть сопоставим с размерами кадра. Это требование обусловлено спецификой фона, который может содержать надписи и другие детализированные объекты.

Другим перспективным методом является статистический анализ с использованием классификаторов [2, 9]. Суть метода сводится к тому, что по изображению движется окно установленного размера, и для каждой области, над которой проходит окно, вычисляется значение признака. Наличие или отсутствие искомого объекта определяется разницей между обучаемым порогом и полученным значением признака. Если в качестве признака используются примитивы Хаара, то такой метод называется алгоритмом Виолы–Джонса.

Алгоритм базируется на следующих принципах обработки изображений:

- используются изображения в интегральном представлении;
- используются признаки Хаара;
- используется бустинг для выбора наиболее подходящих признаков для искомого объекта в данной части изображения;
- все признаки поступают на вход классификатора, который дает результат «верно» либо «ложь»;
- используются каскады признаков для быстрого отбрасывания окон, в которых не найден номер.

В интегральном представлении изображения формируется матрица, совпадающая по размерам с исходным изображением. Элементы матрицы рассчитываются по формуле:

$$L(x, y) = \sum_{i=0; j=0}^{i \leq x; j \leq y} I(i, j) \quad (2)$$

где  $I(i, j)$  – яркость пикселя исходного изображения. Признаки Хаара представляются в виде разности сумм пикселей двух смежных областей внутри прямоугольника, который может занимать различные положения и масштабы на изображении. При последовательном прохождении сканирующего окна по изображению необходимо вычислить все значения признаков в данном окне, что требует значительных вычислительных затрат. В методе Виолы – Джонса признаки Хаара организованы в каскадный классификатор, представляющий собой дерево принятия решений (рис. 5). Необходимо обучить такой классификатор так, чтобы он реагировал только на определенное подмножество всех признаков.

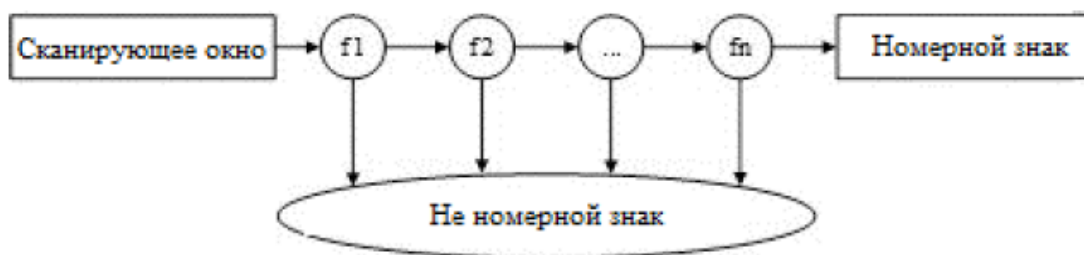


Рис. 5. Структура каскадного классификатора

Подобласти изображения, которые прошли через весь каскад, классифицируются как «номерной знак». Качество работы описанного выше метода зависит от используемых при обучении классификатора положительных и отрицательных образов-изображений, содержащих пластины номерного знака (рис. 6), и изображений фона, на котором номерной знак отсутствует, но на котором также осуществляется поиск знака.



Рис. 6. Примеры номерных знаков для обучения классификатора

Анализ современных и наиболее распространенных методов локализации номерного знака показал, что для решения этой задачи требуются алгоритмы, обладающие высокой степенью адаптивности к априорной неопределенности обрабатываемых данных. При построении таких алгоритмов нужно стараться использовать как можно меньше априорных сведений о свойствах номерного знака: его размерах, соотношении сторон, расположении в кадре, ориентации. Отказ от большей части априорной информации при построении робастного алгоритма распознавания будет обеспечивать устойчивую работу системы при самых разнообразных условиях видеонаблюдения.

### Библиографический список

1. Бусыгин Л. А. Автореферат «Локализация автомобильного номера в потоке видеоданных в режиме реального времени». Томский государственный университет, 2006 г.
2. Амосов О. С., Иванов Ю. С. «Модифицированный алгоритм локализации номерных знаков транспортных средств на основе метода Виолы-Джонса». Информатика и системы управления, 2014 г, № 1(39). Автоматизированные системы и комплексы.
3. Буренин С. Е. автореферат «Распознавание автомобильного регистрационного номерного знака с локализованной области» Томский государственный университет 2006 г.
4. Колюжнов В. В. «Новый метод локализации номерных знаков транспортных средств для систем реального времени». Вестник СибГУТИ, 2012 г, № 4.
5. Martinsky O. «Algorithmic and mathematical principles of automatic number plate recognition systems» B.Sc. Thesis BRNO University of Technology, 2007.
6. Szeliski R. Computer vision: algorithms and applications. Springer, 2010 г.
7. Очаковский Д. О., Елизаров А. И. Локализация автомобильных номеров методом разбиения изображения на блоки. Доклады ТУСУРа, № 2 (18), часть 1, июнь 2008 г.
8. Коваленко Е. Н., Сытник А. В. «Методы выделения номерного знака на изображении». НИИ прикладной электроники, Национальный технический университет Украины «КПИ» г. Киев.
9. Bishop C. M. Pattern recognition and machine learning. Springer, 2006 г.

УДК 629.05

**Я. С. Хомутов**

магистрант кафедры медицинской радиоэлектроники

**Г. И. Худяков**

д-р техн. наук, проф. – научный руководитель

### СИСТЕМА ТЕРРИТОРИАЛЬНОГО ВЗАИМОДЕЙСТВИЯ ГРУППЫ ПОДВИЖНЫХ АБОНЕНТОВ

Для организованной и скоординированной работы большой группы подвижных объектов, имеющих общие цели, остается очевидной необходимость обмена различного рода информацией. Это может быть информация о географических координатах, голосовые сообщения, графиче-

ские сообщения и т. п. Для обмена подобного рода информацией необходимо организовать сеть обмена данными.

В настоящее время такие сети представляют собой многофункциональные автоматизированные цифровые системы распределения данных и привязки к местности на основе пакетной радиосвязи. На них возлагается множество задач, а именно: сбор, распределение и передача информации разного типа по сети в формате времени близком к реальному; дистанционное управление тактическими объектами, выполняющими определенные операции; целеуказания и доведение результатов целеуказаний до всех пользователей; передача сообщений о тактической обстановке и речевых сообщений в интересах всех абонентов, определение географических координат. Следует добавить, что такие сети должны обладать достаточной защитой от преднамеренных помех.

Учитывая современные тенденции развития радиотехнических систем, система распределения данными должна обеспечивать бесперебойную высокоскоростную помехозащищенную и криптозащищенную связь компьютер-компьютер для всех абонентов сети.

Система распределения данных является радиосетью с достаточно гибкой топологией, в которой взаимодействие абонентов организуется на принципах множественного доступа с временным разделением каналов. Использование методов с частотным разделением каналов не представляется возможным ввиду достаточно широкой полосы частот, т. к., работая с большой группой абонентов, порядка ста единиц, необходимо иметь либо многоканальный, либо широкополосный приемник, что не реализуемо для системы передачи информации в формате времени близком к реальному. Использование кодового разделения доступа невозможно ввиду большого динамического диапазона, что требует достаточно высоких вычислительных мощностей обрабатывающей аппаратуры, что не реализуемо для подобного рода систем [1].

Множественный доступ с временным разделением каналов требует кадровой и цикловой синхронизации, что подразумевает наличие собственной подсистемы синхронизации. Глобальная синхронизация шкал времени абонентов является наилучшим способом организации их совместной работы. При построении системы на основе глобального абсолютного синхронизма один из абонентов по определенной иерархии выступает в качестве глобального синхронизатора, и его время ретранслируется по сети всем абонентам [1].

С развертыванием спутниковых радионавигационных систем II поколения, например, GPS и ГЛОНАСС, все больше задач навигации стали возлагаться на эти системы. Так же спутниковые радионавигационные системы являются источниками эталонного времени, что существенно упрощает и ускоряет процесс синхронизации абонентов.

Аналогом системы распределения данных является система JTIDS, использующаяся вооруженными силами США и союзниками для организации сетей обмена информацией. Система является одной из аппаратно-программных реализаций сети Link 16. Это коммуникационная, навигационная и идентификационная система, которая поддерживает обмен данных между тактическим командованием, самолетами, кораблями и наземными подразделениями. Система работает в ультракоротковолновом диапазоне 960–1215 МГц. Разделение каналов реализовано на методе множественного доступа с временным разделением. Система обеспечивает передачу текстовых сообщений, графических изображений, передачу голоса и обладает возможностями относительной навигации. Одной из основных особенностей этой системы является отсутствие управляющей станции. Связь осуществляется по линии прямой видимости до 900 км. Длительность кадра – 7,8125 мс. В течение одного цикла передачи информации каждому участнику информационного обмена выделяется определенный период времени (временной слот, кадр), в течение которого он производит передачу данных в монопольном режиме. Каждому участнику обмена присваивается уникальный восьмеричный пятизначный номер. Система идентифицирует участника обмена и предопределяет расположение временных слотов, в которых абонент передает и принимает информацию. Когда иницируется передача данных по каналу сети, один из участников обмена передает «сетевую временную метку». Получив временную метку, все



остальные участники обмена синхронизируются, после чего работа сети не зависит от участия в ней какого-либо конкретного устройства [2].

Несмотря на все достоинства выше описанной системы, возникает необходимость создания новых систем. Необходимость обуславливается экономическими факторами, так как данная система имеет множество дополнительных функций, которые не используются в данной постановке задачи, и которые определяют конечную стоимость системы в целом.

В своей выпускной работе я исследую возможные варианты построения системы, решающей задачи относительной навигации и обмена данными группы летательных аппаратов.

Требуется организовать сеть обмена данными для группы летательных аппаратов количеством до ста штук одновременно. Как и другие системы обмена данными в режиме реального времени, для данной системы предлагается использовать метод множественного доступа с временным разделением каналов. Спутниковая радионавигационная система является эталоном единого времени, и синхронизация абонентов в сети осуществляется в рамках этой временной сетки. Дальность работы системы – 150 км.

Предлагается использовать длительность кадра равную 10 мс. Помимо полезной информации в кадре необходимо наличие защитных интервалов в начале и конце кадра. Длительность защитных интервалов определяется как удвоенное отношение максимальной дальности распространения сигналов к скорости света [1]. Таким образом, при максимальной дальности в 150 км, длительность защитных интервалов будет равна 1 мс. Минимальный объем навигационной информации составляет порядка 200 бит. Длительность одной информационной посылки составляет 2,4 мс, и чтобы передать 200 бит информации необходимо 7,2 мс [1]. С учетом 2 защитных интервалов по 1 мс каждый, получается 9,2 мс. Таким образом, 10 мс является практически минимально возможной длительностью кадра. Длительность кадра, максимальное число абонентов и период обновления информации являются величинами взаимосвязанными. При максимальном числе абонентов до 100 штук и длительности кадра в 10 мс предлагается использовать период обновления информации 2 с [1]. Конечно, расчет периода обновления информации осуществляется, руководствуясь максимальным числом абонентов, однако, в реальных условиях их значительно меньше. В этом случае не обязательно менять топологию сети, а можно позволить абонентам занимать для своей работы несколько слотов, что повысит темп обновления информации. Эти два обстоятельства позволяют создать сеть с очень гибкой топологией, в свою очередь, требующей эффективных алгоритмов обнаружения конфликтов и алгоритмов назначения каналов.

Для назначения временных интервалов предлагается использовать несколько режимов, а именно два режима: режим статического закрепления временных интервалов и режим с предоставлением на конкурентной основе. В режиме статического закрепления каждый кадр закрепляется за определенным абонентом на время его работы в сети в соответствии с его приоритетом. Это стандартный вариант закрепления временных интервалов за пользователем. Такой режим имеет наименьшую гибкость, но наиболее прост в технической реализации. В режиме предоставления на конкурентной основе каждый абонент использует свой кадр по мере надобности, выбирая его случайным образом. Вполне возможно, что некоторый интервал будет одновременно занят несколькими абонентами. Данный режим требует высокоэффективных алгоритмов обнаружения и устранения конфликтов, но является достаточно гибким.

Организация работы в единой временной сетке осуществляется благодаря спутниковым системам навигации, задающим единые метки времени для всех абонентов. Так же учитывается возможность пропадания сигналов спутниковых систем. Это обстоятельство требует специальных алгоритмов синхронизации абонентов в сети.

Исходя из выше сказанного, перспективной системой взаимодействия абонентов будет являться сеть с временным распределением доступа, использующая два основных метода назначения каналов. Система должна иметь алгоритмы синхронизации при наличии и отсутствии сигналов спутниковых радионавигационных систем.

В заключение хотелось бы сказать, что работа находится на начальной стадии. В дальнейшем планируется исследовать возможность уменьшения периода обмена информацией и времени кадра, а также исследовать предельные расширения по дальности и числу абонентов, решить задачи, касающиеся путей построения системы, выбора варианта построения системы, исходя из технических требований, а также задачи оптимизации выбранного варианта. В результате представляется синтезировать сеть обмена данными с требуемыми характеристиками.

### **Библиографический список**

1. Орлов В. К., Герчиков А. Г., Чернявский А. Г. Локальные радиотехнические системы межсамолетной навигации // СПбГЭТУ «ЛЭТИ», 2011. 123 с.
2. Клименко И. Н., Кисель В. В., Гончар А. Н. Объединенная система распределения тактической информации JTIDS // Зарубежная радиоэлектроника. 1988. № 5. С. 85–96.

УДК 621.396.669.1

**С. А. Хухунин**

магистрант кафедры медицинской радиоэлектроники

**В. Ф. Михайлов**

д-р техн. наук, проф. – научный руководитель

## **ЗАЩИТА ПЕРСОНАЛЬНЫХ ДАННЫХ В ТЕЛЕМЕДИЦИНЕ ПРИ ИСПОЛЬЗОВАНИИ КОРОТКИХ ВОЛН**

Телемедицина – это использование достижений телекоммуникационных технологий в здравоохранении. Телемедицина позволяет поднять эффективность лечения и диагностики на качественно новый уровень. С помощью телемедицинских технологий можно, например, удаленному больному, оказать высококвалифицированную медицинскую помощь. В данной работе используется коротковолновый диапазон для обеспечения эффективной связи между абонентами. Короткие волны способны обеспечить связь на расстоянии до 6000 км и более. Стоимость же коротковолновых радиоканалов на порядок ниже спутниковых. Актуальность задачи состоит в том, что медицинские данные и персональные данные пациента должны быть обеспечены безопасностью при передаче [1–3].

Для решения данной задачи решено было разработать программу на основе Сети Фейстеля. Сеть состоит из ячеек, называемых ячейками Фейстеля. На вход каждой ячейки поступают данные и ключ. На выходе каждой ячейки получают измененные данные и измененный ключ. Все ячейки однотипны, и говорят, что сеть представляет собой определенную многократно повторяющуюся структуру. Ключ выбирается в зависимости от алгоритма шифрования/расшифрования и меняется при переходе от одной ячейки к другой. При шифровании и расшифровании выполняются одни и те же операции; отличается только порядок ключей. Ввиду простоты операций сеть Фейстеля легко реализовать как программно, так и аппаратно [4–6].

Создание таблиц замены и перестановки бит для каждой ячейки данных усилит криптостойкость алгоритма (рис. 1).

Разработанная программа предназначена для шифрования и расшифрования данных пациента. Для создания данного программного обеспечения был использован язык программирования BorlandDelphi, а так же MEXcel для создания таблиц замены и перестановки. Выбор данного языка программирования удовлетворяет всем поставленным требованиям, а именно:

- простота;
- функциональность;
- эффективность;
- скорость разработки.

## РАДИОЭЛЕКТРОНИКА, ЭЛЕКТРОНИКА И СВЯЗЬ

1 ячейка			2 ячейка			3 ячейка		
0000	=	0100	0000	=	0110	0000	=	0101
0001	=	0101	0001	=	0111	0001	=	0110
0010	=	0110	0010	=	0000	0010	=	0111
0011	=	0111	0011	=	0001	0011	=	0000
0100	=	0000	0100	=	0010	0100	=	0001
0101	=	0001	0101	=	0011	0101	=	0010
0110	=	0010	0110	=	0100	0110	=	0011
0111	=	0011	0111	=	0101	0111	=	0100

Рис. 1. Таблицы замены и перестановки бит для каждой ячейки данных

Помимо этого программа позволяет защитить персональные данные пациента, что является важным этическим моментом. Вопросы этики и правовой защиты информации в телемедицине рассматриваются многими авторами и подробно изложены в работах В. В. Овчинникова (2013), К. В. Шматова, Д. В. Мерекина и В. М. Леванова (2015).

В заключение нужно отметить, что основным преимуществом предложенного программного обеспечения является его простота в использовании и возможность интеграции в другие медицинские системы.

### Библиографический список

1. Блажис А., Дюк В. Телемедицина. СПб.: Спецлит, 2001. 137 с.
2. Лучевое приближение и вопросы распространения радиоволн / Под ред. М. Кияновского. М.: Наука, 1971. 311 с.
3. Мигулин В. Как исследуют ионосферу // Радио, 1981. № 5–6. С. 33–34.
4. Информационная безопасность [Электронный ресурс] – режим доступа [http://ru.wikipedia.org/wiki/Информационная\\_безопасность](http://ru.wikipedia.org/wiki/Информационная_безопасность) (Дата обращения: 10.04.2015).
5. Дмитриев А. Системы защиты информации // Мир ПК. 2001. № 05. С. 8.
6. Баричев С. Г. и др. Основы современной криптографии. М.: Горячая линия – Телеком, 2002. 175 с.
7. Овчинников В. В. Телемедицина: проблемы правового регулирования // Тихоокеанский медицинский журнал. 2013. № 3. С. 12–15.
8. Шматов К. В. и др. Телемедицина в здравоохранении. М.: Вена, 2015. 108 с.

УДК 53.072

**А. В. Шатохин**

магистрант кафедры медицинской радиоэлектроники

**Г. И. Худяков**

д-р техн. наук, проф. – научный руководитель

### КАНАЛ СВЯЗИ ВОЛОКОННО-ОПТИЧЕСКОГО ТЕПЛООВОГО ПОЖАРНОГО ИЗВЕЩАТЕЛЯ

Оптические волокна используются в сетях передачи данных вместо металлических проводов, т. к. сигналы проходят по ним с меньшими потерями, и они совершенно не подвержены действию внешних электромагнитных излучений; они имеют меньший вес и стоимость в эквивалентном информационном применении. В наши дни оптическое волокно считается самой совершенной средой для передачи больших потоков информации на большие расстояния.

Применение оптических волокон не ограничивается системами связи. Активно развивается научно-прикладная область, связанная с разработкой волоконно-оптических датчиков различных физических величин. Это, например, датчики радиации, температуры, механических напряжений,

давления. И они могут найти применение в самых широких областях промышленности и хозяйственной деятельности [1].

Особое внимание уделяется волоконно-оптическим датчикам для контроля температуры. Они являются пожарными извещателями и обладают целым рядом преимуществ по сравнению с другими типами подобных устройств. Такой датчик незаменим во многих направлениях современной промышленности. Это устройство имеет малый вес и размер – длина чувствительного элемента составляет 2–10 мм при диаметре 0,1–0,2 мм. Его можно применять в недоступных для других датчиков областях, он нечувствителен к электромагнитным помехам.

Такой датчик может контролироваться на расстоянии до нескольких километров и является абсолютно взрыво- и пожаробезопасным. И, наряду с этими качествами, он обладает высокой чувствительностью. Пока у этих устройств нет аналогов, которые могли бы так же успешно использоваться в нефтяной и газовой промышленности, сушильных установках и различного рода печах, например, в турбинах и генераторах, двигателях, различных областях медицины и инженерии, авионавтике и космонавтике [1].

Все волоконно-оптические датчики можно условно разделить на три типа: точечные, распределенные и квазираспределенные.

Точечные, как следует из названия, позволяют измерить и контролировать параметры в определенной точке объекта, как и большинство других типов неволоконных датчиков. Такие датчики обладают высокой точностью измерения и малым размером. В зависимости от типа сенсорного элемента локализация датчиков может достигать 0.1 см<sup>2</sup>, как в случае брэгговского чувствительного элемента.

Бесспорным преимуществом распределенных датчиков является возможность осуществления непрерывного контроля параметров по длине (объему) объекта в любой его точке, где установлен сенсорный световод. Принцип действия таких сенсорных систем основан на анализе изменения параметров по длине световода и на нелинейных эффектах. Недостатком распределения измеряемого параметра по длине является относительно невысокая точность измерения величины и относительно невысокая точность определения локализации возмущения (несколько метров по длине). Распределенные сенсорные системы могут быть использованы для контроля больших территорий, позволяют анализировать градиенты температур на больших объектах, котлах и т. п.

Система на квазираспределенных датчиках объединяет преимущества первых двух схем. Квазираспределенный датчик представляет собой массив точечных сенсорных элементов, как правило, на основе внутриволоконных решеток, объединенных одним общим световодом. Каждый элемент обладает своими уникальными характеристиками, что позволяет проанализировать его состояние независимо от других сенсорных элементов. Точность таких систем определяется точностью отдельных датчиков, а массив может объединять до 100 и более элементов. Сенсорные массивы позволяют проводить мониторинг сложных объектов, анализировать градиент распределения температуры, контролировать до 100 и более точечных объектов. Причем для этого используется только один волоконный световод и анализатор. Именно квазираспределенные системы по сравнению с электрическими аналогами обладают малым весом и размером, что особенно важно для авиации и космонавтики [2].

В данной работе рассмотрен канал связи волоконно-оптического теплового пожарного извещателя.

Температурный датчик порогового действия, может быть применен в качестве пожарного извещателя. Основной областью применения температурного датчика порогового действия является достоверное обнаружение очагов возгораний по температуре окружающей среды в качестве пожарного извещателя.

Рассмотрим датчик порогового действия. Структурная схема такого датчика приведена на рисунке.

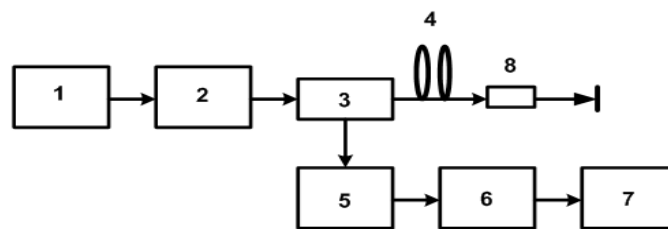


Рис. Структурная схема датчика порогового действия

Устройство состоит из источника оптического излучения 1, возбуждаемого коротким импульсом от задающего генератора 2, волоконно-оптического направленного ответвителя 3, оптического волокна 4, фотоприемника 5, блока обработки информационного сигнала 6 и блока индикации 7. Вдоль оптического волокна равномерно расположены чувствительные элементы 8.

Чувствительный элемент реагирует на повышение температуры. Использован эффект отражения оптического излучения, распространяющегося в оптическом волокне от неоднородностей оптического волокна, которые создаются специальным микроизгибным элементом. Настройка датчиков производится при температуре окружающей среды 20°C. Оптоволокно с помощью регулировочных винтов зажимается так, чтобы потери в таком чувствительном элементе составляли 5–10%. Отражение оптического излучения практически нет.

Таким образом, в штатном режиме лазер излучает импульсы, и они беспрепятственно проходят через чувствительный элемент, отражаясь лишь от концевой сколы волокна – по этому отражению устройство постоянно проверяет свою работоспособность и целостность оптического волокна.

Но при возникновении внештатной ситуации (пожара), при возрастании температуры, до значения 70°C, величина отражения значительно увеличивается и составляет порядка 5%, что фиксируется приемным устройством.

Далее блок обработки сигналов сравнивает напряжение сигнала с выхода фотоприемника с определенным порогом. И на блоке индикации отображается полученная информация.

В данной работе проведен обзор состояния волоконно-оптических датчиков температуры, который показал, что это направление является актуальным и активно развивается. В настоящее время разработан и широко используется целый ряд волоконно-оптических датчиков, использующих различные принципы работы, каждый из которых имеет свои преимущества и недостатки, что позволяет подобрать для каждой конкретной ситуации наиболее подходящий вариант.

Представленные датчики имеют ряд важных преимуществ перед традиционными электрическими датчиками. Отсутствие электрического питания в световоде и чувствительных элементах, возможность измерения температуры в условиях мощных электромагнитных полей, радиации, агрессивных химических сред без потери заданной точности, возможность эксплуатации в условиях пожаро- и взрывоопасности без создания аварийных ситуаций, малые геометрические габариты и многое другое выгодно выделяет волоконно-оптические датчики среди аналоговых электрических устройств.

### Библиографический список

1. Томашук А. Л., Голант К. М., Забейайлов М. О. Разработка волоконных световодов для применения при повышенном уровне радиации // Волоконно-оптические технологии, материалы и устройства. 2001. № 4. С. 52–65.
2. Гармаш В. Б., Егоров Ф. А., Коломиец Л. Н., Неугодников А. П., Поспелов В. И. Возможности, задачи и перспективы волоконно-оптических измерительных систем в современном приборостроении // Фотон-Экспресс. 2005. № 6. С. 128–140.

## Вычислительные системы и программирование

УДК 629.735.33

**В. А. Алексеев**

магистрант кафедры компьютерной математики и программирования

**В. И. Сеньченков**

д-р техн. наук, проф. – научный руководитель

### ОСНОВЫ ПРОГРАММНОЙ РЕАЛИЗАЦИИ БАЗОВЫХ МЕТОДОВ РЕГРЕССИОННОГО АНАЛИЗА

Регрессионный анализ представляет собой совокупность методов обработки экспериментальных данных, позволяющих определить и исследовать форму зависимости между переменными, имеющими определенную статистическую связь. Форма зависимости вычисляется в виде уравнения регрессии, которое задает исследуемую регрессионную модель, после чего данная модель подлежит проверке на достоверность и соответствие экспериментальным данным [1]. Основным методом для вычисления оценок коэффициентов уравнения регрессии является метод наименьших квадратов, позволяющий минимизировать сумму квадратов отклонений экспериментальных значений от значений, полученных через уравнение регрессии. В таком случае наилучшую аппроксимацию исследуемой зависимости будет давать то уравнение регрессии, при котором сумма квадратов отклонений

$$S = \sum_{i=1}^n (y_i - f(x_{i1}, x_{i2}, \dots, x_{im}))^2 \quad (1)$$

будет иметь наименьшее значение. В выражении (1) используются следующие обозначения:  $S$  – сумма квадратов отклонений экспериментальных значений от значений, полученных через уравнение регрессии;  $n$  – количество наблюдений в массиве экспериментальных данных;  $m$  – количество независимых переменных;  $f()$  – функция регрессии [2].

К основным методам регрессионного анализа, помимо метода наименьших квадратов стоит отнести процедуру проверки уравнения регрессии на соответствие экспериментальным данным, а также процедуру по исключению статистически незначимых коэффициентов из уравнения регрессии.

Проверка уравнения на соответствие экспериментальным данным происходит в том случае, если выполняется условие:

$$F = \frac{\tilde{\sigma}^2}{\tilde{\sigma}_1^2} = \frac{\frac{\sum_{i=1}^n (y_i - \bar{y})^2}{n-1}}{\frac{\sum_{i=1}^n (y_i - \tilde{y}_i)^2}{n-k-1}} > F_{(\alpha; n-1; n-k-1)}, \quad (2)$$

где  $\tilde{\sigma}^2$  – общая дисперсия результатов наблюдений;  $\tilde{\sigma}_1^2$  – остаточная дисперсия, которая характеризует рассеивание результатов наблюдений относительно регрессионной зависимости;

$\bar{y}$  – среднее значение зависимой переменной  $y$  по массиву данных;  $\tilde{y}_i$  – значение переменной  $y$ , полученное через уравнение регрессии в  $i$ -м опыте;  $k$  – количество несвободных членов уравнения регрессии;  $\alpha$  – уровень значимости предположения об адекватности уравнения. В выражении (2) значение  $F$  является наблюдаемым значением  $F$ -критерия для рассматриваемого уравнения регрессии, а значение  $F_{(\alpha; n-1; n-k-1)}$  определяет критическое значение  $F$ -критерия для этого уравнения. Критическое значение определяется по таблице критических точек распределения Фишера с учетом степеней свободы  $f_1 = n - 1$  и  $f_2 = n - k - 1$ .

Исключение статистически незначимых коэффициентов из уравнения регрессии происходит по результатам анализа корреляционной матрицы вектора оценок коэффициентов уравнения регрессии:

$$K_{\tilde{A}[k+1]} = \tilde{\sigma}_1^2 \left( F_{[k+1; n]}^T F_{[n; k+1]} \right)^{-1}. \quad (3)$$

На главной диагонали матрицы (3) будут находиться дисперсии оценок коэффициентов регрессии:

$$\text{diag} K_{\tilde{A}} = \left( \tilde{\sigma}_{\tilde{a}_0}^2, \tilde{\sigma}_{\tilde{a}_1}^2, \dots, \tilde{\sigma}_{\tilde{a}_k}^2 \right). \quad (4)$$

На основе дисперсий из выражения (4) вычисляются оценки средних квадратических отклонений коэффициентов  $\tilde{a}_0, \tilde{a}_1, \dots, \tilde{a}_k$ :

$$\tilde{\sigma}_{a_0} = \sqrt{\tilde{\sigma}_{\tilde{a}_0}^2}, \tilde{\sigma}_{a_1} = \sqrt{\tilde{\sigma}_{\tilde{a}_1}^2}, \dots, \tilde{\sigma}_{a_k} = \sqrt{\tilde{\sigma}_{\tilde{a}_k}^2}, \quad (5)$$

где  $k$  – количество несвободных членов уравнения регрессии.

Оценки квадратических отклонений из выражения (5) используются при проверке коэффициентов уравнения регрессии на значимость по условию:

$$t_j = \frac{|\tilde{a}_j|}{\tilde{\sigma}_{\tilde{a}_j}} > t_{(\alpha; n-k-1)}, \quad j = \overline{0, k} \quad (6)$$

где  $t_j$  – наблюдаемое значение  $t$ -критерия для коэффициента  $\tilde{a}_j$ . Критическое значение  $t_{(\alpha; n-k-1)}$  берется из таблицы критических точек распределения Стьюдента с учетом уровня значимости предположения о значимости коэффициента  $\alpha$  и с учетом степени свободы  $f_2 = n - k - 1$ . Коэффициент уравнения регрессии признается значимым, если для него выполняется условие (6) [3].

Следует отметить, что вычисление показателей из условий (2) и (6) возможно, в первую очередь, для функций регрессии, имеющих полиномиальный тип. Для остальных типов функций в качестве меры качества описания исследуемой регрессионной зависимости может выступать сумма квадратов отклонений  $S$  (выражение 1), а проверка коэффициентов уравнения на значимость может не выполняться. Вычисление показателей из условий (2) и (6) для неполиномиальных типов функций возможно не всегда и зависит от того, можно ли в таких функциях определить количество связей, накладываемых функцией регрессии на массив экспериментальных данных.

Вычисление оценок коэффициентов уравнения регрессии можно выполнять на основе системы линейных уравнений, либо на основе матричных расчетов. Для программной реализации можно использовать матричный метод расчетов, когда вычисление оценок коэффициентов для функций регрессии полиномиального типа происходит по выражению:

$$\tilde{A}_{<k+1>} = \left( F_{[k+1; n]}^T X_{[n; k+1]} \right)^{-1} F_{[k+1; n]}^T Y_{<n>} \quad (7)$$

где  $\tilde{A}_{<k+1>}$  – вектор оценок коэффициентов уравнения регрессии;  $F_{[n; k+1]}$  – матрица значений базисных функций;  $Y_{<n>}$  – вектор значений зависимой переменной.

Для функций регрессии, имеющих неполиномиальный тип, требуется приведение к линейному типу. Для экспоненциальной функции регрессии это происходит путем логарифмирования, и оценки уравнения регрессии в этом случае можно вычислить по выражению:

$$\tilde{A}_{<k+1>} = \left( F_{[k+1; n]}^T F_{[n; k+1]} \right)^{-1} F_{[k+1; n]}^T \ln Y_{<n>} \quad (8)$$

Выяснив сущность ряда основных методов регрессионного анализа, можно переходить к описанию структуры программной реализации этих методов. При разработке структуры может быть применен объектно-ориентированный подход, основная цель которого – выражать понятия предметной области решаемой задачи при помощи типов данных, определяемых разработчиком. В этом случае на основе таких пользовательских типов данных можно создавать объекты, призванные моделировать свойства и поведение определенных фрагментов решаемой задачи, объединяя в одно целое данные и методы, относящиеся к таким объектам. Данные объекта задают его характеристики, а методы – операции, которые применяются для обработки этих данных. Взаимодействие объектов также обеспечивается при помощи методов. Пользовательские (введенные разработчиком) типы реализуются в виде классов, а каждый объект является конкретным представителем класса, имеющим свои, индивидуальные значения данных, которые имеет класс объекта [4].

В случае программной реализации вышеописанных методов регрессионного анализа объектно-ориентированную структуру для программы можно на примере однофакторного анализа представить следующим образом (табл. 1):

Таблица 1

Объектно-ориентированная структура программной реализации методов регрессионного анализа

Обозначение класса	Основные методы	Назначение класса
Калькулятор	<ol style="list-style-type: none"> <li>1. Создание массива экспериментальных данных</li> <li>2. Выделение в массиве векторов значений зависимой и независимой переменной</li> <li>3. Формирование матрицы базисных функций <math>F_{[n; k+1]}</math></li> <li>4. Формирование матрицы <math>F_{[k+1; n]}^T</math>, вызов метода перемножения этой матрицы с матрицей базисных функций.</li> <li>5. Вызов метода вычисления определителя матрицы <math>F_{[k+1; n]}^T F_{[n; k+1]}</math></li> <li>6. Определение матрицы алгебраических дополнений для матрицы <math>F_{[k+1; n]}^T F_{[n; k+1]}</math></li> <li>7. Определение матрицы <math>F_{[k+1; n]}^T Y_{&lt;n&gt;}</math></li> <li>8. Определение результата умножения матрицы алгебраических дополнений и матрицы <math>F_{[k+1; n]}^T Y_{&lt;n&gt;}</math></li> <li>9. Вычисление коэффициентов уравнения регрессии</li> <li>10. Формирование уравнения регрессии</li> <li>11. Вычисление суммы квадратов отклонений <math>S</math></li> <li>12. Вычисление общей дисперсии <math>\tilde{\sigma}^2</math></li> <li>13. Вычисление остаточной дисперсии <math>\tilde{\sigma}_1^2</math></li> <li>14. Вызов процедуры проверки уравнения по <math>F</math>-критерию</li> </ol>	Управление ходом вычисления всех процедур



ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ

	<p>15. Вычисление оценок средних квадратических отклонений коэффициентов <math>\tilde{\sigma}_{a_0}, \tilde{\sigma}_{a_1}, \dots, \tilde{\sigma}_{a_k}</math></p> <p>16. Вызов процедуры проверки коэффициентов на статистическую значимость</p> <p>17. Вызов процедуры построения нового уравнения после исключения незначимых коэффициентов</p> <p>18. В зависимости от наличия незначимых коэффициентов повторный вызов процедуры проверки нового уравнения по <math>F</math>-критерию</p>	
Матрица	<p>1. Метод умножения матриц</p> <p>2. Метод вычисления определителя заданной матрицы</p> <p>3. Метод по вычислению обратной матрицы по заданной матрице</p>	Содержит операции с матрицами
Адекватность по $F$ -критерию	<p>1. Метод вычисления наблюдаемого и критического значения <math>F</math>-критерия для первоначального уравнения регрессии</p> <p>2. Метод повторного вычисления <math>F</math>-критерия в случае проверки адекватности для уравнения, из которого исключены незначимые коэффициенты</p>	Вычисляет наблюдаемые и критические значения $F$ -критерия
Таблица критических точек распределения Фишера	<p>1. Метод возврата критического значения <math>F</math>-критерия</p>	Содержит таблицы критических точек распределения Фишера для уровней значимости $\alpha = 0,01$ и $\alpha = 0,05$
Значимость по $t$ -Критерию	<p>1. Вычисление наблюдаемого и критического значения <math>t</math>-критерия для каждого коэффициента уравнения регрессии</p> <p>2. Формирование нового уравнения регрессии после исключения незначимых коэффициентов.</p>	Вычисляет наблюдаемые и критические значения $t$ -критерия, формирует новые уравнения регрессии
Таблица критических точек распределения Стьюдента	<p>1. Метод возврата критического значения <math>t</math>-критерия</p>	Содержит таблицу критических точек распределения Стьюдента
Окно вывода информации для случая парной полиномиальной регрессии	<p>1. Вывод информации по уравнению регрессии, тип функции которого – полиномиальная функция, которая выражает зависимость между зависимой и независимой переменной. Степень полинома может варьироваться.</p>	Вывод информации по исследуемой модели регрессии
Окно вывода информации для случая парной экспоненциальной регрессии	<p>1. Вывод информации по уравнению регрессии, тип функции которого – экспоненциальная функция, которая выражает зависимость между зависимой и независимой переменной. Степень показателя степени функции может варьироваться.</p>	Вывод информации по исследуемой модели регрессии
Окно указания параметров регрессии	<p>1. Основные методы – обработка полей, в которых указаны параметры исследуемой модели (уровни значимости, тип функции, степень функции)</p>	Окно, в котором происходит выбор класса функции, в котором будет выполняться вычисление уравнения регрессии
Окно ввода массива экспериментальных данных	<p>1. Формирование таблицы для ввода экспериментальных данных и ее первоначальная обработка</p>	Основное назначение – ввод массива экспериментальных данных

Стоит отметить, что реализация графического интерфейса (в данном случае используется технология реализации обыкновенного интерфейса WindowsForms) также предполагает использование классов. В табл. 1 представлены четыре, наиболее важные с точки зрения текущего рассмотрения, класса – они обозначены как окна. Рассмотрение всех классов на данном этапе нецелесообразно.

Вышеописанная структура (без учета графического интерфейса) может быть реализована на любом объектно-ориентированном языке программирования, конечно, если не учитывать детали программирования на том или ином языке. Что касается графического интерфейса, то он зависит от языка программирования и среды разработки.

Структуру из табл. 1 можно использовать с незначительными изменениями для программной реализации многофакторного регрессионного анализа. В этом случае класс, выполняющий функции калькулятора, следует дополнить изменениями, связанными с формированием матрицы значений базисных функций, и создать окна для вывода соответствующей информации по многофакторным регрессионным моделям. При этом наибольшее внимание следует уделить проверке коэффициентов уравнения регрессии на значимость. Можно отметить, что данная процедура является трудоемкой, поскольку предполагает определение всех новых видов уравнения, в котором были исключены незначимые коэффициенты. В качестве примера рассмотрим полиномиальную функцию второй степени, характеризующую воздействие трех независимых переменных на одну зависимую:

$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + a_{12}x_1x_2 + a_{13}x_1x_3 + a_{23}x_2x_3 + a_{11}x_1^2 + a_{22}x_2^2 + a_{33}x_3^2 \quad (9)$$

При проверке коэффициентов на значимость в случае уравнения вида (9) такое уравнение в случае исключения коэффициентов в какой-либо комбинации может принять

$$\sum_{u=1}^{10} \frac{n!}{k_u!(n-k_u)!} = 1023 \quad (10)$$

различных вида.

Реализация цикла при проверке коэффициентов уравнения на значимость представляется трудновыполнимой, поскольку в случае исключения коэффициентов требуется каждый раз формировать новое уравнение, а в многофакторном случае – выполнять еще и перерасчет коэффициентов. Поэтому при проверке коэффициентов уравнения на значимость следует использовать другой законный путь – рассмотрение всех новых вариантов уравнения через конструкцию ветвления. Такой путь является трудоемким и предполагает написание большого количества программного кода. Поэтому программную реализацию исключения коэффициентов (формирования нового уравнения) для однофакторной модели желательно выполнить отдельно от реализации процедуры исключения коэффициентов для многофакторной модели регрессии.

### Библиографический список

1. Живописцев Ф. А., Иванов В. А. Регрессионный анализ в экспериментальной физике. – М.: Изд-во МГУ, 1995. – 208 с.
2. Пустыльник Е. И. Статистические методы анализа и обработки наблюдений. – М.: Наука, 1968. – 288 с.
3. Сеньченков В. И. Статистические методы обработки экспериментальных данных: учеб. пособие. – СПб. – ГУАП, 2006. 244 с.
4. Подбельский В. В. Язык Си++: учеб. пособие. – М.: Финансы и статистика, 2006. – 560 с.

УДК 621.391.

**Н. В. Апанасенко**

магистрант кафедры проблемно-ориентированных вычислительных комплексов

**Е. А. Бакин**

канд. техн. наук, доц. – научный руководитель

### МОДЕЛЬ СИСТЕМЫ ПЕРЕДАЧИ ДАННЫХ С ПОЗИЦИОННО ИМПУЛЬСНОЙ МОДУЛЯЦИЕЙ ПРИ ВОЗДЕЙСТВИИ ИМПУЛЬСНОЙ ПОМЕХИ

В современных системах связи, локации и навигации все большую популярность приобретают сигналы с шириной спектра более 500 МГц, которые получили название сверхширокополосных (СШП) сигналов [1]. Наиболее распространенным видом модуляции для СШП систем передачи данных является позиционно-импульсная модуляция (Pulse-Position Modulation, PPM). PPM модуляция представляет собой вид модуляции, в которой положение импульсов соответствует передаче информационного нуля или единицы. При этом передача одного бита разбивается на  $N$  фреймов, в каждом из которых передается один кодовый символ. При этом один фрейм разбивается на  $M$  интервалов, называемых слотами. Каждый слот разбивается на два полуслота. В зависимости от того, какой передается кодовый символ (ноль или единица), импульс формируется в правом либо в левом полуслоте. Актуальной задачей является анализ устойчивости подобных систем к различным видам помех. Одним из распространенных видов помех является хаотическая импульсная помеха (ХИП), которая представляет собой последовательность импульсов произвольной длительности и амплитуды, следующих друг за другом через случайные промежутки времени [2]. Данная работа посвящена созданию моделирующей программы для анализа системы СШП при воздействии ХИП.

На рис. 1 представлена типовая схема приемника PPM сигнала. Сигнал  $x(t)$ , являющийся суммой полезного сигнала  $s(t)$  и шума  $n(t)$ , поступает на два перемножителя. Генератор опорной последовательности (ГОП), будучи синхронизированным с передатчиком, формирует две последовательности импульсов:  $g_i^{(1)}(t) = \sum_{i=1}^N g_i^{(1)}(t)$  и  $g_i^{(0)}(t) = \sum_{i=1}^N g_i^{(0)}(t)$ . Сигнал  $g_i^{(1)}(t)$  – это импульс, положение которого соответствует передаче одного кодового символа, равного единице. Второй сигнал –  $g_i^{(0)}(t)$ , представляет собой импульс, положение которого соответствует передаче одного кодового символа, равного нулю.  $y_i^{(0)}$  и  $y_i^{(1)}$  – значение интегралов для первого и второго ожидаемых сигналов, определенные за время  $T_f$ , где  $T_f$  – длительность одного фрейма. Далее последовательности  $y_i^{(0)}$  и  $y_i^{(1)}$  поступают на вход устройства принятия решения (УПР).

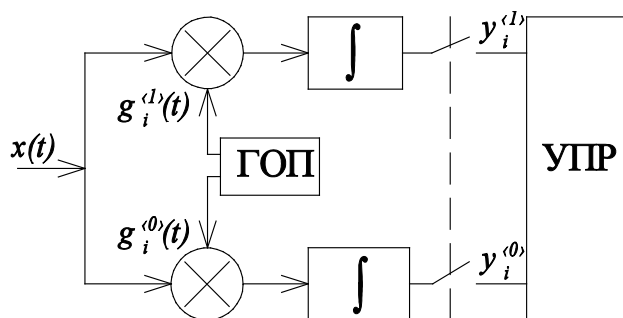


Рис. 1. Общая схема приемника PPM сигнала

В данной работе рассматривается два алгоритма, выполняемые устройством принятия решения: корреляционный прием и прием с двоичным квантованием [3]. Оценка переданного бита  $\hat{b}$  для корреляционного приема может быть записана в виде следующего математического индикатора:

$$\hat{b} = I \left\{ \sum_{i=1}^N (y_i^{(1)} - y_i^{(0)}) \geq 0 \right\},$$

а решающее правило для приемника с двоичным квантованием – следующим выражением:

$$\hat{b} = I \left\{ \sum_{i=1}^N \text{sign}(y_i^{(1)} - y_i^{(0)}) \geq 0 \right\},$$

где  $\text{sign}$  – двоичный квантователь.

Рассматривается помеха, отсчет отклика которой на выходе приемника описывается следующей плотностью распределения вероятности:

$$f_n(x) = (1-p)\delta(x) + p \frac{1}{\sqrt{2\pi D}} \exp\left(-\frac{x^2}{2D}\right).$$

Здесь  $\delta(x)$  – дельта-функция,  $p$  – вероятность возникновения помехи в одном полуслоте,  $D$  – дисперсия помеховой компоненты в случае ее появления. В зависимости от параметра  $p$  помеха может представлять либо отклик входного тракта приемника на ХИП (при  $p \rightarrow 0$ ), либо флуктуационный шум (при  $p \rightarrow 1$ ).

Для такой помехи средняя мощность определяется следующим выражением:

$$P_d = pD.$$

Одним из основных параметров системы является отношение сигнал/помеха на входе УПП в децибелах, которое равно:

$$q = \frac{1}{2MpD}.$$

Зачастую вывод замкнутых выражений для вероятностных характеристик качества приема СШП сигнала является затруднительным. В этих случаях прибегают к методу имитационного моделирования. С данной целью была разработана программа на языке Matlab, целью которой является нахождение вероятностных характеристик качества приема сигнала от параметров системы [4]. Перечень входных и выходных параметров приводятся в таблице.

Таблица

Перечень параметров системы имитационного моделирования

Параметр	Обозначение	Тип переменных
Количество фреймов	$N$	Целочисленный (int)
Количество слотов	$M$	Целочисленный (int)
Количество переданных бит	$nb$	Целочисленный (int)
Вероятность возникновения помехи	$p$	Значение с плавающей точкой (double)
Отношение сигнал/помеха	$q$	Значение с плавающей точкой (double)
Вероятность возникновения битовой ошибки	$P_e$	Значение с плавающей точкой (double)

Программа позволяет моделировать два вида зависимостей: вероятность возникновения битовой ошибки от значения отношения сигнал/помеха, при фиксированном параметре  $p$  согласно блок схеме, представленной на рис. 2,а и вероятность возникновения битовой ошибки от значения вероятности появления помехи в одном полуслоте, при фиксированном параметре  $q$ , согласно блок схеме, представленной на рис. 2,б.

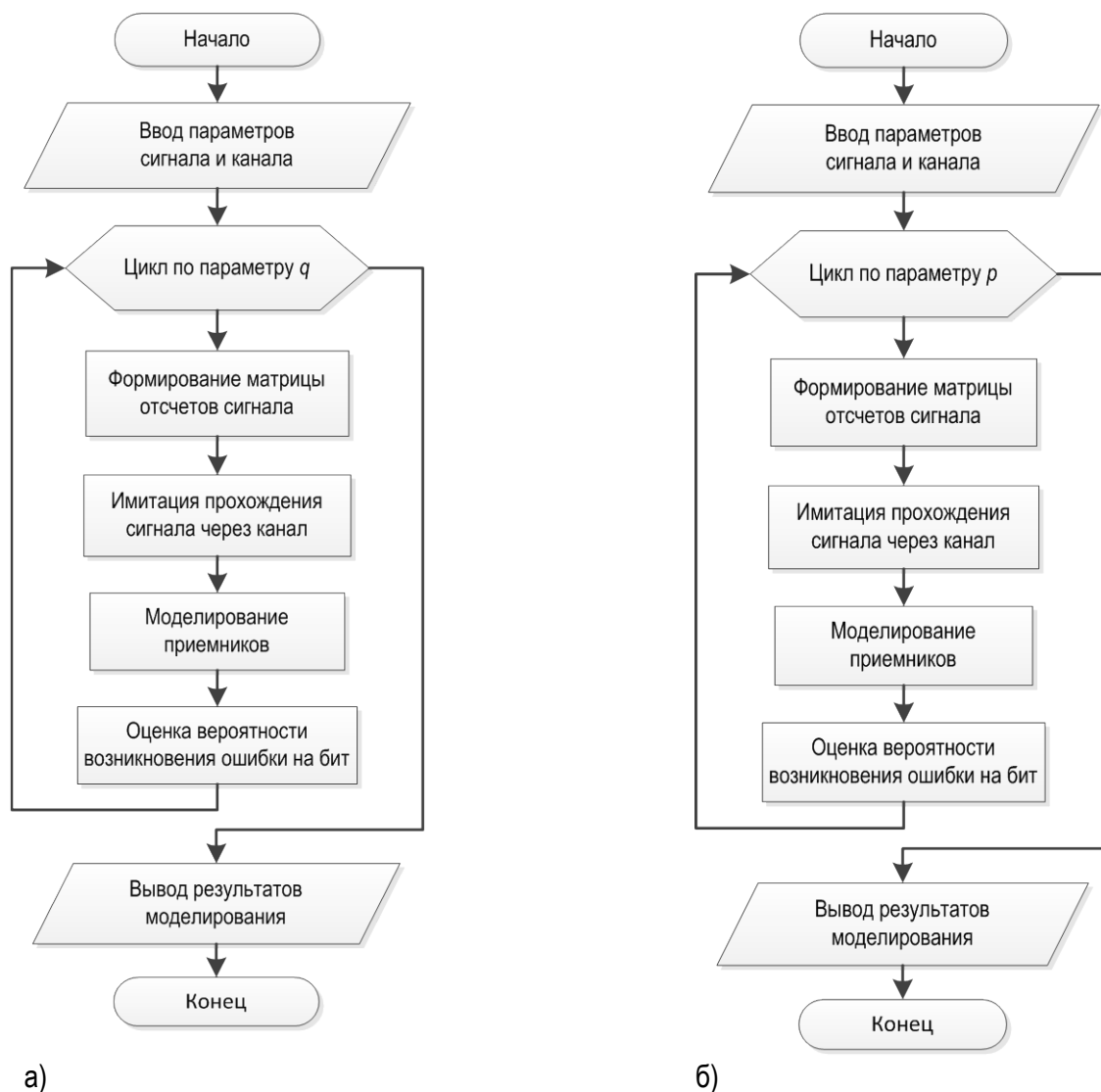


Рис. 2. Блок схема программы имитационного моделирования

Пример работы программы, соответствующей блок схеме, приведенной на рис. 2,а представлен на рис. 3. Моделирование осуществлялось при следующих параметрах:  $nb = 10^6$ ,  $M = 5$ ,  $N = 4$ ,  $p = 0.05$ ,  $q = [-15 \ 10]$  дБ, что соответствует типовым параметрам для подобных систем [5].

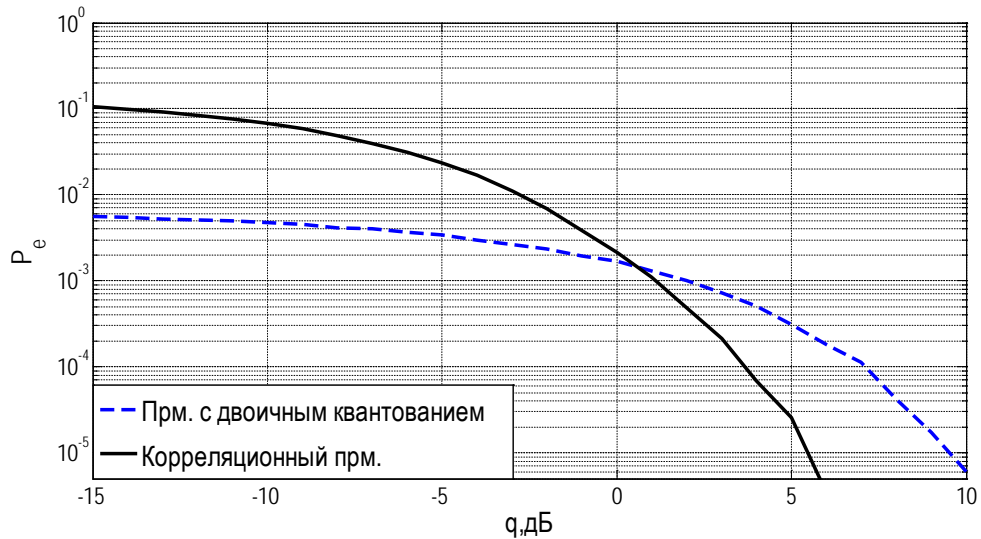


Рис. 3. График зависимости возникновения битовой ошибки при воздействии ХИП

Пример работы программы, соответствующей блок схеме, приведенной на рис. 2,б представлен на рис. 4. Моделирование осуществлялось при следующих параметрах:  $nb = 10^6$ ,  $M = 5$ ,  $N = 4$ ,  $p = [0.001 \ 1]$ ,  $q = -2$  дБ.

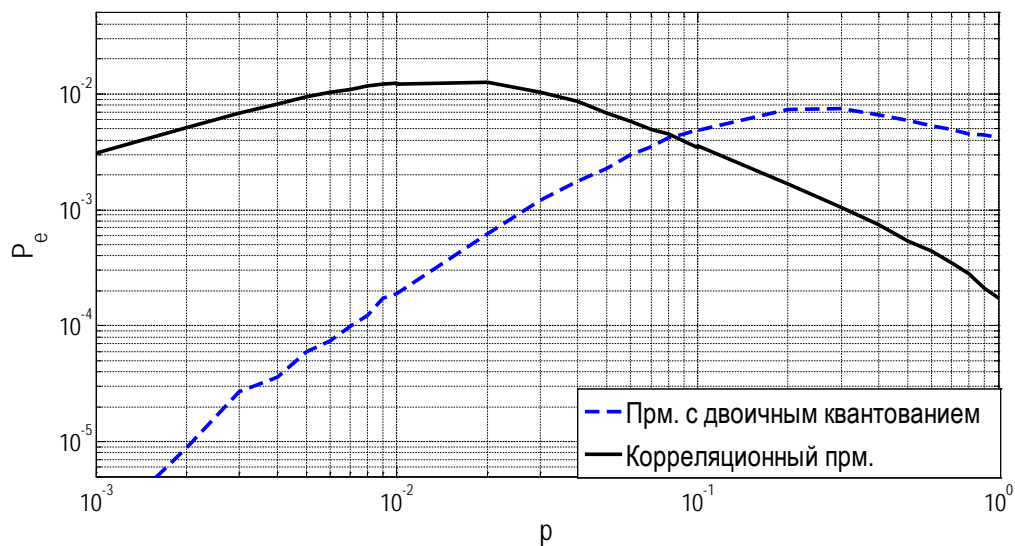


Рис. 4. График зависимости вероятности битовой ошибки от вероятности возникновения помехи в одном полуслоте

В заключение следует отметить, что планируется расширение возможностей имитационной модели, которое позволит проводить анализ систем передачи данных сверхширокополосных сигналов при воздействии других видов аддитивных помех, при наличии многолучевого распространения и перекрестных помех [6].

**Библиографический список**

1. Радзиевский В. Г., Трифионов П. А. Обработка сверхширокополосных сигналов и помех. – М.: Радиотехника, 2009. 290 с.
2. Richard Poisel. Modern Communications Jamming Principles and Techniques. – Boston, MA: House Publishers. 2011. – 870 p.
3. Финк Л. М. Теория передачи дискретных сообщений. – М.: Советское радио, 1970. – 728 с.
4. Дьяконов В. П. Matlab. Полный самоучитель. – М.: ДМК Пресс, 2014. – 768 с.
5. Bo Nu, Beaulieu N. C. Accurate Evaluation of Multiple-Access Performance in TH-PPM and TH-BPSK UWB Systems // IEEE Transactions on Communications. Oct. 2004. Vol. 52. N. 10. P. 1758–1766.
6. Гранкин М. А., Пустовалов Е. В., Тюрликов А. М. Анализ процедуры погашения интерференции в OFDM-системе со случайным множественным доступом // Известия вузов. Приборостроение. 2013. № 8. С. 35–41.

УДК 004.42, 316.47

**К. В. Бабанов**

студент кафедры компьютерной математики и программирования

**С. А. Рогачев**

научный руководитель

**ВЫДЕЛЕНИЕ ОБЪЕКТОВ СОЦИАЛЬНОЙ СЕТИ  
ПО ЗАДАНЫМ ИНФОРМАЦИОННЫМ ПРИЗНАКАМ**

С появлением информационных технологий появилась возможность работать с огромными наборами данных. Использование этих наборов данных с целью выявления закономерностей играет важную роль в исследовательских задачах. Анализируя эти взаимоотношения можно построить модели, описывающие некоторые группы людей со схожими признаками. Данные показатели в первую очередь будут интересны социологам, психологам, а так же людям, связанным с маркетингом. Благодаря такой классификации по группам, можно найти единомышленников, что может помочь людям, ищущим определенное содержимое на страницах социальной сети, за счет того, что у единомышленников выше вероятность найти интересующие материалы, нежели простым поиском.

Под выделением объектов по заданным информационным признакам будем понимать нахождение общего содержимого на страницах этих объектов. В качестве объектов анализа выступают выбранный пользователем объект и объекты, имеющие какие-либо совпадения с выбранным объектом. Интерфейс социальной сети «ВКонтакте» предоставляет возможность получения набора объектов, имеющих заданный информационный признак. Спроектированная программа выполняет множество запросов к социальной сети с целью получения наборов объектов и производит дальнейший анализ этих наборов с целью выявления совпадений. Работа программы представлена на рисунке ниже.

Здесь, на рисунке в центре под номером 1 изображен исследуемый объект. Все объекты, имеющие наибольшее число совпадений информационных признаков, помещены на рабочей области программы и для наглядности выделены белыми кружками. Объекты, имеющие наибольшее количество совпадений признаков, выделяются на фоне остальных и «светятся» ярче. Это можно заметить, сравнивая объекты с номерами 2 и 3 на рисунке. Данные об исследуемом объекте отображаются в левом нижнем углу вместе с индикацией текущего прогресса, в виде количества найденных и уже разобранных информационных признаков. Наведя курсор на любой объект можно получить дополнительную информацию, в частности, что это за объект: пользователь или сообщество, а так же какое количество баллов за совпадения имеет этот объект. Баллы за совпадения того или иного информационного признака задаются пользователем при запуске программы, что позволяет присваивать одним признакам больший приоритет, чем другим.

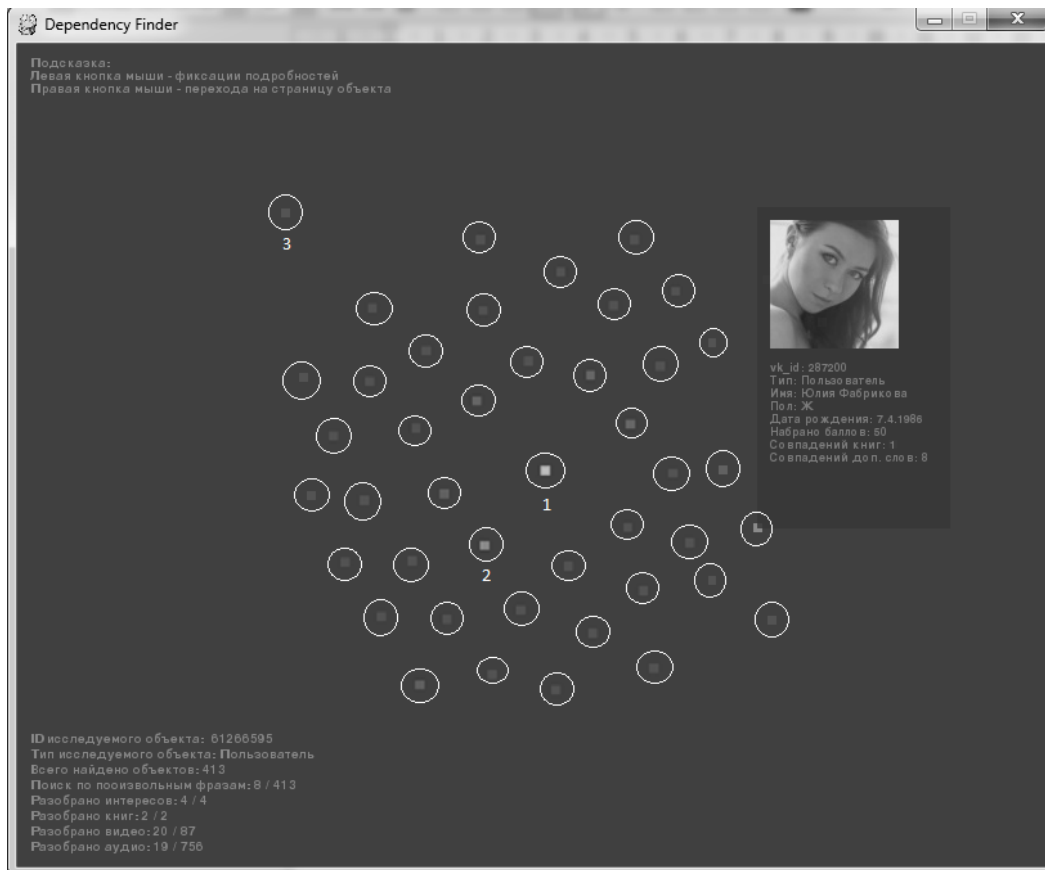


Рис. Работа программы

Также результаты поиска заносятся в таблицу и сохраняются на диск для дальнейшего анализа пользователем. Пример табличных данных представлен ниже.

Таблица

Результаты поиска схожих объектов

ID объекта	Тип	Имя	Всего набрано баллов	Ссылка
-17 463 146	Группа	* yda4aTV *	137	<a href="http://vk.com/club17463146">http://vk.com/club17463146</a>
-54 423 481	Группа	Clever – познавательн	79	<a href="http://vk.com/club54423481">http://vk.com/club54423481</a>
287 200	Пользователь	Юлия Фабрикова	50	<a href="http://vk.com/id287200">http://vk.com/id287200</a>

В результате проделанной работы была разработана программа выделения объектов по заданным информационным признакам. Программа позволяет задавать информационные признаки, на наличие которых проверяются все обрабатываемые объекты. Все полученные данные в результате работы программы сохраняются в файл для возможного дальнейшего анализа этих данных.

**Библиографический список**

1. YASIV // Построение графа дружеских отношений, URL: <http://www.yasiv.com/vk>
2. Facebook data science // The formation of love, URL: <https://www.facebook.com/notes/facebook-data-science/the-formation-of-love/10152064609253859>
3. Документация API ВКонтакте // Описание методов API, URL: [https://vk.com/page-1\\_2369282](https://vk.com/page-1_2369282)



**К. В. Бабанов**

студент кафедры компьютерной математики и программирования

**С. А. Розачев**

научный руководитель

### МОДЕЛИРОВАНИЕ АКСОНАЛЬНОГО НАВЕДЕНИЯ

Целью работы является моделирование процесса аксонального наведения в нейронных сетях нервной системы. Аксональное наведение – сложный процесс роста аксона к своей цели, зависящий от множества факторов. Детальное моделирование данного процесса позволит описывать структуру нейронной сети нервной системы параметрами модели аксонального наведения. Одним из наиболее важных факторов аксонального наведения является уровень концентрации сигнальных молекул – специальных белков, участвующих в аксональном наведении. В данной работе рассматривается упрощенный вариант аксонального наведения, где используются модели только двух типов белков: нетринов, оказывающих притягивающее действие и семафоринов, оказывающих отталкивающее действие. Концентрация этих белков создает так называемый градиент роста, который управляет ростом аксонов и позволяет формировать нужную структуру сети.

В программе будут использоваться градиентные точки – объекты, эмулирующие скопления сигнальных молекул биологических сетей. Значение градиента будет уменьшаться линейно от расстояния, а максимальный уровень градиента может колебаться от  $-1.0$  до  $1.0$ .

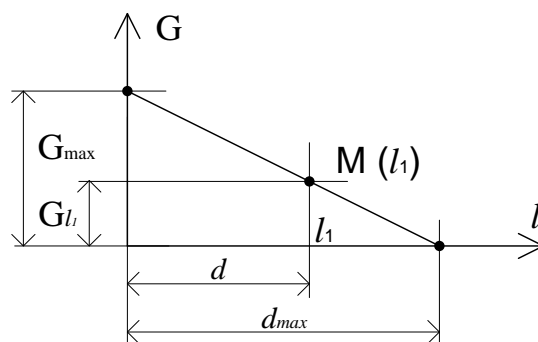


Рис. 1. Вычисление уровня градиента роста

Значение градиента в какой-либо точке будет вычисляться следующим образом: пусть имеется градиентная точка со значением градиента  $G_{\max}$  и необходимо вычислить значение градиента на расстоянии  $d$  (рис. 1). Если расстояние  $d > d_{\max}$ , то значение градиента равно нулю, иначе значение градиента вычисляется по следующей формуле:

$$G(l_1) = \frac{G_{\max}(l_1 - d_{\max})}{-d_{\max}} \quad (1)$$

Функцию взятия градиента в заданной точке на плоскости можно представить в следующем виде:

$$G(x, y) = G(\sqrt{(g_x - x)^2 + (g_y - y)^2}) \quad (2)$$

где  $g_x$  и  $g_y$  – координаты градиентной точки на плоскости.

Если поблизости имеется несколько градиентных точек, то значение градиента в точке  $M(x, y)$  будет равно сумме всех градиентов, действующих на эту точку:

$$G(M(x, y)) = \sum_{i=1}^n G_i(M(x, y)) \quad (3)$$

где  $n$  – количество близлежащих градиентных точек.

Рис. 2 отражает схему нахождения направления роста аксона. В каждой точке вычисляется значение градиента роста и рост аксона продолжается по тому направлению, где градиент выше. Точки  $M_1$  и  $M_2$  являются частью уже сформированного аксона.

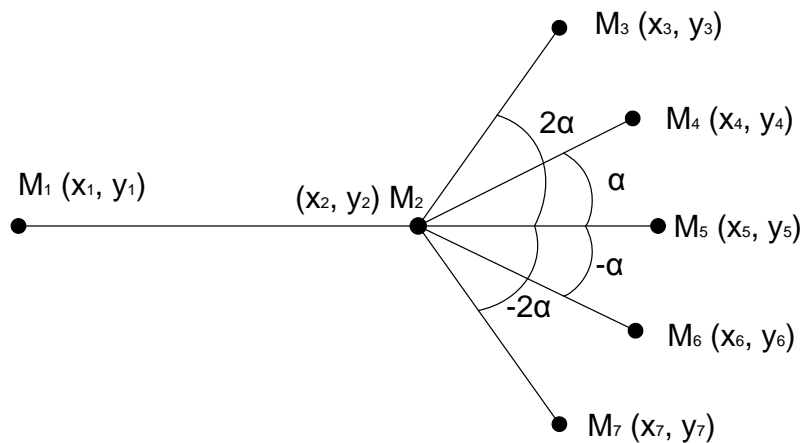


Рис. 2. Нахождение направления роста аксона

Сначала находится направление вектора, образуемого линией между точками  $M_1$  и  $M_2$  следующим образом:

- вычисляется длина  $r$  отрезка  $[M_1, M_2]$  по формуле:

$$r = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}; \quad (4)$$

- вычисляется единичный вектор  $v$ :

$$v = \left( \frac{x_2 - x_1}{r}, \frac{y_2 - y_1}{r} \right); \quad (5)$$

- используя матрицу поворота

$$M(\theta) = \begin{pmatrix} \cos \theta & \mp \sin \theta \\ \pm \sin \theta & \cos \theta \end{pmatrix} \quad (6)$$

домножаем вектор  $v$  на матрицу поворота, указывая различные углы  $\theta$ :

$$\dot{v} = \begin{bmatrix} \dot{v}_x \\ \dot{v}_y \end{bmatrix} = \begin{bmatrix} \cos \theta & \mp \sin \theta \\ \pm \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}. \quad (7)$$

Конкретные знаки в формуле зависят от того, является ли система координат правосторонней или левосторонней, и выполняется ли вращение по или против часовой стрелки. Верхний знак указан для обычного соглашения: правосторонняя система координат и положительное

направление вращения против часовой стрелки (тот же знак верен для левосторонней координатной системы при выборе положительного направления вращения по часовой стрелке; в оставшихся двух комбинациях – нижний знак).

В качестве угла  $\theta$  подставляются значения от  $-60$  до  $60$  градусов, образуя направления роста  $[M_2, M_3]$ ,  $[M_2, M_4]$ ,  $[M_2, M_5]$ ,  $[M_2, M_6]$  и  $[M_2, M_7]$  соответственно. Затем полученный вектор  $v'$  домножается на случайную величину приращения  $d$  следующим образом:

$$M_i = (M_{2_x} + dv_{i_x}, M_{2_y} + dv_{i_y}), \quad i = \overline{3,7} \quad M_i, i = \overline{3,7}. \quad (8)$$

В полученных точках вычисляется градиент роста, и рост аксона продолжается в том направлении, где градиент оказывается ниже. Если в нескольких точках значение градиента одинаково, то следующее направление роста выбирается случайно.

Работа программы, моделирующая аксональное наведение представлена на рис. 3.

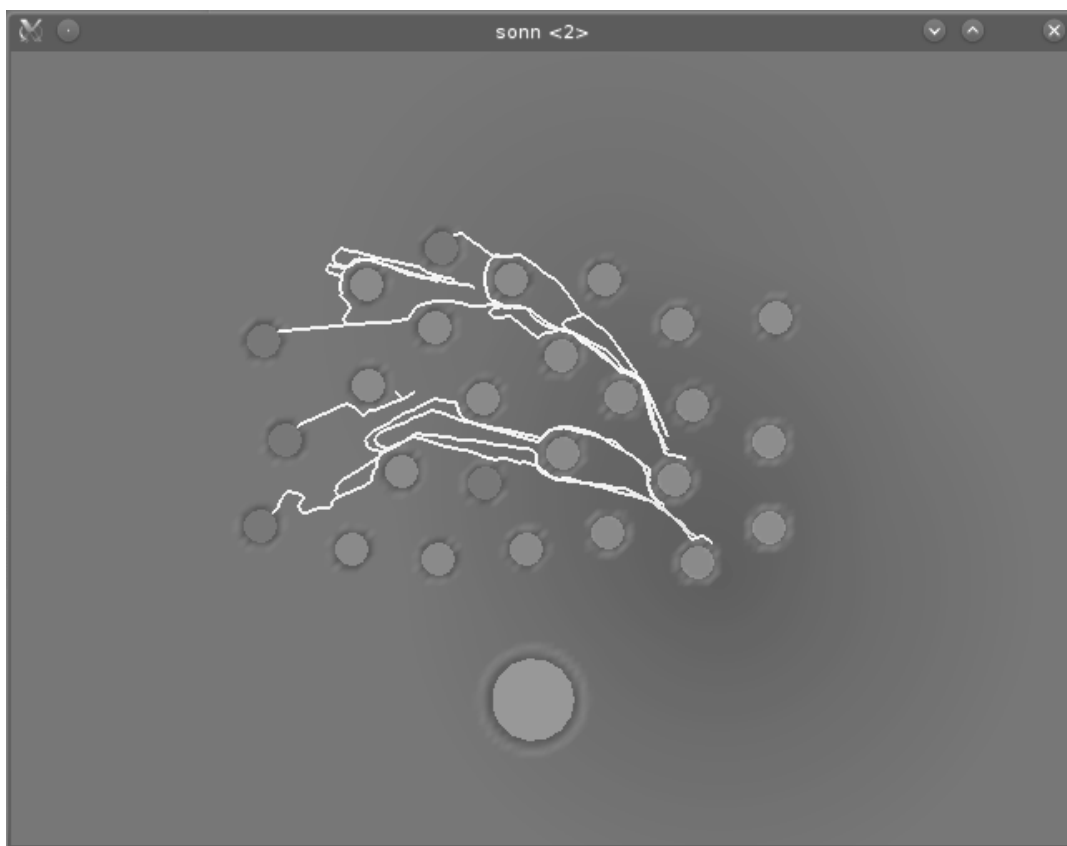


Рис. 3. Работа программы

Здесь, на рис. 3 белыми линиями изображены аксоны, которые растут в сторону градиентной точки, находящейся в правом нижнем углу рабочего окна программы, по градиенту концентрации. Круглые объекты – нейроны, имеющие отрицательный градиент концентрации.

### Библиографический список

1. Баарс Б., Гейдж Н. Мозг, познание, разум. Введение в когнитивные нейронауки. М.: БИНОМ. Лаборатория знаний. 2014. 541 с.
2. Айзек Азимов. Мозг человека. От аксона до нейрона. М.: ЗАО Изд-во Центрполиграф, 2003. 461 с.
3. Dominique Bagnard. Axon growth and guidance. Springer-Verlag New York, 2007. 170 с.

УДК 004.627

**А. А. Балян**

магистрант кафедры компьютерной математики и программирования

**А. А. Ключарев**

канд. техн. наук, доц. – научный руководитель

### ПРЕОБРАЗОВАНИЕ СЖАТОГО ЦВЕТНОГО ИЗОБРАЖЕНИЯ В СЖАТОЕ ИЗОБРАЖЕНИЕ С ГРАДАЦИЕЙ СЕРОГО В МИКРОКОНТРОЛЛЕРНЫХ СИСТЕМАХ

Видеосъемка и видеотрансляция – широко распространенные задачи, и, как правило, к передаваемой картинке предъявляются такие требования как высокое разрешение и частота кадров, хорошая цветопередача. Но есть случаи, в которых размер и качество картинки играют не самую важную роль. Например, когда видеонаблюдение ведется в темное время суток и основная задача сводится к выделению характерных объектов на местности. В этом случае достаточно иметь небольшое черно-белое изображение (с 256 градациями серого).

Если есть необходимость вести видеотрансляцию с удаленного портативного устройства по радиоканалу, то актуальной проблемой становится энергопотребление, которое в первую очередь связано с потреблением приемопередатчика, видеосенсора и контроллера, установленных на устройстве. От выбора этих компонентов зависит продолжительность автономной работы видео-передатчика.

Трансиверы с низким энергопотреблением, обеспечивающие дальность порядка трехсот метров, имеют пропускную способность радиоканала не более 1250 кБит при мощности 14 дБм [1]. По этой причине необходимо сокращать количество передаваемых данных с целью увеличения частоты кадров или разрешения. Для уменьшения потока видео используется сжатие кадров. А значит, в данной ситуации необходим маломощный видеосенсор с небольшим разрешением и возможностью компрессии изображений. Но у миниатюрных КМОП сенсоров, представленных на рынке и отвечающих описанным выше требованиям, есть ограничения.

Видеосенсор, оснащенный контроллером сжатия картинки, может подавать на выход как цветные так и черно-белые сжатые по стандарту JPEG изображения, но они практически не отличаются по объему занимаемой памяти. Это связано с тем, что и цветная и черно-белая картинки содержат в себе цветоразностные компоненты, то есть мы лишились возможности видеть цвета объектов благодаря применению соответствующего фильтра, но файл изображения не стал от этого меньше.

Эта проблема и подводит нас к задаче, решение которой представлено в данной статье: задаче преобразования цветного изображения, сжатого по стандарту JPEG, в изображение с градацией серого, также сжатого по стандарту JPEG.

Для хранения изображений, сжатых по алгоритму JPEG, предусмотрен формат, описанный в стандарте JFIF (JPEG File Interchange Format). Стандарт JPEG предоставляет четыре варианта процесса кодирования:

- 1) последовательный;
- 2) прогрессивный;
- 3) иерархический;
- 4) сжатие без потерь [2].

Предложенный алгоритм предназначен для работы с последовательными файлами JFIF, поскольку они наиболее распространены.

Упрощенная схема алгоритма последовательного процесса кодирования представлена на рис. 1.

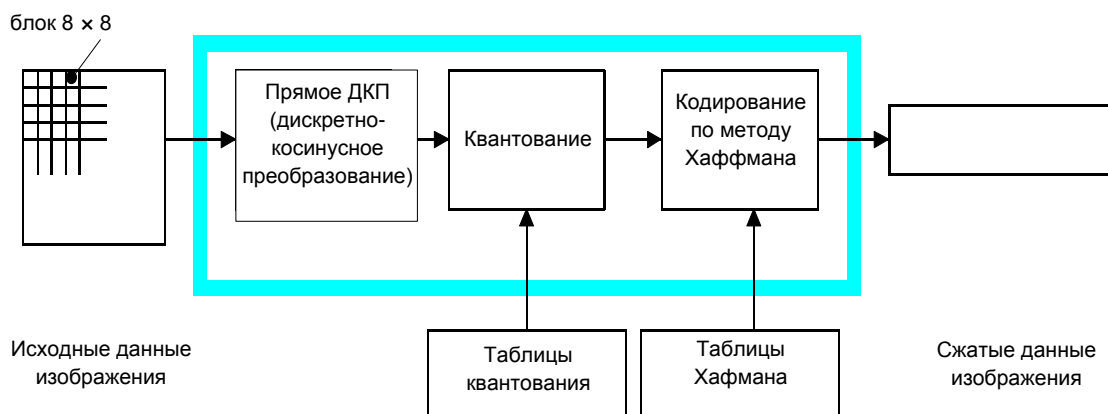


Рис. 1. Алгоритм последовательного процесса кодирования

Цветовое пространство исходных данных в стандарте не указывается, но наиболее широко используется пространство YCbCr. Если исходное изображение представлено в наиболее привычном цветовом пространстве RGB (Red Green Blue), то перед реализацией компрессии оно переводится в пространство YCbCr. YCbCr или  $Y C_B C_R$  – семейство цветовых пространств, которые используются для передачи цветных изображений в компонентном видео и цифровой фотографии. Y – компонента яркости,  $C_B$  и  $C_R$  являются синей и красной цветоразностными компонентами [3].

Из рис. 2 видно, что яркостная компонента Y исходного изображения представляет собой изображение с градацией серого. Таким образом, можно сделать вывод, что для преобразования сжатого цветного изображения в изображение с градацией серого, необходимо вычлнить из общего объема сжатых данных компоненту Y. Но для этого необходимо иметь представление о том, как хранятся сжатые данные и где среди них находится интересующая нас яркостная компонента.



Рис. 2. Цветовые пространства RGB и YCbCr

Одним из подготовительных этапов в алгоритме сжатия JPEG является дискретизация. Сначала изображение разбивается на блоки 8x8, называемые единицами данных. Каналы  $C_b$  и  $C_r$  прореживаются, то есть блоку пикселей присваивается усредненное значение. Например, после прореживания в два раза по горизонтали компоненты Y, единицы данных будет иметь место соответствие, показанное на рис. 3.

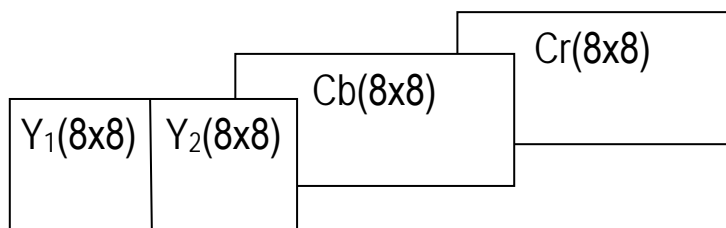


Рис. 3. Минимальный кодируемый блок

Таким образом, четырем единицам данных (блокам 8x8) компонента Y соответствует по одной единице данных компонентов Cb и Cr.

Представленный на рис. 3 участок изображения называется минимальным кодированным блоком MCU (Minimum Coded Unite), он имеет размер 2x8 X 1x8.

В данном примере получаем частоту дискретизации по горизонтали для Y равной 2, по вертикали – 1, а для Cb и Cr частоты дискретизации равны 1 по обоим направлениям.

Блоки MCU поочередно (слева направо, сверху вниз) кодируются и записываются в файл. То есть в сжатом виде в файле данные хранятся в следующем виде[3]:

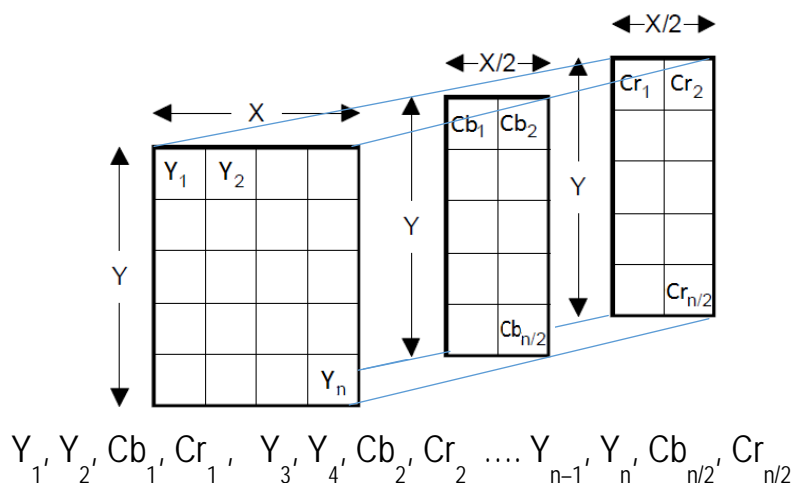


Рис. 4. Структура сжатых данных

Частоты дискретизации используются разные, а также изображения бывают разных размеров, используются различные таблицы квантования и таблицы Хаффмана, поэтому перед началом прохождения по сжатым данным, необходимо проанализировать вспомогательную информацию, которая содержится в файле JFIF.

JFIF файл состоит из блоков, каждый из которых начинается с маркера – двухбайтового слова-разделителя. Также существуют маркеры, после которых не следует никакой информации, например маркер SOI (Start Of Image) – признак начала изображения [4].

Структура последовательного JFIF файла представлена в табл. 1.

Маркер SOI	Start Of Image (FFD8)– признак начала изображения
Маркер APP <sub>0</sub>	Идентификация JFIF-файла и вспомогательные данные
Таблицы квантования	Таблицы квантования для разных компонентов, каждой таблице предшествует маркер DQT (FFDB) и номер таблицы
Таблицы Хаффмана	Таблицы Хаффмана для разных компонентов изображения, каждой таблице предшествует маркер DHT (FFC4), номер таблицы и ее тип
Маркер SOF	Маркер начала кадра (StartOfFrameFFC0), содержит информацию о количестве компонентов, частотах дискретизации и номера таблиц квантования каждого компонента (номера соответствуют номерам из маркеров DQT)
Маркер SOS	Маркер начала скана (StartOfScanFFDA), содержит номера таблиц Хаффмана для каждого компонента (номера соответствуют номерам из маркеров DHT)
Сжатые данные	Последовательность битов, полученная в результате сжатия изображения по алгоритму JPEG
Маркер EOI	End Of Image (FFD9) – признак конца файла

Преобразование сжатого цветного изображения в сжатое изображение с градацией серого без полноценного декодирования выполняется по алгоритму, состоящему из трех основных этапов:

- идентификация;
- изменение содержимого маркеров и запись их в результирующий файл;
- выборка из сжатых данных информации о яркостной компоненте и запись ее в результирующий файл.

Рассмотрим подробнее каждый из этапов.

*Идентификация.* Необходимо убедиться в том, что поступившие данные являются JFIF-файлом. Для этого достаточно проверить первые одиннадцать байт, они должны составлять следующую строку:

FFD8 FF E0 00 10 4A 46 49 46 00,

где [FFD8] – маркера начала изображения (SOI – StartOfImage); [FFE0] – специфические данные программы APP<sub>0</sub>; [00 10] – длина поля маркера APP<sub>0</sub>, она может быть отлична от этого значения, поэтому эти 2 байта при проверке не учитываются; [4A 46 49 46 00] – коды символов J F I F из таблицы ASCII и нулевой байт.

*Изменение содержимого маркеров и запись их в результирующий файл.* На данном этапе отбрасываются таблицы квантования и Хаффмана, предназначенные для цветоразностных компонентов, так как для изображения с градацией серого эта информация будет лишней. А также изменяется содержимое маркеров SOF и SOS, таким образом, чтобы декодеры в дальнейшем распознавали изображение как однокомпонентное. Измененные данные записываются в результирующий файл.

*Выборка из сжатых данных информации о яркостной компоненте и запись ее в результирующий файл.* Для получения из сжатых данных компоненты Y, необходимо определить границы между минимальными кодируемыми блоками и единицами данных компонентов, что является непростой задачей. Сложность состоит в том, что данные записаны в виде битового потока, который формируется в результате применения энтропийного алгоритма сжатия Хаффмана. Для того чтобы определить границы единиц данных, необходимо реализовать декодирование Хаффмана.

Декодирование выполняется для всех трех компонентов с параллельной записью битов компоненты Y в результирующий файл.

Данный алгоритм был реализован и протестирован на платформе для разработок STM3220G-EVAL, оснащенной 32-х разрядным микроконтроллером STM32F207, дополнительной оперативной памятью объемом 1Мб и 2-х мегапиксельным КМОП видеосенсором со встроенным контроллером для сжатия изображения. В тестовом режиме данный аппаратно-программный модуль передает как цветные изображения, так и преобразованные с помощью описанного алгоритма изображения с градацией серого на ПК по интерфейсу USB. Сравнение цветных и черно-белых изображений разных разрешений приведено в табл. 2. Были взяты усредненные значения размеров изображений, полученных в комнатной съемке при освещении лапами накаливания.

Таблица 2

Разрешение изображения	Цветное изображение		Изображение с градацией серого, Кб		Сжатие, %	
	Все изображение, Кб	Заголовок, б	Все изображение, Кб	Заголовок, б	Всего изображения	Данных без заголовка
160x120	1,95	623	1,4	328	28,2 %	22,4 %
320x240	4,8	623	3,8	328	20,8 %	17,5 %

Таким образом, предложенный алгоритм обеспечивает преобразование цветного изображения, сжатого по стандарту JPEG, в изображение с градацией серого, также сжатого по стандарту JPEG, не выполняя при этом полноценного декодирования изображения. В процессе преобразования изображения для выявления яркостной компоненты выполняется только один этап алгоритма JPEG-декодирования Хаффмана. На этом этапе в основном используются логические операции и операции сравнения, являющиеся одними из самых быстрых в процессоре, что позволяет использовать данный алгоритм в микроконтроллерах широкого применения и обрабатывать изображения в процессе видео трансляции.

А также, благодаря избавлению от данных о цветоразностных компонентах, данный метод позволяет избавиться в среднем от 20% передаваемых данных и тем самым увеличить пропускную способность канала связи.

### Библиографический список

1. <http://www.mt-system.ru/news/texas-instruments/cc1200-novyj-sub-1-ghz-priemperedatchik-ot-kompanii-texas-instruments>
2. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных – ISBN 5-86404-170-X, 202
3. Стандарт Т. 81 INFORMATION TECHNOLOGY – DIGITAL COMPRESSION AND CODING OF CONTINUOUS-TONE STILL IMAGES – REQUIREMENTS AND GUIDELINES / CCITT (THE INTERNATIONAL TELEGRAPH AND TELEPHONE CONSULTATIVE COMMITTEE). 1992.
4. Миано Дж. Форматы и алгоритмы сжатия изображений в действии – М.: Издательство «Триумф», 2003 – 336 с.

УДК 681.3.06

**А. Д. Бахмут**

студент кафедры компьютерной математики и программирования

**М. Ю. Охтилеев**

д-р техн. наук, проф. – научный руководитель

### ПРОГРАММНЫЙ КОМПЛЕКС ВТОРИЧНОЙ ОБРАБОТКИ И АНАЛИЗА ИНФОРМАЦИИ ПРИ ОЦЕНКЕ СОСТОЯНИЙ СЛОЖНЫХ ТЕХНИЧЕСКИХ ОБЪЕКТОВ В СОСТАВЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

Всемирная информатизация создает потребность в новых способах структурирования и обработки больших объемов информации. Ограниченность человеческих ресурсов



и постоянное желание сократить расходы привели к созданию систем, которые могут учитывать различные аспекты, способные повлиять на выбор того или иного варианта в процессе принятия решений, а также рассчитать наиболее привлекательные из них.

Человек постоянно принимает решения, но для них требуется информация, чем ее больше, чем она многогранней, тем точнее будет принято решение. При этом увеличение информации приводит к значительному возрастанию времени принятия решения и росту человеческих ресурсов.

Системы поддержки принятия решений (СППР) обеспечивают получение количественной оценки показателей работы на основании выбранных критериев, помогают оценивать последствия возможных решений, выбирать технические средства (или формулировать требования к ним) для эффективного решения поставленной задачи. Эти системы принципиально не исключают человека из процесса принятия решения, а подготавливают необходимую ему информацию о вариантах оптимального решения поставленной задачи [1].

Условно, работу системы поддержки принятия решений можно разделить на два блока (рис. 1).

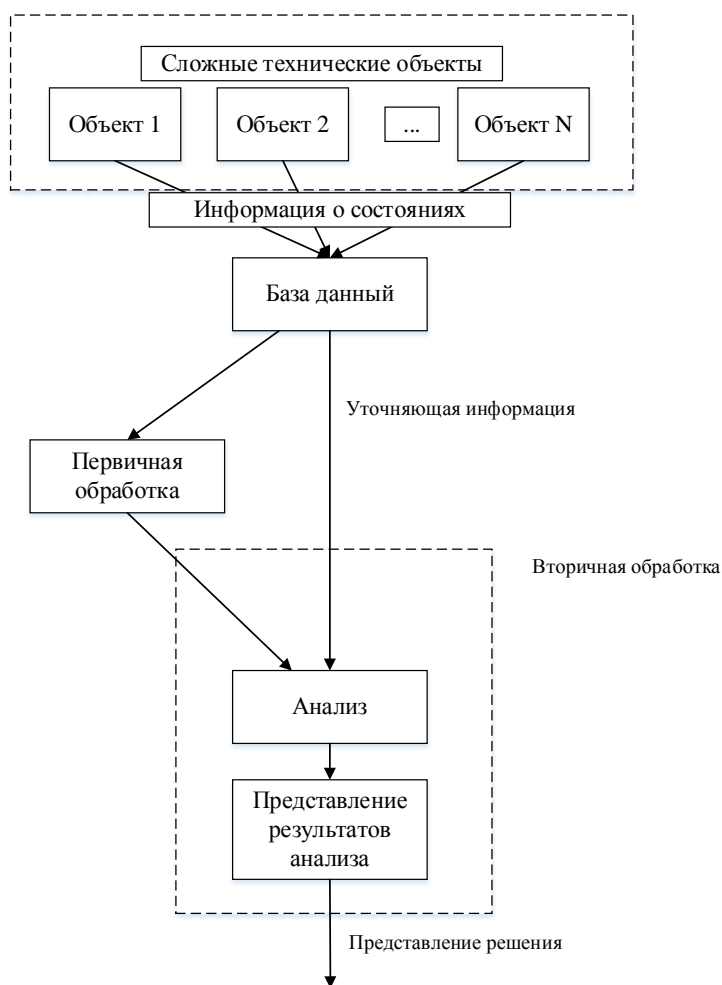


Рис. 1. Схематическое представление СППР

Первичная обработка информации о состоянии объектов, в ходе которой была отсеяна не актуальная или не требуемая в данный момент информация, а оставшаяся дополнена необходимыми вычислениями и представлена в виде максимально удобном для последующих задач анализа.

Вторичная обработка, в рамках которой можно выделить отдельно задачу интеллектуального анализа полученных в ходе первичной обработки данных и задачу представления результатов анализа пользователю или, другими словами, проектирование интерфейса системы. Результатом вторичной обработки является преобразование сотен строк буквенно-цифровых обозначений, не пригодных для использования конечным пользователем, в несколько десятков числовых и словесных параметров, подкрепленных цветовой индикацией и завершенных общим словесным выводом.

К сожалению, создание единых и универсальных средств анализа данных крайне затруднительно в силу обширности и разносторонности сфер применения систем поддержки и принятия решений. Но можно выделить несколько принципиально различающихся подходов к созданию систем анализа данных.

1. Правила вывода – удобны в тех случаях, когда схемы формирования выводов представлены в виде «если–то–иначе». При большом количестве правил теряется наглядность и возрастает сложность восприятия и не во всех сферах можно выделить четкие отношения «если–то–иначе».

2. Нейронные сети – удобны при работе с нелинейными зависимостями, зашумленными и неполными данными, самообучаются в процессе работы на основе ранее полученных результатов. Нейронные сети крайне похожи на «черный ящик»: если входные параметры и есть полученный результат, но нет возможности проследить его получение. Также для успешной работы нейронные сети требуют предварительного обучения.

3. Нечеткая логика – данные ранжируются по степени близости к желаемым результатам. Удобно в случае нечеткого представления данных.

4. Визуализация – графическое представление данных в виде графиков и диаграмм, по которым пользователь сам выявляет закономерности [2, 3].

При вторичной обработке информации важна сохранность промежуточных результатов вычислений, т. к. крайне часто они используют в расчетах не одного, а нескольких параметров. Данный подход приводит к увеличению задействованной памяти, но значительно повышает скорость вычислений и обработки состояний сложных технических объектов. На текущий момент в большинстве автоматизированных рабочих мест нет недостатка в объеме ОЗУ. Таким образом, концентрация внимания на повышении скорости вычислений и, как следствие, получения оценки общей обстановки оказывается вполне оправданной.

Не менее важным является и организация графического пользовательского интерфейса СППР. Именно от него зависит, насколько быстро пользователь сможет воспользоваться предоставленной информацией и принять соответствующее решение.

При проектировании будущего интерфейса системы важно четко определить следующие пункты:

- какую информацию следует предоставить пользователю;
- в каком виде следует предоставлять информацию;
- какие действия предоставить пользователю;
- какие действия считать ошибочными.

Важно не перегружать основное рабочее окно выводимыми параметрами, для того, чтобы внимание пользователя не было расфокусировано (рис. 2). При этом удобно снабжать сугубо числовые параметры словесными пояснениями и сопровождать общим описанием обстановки [4].

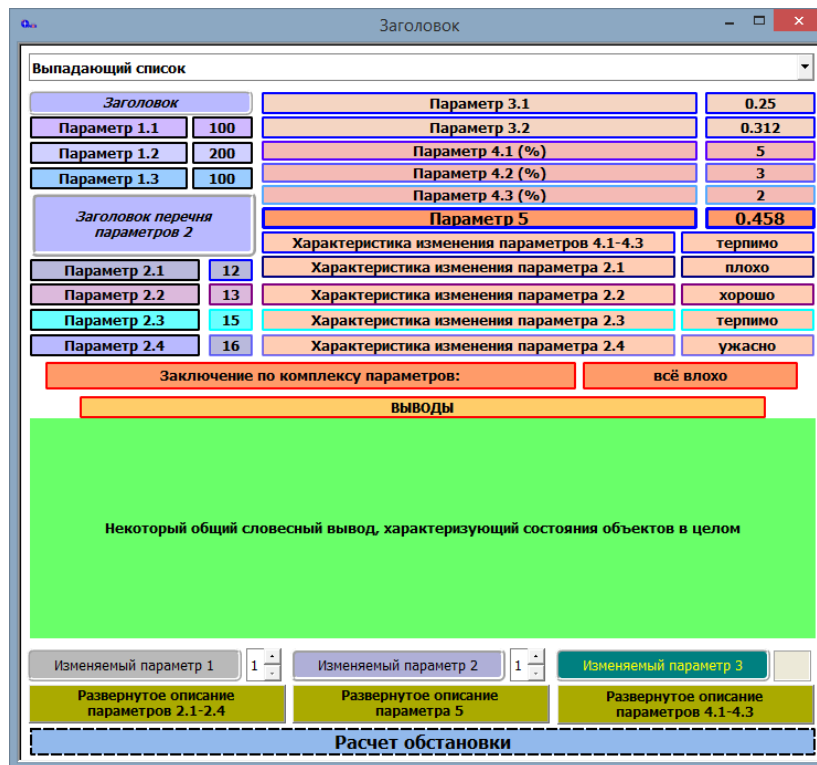


Рис. 2. Схематическое представление основного окна

Необходимо формирование общего вывода, характеризующего ситуацию в целом. В случае необходимости общий вывод дополняется предложением дальнейших действий с возможностью их корректировки и выполнения, а также прогнозированием изменения обстановки.

Хотя основное рабочее окно не должно быть перегружено выводимой информацией, пользователь должен иметь доступ к более развернутому представлению отдельных параметров. Это удобно реализовать в виде дополнительных окон, вызываемых по нажатию определенных кнопок в основном окне (рис. 3).

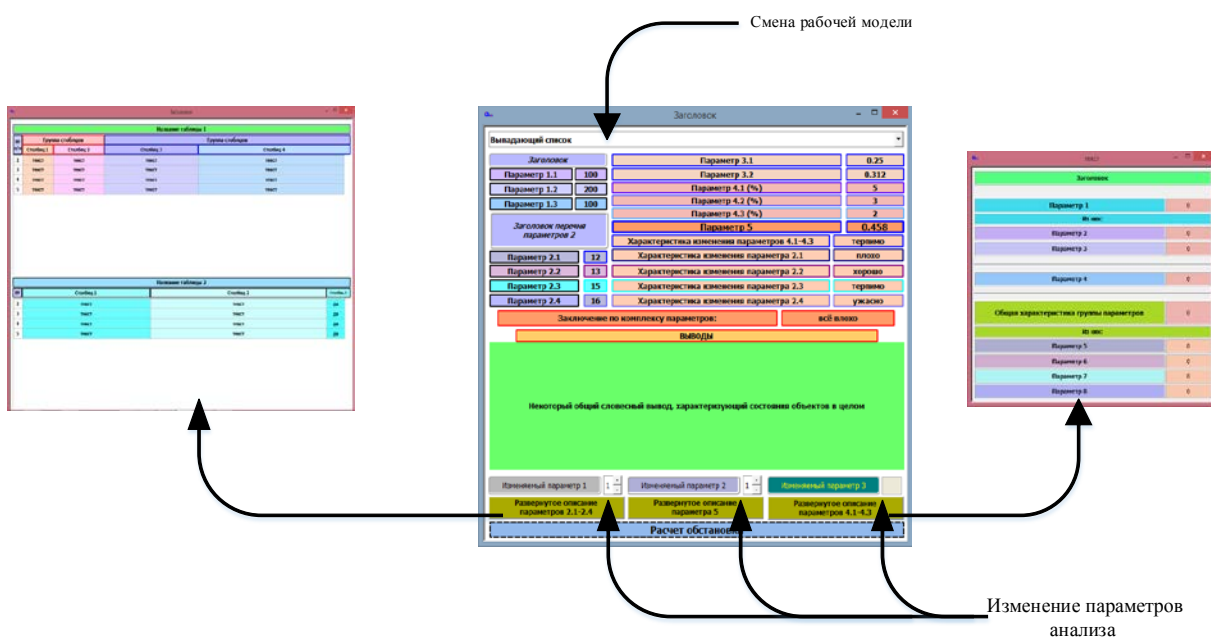


Рис. 3. Схематическая взаимосвязь окон

Так же в основном окне можно расположить средства управления прогнозированием в виде отдельных счетчиков, переключателей, флажков и т. п.

Таким образом, можно заключить, что при создании систем поддержки принятия решения важно не только представление математических моделей первичной обработки информации о состоянии сложных технических объектов, но и способы представления конечных данных и стиль их подачи пользователю в составе вторичной обработки информации. Таким образом, оформление интерфейса системы существенно влияет на скорость осознания ситуации и принятия окончательного решения.

### Библиографический список

1. Быков В. П. Система поддержки принятия решений по управлению движением поездов на участках железных дорог: Конспект лекций. – Хабаровск: Изд-во ДВГУПС, 2001. – 92 с.: ил.
2. Аджиев В. MineSet - визуальный инструмент анализа. Открытые системы, № 3, 1997. С. 73–77.
3. Киселев М., Соломатин Е. Средства добычи знаний в бизнесе и финансах. Открытые системы, № 4, 1997. С. 41–44.
4. Лебедев В. Г. Принципы построения интеллектуального интерфейса пользователя для систем поддержки принятия решений оператором – Проблемы управления, № 3, 2004. С. 43–48.

УДК 004.421.2

**А. О. Беляевский**

студент кафедры компьютерной математики и программирования

**А. А. Ключарев**

канд. техн. наук, доц. – научный руководитель

## АЛГОРИТМ ВЫЧИСЛЕНИЯ ТРАЕКТОРИИ ДВИЖЕНИЯ ВОЛНОМЕРНОГО БУЯ

Измерение морского волнения является важной задачей для большинства отраслей техники, связанных с морем. Приборы для измерения морского волнения используются в гидростроительстве, океанологии, морской геологии, судостроении, мореплавании, военно-морском флоте. Особое место среди указанного класса приборов занимают волномерные буй, получившие широкое распространение в мировой практике волновых измерений. Одним из типов таких буюв является буй в виде плоского поплавка, свободно плавающий на поверхности волны. Принцип работы этих приборов основан на том, что сам буй отслеживает профиль волны, а датчики различных механических величин, установленные на его борту, измеряют параметры его движения. Стандартным набором датчиков является набор из трех датчиков линейных ускорений по трем осям (оси  $X$ ,  $Y$  и  $Z$ ), трех датчиков угловых ускорений (повороты вокруг осей  $X$ ,  $Y$  и  $Z$ ), и трех-осевой компас. На основании данных этих датчиков можно определить траекторию буюа и, соответственно, параметры волнения. Однако методы восстановления траектории движения буюа по отсчетам датчиков разработаны недостаточно (коррекция ошибок, ориентация по магнитному компасу и т.д.). В данной работе была рассмотрена методика восстановления траектории вертикального движения волномерного буюа по его вертикальному ускорению, и проблемы, возникающие в процессе решения этой задачи.

Для того чтобы определить параметры волнения моря, достаточно по данным ускорений по трем осям, трем поворотам и магнитному полю восстановить ускорения по вертикали. Траектория движения датчика, восстановленная из его ускорения по вертикальной оси, позволит вычислить такие параметры как частота и период волнения, высота волны 3%-й обеспеченности.

В данной работе рассматривается алгоритм восстановления траектории движения датчика из информации о его ускорении по вертикальной оси. Для определения параметров волнения

необходимо проводить измерения в течение примерно 20–40 мин.<sup>1</sup> Часть этого времени необходима для завершения переходных процессов в алгоритмах обработки данных датчиков.

Для восстановления траектории движения датчика необходимо провести двойное интегрирование отсчетов вертикального ускорения, однако при этом мы вынуждены сделать два предположения: начальная координата по вертикальной оси и начальная скорость по вертикальной оси равны нулю. Из-за несоответствия этих значений истинным, в результате интегрирования мы получим неконтролируемые ошибки по положению и скорости.

Для решения этой проблемы было решено применить фильтр низких частот второго порядка, который позволит убрать ошибки в определении и начального смещения, и начальной скорости.

В результате получается точная картина относительных перемещений датчика по вертикальной оси (рисунок), что позволяет провести анализ параметров волнения, однако необходимо разработать фильтр с достаточно коротким переходным процессом, чтобы успеть провести все вычисления в пределах минимального интервала времени (с учетом необходимости накопления и усреднения данных). Как видно из рисунка, следует начинать анализ после того как закончится переходный период импульсной характеристики фильтра.

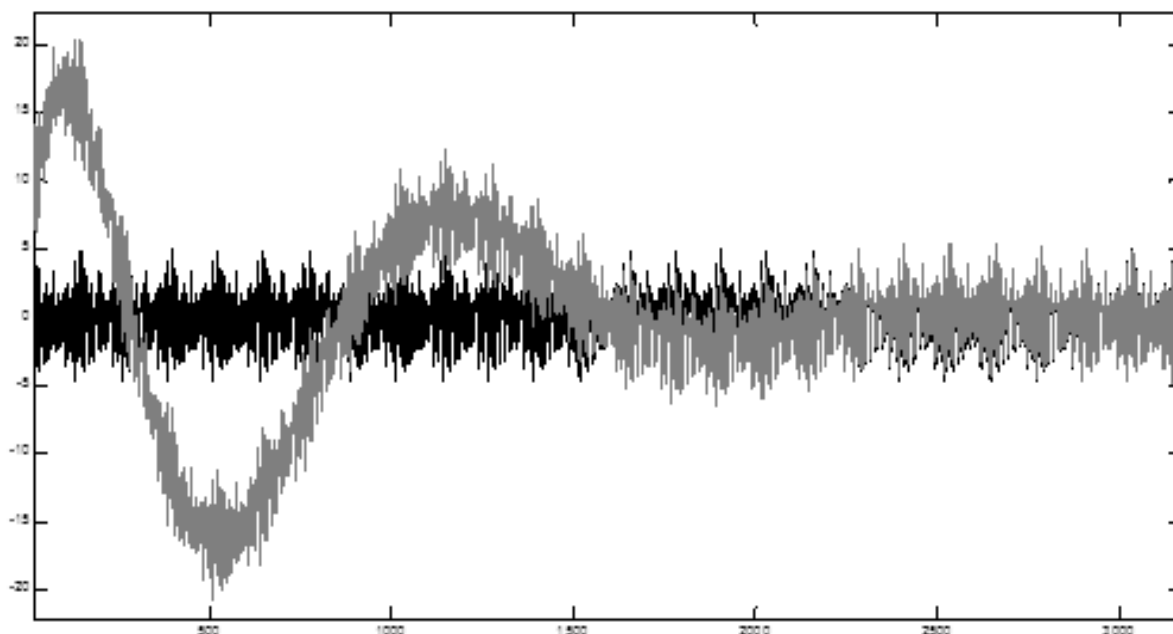


Рис. Результат интегрирования с применением фильтра нижних частот второго порядка. Черное – смоделированная исходная картина волнения, серое – восстановленная исходная картина волнения

Поскольку данные от датчиков прибора получены в локальной системе координат, привязанной к прибору, для определения параметров по вертикальной оси из исходных данных (3 ускорения, 3 поворота, магнитное поле) необходимо преобразовать их к абсолютной системе координат.

Для определения углов отклонения локальной системы координат от глобальной необходимо измерять локальные направления как минимум двух непараллельных векторов, направления которых всегда точно известны в глобальной системе координат. Одним из таких векторов

<sup>1</sup> Ветровое волнение как вероятностный гидродинамический процесс / И. Н. Давидан, Л. И. Лопатухин, В. А. Рожков; Ленинград: Гидрометеиздат, 1978. 269 с.

может быть вектор магнитного поля, положение которого фиксируется при калибровке прибора, производимой перед экспериментом. В качестве второго вектора можно использовать направление гравитационного поля Земли, а, точнее, то, что средние значения  $g_x$  и  $g_y$  (проекция ускорения свободного падения на оси  $X$  и  $Y$  соответственно) равны 0. Отсчеты магнитного поля мгновенны, и его направление можно использовать непосредственно при вычислениях. Использование вектора гравитационного поля затруднено тем, что его локальные проекции на оси  $X$  и  $Y$  могут колебаться при наклонах поплавка и приходится ориентироваться на их средние значения, что сильно затрудняет расчеты. Поэтому его использование требует отдельного рассмотрения, что выходит за рамки данной работы. Отдельному рассмотрению также подлежит задача коррекции аппаратных погрешностей, смещений нуля, погрешностей калибровки, шумов, температурных и временных их зависимостей, погрешностей связанных с дискретными отсчетами данных (ускорений, поворотов) и т. п.

Применение предложенной выше методики восстановления траектории движения волнового буя позволяет выполнить анализ его траектории и вычислить параметры ветрового волнения. Следует также отметить, что данная методика применима к анализу пространственной (3-х мерной) структуры ветрового волнения моря.

УДК 004.05

**В. В. Богданов**

магистрант кафедры вычислительных систем и программирования

**В. В. Бураков**

д-р техн. наук, доц. – научный руководитель

### СОВМЕСТНОЕ ПРИМЕНЕНИЕ СТАТИЧЕСКОГО И ЭВРИСТИЧЕСКОГО АНАЛИЗАТОРОВ БЕЗОПАСНОСТИ КОДА

В силу повсеместного применения вычислительной техники во всех сферах деятельности человека проблема надежности программного обеспечения (ПО) является актуальной и приоритетной. Разработка новых средств анализа надежности ПО производится с целью повышения скорости разработки и качества выпускаемых продуктов. Качество и надежность могут оцениваться как с точки зрения безопасности программного обеспечения, так и с точки зрения других критериев, например, время выполнения программы, ограничения на потребляемые ресурсы, ограничения на используемые технологии.

В настоящее время, для повышения эффективности процессов разработки и качества конечного продукта используются подходы, которые представляют собой наборы правил и рекомендаций, а также экспертных оценок, но внедрение только лишь организационных и методологических правил проектирования программных комплексов не предотвращает появления ошибок при эксплуатации программ. Кроме того, очень часто современная практика создания программ базируется лишь на квалификации и интуиции специалистов, что приводит к наличию большого количества не выявленных при тестировании ошибок и низкому качеству программного продукта.

Большинство представленных на рынке и разрабатываемых коммерческих программ представляют собой автоматизированные информационные системы (АИС). При итерационном процессе разработке систем их структура не является постоянной и на каждой итерации может изменяться. В систему постоянно вносятся усовершенствования, что приводит к появлению новых классов и новых связей между классами или к модификациям уже существующих. В процессе доработки первоначальная структура системы может существенно измениться по сравнению с первоначальным состоянием, причем не в лучшую сторону. Ошибки, допущенные при плохо

продуманных преобразованиях структуры системы, могут привести к таким отрицательным последствиям как: ухудшение надежности ПО, повышение стоимости разработки, увеличение сроков разработки и многим другим.

Для выявления допущенных ошибок в структуре ПО применяются различные технологии тестирования и соблюдаются заданные требования безопасности, которым не всегда уделяется должное внимание, в связи с ограниченностью ресурсов. В тоже время квалификация злоумышленников растет каждый день, они становятся опытнее и применяют новейшие технологии атак и взломов ПО. Эти причины обуславливают актуальность задачи разработки менее затратного и более быстродействующего средства для тестирования. Поэтому анализ кода по требованиям безопасности важный этап разработки, который нельзя исключать из этапов тестирования, как, впрочем, и другие виды проверок исходного кода, например, комплексное тестирование ПО. Для автоматизированного анализа кода применяются различные виды сценариев проверок исходных файлов, одним из этих сценариев является тестирование с помощью статического анализа кода [1].

Статический анализ кода – это процесс выявления ошибок и недочетов в исходном коде программ. Статический анализ можно рассматривать как автоматизированный процесс обзора кода [2].

Этапы работы статического анализатора программного кода представлены на рис. 1.

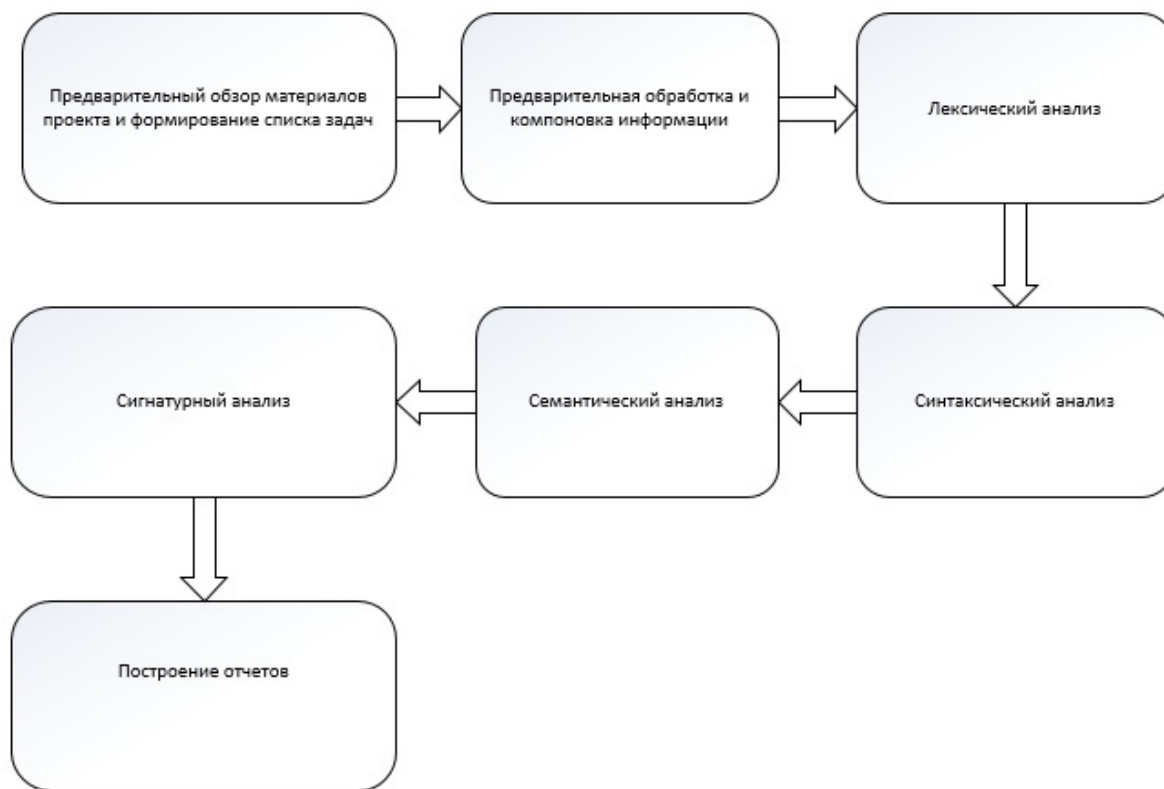


Рис. 1. Общий вид процесса работы статического анализатора кода

#### Методы статического анализа

Существует множество различных методов статического анализа, в частности, анализ с обходом дерева кода, анализ потока данных, анализ потока данных с выбором пути и т. д. Конкретные реализации этих методов различны в разных анализаторах. Тем не менее, статические анализаторы для различных языков программирования могут использовать один и тот же базо-

вый код, или инфраструктуру. Эти инфраструктуры содержат набор основных алгоритмов, которые могут использоваться в разных анализаторах кода вне зависимости от конкретных задач и анализируемого языка. Набор поддерживаемых методов и конкретная реализация этих методов, опять же, будет зависеть от конкретной инфраструктуры. Например, инфраструктура может позволять легко создавать анализатор, использующий обход дерева кода, но не поддерживать анализ потока данных [3].

Хотя все три перечисленные выше метода статического анализа используют дерево кода, построенное синтаксическим анализатором, эти методы различаются по своим задачам и алгоритмам.

Анализ с обходом дерева, как видно из названия, выполняется путем обхода дерева кода и проведения проверок на предмет соответствия кода принятому стандарту кодирования, указанному в виде набора правил и рекомендаций. Именно этот тип анализа проводят компиляторы [4].

Анализ потока данных можно описать как процесс сбора информации об использовании и определении данных, а также об их зависимостях в анализируемой программе. При анализе потока данных используется граф потока команд, генерируемый на основе дерева кода. Этот граф представляет все возможные пути выполнения данной программы: вершины обозначают «прямолинейные», без каких бы то ни было переходов, фрагменты кода, а ребра – возможную передачу управления между этими фрагментами. Поскольку анализ выполняется без запуска проверяемой программы, точно определить результат ее выполнения невозможно. Иными словами, невозможно выяснить, по какому именно пути будет передаваться управление. Поэтому алгоритмы анализа потока данных аппроксимируют возможное поведение, например, рассматривая обе ветви оператора if-then-else, или выполняя с определенной точностью тело цикла while. Ограничение точности существует всегда, поскольку уравнения потока данных записываются для некоторого набора переменных, и количество этих переменных должно быть ограничено, поскольку мы рассматриваем лишь программы с конечным набором операторов. Следовательно, для количества неизвестных всегда существует некий верхний предел, дающий ограничение точности. С точки зрения графа потока команд при статическом анализе все возможные пути выполнения программы считаются действительными. Из-за этого допущения при анализе потока данных можно получать лишь приблизительные решения для ограниченного набора задач [5].

#### Направления статистического анализа

Применение статических анализаторов по своей цели может быть различным и для того, чтобы понимать общие принципы направлений использования анализатора, надо ввести несколько основополагающих терминов, а именно:

- качество (программного средства) – совокупность свойств программного средства, которые обуславливают его пригодность удовлетворять заданные или подразумеваемые потребности в соответствии с его назначением [6];
- надежность (программного средства) – совокупность свойств, характеризующая способность программного средства сохранять заданный уровень пригодности в заданных условиях в течение заданного интервала времени [7];
- дефект – каждое отдельное несоответствие продукции установленным требованиям [7];
- дефект безопасности ПО – ошибка в программном продукте, из-за которой возможно нарушение целостности или конфиденциальности программного продукта при его функционировании.
- эвристический анализатор – это компонент ИС который обнаруживает вредоносные программы путем контроля всех действий выполняемых проверяемой программой. В ходе данной проверки отслеживаются потенциально опасные действия, характерные для вирусов и вредоносных программ других типов. По шаблонам типовых опасных действий на ИС эвристический анализатор современных статических анализаторов способен обнаружить новые, неиз-



вестные вирусы еще до того, как эти вирусы начали действовать [8]. Данное определение применяется в области антивирусных программ, но также его можно применить и обобщить для области статического анализа.

Эвристический анализатор (в области статического анализа) – это компонент ИС, который обнаруживает сложнотерминированные дефекты путем анализа действий пользователя, а также путем обучения, а значит совершенствования алгоритмов обнаружения уязвимостей. На рис. 2 показан общий алгоритм работы эвристического анализа. На вход эвристического анализатора поступают данные в виде шаблона дефекта, полученные в результате работы статического анализатора.

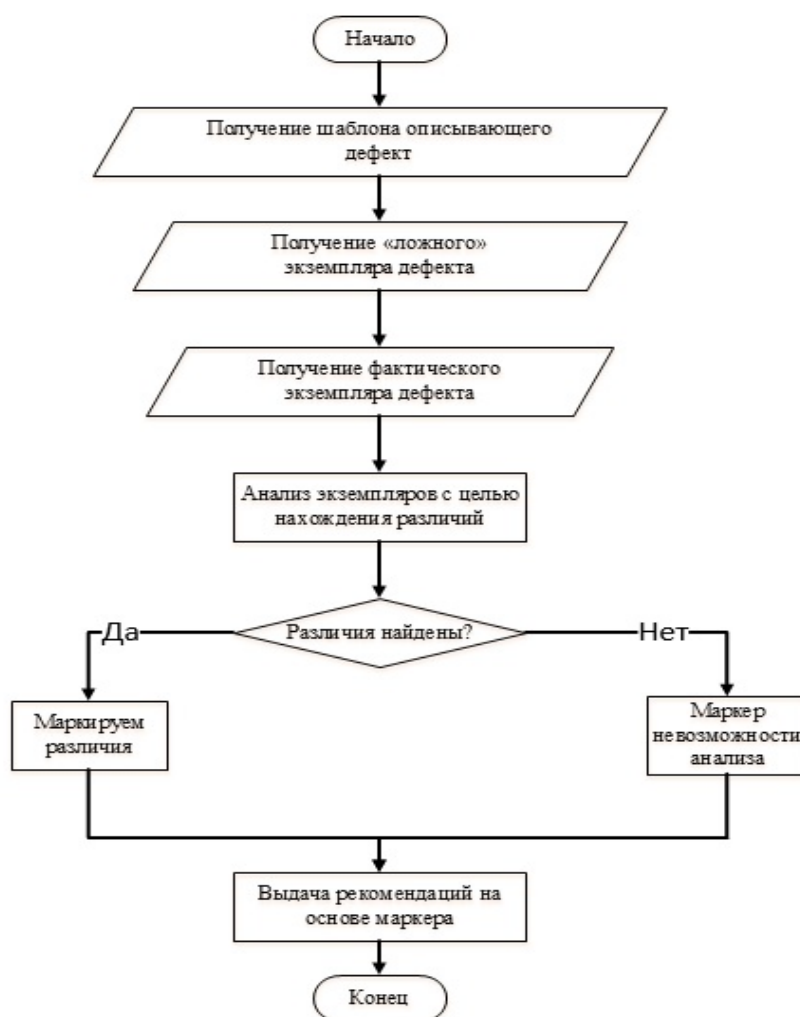


Рис. 2. Общий алгоритм работы эвристического анализа

Направления для применения статических анализаторов можно разделить на несколько подгрупп, а именно: декомпозиции и анализа структуры ПО (с целью анализа качества ПП), проверки стиля программирования и подсчета метрик с целью анализа архитектуры (designreview) (с целью анализа как качества, так и надежности ПП), проведения рецензирования или инспекции кода (codereview) (с целью анализа качества и поиска дефектов). Так же основными направлениями работы анализаторов являются проверка свойств (ограничений) программы на моделях, поиск дефектов (багов) по образцу (сигнатурный анализ). Дефекты безопасности ПО являются сложнотерминированным случаем, как со стороны поиска таких дефектов, так и со стороны их

исправления (имеется ввиду автоматизированное исправление дефектов), поэтому частично решить данную проблему может совместное применение статического и эвристического анализаторов, а соответственно сигнатурно-эвристический поиск дефектов.

Построение шаблонов для обнаружения дефектов ПО логично будет начать с классификации уязвимостей и дефектов.

В таблице 1 приведена общая классификация уязвимостей и дефектов ПО [9].

Таблица 1

Общая классификация уязвимостей и дефектов ПО

Уязвимость кода	Метод тестирования	Комментарии
Ошибки, возникающие в случае применения редко используемых входных данных	Функциональное тестирование на редко используемых входных данных	Проверка декларированных в документации функциональных возможностей для редко используемых входных данных, таких как недопустимые входные данные, граничные значения и др.
Недекларированные входные параметры и режимы, связанные с деградацией производительности системы	Профилирование производительности	Выявление фрагментов кода, в которых возникает деградация производительности (несанкционированная передача данных, переход в «бесконечный цикл», обработка необъявленных /входных параметров и др.)
Ошибки, связанные с отказом в обслуживании	Нагрузочное и стрессовое тестирование	Создание специфических условий для работы программы с большим объемом входных данных, при увеличенной нагрузке на процессор, наличии бесконечно посылаемых сообщений, в условиях минимально допустимых аппаратных ресурсов.
Уязвимости подсистемы парольной защиты	Тестирование подсистемы аутентификации, управления доступом и других подсистем безопасности	Определение режимов управления безопасностью, трассировка парольной системы, поиск встроенных паролей
Программные закладки	Сигнатурно-эвристический анализ	Выявление программных закладок по сигнатуре потенциально опасных операций и по событиям
Некорректности кодирования	Сигнатурно-эвристический анализ	Выявление программных закладок по сигнатурам конструкций
Хулиганский код, «подписи программистов», «пасхальные яйца»	Сигнатурно-эвристический анализ	Выявление программных закладок по осмысленным сообщениям, константам, идентификаторам, фактам скрытия «горячих» клавиш (по признакам нелегитимной деятельности программистов)

## ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ

Скрытые каналы	Анализ трафика и памяти в изолированной среде	Мониторинг и аудит журналов
Программные закладки, иницируемые скрытым переходом	Контроль над избыточностью, сигнатурно-эвристический анализ	Контроль над избыточностью и выявление программных закладок по конструкциям скрытой передачи управления

Число методов статического анализа и самих анализаторов увеличивается из года в год, и это означает, что интерес к статическим анализаторам кода растет. Причина заинтересованности заключается в том, что разрабатываемое программное обеспечение становится все более и более сложным и, следовательно, проверять код вручную становится невозможно.

В этой статье было приведено краткое описание процесса статического анализа и различных методов его проведения. Так же были даны определения важным терминам в сфере статического анализа, такие как качество программного средства, его надежность, дефект (как общий термин) и дефект безопасности ПО. В табл. 1 была дана краткая классификация уязвимостей.

Обнаружение уязвимостей в исходном коде программы с помощью статического анализатора, зависит от нескольких критериев, таких как: проверяемый язык программирования, библиотека шаблонов для сигнатурного анализа, гибкость модели статического анализатора. Все перечисленные критерии (при успешной их реализации) существенно повышают качество обработки исходного кода программы. Под качеством в данном случае понимается общее количество найденных уязвимостей к общему количеству ложных срабатываний СА.

Перечисленные выше критерии качества являются основой любого статического анализатора, но не отвечают полностью существующим требованиям по качеству оценки кода. С развитием как статических анализаторов, так и проверяемых программ все большую популярность набирает идея разработки эвристического анализатора кода (в дополнение к статическому анализу), который при грамотно проработанной модели уменьшает количество ложных срабатываний статического анализатора и увеличивает общее количество найденных уязвимостей в исходном коде программы, но разработка эвристического анализатора, который будет отвечать всем современным требованиям ИБ подчас становится сложной задачей, которая требует как большого количества времени, так и большого багажа знаний конкретной предметной области разрабатываемого ПП.

### Библиографический список

1. Статический анализ кода: что могут инструментальные средства? Е. Трошина – 2014 г. – [Электронный ресурс] – режим доступа: <http://www.jetinfo.ru/stati/staticheskij-analiz-koda-cto-mogut-instrumentalnye-sredstva>
2. Статический анализ кода [Электронный ресурс] – режим доступа [https://ru.wikipedia.org/wiki/Статический\\_анализ\\_кода](https://ru.wikipedia.org/wiki/Статический_анализ_кода)
3. Выявление уязвимостей в программном коде / А. С. Марков, С. В. Миронов, В. Л. Цирлов // Открытые системы. СУБД. 2005. № 12. С. 64–69.
4. Ciera Nicole Christopher Evaluating Static Analysis Frameworks [Электронный ресурс] – Ciera Nicole, cop. 2006. – режим доступа: <http://www.viva64.com/go.php?url=64>
5. Leon Moonen A Generic Architecture for Data Flow Analysis to Support Reverse Engineering [Электронный ресурс]. – Proceedings of the 2nd International Workshop on the Theory and Practice of Algebraic Specifications, cop. 1997.
6. ГОСТ Российской Федерации 28806-90 Качество программных средств. Термины и определения.
7. ГОСТ Российской Федерации 27.002-89 Надежность в технике основные понятия. Термины и определения.
8. Эвристический анализ [Электронный ресурс] – режим доступа: [https://ru.wikipedia.org/wiki/Эвристическое\\_сканирование](https://ru.wikipedia.org/wiki/Эвристическое_сканирование)
9. Выявление уязвимостей в программном коде, Алексей Марков, Сергей Миронов, Валентин Цирлов [Электронный ресурс] – режим доступа: <http://www.osp.ru/os/2005/12/380655/>

**В. В. Виноградов**

студент военной кафедры

**П. М. Безняков**

канд. техн. наук, доц. – научный руководитель

### ДЛИННАЯ АРИФМЕТИКА

ЭВМ могут оперировать числами, количество бит которых ограничено в их двоичном представлении. Современные персональные ЭВМ оперируют 32-х или 64-х битными числами. Но при создании программного обеспечения в областях криптографии, математики, финансов и интернет безопасности часто возникает необходимость вычислений с числами, которые «не помещаются» в 64-х разрядный или 32-х разрядный тип данных. Например, криптографический алгоритм шифрования с открытым ключом RSA оперирует числами, двоичное представление которых значительно превышает разрядную сетку персональных ЭВМ. В данном алгоритме при шифровании информации необходимо возводить число в степень, что даже при относительно небольших значениях может вызвать переполнение стандартного типа данных. А так как в настоящее время лаборатория RSA рекомендует для обычных задач ключи размером 1024 бита, а для особо важных задач 2048 битов, то очевидно, что стандартных типов данных даже для хранения таких ключей будет недостаточно. В настоящей статье рассматриваются программные способы решения вычислительных задач с числами, разрядность которых превосходит длину машинного слова ЭВМ. Такие способы объединены названием – «длинная арифметика».

Длинная арифметика – в вычислительной технике это операции: сложение, умножение, вычитание, деление, возведение в степень и другие над числами, разрядность которых превышает длину машинного слова данной вычислительной машины. Эти операции реализуются не аппаратно, а программно, используя базовые аппаратные средства работы с числами меньших порядков. Это позволяет в некоторых случаях ограничивать размер чисел лишь доступной оперативной памятью.

Хранить длинные числа можно разными способами. Рассмотрим один из вариантов. Для хранения длинного целого числа можно использовать целочисленный массив, где в качестве элемента массива будет одна цифра десятичного числа. Так в 1-м элементе массива хранится число единиц, во 2-м – число десятков и т. д. до самого старшего разряда числа. В 0-м элементе хранится общее количество цифр в числе. Таким образом, число размещается в массиве как бы задом наперед. Это необходимо для удобства организации вычислительного процесса т. к. все арифметические операции выполняются от младших разрядов к старшим. Для знаковых чисел знак учитывается в отдельной переменной. Для ускорения операций с длинными числами в одном элементе массива можно хранить не одну, а несколько цифр числа.

Операции с длинными числами производятся с помощью классических алгоритмов сложения, вычитания, умножения, деления «в столбик», то есть фактически мы работаем с небольшими числами, сводя более сложную задачу к набору более простых подзадач. К ним также применимы и другие алгоритмы, например алгоритмы быстрого умножения [1]:

- Карацубы;
- Тоома;
- быстрое преобразование Фурье.

Рассмотрим кратко данные алгоритмы умножения.

Базовый алгоритм умножения это, по сути, алгоритм умножения двух чисел в столбик. При умножении двух длинных чисел в столбик, первое число последовательно умножается на каждую цифру второго. Результат умножения на  $i$  цифру прибавляется к общему результату со сдвигом на  $i - 1$ . Сложность вычисления алгоритма равна  $O(n^2)$ , где  $n$  – количество разрядов в перемножаемых числах.

Алгоритм Карацубы сводит задачу умножения двух чисел к нескольким задачам умножения чисел меньшей разрядности. Разбиение может осуществляться рекурсивно до тех пор, пока разрядность не уменьшится до поддерживаемой аппаратно (т. е. пока  $n$  не достигнет размера машинного слова). Сложность вычисления данного алгоритма равна  $O(n^{\log_2 3})$ .

Алгоритм Тоома является обобщением алгоритма Карацубы. Алгоритм Тоома делит длинные числа на  $k$  меньших частей, длина каждой из которых равна длине машинного слова, и производит операции над этими частями. Сложность вычисления алгоритма равна  $O(n^{\ln(2k-1)/\ln(k)})$ .

Преобразование Фурье, позволяют перемножить два полинома длиной  $n$ . Очевидно, что умножение двух длинных чисел можно свести к умножению полиномов, поэтому два длинных числа также можно перемножить данным алгоритмом. Сложность вычисления алгоритма равна  $O(n \log n)$ .

Рассмотренные алгоритмы реализованы в современных библиотеках языков программирования. Так на платформе .NET Framework (начиная с 4-й версии) появилась поддержка длинной арифметики [2]. Например, для создания приложения, в котором она необходима, на языке программирования C++ в среде разработки Visualstudio нужно создать проект среды CLR. В программе директивой «#using<System.Numerics.dll>» подключить библиотеку System.Numerics.dll. Затем строкой «using namespace System::Numerics;» объявить пространство имен System.Numerics содержащее числовые типы, которые дополняют числовые типы-примитивы:

- тип BigInteger который является сложным целочисленным типом данных не имеющим нижней или верхней границы. Он является полноценной реализацией длинной арифметики.

- тип Complex, который представляет собой комплексное число в форме  $a + ib$ , где  $a$  – действительная часть,  $b$  – мнимая часть.

Экземпляр BigInteger можно использовать точно так же, как и любой другой целочисленный тип. BigInteger перегружает стандартные числовые операторы для выполнения основных математических операций, таких как сложение, вычитание, деление, умножение, отрицание и унарное отрицание. Можно также использовать стандартные числовые операторы для сравнения двух значений BigInteger друг с другом. Как и другие типы целого числа BigInteger поддерживает битовые операторы And, Or, XOR, сдвиг влево и сдвиг вправо.

Также в BigInteger реализованы следующие функции:

- IsEven – определяет, является ли число четным;
- IsPowerOfTwo – определяет, является ли число степенью числа 2;
- Sign – возвращает значение, указывающее знак числа BigInteger;
- Abs – возвращает абсолютное значение числа BigInteger;
- DivRem – возвращает частное и остаток от операции деление;
- GreatestCommonDivisor – возвращает наибольший общий делитель для двух чисел;
- Log – возвращает логарифм указанного числа в системе счисления с указанным основанием;
- Max / Min – возвращает наибольшее (наименьшее) из двух чисел;
- ModPow – выполняет модульное деление числа, возведенного в степень другого числа;
- Pow – возводит значение BigInteger в заданную степень.

В настоящее время объемы обрабатываемой информации и вычислительные мощности персональных ЭВМ растут. Вместе с ними и растет потребность в использовании длинной арифметики, что приводит к тому, что многие современные языки программирования, такие как Java, Ruby, Python, Lisp, Perl, Haskell имеют встроенную поддержку длинной арифметики, что позволяет в разы сократить время написания программы. Для других языков программирования существуют библиотеки для работы с длинными числами. Одна из таких библиотек, GMP (GNU Multi-Precision Library). Таким образом, на сегодняшний день существует большой выбор инструментов

позволяющих решать задачи вычислений с числами, разрядность которых превосходит длину машинного слова ЭВМ.

### Библиографический список

1. Кнут Д. Э. Искусство программирования. М.: Вильямс, 2001. Т. 2. Получисленные алгоритмы. 832 с.
2. Ватсон Б. C# 4.0 на примерах. СПб.: БХВ-Петербург, 2011. 608 с.

УДК 004.021

**В. В. Виноградов**

магистрант кафедры вычислительных систем и сетей

**А. В. Никитин**

канд. техн. наук, доц. – научный руководитель

## ОПТИМИЗАЦИЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ ПРИ ТРЕХМЕРНОМ МОДЕЛИРОВАНИИ

Область применения и задач компьютерной графики постоянно расширяется вместе с непрерывным ростом аппаратных возможностей компьютеров. Вместе с этим разрабатываются новые алгоритмы для решения возникающих задач, которые рассчитаны на современное аппаратное обеспечение. Большая часть актуальных проблем визуализации связана с эффективным хранением и отображением больших объемов данных с минимальными задержками.

Одной из самых широких сфер применения компьютерной графики являются интерактивные системы визуализации трехмерных сцен. Под интерактивностью понимается возможность визуализации сцены в режиме реального времени, причем частота выводимых кадров (FPS) итогового двумерного изображения сцены должна быть не меньше частоты кадровой развертки монитора. К таким системам относятся компьютерные игры, тренажеры имитирования управления реальной техникой, системы виртуальной и дополненной реальности. При сохранении такого уровня временной эффективности, должен обеспечиваться оптимальный уровень качества картинки. Кроме того, структура данных сцены должна занимать объем памяти не более определенного размера.

С целью повышения эффективности отображения трехмерных сцен разрабатываются специальные алгоритмы, основанные на методах компьютерной геометрии и машинной графике. Задачами этих алгоритмов является отсечение невидимых частей и отображение видимых оптимальным образом. Для этих операций разрабатываются специальные структуры данных и алгоритмические решения. Широкая сфера применения компьютерной графики и многообразие практических задач обуславливает необходимость разработки новых и усовершенствования старых алгоритмов и структур данных для эффективного воспроизведения этих сцен и улучшения временных показателей визуализации. Таким образом, разработка и реализация методов увеличения временной и емкостной эффективности процесса визуализации является актуальной проблемой в области создания эффективного программного обеспечения вычислительных машин в сфере компьютерной графики.

Одним из алгоритмов повышения временной эффективности отображения трехмерных сцен является механизм управления уровнем детализации (levelofdetail – LOD). Это позволяет добиваться компромисса между точностью отображения сцены и скоростью ее реализации. Основная идея этого алгоритма заключается в том, что при рендеринге объектов, которые на данный момент удалены от камеры и плохо видны, следует использовать их менее детализированное представление, нежели когда эти объекты оказываются близко к камере (рис. 1). Уровнем детализации называется некоторая аппроксимация начальной модели объекта с использованием меньшего числа геометрических объектов – полигонов. Уменьшение количества полигонов

на сцене позволяет снизить нагрузку на вычислительные ресурсы и повысить производительность визуализации.



Рис. 1. Уровни детализации LOD

Данный алгоритм активно применяется в различных системах и встроен во многие системы производства мультимедиа продуктов, что обеспечивает простоту его использования.

Проведя анализ существующих решений по использованию механизма LOD в различных системах, можно сделать вывод, что все решения базируются на использовании механизма статического LOD, при котором уровень детализации объекта подгружается в соответствии с величиной метрик [2]. Главным недостатком подобных методов является необходимость создания нескольких представлений каждого объекта, а также хранение их на жестком диске. Другим не менее важным недостатком является эффект «перепрыгивания», заключающийся в заметной для пользователя смене детализации объекта. Данный эффект наблюдается при недостаточном большом количестве созданных представлений различной детализации. Кроме того данный метод неприменим для достаточно крупных объектов, таких как ландшафт, поскольку хранение и подгрузка нескольких представлений объекта, обладающего большим объемом информации дело затратное.

Для решения этих недостатков предлагается реализовать динамический видозависимый алгоритм LOD [1]. Этот подход применяется для избирательного улучшения произвольного трехмерного объекта в соответствии с изменением параметров обзора. Отличие этого метода от базового состоит в том, что при использовании техники статического алгоритма LOD уровни детализации объектов создаются заранее, в то время как в динамическом видозависимом алгоритме LOD вместо этого создается структура данных, из которой геометрия с необходимой детализацией может быть извлечена в реальном времени во время визуализации.

В этом случае предлагается хранить визуализируемый объект в виде набора, состоящего из грубого приближения полигональной сетки и карты смещения. Степень детализации объекта увеличивается с использованием разбиения полигонов на более мелкие в реальном времени. Разбиение позволяет увеличить степень детализации визуализируемого объекта посредством разбиения полигонов исходного объекта на более мелкие в соответствии с картой смещения.

Недостатком данного решения может являться повышение нагрузки на вычислительные устройства, на которые дополнительно накладывается работа по разбиению полигонов.

Реализация динамического видозависимого алгоритма LOD и адаптация его для использования в существующих игровых движках позволит улучшить качество использования управления уровнями детализации и избежать недостатков использования статического алгоритма.

**Библиографический список**

1. Плессер П. Л. Видозависимая tessеляция 3D-объектов как инструмент управления динамическим уровнем детализации // Научно-технический журнал «Информационные технологии в проектировании и производстве». 2011. № 3. С. 82–87.
2. UnityLevelOfDetail [Электронный ресурс] – Режим доступа: <http://docs.unity3d.com/Manual/LevelOfDetail.html>. свободный, дата обращения 12.04.2015.
3. MeshBakerLOD [Электронный ресурс] – Режим доступа: <http://forum.unity3d.com/threads/mesh-baker-lod-released.208359>. свободный, дата обращения 17.04.2015.

УДК 004.832.24

**В. В. Виноградов**

магистрант кафедры вычислительных систем и сетей

**А. В. Никитин**

канд. техн. наук, доц. – научный руководитель

**ПРОЕКТ СТРАТЕГИЧЕСКОЙ ИГРЫ. ИНТЕЛЛЕКТУАЛИЗАЦИЯ ГЕЙМПЛЕЯ**

По данным исследований рынок компьютерных игр растет с каждым годом и обладает большим потенциалом для развития. Особенно быстро развивается рынок многопользовательских и социальных игр. Одним из примеров таких игр являются стратегические игры. Стратегические игры – жанр компьютерных игр, в которых залогом достижения победы является планирование и стратегическое мышление. Смысл таких игр заключается в управлении некоторыми ресурсами, которые необходимо преобразовать в преимущество над соперником при помощи плана, который может меняться с учетом меняющейся обстановки. Управление этими ресурсами происходит с помощью игровых персонажей (так называемых юнитов).

Для придания реалистичности игре, управляемые персонажи обладают игровым искусственным интеллектом (ИИ), что позволяет им действовать и принимать решение подобно тому, как это делают живые организмы.

Основными классами задач в игровом искусственном интеллекте являются реакция и поиск пути.

Под реакцией понимаются действия персонажа при попадании в зону влияния какого-либо объекта. Например, если персонаж попал в поле зрения какого-либо враждебного объекта, данный объект может начать преследование персонажа до тех пор, пока персонаж не выйдет из его поля видимости.

Поиск пути – задача нахождения оптимального маршрута между двумя точками пространства [1]. Критерии оптимальности могут быть различными в зависимости от решаемой задачи. Как правило, к этим критериям относится кратчайшее расстояние и отсутствие препятствий на найденном маршруте.

Алгоритмы поиска пути используются в симуляторах и компьютерных играх, чтобы обеспечивать корректное перемещение персонажей, управляемых компьютером. Также алгоритмы поиска пути используются в GPS навигаторах, для того чтобы подсказать нам кратчайший по времени и расстоянию путь. Это показывает то, что разработка ИИ в компьютерных играх, позволяет в дальнейшем переносить найденные решения в повседневную жизнь.

В контексте разрабатываемой игры необходимо определить алгоритм поиска пути для игровых персонажей, которые должны проходить через игровой уровень (карту) по заданным игроком начальной и конечной точкам движения, минуя препятствия. Препятствиями в данном случае будут являться элементы ландшафта, строительные сооружения, природные объекты, а также другие персонажи. Кроме того, необходимо учесть, что по карте может перемещаться одновременно не один игрок, а несколько, что требует учесть разрешение конфликтов при столкновении объектов друг с другом и с динамическими объектами.



Разрабатываемая игра относится к жанру стратегии реального времени, где в отличие от пошаговых стратегий, ходы игроков выполняются не поочередно с задержкой, а в режиме реального времени, что накладывает определенные ограничения на ресурсоемкость вычислений, так как реакция игры на действие пользователя должна быть мгновенной. Проблема экономии вычислительных ресурсов особенно актуальна, так как предполагается, что проектируемая игра будет являться кроссплатформенной и исполняться на малопроизводительных платформах, таких как планшет или смартфон.

Также необходимо учесть, что разрабатываемая игра является мультиплеерной, что требует отображать движения игроков другим пользователям. Расчет пути персонажа не должен влиять на скорость синхронизации игроков и количество передаваемых данных.

Основным методом поиска кратчайшего пути с учетом препятствий является алгоритм A\* (A звездочка). Он предполагает разбиение всего пространства на клетки одинаковой формы. Такая структура позволяет представить область поиска в форме двумерного массива, каждый элемент которого представляет собой одну из клеток сетки, а его значением будет проходимость этой клетки (проходима или непроходима) [1].

Для нахождения пути необходимо определить, какие клетки использовать для перемещения из точки A в точку B. Для начальной точки составляется список соседних клеток. Для каждой клетки определяется стоимость прохождения F. Способом определения того, какую клетку использовать является выражение

$$F = G + H,$$

где G – стоимость передвижения из стартовой точки A к данной клетке, следуя найденному пути; H – эвристическая оценка передвижения от данной клетке к целевой. Стоимость H рассчитывается с помощью вычисления Манхэттенского расстояния к красной целевой клетке, двигаясь только по горизонтали и вертикали, игнорируя все стены на пути [2].

На каждом этапе алгоритм оперирует с множеством путей из начальной точки до всех еще не раскрытых вершин (рис. 1) [3]. Из множества решений выбирается решение с меньшей стоимостью. Алгоритм продолжает работу до тех пор, пока значение F целевой вершины не станет меньше чем любое значение в списке открытых вершин, либо пока все дерево не будет просмотрено.

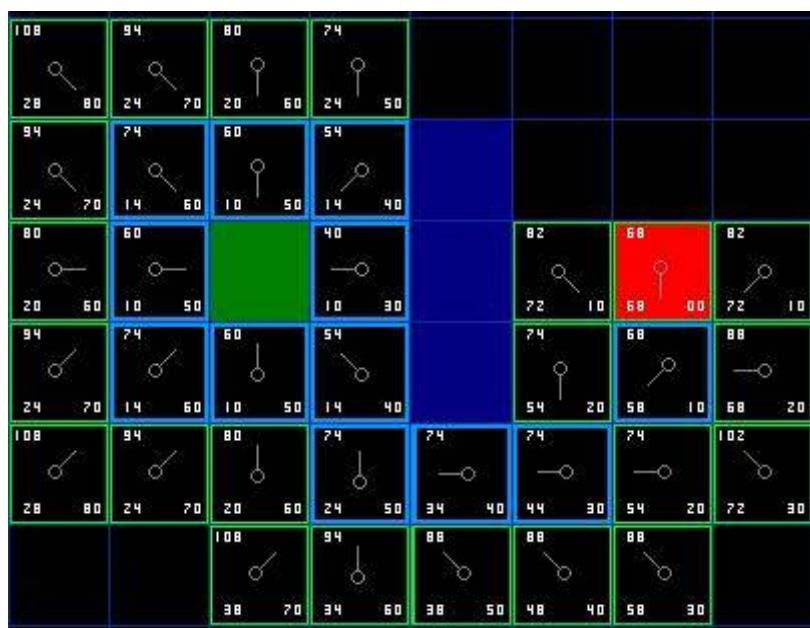


Рис. 1. Разбиение пространства поиска пути

Результат работы алгоритма представлен на рис. 2. Данный алгоритм имеет существенный недостаток. При большом размере карты приходится генерировать большое количество ячеек с последующей их обработкой.

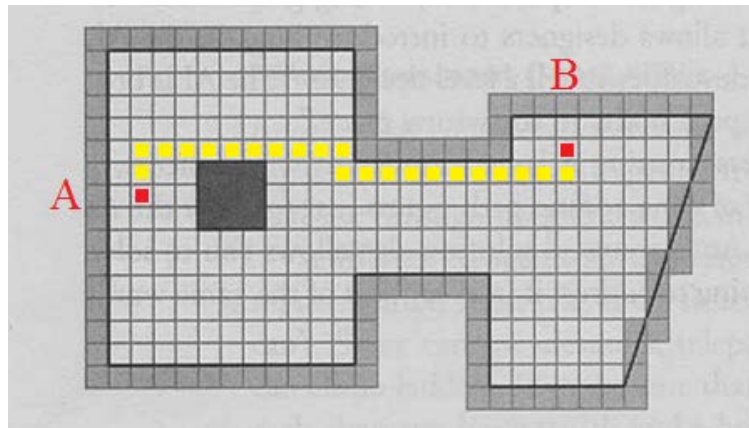


Рис. 2. Результат работы алгоритма A\*

Данный недостаток можно исправить путем изменения структуры данных алгоритма. Пространство поиска разбивается не на ячейки одинакового размера, а на так называемые навигационные меши – полигоны, которые описывают проходимое пространство на карте (рис. 3). Кроме полигонов задаются также точки переходов между полигонами.

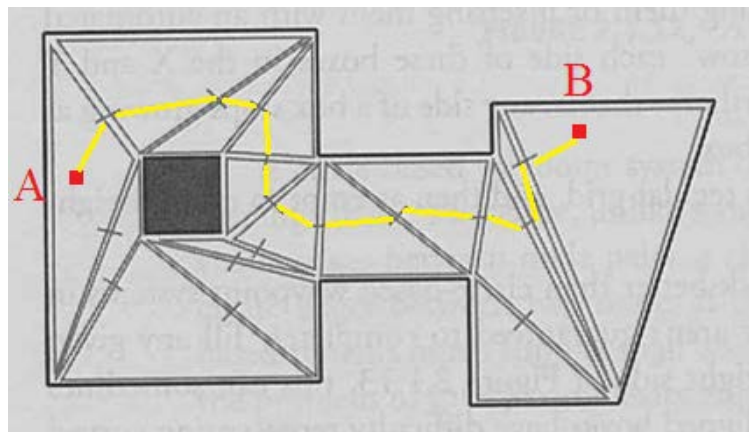


Рис. 3. Разбиение пространства поиска с использованием мешей

Объект движется из точки A в точку B. В этом случае стартовая точка оказывается не в том же полигоне, что конечная, поэтому алгоритм должен выбрать, в какой следующий полигон ему двигаться. После выбора полигона, программа определяет точку перехода между этими полигонами и строит алгоритм по прямой к этой точке. Это повторяется до тех пор, пока объект и конечная точка не окажутся в одном полигоне. В этом случае объект может перейти к точке назначения по прямой. В сравнении с базовым алгоритмом A\* алгоритм навигационных мешей находит кратчайший путь, обрабатывая при этом в разы меньше данных, чем при разбиение на регулярную сетку. Кроме того, такой путь выглядит более естественным для человека.

Недостатком данной методики является требование заранее рассчитывать полигоны при каждом изменении сцены, что недопустимо для сцены с часто меняющимся расположением объектов.

Еще одним возможным решением проблемы производительности при расчете маршрута является запоминание наиболее популярных маршрутов. Игровым персонажам зачастую приходится передвигаться по одним и тем же маршрутам регулярно. На наиболее популярных маршрутах движения персонажей образуются тропы и уже не требуется рассчитывать маршрут каждый раз, достаточно взять его из памяти.

Так как описанные структуры данных имеют свои достоинства и недостатки, разрабатываемой игре предлагается совмещение методик разбиения пространства для алгоритма A\*. Карта игры имеет большой размер и элементы ландшафта, не меняющие своего положения по ходу игры, что позволяет использовать разбиение карты на полигоны на этапе разработки игры. В то же время персонажи перемещаются, как правило, в небольшом пространстве, которое может часто меняться путем добавления новых объектов на нем, чаще всего в пределах одного полигона, что позволяет динамически разбить полигон на регулярную сетку с ячейками квадратной формы при каждом расчете пути.

Так как в момент следования по пути, некоторые динамические объекты могут изменить свою позицию и угол поворота, то необходим механизм разрешения коллизии. В таких случаях предлагается находить путь заново из точки столкновения, учитывая динамический объект как препятствие.

Игра является мультиплеерной, что означает необходимость отображения движения персонажа другим пользователям. Для упрощения синхронизации путь рассчитывается только для своего персонажа, а другим пользователям отображается лишь фактическое передвижение персонажей по карте.

Использование предложенного подхода расчета пути должно уменьшить объем вычислений и позволить применять алгоритм в реальном времени на малопроизводительных платформах.

### **Библиографический список**

1. Международная конференция по компьютерной графике и зрению: Москва, МГУ имени М. В. Ломоносова, 26–30 сентября 2011 г.: Труды конференции. – М.: МАКС Пресс, 2011. – 270 с. ISBN 978-5-317-03808-3, УДК 004.9, ББК 32.973.26-018.2, режим доступа: <http://www.graphicon.ru/proceedings/2011/conference/gc2011vinokurova.pdf>.
2. AungSithuKyaw, Clifford Peters, ThetNaing Swe. Unity 4.x Game AI Programming/Packt Publishing, 2013.232 p.
3. TheMechanicsofStarcraft 2 [Электронный ресурс] – Режим доступа: <http://www.teamliquid.net/forum/starcraft-2/132171-the-mechanics-of-sc2-part-1>, свободный, дата обращения 10.04.2015.

УДК 004.42

**Ю. Д. Гнутова**

студент кафедры компьютерной математики и программирования

**А. А. Ключарев**

канд. техн. наук, доц. – научный руководитель

### **ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ СОЗДАНИЯ РАБОЧИХ ПРОГРАММ ДИСЦИПЛИН НА КАФЕДРЕ**

Планирование и организация учебного процесса невозможна без мощной информационной поддержки. Слабое развитие информационных систем в образовательной отрасли делает актуальной задачу разработки автоматизированных информационных систем поддержки планирования и организации учебного процесса.

Внедрение информационных технологий в процессы управления кафедрой университета позволяют повысить эффективность и качество учебного процесса и, как следствие, выпуска компетентных и востребованных на рынке труда специалистов.

Образовательная деятельность вуза представлена такими процессами, как довузовская подготовка абитуриентов, учебно-организационная деятельность, методическая деятельность, научно-исследовательская работа и реализация образовательного процесса.

Основная цель методической деятельности вуза – создание условий, способствующих повышению эффективности и качества образовательного процесса (обеспечение учебными изданиями, повышение педагогического мастерства преподавателей, улучшение материально-технического обеспечения, совершенствование аудиторной и самостоятельной работы студентов) [1]. Основным результатом методической деятельности являются учебно-методические комплексы дисциплин и учебные издания.

Автоматизация учебных процессов на кафедрах университета является важным шагом в комплексной автоматизации всего вуза.

Под учебно-методическим комплексом дисциплины (далее – УМКД) понимается комплекс, объединяющий учебно-методические, программно-технические и иные средства, обеспечивающие качественное освоение обучающимися содержания дисциплины, и позволяющий эффективно формировать общие и профессиональные компетенции. УМКД предназначен как преподавателям, так и обучающимся. В состав компонентов УМКД входят утвержденные рабочие программы дисциплин по всем реализуемым формам обучения.

Под рабочей программой дисциплины (РПД) понимается нормативный документ, определяющий объем, содержание, порядок изучения и преподавания каждой учебной дисциплины базового учебного плана, а также способы контроля результатов ее усвоения, соответствующие требованиям направлений подготовки и учитывающие специфику подготовки студентов по избранному направлению.

Необходимость автоматизации системы рабочих программ дисциплин продиктована проблемами, возникающими при их создании, редактировании, утверждении, хранении и поиске:

- потеря документа, либо его долгий поиск (большой объем документации по рабочим программам);
- поиск документов в рамках университета затруднен из-за их децентрализованного хранения;
- возможная задержка создания и утверждения РПД (сложные схемы согласования документов и человеческий фактор);
- проблемы контроля создания рабочей программы, в том числе сложность проверки на полную укомплектованность РПД и отслеживания ошибок в рабочих программах дисциплин;
- избыточность документооборота (в т.ч. дублирование документов);
- большой штат сотрудников, работающих с документами.

В ходе работы были решены следующие задачи:

- изучены требования к структуре, содержанию и оформлению рабочей программы учебной дисциплины, порядок разработки и утверждения;
- проанализирован процесс разработки и утверждения рабочих программ дисциплин и построена процессная модель;
- разработано техническое задание к проектируемой системе;
- спроектирована информационная система;
- реализован базовый функционал системы;
- проанализированы полученные результаты.

Структурными элементами рабочей программы являются [2]:

- титульный лист;
- лист согласования рабочей программы;
- основная часть.

На рис. 1 показаны структурные элементы РПД и то, в какие части рабочей программы поступают данные из учебного плана (далее УП).

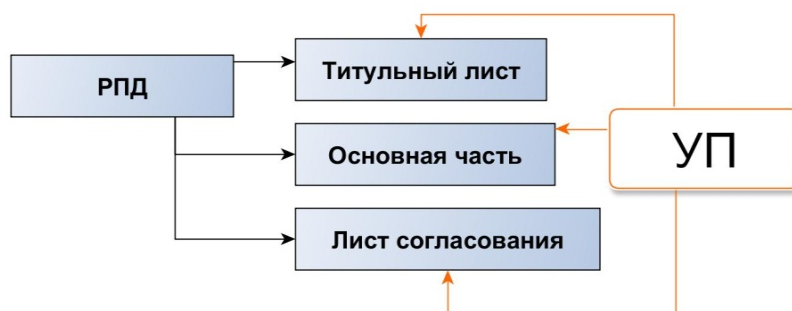


Рис. 1. Структура рабочей программы дисциплины

На рис. 2 показаны элементы титульного листа РПД и то, что эти данные поступают из учебного плана.



Рис. 2. Структура титульного листа РПД

Основная часть рабочей программы дисциплины содержит:

- цели и задачи дисциплины;
- компетенции обучающегося, формируемые в результате освоения дисциплины;
- междисциплинарные связи;
- объем дисциплины и виды учебной работы;
- разделы дисциплины, их содержание и виды занятий;
- тематический план изучения дисциплины (по семестрам);
- учебно-методическое обеспечение дисциплины;

- перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине;
  - материально-техническое обеспечение дисциплины.
- На рис. 3 показана структура основной части РПД и то, в какие данные поступают из учебного плана.

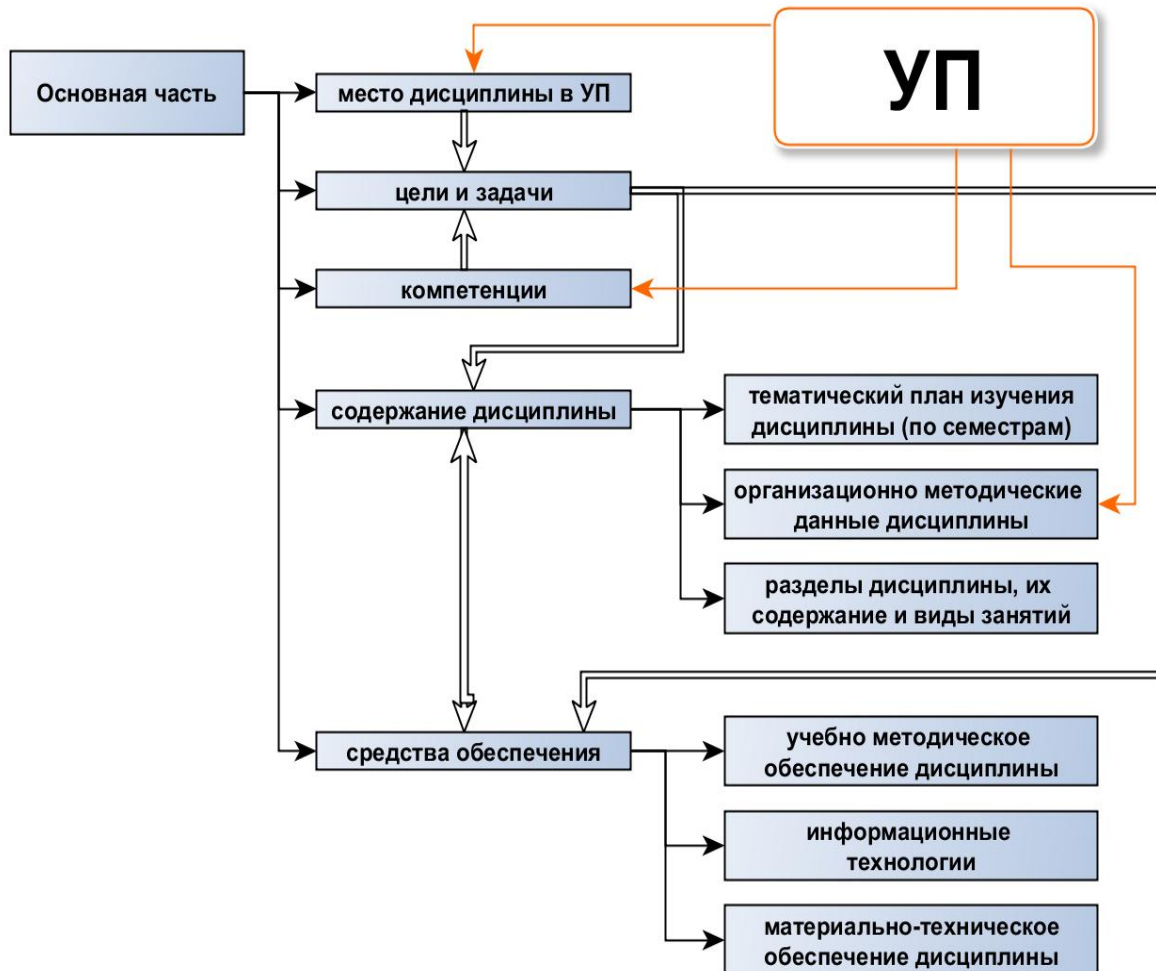


Рис. 3. Структура основной части рабочей программы дисциплины

Система имеет разграниченный доступ к данным. В системе есть три вида прав доступа, это Администратор, Методист, Составитель.

Составителю доступен просмотр списков всех РПД, которые он имеет право редактировать. Одним из предложенных решений было составление РПД поэтапно, т. е. составитель имеет возможность выбирать, какую часть рабочей программы он будет разрабатывать в следующий момент. Выбрав РПД из списка, он имеет право изменять все ее составные части, автоматически генерировать печатную форму РПД и отправлять на утверждение документа.

Администратор имеет наиболее полные права доступа. Администратор может составлять списки пользователей, присваивать статусы уже существующим пользователям системы, редактировать учетные записи.

Методисту доступен просмотр списков всех РПД, за которые он ответственен:

- список готовых РПД – составленных и утвержденных;

- список разрабатываемых РПД – у РПД есть составитель, который разрабатывает РПД, но еще не отправил на утверждение Методисту;
- список отправленных на доработку РПД – разработанные и отправленные составителями РПД, которые требуют доработки;
- список ждущих РПД, которые уже есть в системе, но у которых еще не определен составитель.

В отличие от Составителя, Методист имеет доступ к тем же окнам, что и составитель только он не имеет права изменять РПД, в его праве утвердить РПД, после чего статус РПД изменится на «Готов», или отправить на доработку составителю с запиской о том, что следует изменить, после чего статус РПД изменится на «Отправлен на доработку». Методист имеет право выбирать составителей для РПД.

Разработанный базовый функционал обеспечивает:

- возможность импорта данных из базы данных УМО во внутреннюю БД системы на уровне файлов;
- разграничение прав доступа к системе;
- выгрузку данных из внутренней БД системы в пользовательский интерфейс с учетом всех требуемых правил представления данных;
- удобный пользовательский интерфейс и наглядность представления информации;
- поддержку уже имеющихся РПД и составление новых;
- автоматический расчет трудоемкости учебных работ;
- возможность согласования и утверждения составленных рабочих программ;
- сохранение всех проделанных изменений в базу данных;
- возможность автоматического формирования документа.

Предложенная структура проектируемой системы, разработанное техническое задание и реализованный базовый функционал являются основой для создания автоматизированной информационной системы создания рабочих программ дисциплин на кафедре.

### Библиографический список

1. Положение об учебно-методическом комплексе дисциплины в НЧОУ ВПО АЛСИ [Электронный ресурс] URL: <http://alsi.itech.ru/main/putev/norm-doc/UMK%20ALSI.pdf> (Дата посещения 15.06.2015).
2. Порядок разработки и утверждения в ГУАП образовательной программы высшего образования – программы бакалавриата, программы специалитета, программы магистратуры [Электронный ресурс] URL: [http://download.guap.ru/local/iuko/rds/razr\\_op\\_2014.doc](http://download.guap.ru/local/iuko/rds/razr_op_2014.doc) (Дата посещения 15.06.2015).

УДК 681.3.06

**Е. Н. Головин**

студент кафедры компьютерной математики и программирования

**М. Ю. Охтилев**

д-р техн. наук, проф. – научный руководитель

## ПРОТОКОЛ ИНФОРМАЦИОННОГО ВЗАИМОДЕЙСТВИЯ ПРИКЛАДНОГО УРОВНЯ И ЕГО ПРИМЕНЕНИЕ В АСУ КОСМИЧЕСКИМИ ОБЪЕКТАМИ

В современной промышленности автоматизированные системы управления (АСУ) используются повсеместно. Они позволяют осуществлять мониторинг состояния и управление в реальном времени сложными техническими объектами, такими как: искусственные спутники, АЭС, ракеты и многое другое.

Как правило, общий принцип работы АСУ заключается в сборе, анализе и визуализации информации об объектах управления, а также формировании управляющих сигналов на основе оценки их состояния. На рис. 1 схематически изображен процесс транспортировки информации, поступающей от объектов управления, начиная с пункта первичной обработки данных и заканчивая потребителями обработанной, подготовленной для визуализации или сбора статистики информации.

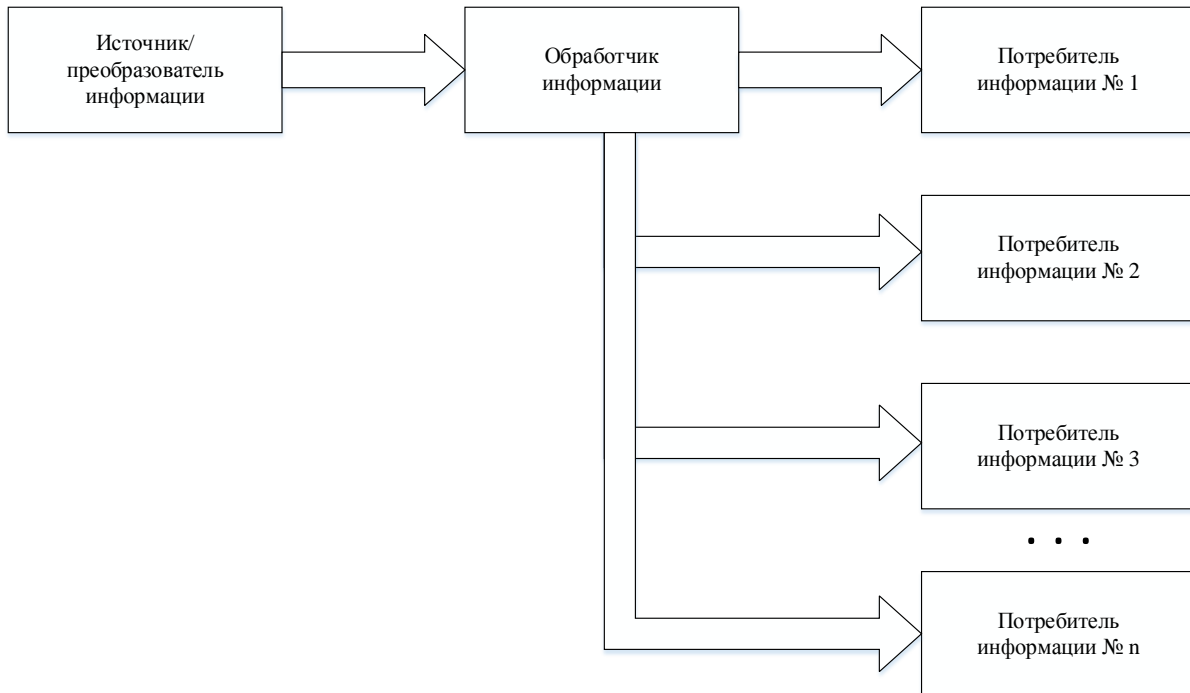


Рис. 1. Общая схема потоков информации в АСУ

Потребителями информации могут быть программы или программные модули, реализующие перечисленные выше функции. Они могут быть на одном вычислительном устройстве с блоком обработки информации, могут входить в состав одной программы вместе с упомянутым блоком или находиться на удаленном вычислительном устройстве и взаимодействовать с блоком обработки информации по сети. В данной работе рассматривается последний вариант.

Поскольку в рассматриваемом случае основной задачей является обеспечение информационного взаимодействия, возникает задача разработки протокола прикладного уровня (в соответствии с моделью ISO OSI) [1, 2]. В самом общем случае протокол информационного обмена представляет собой последовательность передачи сообщений с применением вычислительных, в том числе криптографических, алгоритмов в процессе обработки данных [3].

#### Описание итерационного процесса разработки протоколов

На сегодняшний день нет общепринятой технологии разработки безопасных прикладных протоколов информационного обмена [3]. Предлагается использовать следующий подход к организации процесса разработки протоколов (см. рис. 2).

Данный подход носит название «итерационный процесс разработки протоколов информационного обмена». Он был предложен и исследован на кафедре «Кибернетика» Национального исследовательского ядерного университета МИФИ [3].



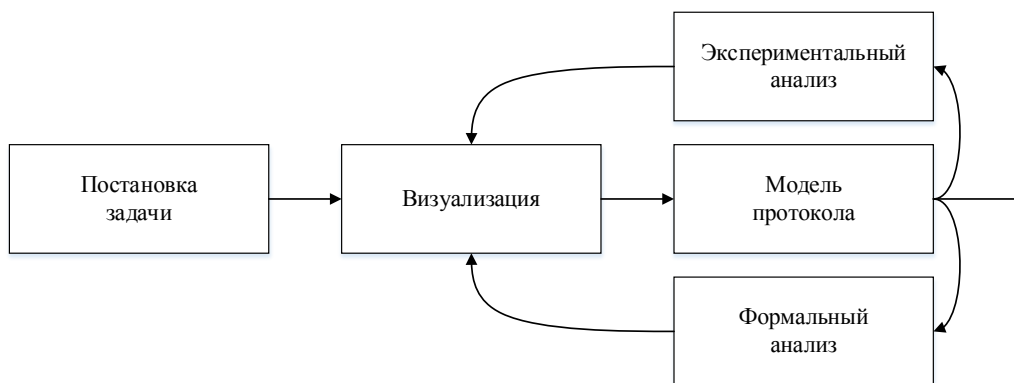


Рис. 2. Итерационный процесс разработки протоколов

Процесс начинается с постановки задачи. Под этим понимается определение требований к протоколу. Затем создается визуальная модель протокола. Она преобразуется в детализированное описание и формальную модель протокола. Исследование полученной модели осуществляется с применением формального и экспериментального методов. По результатам анализа могут быть внесены изменения в визуальную и формальную модели протокола. Процесс итерационно продолжается до тех пор, пока не будет получена модель протокола, отвечающая начальным требованиям.

В данной статье автор ограничился использованием упрощенной модели итерационного процесса разработки протоколов информационного обмена, в которой исключен цикл формального анализа. Таким образом, задача разработки сводится к:

- постановке задачи;
- визуализации общего принципа работы протокола;
- детальному описанию протокола;
- экспериментальному анализу полученного протокола.

В качестве постановки задачи приводится список требований, соблюдение которых должно обеспечиваться протоколом. В частности, протокол должен обеспечивать для обработчика информации возможность сообщать потребителю информации сведения о:

- процессе выполнения инициализации исполнительных модулей удаленного обработчика информации;
- нештатных ситуациях, критических ошибках на различных этапах работы удаленного обработчика информации;
- графическом интерфейсе, необходимом для визуализации последующих данных об объектах управления;
- управляющих сигналах, посылаемых управляемым объектам;
- непосредственно данных (данные о температуре, давлении, влажности, снимаемые с датчиков, расположенных на управляемом объекте);
- завершении сеанса информационного обмена.

Протокол также должен обеспечивать для потребителя информации возможность сообщать обработчику информации сведения о:

- начале сеанса информационного обмена;
- событиях, происходящих на стороне потребителя информации;
- успешном/с ошибками получении информации.

Результаты работы

Общий принцип работы протокола изображен на рис. 3.

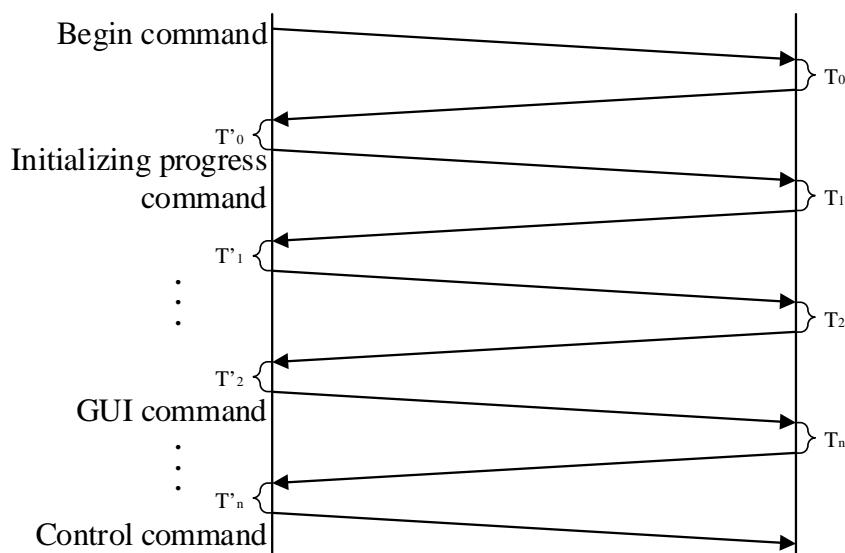


Рис. 3. Общий принцип работы протокола

Здесь  $T_0, T_1, T_2, \dots, T_n$  – время, затраченное обработчиком информации на обработку полученных данных, формирование и отправку нового пакета;  $T'_0, T'_1, T'_2, \dots, T'_n$  – время, затраченное потребителем информации на обработку полученных данных, формирование и отправку нового пакета; *Begin command* – команда начала сеанса информационного обмена; *Initializing progress command* – команда, сообщающая потребителю информации сведения о готовности обработчика информации, как правило, в процентном выражении; *GUI command* – команда, сообщающая потребителю информации сведения о способе представления получаемых впоследствии данных; *Control command* – команда, сообщающая потребителю информации сведения об управляющих сигналах, высылаемых на объекты управления.

Весь процесс информационного взаимодействия в данном случае можно разделить на четыре этапа.

1. Этап подключения к обработчику информации. На данном этапе потребитель информации осуществляет соединение с обработчиком по сети и высылает ему пакет со сведениями о начале сеанса информационного обмена (тип потребителя информации, наблюдаемый объект управления, режим обмена данными и т.д.). Уже на этом этапе обработчик информации может определить, в каком формате необходимо поставлять данные, о каких объектах управления и т.д.

2. Этап инициализации обработчика информации. На данном этапе обработчик сообщает потребителю сведения о загружаемых модулях, об их готовности. Также, в случае непредвиденных ситуаций обработчик высылает потребителю сведения об их возникновении. Отправив пакет, обработчик ждет от потребителя подтверждения об успешном получении им пакета. В случае ошибки (данные пришли не полностью, не соответствуют предусмотренной структуре) обработчик осуществляет повторную отправку пакета.

3. Этап построения графического интерфейса. На данном этапе обработчик информации сообщает потребителю сведения о способе представления данных, о графических компонентах, предназначенных для визуализации данных, свойствах этих компонентов и т. д. В зависимости от информации, поступившей на первом этапе, обработчик решает в каком формате сообщить эти сведения потребителю. Это может быть описание в формате HTML, XML, байтовая последовательность и т. д. Отправив данную информацию, обработчик ждет подтверждения об успешном получении сведений и построении графического интерфейса.

4. Этап визуализации получаемых об объектах управления данных. На данном этапе обработчик высылает потребителю блоки информации, подготовленной для визуализации, потреби-

тель же соответствующим образом их отображает. Как и на предыдущих двух этапах, успешное или неудачное получение информации квитируется отправкой пакета.

Общая структура пакета, отправляемого обработчиком информации потребителю, изображена на рис. 4.

DATA_BEGIN	eAnNetCmd	Длина данных (L)	Данные	DATA_END
(int)	(int)	(int)	L	(int)

Рис. 4. Общая структура пакета, отправляемого обработчиком информации потребителю

Общая структура пакета, отправляемого потребителем информации обработчику, изображена на рис. 5.

DATA_BEGIN	eAnNetCmd	Длина данных (L)	Код возврата	Данные (если есть)	DATA_END
(int)	(int)	(int)	(int)	L - 4	(int)

Рис. 5. Общая структура пакета, отправляемого потребителем информации обработчику

Здесь DATA\_BEGIN, DATA\_END – маркеры начала и конца пакета; eAnNetCmd – код команды; L – длина данных (в байтах).

Расположение данных в обоих случаях зависит от команды, содержащейся в пакете.

Перечень команд, фигурирующих в процессе информационного взаимодействия, приведен в табл. 1.

Таблица 1

Перечень команд

Команда	Описание
CmdStartAn	Запуск проекта, посылается клиентом при установлении соединения
CmdStopAn	Завершение проекта, посылается сервером при завершении проекта
CmdLoadWindow	Загрузка формы отображения
CmdDeleteObject	Удаление объекта
CmdSetValue	Установить значение свойства объекта
CmdGetValue	Получить значение свойства объекта
CmdGetPropertyType	Получить тип свойства
CmdGetInput	Данные по событию (Event), посылает клиент без запроса от сервера
CmdShowMessageBox	Вывести стандартное окно сообщения (вопроса)
CmdSetTableRowInfo	Устанавливает параметры нескольких ячеек в строке таблицы
CmdSetTableColumnInfo	Устанавливает параметры нескольких ячеек в столбце таблицы
CmdChooseProject	Открывает окно выбора проекта и возвращает индекс выбранного проекта
CmdSetActiveWindow	Сделать форму активной
CmdFloodBar	Открывает (изменяет) индикатор выполнения
CmdSetState	Устанавливает графическое состояние для объекта

Итак, в данной работе была рассмотрена задача разработки протокола информационного взаимодействия прикладного уровня и его применение в АСУ космическими объектами. В процессе разработки была рассмотрена общая структура пакетов и данных, содержащихся в этих пакетах. Кроме того, в соответствии с изменяющимися требованиями, были введены новые команды. Согласно данному протоколу, минимальный объем информативного пакета составляет 20 байт. В целом, итерационный подход к разработке позволяет добиться нужного результата, поступательно вводить во взаимодействие новые команды, устранять избыточность и т. д. Однако, для достижения большей гибкости, для решения вопросов безопасности этого может оказаться недостаточно. Поэтому следующим шагом в данном исследовании является, конечно же, рассмотрение методов формального синтеза и анализа протоколов, поисков его уязвимостей и т. д.

### **Библиографический список**

1. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. М.: Питер, 2010. – 943 с.
2. Танненбаум Э. Компьютерные сети/Э. Танненбаум; пер. с англ. В. Шрага. М.: Питер, 2010. – 993 с.
3. Михайлов А. С. Итерационный процесс разработки протоколов информационного обмена / А. С. Михайлов, С. С. Селезнев // Прикладная информатика: сб. науч. ст. Вып. 5(23), 2009. С. 59–60.

УДК 629.735.3, 004.921

***М. Е. Гончаров***

студент кафедры вычислительных систем и сетей

***Н. Н. Решетникова***

канд. техн. наук, доц. – научный руководитель

## **РАЗРАБОТКА ИНТЕРАКТИВНОГО ЭЛЕКТРОННОГО РУКОВОДСТВА ПО СБОРКЕ КВАДРОКОПТЕРА**

Квадрокоптер представляет собой платформу с четырьмя двигателями. По сравнению с беспилотными летательными аппаратами вертолетного типа с несущим и рулевым винтами, квадрокоптеры обладают рядом преимуществ, таких как: надежность и простота конструкции, большая стабильность, компактность и маневренность, малая взлетная масса при существенной массе полезной нагрузки. Во время полета квадрокоптер поддерживает горизонтальное положение относительно земли, может парить на месте, вращаться относительно вертикальной оси, перемещаться в любом направлении.

Область применения квадрокоптеров достаточно широка. Он может быть использован как недорогое и эффективное средство для получения фото и видео изображений с воздуха, в том числе в плохих погодных условиях. Благодаря тому, что квадрокоптер – дистанционно управляемый летательный аппарат, он хорошо подходит для наблюдения и контроля объектов и зон, доступ к которым затруднен (например, в случае естественных или техногенных катастроф) или в условиях непригодных для человека, таких как повышенный уровень радиации или сильное загрязнение воздуха.

Квадрокоптеры начали набирать популярность недавно, а потому, на данный момент, процесс их изучения налажен плохо. Огромную роль для восприятия имеет возможность взаимодействия с объектом или его 3D моделью в рамках его окружения, изменения его свойств или поведения. Но в большинстве случаев, во время обучения, студенты видят лишь несколько картинок изучаемых ими предметов, что негативно сказывается на усвоении материала. С помощью внед-

рения интерактивной компьютерной графики в процесс обучения, можно повысить не только интерес к изучаемому предмету, но и увеличить эффективность предоставления учебной информации.

На данный момент, изучение процесса сборки квадрокоптера происходит одним из трех способов:

- чтение текстовой документации или информации с сайта производителя;
- просмотр видеороликов, демонстрирующих процесс сборки;
- курсы по изучению процесса сборки квадрокоптера.

Каждый из этих способов имеет ряд своих недостатков. Так – в документациях зачастую встречается множество терминов, которые не знакомы человеку, собирающему свой первый квадрокоптер. Кроме того, не все могут понять изображенные в них схемы подключения, что может привести к неверной сборке аппарата и его поломке. Просмотр видеороликов куда более нагляден, но большинство из них записано на английском языке, что усложняет процесс обучения. Так же, чаще всего не удается разглядеть детали, вследствие плохого качества видео или слишком большой удаленности камеры от демонстрируемых элементов. Курсы по изучению процесса сборки квадрокоптера лишены данных минусов – вам наглядно все покажут, помогут собрать ваш собственный квадрокоптер, и объяснят, как лучше с ним обходиться. Проблема данных курсов состоит в их стоимости и наличии только в больших городах.

Описываемое в данной статье интерактивное приложение, демонстрирующее процесс сборки квадрокоптера, лишено перечисленных недостатков. В нем имеется возможность рассмотреть детали с любого ракурса и расстояния, проанализировать информацию о каждой из них, а так же, шаг за шагом, изучить процесс сборки квадрокоптера.

Рассмотрим процесс создания интерактивного 3D приложения по шагам.

Шаг 1. Создание трехмерной модели.

Квадрокоптер состоит из следующих элементов [1]:

- рама;
- контроллер;
- GPS модуль;
- светодиодный модуль;
- двигатели;
- винты;
- регуляторы оборотов;
- аккумулятор;
- ресивер;
- соединительные провода.

Для создания трехмерных моделей деталей квадрокоптера использовалась среда трехмерного моделирования Cinema 4D [2] и различные способы моделирования. К примеру, чтобы создать контроллер, в основном использовались булевы операции, для вычитания области пересечения двух объектов. Чтобы придать оболочке регулятора оборотов мягкий вид, использовался скульптинг, позволяющий с помощью кистей придавать 3D объекту желаемую форму. Для создания рамы использовались нарисованные в Adobe Illustrator контуры, выдавливая которые инструментом Extrude, можно легко получить конечную модель.

В табл. 1 приведен полигонаж каждой из разработанных 3D моделей, а так же количество объектов, из которых они состоят.

Полигонаж моделей и количество объектов, из которых они состоят

Название	Количество полигонов	Количество объектов
Рама	379790	4210
Контроллер	118545	2570
GPS модуль	43133	540
Светодиодный модуль	106961	1027
Мотор	27640	2740
Регулятор оборотов	40394	138
Винт	10918	1
Аккумулятор	51238	284
Ресивер	30752	199

На рис. 1 приведена трехмерная модель квадрокоптера.



Рис. 1. Трехмерная модель квадрокоптера

Шаг 2. Текстурирование.

Для создания текстур использовались редакторы растровой и векторной графики – Adobe Photoshop и Adobe Illustrator. Adobe Photoshop использовался для цветокоррекции текстур, а так же получения карт нормалей. С помощью Adobe Illustrator создавались карты прозрачности, позволяющие нанести надписи поверх основной текстуры.

На рис. 2 приведена текстурированная трехмерная модель квадрокоптера.



Рис. 2. Текстурированная трехмерная модель квадрокоптера

### Шаг 3. Анимирование.

Анимация выполнялась двумя способами – анимация по ключам и костная анимация. При анимации по ключам, аниматор задает значение некоторых характеристик сцены в определенных кадрах, называемых ключевыми. Значение характеристик в остальных кадрах рассчитывается автоматически. Костная анимация – позволяет создавать кости скелета и анимировать объекты с их помощью.

В качестве примера рассмотрим первый шаг сборки квадрокоптера – он демонстрирует процесс припаивания проводов питания регулятора оборотов к нижней центральной панели рамы квадрокоптера. Для изменения позиции всего объекта используется анимация по ключам, а для изгиба проводов – костная анимация.

### Шаг 4. Создание трехмерного приложения.

Для разработки элементов управления процессом сборки 3D модели квадрокоптера используется игровой движок Unity 3D [3]. Интерактивное приложение состоит из двух 3D сцен – одна позволяет рассмотреть все детали, из которых состоит квадрокоптер, вторая – изучить процесс его сборки. Иерархические структуры 3D сцен приведены на рис. 3.

Как можно заметить из рис. 3, принципы их построения аналогичны. Одна из камер, отображает фон, вторая камера направлена на элементы пользовательского интерфейса, третья – включается при нажатии кнопки перехода на следующую сцену и отображает процесс загрузки. Еще по одной камере хранится в каждом игровом объекте, содержащем трехмерные модели деталей. К каждой из этих камер подключен скрипт обзора с помощью мыши. Кроме трехмерной модели и камеры в этих игровых объектах хранятся по 6 источников света – по одному с каждой стороны детали, для равномерного ее освещения. Так же существует игровой объект «О программе», и как понятно из названия, он хранит информацию об управлении интерактивным приложением. В сцене обзора деталей квадрокоптера присутствует дополнительный ряд объектов – «Информация». В таких объектах хранится информация о технических характеристиках детали, отображающаяся после нажатия одноименной кнопки.

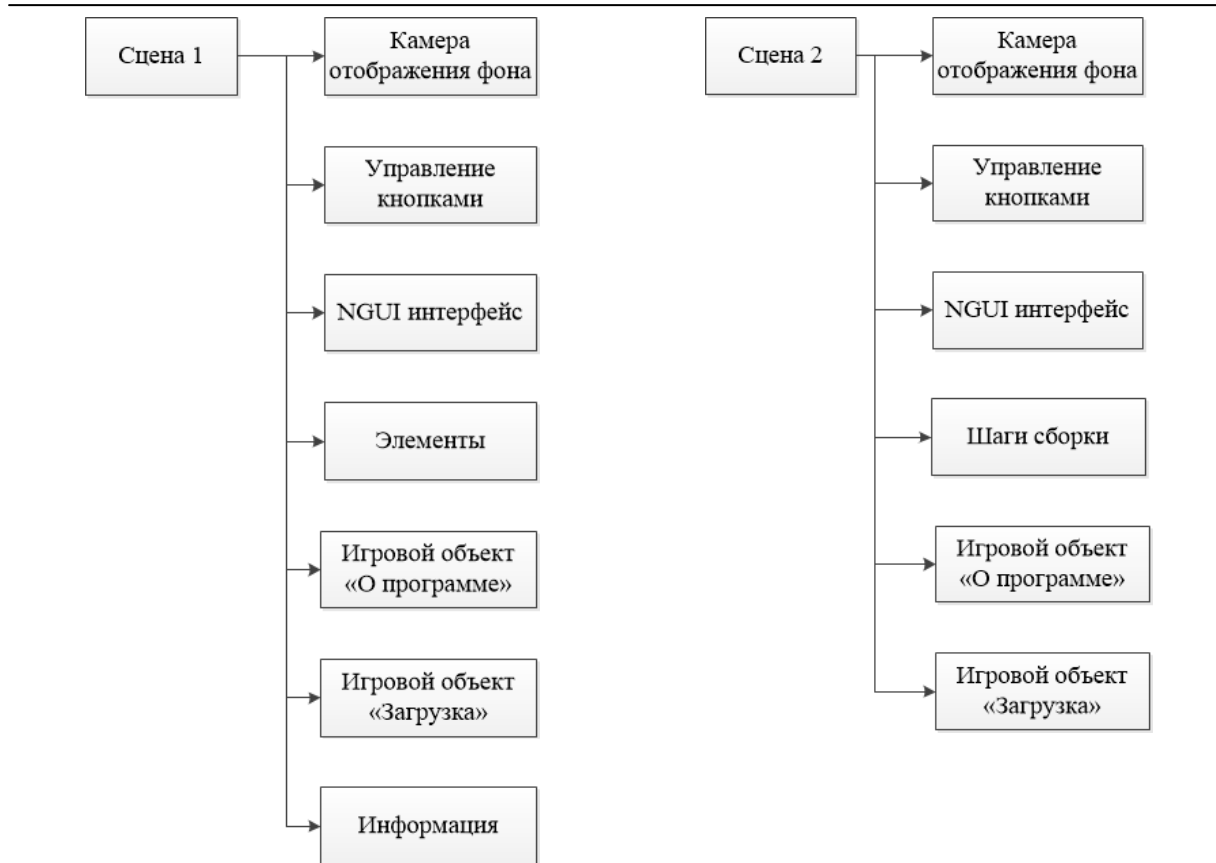


Рис. 3. Иерархические структуры 3D сцен приложения

Шаг 5. Написание скриптов.

Для управления приложением были написаны три скрипта (табл. 2) с использованием языков программирования C#[4] и JavaScript [5]. Первый из этих скриптов позволяет с помощью мыши вращать камеру вокруг детали. Второй и третий – отвечают за функционал кнопок в отдельных сценах. Алгоритмы скриптов разработаны так, что они не зависят от моделей и их количества. Таким образом, при замене 3D моделей, функциональность приложения не изменится.

Таблица 2

Функционалы скриптов

Управление камерой	Управление сценой обзора деталей квадрокоптера	Управление сценой изучения процесса сборки квадрокоптера
<ul style="list-style-type: none"> <li>– Вращение камеры с помощью мыши</li> <li>– Приближение камеры с помощью колесика мыши</li> </ul>	<ul style="list-style-type: none"> <li>– Выбор детали</li> <li>– Вращение детали</li> <li>– Вывод информации о детали</li> <li>– Вывод информации об управлении</li> <li>– Отображение процесса загрузки</li> <li>– Переход на следующую сцену</li> <li>– Выход из программы</li> </ul>	<ul style="list-style-type: none"> <li>– Изменение шага сборки</li> <li>– Изменение скорости воспроизведения анимации</li> <li>– Запуск и рестарт анимации</li> <li>– Отображение процесса загрузки</li> <li>– Переход на следующую сцену</li> <li>– Выход из программы</li> </ul>



Шаг 6. Отладка программы.

Перед завершением работы над программным продуктом, необходимо протестировать его и внести правки при возникновении каких-либо недочетов.

На рис. 4 и рис. 5 приведены скриншоты обеих сцен готового интерактивного приложения.



Рис. 4. Сцена обзора деталей квадрокоптера



Рис. 5. Сцена изучения процесса сборки квадрокоптера

### Библиографический список

1. Flame Wheel ARF KIT – Multirotor Flying Platform For Entertaining And Amateur Ap. [Электронный ресурс]. – Режим доступа: <http://www.dji.com/product/flame-wheel-arf>, свободный
2. MAXON | 3D FOR THE REAL WORLD: Документация [Электронный ресурс]. – Режим доступа: <http://www.maxon.net/ru/support>, свободный
3. Unity – Scripting API [Электронный ресурс]. – Режим доступа: <http://docs.unity3d.com/ScriptReference/>, свободный
4. Либерти Дж. Программирование на C# / Дж. Либерти; Пер. с англ. – СПб.: Символ, 2003. – 688 с.
5. Флэнаган Д. JavaScript Подробное руководство / Д. Флэнаган; Пер. с англ. – СПб.: Символ, 2008. – 982 с.

УДК 681.3.06

**Н. С. Долгов**

студент кафедры компьютерной математики и программирования

**М. Ю. Охтилев**

д-р техн. наук, проф. – научный руководитель

### ПРОГРАММНЫЙ КОМПЛЕКС ДИАГНОСТИРОВАНИЯ ЭЛЕМЕНТОВ ОБОРУДОВАНИЯ СИСТЕМЫ БЕЗОПАСНОСТИ И КОНТРОЛЯ

Реалии современного мира диктуют жесткие условия для выживания промышленных предприятий. Увеличивается конкуренция на рынках сбыта. Чтобы оставаться конкурентоспособными, производители вынуждены ужесточать требования к качеству выпускаемой продукции. Для достижения желаемого результата, требуется, чтобы производственное оборудование работало без каких-либо погрешностей и сбоев, именно для этих целей и используются автоматизированные системы диагностики и контроля технического состояния оборудования. Проведение диагностики позволяет решить множество насущных проблем:

- обеспечить бесперебойное и надежное функционирование предприятия;
- планировать профилактические ремонты, во избежание возможных поломок;
- обеспечить экономию средств на дорогостоящие ремонты при неисправности оборудования;
- повысить производительность труда за счет бесперебойной работы оборудования;
- получать стабильно высокое качество выпускаемой продукции.

В рамках решения задачи диагностирования технического состояния элементов оборудования системы безопасности и контроля автоматизированной системы управления подготовкой и пуском на унифицированном техническом комплексе, требовалось разработать программный комплекс, позволяющий проводить диагностирование следующих блоков:

- диагностика сети (контроль подключения аппаратуры по сети);
- диагностика дискретных сигналов, поступающих от стойки;
- диагностика состояния источника бесперебойного питания;
- диагностика состояния переключателя фаз;
- диагностика питания (контроль наличия питания 220 В на фидерах питания);
- диагностика состояния видеокамер.

Задачи такого типа все больше и больше набирают свою актуальность в различных отраслях экономики. Растет количество оборудования, которое становится все сложнее. Диагностика такого оборудования, с применением человеческого труда становится недопустимой, т. к. это требует больших временных затрат, и к тому же знаний и классификации обычных специалистов становится недостаточно. Автоматизация процессов мониторинга и, в частности, диагностирования позволит сократить количество аппаратно-программных средств и, соответственно, численность персонала, унифицировать аппаратные и программные средства и сократить затраты на эксплуатацию автоматизированной системы управления.

На современном отечественном рынке присутствуют различные программные продукты для автоматизации процессов диагностики, которые обладают собственными особенностями и преимуществами, например:

ITV/AxxonSoft – эта компания недавно выпустила модуль для диагностики состояния оборудования, который предназначен для применения в качестве составной части специального программного комплекса под названием «Интеллект», позволяющий в автоматическом режиме осуществлять контроль за работоспособностью всех частей системы и оперативно выявлять любые типы отклонений. Пример работы ITV/AxxonSoft приведен на рис. 1.

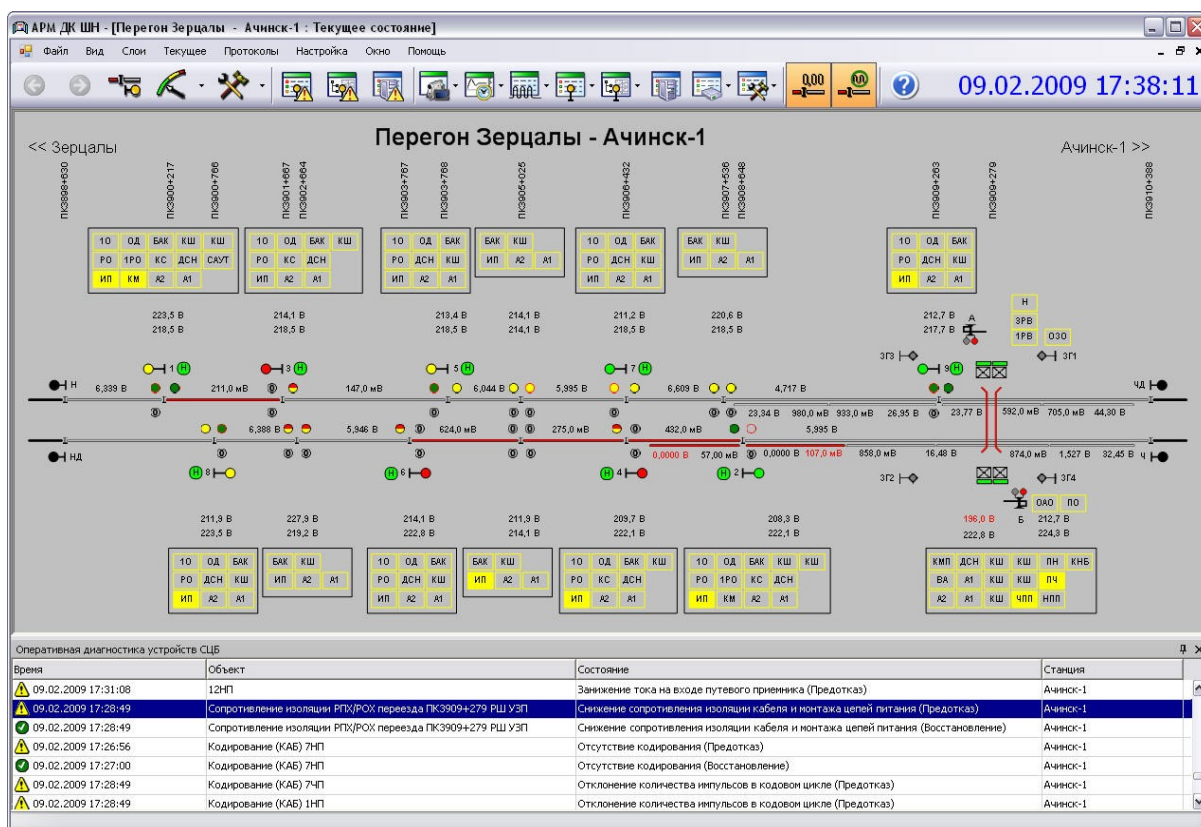


Рис. 1. Пример работы программы «ITV/AxxonSoft»

НПЦ «Динамика» – внедрение этой системы диагностики дает возможность следить за состоянием всего оборудования в режиме реального времени и своевременно устранять возникающие неполадки, что, в свою очередь, позволяет свести к минимуму все эксплуатационные потери. Использование этой мониторинговой системы позволяет практически полностью устранить вероятность внеплановой остановки оборудования. Эта мониторинговая система рекомендована для использования на нефтехимических предприятиях. Пример работы НПЦ «Динамика» приведен на рис. 2.

«АПИД» – это автоматизированная система подготовки исходных данных, которая предназначена для занесения измерительных данных об измерительной информации произвольной природы, алгоритмах, моделях и т. п. В базу данных, просмотра и корректировки данных, содержащихся в БД и описывающих объект анализа, в качестве которого могут рассматриваться как отдельные системы и агрегаты в целом. Пример работы «АПИД» приведен на рис. 3.

# ТЕХНИЧЕСКИЕ НАУКИ

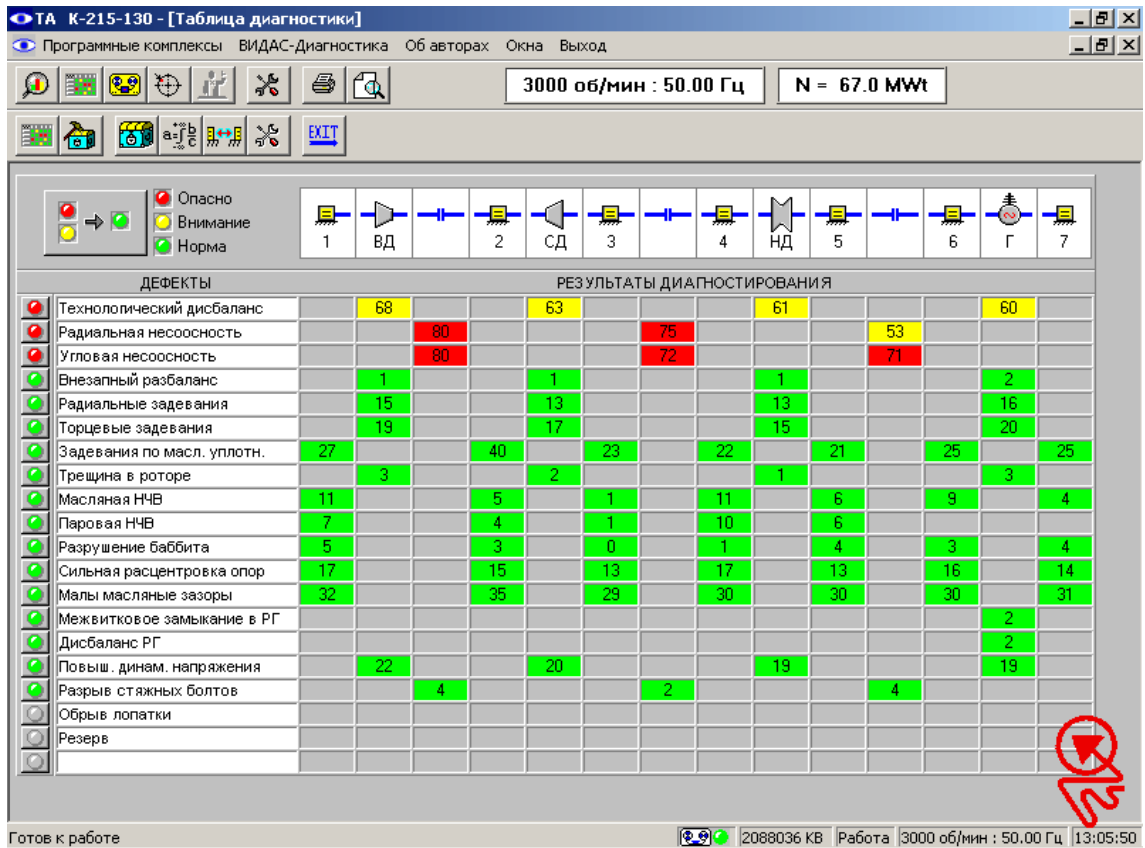


Рис. 2. Пример работы программы «Динамика»

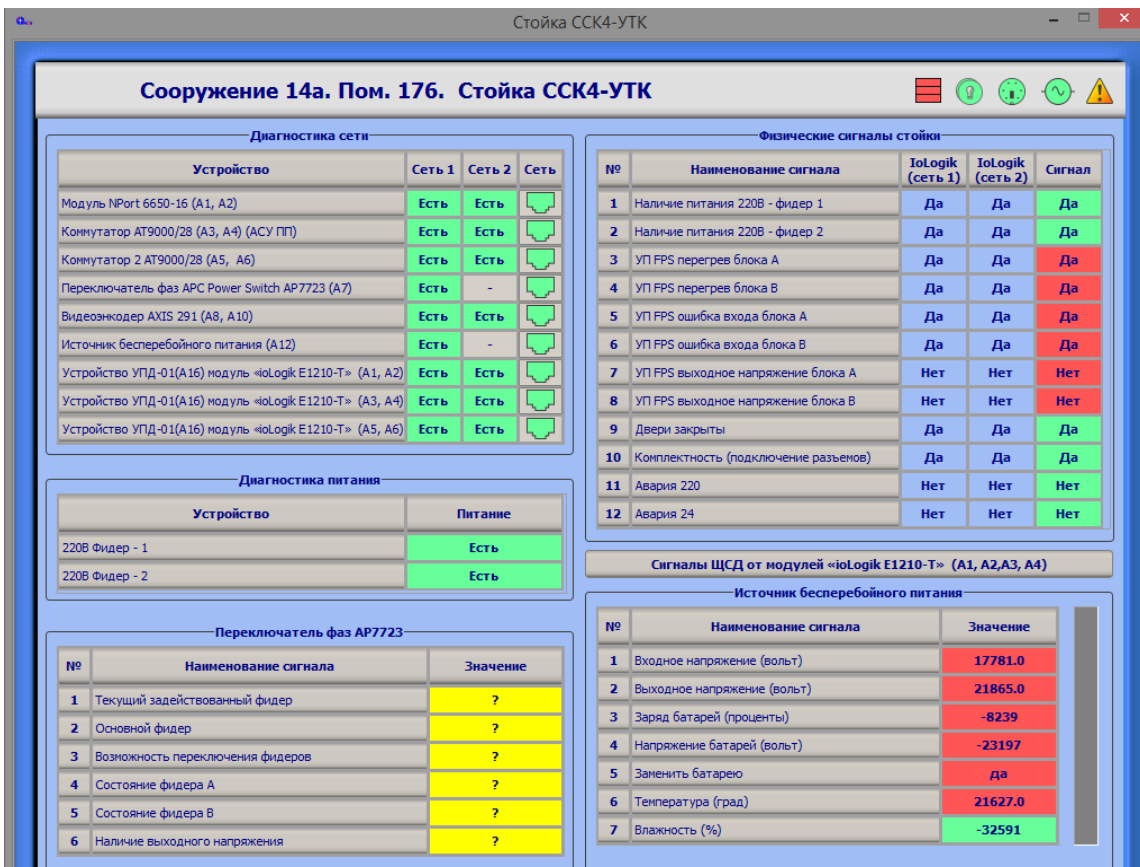


Рис. 3. Пример работы программы «АПИД»

Процесс получения оценки о техническом состоянии объекта диагностики схематично представлен на рис. 4. Как видно из рисунка, в каждый момент времени объект диагностики находится в одном из состояний из потенциально бесконечного множества состояний. В процессе наблюдения за объектом диагностики на нем производятся измерения, результаты которых в виде измерительной информации передаются в систему оценивания состояния. В результате проведения необходимых операций по оцениванию технического состояния определяются значения вычисляемых параметров технического состояния (ТС). В целом измеряемые и вычисляемые параметры ТС, значения которых используются при оценивании ТС, образуют некоторое множество параметров ТС. Целью мониторинга и, в частности, диагностирования является получение на основе анализа измерительной информации, обобщенных оценок совокупности параметров технического состояния, значения которых в явном виде либо указывают степень работоспособности рассматриваемого объекта управления или место и вид возникшей неисправности, либо являются оценками прогнозируемых явлений и процессов с заданной точностью и интервалом прогноза и т. п. – с учетом конкретных целей и условий эксплуатации объекта диагностики на различных этапах его функционирования [1–3].

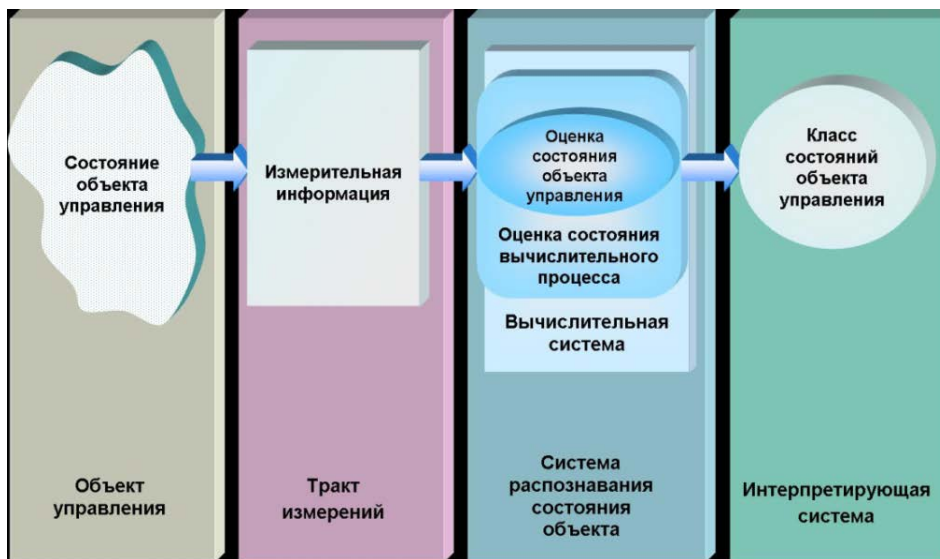


Рис. 4. Схема оценивания технического состояния объекта анализа

В качестве объекта диагностики используется система безопасности и контроля автоматизированной системы управления подготовкой и пуском на унифицированном техническом комплексе, которая включает в себя следующее оборудование: сервер системы безопасности и контроля, стойки, щиты, пульта автоматизированных рабочих мест.

Программный комплекс диагностики позволяет в режиме реального времени отслеживать состояние контролируемых блоков и модулей объекта. В качестве примера рассмотрим «Сервер системы безопасности и контроля», структурная схема которого изображена на рис. 5.

Разработка программного комплекса (ПК) диагностики технического состояния элементов оборудования системы безопасности и контроля проводилась с помощью автоматизированной системы подготовки исходных данных. Создание ПК включает следующие этапы:

- проектирование структур и типов данных. На данном этапе описываются параметры, переменные, таблицы текстов и другие структуры данных, используемые при мониторинге;
- визуальное проектирование форм. Создаются мнемосхемы, диалоговые окна с которыми пользователь будет работать на этапе анализа;
- логическое проектирование программы диагностики.

На рис. 6 представлен главный экран программного комплекса, где отображены все объекты диагностики, а также степень их работоспособности.

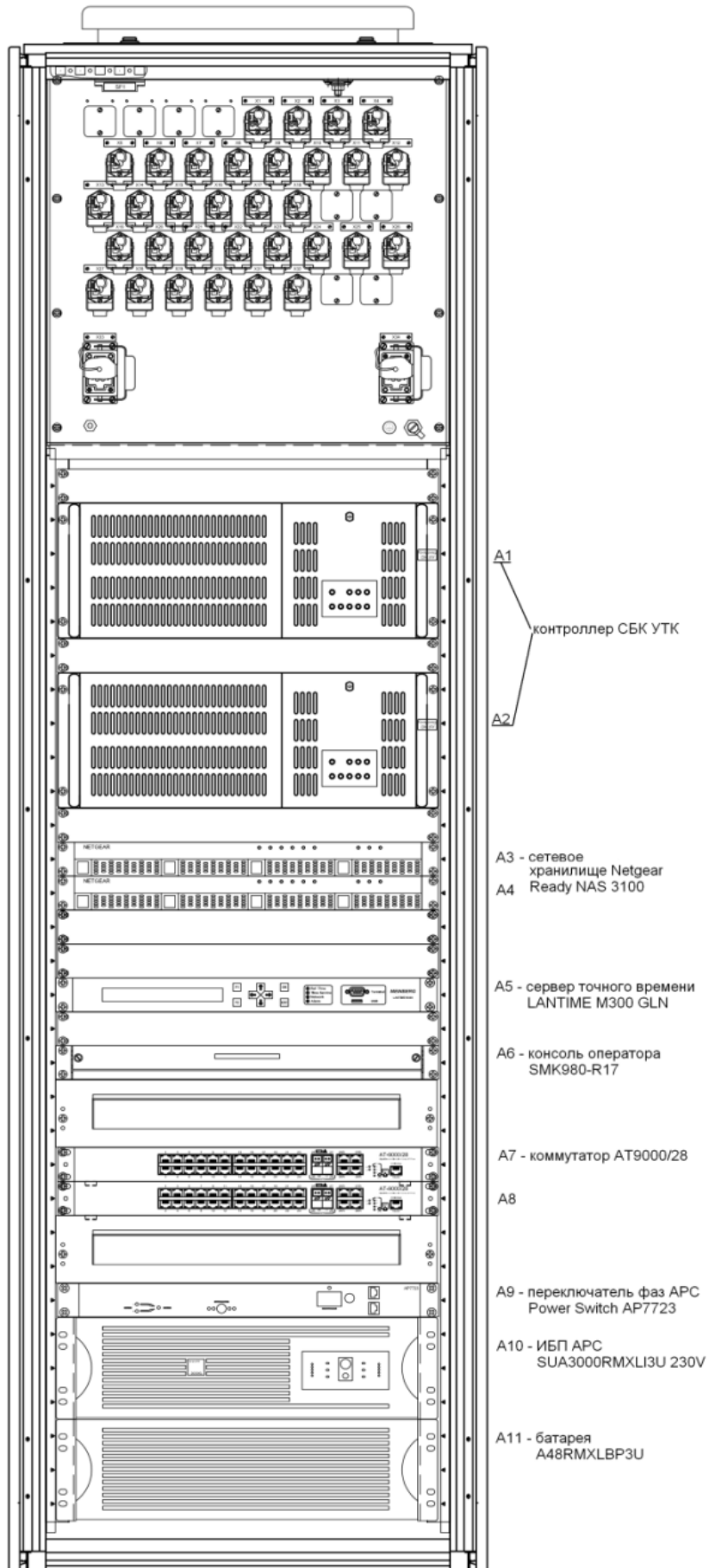


Рис. 5. Стойка «Сервер системы безопасности и контроля»

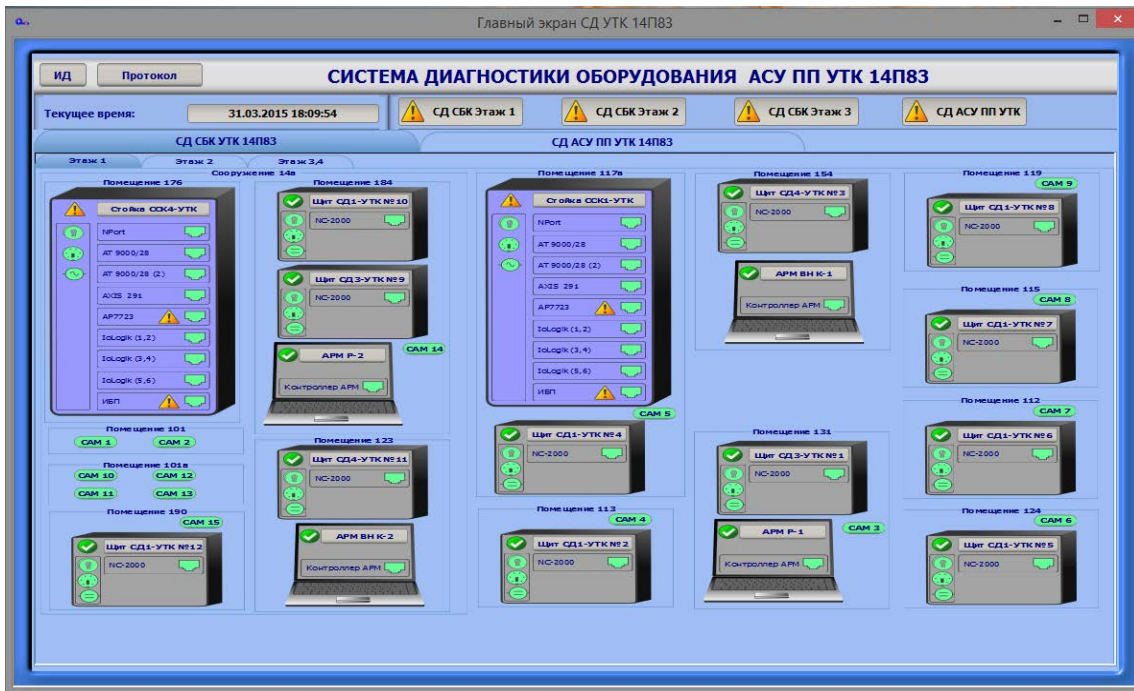


Рис. 6. Главный экран программного комплекса диагностики элементов оборудования системы безопасности и контроля

Диагностика «Сервера системы безопасности и контроля», позволяет определить работоспособность его элементов, таких как: коммутатор, переключатель фаз, консоль оператора и т. д. (рис. 7).

Информация о наступивших событиях заносится в протокол (рис. 8).

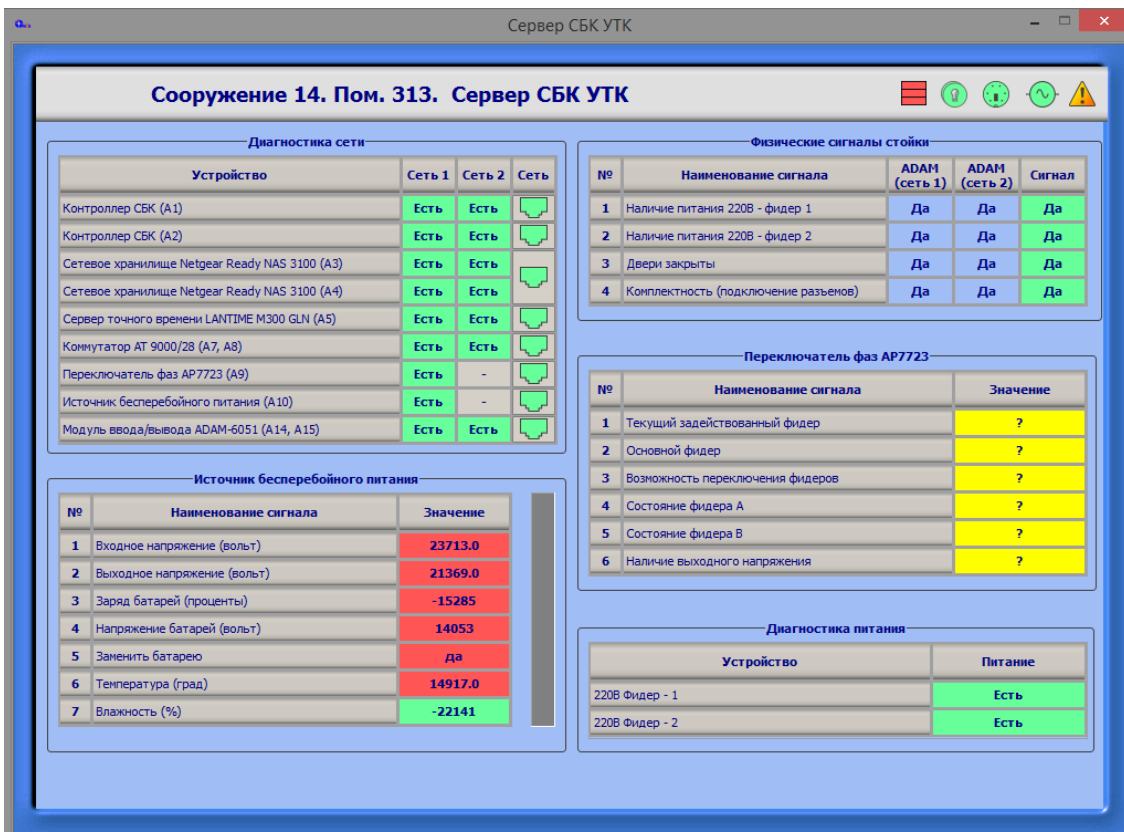


Рис. 7. Диагностика технического состояния элементов «Сервера безопасности и контроля»

Время	Событие
2000/01/01 00:00:29.000	Сооружение 14. Пом. 313. Стойка ССК3-УТК № 0003. ИБП SMX3000HV4U (A12). Заряд батарей - норма
2000/01/01 00:00:28.000	Сооружение 14. Пом. 117а. Стойка ССК1-УТК № 0001. ИБП SMX3000HV4U (A12). Заряд батарей - норма
2000/01/01 00:00:28.000	Сооружение 14. Пом. 207. Стойка ССК2-УТК № 0002. ИБП SMX3000HV4U (A12). Заряд батарей - норма
2000/01/01 00:00:28.000	Сооружение 14. Пом. 313. Сервер СБК УТК № 0001. ИБП SUA3000RMXL13U (A10). Заряд батарей - норма
2000/01/01 00:00:27.000	Сооружение 14. Пом. 207. Стойка ССК2-УТК № 0002. ИБП SMX3000HV4U (A12). Низкий уровень заряда батарей
2000/01/01 00:00:27.000	Сооружение 14а. Пом. 176. Стойка ССК4-УТК № 0004. ИБП SMX3000HV4U (A12). Низкий уровень заряда батарей
2000/01/01 00:00:27.000	Сооружение 14. Пом. 313. Сервер СБК УТК № 0001. ИБП SUA3000RMXL13U (A10). Низкий уровень заряда батарей
2000/01/01 00:00:27.000	Сооружение 14. Пом. 313. Сервер БД АСУ ПП УТК № 0001. ИБП SUA3000RMXL13U (A9). Заряд батарей - норма
2000/01/01 00:00:26.000	Сооружение 14а. Пом. 176. Стойка ССК4-УТК № 0004. ИБП SMX3000HV4U (A12). Заряд батарей - норма
2000/01/01 00:00:25.000	Сооружение 14. Пом. 207. Стойка ССК2-УТК № 0002. ИБП SMX3000HV4U (A12). Заряд батарей - норма
2000/01/01 00:00:25.000	Сооружение 14. Пом. 313. Стойка ССК3-УТК № 0003. ИБП SMX3000HV4U (A12). Низкий уровень заряда батарей
2000/01/01 00:00:25.000	Сооружение 14а. Пом. 176. Стойка ССК4-УТК № 0004. ИБП SMX3000HV4U (A12). Низкий уровень заряда батарей
2000/01/01 00:00:25.000	Сооружение 14. Пом. 313. Сервер СБК УТК № 0001. ИБП SUA3000RMXL13U (A10). Заряд батарей - норма
2000/01/01 00:00:25.000	Сооружение 14. Пом. 313. Сервер БД АСУ ПП УТК № 0001. ИБП SUA3000RMXL13U (A9). Низкий уровень заряда батарей
2000/01/01 00:00:24.000	Сооружение 14. Пом. 117а. Стойка ССК1-УТК № 0001. ИБП SMX3000HV4U (A12). Низкий уровень заряда батарей
2000/01/01 00:00:24.000	Сооружение 14. Пом. 313. Сервер БД АСУ ПП УТК № 0001. ИБП SUA3000RMXL13U (A9). Заряд батарей - норма
2000/01/01 00:00:23.000	Сооружение 14. Пом. 207. Стойка ССК2-УТК № 0002. ИБП SMX3000HV4U (A12). Низкий уровень заряда батарей
2000/01/01 00:00:23.000	Сооружение 14. Пом. 313. Сервер БД АСУ ПП УТК № 0001. ИБП SUA3000RMXL13U (A9). Низкий уровень заряда батарей
2000/01/01 00:00:22.000	Сооружение 14. Пом. 313. Стойка ССК3-УТК № 0003. ИБП SMX3000HV4U (A12). Заряд батарей - норма
2000/01/01 00:00:21.000	Сооружение 14. Пом. 313. Стойка ССК3-УТК № 0003. ИБП SMX3000HV4U (A12). Низкий уровень заряда батарей
2000/01/01 00:00:21.000	Сооружение 14. Пом. 313. Сервер СБК УТК № 0001. ИБП SUA3000RMXL13U (A10). Низкий уровень заряда батарей
2000/01/01 00:00:20.000	Сооружение 14а. Пом. 176. Стойка ССК4-УТК № 0004. ИБП SMX3000HV4U (A12). Заряд батарей - норма
2000/01/01 00:00:20.000	Сооружение 14. Пом. 313. Сервер БД АСУ ПП УТК № 0001. ИБП SUA3000RMXL13U (A9). Заряд батарей - норма
2000/01/01 00:00:19.000	Сооружение 14. Пом. 117а. Стойка ССК1-УТК № 0001. ИБП SMX3000HV4U (A12). Заряд батарей - норма
2000/01/01 00:00:19.000	Сооружение 14. Пом. 207. Стойка ССК2-УТК № 0002. ИБП SMX3000HV4U (A12). Заряд батарей - норма

Рис. 8. Протокол наступивших событий

Разработка автоматизированной системы диагностики технического состояния объектов достаточно трудоемкий процесс. С усложнением объектов диагностики требуется существенное увеличение числа контролируемых параметров, характеризующих процессы их функционирования. Сбор, обработка, анализ и отображение информации в удобном для пользователя виде позволяет применять данные системы повсеместно, где присутствует оборудование, и поэтому разработка таких систем является одним из ведущих направлений автоматизации управления современным производством.

### Библиографический список

1. Охтилев М. Ю. Использование топологического пространства вычислений при оценивании технических состояний объектов управления // Изв. вузов. Приборостроение. 2000. Т. 43. № 7. С. 27–30.
2. Охтилев М. Ю., Соколов Б. В., Юсупов Р. М. Интеллектуальные технологии мониторинга и управления структурной динамикой сложных технических объектов. М.: Наука, 2006. 410 с.
3. Майданович О. В., Соколов Б. В., Охтилев М. Ю. Новый подход к созданию интеллектуальных информационных технологий проектирования систем мониторинга состояния сложных объектов // XI международная научно-техническая конференция 12–14 мая 2010 г. «Кибернетика и высокие технологии XXI века»: Сборник докладов в 2-х томах. Воронеж: НПФ «Саквое», 2010. Т. 2. С. 601–608.

УДК 004.514, 04.921

**И. А. Дубков, Г. К. Логач**

магистранты кафедры вычислительных систем и сетей

**Н. Н. Решетникова**

канд. техн. наук, доц. – научный руководитель

### РАЗРАБОТКА ТРЕХМЕРНОЙ МОДЕЛИ ПЕРСОНАЖА ДЛЯ ИГРОВОГО ДВИЖКА

На сегодняшний день разработка трехмерных моделей персонажей занимает одно из главных мест в создании современных фильмов, мультфильмов, игр, презентаций. Развитие



технологий анимации трехмерных персонажей открывает возможности не только для моделирования фантастических героев компьютерных игр или ситуаций современной жизни (симуляторы, тренажеры и т.п.), но и картин далекого прошлого при создании исторических виртуальных реконструкций [1].

Сложившийся процесс создания анимированного персонажа в пакетах 3D-моделирования можно условно разделить на следующие основные этапы: проектирование и разработка модели персонажа; проектирование и описание атомарных движений персонажа; генерация сложных движений на основе последовательности атомарных движений; сохранение информации о движении и последующее воспроизведение этой информации.

Большинство анимируемых персонажей состоят из оболочки и скелета. При движении отдельных частей скелета оболочка будет деформироваться, поэтому необходимо привязать кости к соответствующим частям тела таким образом, чтобы при изменении положения скелета оболочка деформировалась реалистично. Создание скелета персонажа, связывание его с оболочкой и дальнейшая анимация с использованием метода ключевых кадров достаточно длительный и трудоемкий процесс.

Технологический процесс разработки реалистичной трехмерной модели персонажа рассмотрим на примере двух трехмерных моделей: модели скифа для приложения «Интерактивная компьютерная реконструкция Костромского кургана» и модели волшебника для интерактивной 3D-игры. Технология моделирования включает следующие этапы:

- поиск референса и создание заготовки;
- создание высоко полигональной модели персонажа;
- ретопология модели;
- текстурирование:
  - создание развертки UV низко полигональной модели;
  - запекание карт нормалей;
  - создание Diffuse и Specular карт;
  - создание анимаций;
- подготовка анимации:
  - создание скелета;
  - связывание оболочки и скелета;
  - анимирование скелета;
- внедрение в игровой движок.

На первом этапе производится поиск двумерных изображений – референса, после чего на основе этих изображений создается заготовка. При работе с заготовкой в любом удобном пакете трехмерного моделирования создается простая трехмерная модель в T-позе, силуэт которой соответствует задуманной идее.

Затем на основе созданной заготовки и ее частей, строится высоко полигональная модель в таких приложениях как ZBrush [2] или Autodesk Mudbox [3]. Процесс разработки высоко полигональной модели значительно отличается от работы в пакете 3D моделирования. Он больше напоминает лепку скульптуры с использованием всевозможных кистей, каждая из которых обладает особенным поведением при взаимодействии с моделью персонажа. В ходе процесса детализации у модели увеличивается количество полигонов, что позволяет наносить более мелкие детали. Рассмотрим технологию разработки создания трехмерного анимированного персонажа более подробно.

#### Поиск референса и создание заготовки

Референс – это подробное графическое описание персонажа. В более широком смысле референс – вспомогательное изображение: рисунок или фотография, которые художник изучает перед работой, чтобы точнее передать детали. На рис. 1 приведен пример создания высоко полигональной модели персонажа в редакторе Pixologic ZBrush. Для трехмерной модели скифа

были представлены фотографии сосудов периода скифской архаики, на которых было изображено обмундирование воинов. По референсу создаются общие формы модели. Ключевым этапом моделирования персонажа является создание его каркаса. Разработка каркаса ведется в пакете трехмерного моделирования Zbrush.



Рис. 1. Моделирование по исходным данным

#### Создание высоко полигональной модели персонажа

С использованием кисти виртуальной лепки создается высокополигональная модель, состоящая из более миллиона полигонов. Zbrush имеет огромное количество кистей для работы с полигонами. Так как программа позиционируется как программа виртуальной лепки, она имеет ряд кистей для вдавливания и выдавливания полигонов. В данных работах для этой цели используется кисть *Standard*. Кисть *Standardadd* выдавливает полигоны в зависимости от интенсивности и размера кисти. Настраивать параметры кисти можно с помощью правой кнопки мыши, при нажатии на которую появляется окно параметров кисти. Для вдавливания полигонов используется кисть *Standarddiv*. При разработке высоко полигональной модели использовались также следующие кисти:

- кисть *Move* используется для перемещения полигонов, не отрывая их от соседних;
- кисть *Smooth* необходима для разглаживания полигонов на поверхности объекта;
- кисть *Clay* позволяет имитировать глину;
- кисть *InFlat* имитирует надувание модели.

Путем добавления кисти имитации глины, разглаживания отдельных частей полигонов, и вдавливания/выдавливания получается модель волшебника (рис. 2).

#### Ретопология модели

Работать с высокополигональной моделью в игровом движке невозможно. Чтобы избежать этих трудностей, необходимо произвести ретопологию модели.

Ретопология – это изменение топологической сетки объекта, с сохранением, по возможности, его формы.

Другими словами, ретопология – это изменение полигональной сетки, создание новой геометрии для дальнейшего его использования.

Ретопология используется для:

- создания «чистой» сетки для дальнейшей анимации;
- создания низкополигональной модели;
- создания UV карты.



Рис. 2. Процесс моделирования в Zbrush

Для того чтобы произвести ретопологию модели, используется программа ТороGun.

ТороGun – программа ретопологии, предназначенная для построения низко полигональной модели по поверхности высоко полигональной, отсканированной с помощью 3D сканера или созданной в среде Zbrush, Mudbox. В результате работы была создана топология путем добавления вершин и полигонов на поверхности высоко полигональной сетки.

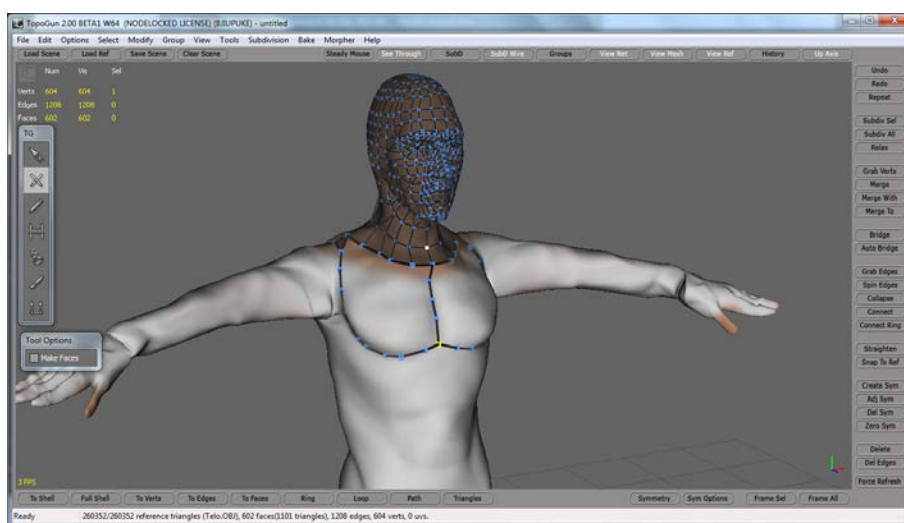


Рис. 3. Процесс ретопологии

### Текстурирование

Текстурирование – это процесс наложения текстурных карт на трехмерную модель. Для того чтобы текстуры легли ровно, необходимо произвести развертку трехмерной модели.

#### *Создание развертки UV низко полигональной модели*

UV-развертка – наиболее гибкий способ отображения 2D-текстур на трехмерном объекте. Данный процесс располагает трехмерную модель на плоскости, которая называется UV-мар. Использование UV-развертки обеспечивает реализм моделям и очень высокую детализацию, которых практически невозможно достичь с помощью процедурных текстур.

Благодаря UV-развертке на модель накладываются не только текстуры, но и карты нормалей с картами цветов.

#### *Запекание карт нормалей*

Карты нормалей – это один из видов рельефного текстурирования, то есть метод компьютерной графики для придания более реалистичного и насыщенного вида поверхности объектов.

Normalmapping – техника, позволяющая изменять нормаль отображаемого пикселя основываясь на цветной карте нормалей, в которой эти отклонения хранятся в виде текселя, цветовые составляющие которого [r, g, b] интерпретируются в оси вектора [x, y, z], на основе которого вычисляется нормаль, используемая для расчета освещенности пикселя.

#### *Создание Diffuse и Specular карт*

Diffuse Map – карта представляющая собой диффузное отражение и цвет поверхности объекта. Простыми словами – это базовый цвет объекта. Эта карта используется в качестве основной текстуры объекта.

Specular Map – карта передает интенсивность и цвет бликов на поверхности объекта. Другими словами, она определяет блик и цвет световых отражений.

Обе карты запекаются в программе 3Ds Max.

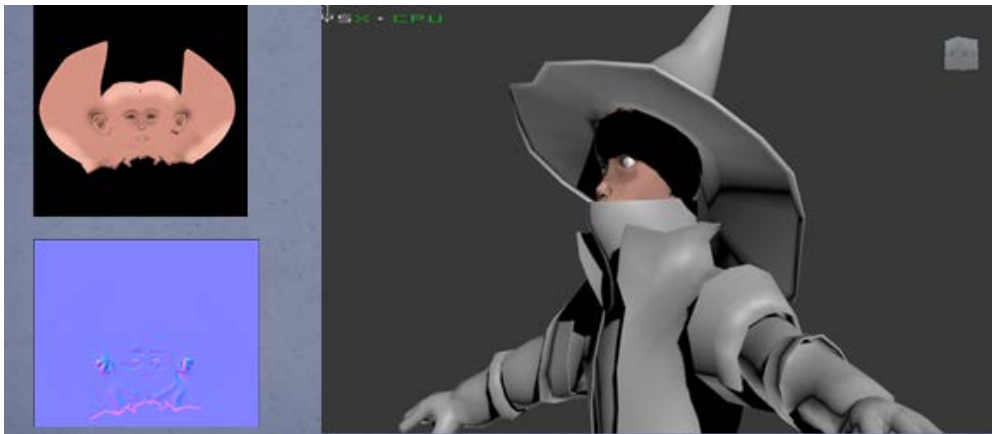


Рис. 4. Процесс текстурирования трехмерной модели

#### *Подготовка анимации*

##### *Создание скелета*

Почти все персонажи анимируются с помощью какой-либо скелетной анимационной системы. Это значит, что у персонажа есть геометрия, состоящая из полигональной сетки, под которой находится скелет, используемый для деформации кожи.

В данной работе был создан скелет *Biped*, состоящий из готовых иерархически связанных друг с другом костей. Это готовый скелет, который можно подогнать под любую форму. Его первое преимущество – быстрота создания. Второе – снабдив персонажа таким скелетом можно использовать настроенные для него стандартные анимации, например ходьбы или бега. Для того, чтобы персонаж правильно анимировался, необходимо правильно выровнять *Biped* по сетке персонажа, задав его так, чтобы сетка деформировалась должным образом.

##### *Связывание оболочки и скелета*

Для того чтобы движения суставов человека были реалистичными, необходимо применить для модели модификатор *Skin*. Модификатор *Skin* – это скелетно-деформационный инструмент. Он позволяет деформировать объект другим объектом. После применения модификатора *Skin*

и назначения костей, каждая кость имеет капсулообразную оболочку *Envelope*. Вершины внутри этой оболочки движутся с костями. Где оболочки совпадают, там движение вершин плавно сочетается с каждой из оболочек.

Для того чтобы задать необходимые кости, в параметрах модификатора *Skin*, нужно нажать кнопку *Add* и выделить все кости, которые должны определять изменения в геометрии оболочки. При учете воздействия на одну вершину нескольких костей, достигается наибольшая реалистичность анимации. Степень влияния каждой кости на вершину задается соответствующим весовым коэффициентом *Weight*, который может быть не больше единицы. В результате работы, необходимо настроить коэффициенты таким образом, чтобы минимум две кости воздействовали одинаково на один сустав или какое-либо другое место сгиба. Чем ближе параметр весов приближается к единице, тем сильнее кости будут воздействовать на выделенную точку. При изменении параметра весов будет меняться цвет точки.

#### *Анимация с использованием ключевых кадров*

Суть использования ключевых кадров заключается в создании ключей анимации для начального и конечного положения объекта, при этом состояние объекта в промежуточных стадиях просчитывает компьютер.

Для того чтобы начать работать с анимацией, необходимо выделить скелет *Biped*, войти во вкладку *motion/keyinfo*. Появится специальное меню с созданием ключевых кадров. С помощью инструментов перемещения и поворота необходимо изменять расположение костей скелета и ставить ключ для фиксации позы скелета.

В основном используются три ключа: *Set Sliding Key* для задания обратной кинематики, *Set Planted Key* для задания прямой кинематики и свободный ключ *Set Free Key*.

Вся работа проходит в окне, которое называется *TrackView*. Это окно представляет собой шкалу с обозначениями кадров. Верхний ползунок используется для выбора кадра. На каждый выбранный кадр ставится ключ, который фиксирует позу. При перемещении ползунка к следующему ключевому кадру программа автоматически покадрово сдвигает кости скелета.

В результате работы создаются базовые анимации покоя, ходьбы и бега.

#### Внедрение в игровой движок

Для того чтобы внедрить 3D-модель персонажа с готовой анимацией в игровой движок, необходимо эту модель экспортировать из *3ds Max* в формате *.fbx*. Для просмотра модели необходимо создать сцену, разработать ландшафт и настроить освещение.



Рис. 5. Внедрение в игровой движок

Все полученные файлы трехмерной модели должны быть собраны в одной папке. После чего необходимо запустить Unity3D, и проверить, чтобы файлы отобразились в окне Project. Требуемую модель нужно выбрать и перетащить в окно Scene. В модель нужно поместить Character controller – специальный стандартный инструмент, использующийся в качестве collider'a и дающий возможность помещать камеру. Этот компонент также необходим для того, чтобы перемещать модель в пространстве. Камера была размещена позади персонажа, так как это изображено на рис. 5.

Все анимации находятся в файле Take001. Для разбиения анимации требуется выделить файл Take001 и зайти во вкладку animation. В этой вкладке создать четыре файла, выбрать для них кадры запуска и остановки анимации. В пункте *WrapMode* следует выбрать, будет ли анимация зациклена или нет.

Таким образом настраиваются анимации. Для работы этих анимаций был написан скрипт движения персонажем с помощью клавиатуры.

В заключение следует отметить, что при разработке трехмерной модели персонажа для игрового приложения выявляется ряд особенностей.

Во-первых, трехмерная модель должна быть низко полигональной, имеющей не более ста тысяч полигонов.

Во-вторых, для готовой модели требуется обязательно сделать UV-развертку, иначе в текстурах будут появляться артефакты.

В-третьих, модель использует неиерархическую систему костей, для снятия ограничений при создании анимации.

### Библиографический список

1. Виртуальная археология(эффективность методов: материалы Второй Международной конференции, состоявшейся 1–3 июня 2015 года в Государственном Эрмитаже / Государственный Эрмитаж. – СПб.: Изд-во Гос. Эрмитажа, 2015. – С. 208–213.
2. Introducing ZBrush 4 / Eric Keller, 2011[Электронный ресурс]. – Режим доступа: <http://docs.Zbrush.com/Ek>
3. Официальный сайт Autodesk 3Ds Max [Электронный ресурс]. – Режим доступа <http://www.autodesk.ru/>свободный.
4. Официальный сайт Unity3D Max [Электронный ресурс]- режим доступа <http://www.unity3d.ru/>свободный.
5. Unity3D: Tutorials and Documentation [Электронный ресурс]. – Режим доступа: <http://docs.unity3d.com/ScriptReference/>.
6. 3D modelling and texturing demo reel focusing spartan warrior [Электронный ресурс]. – Режим доступа [http://www.youtube.com/ /watch?v=V8Zu1ltebk0.html](http://www.youtube.com/watch?v=V8Zu1ltebk0.html).свободный.
7. Урок № 29 Анимация врага и его боевых действий [Электронный ресурс]. – Режим доступа [http://advicez.ru/index.php?id\\_catalog=/56&id\\_position=191.html](http://advicez.ru/index.php?id_catalog=/56&id_position=191.html).свободный.
8. Виртуальная реконструкция объектов историко-культурного наследия [Электронный ресурс]. – Режим доступа: <http://books.google.ru/books?id=virtualreconstr/>.html.свободный.
9. Сайт компании в сфере компьютерной графики United 3d Labs, [Электронный ресурс]. – Режим доступа: <http://www.uni3dlabs.com.htm>.свободный.

УДК 51-73

**С. А. Жукова**

студент кафедры компьютерной математики и программирования

**В. И. Хищенко**

д-р техн. наук, проф. – научный руководитель

### АНАЛИЗ ЭКСПЕРИМЕНТАЛЬНЫХ РЕЗУЛЬТАТОВ ПО ИССЛЕДОВАНИЯМ ОКЕАНОЛОГИЧЕСКИХ ПРОЦЕССОВ

Вода занимает особое положение среди природных богатств Земли. Поверхностные воды в основном сосредоточены в Мировом океане, содержащем около 91% всей воды на Земле.

Из этого количества всего 6% составляет пресная вода, которую человек может использовать для своих нужд. При нынешних темпах роста потребления воды имеющегося количества хватит на ближайшие 25–30 лет [1].

Огромное значение вода имеет в промышленном и сельскохозяйственном производстве. Общеизвестна необходимость ее для бытовых потребностей человека, всех растений и животных. Для многих живых существ она служит средой обитания. Рост городов, бурное развитие промышленности, интенсификация сельского хозяйства, значительное расширение площадей орошаемых земель, улучшение культурно-бытовых условий и ряд других факторов все больше усложняют проблемы обеспечения водой. Много воды потребляют химическая и целлюлозно-бумажная промышленность, черная и цветная металлургия. Развитие энергетики также приводит к резкому увеличению потребности в воде. Значительное количество воды расходуется для потребностей отрасли животноводства, а также на бытовые потребности населения. Большая часть воды после ее использования для хозяйственно-бытовых нужд возвращается в реки в виде сточных вод.

Дефицит пресной воды уже сейчас становится мировой проблемой. Все более возрастающие потребности промышленности и сельского хозяйства в воде заставляют ученых всех стран мира искать разнообразные средства для решения этой проблемы [2, 3].

В настоящей работе рассмотрены источники данных об океанологии, находящиеся в свободном доступе, а также проведен анализ этих данных. В качестве источника данных в данной работе используется информация, собранная организацией TAO Information и представленная для общего доступа в сети Интернет [4]. На рис. 1 представлена карта с изображением станций-буйков – станций по сбору данных. Этих станций большое количество, расположены они в различных точках земного шара.



Рис. 1. Расположение станций по сбору данных

На рис. 2 представлена детализированная карта со станциями. На данном рисунке пользователю предоставляется возможность выбора станций по координатам.

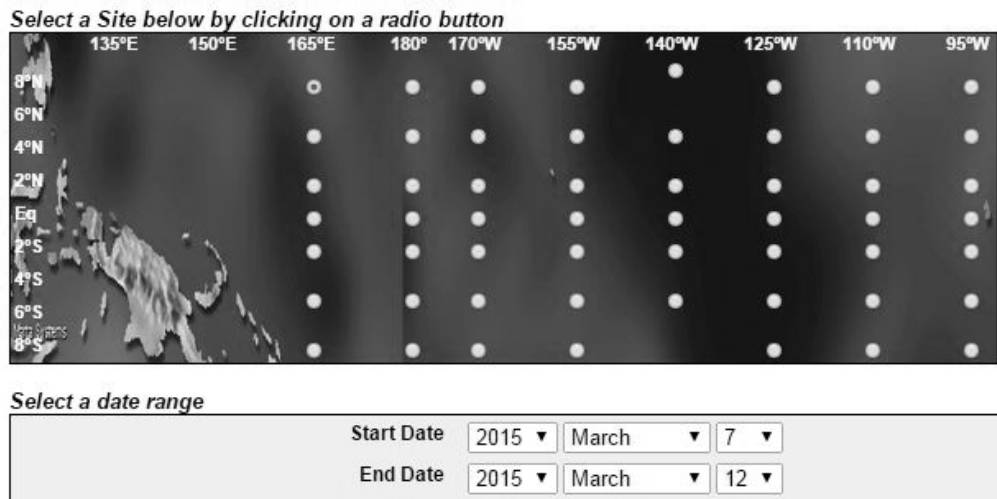


Рис. 2. Детализированная карта

Каждая станция по сбору информации является сложной системой с множеством датчиков и устройств, позволяющих собирать данные и регистрировать их в режиме реального времени. На рис. 3 представлена схема строения станции по сбору информации. В табл. 1 имеются названия элементов станции.

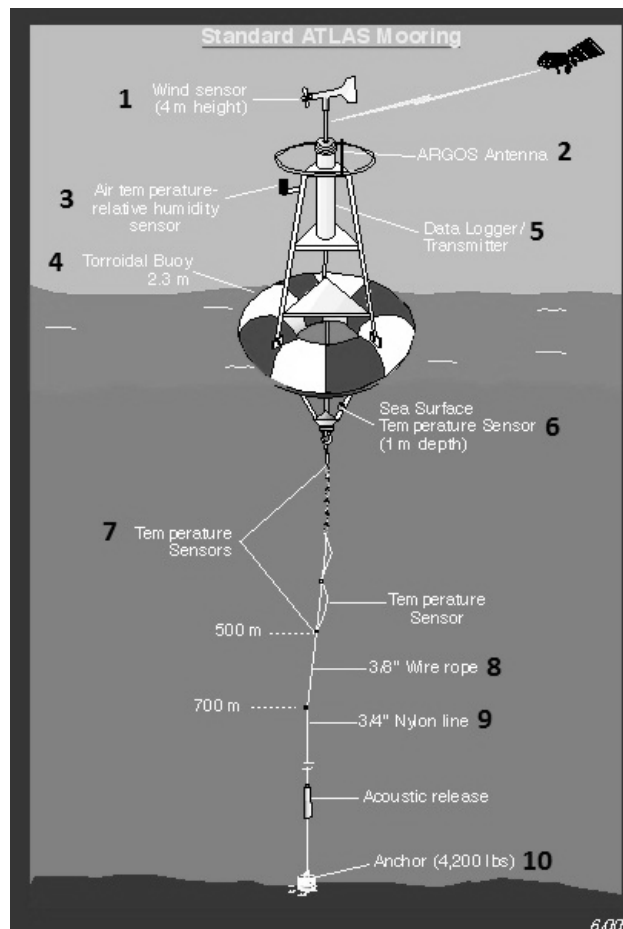


Рис. 3. Станция по сбору информации



Элементы станции по сбору информации

№ устройства	Название устройства
1	Датчик ветра, расположенный на высоте 4 м над поверхностью водоема
2	Спутниковая антенна АРГОС
3	Датчик температуры окружающей среды
4	Буй (в форме тора, диаметр 2,3 м)
5	Устройство регистрации данных
6	Датчик температуры поверхности воды (на глубине 1 м)
7	Набор датчиков измерения температуры воды на глубине от 1 м до 500 м)
8	Удерживающий трос
9	Удерживающая нейлоновая нить (эластичный трос)
10	Якорь

Станция по сбору информации позволяет получить данные разного рода и характера. Примеры данных, взятых со станции, приведены в табл. 2.

Таблица 2

Виды данных

Описание	Пример
Данные с датчика ветра – скорость ветра в определенном месте. Предполагается выбор интересующего диапазона времени для оценивания	<p>Wind Speed, at -4 m</p> <p>TAO/NDBC/NOAA, 0N124W</p> <p>2015</p> <p>Feb 16 2015</p>
Температура и относительная влажность	<p>Relative Humidity, at -3 m</p> <p>TAO/NDBC/NOAA, 0N124W</p> <p>2015</p> <p>Feb 16 2015</p>
Температура воды с датчиков на глубине	<p>Isotherms</p> <p>TAO/NDBC/NOAA, 0N124W</p> <p>2015</p> <p>Feb 16 2015</p>

## ТЕХНИЧЕСКИЕ НАУКИ

Описание		Пример
Данные с датчика на глубине 1 м	Температура воды	<p style="text-align: center;">Water Temperature, at 1 m</p> <p style="text-align: center;">deg c</p> <p style="text-align: center;">2015</p> <p style="text-align: center;">Feb 16 2015</p>
	Проводимость	<p style="text-align: center;">Conductivity, at 1 m</p> <p style="text-align: center;">S/m</p> <p style="text-align: center;">2015</p> <p style="text-align: center;">Feb 16 2015</p>
	Соленость	<p style="text-align: center;">Salinity, at 1 m</p> <p style="text-align: center;">PSU</p> <p style="text-align: center;">2015</p> <p style="text-align: center;">Feb 16 2015</p>
Данные с датчиков температуры на разных глубинах	На глубине 40 м	<p style="text-align: center;">Water Temperature, at 40 m</p> <p style="text-align: center;">deg c</p> <p style="text-align: center;">2015</p> <p style="text-align: center;">Feb 16 2015</p>
	На глубине 500 м	<p style="text-align: center;">Water Pressure, at 500 m</p> <p style="text-align: center;">dBar</p> <p style="text-align: center;">2015</p> <p style="text-align: center;">Feb 16 2015</p>

Также данный источник позволяет получить информацию, распространенную широко как географически, так и по времени. Иными словами, сбор данных ведется достаточно давно, и пользователю предоставляется возможность получить данные за интересующий период времени. Это также позволяет проводить анализ собранных данных, опираясь на их периодичность, отслеживая закономерность.

Так как данные представлены в виде случайных процессов, рассматривать их можно с помощью двух теорий.

1. Имея в качестве данных случайный процесс, мы можем рассматривать его с точки зрения теории выбросов. На рис. 4 представлены исходные данные.

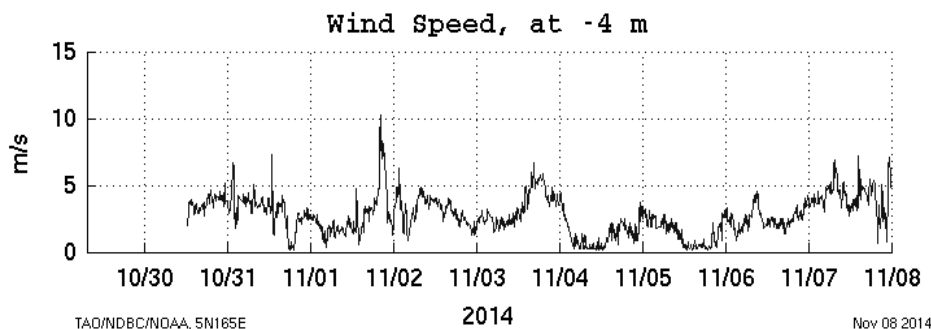


Рис. 4. Исходные данные. Ветровое волнение

Если обозначить некоторое среднее значение, вокруг которого должна изменяться величина данного параметра, то те значения, которые будут образовывать «пики», и будут выбросами. На рис. 5 представлены исходные данные с отмеченными выбросами.

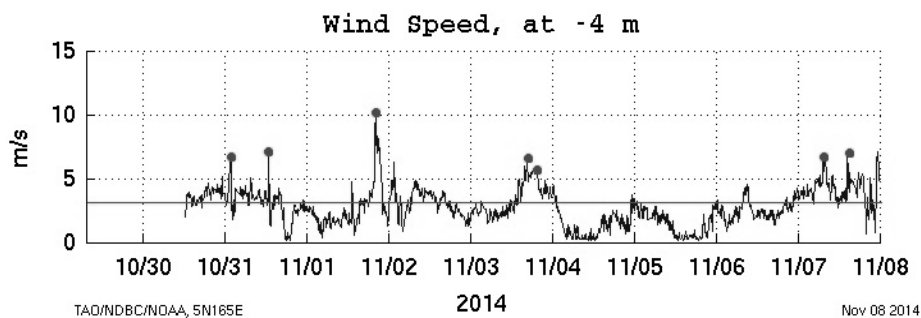


Рис. 5. Исходные данные с отмеченными выбросами

На основе данной теории можно реализовывать исследования по таким данным, как среднее число выбросов на интервале времени, средняя продолжительность выбросов и т. д.

2. Исходные данные можно рассматривать, опираясь на теорию пересечения уровней, к основным разделам которой можно отнести:

- исследование вероятностных характеристик выбросов для различных моделей случайных процессов;
- исследование вероятностных характеристик выбросов при линейных и нелинейных преобразованиях исходного процесса;
- исследование статистики выбросов и вероятностный анализ прикладных задач, решение которых основано на использовании характеристик выбросов [5].

### Библиографический список

1. Новиков Ю. В. Экология, окружающая среда и человек. М.: ФАИР-ПРЕСС, 2000.
2. Лавёров Н. П. Актуальные проблемы океанологии. М.: Наука, 2003.
3. Степанов В. Н. Мировой океан. М.: Знание, 1974.
4. TAO NDBC [Электронный ресурс] – Режим доступа: <http://tao.ndbc.noaa.gov/> (Дата обращения: 26.04.2015).
5. Тихонов В. И., Хименко В. И. Проблема пересечения уровней случайными процессами. Радиотехнические приложения. Радиотехника и электроника. Т. 43. № 5. 1998.

УДК 681.3.06

**А. Э. Зянчурин**

студент кафедры компьютерной математики и программирования

**М. Ю. Охтилев**

д-р техн. наук, проф. – научный руководитель

## **РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРАКТИВНОГО ИНТЕРФЕЙСА 3D-МНЕМОСХЕМ В СОСТАВЕ АСУ ПОДГОТОВКИ И ПУСКА РАКЕТ КОСМИЧЕСКОГО НАЗНАЧЕНИЯ**

Взаимодействие пользователя с вычислительной машиной является важнейшим звеном вычислительного процесса при решении различных прикладных задач как научного, так и производственного плана. При разработке программ в области организации рыночных отношений, создания информационных ресурсов, приложений, выполняющих задачи исследовательского характера, а также приложений в составе автоматизированных систем управления, требуется реализация удобного для пользователя взаимодействия с ЭВМ.

В настоящее время для определения взаимодействия пользователя с ЭВМ используется понятие интерфейс.

Интерфейс – совокупность средств и правил, обеспечивающих взаимодействие устройств вычислительной системы и программ, а также взаимодействие их с пользователем. Кроме того, интерфейс определяют, еще и как средство общения одной системы с другой [3].

### Постановка задачи

Целью реализации взаимодействия пользователя с программным комплексом является разработка пользовательского интерактивного интерфейса 3D-мнемосхем в составе АСУ подготовки и пуска ракет космического назначения. Для достижения данной цели решались следующие задачи:

- поиск программного инструментария для разработки. То есть, вспомогательных библиотек, которые изначально нацелены на создание пользовательских интерфейсов;
- сравнительный анализ вспомогательных средств разработки, по выводам которого были отобраны самые оптимальные для конкретной задачи из первоначального списка;
- изучение основных методов библиотеки OpenSceneGraph;
- изучение основных методов библиотеки MyGUI;
- разработка платформы взаимодействия библиотек OpenSceneGraph и MyGUI, на основе которой достигалась конечная цель разработки.

Задача актуальна, поскольку является модернизацией и дополнением к уже существующей системе, и направлена на увеличение эффективности взаимодействия пользователя с автоматизированной системой управления. Далее будут рассмотрены варианты решения данной задачи и изложен алгоритм решения в общем виде.

### Анализ вариантов решения задачи

Для рассмотрения данного раздела введем понятие графического пользовательского интерфейса [2].

Графический пользовательский интерфейс – обеспечение возможности управления поведением вычислительной системы через визуальные элементы управления – окна, списки, кнопки, гиперссылки и так далее.

Первый из вариантов решения данной задачи заключался исключительно в использовании методов библиотеки OpenSceneGraph без привлечения сторонних пакетов, реализующих графический интерфейс. Но в процессе разработки возникла проблема временных затрат на реализацию. Дело в том, что создание графического пользовательского интерфейса требует приличный

по объему набор визуальных элементов, реализация которых «с нуля» и объединение их после этого в систему ведет к необходимости затрат по времени недопустимых в рамках решения данной задачи. Исходя из этого, был разработан второй, более эффективный, вариант решения, который заключался в объединении средств и методов библиотеки OpenSceneGraph и библиотеки MyGUI. Данный подход оптимизирует разработку интерфейса, что существенно сокращает временные затраты и повышает качество программного продукта.

#### Использование библиотеки MyGUI

Для большего понимания, что такое MyGUI, и по каким причинам выбор был сделан в пользу данной библиотеки, рассмотрим ее преимущества.

Основным строительным материалом пользовательского интерфейса является виджет. Виджет – это прямоугольная область с некоторыми свойствами и логикой взаимодействия с пользователем. Виджеты поддерживают иерархию для создания более сложных виджетов, загрузку из XML и другие особенности [1]:

- поддержка управления прозрачностью;
- поддержка выравнивания: виджет можно сделать растягивающимся вместе с родителем либо привязать к какой-либо грани родителя;
- поддержка маски (маской служит любая картинка) для управления «пикинг» мыши, т. е. присутствует возможность создания виджетов произвольной формы, например круглых;
- поддержка произвольного курсора: при наведении на виджет, курсор будет принимать указанную форму;
- поддержка окон подсказок (tooltip): у виджета нет прошитых окон подсказок, но виджет может сообщить, когда нужно показать окно и когда его скрыть. Соответственно, есть возможность конструирования любых по сложности окон подсказок;
- поддержка отсоединения и присоединения в иерархии: это дает возможность конструирования «плавающих» панелей;
- независимая иерархия от иерархии рендринга (процесс получения изображения): для сложных виджетов, например, окон, имеющих клиентскую зону (Window, ScrollView и т. п.) или виджетов, имеющих всплывающих «детей» (ComboBox) используются логические «дети», которые не совпадают с физическими. Тем не менее, они ведут себя как реальные «дети». Например, при скрытии ComboBox скрывается его выпадающий список, а в Window в списке «детей» находятся те «дети», которых мы добавили, хотя физически они на клиентском виджете;
- поддержка смены «скина» (шкурки) на лету;
- поддержка смены стиля (стиль поведения) на лету. Например, любого «дитя», можно сделать всплывающим – оно не будет обрезаться по размерам родителя, или перекрывающимся – оно будет перекрываться с другими «детьми».

На основе данных пунктов можно заключить, что данная библиотека довольно гибкий инструмент для создания пользовательского интерактивного интерфейса. Для решения задачи модернизации уже существующей системы это неоспоримый плюс. Также немаловажным достоинством является, то, что данная библиотека написана на языке C++, что упрощает процесс интеграции в программный комплекс.

#### Платформа взаимодействия библиотек OpenSceneGraph и MyGUI

Написание платформы взаимодействия OpenSceneGraph и MyGUI является основополагающей задачей модернизации уже существующего программного комплекса, потому как здесь мы реализуем процесс интеграции в систему и существенно экономим время на разработку.



Рис. 1. Общая схема платформы взаимодействия

Из данной схемы видно, что процесс делится на два больших блока, которые, в свою очередь, содержат определенный набор классов, выполняющих закрытое взаимодействие между собой. Реализация оболочки для методов MyGUI это тот момент, где в разработке мы максимально близки к дизайну. Разработка дизайна – это одна из важнейших задач в написании интерфейса, потому как в данной точке происходит взаимодействие пользователя со всей системой, и от того, насколько качественно будет реализован интерфейс, зависит и качество вычислительного процесса.

### Разработка дизайна

Дизайн – (англ. Design – проект, чертеж, проектирование) – вид проектной междисциплинарной художественно-технической деятельности по формированию предметной среды [2]. Особенность дизайнерской деятельности заключается в специфически эстетическом способе целостного осмысления и формирования объектов. Дизайн имеет дело с формальными качествами предмета, под которыми понимают не только особенности его внешнего вида, но и его структурные связи, придающие ему необходимое функциональное и композиционное единство.

Таким образом, можно заключить, что дизайн не является тривиальной задачей, а значит и подход к данной задаче должен быть на высоком уровне. Первое, что необходимо понимать при разработке, это важность того, что интерфейс обязан быть интуитивно понятным. Затем, для формирования предметной среды, должна быть хорошо изучена сама предметная область, чтобы вычислительный процесс был максимально оптимизирован. В этой части происходит слияние двух творческих блоков – от функции к форме и от формы к функции. На этом этапе мы достигаем цели – удобства эксплуатации, рациональность компоновки и высокий эстетический уровень.

Библиотека MyGUI располагает средами разработки дизайна интерфейсов. На их основе и будет построена дизайнерская часть проекта. Для большего понимания раскроем понятие «скин». Скин – это визуальный элемент отображения для виджетов или некая графическая оболочка для виджета.

Чем сложнее виджет, тем сложнее его скин. Например, кнопка состоит всего из одного квадрата текстуры, а вот уже полоса прокрутки состоит из подложки, двух кнопок по бокам и кнопки, которая перемещается посередине. Получается что скин – это иерархическое дерево простых элементов, некий шаблон. Сложные элементы могут быть частью еще более сложных элементов. Например, полоса прокрутки является частью списка.

Для упрощения создания скина процесс разбит на две части [1]:

- скиннинг – создание простейших элементов на основе текстуры;
- макетирование – создание иерархии на основе простейших элементов.

### Скиннинг

Этот этап выполняется.

На этапе скиннинга, осуществляемом в редакторе скинов SkinEditor, выполняются следующие операции:

- размечается текстура для различных состояний элемента;
- размечаются места растягивания элемента;
- размечается место расположения текста, если он нужен элементу.

После скиннинга мы имеем простейший элемент, который можно уже использовать в редакторе макетов и даже непосредственно при создании элементов, но в таком виде элемент не поддерживает свойства и иерархию. Элемент будет иметь состояния и растягиваться, но не будет, например, иметь текст. Такой элемент будет похож на простую кнопку без текста.

### Макетирование

Этот этап выполняется в редакторе макетов LayoutEditor и включает в себя следующие операции:

- выбор шаблона для типа виджета;

- создание иерархии элементов для шаблона из простых или более сложных элементов;
- указание свойств шаблона.

После макетирования мы имеем полноценный шаблон, который можно указывать в качестве скина элемента.

Вот таким образом, если описывать процесс без углубления в тематику, выглядит дизайн в MyGUI.

#### Заключение

Данная разработка является ключевым звеном в системе автоматизированного управления, так как, без всякого преувеличения, можно сказать, что данный процесс невозможен без интерактивного пользовательского интерфейса, и от того, насколько качественно он выполнен, будет зависеть эффективность и работоспособность всей системы, а значит, и влиять на качество вычислений.

В данной статье была освещена лишь небольшая часть разработки программной системы, которая повествует о проблематике организации взаимодействия пользователя с вычислительной машиной или неким программным продуктом, что еще раз указывает на актуальность и важность решения данной задачи и похожих задач в технической сфере, при проектировании интерфейсов в АСУ подготовки и пуска ракет космического назначения.

#### Библиографический список

1. Семенов А., Евменов Дж. Документация библиотеки MyGUI // Users Interfaces, Skining MyGUI. 2013 г.
2. Патий Е. графический интерфейс пользователя. – URL: <http://mygui.info/>.

УДК 519.876.5

**И. С. Иванов**

студент кафедры проблемно-ориентированных вычислительных комплексов

**Е. А. Бакин**

канд. техн. наук, доц. – научный руководитель

### АНАЛИЗ АЛГОРИТМОВ ПОДАВЛЕНИЯ БОКОВЫХ ЛЕПЕСТКОВ ФАЗОМАНИПУЛИРОВАННЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

В телекоммуникационных системах при установлении синхронизации приемной и передающей станций, а также для оценки импульсной характеристики канала связи широко применяются сложные сигналы [1]. Сжатие сложных сигналов осуществляется, как правило, с помощью согласованных фильтров. Сигнал на выходе согласованного фильтра совпадает с автокорреляционной функцией сложного сигнала, уровень боковых лепестков которой может превышать уровень, необходимый в рамках решаемой задачи. Тогда используются рассогласованные фильтры подавления боковых лепестков [2]. Для количественной оценки эффективности применения таких фильтров актуальной задачей является создание программного средства, моделирующего существующие алгоритмы подавления боковых лепестков.

Наиболее распространенным и широко применимым классом сложных сигналов являются радиосигналы на основе бинарных последовательностей [3]. Бинарные последовательности можно записать в виде числового вектора  $S = \{s_1, s_2, \dots, s_N\}$  длины  $N$ , состоящего из элементов  $\{1, -1\}$ . Задача подавления боковых лепестков подразумевает поиск коэффициентов фильтра  $C = \{c_1, c_2, \dots, c_M\}$  длины  $M$ , который позволяет минимизировать одну из метрик сжатия сложного сигнала (например, суммарную энергию или пиковый уровень боковых лепестков).

Одной из метрик является энергия боковых лепестков сжатой фазоманипулированной последовательности. Можно показать, что задача нахождения коэффициентов соответствующего фильтра решается на основе метода наименьших квадратов [4]. Тогда:

$$C_{ME} = (\mathbf{S}^T \mathbf{S})^{-1} \mathbf{S}^T \hat{F} \quad (1)$$

где  $C_{ME}$  – коэффициенты фильтра сжатия,  $\mathbf{S}$  – матрица сдвигов фазоманипулированной последовательности размера  $M \times M + N - 1$ ,  $\hat{F}$  – отсчеты желаемого сигнала на выходе фильтра. В строки матрицы  $\mathbf{S}$  записываются отсчеты бинарной последовательности, сдвинутые на величину, соответствующую номеру строки. Например, при  $M = 5$  и  $N = 3$  матрица  $\mathbf{S}$  имеет следующий вид:

$$\mathbf{S} = \left[ \begin{array}{cccccc} \overbrace{s_1} & 0 & 0 & 0 & 0 & \\ s_2 & s_1 & 0 & 0 & 0 & \\ s_3 & s_2 & s_1 & 0 & 0 & \\ 0 & s_3 & s_2 & s_1 & 0 & \\ 0 & 0 & s_3 & s_2 & s_1 & \\ 0 & 0 & 0 & s_3 & s_2 & \\ 0 & 0 & 0 & 0 & s_3 & \end{array} \right] \left. \vphantom{\begin{array}{cccccc} \end{array}} \right\} M + N - 1 \quad (2)$$

Соответствующий вектор-столбец  $\hat{F}$  при  $M = 5$  и  $N = 3$  имеет вид:

$$\hat{F}^T = \underbrace{[0 \ 0 \ 1 \ 0 \ 0]}_M \quad (3)$$

Для моделирования алгоритма минимизации энергии боковых лепестков разработана моделирующая программа на языке высокого уровня MATLAB. Для демонстрации работы программы выбран код Баркера длины 5 (рис. 1). Длина фильтра  $M$  равна 25. Уровень подавления боковых лепестков составляет примерно  $-50$  дБ, что намного ниже, чем при применении согласованной фильтрации.

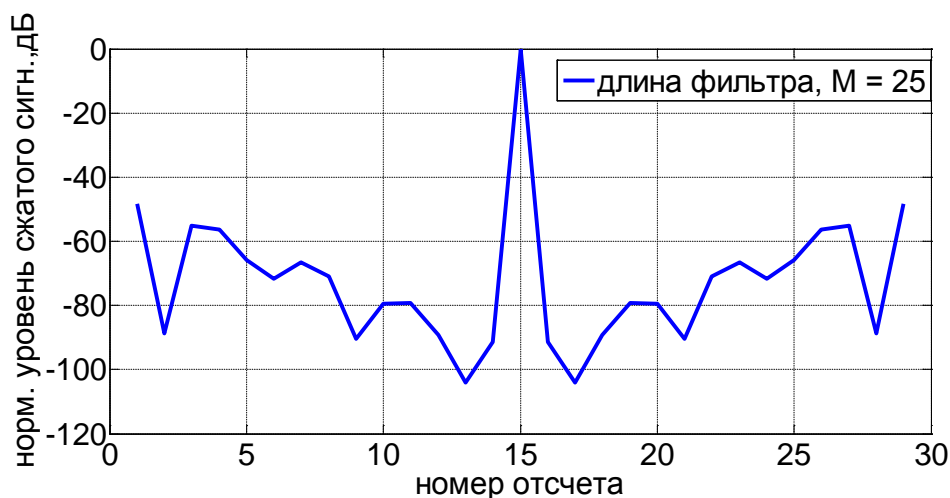


Рис. 1. Результат работы программы минимизации энергии боковых лепестков



Второй метрикой сжатия является пиковый уровень боковых лепестков фазоманипулированной последовательности [5]. Пиковая мощность отклика фильтра на бинарную последовательность заданного вида соответствует целевой функции:

$$\sum_{i=1}^N C_{MP} [(M - N) / 2 + i] \cdot S [i] \rightarrow \max \quad (4)$$

где  $C_{MP}$  – коэффициенты фильтра сжатия,  $S$  – отсчеты фазоманипулированной последовательности. Должны быть выполнены линейные ограничения:

$$\sum_{i=1}^M C_{MP} [i] \cdot S [i - k] \leq 1 \quad (5)$$

$$-\sum_{i=1}^M C_{MP} [i] \cdot S [i - k] \leq 1 \quad (6)$$

где  $k$  – коэффициент сдвига, принимает целочисленные значения в интервале  $1 - N \leq k \leq M - 1$ , причем  $k \neq \frac{M - N}{2}$ . Видно, что поиск коэффициентов соответствующего фильтра сводится

в задаче линейного программирования. В теории оптимизации существуют различные методы решения задачи линейного программирования, наиболее распространенный – симплекс-метод, разработанный Г. Данцигом, который позволяет найти базисное решение, если оно существует, за конечное число итераций [6].

В среде MATLAB решение задачи линейного программирования осуществляется с помощью встроенной функции  $C_{MP} = \text{linprog}(G, \mathbf{L}, B)$ , которая реализует обобщенный симплекс-метод. Функция возвращает вектор  $C_{MP}$  длины  $M$ , являющийся коэффициентами фильтра сжатия. Аргументами функции являются матрица  $\mathbf{L}$  и векторы  $G, B$ . Строки матрицы  $\mathbf{L}$  содержат коэффициенты линейных неравенств. Вектор-строка  $G$  содержит коэффициенты целевой функции, а в вектор-столбец  $B$  записываются свободные члены системы линейных неравенств. Для вычислений с помощью встроенной функции требуется представить аргументы в корректной форме. Примеры заполнения матрицы  $\mathbf{L}$  и векторов  $G, B$  при  $M = 5$  и  $N = 3$  представлены далее:

$$\mathbf{L}^T = \left[ \begin{array}{cccccccccccc} \overbrace{0 \ 0 \ 0 \ s_1 \ s_2 \ s_3 \ 0 \ 0 \ 0 \ -s_1 \ -s_2 \ -s_3}^{2(M+N-2)} \\ 0 \ 0 \ 0 \ s_2 \ s_3 \ 0 \ 0 \ 0 \ 0 \ -s_2 \ -s_3 \ 0 \\ 0 \ 0 \ s_1 \ s_3 \ 0 \ 0 \ 0 \ 0 \ -s_1 \ -s_3 \ 0 \ 0 \\ 0 \ s_1 \ s_2 \ 0 \ 0 \ 0 \ 0 \ -s_1 \ -s_2 \ 0 \ 0 \ 0 \\ s_1 \ s_2 \ s_3 \ 0 \ 0 \ 0 \ -s_1 \ -s_2 \ -s_3 \ 0 \ 0 \ 0 \end{array} \right] \quad M \quad (7)$$

$$G = \left[ \overbrace{0 \ s_1 \ s_2 \ s_3 \ 0}^M \right] \quad (8)$$

$$B^T = \left[ \overbrace{0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1}^{2(M+N-2)+1} \right] \quad (9)$$

Разработанное программное средство вычисляет свертку импульсной характеристики фильтра сжатия и исходной бинарной последовательности. В данном случае длина фильтра выбрана равной 25, а в качестве пилотного сигнала использован код Баркера длины 5. Пиковый уровень боковых лепестков в сжатом сигнале составляет –53 дБ (рис. 2).

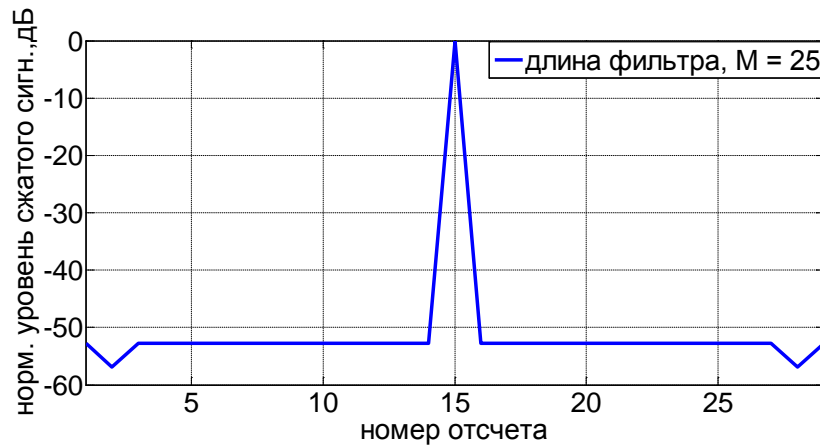


Рис. 2. Результат работы программы минимизации пикового уровня боковых лепестков

Произведен сравнительный анализ алгоритма минимизации энергии БЛ и алгоритма минимизации пикового значения БЛ. Первый алгоритм обладает лучшими характеристиками по подавлению энергии боковых лепестков. Оба алгоритма обладают спадающей зависимостью энергии БЛ от длины фильтра (рис. 3)

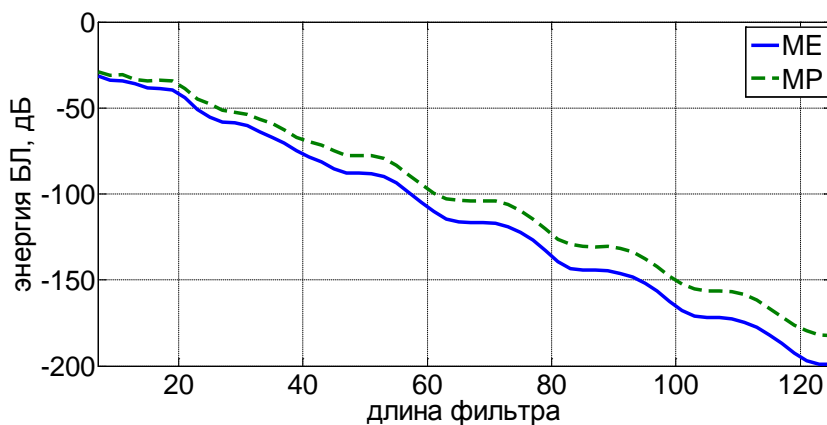


Рис. 3. Зависимости энергии БЛ от длины фильтра

Сравнение качества сжатия также можно провести на основе вычисления пикового уровня БЛ. Как и можно было ожидать, алгоритм подавления пикового уровня БЛ обладает лучшими показателями по данному параметру (рис. 4).

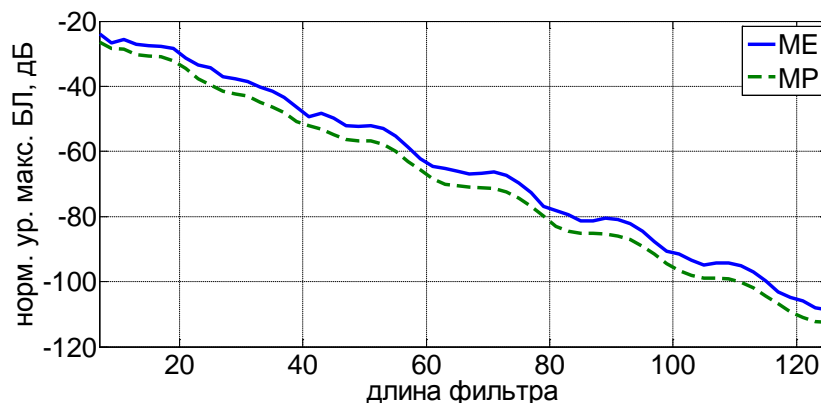


Рис. 4. Зависимости пикового уровня БЛ от длины фильтра

Рассмотренные в работе фильтры относятся к классу рассогласованных, поэтому вносят потери в отношении сигнал/шум. Специфика алгоритмов такова, что минимизация энергии и пикового уровня боковых лепестков происходит без учета потерь на рассогласование. В дальнейшем планируется количественно оценить потери в отношении сигнал/шум и модифицировать существующие алгоритмы с учетом ограничения потерь на рассогласование.

### Библиографический список

1. Sklar Bernard. Digital Communications: Fundamentals and Applications. – Prentice-Hall PTR, 2001. – 1079 p.
2. Levanon N. Radarsignals / N. Levanon, E. Mozeson. N.J.: Wiley-Interscience, 2004. 411 p.
3. Кук Ч. Радиолокационные сигналы / Ч. Кук, М. Бернфельд; пер. с англ. В. С. Кельзона. М.: Советское радио, 1971. 568 с.
4. Daniels R. Code inverse filtering for complete sidelobe removal in binary phase coded pulse compression systems / Robert C. Daniels, V. Gregers-Hansen. Washington DC: RadarDivision, NavalResearchLaboratory, 1980. 10 p.
5. Norman J. Minimum Peak Range Sidelobe Filters for Binary Phase-Coded Waveforms / J. Norman, B. Miller, R. Frank, W. Symons. IEEE, 1980. 7 p.
6. Банди Б. Основы линейного программирования / Б. Банди; пер. с англ. О. В. Шихеевой. М.: Радио и связь, 1989. 176 с.

УДК 004.657

**Д. М. Ильинская**

магистрант кафедры проблемно-ориентированных вычислительных комплексов

**Г. С. Евсеев**

канд. техн. наук, доц. – научный руководитель

### ОЦЕНКА ЭФФЕКТИВНОСТИ НОРМАЛИЗАЦИИ ТАБЛИЦ В СУБД MYSQL

При работе с базами данных (БД) обычно предъявляется три важных требования:

- 1) обеспечить целостность данных;
- 2) обеспечить высокую производительность выполнения запросов;
- 3) организовать экономичное хранение информации в БД.

На практике эти требования часто являются антагонистами [1].

Для учета требований 1 и 3 используют нормализацию таблиц базы данных. В результате, общее количество таблиц увеличивается, но объем хранимых данных уменьшается, т. к. исключается дублирование данных. Кроме того, нормализация улучшает условия для поддержания целостности данных.

С другой стороны, запрос на выборку в нормализованной базе данных становится много-табличным, и время его выполнения, как правило, возрастает.

В данной статье представлены результаты количественного исследования этого эффекта.

#### Подготовка эксперимента

В качестве объекта исследования выбрана БД, которая в ненормализованном виде содержит табл. 1 с полями: НЗ, П1, П2. Все поля являются целочисленными. НЗ – ключевое поле, П2 зависит от П1, т.е. его значения однозначно определяются значениями П1.

В результате нормализации (перехода от первой к третьей нормальной форме) вместо табл. 1 получаем табл. 2 с целочисленными полями КЛ и П1 и табл. 3 с полями НЗ, ВК, П2, которые также являются целочисленными. При этом НЗ – ключевое поле, ВК – ссылка на табл. 2.

Ключевые поля в таблицах имеют тип данных «счетчик».

Введем следующие обозначения: N1 – число записей в табл. 1 и в табл. 3; N2 < N1 – число записей в табл. 2.

Таблицы заполнялись следующим образом. Сначала заполнялась табл. 2 (в поле П1 равновероятными случайными числами в интервале от 1 до N2). Затем заполнялась табл. 3 (в поле П2 равновероятными случайными числами в интервале от 1 до N2). После этого заполнялась табл. 1 данными из табл. 2 и табл. 3. Таким образом, обеспечивались равные условия для сравниваемых вариантов.

Ниже представлены два соответствующих запроса [2].

1. Select count(H3)  
From Таблица\_1  
Where П1<A1 and П2<A2 into x;
2. Select count(H3)  
From Таблица\_2 innerjoin Таблица\_3 on Таблица\_2.Кл=Таблица\_3.ВК  
Where Таблица\_2.П1<A1 and Таблица\_3.П2<A2 intox;

Запросы составлены так, чтобы отключить вывод результатов на экран, значения A1 и A2 являются параметрами и определяют долю выбираемых записей.

Исследование времени выполнения запроса проводилось для трех случаев:

- однотабличный запрос с поиском по индексированным полям;
- однотабличный запрос с поиском по неиндексированным полям;
- двухтабличный запрос.

Среднее время выполнения запроса обозначается соответственно  $T_1, T_2, T_3$ .

Ниже представлены алгоритмы выполнения программы эксперимента (рис. 1) и подпрограммы «Оценить  $T_i$ » (рис. 2).

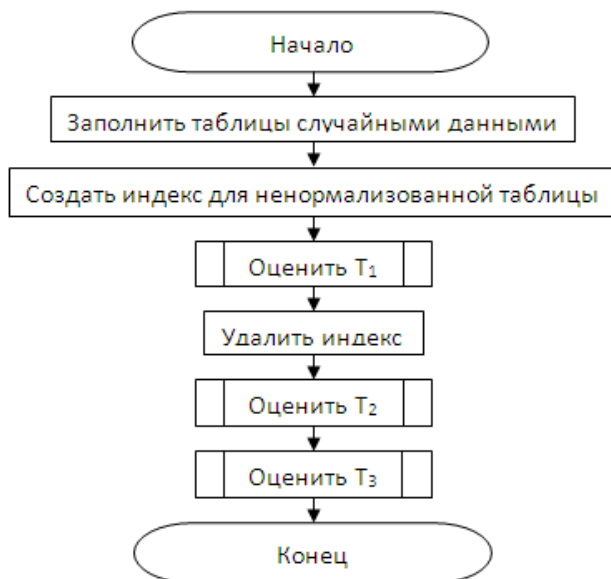


Рис. 1. Алгоритм программы эксперимента

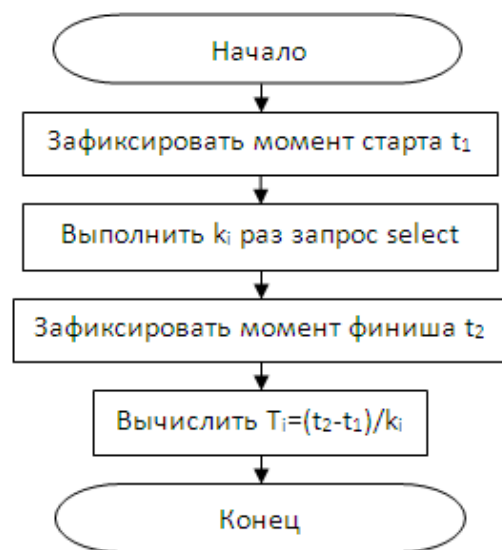


Рис. 2. Алгоритм подпрограммы «Оценить  $T_i$ »

Оценка времени выполнения одного запроса [3] делалась на основе измерения общего времени выполнения запроса  $k$  раз.

Основная программа выполнялась для разного объема таблиц (N1).

Общая погрешность оценки времени содержит следующие компоненты:

- методическая погрешность из-за округления времени до секунд;

– случайная погрешность, обусловленная тем, что периодически включаются на короткое время служебные программы операционной системы (например, программа, которая показывает текущее время).

Как показывает опыт, доля времени, занимаемая служебными программами, составляет менее 0,1%. Поэтому, если на время эксперимента отключить другие приложения, которые могут автоматически выполнять трудоемкие операции, например, автосохранение или загрузку обновлений из интернета, то случайной погрешностью можно пренебречь.

Для ненормализованной таблицы и нормализованных таблиц значения  $k$  выбирались так, чтобы общая погрешность составляла менее 1%. Это достигалось следующим образом. Если общее время выполнения  $k$  запросов более 100 с, то при округлении до секунд, погрешность округления составляет менее 1%.

### Результаты эксперимента

Результатом исследования является график, представленный на рис. 3. На графике показана зависимость времени выполнения запроса от числа записей в таблице. Для того чтобы зависимость имела более гладкий вид, выбраны логарифмические масштабы на осях абсцисс и ординат. На графике сплошными линиями показаны времена для выборки записей в ненормализованной таблице, а пунктирной линией – время для выборки записей из нормализованных таблиц. Все времена представлены в секундах.

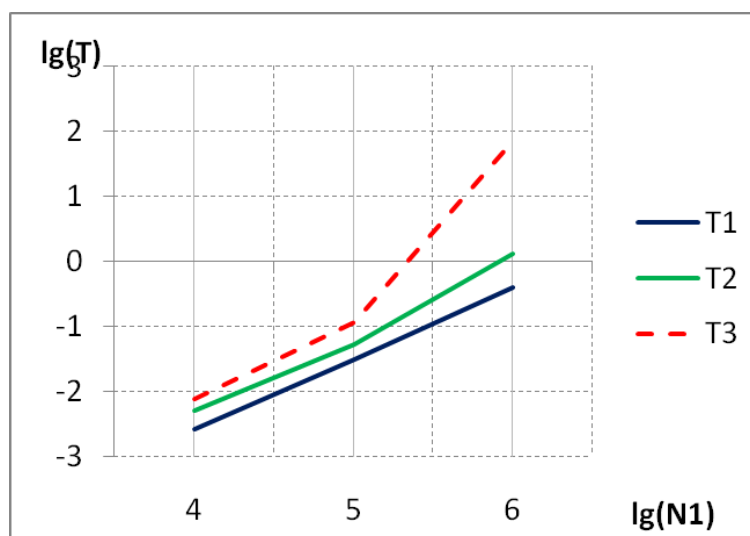


Рис. 3. График зависимости времени выполнения запроса от числа записей в таблице

На графике видно, что однотабличный запрос выполняется существенно быстрее, чем двухтабличный, причем, эта разница возрастает с увеличением числа записей в таблице. Наличие индекса существенно ускоряет выполнение однотабличного запроса и не влияет на время выполнения двухтабличного запроса.

Скорость выборки максимальна (время выборки минимально) в ненормализованной таблице с индексированными полями.

В табл. 1 представлена относительная скорость роста  $T_2$  и  $T_3$  от числа записей в таблице.

Относительная скорость роста  $T_2$  и  $T_3$  от числа записей в таблице

N1	$T_2/T_1$	$T_3/T_1$
10 000	2,38	2,94
100 000	1,94	3,90
1 000 000	2,47	134,82
10 000 000	4,74	329,11

При увеличении N1 относительный выигрыш для ненормализованной таблицы быстро увеличивается, достигая больших значений.

Подводя итоги, можно сделать следующие выводы.

Время выполнения запроса в ненормализованном варианте существенно меньше, чем в двухтабличном нормализованном варианте.

Время выполнения запроса в нормализованном варианте при увеличении объема таблицы растет существенно быстрее, чем для ненормализованного варианта.

Следует отметить, что запросы, касающиеся только данных из табл. 2, будут для обоих случаев однотабличными, следовательно, в нормализованном варианте БД они будут выполняться значительно быстрее, чем в ненормализованном варианте.

#### Библиографический список

1. Шварц Б., Зайцев П., Ткаченко В., Заводны Дж., Ленц А., Бэллинг Д. MySQL. Оптимизация производительности, 2-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2014. – 824 с., ил.
2. Кузнецов М., Симдянов И. MySQL 5. – СПб.: БХВ-Петербург, 2010. – 1007 с.
3. Григорьев Ю. А. Оценка времени выполнения SQL-запросов к базам данных. – Наука и образование: электронное научно-техническое издание. 2012. № 01.

УДК 519-7

#### **В. С. Капитонов**

студент кафедры проблемно-ориентированных вычислительных комплексов

#### **О. О. Жаринов**

канд. техн. наук, доц. – научный руководитель

### ПРИМЕНЕНИЕ РАСПРЕДЕЛЕНИЯ МЕЛ-ЧАСТОТНЫХ КЕПСТРАЛЬНЫХ КОЭФФИЦИЕНТОВ ДЛЯ ГОЛОСОВОЙ ИДЕНТИФИКАЦИИ ЛИЧНОСТИ ДИКТОРА

Задача верификации диктора по голосовым данным в настоящее время находит применение во многих областях, но наиболее широкое, конечно же, – в области охранной деятельности и безопасности информационных систем.

Метод решения поставленной задачи основывается на получении индивидуальных акустических характеристик пользователей, сопровождающейся записью в специальный банк данных. В процессе обработки входных данных (акустические параметры голоса диктора), они сравнива-

ются с теми, что были записаны в банк и, если есть совпадение, происходит идентификация личности пользователя.

Как и любой подход, связанный с биометрическими показателями, голосовая идентификация сильно зависит от факторов, связанных со здоровьем человека-диктора, его расположением относительно микрофона, характеристик записывающего устройства (микрофона), а также алгоритмов предварительной обработки сигнала и получения вектора признаков, характеризующих сигнал.

В данной статье рассмотрен один из самых изученных подходов в получении полезной информации об акустическом сигнале – метод мел-частотных кепстральных коэффициентов(MFCC) [1, 3].

Связь между единицами мел и герц

Мел – это эмпирически подобранная единица высоты звука, основанная на восприятии этого звука нашими органами слуха вследствие того, что воспринимаемая человеческим слухом высота звука не совсем линейно зависит от его частоты.

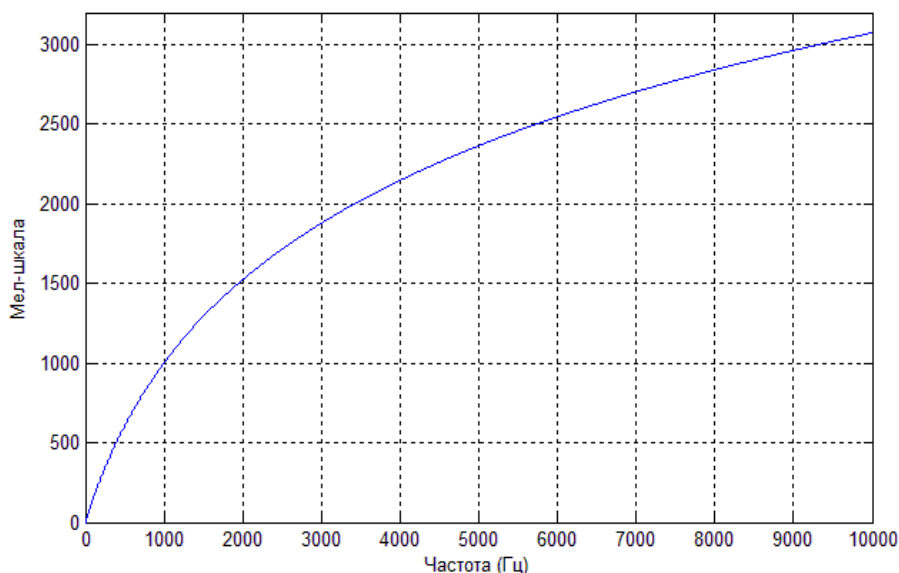


Рис. 1. Соответствие между мел и частотой, выраженной в герцах

Формула преобразования значения частоты звука  $f$  (Гц) в высоту  $m$  (мел) представляет собой следующее выражение:

$$m = 1127.01048 \cdot \ln \left( 1 + \frac{f}{700} \right) \quad (1)$$

Определение кепстра основано на понятии энергетического спектра. После получения амплитудного спектра  $S(\omega)$  сигнала, от каждого его значения в квадрате берется натуральный логарифм, далее применяется обратное преобразование Фурье.

$$C_s(q) = \frac{1}{2 \cdot \pi} \cdot \int_{-\infty}^{\infty} \ln[S(\omega)]^2 \cdot \exp(i \cdot q \cdot \omega) d\omega \quad (2)$$

Аргумент  $q$  имеет размерность время, но это особое, кепстральное время, иногда его называют «сачтота» или «кьюфренси».

Метод MFCC-идентификации диктора

Схема системы идентификации личности по голосу состоит из следующих этапов:

- уровень предварительной обработки входного сигнала;
- уровень обработки;
- уровень сопоставления и принятия решения.

Уровень предварительной обработки сигнала достаточно хорошо проработан в настоящее время и заключается в устранении шумов в оцифрованном сигнале и удалении участков, не содержащих речь.

Уровень обработки разберем подробнее. Сигнал после предобработки делится на отрезки-кадры длиной 25–30 мс ( $N$  отсчетов), к каждому отрезку применяется весовая функция и далее вычисляется дискретное преобразование Фурье.

$$X_k = \sum_{n=0}^{N-1} x_n \cdot w_n \cdot \exp\left(-\frac{2 \cdot \pi \cdot i}{N} \cdot k \cdot n\right), \quad k \in 0, 1..N-1 \quad (3)$$

Значения индексов  $k$  соответствует частоте  $f = \frac{Fs}{N} \cdot k$ , где  $Fs$  – частота дискретизации сигнала.

Полученное представление сигнала в частотной области разбивают на диапазоны набором треугольных фильтров. Границы фильтров вычисляются в шкале мел.

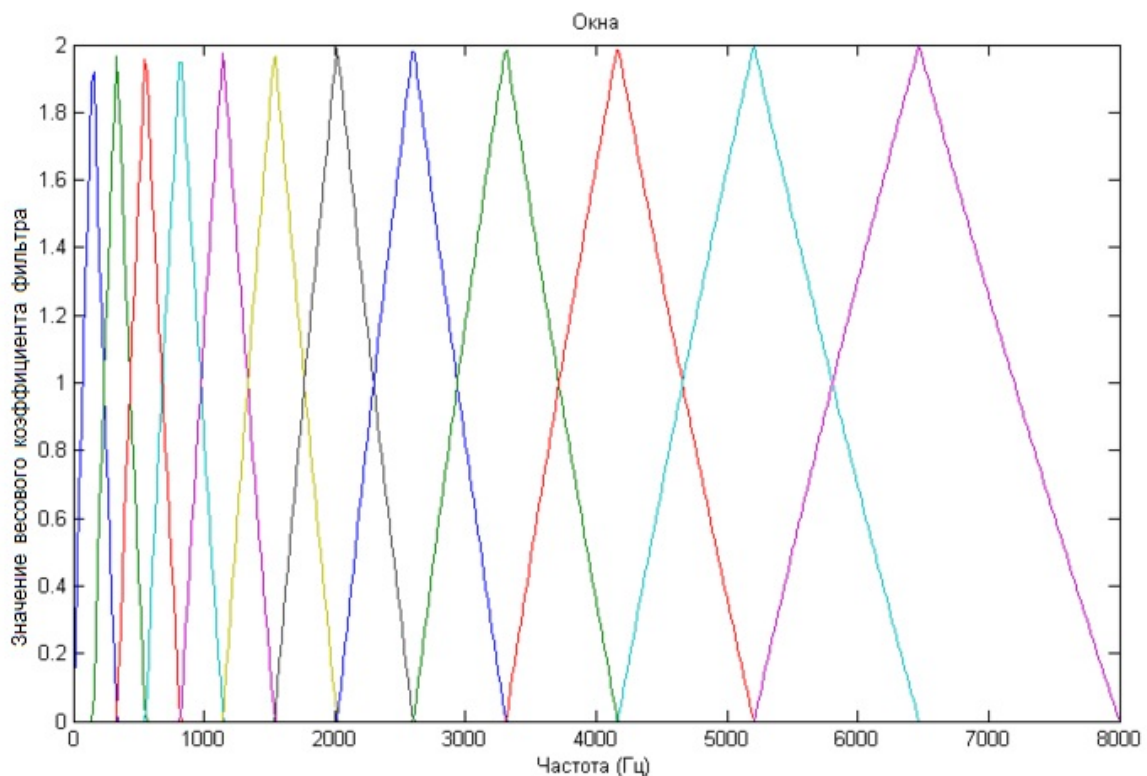


Рис. 2. Весовые коэффициенты набора треугольных фильтров с половинным перекрытием

На графике (рис. 2) заметно, что окна «собираются» в области низких частот, обеспечивая большую плотность покрытия частотного диапазона там, где оно необходимо для распознавания.



Пусть  $H_{m,k}$  – весовые коэффициенты фильтров, где  $m$  – номер фильтра,  $m \in 1..N_{FB}$ ,  $N_{FB}$  – количество треугольных фильтров,  $k$  – номер отсчета спектра,  $k \in 1..N$ ,  $N$  – количество отсчетов в спектре.

Тогда эти фильтры применяются к квадратам модуля спектра и полученные значения логарифмируются.

$$E_m = \ln \left( \sum_{k=0}^N |X_k|^2 \cdot H_{m,k} \right) , m \in 0..N_{FB} - 1 \quad (4)$$

Коэффициенты  $E$  несут в себе информацию об интенсивности сигнала на данной частоте. Заключительным этапом является взятие дискретного косинусного преобразования.

$$C_p = \sum_{m=0}^{N_{FB}-1} \left( E_m \cdot \cos \left( \frac{\pi \cdot p \cdot (m + 0.5)}{N_{FB}} \right) \right) , p \in 1 \dots N_{MFCC} \quad (5)$$

Вектор  $C$  содержит искомые мел-частотные кепстральные коэффициенты,  $N_{MFCC}$  – количество рассчитываемых коэффициентов.

На практике нулевой коэффициент не используется, так как несет информацию о полной энергии сигнала.

Для реализации метода была написана программа в среде моделирования MATLABR2012, обрабатывающая по формулам (2–5) оцифрованную аудиодорожку. Результат тестирования программы представлен на рис. 3 и рис. 4.

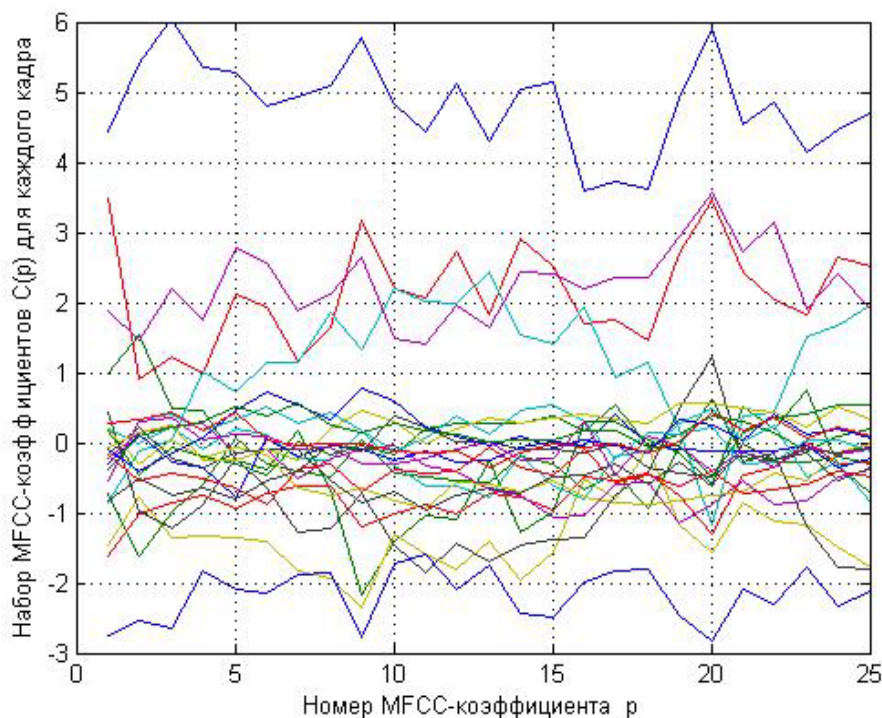


Рис. 3. Набор MFCC-коэффициентов по каждому кадру разбиения для фразы «Привет»

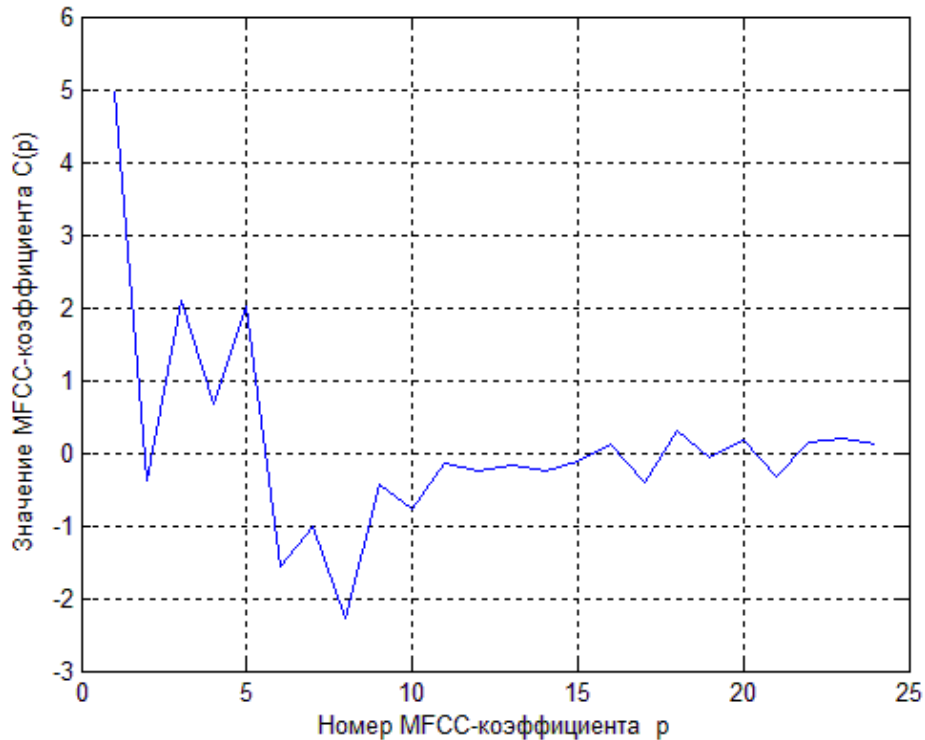


Рис. 4. Усредненный по каждому кадру набор мел-частотных кепстральных коэффициентов для фразы «Привет»

Коэффициенты по каждому кадру разбиения исходного сигнала усредняются в один набор, что является самой простой из обработок для получения одного вектора признаков при разбиении сигнала на кадры.

Последней стадией является классификация говорящего. Классификация производится вычислением меры схожести пробных данных и уже известных. Мера схожести выражается расстоянием от вектора признаков пробного сигнала до вектора признаков уже классифицированного. Наиболее простое решающее правило – «расстояние городских кварталов» (норма L1) [3].

$$D(C, CB) = \sum_{i=1}^{N_{MFCC}} |C_i - CB_i| \quad (6)$$

где  $CB$  – эталонный вектор MFCC-коэффициентов.

Сравнение наборов мел-частотных кепстральных коэффициентов для фраз, произнесенных разными дикторами

На рис. 5 представлены векторы-признаки мел-кепстральных коэффициентов для фразы «Привет» произнесенной десять раз и две фразы «One, Two, Three» одним диктором и две фразы «Привет» произнесенные другими дикторами. Для наглядности коэффициенты возведены в квадрат, а их количество выбрано равным 7.

Видно, что векторы-признаки двух других дикторов отличаются от общей массы векторов-признаков первого диктора.

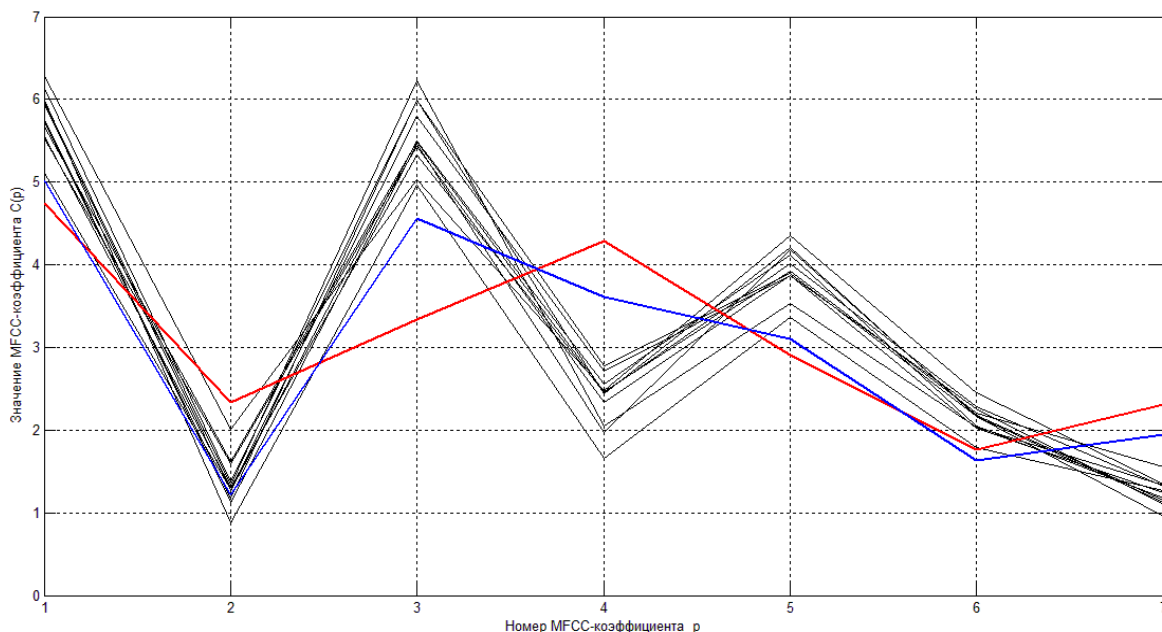


Рис. 5. Векторы MFCC-коэффициентов для 14 аудиозаписей:  
 черные – диктор № 1, красная и синяя – диктор № 2 и диктор № 3 соответственно

Перед началом классификации говорящего необходимо задать пороговое значение расстояния  $D$  (6), которое и будет определять вероятность точной верификации диктора, но следует учесть, что нельзя брать  $D$  слишком малым, иначе возрастет вероятность ошибки первого рода (несрабатывания даже при условии произношения фразы диктором, на котором и производилось обучение программы). Одним из выходов может послужить оценка математического ожидания и дисперсии расстояния для нескольких обучающих векторов признаков. Такие оценки улучшат качество работы алгоритма и соответственно уменьшат вероятность ложного срабатывания.

#### Вывод

В данной статье рассмотрен подход в голосовой идентификации личности диктора, основанный на применении распределения мел-частотных кепстральных коэффициентов. Разобраны три стадии голосовой идентификации, включающие предварительную обработку сигнала, выделение вектора-признака характеризующего диктора, обработку данных. В среде моделирования MATLABR2012 написан оригинальный текст программы, которая использовалась как средство моделирования разработанных алгоритмов. Программа для обработки всех дикторов содержит встроенную функцию системы по вычислению MFCC-коэффициентов и позволяет рассчитать выбранное пользователем количество MFCC-коэффициентов для массива данных, количество кадров разбиения задается пользователем.

#### Библиографический список

1. Агашин О. С., Карелин О. Н. Методы цифровой обработки речевого сигнала в задаче распознавания изолированных слов с применением сигнальных процессоров, НГТУ им. П. Е. Алексеева, 2012. С. 37–40.
2. Заковряшин А. С., Малинин П. В., Лелендин А. А. Вып. Управление, вычислительная техника и информатика, Известия АГУ, 2012. С. 156–160.
3. Первушин Е. А. Обзор основных методов распознавания диктора // Математические структуры и моделирование, Омск, 2011. Вып. 24. С. 41–54.
4. Расстояние городских кварталов // [https://ru.wikipedia.org/wiki/Расстояние\\_городских\\_кварталов](https://ru.wikipedia.org/wiki/Расстояние_городских_кварталов).

УДК 579.8:004.896

**Н. В. Кривец**

студент кафедры компьютерной математики и программирования

**М. Д. Поляк**

научный руководитель

### СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В ЛАБОРАТОРНОЙ ДИАГНОСТИКЕ ЭНТЕРОБАКТЕРИЙ

Довольно широко распространено мнение, что необходимость анализа данных в биологических экспериментальных исследованиях отсутствует, «если требуется анализ данных, это значит, что сам эксперимент плох» (в том смысле, что фиксируемый эффект незначителен). Однако применение статистики в медицинских и биологических исследованиях не ограничивается анализом данных. Статистические методы следует использовать также на этапе планирования биологического эксперимента или медицинского исследования. Мы придерживаемся мнения, что и в биологическом эксперименте для анализа данных необходимо применение статистики, в противном случае выводы нельзя считать научно обоснованными [1].

Несмотря на то, что современная медицина создала новые эффективные лекарства и технологии, эффективность здравоохранения серьезно проседает, в том числе и по причине отсутствия достаточного уровня автоматизации.

Энтеробактерии – это семейство грамотрицательных, палочкообразных, споронеобразующих бактерий, длиной 1–5 мкм, со жгутиками для передвижения. И, как известно, энтеробактерии включают в себя такие патогены, как: сальмонеллы, кишечные палочки и чумные палочки. Успешное лечение заразившихся пациентов зависит от своевременного и точного определения вида энтеробактерии-возбудителя. Для предотвращения массовых эпидемий необходимо своевременно выявлять патогенные бактерии в окружающей среде, прежде всего в водоемах и в продуктах питания [2].

В микробиологических лабораториях идентифицируют микроорганизмы в исследуемом материале и определяют их видовую принадлежность. Факт наличия или отсутствия возбудителя может устанавливаться как в организме больного, так и на объектах окружающей среды. Таким образом, различают медицинскую и санитарную диагностику.

Именно поэтому задача минимизации человеческого фактора и помощь в принятии решения при лабораторной диагностике является особенно важной и актуальной, так как бактерии семейства Enterobacteriaceae в случае непринятия своевременных мер представляют смертельную опасность для человека.

При дифференциации родов энтеробактерий ключевое значение имеет изучение биохимических и физиологических свойств, являющихся генетически более стабильными. Для идентификации энтеробактерий могут быть использованы различные диагностические системы, позволяющие сократить и упростить исследования. Так, тестовые полоски API-20E, представленные на рис. 1, используются для идентификации грамотрицательных стержней.



Рис. 1. Тестовые полоски API-20E

Номер профиля определяется из последовательности положительных и отрицательных тестовых результатов, а затем производится поиск в таблице кодирования, имеющей корреляцию между числами и видами бактерий.

Именно результаты биохимических исследований представляют собой одну из частей исходных данных. Другой же частью является таблица, которая предоставляет виды *Enterobacteriaceae*, составляющие до 99% наиболее часто встречаемых бактерий и самые полезные тесты для идентификации.

Таким образом, в нашем распоряжении имеется таблица с данными для каждого из известных классов энтеробактерий, а именно 26 тестов, среди которых присутствуют тесты на выявление индола, ферментацию лактозы и реакция Фогес-Проскауера. В ячейках данной таблицы указан процент случаев, в которых данный вид бактерии давал положительную реакцию на тест. А так же используются результаты проведенных тестов в лаборатории со значениями типа "+" – если реакция была положительная, "-" – отрицательная.

The image shows a software interface for entering test results. It consists of two columns of test controls. Each control has a label and three radio buttons labeled '+', '-', and '0'. The results are as follows:

Test Name	Result
Индол	-
Метиловый красный	-
Фогес-Проскауера тест	+
Цитрат	-
Сероводород H2S	-
Мочевина	-
Фенилаланин ДА	-
Лизин ДК	-
Аргинин ДГ	-
Орнитин ДК	-
Подвижность	-
Желатина	-
Глюкоза	-
Лактоза	0
Сахароза	+
Манит	-
Дульцит	-
Адонит	-
Сорбит	+
Арабиноза	-
Рафиноза	-
Рамноза	-
Ксилоза	-
Мелибиоза	+
ДНКаза	+
ОНПГ	0

Рис. 2. 26 тестовых значений программы

Если рассматривать значения результатов тестов для каждого класса бактерий в качестве вероятностей, тогда логичным было бы использовать методы теории вероятностей. В данном случае необходимо оперировать условными вероятностями, так как вероятности, указанные в рис. 3, имеют следующий вид  $P(T_i | A_j)$ , где  $T_i$  – это  $i$ -й тест из рис. 3, а  $A_j$  является  $j$ -м видом бактерии.

Следовательно, искомая вероятность того, что при определенных биохимических результатах будет именно указанная бактерия, приобретает следующий вид:

$$P(A_j | T_1, T_2, \dots, T_n) \sim \sum q_i, \text{ где } q_i = \begin{cases} P(T_i | A_j), \text{ если } H_i = 1 \\ 1 - P(T_i | A_j), \text{ если } H_i = 0 \end{cases}$$

В этой формуле  $P(A_j | T_1, T_2, \dots, T_n)$  является вероятностью того, что результатам тестов  $T_1, T_2, \dots, T_n$  в исследуемом посеве будет соответствовать бактерия  $A_j$ .

	Индол	Метил	Фосек	Цитра	Серос	Мочев	Фени	Лизин	Аргин	Орнит	Подви	Жела	Глюко	Лакто	Сахар	Манит	Дульц	Адопн	Сорби	Араби	Рафин	Рамин	Ксило	Мелис	ДНКз	ОНПГ
Citrobacter diversus	99	100	0	99	0	75	0	0	65	99	95	0	98	35	45	100	50	98	99	100	0	100	100	0	0	96
Citrobacter freundii	5	100	0	95	80	70	0	0	65	20	95	0	95	50	30	99	55	0	98	100	30	99	99	50	0	95
Enterobacter aerogenes	0	5	98	95	0	2	0	98	0	98	97	0	100	95	100	100	5	98	100	96	99	100	99	0	100	
Enterobacter cloacae	0	5	100	100	0	65	0	0	97	96	95	0	100	93	97	100	15	25	95	100	97	92	99	90	0	99
Escherichia coli	98	99	0	1	1	1	0	90	17	65	95	0	95	95	50	98	60	5	94	99	50	80	95	75	0	95
Hafnia alvei	0	40	85	10	0	4	0	100	6	98	85	0	98	5	10	99	0	0	0	95	2	97	98	0	0	90
Klebsiella oxytoca	99	20	95	95	0	90	1	99	0	0	0	0	97	100	100	99	55	99	99	98	100	100	100	99	0	100
Klebsiella pneumoniae	0	10	98	98	0	95	0	98	0	0	0	0	97	98	99	99	30	90	99	99	99	99	99	99	0	99
Morganella morganii	98	97	0	0	5	98	95	0	0	98	95	0	90	1	0	0	0	0	0	0	0	0	0	0	0	5
Proteus mirabilis	2	97	50	65	98	98	98	0	0	99	95	90	96	2	15	0	0	0	0	0	1	1	98	0	50	0
Proteus vulgaris	98	95	0	15	95	95	99	0	0	0	95	91	85	2	97	0	0	0	0	0	1	5	95	0	80	1
Providencia alcalifaciens	99	99	0	98	0	98	0	0	1	96	0	85	0	15	2	0	98	1	1	1	0	1	0	0	0	1
Providencia rettgeri	99	93	0	95	0	98	98	0	0	0	94	0	10	5	15	100	0	100	1	0	5	70	10	5	0	5
Providencia stuartii	98	100	0	93	0	30	95	0	0	0	85	0	0	2	50	10	0	5	1	1	7	0	7	0	10	10
Salmonella paratyphi A	0	100	0	0	10	0	0	0	15	95	95	0	99	0	0	100	90	0	95	100	0	100	0	95	0	0
Salmonella typhi	0	100	0	0	97	0	0	98	3	0	97	0	0	1	0	100	0	0	99	2	0	0	82	100	0	0
Salmonella, most serotypes	1	100	0	95	95	1	0	98	70	97	95	0	96	1	1	100	96	0	95	99	2	95	97	95	2	2
Serratia marcescens	1	20	98	98	0	15	0	99	0	99	97	90	55	2	99	99	0	40	99	0	2	0	7	0	98	95
Shigella serogroups A, B, and C	50	100	0	0	0	0	0	0	5	1	0	0	2	0	0	93	2	0	30	60	50	5	2	50	0	2
Shigella sonnei	0	100	0	0	0	0	0	0	2	98	0	0	0	2	1	99	0	0	2	95	3	75	2	25	0	90
Yersinia enterocolitica	50	97	2	0	0	75	0	0	0	95	2	0	5	5	95	98	0	0	99	98	5	1	70	1	5	95
Yersinia pestis	0	80	0	0	0	5	0	0	0	0	0	0	0	0	0	97	0	0	50	100	0	1	90	20	0	50
Yersinia pseudotuberculosis	0	100	0	0	0	95	0	0	0	0	0	0	0	0	0	100	0	0	0	50	15	70	100	70	0	70

Рис. 3. Биохимические реакции именованных видов, биогрупп и энтерогрупп семейства энтеробактерий

Используемая в формуле функция  $H_i$  может принимать следующие значения:

$$H_i = \begin{cases} 1, \text{ если } T_i = "+" \text{ и } \Delta_{\max} \leq P(T_i | A_j) \leq 1 \\ 1, \text{ если } T_i = "-" \text{ и } 0 \leq P(T_i | A_j) \leq \Delta_{\min} \\ 0, \text{ если } T_i = "+" \text{ и } 0 \leq P(T_i | A_j) \leq \Delta_{\max} \\ 0, \text{ если } T_i = "-" \text{ и } \Delta_{\min} \leq P(T_i | A_j) \leq 1 \end{cases}$$

Таким образом,  $H_i$  принимает значение 1, только в тех случаях, когда результаты теста соответствуют виду бактерии  $A_j$ . А так же принимает значение 0 в тех случаях, когда результаты теста не соответствуют бактерии  $A_j$ . И данная искомая вероятность пропорциональна сумме  $q_i$ .

Однако произвести расчет искомой вероятности не представляется возможным, в виду чрезмерного количества условных параметров, а именно: 26 тестов для каждой бактерии

и отсутствия дополнительных данных об априорных вероятностях. Поэтому предлагается использовать эвристический алгоритм.

Рассмотрим функцию  $h(t)$ , которая по своей сути является индикаторной функцией и способна принимать одно из трех возможных значений: «-1», если результат биохимического теста отрицателен, «0», если тест не проводился, и «1», если результат теста положителен.

$$h(t) = \begin{cases} -1, & \text{if } t = "-" \\ 0, & \text{if } t = "undefined" \\ 1, & \text{if } t = "+" \end{cases} ,$$

где  $t$  – один из 26 тестов.

Введем понятие баллов, «плохих тестов», а так же минимального и максимального порогов значений.

Балл – это величина, которая характеризует значение результата теста для энтеробактерий. То есть, 100 баллов будут соответствовать 100% значению в таблице с тестами.

Максимальный или минимальный порог значений – это пороги значений, которые будут являться условиями для индикации принадлежности к множеству «плохих тестов».

«Плохим тестом» называется тест, неудовлетворяющий критериям вхождения в оптимальный интервал результатов.

Для положительного ("+") теста количество баллов равно соответствующему значению вероятности положительности теста в таблице. Однако, если значение для данного теста меньше, чем минимально установленный порог, то данный тест входит во множество «плохих тестов».

Для отрицательных ("-") тестов количество баллов будет рассчитываться как разность максимального количества баллов за тест (100) и текущего значения в таблице. Так же как и для положительного теста существует индикация принадлежности к множеству «плохих тестов», но здесь она будет исходить из того, что если значение теста больше, чем максимально установленный порог, то тест считается «плохим».

Для «0» значений расчет баллов не производится.

Теперь же, если скомпоновать все полученные правила подсчета, можно составить следующую формулу, которая по своей сути является эвристической:

$$Z_i = \sum_{k=1}^n (h(t_k)(p_{i_k} - 50) + 50h^2(t_k)),$$

где  $Z_i$  – сумма баллов для данной бактерии по результатам тестов, а  $n$  – количество тестов.

Разработанное программное обеспечение представляет собой систему, помогающую в принятии решений при лабораторной диагностике патогенных энтеробактерий. Интерфейс программного продукта представлен следующим образом: с одной стороны на экране вы видите 26 различных биохимических тестов, а так же различные варианты их значений, с другой стороны, окно вывода результатов идентификации рода и вида бактерий. Сначала выбираются бактерии с наименьшим количеством «ошибочных» тестов, т. е. именно так которые лучше всего соответствуют полученным биохимическим результатам (тестам). А баллы, рассчитанные по формуле, используются для ранжирования только тех бактерий, которые одинаково хорошо соответствуют результатам биохимических тестов.

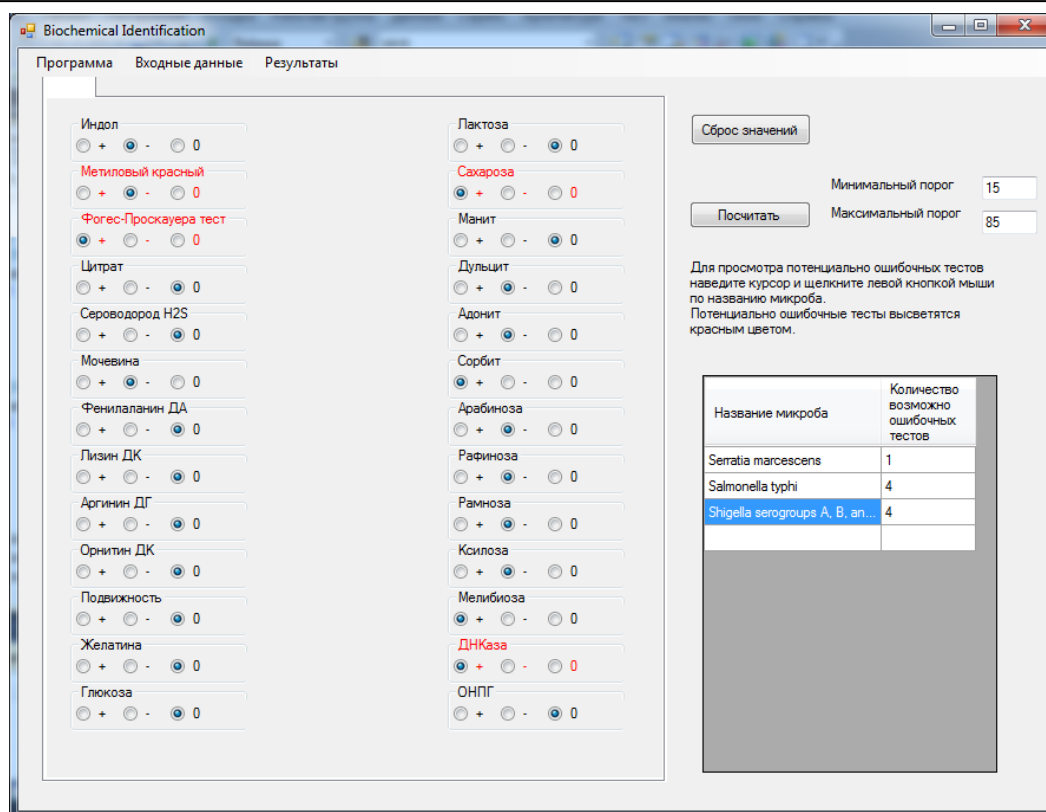


Рис. 4. Интерфейс главного окна при наличии результатов и «плохих» тестов

Данная программа нашла свое применение в медицинских лабораториях, в силу того, что алгоритм достоверно определяет род и вид бактерий, исходя из результатов биохимических тестов, а так же позволяет ускорить процесс определения бактерии, экономя тем самым время проведения лабораторных исследований, что определенно важно в данной области.

### Библиографический список

1. Реброва О. Ю. Статистический анализ медицинских данных. Применение пакета прикладных программ STATISTICA [Текст] / О. Ю. Реброва. – Москва: МедиаСфера, 2002.-312 с.
2. Biochemical identification of new species and biogroups of Enterobacteriaceae isolated from clinical specimens [Текст] / J. J. Farmer 3rd, B. R. Davis, F. W. Hickman-Brenner, A. McWhorter, G. P. Huntley-carter, M. A. Asbury, C. Riddle, H. G. Wathen-Grady, C. Elias, G. R. Fanning.-J. Clin. Microbiol., 1985, 21(1):46.
3. Sturmborg J. P. Handbook of Systems and Complexity in Health [Текст] / J. P. Sturmborg, C. Martin.-Springer, 2013. – 954 p.
4. Peat J. Medical Statistics. A Guide to Data Analysis and Critical Appraisal [Текст] / Peat J., Barton B.-Blackwell Publishing Ltd, 2005. – 324 p.

УДК 681.3.06

**Ю. А. Кузнецова**

магистрант кафедры проблемно-ориентированных вычислительных комплексов

**Г. С. Евсеев**

канд. техн. наук, доц. – научный руководитель

### ТРЕНАЖЕР ДЛЯ ОПЕРАТОРОВ ПО ВВОДУ ИНФОРМАЦИИ

В настоящее время разработано множество клавиатурных тренажеров, выполняющих следующие основные функции [1, 2]:



- увеличение скорости набора;
- уменьшение количества ошибок/опечаток;
- обучение на русском, английском и немецком языках.

Практически все предлагаемые тренажеры выполняют эти функции. Некоторые тренажеры предлагают обучающие уроки, поэтапное изучение клавиатуры и ведут статистический учет.

Разнообразие предлагаемого ПО позволяет не только подобрать тренажер, но и выбрать программу с подходящим уровнем обучения.

Краткий обзор характеристик некоторых популярных тренажеров приведен в табл. 1.

Таблица 1

Характеристики тренажеров

Тренажер	Преимущества	Недостатки
VerseQ	Запоминает допущенные ошибки и чаще выдает комбинации с теми же сочетаниями	Отсутствует полная справочная информация по работе с программой
Stamina	Реализует поэтапное обучение	Отсутствует учет ошибок
Klavarog	Ведет учет ошибок	Не ведется статистический учет
Соло на клавиатуре	Отображает количество ошибок, скорость, ритмичность набора	Длительный процесс обучения. Невозможно перейти на следующий уровень, пока не пройдешь предыдущий
Keybr	Позволяет набирать тексты и оценивать количество ошибок и скорость набора	Длительный процесс обучения

Предлагается расширить функциональность тренажера введением следующих функций:

- изменение условий для ввода текста;
- оценка уровня адаптации оператора;
- тренировка адаптации оператора к изменению условий ввода текста.

В предлагаемом тренажере изменение условий ввода текста обусловлено переключениями режима клавиатуры с одного языка на другой (переключение режимов Rus/Lat).

Идея реализации тренажера основана на том, что оператору предлагаются для ввода два текста: на русском и на английском языке. Качество работы оператора оценивается по двум параметрам:

- средняя скорость набора текста;
- средняя частота опечаток.

Средняя скорость набора текста определяется отношением времени ввода текста к его объему.

Средняя частота опечаток определяется отношением числа опечаток к объему текста.

В предварительном испытании предлагается ввести текст один за другим, чтобы оценить характеристики оператора по вводу каждого текста. В основном испытании оба текста разбиваются на куски. Куски обоих текстов перемешиваются, составляя общий текст. Для ввода такого текста уже требуется переключение алфавита с русского на английский и с английского на русский. При этом можно ожидать, что характеристики оператора ухудшатся. По степени этого ухудшения можно судить об уровне психофизических качеств оператора. При хорошей адаптивности оператора его характеристики в предварительном и основном испытании должны быть близкими.

Предлагаемый тренажер реализован в среде MSACCESS 2010. Для построения графиков зависимостей предполагается использовать MSEXCEL.

Результаты испытаний предполагается накапливать в базе данных для последующего статистического анализа.

Вид пользовательской формы представлен на рисунке.

## Интерфейс тренажера

Старт испытания	ФИ	Иванов А.А
Конец испытания	Дата	25.05.2015
	Врем	12:02

**Исходный текст**

There are many museums in London. One of them is the Tate Gallery in Millbank, which presents modern masters of England and France.  
 В Лондоне много музеев. Один из них - Тейт галерея в Милманке, в которой представлены современные художники Англии и Франции.

**Повторите текст**

7

Рис. Интерфейс тренажера

Порядок выполнения эксперимента.

1. Измерить характеристики оператора при вводе русского текста.
2. Измерить характеристики оператора при вводе английского текста.
3. Измерить характеристики оператора при вводе чередующихся кусков русского и английского текстов для разных значений коэффициента чередования.
4. Построить графики зависимостей характеристик оператора от коэффициента чередования.
5. Оценить способность оператора к адаптации.

Особенности применения тренажера.

Во-первых, тренажер может применяться для входного испытания при приеме на работу. В этом случае перед испытанием устанавливаются требуемые значения характеристик набора текстовой информации, и результаты испытания сравниваются с ними.

Во-вторых, тренажер может быть использован для повышения уровня тренированности сотрудников организации.

В-третьих, тренажер позволяет косвенно оценить уровень владения иностранным языком, а именно: по разности характеристик при вводе русского и альтернативного текстов.

### Библиографический список

1. Попов М. П. Эффективные приемы набора и редактирования текста. СПб.: БХВ Петербург, 2006.
2. <http://www.urikor.net/rus/Art03.html> – общий сайт.

УДК 004.657

**О. С. Лесникова**

студент кафедры проблемно-ориентированных вычислительных комплексов

**Г. С. Евсеев**

канд. техн. наук, доц. – научный руководитель

### НЕЯВНЫЕ БЛОКИРОВКИ ТРАНЗАКЦИЙ В СУБД MYSQL

Главным преимуществом SQL-сервера является высокая производительность при работе с большим количеством потоков запросов пользователей. Процесс обработки таких потоков предполагает параллельное выполнение транзакций. При этом для обеспечения целостности данных необходимо в ряде случаев блокировать выполнение отдельных транзакций на некоторое время. Для этого используется механизм блокировок, в котором предусмотрены как явные, так и неявные (устанавливаемые сервером) блокировки [1].

Явные блокировки осуществляются напрямую пользователем во время выполнения транзакции при помощи команд LOCK и UNLOCK TABLES. Команда LOCK подразумевает описание диапазона блокировки, будь то таблица или несколько ее строк. При ее использовании необходимо описывать сразу весь диапазон данных, которые пользователь собирается изменять, т. к. каждая последующая команда LOCK TABLES отменяет действие предыдущей. Команда UNLOCK TABLES снимает все блокировки, установленные в данной транзакции.

Пример текста транзакции с применением явных блокировок:

```
START TRANSACTION; //начало транзакции
LOCK TAB; //защита от изменения читаемых данных
SELECT ... FROM TAB; //запрос на чтение данных из таблицы
UNLOCK TABLES; //снятие защиты с данных
LOCK TABLES TAB1; //защита от параллельного изменения данных
UPDATE TAB1 ...; //обновление данных таблицы
UNLOCK TABLES; //снятие защиты с измененных данных
COMMIT; //конец транзакции
```

Неявные блокировки устанавливаются автоматически в зависимости от уровня изоляции и системы хранения для обеспечения целостности данных.

В MySQL 5.6 [2] наиболее часто используются системы хранения данных MyISAM и InnoDB. Механизм транзакций реализуется только для таблиц типа InnoDB, поэтому рассматриваются только такие таблицы. В системе InnoDB блокировки задействуют диапазон, содержащий от трех и более строк в зависимости от заданных в транзакции условий.

При использовании неявных блокировок сервер вставляет в текст транзакции инструкции, связанные с блокировками. Рассмотрим следующий пример.

Текст исходной транзакции:

```
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ; //задание уровня изоляции
для конкретной транзакции
START TRANSACTION; //начало транзакции
UPDATE TAB1 SET AGE=3 WHERE ID=1; //обновление данных таблицы
COMMIT; //завершение транзакции
```

Текст исполняемой транзакции:

```
START TRANSACTION;
LOCK TABLES TAB1; //подставленный механизмом MySQL оператор блокировки таблицы
...
```

```
UNLOCK TABLES; //снятие всех блокировок
COMMIT;
```

Механизм неявных блокировок транзакций подразумевает, что обработка транзакции осуществляется по некоторым правилам, установленным для нее по умолчанию или в явном виде. В данном примере задание этого набора правил представлено строкой SET TRANSACTION ISOLATION LEVEL REPEATABLE READ.

Уровни изоляции или разделения транзакций – наборы правил, по которым транзакция взаимодействует с другими транзакциями. Они введены в стандарте ANSI SQL-92 для решения ряда проблем, связанных с параллельным выполнением транзакций, и включают в себя четыре уровня [3]. На каждом уровне к числу запретов предыдущего уровня добавляются новые запреты.

1. Неподтвержденное/«грязное» чтение (read uncommitted). Запрещается одновременное редактирование данных, но разрешается чтение данных, которые изменяются чужими транзакциями.

2. Подтвержденное чтение (read committed). Запрещается чтение данных, которые изменяются чужими транзакциями, но разрешается чтение данных, которые в данный момент не изменяются чужими транзакциями.

3. Повторяемое чтение (repeatable read). Запрещается использование данных, отличных от первой полученной версии этих данных в ходе транзакции.

4. Упорядоченный уровень (serializable). Запрещается вставка и удаление данных в диапазоны, с которыми работают другие транзакции.

Взаимодействие параллельных транзакций на разных уровнях изоляции носит разный характер в зависимости от проблемы, которую устраняет конкретный уровень.

При повышении уровня изоляции возрастает защищенность целостности данных, но уменьшается средняя скорость выполнения транзакций за счет того, что ужесточаются условия для их параллельного выполнения. Продемонстрируем это на нескольких примерах.

Тексты используемых в примерах транзакций:

```
START TRANSACTION tr1;
SELECT * FROM tab1 WHERE Id=1; //чтение записи из tab1
SELECT * FROM tab1 WHERE Id=1; //повторное чтение записи из tab1
COMMIT;
```

```
START TRANSACTION tr2;
UPDATE tab1 SET Age=4 WHERE Id=1; //изменение записи в tab1
COMMIT;
```

Рассмотрим пример взаимодействия транзакций на уровнях Read uncommitted и Read committed (рис. 1).

Читающая транзакция tr1 начинается раньше, чем транзакция на обновление и осуществляет первое чтение данных. После этого выполняется транзакция tr2 на обновление тех же данных, которая завершится позже транзакции tr1. Измененные данные еще не зафиксированы, но на уровне Read uncommitted их чтение уже осуществляется, что нарушает целостность данных, полученных транзакцией tr1.

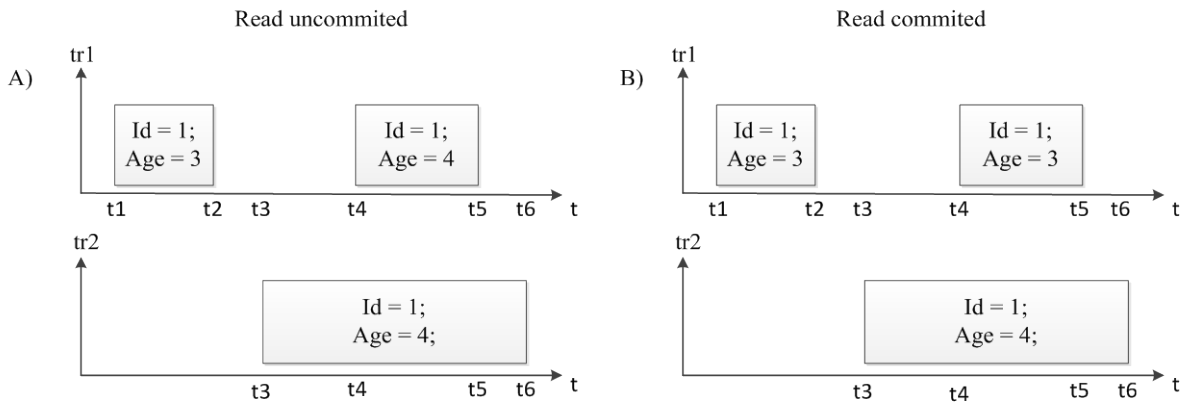


Рис. 1. Взаимодействие транзакций на уровнях изоляции Read committed и Read uncommitted

Рассмотрим следующий пример взаимодействия транзакций на уровнях Read committed и Repeatable read (рис. 2).

Читающая транзакция tr1 начинается раньше, чем транзакция на обновление и осуществляет первое чтение данных. После этого выполняется транзакция tr2 на обновление тех же данных, которая завершится раньше, чем произойдет второй запрос чтения в транзакции tr1. Внесенные ею изменения будут зафиксированы. На уровне Read committed транзакция tr1 при втором запросе чтения получит другие данные, что нарушает их целостность. На уровне Repeatable read в любом случае будут прочитаны те же данные, что в первый раз.

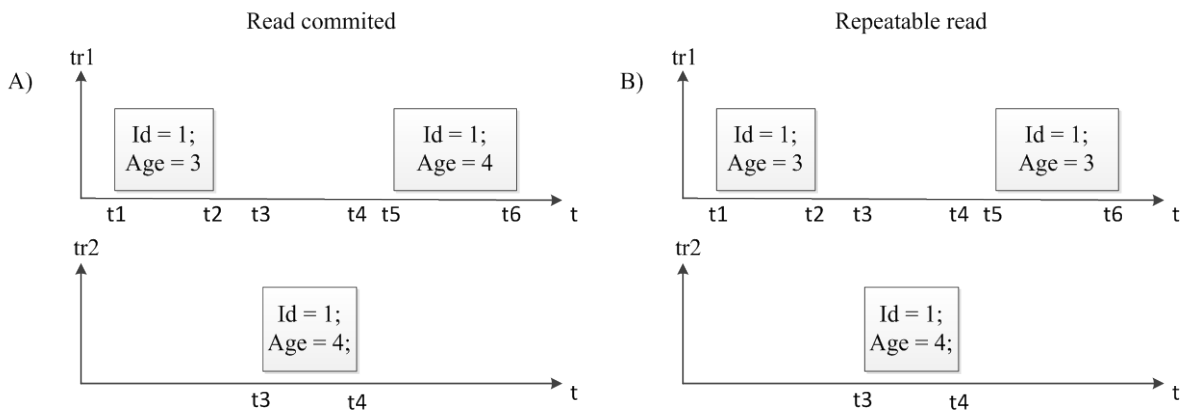


Рис. 2. Взаимодействие транзакций на уровнях изоляции Read committed и Repeatable read

Таким образом, на уровне Repeatable read транзакции часто обрабатывают целостные, но уже устаревшие данные. Для устранения такого эффекта используется блокировка намерения.

SELECT... FOR UPDATE – запрещает изменение прочитанных строк, т. е. устанавливает исключительную блокировку, как и оператор UPDATE. Запрос с таким чтением работает не с копией данных в буфере, а с последней версией данных.

Рассмотрим пример.

Тексты транзакций:

```
START TRANSACTION tr3;
SELECT * FROM Bakalavr FOR UPDATE; //чтение записей таблицы Bakalavr с установкой
исключительной блокировки
```

```

SELECT * FROM Magistr FOR UPDATE; //чтение записей таблицы Magistr с установкой
исключительной блокировки
SELECT sum(Stependia) FROM Bakalavr INTO N1; //чтение и запись в переменную N1 сум-
мы по столбцу таблицы Bakalavr
SELECT sum(Stependia) FROM Magistr INTO N2; //чтение и запись в переменную N2 суммы
по столбцу таблицы Magistr
SELECT N1+N2; //чтение суммы переменных N1 и N2
COMMIT;

START TRANSACTION tr4;
INSERT INTO Magistr(FIO, Stependia) VALUES ('Петечкин А.В.', 8800); //добавление строки
в таблицу Magistr
DELETE FROM Bakalavr WHERE FIO='Петечкин А.В.'; //удаление строки из таблицы Ва-
kalavr
COMMIT;
    
```

При использовании атрибута FOR UPDATE первая транзакция требует, чтобы прочитанные данные были актуальны до ее завершения. Поэтому старт второй транзакции будет задержан до завершения первой транзакции (рис. 3).

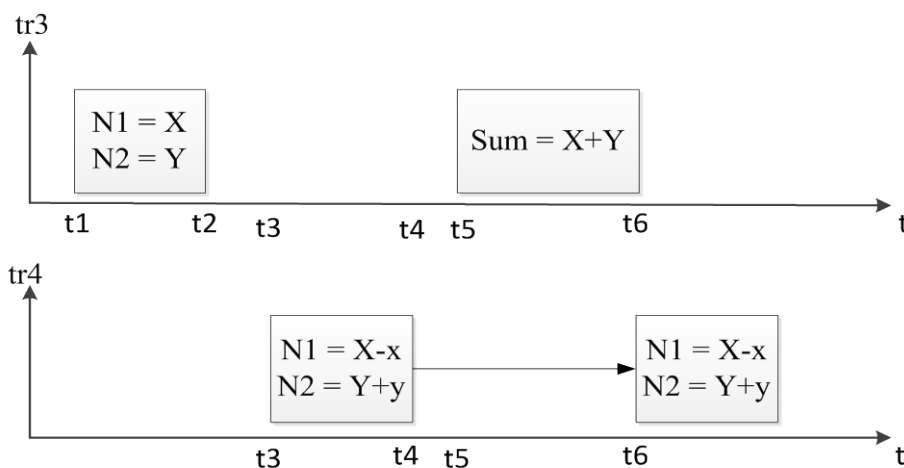


Рис. 3. Взаимодействие транзакций при использовании SELECT ... FOR UPDATE

В заключение можно сделать следующие выводы.

Все неявные блокировки в MySQL реализованы через уровни изоляции транзакций.

При уровне изоляции «read committed» несколько чтений в одной транзакции могут давать разные результаты. Этот уровень подходит для систем с большим числом параллельных редактирующих транзакций.

При уровне изоляции «repeatable read» внутри транзакции читается только значение данных, считанное в первый раз, вне зависимости от параллельных его изменений. Это необходимо для систем с частыми столкновениями читающих и редактирующих транзакций.

Атрибут «...for update» обеспечивает сохранение версии данных до завершения транзакции.

### Библиографический список

1. Крис Дж. Дейт. Введение в системы баз данных / Крис Дж. Дейт; Москва, Вильямс, 2006. 1328 с.

2. MySQL. Оптимизация производительности, 2-е издание / Шварц Б., Зайцев П., Ткаченко В. В. и др. Пер. с англ. – СПб.: Символ-Плюс, 2014. – 824 с., ил.
3. Стандарт ANSI SQL-92; 1992. 624 с.

УДК 004.42

**Е. А. Логвинчук**

магистрант кафедры компьютерной математики и программирования

**А. В. Бржезовский**

канд. техн. наук, доц. – научный руководитель

### ПРОЕКТ ПЕРЕНОСА БАЗЫ ДАННЫХ КОНФИГУРАЦИИ КОЛЛ-ЦЕНТРА В СИСТЕМУ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ КАССАНДРА

Роль баз данных в современном мире очень велика. Базы данных, основанные на реляционной модели данных и предложенные американским математиком Э. Ф. Коддом, существуют уже более 30 лет. Реляционная модель надолго стала стандартом представления структурированной информации по причинам простоты, использования реляционной алгебры в качестве подхода к манипулированию данными и универсальности алгоритмов выполнения запросов.

В настоящее время наблюдается увеличение объема информации, хранящейся в базах данных (БД), активное развитие облачных вычислений и многоядерных систем в различных сферах деятельности. Современные реляционные системы управления базами данных (СУБД), установленные на сегодняшний день, порой не справляются с таким родом задач, начинают постепенно давать сбои и задержки в работе из-за роста обрабатываемых данных. Переход на новую реляционную СУБД влечет за собой вложение определенного количества денежных средств. И поэтому им на помощь приходят нереляционные СУБД или, как их еще называют, NoSQL-СУБД [1]. Они в большинстве своем являются открытыми и не требуют вложений, разве что минимальных. Существуют задачи, в которых использование этих систем имеет преимущества перед реляционными СУБД. Одна из таких задач решалась в рамках данного проекта.

Целью настоящей работы является разработка структуры хранения конфигурационных данных колл-центра в СУБД Кассандра, а также анализ возможностей реализации запросов к создаваемой структуре для более быстрого и эффективного поиска информации.

Колл-центры – это специализированные офисы, которые используются для приема и передачи большого объема запросов по телефону. На сегодняшний день вся информация в одной из компаний по разработке программного обеспечения (ПО) для колл-центров хранится в реляционной СУБД. Конфигурационные данные колл-центра описывают объекты предметной области и их отношения, а также включают в себя описание схемы данных – типы объектов, их атрибуты, связи и сами значения. В качестве такого средства для описания выступают XML-файлы (eXtensible Markup Language), которые в данной задаче представляют собой аналог ORM (Object-relational mapping) – технологии объектно-реляционного отображения. Данная технология во многих системах играет ключевую роль во взаимодействии программных объектов с базами данных. В данной задаче сервер конфигурации по данным XML-файлов понимает, как читать данные из базы данных для дальнейшего их преобразования в объекты, с которыми будут работать клиентские приложения.

В последние годы в колл-центрах и во многих других компаниях из-за высоких темпов роста информации задачи перевода на нереляционные СУБД появляются достаточно часто. Как правило, при решении такого рода задач на общий просмотр не выкладываются полные решения с объяснениями или готовые программные продукты. Это связано и с соглашениями о неразглашении, и с уникальной структурой хранения информации каждой такой компании. Поэтому тяже-

ло найти какие-то общие решения, разве что только разные статьи и методы по переводу [2] или статьи, объясняющие основные термины и преимущества NoSQL-систем [3].

В данной работе в качестве нереляционной СУБД была выбрана СУБД Кассандра – легко масштабируемая NoSQL-СУБД с открытым исходным кодом. Кассандра отлично подходит для управления и обработки больших объемов структурированных, слабоструктурированных и неструктурированных данных в центрах обработки данных и облачных хранилищах. В данной работе требовалось после анализа способов хранения данных в СУБД Кассандра придумать способ хранения объектов колл-центра так, чтобы запросы, которые присутствуют на сегодняшний день, могли выполняться с не меньшей эффективностью. Модель данных Кассандры состоит из пространства ключей (аналог схемы базы данных в реляционной модели), семейств колонок (аналог отношений), строк (аналог кортежей) и колонок (аналог атрибутов). Для обеспечения быстрых функций записи, чтения и для легкой масштабируемости в Кассандре отсутствуют какие-либо связи, то есть если нужно хранить несколько сущностей, то их лучше хранить вместе. Тогда поиск по какому-либо критерию вернет сразу все запрашиваемые значения и не будет искать разбросанные куски данных по разным местам, соединяя их. Поэтому, при проектировании схемы в Кассандре предполагается, что исходить надо не из сущностей и связей, как при реляционном моделировании, а из запросов, которые будут выполняться. Пример, сравнения структуры хранения в реляционной БД и в БД Кассандра для данной задачи представлен на рис. 1.

Структура хранения в реляционной БД



Структура хранения в БД Кассандра

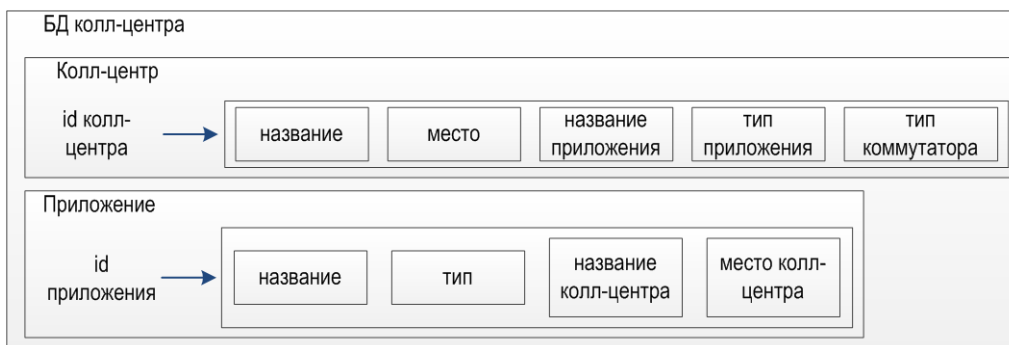


Рис. 1. Пример структуры хранения данных в реляционной БД и БД Кассандра

В примере сравнения структур хранения данных (см. рис. 1) отображены три таблицы базы данных: колл-центр, приложение и коммутатор. Предположим, что если нужны запросы по колл-центру, то все данные, с которыми связан колл-центр, в Кассандре будут храниться в одном колоночном семействе. Для доступа к данным Кассандра использует свой язык запросов CQL (Cassandra Query Language), напоминающий по синтаксису язык структурированных запросов SQL.

При проектировании новой структуры хранения были сделаны следующие заключения. Операция обновления данных в Кассандре по сути своей равна операции вставки и на этот счет есть даже специальный термин – `upsert` (`update+insert`). То есть получается, что делать вставку



данных будет безопасней и проще, чем обновлять какие-то поля. Идея состояла в том, чтобы в новом хранилище данные накапливались в строке по уникальному идентификатору, а там они будут храниться в отсортированном по времени создания порядке, самые свежие записи в начале.

После проектирования структуры хранилища в СУБД Кассандра была поставлена задача рассмотреть вопрос выполнения запросов к данной структуре. Как оказалось, для некоторой исходящей конфигурационной информации данная СУБД не самый подходящий механизм для осуществления поиска. Если колонок, по которым производится поиск, у строки мало, то тогда поиск в Кассандре становится эффективным и оправданным. Но если колонок для поиска много, более четырех или пяти, то ставится задача индексации всех данных, что приводит к большим временным затратам и задержкам в работе программы. Исходя из всего этого, было решено использовать поиск в Кассандре только для определенных групп колоночных семейств, а остальные синхронизировать с одной из поисковых систем. Такой поисковой системой был выбран продукт под названием Elasticsearch.

ElasticSearch – это современный, хорошо масштабируемый инструмент для полнотекстового поиска и аналитики с открытым исходным кодом. Он позволяет хранить, осуществлять поиск и анализировать большие объемы данных быстро и в режиме реального времени. Его работа основывается на индексировании данных и последующем выполнении различного рода сложных запросов к этим данным. В данной системе есть возможность индексировать данные различных СУБД, но с Кассандрой она связи напрямую не имеет. Поэтому следующей задачей было найти способ связи этих двух систем.

Для решения связывания СУБД Кассандра и поисковой системы Elasticsearch был написан программный модуль на языке C#, который выполняет следующие действия:

- создание колоночных семейств по текущим данным XML-файлов, хранящих конфигурационную информацию;
- генерацию тестовых данных по новой структуре хранения;
- выгрузку данных из СУБД Кассандра;
- индексацию данных в систему Elasticsearch.

Пользовательский интерфейс программы представлен на рис. 2.

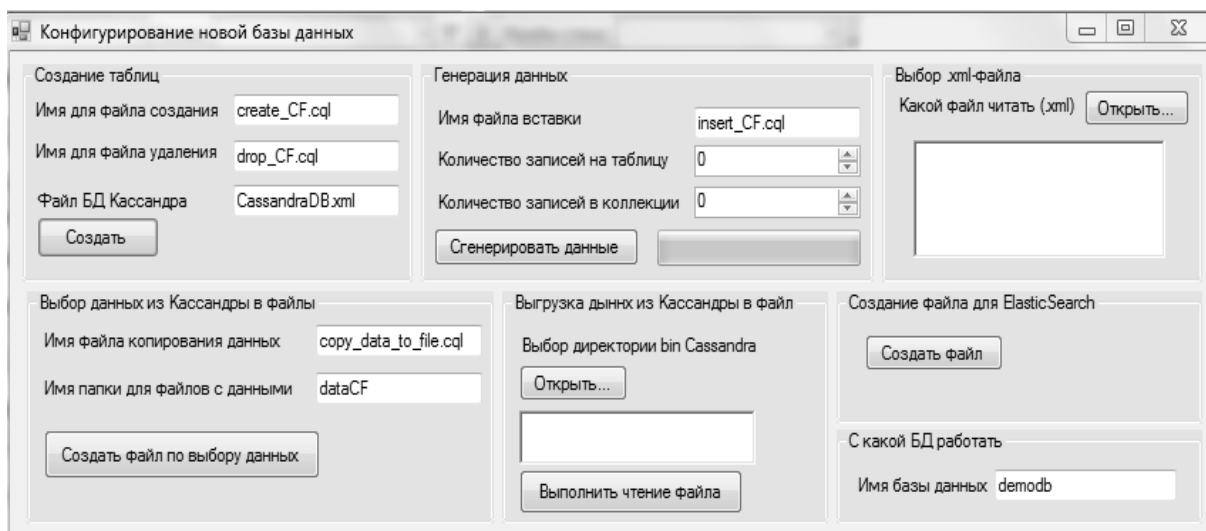


Рис. 2. Пользовательский интерфейс программного модуля

С моей точки зрения, разработанная структура нереляционной базы данных для хранения конфигурационной информации, а также связывание ее с одной из поисковых систем, позволят

оптимизировать весь процесс обработки информации, увеличить объем хранимых данных, обеспечить высокое качество исходящих продуктов и сократить количество сбоев и задержек программного обеспечения в работе колл-центра.

### Библиографический список

1. Константин Селезнев. От SQL к NoSQL и обратно [Электронный ресурс] <<http://www.osp.ru/os/2012/02/13014127/>>
2. Cassandra глазами Operations [Электронный ресурс] <<http://habrahabr.ru/company/lifestreet/blog/146115/>>
3. Препара С. Погружение в СУБД Apache Cassandra [Электронный ресурс] <<http://www.ibm.com/developerworks/ru/library/os-apache-cassandra/index.html>>

УДК 004.8

**О. Н. Максимова**

студент кафедры вычислительных систем и сетей

**А. В. Аксёнов**

научный руководитель

## ИСПОЛЬЗОВАНИЕ ВАРИАЦИИ ТЕСТА ТЬЮРИНГА ДЛЯ ВЫЯВЛЕНИЯ «МАШИННЫХ» ПАТТЕРНОВ ЧЕЛОВЕЧЕСКОГО ПОВЕДЕНИЯ

Одним из направлений развития информационных технологий является создание программ, в той или иной мере имитирующих речевое поведение человека. Подобные программы разного уровня сложности имеют достаточно широкое, хотя и не всегда «полезное» применение, но могут быть использованы для упрощения человеко-машинного взаимодействия. Например, в наши дни уже существует опыт использования программ в качестве продавцов-консультантов в интернет-магазинах [1].

При создании любого виртуального собеседника (чат-бота) решаются две задачи: во-первых, программу необходимо обучить «понимать» сообщения на естественном языке, то есть выделять ключевые фразы, анализировать контекст и предыдущие сообщения. И, во-вторых – формировать на основе полученной информации понятные человеку ответы в форме, максимально близкой к разговорной речи.

Взаимодействие человека с подобной программой будет тем более комфортным, чем ближе ее поведение к «человеческому». Но однозначно определить, какое поведение воспринимается как «человеческое», а какое – нет, на самом деле не просто.

В рамках данного исследования была предпринята попытка найти ответы на два вопроса:

1. По каким признакам человек определяет, что общается не с другим человеком, а с программой? (Выявление наиболее распространенных представлений о «человеческих» и «машинных» паттернах поведения).

2. Существуют ли такие сценарии человеческого поведения, которые при определенных условиях будут восприняты собеседником как «машинные»? (Выявление «машинных» паттернов человеческого поведения).

Первая часть исследования включала в себя онлайн-опрос, участникам которого было предложено ответить на ряд вопросов, связанных с программами-собеседниками. Всего в опросе приняло участие более 100 человек, преимущественно – люди в возрасте от 18 до 25 лет. При этом 40% опрошенных указали в качестве сферы своей деятельности сферу IT.

Согласно результатам опроса, практически все респонденты знают о существовании виртуальных собеседников (92%), при этом большая часть таких программ кажется им неубедительной: такого мнения придерживаются 57% опрошенных, в то время как за убедительность большинства чат-ботов проголосовало только 6,2%. Остальные участники опроса не смогли сделать выбор либо посчитали вероятность принятия программы за человека ситуативной.

Помимо этого, респондентам предлагалось представить себя участниками теста Тьюринга – эмпирического теста, идея которого была предложена Аланом Тьюрингом в статье «Вычислительные машины и разум», опубликованной в 1950 г. Исходной целью теста был поиск ответа на вопрос, может ли машина мыслить. Позднее от этой идеи отказались, и в наши дни тест рассматривается исключительно как способ проверки «убедительности» чат-ботов. Его участники должны, обмениваясь текстовыми сообщениями с невидимыми им собеседниками, определить, кто из них является человеком, а кто – имитирующей человеческое поведение программой. Тест Тьюринга считается пройденным, если программа сумеет обмануть 30% своих собеседников.

Среди стратегий, предложенных респондентам для выявления программ-собеседников, предпочтительными оказались анализ манеры письменной речи собеседника и его поведения. При этом к человеческим паттернам поведения были отнесены:

- чувство юмора;
- наличие ошибок и опечаток в сообщениях при правильном построении фраз;
- наличие собственного мнения по большинству вопросов;
- способность подробно рассказать о себе.

Также достаточно популярным среди респондентов оказался критерий восприятия собеседника как личности, что можно понимать как наличие у собеседника биографии, устойчивой системы взглядов и индивидуальных особенностей речи.

В качестве признаков машинного поведения были выделены такие, как наличие повторяющихся либо совершенно различных ответов на похожие вопросы, отличие манеры речи собеседника от разговорной и «странность» его собственных вопросов. Стоит отметить, что эти признаки относятся по большей части к слабым программам, имеющим небольшую базу знаний и не способным анализировать предыдущие сообщения собеседника. Но так же в некоторых случаях подобное поведение может демонстрировать и человек.

Данное предположение косвенно подтверждается результатами опроса: 57% респондентов соглашались с тем, что им станет сложнее определить, является ли собеседник программой или человеком, если предварительно станет известно, что среди собеседников-людей не все в должной мере владеют языком, на котором происходит общение.

Таким образом, можно сделать вывод, что в ситуации, когда необходимо отличить собеседника-человека от собеседника-программы, восприниматься как полностью «человеческие» будут лишь некоторые из всех возможных сценариев человеческого поведения. При этом значительное внимание будет уделяться владению разговорным языком.

Вторая часть исследования является экспериментальной и направлена на выявление сценариев человеческого поведения, которые в определенных условиях могут быть интерпретированы как «машинные». Для этого была предложена специальная вариация теста Тьюринга, единственным существенным отличием которой от общепринятой является отсутствие среди испытуемых реальных ботов.

В эксперименте принимало участие две группы: «судьи» и «участники», набранные случайным образом из числа студентов ГУАП. «Судьям» было сообщено, что они участвуют в тесте Тьюринга, в то время как «участники» считали, что помогают испытывать программу-мессенджер. Взаимодействие осуществлялось в течение 5 минут строго по одной реплике – «судья» не должен был отправлять следующий вопрос, пока не получил ответ на предыдущий. При этом вся переписка между «судьями» и «участниками» автоматически сохранялась.

Наиболее популярными стратегиями обнаружения ботов стали выявление особенностей личности собеседника («судьи» задавали участникам вопросы, касающиеся их настроения, произошедших в их жизни событий, каких-либо предпочтений и т. д.) и проверка знаний собеседника (несложные вопросы из химии, математики, астрономии).

Также использовались:

- анализ текста сообщений собеседника (наличие опечаток, смайликов и т. п.);
- анализ скорости ответа собеседника;

– проверка способности собеседника распознавать и производить вычисления («Напиши мне, сколько будет 2 в степени 0, 3 в степени 0, 4 в степени 1. Ответ напиши по порядку»).

При этом определить, что все собеседники являются людьми, смог только испытуемый, выбравший последнюю стратегию. Все остальные «судьи» принимали некоторых своих собеседников за ботов. Кроме того, нашлись такие «участники», которые приняли за ботов «судей», несмотря на то, что им не был заранее предложен такой вариант.

Анализируя характеристики, данные собеседниками друг другу, и тексты их переписок, удалось выявить ряд паттернов человеческого поведения, воспринимаемых как «машинные». В ходе эксперимента за ботов были приняты те участники, в диалогах с которыми выполнялось хотя бы одно из следующих условий:

- реакция собеседника на задаваемые вопросы не соответствует ожидаемой. Например, шуточные ответы «участников» на вопросы «судей» воспринимались последними как непонимание сути этих вопросов;
- собеседник отвечает на вопросы слишком медленно или вообще не отвечает на них (воспринимается как «зависание» программы);
- реплики собеседника являются «шаблонными», предсказуемыми в рамках темы диалога;
- собеседник реагирует только на сообщения, касающиеся ответов на его вопросы.

Стремление «судей» за короткий период времени получить как можно больше информации согласно выбранной стратегии было воспринято частью собеседников как проявление «машинного» поведения.

Из всего вышесказанного можно сделать вывод о существовании особых «машинных» сценариев человеческого поведения, влияющих на восприятие человеком его собеседника. Выявление таких сценариев, как и анализ закономерностей человеческого восприятия, может помочь в создании программ, более правдоподобно имитирующих человеческое поведение.

### **Библиографический список**

1. Насакин Р. Эволюция болтунов. // Компьютерра Online. – Дата публикации: 18.04.2007. Режим доступа: <http://old.computerra.ru/focus/315260/>
2. Тьюринг А. М. Могут ли машины мыслить? // в сб.: Информационное общество. – М.: АСТ, 2004. – С. 221–284.
3. Тьюринг А. М. Вычислительные машины и разум // в сб.: Хофштадер Д., Деннет Д. Глаз разума. – Самара: Бахрах-М, 2003. – С. 47–59.
4. Warwick K., Shah H. Human misidentification in Turing tests. // Journal of Experimental & Theoretical Artificial Intelligence. 2014, № 27 (3). PP. 123–135.

УДК 004.921, 004.77

**И. Е. Матусова, М. А. Суменсан**

студенты кафедры вычислительных систем и сетей

**Н. Н. Решетникова**

канд. техн. наук, доц. – научный руководитель

**Е. Ф. Королькова**

канд. искусствоведения, зав. сектором отдела археологии Восточной Европы и Сибири Государственного Эрмитажа – научный консультант

## **3D МОДЕЛИРОВАНИЕ В ИССЛЕДОВАНИИ ЗООМОРФНЫХ ОБРАЗОВ СКИФО-СИБИРСКОГО ЗВЕРИНОГО СТИЛЯ**

Скифо-сибирский звериный стиль – это художественно-символическая знаковая система для выражения мировоззрения раннекочевнических обществ Евразии VIII в. до н.э. – III в. н.э. Искусство звериного стиля – это «художественный язык», использующий иконические знаки

в виде образов животных, абсолютное доминирование которых в искусстве ранних кочевников дало название самому художественному феномену [1].

Исследовательские задачи, встающие при изучении исторического художественного феномена скифо-сибирского звериного стиля, относятся к области гуманитарных наук, однако применение компьютерных технологий и методов моделирования могут помочь реконструировать способы художественного формообразования и стилизации в древнем искусстве и воссоздать последовательность формирования образа и его логику. Такой междисциплинарный подход может оказаться перспективным для приближения понимания и разгадки смысла древних изображений.

3D моделирование различных объектов может быть эффективно использовано с целью сохранения и актуализации исторических событий, верификации исторических гипотез (их уточнения, проверки, выработки новых представлений). Как сами 3D модели, так и возможность взаимодействия с ними используются для аналитической работы, существенно повышая ее эффективность за счет визуальной составляющей и высокой степени информативности исследования с применением трехмерных технологий.

Использование компьютерных моделей при создании исторических реконструкций позволяет добиться нового уровня передачи информации и, как следствие, нового уровня восприятия, что позволяет приблизиться еще на шаг к восстановлению исторической действительности.

Рассмотрим процесс построения 3D моделей зооморфных образов на конкретных примерах.

Бляха от конского убора VII в. до н.э. с образом пантеры (рис. 1) (Сибирская коллекция Петра I) и ее 3D-модель (рис. 2).



Рис. 1. Бляха от конского убора, выполненная в скифо-сибирском зверином стиле



Рис. 2. 3D модель

Создание 3D модели осуществляется в Autodesk 3DMax 2015. Моделирование выполнено на основе примитивов с ручным редактированием элементов трехмерных фигур (граней, ребер, вершин) (рис. 3). Поскольку все объекты в трехмерной графике состоят из вершин, ребер и полигонов, всегда можно изменить геометрию объекта, изменяя эти три параметра (вручную, либо при помощи модификаторов).

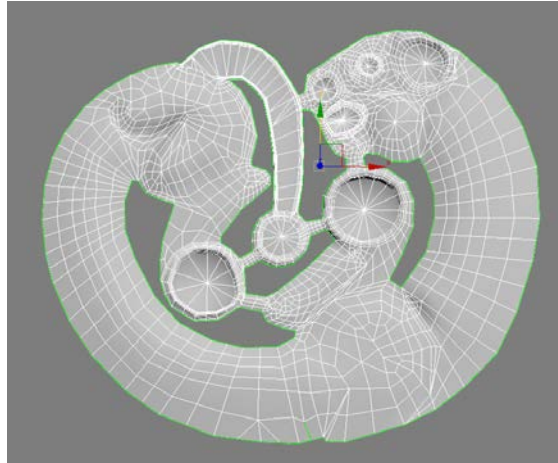


Рис. 3. Полигональная 3D модель

Моделирование проводилось на основе сплайнов Line (Линия), которые использовались для обводки контура исходного изображения. Далее объект был преобразован в EditPoly (Редактируемый многоугольник) и разбит на большое количество полигонов для более точного создания образа.

Для исследователя скифо-сибирского искусства важно выявление механизмов формообразования в древней изобразительной системе, отбора нужных характеристик для воспроизведения и перехода в изобразительном искусстве от реального трехмерного пространства к плоскостным графическим образам. Трансформация объемных объектов живой природы реального мира в визуализированные стилизованные идеограммы искусства звериного стиля представляет собой сложный процесс. Схематически этот процесс сводится к отбору человеческим мозгом признаков изображаемого реального объекта, необходимых для передачи характерных значимых особенностей зверя и достаточных для его опознания. Несущественные признаки отбрасываются. В то же время некоторые характеристики намеренно гиперболизируются и подчеркиваются, придавая изображениям животных специфические формы и образуя устойчивые изобразительные формулы, которые соответствуют определенным идеологическим задачам, семантике и содержанию символического по своей природе искусства кочевников.

Для хищников, в данном случае семейства кошачьих, подчеркиваются признаки, связанные с функцией их плотоядности и опасности: мощные челюсти, оскаленная пасть, когтистые лапы (в изделии могут быть предусмотрены области для инкрустации, например, бирюзой). Пластическая трактовка сильного тела животного основывается на обобщенной передаче мускулатуры с визуально рельефно выраженными зонами плечевого пояса и бедра. Гибкость пантеры часто утрируется удлинением пропорций средней части туловища.

Чтобы понять процесс формообразования и наглядно его представить – интересно «развернуть» имеющийся образ. Без применения компьютерных технологий это сделать сложно.

Для того чтобы манипулировать 3D моделью объекта, необходимо создать для него систему костей. Подобная система костей помещается внутрь модели тела пантеры (то есть имитирует скелет – рис. 4) и определяет последовательность и особенности связывания отдельных его элементов между собой. При этом каждая часть тела привязывается к своей косточке. За создание костей отвечает тип объектов Bones (Кости) категории Systems (Системы).

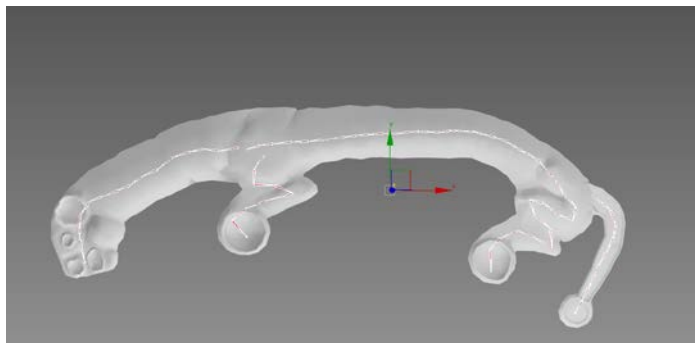


Рис. 4. Развернутая 3D модель со скелетом

Внедрение в модель структуры костей само по себе никак не скажется на поведении модели – необходимо обеспечить между ними взаимодействие, что возможно путем назначения модели модификатора Skin (Кожа), связанного с системой костей. Данный модификатор может воздействовать на модель, обеспечивая ее плавную деформацию с учетом смещения при движении костей.

На рис. 5 представлена трансформация от реального изображения животного до предмета искусства.



Рис. 5. Сопоставление фотографии животного с зооморфным образом скифо-сибирского стиля

Перейдем к описанию зооморфного образа оленя, который является навершием головного убора, выполненного в скифо-сибирском зверином стиле конца IV-го века до н.э, Налиньгаоту, уезд Шэньму, Провинция Шэньси (Soo, Bunker, 1995) [2]. Изображение украшения головного убора и результаты трехмерного моделирования зооморфного образа оленя приведены на рис. 6 и рис. 7 соответственно.



Рис. 6.Навершие головного убора, выполненное в скифо-сибирском зверином стиле



Рис. 7. 3D модель зооморфного образа оленя

Создание 3D модели образа оленя выполнено в Autodesk 3DsMax 2015 [3]. С помощью таких инструментов, как Lines (Линия) и Circles (Окружности), обводятся контуры фигуры, глаз, рогов и т. д. После того как контуры обведены, применяется инструмент EditableSpline (Редактируемый сплайн), который позволяет изменять форму моделируемых объектов (рис. 8,а).

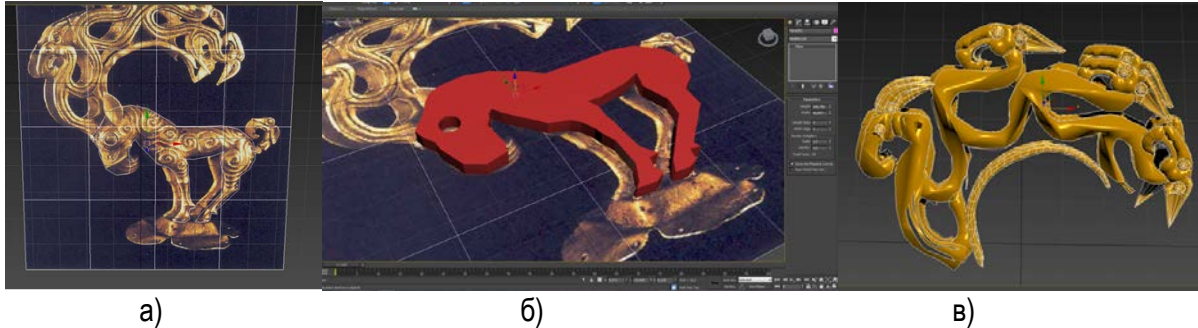


Рис. 8. Преобразование изображения зооморфного образа в 3D модель средствами пакета 3DsMax 2015.

Затем контуры изображений преобразованы в набор редактируемых полигонов (EditablePoly), т.е. представление модели на уровне вершин, ребер, граней и полигонов.

Применение модификатора Extrude (выдавливание) преобразует двухмерный сплайн в геометрическое тело, обладающее поверхностью (рис. 8,б). Модификатор Extrude придает сплайну (замкнутой невидимой кривой Безье) третье измерение. Завитки и ажурные элементы, которые украшают тело зооморфного образа оленя, моделируются при помощи модификации полигонов и вершин.

Для того чтобы сгладить фигуру (рис. 8,в), добавляются дополнительные точки и грани на полигональной поверхности и применяется модификатор Relax, который позволяет изменять неровности объекта путем сдвига вершин. Параметр RelaxValue контролирует, насколько далеко вершина сдвигается на каждой итерации. Iteractions устанавливает количество итераций, сколько раз операция будет осуществлена при тех же настройках. KeepBoundaryPtsFixed контролирует, сдвигаются ли вершины. SaveOuterCorners сохраняет оригинальные позиции вершин наиболее отдаленных от центра объекта.

Образ оленя в скифо-сибирском зверином стиле значительно отличается от реально существующих копытных животных (рис. 6, рис. 9,б). Рога мифологического животного развернуты фронтально в горизонтальной плоскости, ритмичный ряд симметрично расположенных отростков включает скомпонованные головы грифонов. Ноги животного прямые, как бы свисают с туловища, концы копыт направлены вниз. Своеобразна не только поза животного, но и необычно большие круглые глаза, со сквозным отверстием и хвост в виде стилизованной головы ушастой птицы.

Следует отметить, что все изображения зооморфных образов, по преимуществу – копытных животных, прежде всего оленей, а также кошачьих хищников и птиц, имели в глазах скифов некое важное содержание, но вопрос о семантике скифского звериного стиля до сих пор составляет предмет дискуссий [1, 2, 4].

Считается, что изделия древних мастеров «звериного стиля» имели магические функции, были призваны обеспечить их владельцу удачу, силу, быстроту, зоркость – те качества, которые присущи изображаемому животным. Например, грифон – это символ силы и бдительности, а спираль является символом времени, циклических ритмов, смены сезонов года, рождения и смерти, фаз «старения» и «роста» Луны, а также самого Солнца [1].

В современном мире исторические реконструкции (рис. 9,а) широко используются в программах семинаров, конференций и праздников, посвященных культурным традициям



народов для демонстрации традиционных обрядов и ритуалов, поддерживая связь времен и поколений.



Рис. 9. Сравнение фотографии коня в ритуальном уборе и 3D модели зооморфного образа оленя

В заключение следует отметить, что с применением технологий 3D моделирования, появляется новый метод изучения процесса, важного для исследователя древней культуры, перехода от реального мира к стилизованным идеограммам искусства звериного стиля. Пошаговая 3D визуализация позволяет проследить процесс формирования сложного изображения зооморфного образа из отдельных частей, поскольку каждый из них состоит из нескольких образов фантастических существ. Компьютерные реконструкции находок из скифских курганов и поселений VIII в. до н.э. – III в. н.э. VII помогут в исследовании и определении характерных признаков зооморфных образов, а также могут быть представлены в качестве экспонатов в музейных экспозициях.

#### Библиографический список

1. Королькова Е. Ф. "Звериный стиль Евразии. Искусство племен Нижнего Поволжья и Южного Приуралья в скифскую эпоху (VII–IV вв. до н.э.)" – Санкт-Петербург, издательство "Петербургское Востоковедение", 2006 г.
2. Emma C. Bunker with contributions by James C. Y. Watt and Zhixin Sun "Nomadic Art of the Eastern Eurasian Steppes. The Eugene V. Thaw and other New York collections" – The Metropolitan Museum of Art, New York; Yale University, New Haven, London.
3. Справочник Autodesk 3dsMax [Электронный ресурс] – Режим доступа: <http://docs.autodesk.com/3DSMAX/15/ENU/3ds-Max-Help/index.html>, свободный.
4. Минасян Р. С. "Металлообработка в древности и средневековье" – Санкт-Петербург, издательство Государственного Эрмитажа.

УДК 004.514, 004.921

**А. Ю. Меликян, А. Н. Исаяев**

студенты кафедры вычислительных систем и сетей

**Д. А. Булгаков**

научный руководитель

**Е. Ф. Королькова**

канд. искусствоведения, зав. сектором отдела археологии Восточной Европы и Сибири Государственного Эрмитажа – научный консультант

#### ПРЕДСТАВЛЕНИЕ ЭЛЕМЕНТОВ УПРАВЛЕНИЯ В ИНТЕРАКТИВНЫХ ПРИЛОЖЕНИЯХ НА ПРИМЕРЕ РЕКОНСТРУКЦИИ ОБРАЗОВ СКИФО-СИБИРСКОГО ЗВЕРИНОГО СТИЛЯ

Скифо-сибирский звериный стиль остается одним из самых интересных и загадочных явлений древней художественной культуры как в аспекте содержания, так и с точки зрения анализа

формы. Проблема происхождения искусства звериного стиля на протяжении долгих лет заставляет исследователей обращаться к разным культурным регионам древнего мира в поисках прототипов и истоков самобытного зооморфного изобразительного творчества, ставшего одной из главных особенностей культур ранних кочевников Евразии в I тыс. до н. э. [1, 2].

Процесс создания виртуальной исторической реконструкции условно можно поделить на несколько этапов [3].

- первый этап – анализ источников и, по возможности, первоисточников исторической информации;
- второй этап – верификация данных;
- третий этап – анализ археологических данных с помощью компьютерных технологий моделирования;
- четвертый этап – повышение реалистичности построенной модели исторической реконструкции;
- пятый этап – разработка пользовательского интерфейса и интерактивности.

В результате интерактивная виртуальная реконструкция становится одновременно пространством экспериментов и репрезентативным пространством. В ходе создания трехмерных моделей исторических реконструкций возможно проведение различных экспериментов, с целью проверки научных гипотез.

Остановимся на этапе разработки пользовательского интерфейса подробнее.

Для изучения скифо-сибирского стиля было создано интерактивное приложение, которое, благодаря простому и доступному интерфейсу (рис. 1), может заинтересовать как молодое, так и старшее поколение. Интерактивность в приложении подразумевает, что пользователь может не только наблюдать за происходящим, но и управлять им.



Рис. 1. Схема интерфейса приложения

Разработка интерактивного приложения производилась на игровом движке Unity 3D. Все необходимые модели, созданные в 3DsMax, экспортировались в формате fbx (File>Export), при этом сохранялись все материалы и текстуры [4].

Для анимации и взаимодействия с объектами были написаны скрипты, для написания скриптов использовался язык программирования C#.

На главном меню расположено название проекта «Скифо-сибирский звериный стиль» (рис. 2). Под ним находится навигационная панель с кнопками, представленными в виде соответствующих образов. Каждый экран, помимо главного меню, снабжен основными элементами навигации: переход на предыдущий уровень навигации, возврат в главное меню (кнопка «домой»). Некоторые экраны, в случае необходимости, содержат дополнительные элементы навигации (например, кнопка «назад»).

На экране «Определение скифо-сибирского звериного стиля» (рис. 3) расположен текст, раскрывающий понятие и характерные черты скифо-сибирского звериного стиля, а также дающий представление о соответствующей исторической эпохе.



Рис. 2. Главное меню

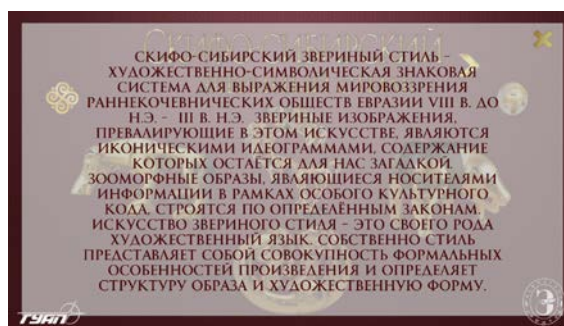


Рис. 3. Определение скифо-сибирского звериного стиля

Экран «Мировоззрение скифов» (рис. 4) знакомит пользователя с мировоззрением скифов, основой которого являлась концепция трех миров: верхнего, среднего и нижнего. Каждый мир представлен изображением олицетворяющего его животного: птицы, оленя и пантеры соответственно. Изображения размещены в левой части экрана, правый же занимает описание миров.

Экран «Карта» (рис. 5) представляет собой интерактивную карту Евразии настоящего времени. На нее нанесены топонимы разного цвета, обозначая крупнейшие города и реки, топонимы интересующего нас исторического периода и названия обнаруженных стоянок, крупных захоронений и кладов (лишь тех, о которых идет речь в нашей программе). Каждый памятник обозначает иконка – изображение предмета, найденного там и ставшего символом той или иной страницы в историографии скифо-сибирского звериного стиля. Каждая иконка является интерактивной и при двойном клике на нее ведет на страницу с фотографиями раскопок того или иного памятника, ежели они проводились.



Рис. 4. Мировоззрение скифов



Рис. 5. Интерактивная карта Евразии настоящего времени

Фотографии моделей, расположенных на главном меню, содержат изображения четырех артефактов, которые служат ссылками на экраны, содержащие их 3D модели. Артефакты представляют важнейшие звериные образы скифо-сибирского искусства: изображения оленя, волка, медведя и пантеры. При нажатии на изображение осуществляется переход к сцене с 3D моделью выбранного животного, например волка (рис. 6).



Рис. 6. Интерактивная 3D-модель волка

Использование компьютерных технологий при создании исторических реконструкций позволяет добиться нового уровня передачи информации и, как следствие, нового уровня восприятия, что приближает нас еще на шаг к восстановлению картины мира кочевников.

Таким образом, было разработано интерактивное приложение, содержащее трехмерные модели зооморфных образов животных. Подобное приложение с трехмерной визуализацией модели дает возможность решать целый комплекс научных и практических задач, стоящих перед научными работниками, хранителями музеев и другими специалистами. Появляется уникальная возможность показывать музейные экспонаты в другом месте, не вывозя их. Можно дополнять музейную выставку экспонатами из других музеев, а также виртуально реконструированными предметами, утерянными в веках.

Однако основная цель данного приложения – позволить посетителям музея увидеть артефакты так, как это невозможно было сделать на обычных выставках. 3D моделирование позволяет повернуть и осмотреть музейные артефакты с разных углов, что привлечет внимание посетителей и поможет увеличить выставочные возможности музея. Благодаря использованию редактора Unity3D сама технология разработки может быть реализована на различных платформах – как на персональных компьютерах с различными операционными системами, так и на мобильных устройствах.

Данное приложение предполагается применять на интерактивных выставочных стендах Государственного Эрмитажа.

### Библиографический список

1. Королькова Е. Ф. Звериный стиль Евразии. Искусство племен нижнего Поволжья и Южного Приуралья в скифскую эпоху (VII–IV вв. до н.э.) – Санкт-Петербург: Петербургское Востоковедение, 2006. – 272 с.
2. Золотые олени Евразии: Каталог выставки в Государственном Эрмитаже (СПб., 18.X.2001 г. – 20.I.2002 г.). – СПб.: Славия, 2001. – 248 с.
3. Бородин Л. И., Жеребятёв Д. И. Технологии 3D-моделирования в исторических исследованиях: от визуализации к аналитике // Историческая информатика. – 2012. – № 2. – С. 49–63.
4. UnityManual (Уроки) [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.docs.unity3d.com/Manual/index.html>, свободный. – Загл. с экрана. – Яз.: русск., англ., кит., исп. (10.04.2015).

**Р. А. Мулеронко**

студент кафедры вычислительных систем и сетей

**Н. Н. Решетникова**

канд. техн. наук, доц. – научный руководитель

**М. Е. Тихомиров**

доц. – научный консультант

## ОСОБЕННОСТИ СОЗДАНИЯ ТРЕХМЕРНОГО ПРИЛОЖЕНИЯ В BLENDER

Трехмерное представление информации является наиболее привычным и наглядным для человека. В связи с этим 3D моделирование незаменимо для презентаций, выставок, а так же в том случае, когда надо показать, каким будет итоговый результат моделирования.

Blender – свободный, профессиональный пакет для создания трехмерной компьютерной графики, включающий в себя средства моделирования, анимации, визуализации, пост обработки и монтажа видео со звуком, а также для создания интерактивных приложений.

Процесс разработки интерактивных трехмерных приложений можно разделить на три этапа: создание 3D моделей для проекта; создание интерфейса и 3D сцен; создание интерактивной части. Рис. 1 показывает поэтапно процесс создания трехмерных приложений: стандартно используемые программы и альтернативный вариант.

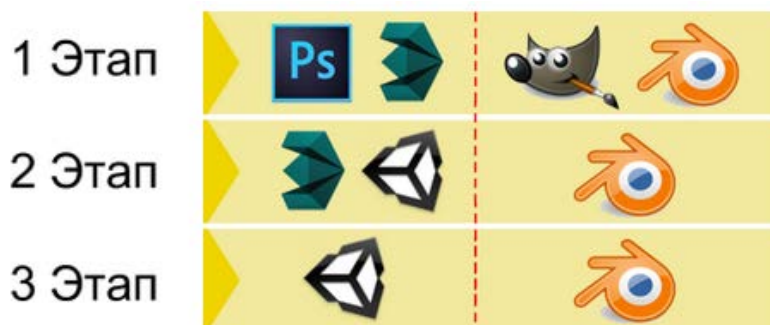


Рис. 1. Стандартный и альтернативный процесс создания интерактивного трехмерного приложения

Если рассматривать стандартный процесс со стороны использования сред разработки, то разработчики используют для каждого этапа разные программные средства, что приводит к появлению проблем: экспортирование моделей, плохая поддержка форматов, следствие из этого: полигональные коллизии, ошибочное наложение текстур и материалов.

Главной особенностью программы Blender является объединение всего цикла разработки трехмерных приложений, что решает проблемы, с которыми обычно сталкиваются разработчики, использующие стандартный процесс разработки.

Создание трехмерных моделей в программе Blender, схоже с аналогичными программами, поэтому в статье подробнее поговорим о процессе создания интерфейса и разработке интерактивной части.

Рассмотрим особенности создания трехмерного приложения на примере интерактивной 3D модели истребителя СУ-30 в программе Blender.

Интерактивное приложение – набор сцен, связанных между собой, с которыми взаимодействует пользователь. Перед тем как начинать непосредственно его программирование, необходимо понять, как пользователь будет взаимодействовать с приложением, рис. 2.

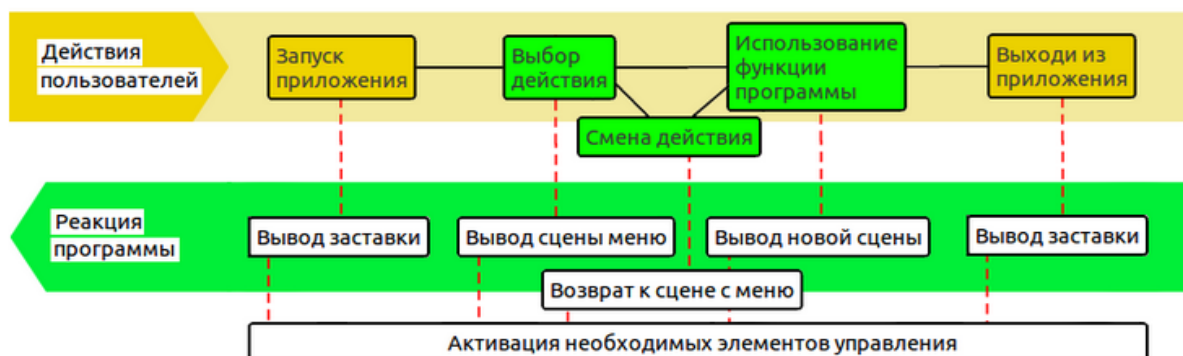


Рис. 2. Структура диалога «Действие – Реакция»

По структуре диалога, можно понять логику работы приложения и необходимое количество сцен. Каждая сцена имеет интерактивные элементы, с которыми взаимодействует пользователь на протяжении всего цикла работы с приложением.

На рис. 3 представлена общая структура приложения с элементами интерфейса. Интерактивное приложение, включающее в себя модель истребителя СУ-30, состоит из 15 сцен.

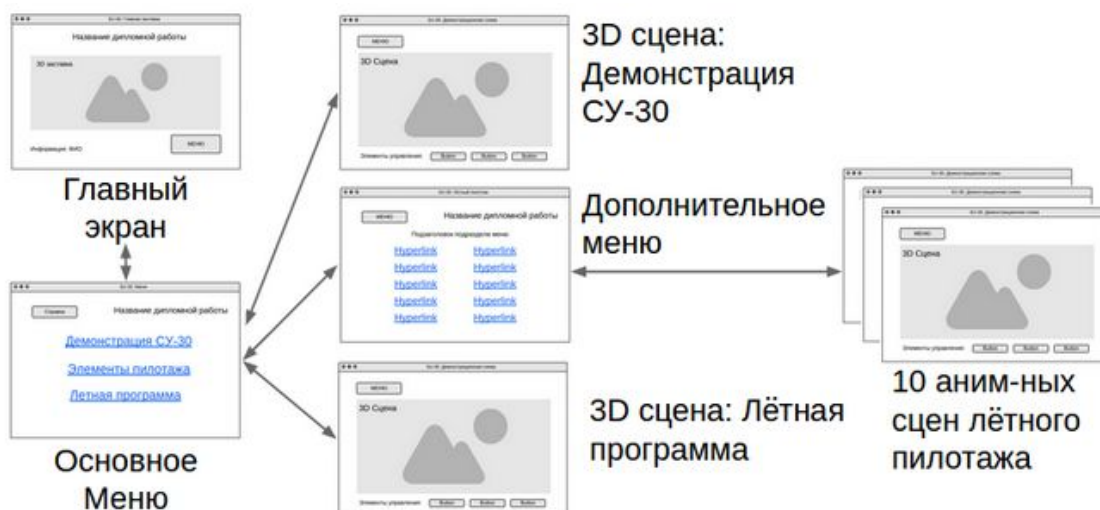


Рис. 3. Общая структура приложения с элементами интерфейса

Главный экран – анимационная заставка приложения; Основное меню – выбор взаимодействия с моделью истребителя СУ-30; Дополнительное меню – выбор элемента пилотажа; 3D сцены – 12 сцен с трехмерной моделью самолета, демонстрирующих элементы высшего пилотажа.

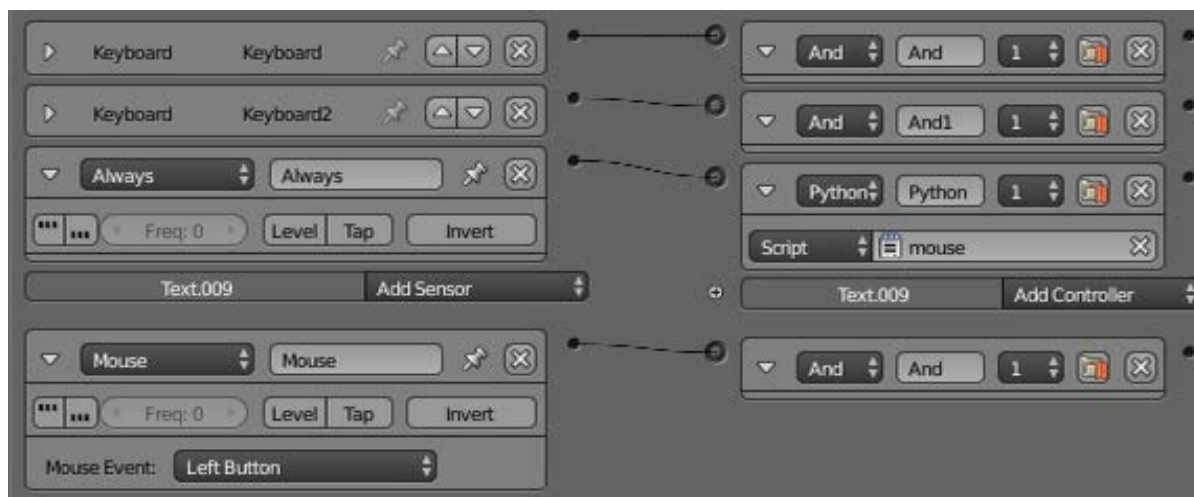


Рис. 4. Панель LogicEditor

На рис. 4 представлена панель LogicEditor. Она используется для разработки интерактивной части в программе Blender, включает в себе объединение визуального и стандартного программирования [1].

Логика визуального языка программирования в BGE (BlenderGameEngine), заключается в следующем, имеется три типа блоков: сенсоры, контроллеры, актуаторы. Каждый блок решает определенные задачи.

Сенсор реагирует на различные виды входных данных. Это могут быть нажатия комбинаций клавиш, кнопок джойстика или события таймера, которые поступают при каждом обновлении экрана приложения.

Контроллеры используются для связи сенсоров с актуаторами. Они дают более полный контроль над тем, как сенсоры и актуаторы взаимодействуют друг с другом.

Актуатор передает параметры изменения модели самолета и выполняет действия в приложении. Это может быть перемещение объекта в пределах сцены, воспроизведение анимации, или проигрывание звука и т. д.

Логические блоки связываются друг с другом путем их визуального соединения. Сенсоры можно соединять только с контроллерами, контроллеры – только с актуаторами. К одному контроллеру можно присоединить несколько сенсоров и несколько актуаторов. Когда все необходимые блоки объединены друг с другом, получается карта действий сцены. Рассмотрим карту для 3D сцены – демонстрация истребителя СУ-30. На сцене имеются интерактивные элементы: СУ-30 на пьедестале, кнопка меню (переход в основное меню). На рис. 5 представлена сцена, на которой можно посмотреть на самолет с различных точек зрения, а также карта действий по сцене.

Рассмотрим детально карту действий. В первой колонке используются следующие сенсоры: 2 шт. – клавиатура (keyboard), всегда (always), мышь (mouse). Во второй колонке используются следующие контроллеры: 3 шт. – и (and), питон (Python). В третьей колонке используются следующие активаторы: 2 шт. – движение (motion), сцена (scene).

Рассмотрим первую строчку карты действий. Сенсор клавиатура срабатывает по нажатию клавиши стрелки на клавиатуре. Далее, сигнал с сенсора поступает на контроллер «и». «И» – простая логическая функция, аналогичная логическому умножению. Далее сигнал проходит на актуатор движения. Актуатор начинает изменять угол объекта по оси Z. И самолет начинает вращаться вокруг своей оси.

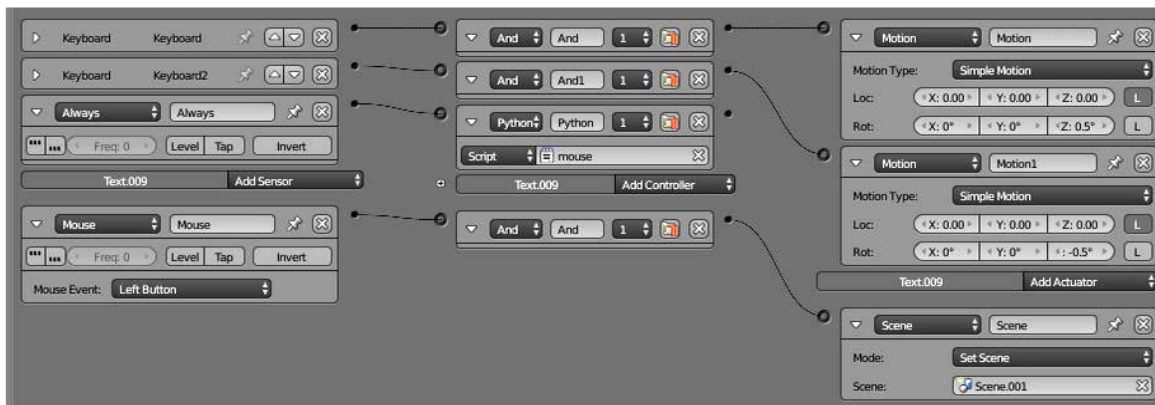


Рис. 5. 3D сцена: демонстрация истребителя СУ-30 и карта действий

Помимо визуального программирования, Blender поддерживает высокоуровневое программирование на языке Python [2, 3]. Для добавления скрипта необходимо воспользоваться специальным контроллером Python Scripts. Пример подключения скрипта приведен на рис. 6 – пример скрипта активации курсора мыши на сцене.



Рис. 6. Подключение скрипта Python к сцене



Для перехода в режим запуска и тестирования приложения необходимо нажать на клавиатуре кнопку Р или в панели Render нажать на кнопку Start. Программа запустится в новом окне. Можно начинать тестирование и отладку.

Из выше сказанного можно сделать вывод, что на данный момент Blender является свободно распространяемой программой и наиболее гибкой средой создания 3D моделей и интерактивных приложений. Разработанное приложение «Интерактивная модель многоцелевого истребителя СУ-30» можно использовать в демонстрационных и учебных целях. Разработанное приложение может быть использовано на разных платформах: Windows, OSX, Linux.

### Библиографический список

1. Справка по написанию скриптов [Электронный ресурс]. – Режим доступа: [www.wiki.blender.org](http://www.wiki.blender.org)
2. Лекции по Python [Электронный ресурс]. – Режим доступа: [www.ideafix.name](http://www.ideafix.name)
3. Michel Anders Blender 2.49 Scripting / Michel Anders - Packt Publishing, 2014, 292 с.

УДК 681.3.06

**Т. В. Нагорный**

студент кафедры компьютерной математики и программирования

**М. Ю. Охтилев**

д-р техн. наук, проф. – научный руководитель

### ГЕНЕРАЦИЯ ТЕКСТУРЫ 3D МНЕМОСХЕМ В АСУ КОСМИЧЕСКИМИ СРЕДСТВАМИ НА ОСНОВЕ МАЛОЙ КАРТИНКИ ДЛЯ ОТКРЫТЫХ ЛАНДШАФТОВ

С резким скачком производительности ЭВМ и развитием средств трехмерного моделирования 3D технологии интегрировались в различные сферы человеческой деятельности, к примеру: в машиностроение, строительство, космическую отрасль, медицину, рекламу, киноиндустрию.

Мониторинг космических объектов предполагает их первоначальное моделирование. Воссозданию подлежат все объекты и локации мест запуска. Очень важным моментом является текстурирование всей сцены. Это необходимо, так как покрытая текстурой модель выглядит более реалистичной, а окружение более естественным. Моделирование сцены проводится в пропорциях, близких к реальным, в результате чего получают огромные открытые пространства, на которые необходимо качественным образом наложить текстуру. Для этого необходимо наличие самого изображения-текстуры. Это, как правило, внешний файл-картинка. Для больших объектов необходимы текстуры соответствующих размеров. Если это условие не соблюдается, изображение подвергается масштабированию, что ведет к потере качества конечного результата. Или же дублирование самого изображения, что сохраняет качество, но создает швы на стыке текстур. Решение этой задачи достаточно тривиальное – необходимо использовать подходящее изображение нужного размера, но проблема заключается в том, что зачастую текстур таких размеров просто не существуют. Цель работы – это поиск решения и его реализация. Другими словами, необходимо из маленького изображения получить бесшовную и качественную текстуру заданного размера.

Данная тема является актуальной по ряду причин. Имея рабочий и быстрый алгоритм формирования текстуры искомых размеров, можно сократить время, затраченное на поиск подходящего изображения. Самым главным моментом является ситуация полного отсутствия текстуры нужного размера, которая решается ее генерированием на основе малой картинки, которая сама по себе не подходит для текстурирования.

При работе с открытыми пространствами также используется метод текстурирования поверхности в режиме реального времени в зоне видимости точки обзора. Это значительно уменьшает количество используемой оперативной памяти для хранения целой текстуры.

Рассматриваемая проблема является не самой распространенной, но, тем не менее, существуют несколько разработанных моделей, решающих похожие задачи. Наиболее подходящим вариантом является комбинирование методов «Кройка изображений» [1] и «Плитки Вонга» [2].

Первая модель может выдавать конечный необходимый результат сам по себе, но это достаточно ресурсоемкий процесс, который может занять много времени, в зависимости от размеров искомой картинки. Однако, использование некоторых методов этого алгоритма в связке с «Плитками Вонга» позволяет добиться формирования бесшовного изображения и значительного сокращения времени генерации текстуры.

Как было сказано выше, для формирования итогового результата используется комбинация из двух алгоритмов. Базовый метод «Кройка изображений» и вспомогательный метод «Плитки Вонга».

#### Алгоритм «Плитки Вонга»

Финальное изображение строится путем комбинаций из восьми плиток (в некоторых вариантах – 12 или 18) между собой. Одна плитка состоит из объединенных четырех блоков, повернутых на 45 градусов по часовой стрелке. Комбинации этих четырех блоков уникальны для каждой плитки. Схематично, каждую плитку можно рассмотреть как изображение с индивидуальным «цветом» ребра (рис. 1, рис. 2).

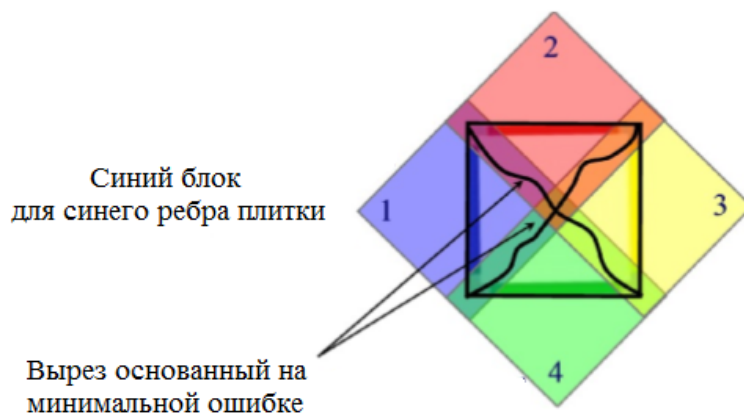


Рис. 1. Структура одной схематичной плитки Вонга

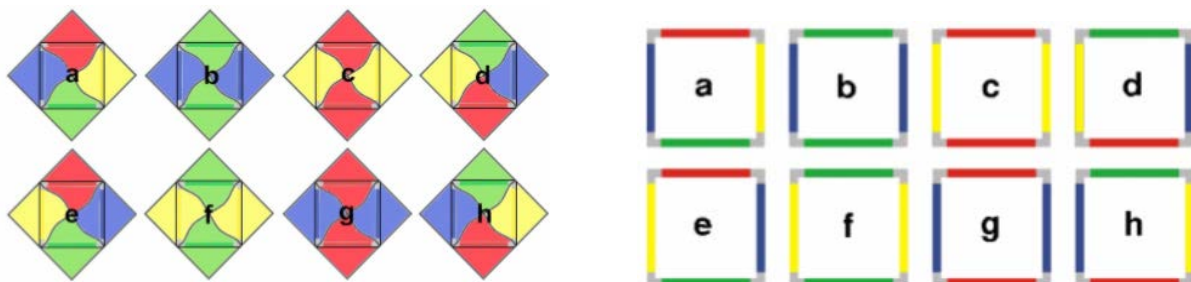


Рис. 2. Восемь схематичных плиток Вонга

Таким образом, финальное изображение генерируется путем произвольных комбинаций плиток с единственным условием равенства «цвета» ребра – при каждой следующей подстановке (рис. 3).

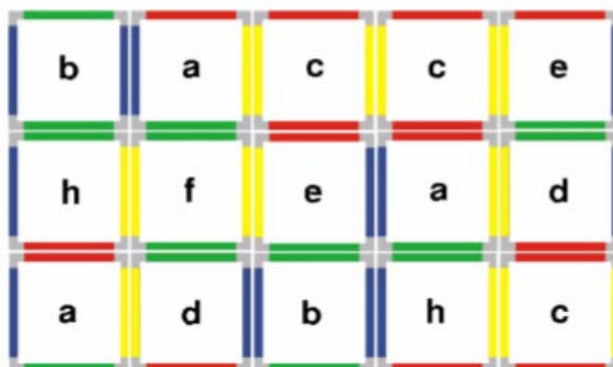


Рис. 3. Схематичный пример сгенерированной текстуры

Как было сказано выше, для формирования одной плитки Вонга, используются четыре изображения. Для корректного смешивания границ каждого из этих изображений используются методы из алгоритма «Кройка изображения».

Метод смешивания границы алгоритма «Кройка изображения»

На вход подается небольшое изображение. Из него, произвольным образом извлекаются четыре блока. Схематично – синий, красный, желтый, зеленый. Формируется плитка Вонга, с соответствующей ей комбинацией цветов. На границе цветов происходит наложение блоков друг на друга с заданной глубиной. Анализируется каждое наложение из четырех в отдельности. Вычитается значение каждого соответствующего пикселя в наложении, таким образом, что пиксели, показавшие минимальную накопленную ошибку, формируют так называемый путь – «Вырез, основанный на минимальной ошибке» (рис. 4).



Рис. 3. Пример реальной плитки Вонга

Результатом работы является искомая текстура заданных размеров и высокого качества. Связка из двух методов позволяет экономить время генерирования изображения, а также обеспечивать бесшовность конечного рисунка.

**Библиографический список**

1. Efros Alexei A., Freeman William T. Image quilting for texture synthesis and transfer // SIGGRAPH '01 Proceedings of the 28th annual conference on Computer graphics and interactive techniques, p. 341–346.
2. Cohen Michael F., Shade Jonathan, Hiller Stefan, Deussen Oliver. Wang Tiles for Image and Texture Generation // SIGGRAPH '03 ACM SIGGRAPH 2003 Papers, p. 287–294.

УДК 004.4

**И. В. Нестерук**

студент кафедры компьютерной математики и программирования

**В. В. Бураков**

д-р техн. наук, доц. – научный руководитель

**МОДЕЛЬ АКТОРОВ**

В начале 70-х годов прошлого века разработчики компьютерных систем верили, что машины следующего поколения будут многопроцессорными, а программы обладать искусственным интеллектом. В этой связи возникла задача обобщить понятие вычислимости на параллельных вычислениях [1].

Теория вычислимости является разделом математики, который находится на стыке математической логики, теории алгоритмов и информатики, возникший в результате изучения понятий вычислимости и не вычислимости функции (в современной интерпретации – доказуемости или недоказуемости). Обычно понятие вычислимости удобнее всего определить через машину Тьюринга или Поста [2].

У машины есть состояние – совокупность значений всех ячеек ленты. Процесс вычислений представляет собой последовательность шагов машины, каждый из которых меняет ее состояние. Каждый шаг машины – выполнение одной атомарной операции. Выполнение вычислений в такой трактовке приводит к появлению глобального времени, когда в один и тот же момент времени может выполняться одна и только одна атомарная операция [3].

При переходе к многопоточным системам, когда несколько параллельных потоков выполнения команд разделяют общую память, требование к глобальному времени не будет выполняться. Кроме того, разработчики параллельных и многопоточных систем столкнулись с таким явлением как «состояние гонки» (racecondition), когда работа программы зависит от того, в какой последовательности выполняются участки кода в параллельной системе. Чтобы избежать подобной проблемы стали использовать различные механизмы синхронизации параллельных вычислений (мьютексы, семафоры, мониторы и т.д.). Очевидным недостатком механизмов синхронизации является установка и снятие блокировок с критических секций программы, которая в значительной степени снижает общую производительность системы. Неправильное же использование механизмов синхронизации грозит появлению тупиковых ситуаций (deadlock) [4].

Альтернативным подходом в применении вычислимости в параллельных и многопоточных системах является модель акторов.

Актор – это универсальная абстракция вычислительной сущности, которая в ответ на получаемое сообщение может выполнить одно из следующих действий:

- отправить конечное число сообщений другим акторам;
- создать конечное число новых акторов;
- выбрать тип поведения, которое будет использоваться для следующего сообщения в свой адрес [5].

Все действия могут выполняться для каждого актора в произвольном порядке, а также все эти действия могут выполняться параллельно. Схема модели акторов изображена на рисунке.

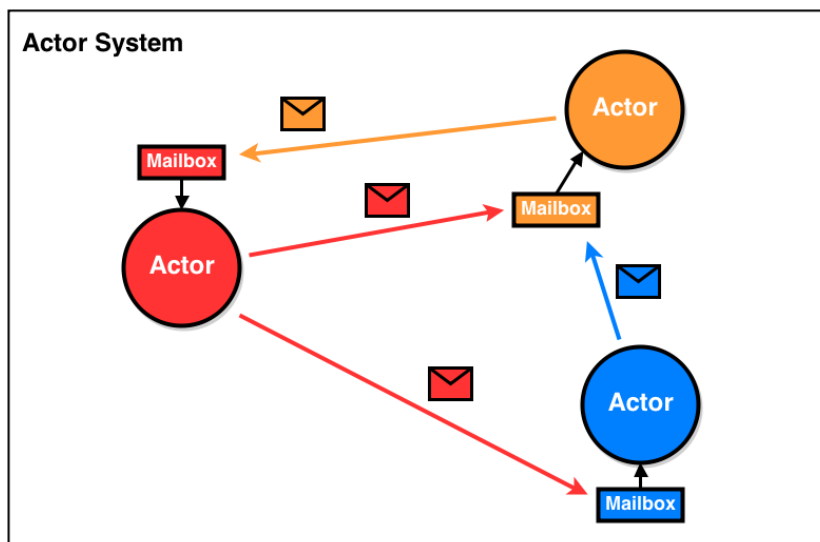


Рис. Схема модели акторов

На рисунке получатели сообщений определяются по адресу, который также называется «почтовым». Актор в данной модели может взаимодействовать только с теми акторами, адреса которых он знает. Адреса могут быть известны заранее (свой адрес, адрес «детей» и непосредственного родителя) либо извлечены из полученных сообщений.

Классическую модель параллельных вычислений можно рассмотреть на примере телефонного звонка. Если несколько человек попытаются дозвониться на один и тот же телефонный номер, то они начинают конкурировать между собой за возможность соединиться с абонентом (классическая ситуация появления состояния гонки потоков в параллельных вычислениях). Чтобы избежать подобных проблем, применяют многоканальные телефоны, собственные АТС и т. д. (в терминах параллельного программирования – это механизмы синхронизации) [1].

Работа же акторов больше похожа на отправку электронных сообщений. Каждый отправитель (актор) знает своего адресата, которому можно отправить сообщение. Каждый принимающий (уже другой актор) также может отправить сообщение и т. д. Применение такого подхода при построении модели акторов позволяет избавиться от задержек, которые присущи параллельным вычислениям с использованием механизмов синхронизации [6].

Модель акторов в настоящее время поддерживается большим числом языков программирования. Языками программирования со встроенной моделью акторов являются: Erlang, Go и многие другие.

В тех же языках программирования, где изначально не была реализована модель акторов, используют различные библиотеки: Akka для языков Scala и Java, Haskell-Actor используют в языке Haskell, Akka.NET имплементация идей Akka в языке C#, Celluloid для языка Ruby и др. [1].

Для наглядности использования модели акторов приведем пример написанный на языке программирования Scala с использованием библиотеки Akka. В данном примере один из акторов будет принимать имена адресатов, а другой категорию сообщения.

```
import akka.actor.{Actor, ActorSystem, Props}

object Main extends App {
  // создание корневого актора
  val system = ActorSystem("SimpleMail")
  val name = system.actorOf(Props(Name), name = "name")
  name ! "Bob"
  for(i <- 1 to 10) {
    println("Do task" + i)
  }
  name ! "Tom"
  name ! "Alice"
  // Актор принимающий имя получателя
  class Name extends Actor {
    def receive = {
      case "Bob" => {
        val subject = context.actorOf(Props(new Subject("Hi Bob!")), name = "subject")
        println("Hi Bob!")
        subject ! "Job"
      }
      case "Alice"=> {
        println("Hi Alice!")
        // Создаем актор потомок
        val subject2 = context.actorOf(Props(new Subject("Hi Alice")), name = "subject2")
        subject2 ! "Weekend"
      }
      case _ => println("Hi Unknown!")
    }
  }
  // Актор принимающий тему сообщения и приветствие
  class Subject(name: String) extends Actor {
    def receive = {
      case "Job" => println("New Idea about new project - " + name)
      case "Weekend" => println("Plan at weekend - " + name)
    }
  }
}
```

Скомпилируем и запустим программу на исполнение несколько раз.

```
Hi Bob!
Do task1
Do task2
New Idea about new project - Hi Bob!
Do task3
Do task4
Do task5
Do task6
Do task7
Do task8
Do task9
Do task10
Hi Unknown!
Hi Alice!
Plan at weekend - Hi Alice
```

```
Hi Bob!
Do task1
Do task2
Do task3
Do task4
Do task5
Do task6
New Idea about new project - Hi Bob!
Do task7
Do task8
Do task9
Do task10
Hi Unknown!
Hi Alice!
Plan at weekend - Hi Alice
```

Как видно из примера, акторы при получении и обработке сообщения не блокируют дальнейший ход вычислений. Вместо этого акторы выполняются параллельно и независимо как друг от друга, так и основного потока.

Моделью акторов активно занимаются два специалиста – Carl Hewitt и его ученик Gul A. Agha. Hewitt, в основном, занимается строгим обоснованием того, что другие подходы к вычислимости являются частным случаем акторной модели, а Agha – различными приложениями для распределенных систем.

В 1977 г. Carl Hewitt и Henry Baker в статье *Actors and Continuous Functionals* обосновали, что модель акторов пригодна для использования на практике. В 1985 г. Gul A. Agha под руководством Hewitt защитил диссертацию *Actors: A Model Of Concurrent Computations in Distributed Systems*. В диссертации Agha описывается синтаксис минимального языка программирования, поддерживающего акторы, а также набор типовых задач и приемов их решения.

В статье Phillip Haller и Martin Odersky «Event-Based Programming with Inversion Control» был предложен новый подход в разработке акторных систем, а именно написание параллельной программы в обычном последовательном стиле [1].

В настоящее время модель акторов, по мнению Carl Hewitt, ставит следующие актуальные вопросы в области компьютерных систем, языков параллельного программирования, веб-сервисов и архитектур связи:

- масштабируемость. Связан с расширением локального и нелокального параллелизма;
- прозрачность. Устранение пропасти между локальными и нелокальным параллелизмом.

В настоящее время некоторые исследователи придерживаются строго разделения модели локального и нелокального параллелизма. Такое деление приводит к проблемам необходимости внесения изменений в оба метода параллелизма, при изменении одного из них;

- противоречивость. Все очень большие системы знаний о взаимодействии информационных систем человечества являются противоречивыми [7].

Модель акторов находит широкое применение в облачных вычислениях, а также в многоагентных системах, которые дополнительно накладывают ограничения на взаимодействие акторов, требуя, чтобы они использовали обязательства и цели [1].

Таким образом с помощью модели акторов, в отличие от многопоточных систем, основанных на синхронизации, можно не только согласовать использование общих вычислительных ресурсов разными потоками, но и осуществить это наименее чувствительным образом, с точки зрения производительности.

### Библиографический список

1. Habrahabr: Модели акторов 40 лет! URL: <http://habrahabr.ru/company/tiktokcoach/blog/206300>.
2. Хопкрофт Д., Мотвани Р., Ульман Д. Введение в теорию автоматов, языков и вычислений: пер. с англ. / Д. Хопкрофт, Р. Мотвани, Д. Ульман. М.: Вильямс, 2002, 528 с.
3. Habrahabr: Машина Тьюринга на формулах Excel, URL: <http://habrahabr.ru/post/189064/>
4. The actors model guide: Глава 1. Акторы. Начало., URL: <http://actorsmodel.ru/glava-1-actori-nachalo/>
5. Stan'swork: О моделях актеров, URL: [http://stanonwork.blogspot.ru/2007/07/blog-post\\_18.html](http://stanonwork.blogspot.ru/2007/07/blog-post_18.html)
6. Msdn: Программная модель актеров, URL: <http://blogs.msdn.com/b/ru-hpc/archive/2009/11/25/9928475.aspx>
7. Scriptorg: SCALA Report, URL: <http://scriptogr.am/dennis-groves/post/scala-report>.

УДК 372.8; 004.75

**А. А. Никифорова**

магистрант кафедры проблемно-ориентированных вычислительных комплексов

**В. С. Павлов**

д-р техн. наук, проф. – научный руководитель

### ВЕБ-ОРИЕНТИРОВАННЫЕ МЕДИАТЕХНОЛОГИИ В МУЗЫКАЛЬНОМ ОБРАЗОВАНИИ

Рассматривается специфика организации образовательного пространства [1] для студентов, преподавателей музыкальных образовательных учреждений, базирующаяся на использова-

нии веб-ориентированных медиатехнологий обучения. Обосновывается эффективность внедрения данных технологий в образовательный процесс и раскрывается их наиболее важная техническая сторона, а именно – способы и форматы представления мультимедийных учебных материалов, значительное место в которых занимают нотные записи.

Главным направлением современного этапа информатизации образования является всеобщее стремление к интеграции различных компьютерных средств обучения, таких как электронные справочники, энциклопедии, обучающие программы, средства автоматизированного контроля знаний обучаемых, компьютерные учебники и тренажеры, в единые программно-методические комплексы.

Музыканты и учащиеся музыкальных учреждений имеют потребность в веб-ресурсах, которые могли бы обеспечить уверенные шаги по ступеням музыкальной карьеры. Другими словами – любой музыкант заинтересован в удобстве, компактности, а так же быстром поиске обучающего материала, нотных партитур.

До сих пор представление нотной литературы в режиме реального времени без предварительного сохранения на устройство, не представлялось возможным, или попросту не имело полной реализации. Веб-ресурс, содержащий нотные партитуры, готовые к просмотру и использованию, в режиме реального времени, актуален в любом музыкальном учебном заведении, ведь использование информационных технологий [2] в обучении приняло массовый характер.

Общепринятые методы музыкального образования основываются на определенных циклических процессах (рис. 1). В первую очередь учащимся музыкального учреждения предлагают ознакомиться с материалом для последующей практической работы над ним. Материалы предоставляются в печатной форме, а так же проигрываются преподавателем. Полученные таким образом практические задания активно используются для достижения наилучшего результата. Самостоятельный способ выполнения заданий является обязательным, так как основной частью образования в музыкальном учреждении является самоанализ, самообразование и готовность к восприятию. Посредством самоанализа учащийся выдвигает гипотезу произведения, самосовершенствуется и корректирует результаты. Подтверждение гипотезы теоретическими знаниями облегчает понимание материала, а рекомендации, данные преподавателем, формируют уверенность и готовность к музыкальной деятельности.

Залогом самосовершенствования в музыке является следование нововведениям, своевременное обновление имеющихся теоретических знаний, а так же постоянная практика. Общепринятые принципы музыкального образования вполне могут быть реализуемы и в медиасреде.



Рис. 1. Общепринятые методы музыкального образования



Проанализировав возможности по имитации образовательного процесса (таблица), следует конкретно выделить все возможные аргументы «за» и «против» применения медиатехнологий в сфере музыкального образования. В число положительных аргументов можно включить удобство, доступность, сокращение времени на поиск, обработку и редактирования материалов, возможность массового использования. К аргументам «против», можно отнести сложность обеспечения обучающихся веб-ресурсами для взаимодействия с источниками информации. Так же существует ряд сложностей в создании веб-ресурсов и программ, так как ученым разных отраслей необходимо объединять свои силы для создания действительно стоящего продукта, отвечающего как высочайшим технологическим требованиям, так и специфике педагогики и юзабилити [4].

Таблица

Специфика музыкального образования и возможности мультимедийных веб-систем

Основные принятые методы музыкального образования	Возможности
<ul style="list-style-type: none"> <li>– слушание музыки</li> <li>– вокально-хоровое</li> <li>– музыкально-ритмическое исполнительство</li> <li>– выполнение творческих заданий</li> </ul>	<ul style="list-style-type: none"> <li>– средства для просмотра музыкальной информации</li> <li>– хранение информации</li> <li>– прослушивание аудио</li> <li>– редактирование информации</li> <li>– самостоятельный подбор необходимых материалов</li> <li>– быстрый поиск</li> <li>– коммуникативность</li> </ul>

Для непосредственного использования в сфере музыкального образования, информационная веб-система должна предоставить пользователю возможность поиска, хранения, использования музыкальной литературы. Одной из особенностей такой веб-системы является использование всех данных в режиме реального времени, без непосредственной загрузки документов на персональный компьютер. Это позволяет пользователю сократить время на поиск данных, уменьшить объемы постоянно-используемых документов, за счет хранения их в базе данных.

Из изложенного следует актуальная и требующая непосредственного решения задача обоснования веб-системы, которая наилучшим образом отвечает требованиям учащихся музыкальных учреждений.

Сравнив программные средства, предназначенные и используемые повседневно в музыкальной сфере, можно выделить характеристики, формирующие общие критерии для оценки возможностей применения медиатехнологий в музыкальном образовании. Абсолютным аргументом в пользу применения мультимедийных веб-систем стала необходимость обеспечения студентов и преподавателей музыкальных учреждений общедоступной мультимедийной базой, с возможностями использования ее в целях образования. Анализ показал целесообразность использования веб-ориентированной системы, адаптированной под образовательные цели.

Примером такой веб-системы может стать, например, мультимедийная библиотека нотных записей. В качестве единого формата для записи нот можно использовать формат MusicXML, являющийся одним из ведущих форматов представления нотных записей. Выбор формата представления основывался на критериях:

- в основе лежит XML [3], что более приемлемо для интернета, нежели проприетарные двоичные форматы;
- MusicXML является общепризнанным открытым форматом;
- 98% нотных редакторов включают в себя импорт/экспорт файлов в формате MusicXML;
- гибкость и расширяемость.

Изначально общепринятым форматом для обработки и обмена нотной информацией являлся формат MIDI. Однако он не решал проблем с нотными записями, так как MP3 и другие аудио форматы представляют собой звуковые записи, а не нотные партитуры. Многие нотные редакторы имеют собственный проприетарный формат нотных записей, но не исключают возможность взаимодействия с другими форматами. При детальном рассмотрении и изучении различных программ была оценена возможность использования и записи формата MusicXML.

В связи с тем, что формат MusicXML стал стандартом для обмена интерактивными нотными записями, у пользователей появилась возможность создавать музыку в одной программе, а после поделиться своими результатами с другими людьми (рис. 2).

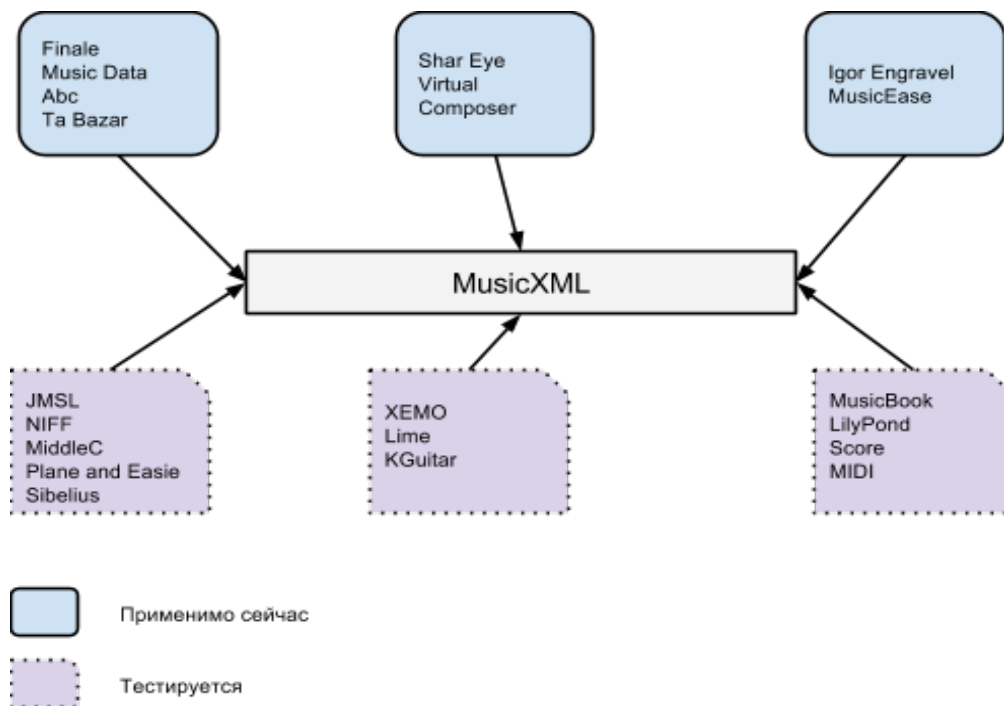


Рис. 2. Формат MusicXML. Взаимодействие с программами и форматами

Причем просмотр и редактирование возможны при помощи любых программ, поддерживающих данный формат.

В музыкальном образовании важным критерием является аудио-визуальный контроль исполнения нотных партитур в режиме реального времени. Поэтому в веб-системе, помимо отображения выбранной нотной партитуры, предполагается и ее проигрывание. Принцип проигрывания аналогичен визуализации.

Этапы отображения выбранного формата (MusicXML) в предполагаемом веб-ресурсе:

- пользователь загружает (или выбирает из уже имеющихся) файл в формате MusicXML;
- файл помещается в внутреннее хранилище (сервер);
- программа осуществляет валидацию формата MusicXML (Если валидация успешна, файл поступает в обработчик javascript. Если валидация не пройдена, процесс останавливается и нотные партитуры не отрисовываются);

используя библиотеку «lib Canvas», программа реализует визуальное отображение нот.

Этапы проигрывания выбранного формата (MusicXML) в предполагаемом веб-ресурсе:

- пользователь выбирает файл;

– веб-система осуществляет валидацию формата MusicXML (Если валидация успешна, файл поступает в обработчик javascript. Если валидация не пройдена, процесс останавливается и нотные партитуры не проигрываются);

– веб-система проигрывает ноты.

Универсальность, расширяемость, открытость спецификации и многофункциональность применения при работе с нотными записями являются весомыми аргументами в пользу выбора формата MusicXML как базового в сфере музыкального образования. Мультимедийные веб-системы и электронные библиотеки нотных записей, приведенные к такому единому формату, позволяют наиболее широко использовать современную техническую базу музыкального образования, а также вводить новые педагогические приемы обучения, направленные на повышение его эффективности и интереса учащихся.

### Библиографический список

1. Веряев А. А., Шалаев И. К. От образовательных сред к образовательному пространству: понятие, формирование, свойства. – Педагог. 1998. [Электронный ресурс]. URL: [http://bspu.secna.ru/Journal/pedagog/pedagog\\_5/a03.html/](http://bspu.secna.ru/Journal/pedagog/pedagog_5/a03.html/)
2. Основы webтехнологий: Интернет-университет информационных технологий / Храмцов П. Б., Брик С. А., Русак А. М., Сурин А. И.: ИНТУИТ.ру, 2003, 520 с.
3. Как программировать на XML / Дейтел Х. М., Дейтел П. Дж., Нието Т. Р., Лин Т. М., Садху П. М.: Бином, 2001, 944 с.
4. Эффективный самоучитель по креативному Web-дизайну. HTML, XHTML, CSS, JavaScript, PHP, ASP, ActiveX / К. Джамса, К. Кинг, Э. Андерсон: изд-во Диасофт, 2005, 672 с.

УДК 004.8

**С. Ю. Овчаров**

студент кафедры компьютерной математики и программирования

**А. В. Фомин**

канд. техн. наук, доц. – научный руководитель

### МЕТОДИКА КОМПЬЮТЕРНОГО ПРОГНОЗИРОВАНИЯ СПРОСА НА ОСНОВЕ ПЕРВИЧНЫХ ДАННЫХ

Управление экономикой на всех уровнях должно стать точной наукой. Достижение наилучшего компромисса между противоречивыми требованиями сокращения расходов на хранение и надежного обеспечения платежеспособного спроса является делом достаточно сложным и требует тщательного количественного анализа. Методы такого анализа составляют предмет теории управления запасами [1].

Для получения преимущества в условиях возрастающей конкуренции необходимо повышать эффективность работы предприятий. Немаловажную роль в этом играет автоматизация бизнес-процессов. Повышение рентабельности – одна из ключевых задач, решение которой обеспечивает успех компании на рынке. Она характеризуется совокупностью действий, направленных на сокращение расходов и увеличение прибыли предприятия. Одно из таких действий – это автоматизация заказов товаров, особенно тех товаров, которые имеют ограниченный срок годности.

Решение проблемы автоматизации процессов в отделе закупок торгового предприятия состоит из нескольких этапов, таких как:

- построение программного обеспечения для автоматизации и учета работы отдела закупок, анализа торговых показателей и составления заказов;
- анализ торговых показателей с целью установления наиболее значимых признаков;
- изучение и выбор наиболее эффективных алгоритмов машинного обучения для автоматического формирования заказов.

В данной статье мы рассмотрим алгоритм машинного обучения для прогнозирования спроса и рассчитаем значение предстоящего спроса на основе данных о предыдущих продажах товара.

Количество факторов, которые влияют на принятие решений в реальной жизни, достаточно велико. Приведем их здесь.

Факторы, влияющие на принятие решений следующие:

- количество заказанного товара;
- количество продаваемого товара со скидкой за день в аналогичных условиях;
- количество продаваемого товара без скидки за день в аналогичных условиях;
- количество списываемого (непроданного) товара за день в аналогичных условиях;
- день недели, на который делается заказ;
- время, когда закончился товар;
- фактор «праздничного дня», на который делается заказ;
- фактор «Пасхи» (количество дней до праздника);
- фактор «Нового года» (количество дней до праздника);
- прогноз погоды на день;
- порядковый номер недели со дня запуска магазина;
- сезон.

В данной статье рассмотрим один из этих факторов – количество заказанного товара.

Данный фактор выбран исключительно в методических целях для примера реализации работы алгоритма прогнозирования. Количество факторов можно постоянно изменять и добавлять, но, для наглядности работы алгоритма, будет использован один.

Основными информативными параметрами, которые больше всего интересуют нас для решения поставленной задачи это:

- количество заказанных единиц товара;
- количество полученной прибыли от продаж.

Построим график зависимости прибыли от заказанного количества. Данные для построения получены опытным путем, взята выборка по продажам одного товара за три месяца в магазине по продаже хлеба и кондитерских изделий. Товар скоропортящийся и весь нераспроданный за день товар подлежал списанию. Отсюда мы можем на графике увидеть отрицательную прибыль.

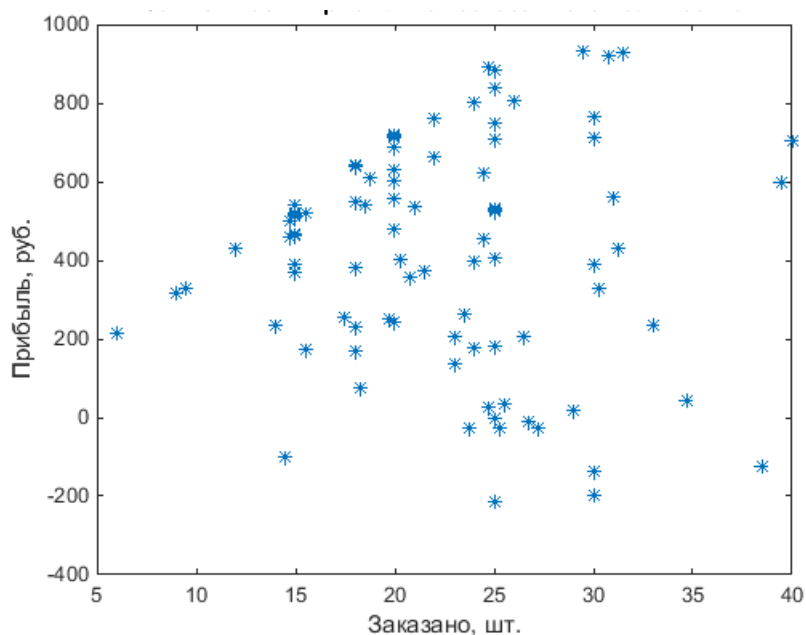


Рис. 1. График зависимости прибыли от количества в заказе

Назовем эти зависимости обучающей выборкой. Конечно, мы видим, что данный график не показывает нам явных зависимостей, однако, прибегая к алгоритмам машинного обучения, мы можем выдвинуть гипотезу о существовании функции, которая в своем максимуме может дать нам максимально подходящее значение для заказа [2]. В качестве такой функции-гипотезы, может быть выбран полином второй степени.

$$h(x) = \theta_0 \cdot x^0 + \theta_1 \cdot x^1 + \theta_2 \cdot x^2.$$

За счет подбора коэффициентов  $\theta$  при переменной  $x$  можно построить функцию, которая в своем максимуме (экстремуме) даст значение, которое затем используется для определения количества заказываемого товара, т. е. как прогноз спроса. Подбор таких коэффициентов основан на принципе метода наименьших квадратов.

Метод наименьших квадратов (МНК, англ. Ordinary Least Squares, OLS) – математический метод, применяемый для решения различных задач, основанный на минимизации суммы квадратов отклонений некоторых функций от искомым переменных. Он может использоваться для «решения» переопределенных систем уравнений (когда количество уравнений превышает количество неизвестных), для поиска решения в случае обычных (не переопределенных) нелинейных систем уравнений, для аппроксимации точечных значений некоторой функцией. МНК является одним из базовых методов регрессионного анализа для оценки неизвестных параметров регрессионных моделей по выборочным данным.

Пусть  $x$  – набор  $m$  неизвестных переменных (параметров),  $f_i(x), i = 1, \dots, n, n > m$  – совокупность функций от этого набора переменных. Задача заключается в подборе таких значений  $x$ , чтобы значения этих функций были максимально близки к некоторым значениям  $y_i$ . По существу речь идет о «решении» переопределенной системы уравнений  $f_i(x) = y_i, i = 1, \dots, n$  в указанном смысле максимальной близости левой и правой частей системы. Сущность МНК заключается в выборе в качестве «меры близости» суммы квадратов отклонений левых и правых частей  $|f_i(x) - y_i|$ . Таким образом, сущность МНК может быть выражена следующим образом:

$$\sum_i e_i^2 = \sum (y_i - f_i(x))^2 \rightarrow \min_x.$$

В случае если система уравнений имеет решение, то минимум суммы квадратов будет равен нулю и могут быть найдены точные решения системы уравнений аналитически или, например, различными численными методами оптимизации. Если система переопределена, то есть, говоря нестрого, количество независимых уравнений больше количества искомым переменных, то система не имеет точного решения и метод наименьших квадратов позволяет найти некоторый «оптимальный» вектор  $x$  в смысле максимальной близости векторов  $y_i$  и  $f_i(x)$  или максимальной близости вектора отклонений  $e$  к нулю (близость понимается в смысле евклидова расстояния) [3].

Для того, чтобы найти коэффициенты, при которых будет найден минимум сумм квадратов отклонений необходимо решение нормального уравнения [4] вида:

$$\theta = (X^T \cdot X)^{-1} \cdot X^T \cdot y.$$

где  $\theta$  – вектор искомым коэффициентов  $[\theta_0, \theta_1, \theta_2, \dots, \theta_n]^T$  при переменной  $x$ ;  $X$  – матрица влияющих параметров;  $y$  – вектор результирующих параметров.

Рассмотрим пример нахождения коэффициентов и величины прогнозируемого спроса на основе данных, полученных от продаж одного вида товара в розничном магазине.

Частичная выборка данных приведена в табл. 1.

Частичная выборка данных

Матрица X (x – заказ, шт)			Вектор y.
$x^0$	$x^1$	$x^2$	Прибыль, руб
1	18,25	333,06	198,33
1	18	324	341,48
1	18	324	221,72
1	15	225	644,69
1	9	81	327,58

Проведя расчеты в Matlab получим вектор

$$\theta = [2.296149716293742e + 02, 17.821949959375320, -0.387247528054273 ]^T .$$

Далее построим функцию с полученными коэффициентами и построим график.

$$h(x) = 229,615 \cdot x^0 + 17,8219 \cdot x^1 + 0,3872 \cdot x^2 .$$

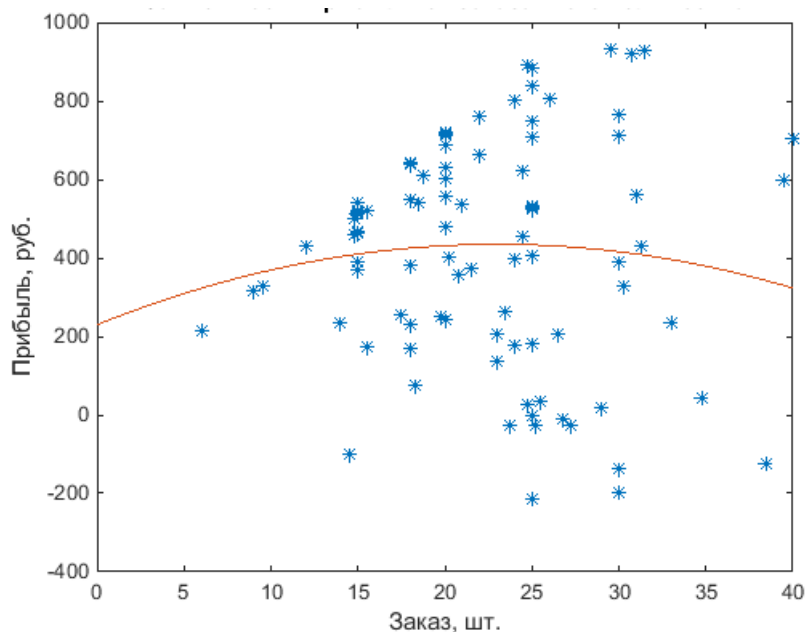


Рис. 2. График функции-гипотезы

Далее найдем значение  $x$  в максимуме полученной функции. Для этого приравняем производную от функции к нулю и решим уравнение  $17,8219 + 0,3872 \cdot 2 \cdot x = 0$ . Получим  $x \approx 23$ .

Итак, сделано предсказание, что на следующий день спрос будет равняться 23 штуки рассматриваемого товара.

В заключение нужно отметить, что достоинствами данного метода является простота и наглядность формул, простота реализации вычислений при малом количестве влияющих параметров. К недостаткам же следует отнести то, что в некоторых случаях можно столкнуться с неинвертируемостью матриц, например, когда влияющие параметры линейно зависимы друг от

друга. Также обратим внимание, что если количество влияющих параметров будет превышать 100 тыс., то, вследствие сложности вычисления по предложенной формуле, предлагается другой подход к нахождению коэффициентов при  $x$  с помощью метода понижающего градиента.

### Библиографический список

1. Рыжиков Ю. И. Логистика, очереди и управление запасами. – СПб.: ГУАП, 2011 – 477 с.
2. Andrew Ng. Курс лекций по машинному обучению в университете Stanford. <https://class.coursera.org/ml-005/lecture>. 25.05.2015.
3. [https://ru.wikipedia.org/wiki/Метод\\_наименьших\\_квадратов](https://ru.wikipedia.org/wiki/Метод_наименьших_квадратов). 20.05.2015.
4. [www.MachineLearning.ru](http://www.MachineLearning.ru). 22.05.2015.

УДК 681.3.06

**П. А. Охтилев**

студент кафедры компьютерной математики и программирования

**М. Ю. Охтилев**

д-р техн. наук, проф. – научный руководитель

## ПРОГРАММНЫЙ КОМПЛЕКС ПЕРВИЧНОЙ ОБРАБОТКИ ИНФОРМАЦИИ ПРИ ОЦЕНИВАНИИ СОСТОЯНИЙ СЛОЖНЫХ ТЕХНИЧЕСКИХ ОБЪЕКТОВ В СОСТАВЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

В различных сферах деятельности мирового сообщества существуют сложные задачи, решение которых требует многогранной оценки. Это означает, что эти задачи возникают в сложной информационной среде, где для принятия правильного решения необходимо учесть множество критериев, множество деталей, которые чрезвычайно сложно отследить одному человеку, даже если он является высококвалифицированным специалистом в своей области. Зачастую сложность такой задачи доходит до такого уровня, что и группа специалистов не в состоянии решить ее, или эта группа потратит на ее решение неприемлемое для конкретной ситуации время. В таком случае, решение задачи требует автоматизации при помощи компьютера, вычислительная мощность которого позволит не только учесть все факторы, но и прийти к абсолютно верному решению в кратчайшие сроки. В рамках таких задач возникли системы поддержки принятия решений (СППР), позволяющие собрать все критерии воедино, оценить их и предложить пользователю – эксперту в своей области – решение [1]. В идеальной ситуации такая система для конкретной задачи должна выполнять следующие функции:

- извлекать информацию (критерии оценки) из разнородных источников;
- консолидировать собранную информацию;
- обработать полученные данные по определенным заданным алгоритмам;
- моделировать определенные ситуации на основе критериев оценки и результатов расчетов;
- прогнозировать ситуации на основе всех полученных данных;
- визуализировать данные в удобном для конечного пользователя виде.

Одной из актуальных задач, требующих своего решения, является поддержка принятия решений (ППР) при функционировании автоматизированной системы управления Вооруженными Силами (АСУ ВС). В рамках решения задачи по прогнозированию состояния противостоящих наземных войск, военных объектов на возможных направлениях ударов противника, а также по оценке требующихся в конкретных ситуациях действий боевой авиации, требуется разработать автоматизированную СППР, позволяющую:

- осуществлять прогноз исхода боевых действий на направлениях;
- формировать выводы по оценке состояния своих войск и войск противника на направлениях после боевых действий;
- формировать возможные решения предупредительных действий по стабилизации состояния войск на направлениях перед боевыми действиями.

В рамках первичной обработки информации о состояниях вышеизложенных технических объектов требуется создать программный комплекс, осуществляющий:

- извлечение данных об объектах с учетом необходимых критериев из источника информации;
- обработку данных по определенным алгоритмам для формирования полной информации о состояниях сложных технических объектов в структурированном виде;
- запись данных в источник информации для последующей вторичной обработки и анализа.

Многокритериальные задачи возникают все чаще в различных отраслях, в том числе в военной сфере, в экономике на предприятиях, в инфраструктуре и других отраслях. В принципе, любой механизм, помогающий человеку принимать обоснованные решения в сложной информационной среде, а не действовать на основе только интуиции и своего профессионального опыта, можно назвать СППР. Поэтому элементы СППР есть практически в любых программах, где имеется возможность получить отчет либо построить график. В условиях большого количества исходных данных для анализа, множества критериев и малого количества времени для принятия решения использование человеческого труда, как решения, становится неприемлемым, а СППР становятся необходимостью. В частном случае, для прогнозирования состояния войск непозволительно потратить несколько часов, когда единица времени является стратегически важным элементом – на принятие решения должно отводиться не более нескольких минут. Кроме того, для такой задачи необходимо свести к минимуму ошибки расчетов, что может достигаться благодаря автоматизации процесса решения задачи в рамках первичной обработки исходных данных.

На данный момент большинство организаций, которые должны предпринимать определенные действия в сложной информационной обстановке, стараясь действовать максимально эффективно, прибегают к различным способам принятия решений:

- организация специальной аналитической группы для оценки той или иной обстановки или состояния объекта (системы объектов);
- использование различных программных сред с математическим аппаратом;
- использование узкоспециализированных АСУ.

Каждый из вышеназванных вариантов обладает своими плюсами и минусами. Однако система поддержки принятия решений является наиболее мощным инструментом, как по скорости вычислений, так и по удобству организации и представления данных.

СППР основана на взаимодействии программного комплекса с базой данных (БД) на основе клиент-серверной архитектуры. Таким образом, имеется распределенная система, в которой есть некое хранилище данных с заложенной туда информацией об объектах и критериях оценки, а также – группа алгоритмов первичной и вторичной обработки данных, где на основе первичной обработки осуществляется сбор и обработка этих данных для формирования состояний как отдельных объектов, так и системы объектов, а на основе вторичной – формулирование неких выводов с использованием результатов первичной обработки и их анализа, с помощью которых СППР не только визуально сообщит пользователю о результатах, но и попытается сформировать определенные предписания действий для достижения того результата, который необходим пользователю. Структура подобной СППР представлена на рис. 1.





Рис. 1. Схема взаимодействия СППР с пользователем

Первичная обработка информации о состояниях сложных технических объектов основана на работе двух программных модулей СППР, написанных на языке C++, взаимодействующих с базой данных посредством SQL-запросов и хранимых процедур.

Модуль прогнозирования состояния войск, выполняющий следующие функции (рис. 2):

- формирование списка средств ведения боя противоборствующих сторон на направлении удара;
- анализ возможностей уничтожения средств ведения боя и военных объектов, а также отражения удара на основе множества критериев;
- формирование математического ожидания уничтожения противника;
- формирование результатов ведения боя.

1. Модуль поиска и формирования списка объектов для осуществления стабилизации войск на направлении удара, выполняющий следующие функции (рис. 3):

- анализ результатов ведения боя;
- расчет необходимого количества средств для стабилизации войск;
- поиск средств ведения боя на основе множества критериев;
- формирование списка передислоцируемых на направление удара средств, запись списка в БД.



Рис. 2. Схема работы модуля прогнозирования состояния войск

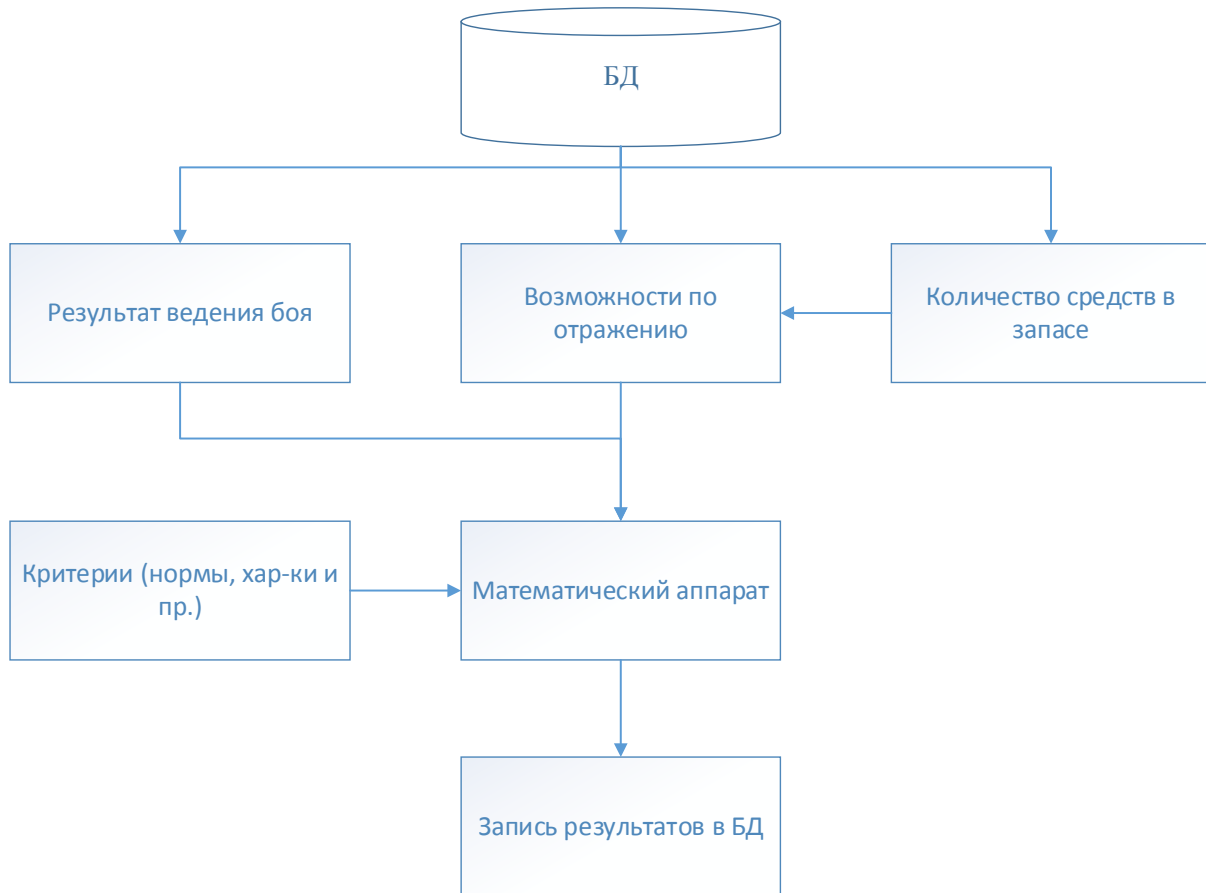


Рис. 3. Схема работы модуля стабилизации войск на направлении удара

К достоинствам вышеописанного подхода к решению задачи можно отнести высокую скорость работы: в сравнении с аналитической группой, которая решала бы подобную задачу несколько часов, или программной среды с математическим аппаратом, где приходилось бы каждый раз задавать алгоритм работы, элементы подсистемы СППР решают подобного рода задачи за несколько минут. Кроме того, важно учесть удобную организацию данных, которая позволяет с легкостью находить необходимые данные, масштабировать систему и реорганизовывать ее под другие задачи такого класса заменой данных в БД на необходимые.

К недостаткам системы можно отнести высокую сложность разработки первичной обработки данных (сложные ветвящиеся алгоритмы, обработка всех ситуаций и пр.), что требует наличия высококвалифицированных специалистов-экспертов и приводит к длительному времени разработки [2].

В заключение следует отметить, что, несмотря на сложность разработки, программный комплекс первичной обработки информации является удобным и мощным инструментом организации данных и вычислений в составе СППР: благодаря автоматизации процесса расчетов принятие решения в сложной информационной обстановке достигается в кратчайшие сроки, а ошибки расчетов сведены к минимуму.

### Библиографический список

1. Ларичев О. И., Петровский А. Б. Системы поддержки принятия решений. Современное состояние и перспективы их развития. // Итоги науки и техники. Сер. Техническая кибернетика. – Т. 21. М.: ВИНТИ, 1987. С. 131–164.
2. Трахтенгерц Э. А. Компьютерные системы поддержки принятия управленческих решений // Проблемы управления. М.: СенСидат, 2003. С. 13–28.

**Д. В. Печенин**

студент кафедры вычислительных систем и сетей

**Р. И. Дамизов**

студент учебного военного центра

**В. В. Котович**

канд. техн. наук – научный руководитель

**ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО КОМПЛЕКСА  
ДЛЯ ОБУЧЕНИЯ РАБОТЕ С МИКРОКОНТРОЛЛЕРАМИ  
С ИСПОЛЬЗОВАНИЕМ CASE-ТЕХНОЛОГИЙ**

Данный программный комплекс является частью обучающего аппаратно-программного комплекса, ориентированного на повышение у студентов теоретических и практических навыков работы с микроконтроллерами [1]. Комплекс учитывает возможность дистанционного обучения студентов и позволяет получать доступ к лабораторному оборудованию в любое время, используя сеть Internet.

При разработке программного комплекса использовались CASE технологии. CASE средства (англ. Computer-Aided Software Engineering) – это набор инструментов для автоматизации процесса проектирования и разработки программного обеспечения. CASE технологии позволяют отграничить процесс проектирования программных продуктов от процесса кодирования.

Существует два основных подхода к проектированию:

- структурный – подразумевает декомпозицию задачи;
- объектно-ориентированный – основным инструментом является унифицированный язык моделирования – UML.

В составе CASE средств имеются:

- средства анализа;
- средства проектирования баз данных;
- средства разработки приложений;
- средства реинжиниринга приложений;
- средства планирования и управления проектом;
- средства тестирования;
- средства документирования.

При разработке данного программного комплекса в качестве среды проектирования был выбран программный продукт RationalRose (лидер на рынке средств проектирования) и UML 2.4.1 спецификации 2011 г.

На начальном этапе была составлена *диаграмма USE-CASE* (рис. 1). Она играет важную роль в процессе проектирования и позволяет определиться с функциональными требованиями к системе. Варианты использования являются основой для анализа, проектирования и тестирования.

При реализации диаграммы USE-CASE было создано 4 «актера» (внешних сущностей, взаимодействующих с системой):

- «Пользователь»;
- «Автор» (расширяет сущность «Пользователь»);
- «Администратор» (расширяет сущность «Пользователь»);
- «Внешнее устройство».

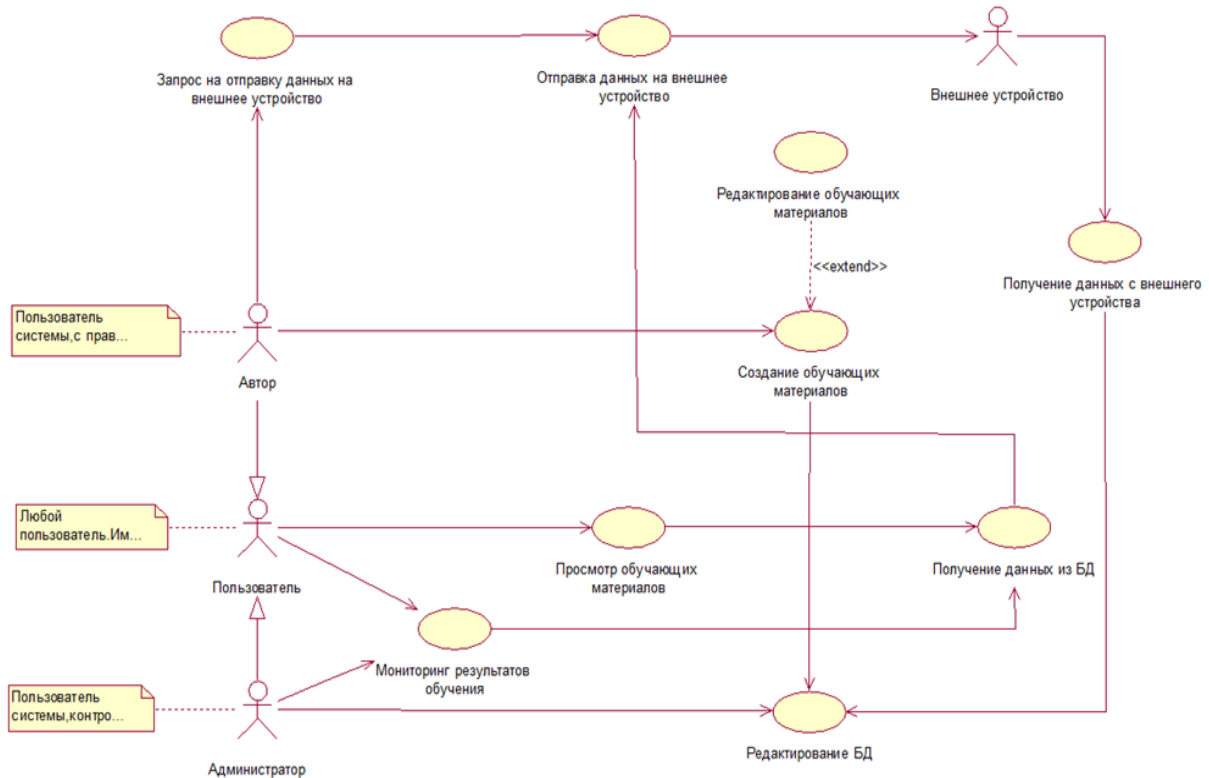


Рис. 1. USE-CASE диаграмма системы

Также были определены основные варианты использования («Создание обучающих материалов», «Редактирование БД»).

Далее была создана диаграмма классов (рис. 2), являющаяся статическим описанием системы и позволяющая оценить общую структуру системы.

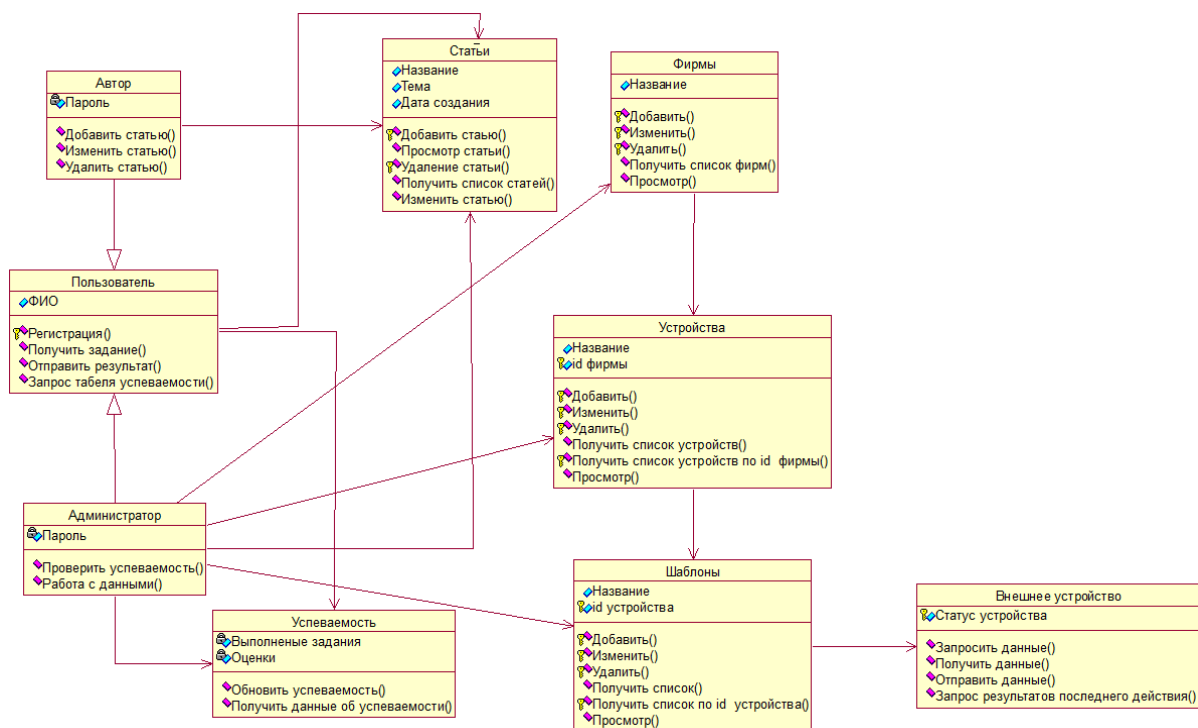


Рис. 2. Диаграмма классов

Для реализации проекта был выбран формат Web 2.0 приложения с клиент-серверной архитектурой на языке php, что обеспечило кроссплатформенность и позволило пользователям получить удаленный доступ к виртуальным рабочим местам. Для реализации проекта также были использованы html, css, javascript, и технология ajax, позволяющая производить обмен данными с пользователем в фоновом режиме и производить асинхронное обновление страницы (производится обновление определенного элемента страницы). При выборе структуры приложения был проведен анализ различных моделей (MVC, MVP, MVVM). Наиболее подходящей по своей структуре была модель MVC («модель–вид–контроллер») (рис. 3). Данный вариант реализации позволил разделить приложение на три части:

- модель;
- вид;
- контроллер.

Это дает возможность изменять любой из элементов с минимальным воздействием на другие. Данный факт не оказывает большого влияния на начальных этапах реализации, но его польза возрастает прямо пропорционально объему проекта.

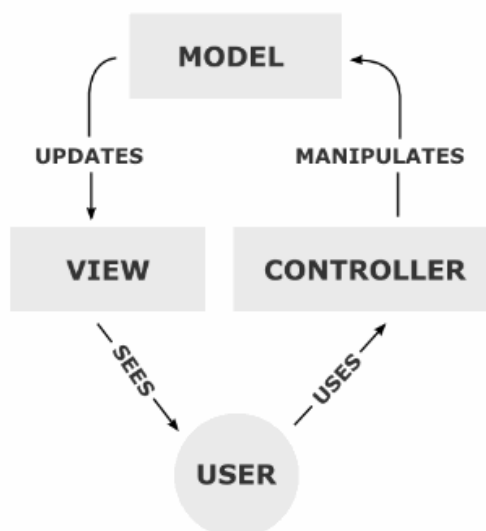


Рис. 3. Модель MVC

Работу web-приложения с данной структурой можно коротко описать следующим образом. Контроллер получает событие извне и, в соответствии с заложенной в него логикой, реагирует на это событие, изменяя модель посредством вызова соответствующего метода.

Для разработки были использованы программы XAMPP и IDENetBeans. XAMPP позволяет проводить локальную разработку web-приложения и включает в себя множество компонентов, но при разработке данного программного комплекса использовались только система управления базами данных SQL и web-сервер Apache. NetBeans является интегрированной средой разработки с большим количеством поддерживаемых языков и возможностей. Среди существующих IDE (IDENetBeans, PhpStorm, Eclipse, IntelliJIDEA) предпочтение было отдано NetBeans за ее возможность тестирования web-приложений, поддержку работы с репозиториями и наличие функция autocomplete, что значительно ускорило разработку программного комплекса. Оба программных продукта являются бесплатными.

Данное приложение позволило нам проводить компиляцию, сборку исходных кодов и прошивку микроконтроллера, используя usb-порт. Также стоит отметить, что данный продукт может работать с большинством современных программаторов используя usb-порт или UART.

В дальнейшем планируется реализовать клиенты данного приложения для различных платформ, чтобы обеспечить максимальный набор возможностей при работе с приложением. Для обмена информацией между клиентом и сервером был выбран формат json (JavaScriptObjectNotation). Этот формат даёт возможность легко обрабатывать данные как клиенту, так и серверу.

### **Библиографический список**

1. Разработка системы автоматизации создания программ для микроконтроллеров / А. А. Соколов // 66 международная студенческая научная конференция ГУАП: Сб. докл.: Ч. 2. Гуманитарные науки / СПбГУАП Санкт-Петербург, 2013.

УДК 651.012.12

**О. В. Полех**

студент кафедры компьютерной математики и программирования

**В. В. Бураков**

д-р техн. наук, доц. – научный руководитель

## **СРАВНИТЕЛЬНЫЙ АНАЛИЗ СИСТЕМ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА**

Документооборот является информационной основой деятельности предприятия. Документы сопровождают все основные бизнес-процессы и являются важнейшим управленческим ресурсом.

В данной статье проведен анализ, какая система лучше подходит для решения определенных задач – в частном и государственном секторе.

Электронный документооборот (ЭДО) – это способ организации работы с документами, при котором основная масса документов используется в электронном виде и хранится централизованно. При этом введение электронного документооборота не является процессом искоренения бумажных документов, однако считается приоритетным направлением оптимизации и коррекции работы с документами [1].

Использование систем электронного документооборота дает такие преимущества, как повышение производительности компании, мгновенный доступ к актуальной информации, уменьшение ошибок типа «человеческий фактор», снижение расходов на хранение бумажных документов, коллективная работа с электронным документом, его защита от повреждений и несанкционированного искажения, эффективный контроль и прозрачность работы.

Рынок систем электронного документооборота в последние годы является одним из самых динамично развивающихся сегментов отечественной ИТ-индустрии. Его развитию способствует растущая с каждым годом потребность предприятий в оптимизации документооборота. На российском рынке присутствует несколько десятков отечественных и зарубежных СЭД.

По данным TAdviser на сентябрь 2014 г., наиболее активно внедрения СЭД в России реализуются в государственном секторе, в первую очередь это относится к органам власти, но также и компаниям с государственным участием. Также спрос на системы электронного документооборота высок в финансовом секторе (банки, страховые компании), в торговле, строительстве, машиностроении и других отраслях (рис. 1) [2].



Рис. 1. Распределение СЭД по отраслям

Лидерами по количеству проектов внедрения на российском рынке являются системы Дело, DocsVision, Directum, E1 Евфрат, Documentum и 1С:Документооборот (рис. 2) [3].

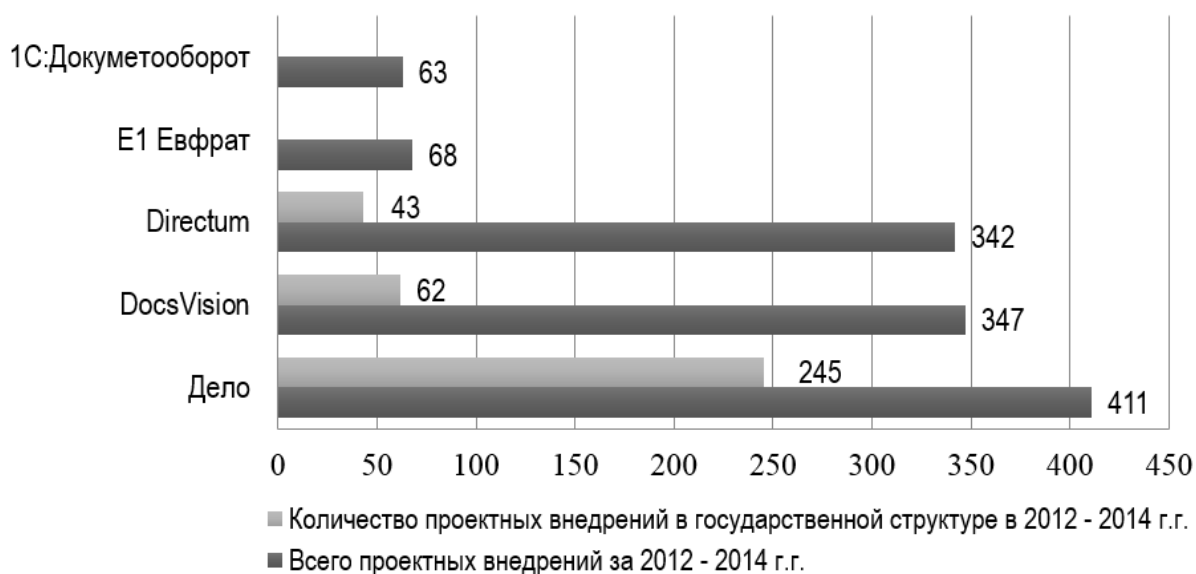


Рис. 2. Системы по количеству проектов внедрений за 2012–2014 гг.

Для сравнительного анализа были выбраны наиболее популярные системы документооборота E1 Евфрат, Дело, Directum, DocsVision, 1С:Документооборот и одна молодая система – VerDox. Сравнение происходило по их основным функциональным возможностям (табл. 1) и общим характеристикам (табл. 2).

Подсчет баллов по критериям проводился по трехбалльной системе: 1 балл – если функция присутствует в программном продукте; 0,5 балла – функция заявлена разработчиком, однако не является собственной разработкой компании или ограниченность функции; 0 баллов – функция не заявлена.

Таблица 1

Сравнение СЭД по основным функциональным возможностям

Функция	Е1 Евфрат	Docs Vision	1С:Документо- оборот	VerDox	Directum	Дело
Регистрация документов	1	1	1	1	1	1
Контроль исполнения поручений	1	1	1	1	1	1
Сканирование и распознавание документов	1	0,5	0,5	0	0,5	0,5
Графический редактор маршрутов бизнес-процессов	1	1	1	0	1	1
Электронный архив	1	1	1	1	1	1
шифрование данных	0,5	0,5	0,5	0	0,5	0,5
Электронная подпись	0,5	0,5	0,5	0	0,5	0,5
Мандатный доступ	0	1	0	0	0	0
Работа с обращениями граждан	0	1	1	0	1	1
Итого	6	7,5	6,5	3	6,5	6,5

Подсистема оповещений предназначена для автоматической рассылки оповещений в электронную почту пользователей о различных событиях системы.

Бизнес-процесс – это процесс обращения информации (документов) по заранее заданной схеме. В схеме бизнес-процесса могут быть заданы одна или несколько точек входа и выхода, а также последовательность действий, совершаемых в заданном порядке и в определенных условиях [4].

Электронная подпись позволяет подтвердить личность автора или отправителя корреспонденции, а также является гарантией того, что в документ после его подписания не были внесены изменения.

Таблица 2

Сравнение СЭД по общим характеристикам

Функция	Е1 Евфрат	Docs Vision	1С:Документо- оборот	VerDox	Directum	Дело
Интеграция с другими приложениями (1С, MS Office и др.)	1	1	0,5	0	1	1
Легкость внедрения	0	0	0	1	0	0
Системы средств резервного копирования и восстановления данных	1	1	1	0	1	1
Мобильное решение	1	1	1	0	1	1
Тонкий клиент	1	1	0	0	1	1
Итого	4	4	2,5	1	4	4

Объединим итоги сравнения таблиц 1 и 2 в таблице 3.

Таблица 3

Итог сравнения СЭД

Критерий	Е1 Евфрат	Docs Vision	1С:Документо- оборот	VerDox	Directum	Дело
Функциональные возможности	6	7,5	6,5	3	6,5	6,5
Общие характеристики	4	4	2,5	1	4	4
Итого	10	11,5	9	4	10,5	10,5



Одним из определяющих критериев выбора покупателя СЭД была и остается цена продукта. Для наибольшего представления рассмотрим два сравнительных среза по расширенной комплектации (из расчета продажи лицензии на 100 пользователей) и базовой комплектации (из расчета продажи лицензии на 10 пользователей) (табл. 4).

Таблица 4

Сравнение СЭД по стоимости базовой и расширенной версий (тыс. руб.)

Версии	Е1 Евфрат	DocsVision	1С:Документооборот	VerDox	Directum	Дело
Базовая	73	18,5	77,4	50	26	92,5
Расширенная	520	568,5	396	450	560	837,5

Системы Е1 Евфрат, 1С:Документооборот и VerDox представляют собой полностью готовое решение, в то время как DocsVision, Directum и Дело построены по принципу отдельных модулей, поэтому их финальная стоимость может быть больше. Плюс модульной архитектуры в том, что заказчик может выбрать только необходимый набор функций, не переплачивая за лишний функционал.

По итогам сравнения системы Дело и Directum имеют одинаковый набор основных функций. Система Дело реализует не только интеграцию с автоматизированной системой «Обращения граждан», но и интеграцию с «Единым порталом государственных и муниципальных услуг», а также взаимодействие с системой межведомственного электронного взаимодействия (СВЭМ) и межведомственным электронным документооборотом (МЭДО). Поэтому СЭД Дело востребована в госсекторе [5].

Минусом системы Directum является ее не кроссплатформенность – работает только на операционной системе Microsoft Windows. СЭД Directum включает в себя игру под названием Directum Восхождение. Данный модуль представляет из себя квест-игру, целью которой является получение наград за выполненные задания и тем самым быстрее и легче проходит процесс освоения системы сотрудниками предприятия [6].

Преимуществом системы DocsVision в плане безопасности данных можно выделить наличие мандатного управления доступом. Мандатная модель разграничения доступа предполагает назначение объекту грифа секретности, а субъекту – уровня допуска. Доступ субъектов к объектам в мандатной модели определяется на основании правил «не читать выше» и «не записывать ниже». Это означает, что пользователь не может прочитать информацию из объекта, гриф секретности которого выше, чем его уровень допуска. Также пользователь не может перенести информацию из объекта с большим грифом секретности в объект с меньшим грифом секретности. Использование мандатной модели предотвращает утечку конфиденциальной информации, но снижает производительность компьютерной системы [7, 8].

Система Е1 Евфрат единственная, из рассмотренных, имеет свои встроенные средства сканирования и распознавания документов, остальные же интегрируются с решениями компании АВВУУ. СЭД Е1 Евфрат по функционалу не сильно отстает от лидирующей тройки, но стоимость продукта в разы меньше, так как все функции системы входят в стоимость рассчитанную в табл. 4 [9].

Система 1С:Документооборот по сути является большим модулем системы 1С:Предприятие, так как без предустановленной системы 1С:Предприятие невозможно установить 1С:Документооборот. СЭД 1С:Документооборот выпускается в трех вариантах: 1С:Документооборот КОРП – для предприятий со сложной организационной структурой и сложным документооборотом, 1С:Документооборот ПРОФ – для малых и средних предприятий и учреждений, 1С:Документооборот государственного учреждения – включает опцию «Учет обра-

щения граждан» и взаимодействие с МЭДО. Существует отдельная система 1С:Архив. Не смотря на свои недостатки, СЭД 1С:Документооборот стоит не дорого для среднего бизнеса [10].

Система VerDox больше подходит для компаний малого и среднего бизнеса – при небольшом количестве лицензий заявленный разработчиком базовый функционал вполне достаточен для работы. Относительно систем Е1 Евфрат и 1С:Документооборот стоимость продукта Verdox завышена [11].

Для работы с грифованными документами в СЭД интегрируется средство криптографической защиты информации «КриптоПроСР», сертифицированное ФСБ РФ.

Там, где необходим высокий уровень конфиденциальности, стоит использовать систему DocsVision, так как она использует мандатную модель управления доступом. Системы Дело и Directum подходят для использования в госсекторе федерального и регионального уровней соответственно. Для крупного бизнеса хорошо подходит система Е1 Евфрат. 1С:Документооборот рассчитана для предприятий разного уровня, но больше подходит для среднего бизнеса. Verdox – мало функциональная система, поэтому лучше использовать на небольшом предприятии.

Для подготовки сравнительного анализа использовались публично доступные материалы с официальных сайтов производителей СЭД.

### Библиографический список

1. ECM-journal: Просто об электронном документообороте, URL: <http://ecm-journal.ru/mustknow>
2. Tadviser: СЭД в России: отраслевая специфика, URL: [www.tadviser.ru/index.php](http://www.tadviser.ru/index.php) / Статья: СЭД в России: отраслевая специфика
3. Tadviser: Системы по количеству проектов внедрений (СЭД), URL: [http://www.tadviser.ru/index.php?title=СЭД&ptype=system&cache=no&utm\\_source=tadviser\\_inner&utm\\_medium=tech\\_search&utm\\_campaign=tech\\_main#ttop](http://www.tadviser.ru/index.php?title=СЭД&ptype=system&cache=no&utm_source=tadviser_inner&utm_medium=tech_search&utm_campaign=tech_main#ttop)
4. Artw: Бизнес-процессы, URL: [artw.ru/services/erp/business](http://artw.ru/services/erp/business)
5. ЭОС: Функциональные возможности, URL: [http://www.eos.ru/eos\\_products/eos\\_delo/functions.php](http://www.eos.ru/eos_products/eos_delo/functions.php)
6. Directum: ОсистемеDirectum, URL: [www.directum.ru/system](http://www.directum.ru/system)
7. Docsvision:Docsvision 5, URL: [www.docsvision.com/o-sisteme/dv5/dv5](http://www.docsvision.com/o-sisteme/dv5/dv5)
8. Интуит: Общие вопросы технической защиты информации, URL: [www.intuit.ru/studies/courses/2291/591/lecture/12689?page=3](http://www.intuit.ru/studies/courses/2291/591/lecture/12689?page=3)
9. Е1Евфрат: Сканирование и распознавание, URL: [www.evfrat.ru/about/configurations/e-archive](http://www.evfrat.ru/about/configurations/e-archive)
10. 1С:Предприятие: Конфигурации 1С:Документооборота, URL: [v8.1c.ru/doc8/versions.htm](http://v8.1c.ru/doc8/versions.htm)
11. Verdox: Решаемые задачи, URL: [verdox.ru/solutions](http://verdox.ru/solutions)

УДК 519.688

### **Р. С. Прокуратов**

студент учебного военного центра

### **П. М. Безняков**

канд. техн. наук – научный руководитель

## **МОДЕЛИРОВАНИЕ И ОЦЕНКА ЭФФЕКТИВНОСТИ РАЗМЕЩЕНИЯ СИЛ И СРЕДСТВ ПРИ НАНЕСЕНИИ ПРОТИВНИКОМ ГРУППОВОГО РАКЕТНОГО УДАРА**

В своей боевой деятельности подразделения сталкиваются с вопросами прогнозирования хода и исхода боевых действий. Одной из важнейших проблем для подразделения является оценка эффективности распределения его сил и средств на участке местности. Проблема заключается в том, что подразделение на основе поступившей разведывательной информации размещает свои силы и средства в таком порядке, который является наиболее эффективным с точки зрения некоторого выбранного критерия.

Постановка задачи. На некотором участке местности (рис. 1), длиной  $a$  км, шириной  $b$  км, рассредоточена ротно-тактическая группа (РТГ) войск, расположенная вне линии соприкосновения с противником на удалении  $p$  км. РТГ занимается совершенствованием боевого слаживания подразделений, подготовкой вооружения и военной техники к использованию, пополнением боеприпасами и запасами материально-технических средств (МТС). Подразделения РТГ располагаются в произвольном порядке, но сгруппированы по своему предназначению.

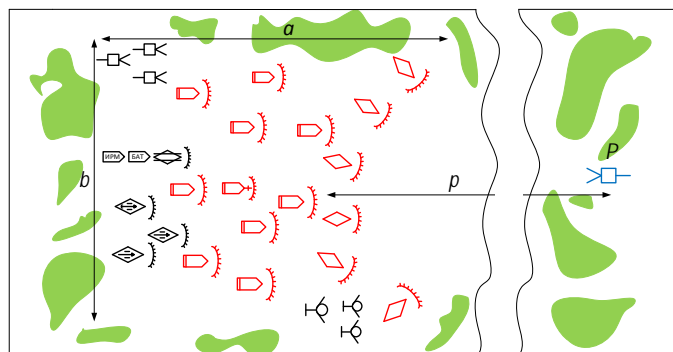


Рис. 1. Постановка задачи

В результате проведения воздушной разведки противник определил местонахождение подразделений и принял решение нанести превентивный огневой удар по расположению РТГ.

Противник может произвести по одному прицельному выстрелу по каждому подразделению РТГ неуправляемыми ракетами MGR-1 Honest John. Необходимо выбрать безопасный вариант распределения подразделений РТГ на местности с точки зрения минимизации среднего ущерба.

Задача рассматривается со следующими допущениями и ограничениями: рельеф местности – плоскость; естественные преграды отсутствуют. Силы и средства РТГ расположены на плоскости, ограниченной прямоугольником. Противнику известны точные координаты целей. Влияние атмосферных и погодных условий не учитываются. Силы и средства своего подразделения считаем однородными, то есть равнопоражаемыми.

Формализация задачи. Введем на плоскости прямоугольную систему координат  $OXY$  (рис. 2). Начало координат расположим в центре прямоугольника длиной  $a$  км и шириной  $b$  км. Оси системы координат параллельны сторонам прямоугольника. Представим каждую позицию  $i$ -го подразделения РТГ точкой  $D_i$  на плоскости с координатами  $(x_i, y_i)$ , где  $i = 1, \dots, n$  (рис. 2). Всего в РТГ  $n$  позиций. При этом  $x_i \in \left[-\frac{a}{2}, \frac{a}{2}\right]$ ,  $y_i \in \left[-\frac{b}{2}, \frac{b}{2}\right]$ . Огневое средство противника представим в виде точки  $P$  на оси абсцисс с координатами  $(p, 0)$ .

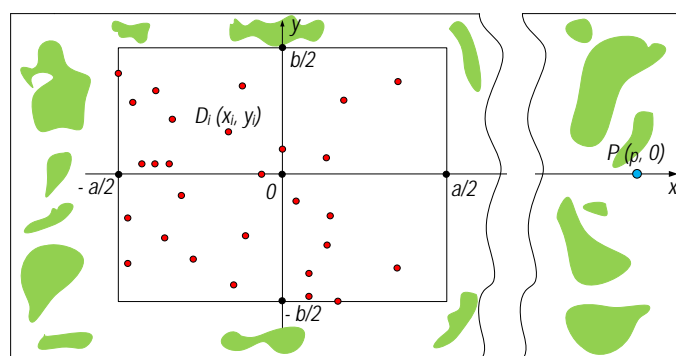


Рис. 2. Формализация задачи

Точку падения каждой ракеты, запущенной по  $i$ -ой цели, представим в виде двумерной случайной величины  $\Psi_i$  с координатами  $(\xi_i, \eta_i)$  (рис. 3).

Опыт артиллерийских стрельб и пусков ракет показывает [1], что координаты точки падения каждого снаряда противника являются двумерной случайной величиной, распределенной по нормальному закону с параметрами  $m_{\xi_i}, m_{\eta_i}, \sigma_{\xi_i}, \sigma_{\eta_i}, r_i$  с соответствующей плотностью распределения

$$f_{\Psi_i}(x, y) = \frac{1}{2\pi\sigma_{\xi_i}\sigma_{\eta_i}\sqrt{1-r_i^2}} \exp\left\{-\frac{1}{2(1-r_i^2)} \left[ \frac{(x-m_{\xi_i})^2}{\sigma_{\xi_i}^2} - \frac{2r_i(x-m_{\xi_i})(y-m_{\eta_i})}{\sigma_{\xi_i}\sigma_{\eta_i}} + \frac{(y-m_{\eta_i})^2}{\sigma_{\eta_i}^2} \right]\right\},$$

где  $m_{\xi_i} = x_i, m_{\eta_i} = y_i$  – математические ожидания одномерных случайных величин  $\xi_i$  и  $\eta_i$  соответственно;  $\sigma_{\xi_i} = \sigma_{\xi_i}(l_i), \sigma_{\eta_i} = \sigma_{\eta_i}(l_i)$  – среднеквадратические отклонения случайных величин  $\xi_i$  и  $\eta_i$  соответственно ( $l_i$  – расстояние от точки  $P$  до  $D_i$ );  $r_i = r_i(\alpha)$  – коэффициент корреляции величин  $\xi_i$  и  $\eta_i$  ( $\alpha$  – угол между отрезком  $PD_i$  и осью  $OX$ ). Случайная величина  $\Psi_i(\xi_i, \eta_i)$  в системе координат  $O'X'Y'$  является некоррелированной (рис. 3).

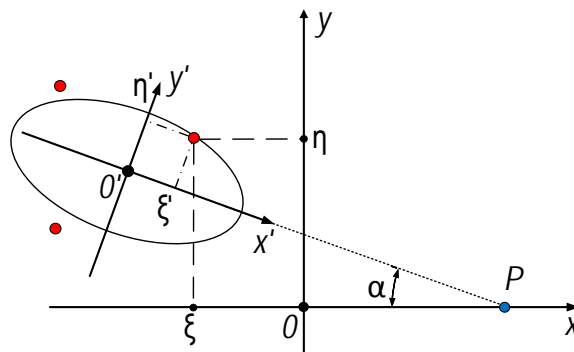


Рис. 3. Генерация обстрела для цели

Рис. 4,а иллюстрирует стрельбу по двум точкам  $D_i$  и  $D_k$  двумя ракетами. Реализация случайной величины  $\Psi_k$  располагается вблизи эллипса рассеивания случайной величины  $\Psi_j$  на расстоянии  $d_{ik}$  от его центра, тогда как реализация величины  $\Psi_j$  попала в эллипс и располагается на расстоянии  $d_{ji}$  от центра. Очевидно, что существует вероятность реализации случайных величин и в непосредственной близости от точки  $D_i$ , что означает влияние и «целевых» и «нецелевых» ракет на состояние любой  $i$ -ой цели. Для оценки этого влияния введем функцию ущерба  $U(d)$  цели от одной ракеты с областью значений  $(0, 1]$  (рис. 4,б). Здесь коэффициент  $K_{100\%}$  определяет радиус круга, при попадании в который цель будет полностью уничтожена, а кривая в области  $d \in (K_{100\%}, \infty)$  прямо пропорциональна величине  $\frac{1}{d^2}$ .

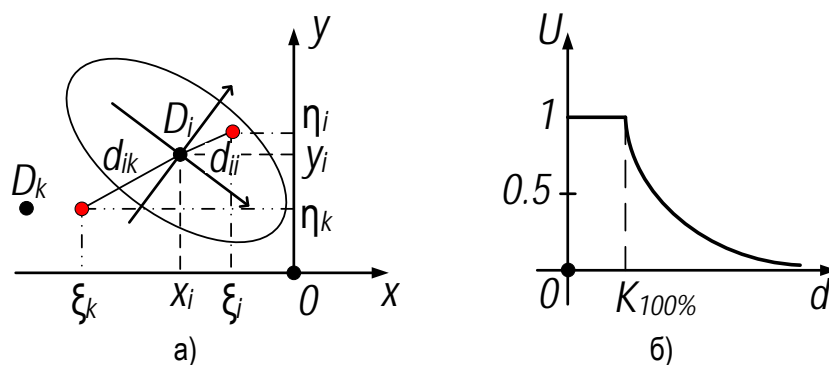


Рис. 4. Иллюстрация стрельбы (а) и функция ущерба (б)

Результирующий ущерб, нанесенный  $i$ -ой цели на  $j$ -ом испытании от всех  $n$  выпущенных ракет, будем оценивать аддитивно с помощью выражения

$$U_{ij} = \begin{cases} 1, & \text{если } \sum_{k=1}^n U(d_{ik}) \geq 1, \\ \sum_{k=1}^n U(d_{ik}), & \text{иначе.} \end{cases}$$

Средний ущерб, нанесенный  $i$ -ой цели, по результатам  $s$  испытаний, определяется выражением

$$U_i = \frac{\sum_{j=1}^s U_{ij}}{s},$$

тогда максимальный средний ущерб среди всех целей будет равен

$$U_{\Sigma} = \max_{i \in 1..n} \{U_i\}.$$

Если имеется  $m$  схем размещения подразделений на местности, и этим схемам соответствуют элементы индексного множества  $W$  ( $|W| = m$ ) и максимальный средний ущерб  $U_{\Sigma i}$  ( $i \in W$ ), то исходную задачу можно сформулировать так: найти индекс  $j \in W$ , такой что  $U_{\Sigma j} = \min_{i \in W} \{U_{\Sigma i}\}$ . Схема размещения с индексом  $j$  подразделений будет в этом случае наиболее безопасной с точки зрения минимизации максимального среднего ущерба.

Решение задачи. Поставленная задача решается методами статистического моделирования при помощи ЭВМ [2]. На рис. 5 представлен алгоритм моделирования обстрела и расчета максимального среднего ущерба.

Блок 1 предназначен для ввода всех исходных данных – координат целей, координат противника, параметров плотности распределения случайных величин, параметров функции ущерба. Блоки 2, 3, 4 предназначены для реализации метода статистических испытаний с целью расчета средних ущербов для каждой поражаемой цели. Блок 5 реализует поиск максимального среднего ущерба среди всех целей.

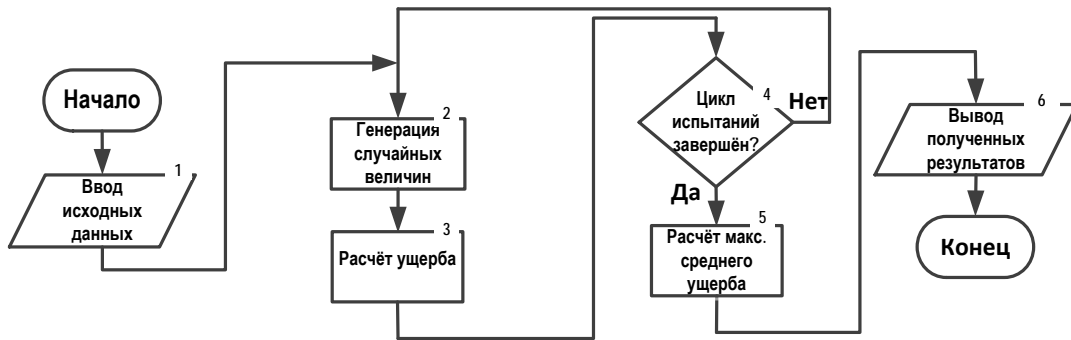


Рис. 5. Алгоритм моделирования

Вычислительный эксперимент. На участке с параметрами  $a = 10$  км и  $b = 8$  км рассмотрим четыре схемы размещения подразделений РТГ на местности в количестве 21 единицы ( $n = 21$ ): 1 – равномерная схема (рис. 6,а), треугольная схема (рис. 6,б), случайная схема (рис. 6,в), смещенная схема (6,г).

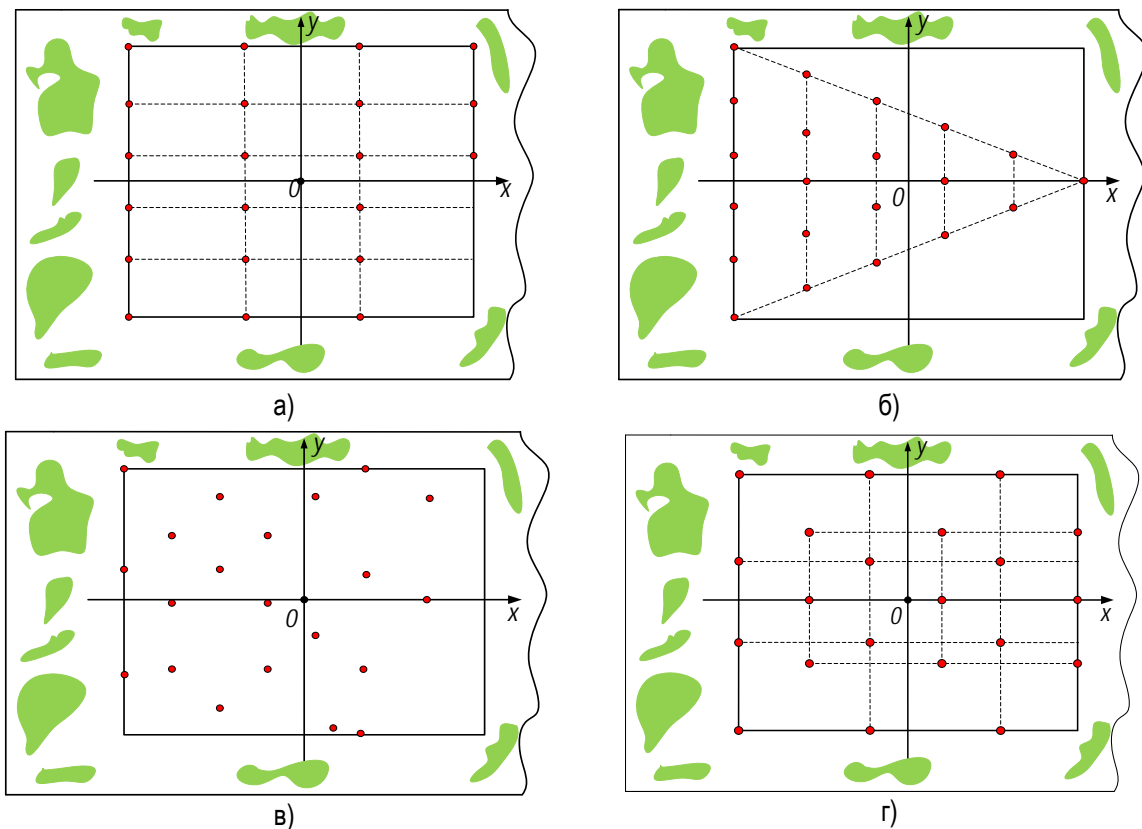


Рис. 6. Схемы размещения подразделений: равномерная (а), треугольная (б), случайная (в), смещенная (г)

Исходные координаты средства огневого поражения противника в километрах –  $P(40,0)$ . Параметры распределения:  $\sigma_{\xi_j} = 0,005/l_j$ ,  $\sigma_{\eta_j} = 0,005/l_j$ . Функция ущерба для каждой цели определяется выражением  $U(d) = \frac{0,01}{d^2}$ .

Округленные до целого результаты решения задачи представлены гистограммами на рис. 7.

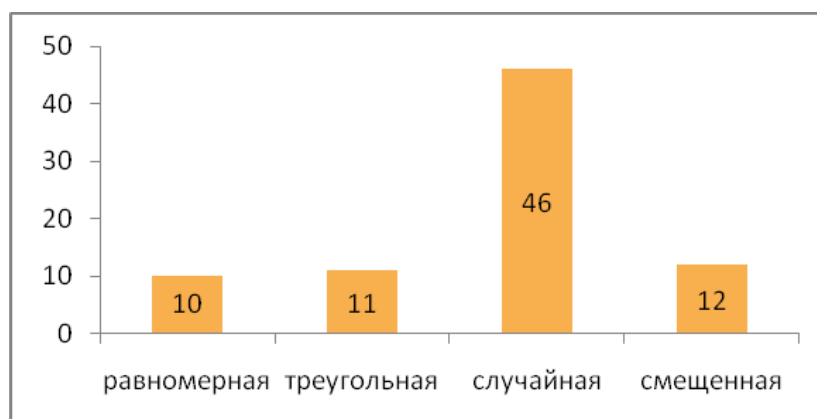


Рис. 7. Результаты решения задачи (%)

Выводы. Анализ полученных результатов показывает, что при распределении сил и средств на местности необходимо исключать случайность и руководствоваться результатами оперативных расчетов. В нашем примере из четырех рассмотренных схем в случае равномерного размещения получаем наименьший вероятный ущерб (10%).

#### Библиографический список

1. Разоренов Г. Н. и др. Системы управления летательными аппаратами: учебник для вузов. М.: Машиностроение, 2003. 584 с.
2. Ермаков С. М., Михайлов Г. А. Статистическое моделирование. М.: Наука. Главная редакция физико-математической литературы, 1982. 296 с.

УДК 681.3.053

**Д. С. Рожко**

студент кафедры компьютерной математики и программирования

**А. В. Яковлев**

канд. техн. наук, доц. – научный руководитель

### ИСПОЛЬЗОВАНИЕ ГИБРИДНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ В ПОМЕХОУСТОЙЧИВОМ КОДИРОВАНИИ. КОД РИДА-СОЛОМОНА

Передача данных происходит в физической среде, которая априори не может быть абсолютно надежной. Кроме того, в процессе передачи могут возникать шумы, помехи и случайные ошибки. Для надежности передачи данных были созданы помехоустойчивые коды, которые позволяют определить наличие ошибок в полученных данных и даже их исправить [1]. Большинство помехоустойчивых кодов, используемых на сегодняшний день, имеют высокую сложность вычислений. Следовательно, при их использовании в процессе кодирования и декодирования информации требуется большая вычислительная мощность [2].

Повышение производительности вычислительных систем довольно долгое время происходило путем улучшения характеристик центральных процессоров, а не путем изменения подходов

к структуре самих систем. Данное решение имеет свои ограничения, и как мы видим на сегодняшний день, производительность вычислительных систем с центральными процессорами практически перестала расти. Основные изменения происходят только в уменьшении тепловыделения и энергопотребления. Поэтому все будущие улучшения производительности вычислительных систем могут происходить только за счет увеличения параллельности и использования большого числа простых процессоров, работающих на низкой частоте. Следовательно, кардинально новым подходом к архитектуре вычислительных систем, позволяющим существенно увеличить вычислительную мощность, служит создание гибридных вычислительных систем.

Исходя из этого – исследование возможности использования гибридных вычислительных систем, как одной из наиболее эффективных архитектур для высоконагруженных вычислений, в области помехоустойчивого кодирования является актуальным.

В данном докладе будет рассмотрена возможность применения гибридных вычислительных систем в задачах кодирования и декодирования данных, а также эффективность такого решения, на примере кодов Рида-Соломона. Данные коды обладают огромной популярностью, благодаря тому, что они являются оптимальными с точки зрения соотношения длины пакета и возможности исправления ошибок, а также благодаря способности исправлять многократные пакеты ошибок [3]. Они нашли свое применение в различных стандартах кодирования информации как внешние коды и применяются в следующих областях: запоминающие устройства (жесткие диски, магнитные ленты, CD/DVD, штрих коды), беспроводная и мобильная связь (сотовые телефоны, микроволновая связь, мобильные модемы), спутниковая связь, цифровое телевидение, высокоскоростные линии связи (ADSL, xDSL, ВОЛС).

В данном исследовании использовался код Рида-Соломона в поле Галуа  $GF(2^8)$ , имеющий в каждом блоке данных 239 информационных символов ( $k$ ), каждый из которых кодирует один байт информации, и 16 символов избыточности. Всего символов в блоке ( $n$ ) 255. Код с данными характеристиками исправляет до 8-ми случайных ошибок.

В данном исследовании использовался алгоритм кодирования информации, который можно представить следующим образом [3, 4]:

- к исходному слову добавляется избыточность, содержащая 16 нулей. Получается полином:  $v(x) = u(x)x^{16}$ , где  $v(x)$  – полином, сопоставляемый кодируемому слову  $v$  с избыточностью;  $u(x)$  – полином, сопоставляемый кодируемому слову;

- производится прямое преобразование:  $V_j = \sum_{i=0}^{n-1} a^{ij} v_i = v(a^j)$ , где  $V$  – элемент кодового слова;  $v$  – элемент кодируемого слова;  $a$  – примитивный элемент поля Галуа, в моем случае – 2.

Алгоритм декодирования можно представить следующим образом [3, 4]:

- производится обратное преобразование  $v_i = \sum_{j=0}^{n-1} a^{-ij} V_j = v(a^{-i})$ , где  $v$  – элемент кодируемого слова;  $V$  – элемент кодового слова;  $a$  – примитивный элемент поля Галуа, в моем случае – 2;

- проверяются элементы, являющиеся избыточностью, называемые синдромом ошибки ( $S$ );

- если синдром ошибки не равен 0, то строится система уравнений в матричной форме для нахождения локатора ошибки ( $L$ ):  $MV = L$ , где  $M$  – матрица коэффициентов ( $M_{ij} = S_{t-1+i-j}$ );  $V$  – вектор свободных членов ( $V_i = S_{t+i}$ );  $t$  – количество ошибок;

- находится маска ошибки ( $f$ ) методом Форни:

- в позиции с  $k$  по  $n-1$  маски ошибки записывается синдром ошибки;

- производится вычисление остальных элементов на позициях с  $k-1$  по 0 по следующей формуле:



$$f_i = \left( \sum_{j=0}^{t-1} f_{i+1+j} / t_{-1+j} \right) / l_t,$$

где  $l$  – элемент локатора ошибок  $L$ ;

- из декодированного сообщения вычитается маска ошибки.

В ходе выполнения данного исследования была создана программа для кодирования и декодирования данных кодами Рида-Соломона. Данные генерировались байтами с диапазоном значений от 1 до 239, а кодировались и декодировались сначала при помощи центрального процессора, а затем и с задействованием графического. При каждом кодировании и декодировании замерялось время работы алгоритма. Результаты измерений заносились в файл. После декодирования полученные данные сравнивались с исходными.

Все данные при кодировании и декодировании кодами Рида-Соломона обрабатываются блоками. Поэтому весь эксперимент проводился с разным количеством блоков: 100, 300, 500, 1000, 3000, 5000, 10 000, 13 000, 20 000, 30 000, 40 000. Наименьшее количество блоков было выбрано в соответствии с тем, что обработка данных размером меньше 100 блоков обладает низким параллелизмом, а значит, не имеет смысла использование для этих целей графического процессора. При этом данные размером более 40 000 блоков могут обрабатываться на графическом процессоре более двух секунд, что вызовет срабатывание механизма Timeout Detection and Recovery (TDR) и вычисления сбросятся.

Эксперимент проводился на следующем оборудовании:

- центральный процессор Intel Core I5-2430M;
- видеокарта NVIDIA GeForce GTX 550 Ti.

Каждый раз вычислялось время выполнения алгоритма. Все полученные результаты были проанализированы и построены графики зависимости времени кодирования (рис. 1) и декодирования (рис. 2) данных от их количества для центрального и графического процессора.

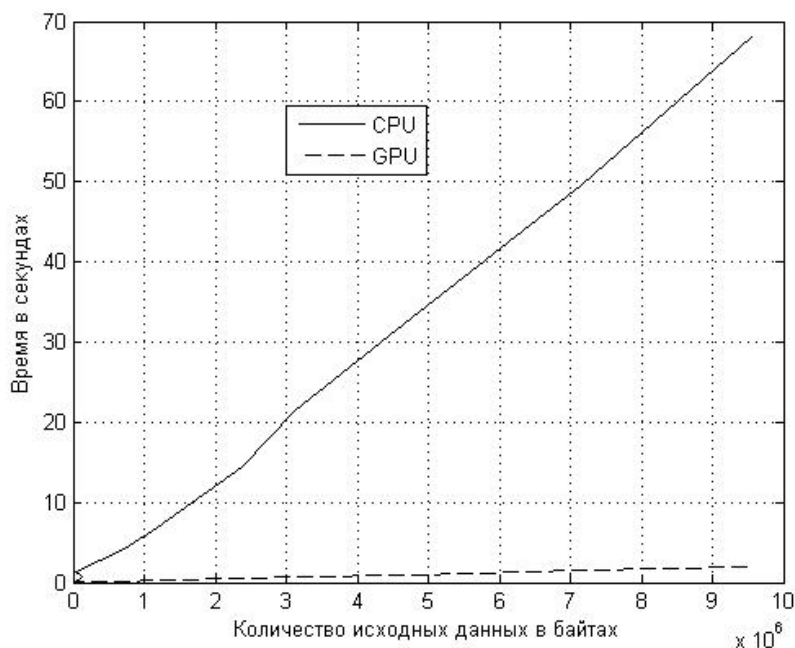


Рис. 1. Зависимость времени кодирования данных от их количества

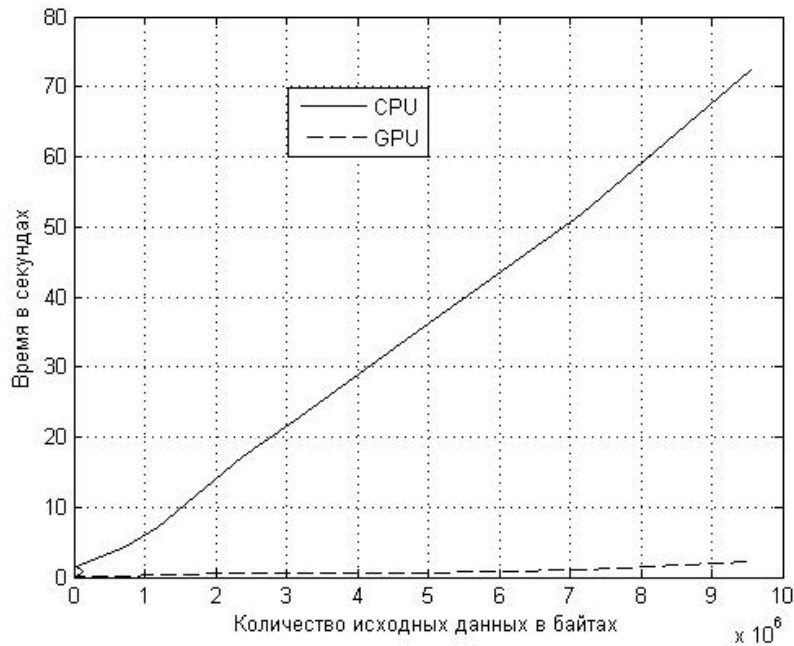


Рис. 2. Зависимость времени декодирования данных от их количества

Как видно из графиков, время кодирования (рис. 1) и декодирования (рис. 2) данных с помощью графического процессора значительно ниже, чем с помощью центрального. Стоит отметить то, что чем больше количество данных, тем больше абсолютное значение разницы времени решения задач. Это объясняется тем, что на большем объеме данных удастся равномерно распределить нагрузку на графическом процессоре. При кодировании данных прирост производительности при использовании графического процессора составил от 25 до 34 раз, а при декодировании от 18 до 50. При этом время решения данных задач сократилось в среднем на 17.63 и 18.82 с соответственно.

Полученные данные свидетельствуют о том, что использование гибридных вычислительных систем в задачах помехоустойчивого кодирования позволяет существенно повысить скорость выполнения этого процесса. Результаты данного исследования могут быть применены для реализации помехоустойчивого кодирования кодами Рида-Соломона с использованием возможностей гибридных вычислительных систем.

### Библиографический список

1. Золотарев В. В., Овечкин Г. В. Помехоустойчивое кодирование. Методы и алгоритмы. М.: Горячая линия – Телеком, 2004. 126 с.
2. Мясин В. В., Шашков Б. Д. Эффективность параллельной реализации алгоритмов помехоустойчивого кодирования Рида-Соломона // Прикладная информатика. 2006. № 3. С. 120–129.
3. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение: Пер. с англ. М.: Техносфера, 2005. 320 с.
4. Вернер М. Основы кодирования: Пер. с англ. М.: Техносфера, 2004. 288 с.

**Д. А. Смолиенко**

студент кафедры компьютерной математики и программирования

**П. А. Степанов**

научный руководитель

### ТРАНСЛЯЦИЯ ВИЗУАЛЬНЫХ ЯЗЫКОВ

Несмотря на то, что уже имеется большое количество различных визуальных языков, новые продолжают появляться. Для этих новых языков становится актуальной задача создания инструментальных средств их поддержки – редактирования и исполнения (трансляции). В виду того, что разработка редактора под каждый конкретный язык является очень дорогой, логичным решением является создание универсального редактора, настраиваемого на язык. Таким средством является программа Diagen [1].

В этом средстве для описания синтаксиса и семантики визуального языка предлагается формализм на основе семантической сети. Пример изображения прямоугольника с перемещаемыми краями в таком подходе изображен на рис. 1.

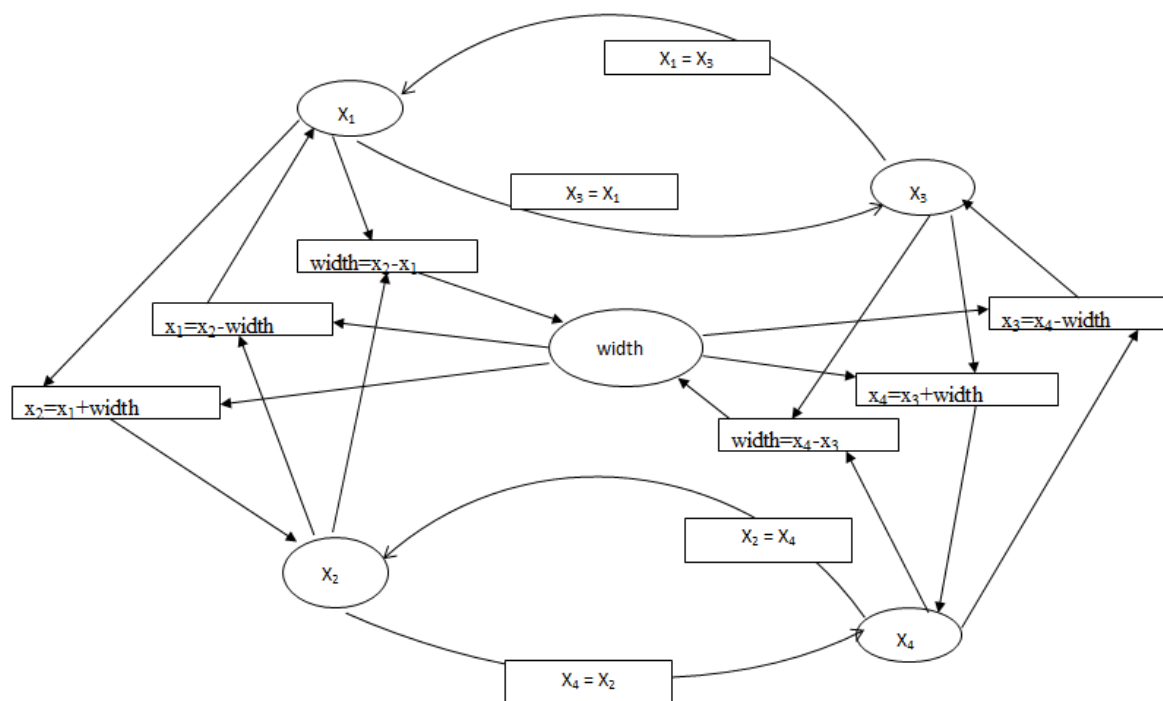


Рис. 1. Пример семантической сети прямоугольника с перемещаемыми краями

Диаграмма представляется в виде концептов-значений, между которыми установлены математические соотношения. Визуальный синтаксис такой диаграммы обеспечивается за счет сопоставления значения в концепте с некоторым базовым элементом визуального алфавита (текст, отрезок, окружность и пр.), взаимодействие пользователя с диаграммой представляет собой изменение значений концептов или топологии семантической сети, в ответ на которое происходит пересчет сети по некоторому алгоритму и дальнейшая перерисовка.

Очевидным образом подобный диаграммер будет неполным по своему функционалу, если он не сможет производить трансляцию визуальных языков. Именно такая задача была поставлена в этой работе. Для возможности транслирования диаграммы в текстовые языки был разработан специальный метаязык для описания хода трансляции.

Метаязык основывается на LL(1)-грамматике. LL(1)-грамматики очень распространены, потому что соответствующие им LL-анализаторы просматривают поток только на один символ вперед при принятии решения о том, какое правило грамматики необходимо применить. Причина привлекательности синтаксического анализа LL(1) заключается в его естественности, данный метод является крайне наглядным и удобным для создания основы для последующей компиляции языка программирования [2]. Приведем небольшой пример правил для описания оператора if:

- 1) P ~ if(A)O
- 2) A ~ !значениеB | значениеB
- 3) B ~ ==A | !=A | <A | >A | >=A | <=A | &A | e
- 4) O ~ TL
- 5) T ~ {K} | S
- 6) L ~ elseT | e

(K и S – блок действий, который необходимо выполнить при истинности или ложности условия.). Данная грамматика является LL(1) потому что операции, подобные < и <= разбираются парсером как единая лексема.

Такого рода правила позволяют нам создать специальный интерпретатор, основанный на методе рекурсивного спуска. Суть метода заключается в том, что при определенных свойствах LL(1)-грамматики в любой момент транслятор по следующему символу на ленте может понять, какое именно производящее правило должно применяться следующим. Сам интерпретатор состоит из трех частей – лексического и синтаксического анализатора и собственно исполняющей машины. Такая структура крайне проста и, вместе с тем, достаточно эффективна, хотя и не может обладать высокими показателями производительности. Лексический анализатор предлагается строить на основе JFlex – Java-порта GNU-версии лексического анализатора Lex. Этот анализатор позволяет очень быстро строить парсеры, если известны регулярные выражения, с помощью которых можно распознавать лексемы языка.

В языке реализовано пять основных типов данных: числовой, строковый, логический, объект и список. Для простых типов (числовой, строковый, логический) в языке предусмотрены следующие арифметические операции: сложение «+», вычитание «-», умножение «\*», деление «/».

В языке восемь операций сравнения и операция отрицания:

- 1) больше > ;
- 2) меньше < ;
- 3) больше или равно >= ;
- 4) меньше или равно <= ;
- 5) равно == ;
- 6) не равно != ;
- 7) логическое И & ;
- 8) логическое ИЛИ | ;
- 9) отрицание ! .

Логические операции 1–6 можно использовать как с логическим типом данных, так и с числовым и строками.

Язык содержит в себе следующие операторы: оператор определения переменной var, оператор блока { }, оператор условия if-then-else, оператор цикла foreach и операторы вывода в консоль write и writeln. В виду того, что основной задачей является преобразование диаграммы в текст, этот набор операторов представляется достаточным.

Для объявления переменных существует оператор `var`. Заметим, что тип переменной указывать не обязательно. После того, как интерпретатор посчитает выражение справа от знака «=» автоматически присвоит переменной тип этого выражения.

Синтаксис оператора выглядит следующим образом:

- 1) `<оператор var> ::= var<переменная>{=<выражение>};`
- 2) `<выражение> ::= <переменная> | <строка> | <число> | <объект> | <функция> | <список>;`

Оператор `foreach` позволяет организовывать цикл по элементам списка. Для трансляции диаграммы обычно необходимо выделить некоторую группу визуальных объектов и разом преобразовать их свойства в текстовый язык. Оператор `foreach` хорошо подходит для этих целей.

Синтаксис оператора выглядит следующим образом:

- 1) `<оператор foreach> ::= foreach(<условие>) <{>< блок операторов><}>;`
- 2) `<условие> ::= <переменная>:<список>;`
- 3) `< блок операторов > ::= <> | <оператор><блок операторов>;`

Конструкция `foreach` не требует ручного изменения переменной-шага для перебора – цикл автоматически выполняет эту работу. Для того чтобы определять, сколько еще элементов осталось в цикле, оператор `foreach` экспортирует в контекст переменную `Selector`, содержащую список непросмотренных элементов.

Операторы `write` и `writeln` предназначены для вывода информации в консоль (т. е. в результат). Операторы позволяют выводить как символьную, так и числовую информацию.

Синтаксис описывается следующим образом:

- 1) `<оператор write> ::= write(<выражение>);`
- 2) `<выражение> ::= <переменная> <строка> | <число>;`

Другим весьма важным оператором является оператор ветвления – `if`. Оператор `if` позволяет программе в зависимости от условий выполнить оператор или группу операторов, основываясь на значении булевой переменной или выражения. Оператор `if` начинается с ключевого слова `if`, которое должно сопровождаться булевым выражением, заключенным в скобки. Синтаксис описывается следующим образом:

- 1) `<оператор if> ::= if(<булево выражение>)<{><блок операторов><}> [else<{>< блок операторов ><}>];`
- 2) `< блок операторов > ::= <> | <оператор><блок операторов>;`
- 3) `<булево выражение> ::= <булево выражение><булево выражение> | <выражение><логический оператор><выражение>;`
- 4) `<логический оператор> ::= <|> | == | != | > | < | <= | >= | & | <|>;`
- 5) `<выражение> ::= <переменная> | <строка> | <число>;`

Булево выражение принимает значение `true` или `false`. Если условие истинно, то оператор или группа операторов выполняется, если ложно, то оператор не выполняется. Очень часто булево выражение в операторе `if` содержит какое-нибудь сравнение, но можно использовать булеву переменную или константу. Оператор `if` продолжается заключенным в фигурные скобки фрагментом, который называют блоком операторов.

Синтаксис оператора:

- 1) `< оператор {> ::= <{>< блок операторов ><}>;`
- 2) `< блок операторов > ::= <> | <оператор><блок операторов>;`

Приведем пример трансляции ER-диаграммы с использованием описываемого языка:

```
foreach (X:getObjects(type=="entity"))
{
    write("create table ");write(name);
    write("(");writeln("");
    var attribute = X.getAttribute("content");
    foreach(Y:attribute){
```

```

write(Y.getListElement(2)); write(" ");
write(Y.getListElement(1));write(" ");
write(Y.getListElement(0));
if (selector.hasNext()){
    writeln(","); };
}
writeln("");writeln("go");
}
foreach (X:getObjects(type=="connection"))
{
write("alter table ");
var toname = X.getAttribute("to");
var fromname = X.getAttribute("from");
var fkfield = X.getAttribute("fkfield");
writeln(toname);write("add foreign key (");
foreach (Y:getObjects(name==toname))
{
var attribute = Y.getAttribute("content");
foreach(Z:attribute){
var modifier = Z.getListElement(0);
if (modifier == "primary key"){
write(Z.getListElement(2));
};
}
}
writeln("");write ("references "); write(fromname);write ("(");write(fkfield);
writeln ("");writeln ("go");
}
}

```

Данный код описывает трансляцию ER-диаграммы в SQL. После применения данного кода к некоторой диаграмме мы получим код на языке SQL, позволяющий создать таблицы и отношения между ними. Здесь мы перебираем в цикле не только сущности, но и содержимое каждой сущности с помощью вложенного цикла. В условии мы проверяем, следует ли за текущим полем еще одно, и если следует, то мы можем поставить запятую, если же нет, то запятая не нужна. Применяя данный код к простой диаграмме, состоящей из таблиц, как на рис. 2, мы получим код на языке SQL, описывающий эти таблицы.

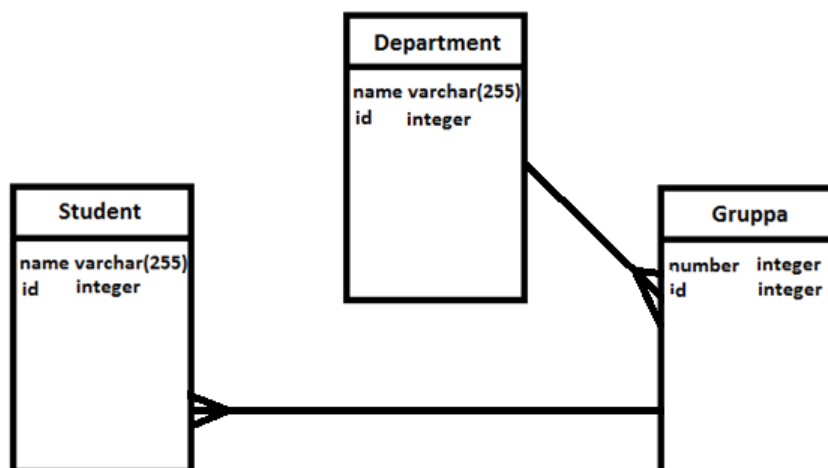


Рис. 2. Диаграмма с таблицами

Таким образом, была поставлена и реализована задача создания универсального транслятора для уже имеющегося универсального редактора диаграмм. Метод рекурсивного спуска, на основе которого сделан интерпретатор, способствует дальнейшему развитию языка и простому добавлению нового функционала.

### Библиографический список

1. Степанов П. А. Вычислительная модель визуального языка. Сб. науч. тр. / Под ред. Охтилев М. Ю. Изв. вузов. 2006. № 11.– С. 28–32.
2. Ахо А. В., Ульман Д. Д. Компиляторы: принципы, технологии и инструменты. Вильямс, 2003.

УДК 004.932

**А. Р. Сытин**

студент кафедры компьютерной математики и программирования

**С. А. Рогачев**

научный руководитель

## КЛАССИФИКАЦИЯ АЭРОКОСМИЧЕСКИХ ИЗОБРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ РАДИАЛЬНЫХ БАЗИСНЫХ НЕЙРОННЫХ СЕТЕЙ

Актуальность задачи классификации аэрокосмических изображений обусловлена тем, что данные дистанционного зондирования Земли могут быть использованы в условиях, когда невозможны наземные методы исследования, таких как получение данных о больших территориальных площадях, решение глобальных исследовательских задач. На данный момент сформулировано более 318 тематических задач, которые решаются с помощью данных дистанционного зондирования Земли. Анализ аэрокосмических снимков позволяет решать очень важные задачи, такие как мониторинг пожаров, анализ воздействия антропогенных объектов на окружающую среду и так далее.

Аэрокосмические снимки ведутся в различных частях электромагнитного спектра. Таким образом, космический снимок является серией снимков, сделанных примерно в одно и то же время для одного участка территории в различных спектральных зонах, поэтому такие снимки называются многозональными. Благодаря различной отражательной способности объектов на земной поверхности в различных спектральных каналах проявляются различные свойства сфотографированных объектов, предоставляя широкие возможности их классификации [1]. Комбинируя спектральные каналы как RGB-компоненты цветного изображения, можно как наглядно отобразить свойства объектов на мониторе, так и выделить различные свойства объектов для классификации. Например, разделить области с чистой и загрязненной водой.

Сеть радиальных базисных функций – это нейронная сеть, использующая радиальные базисные функции как функции активации. В общем случае функция активации в радиальных базисных сетях имеет вид [2]:

$$f_j(x) = \frac{x^2}{\sigma_j^2},$$

где  $f_j(x)$  – радиальная базисная функция нейрона;  $\sigma_j$  – стандартное отклонение, или ширина окна функции;  $x$  – входные значения.

Активационная функция имеет нормальное распределение, где вершина гауссианы – центр притяжения нейрона (рис. 1).

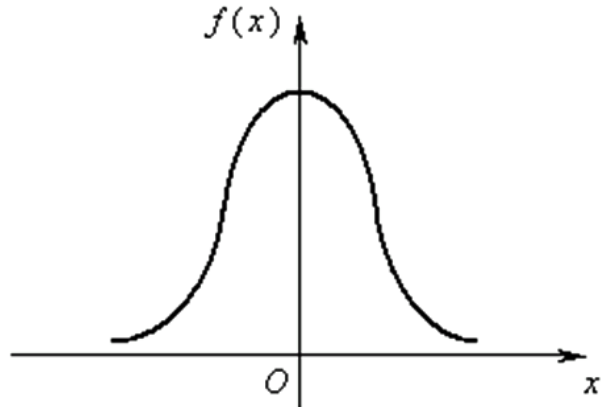


Рис. 1. График радиальной базисной функции

Радиальная базисная нейронная сеть имеет структуру, представленную на рис. 2.

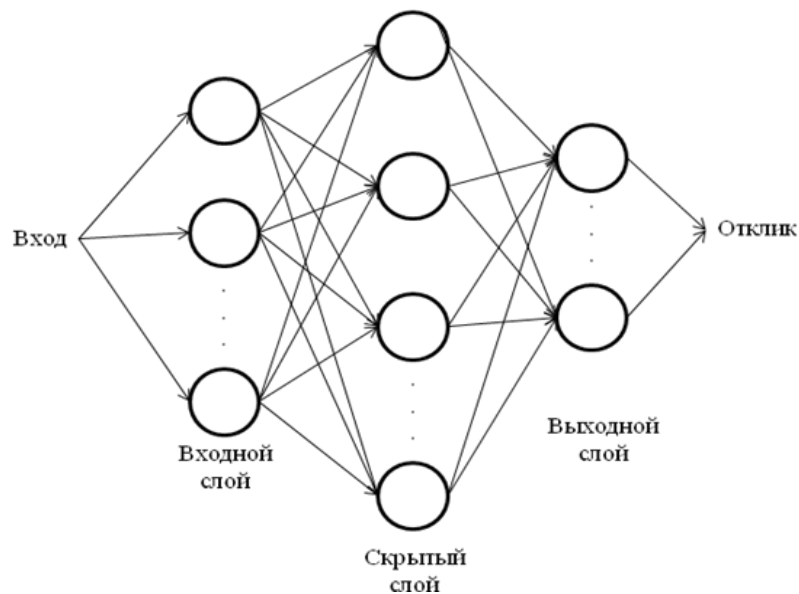


Рис. 2. Структура радиальной базисной нейронной сети

Входной слой является вектором входных значений, скрытый слой содержит один уровень скрытых нейронов. Данный слой выполняет нелинейное преобразование входного пространства в скрытое. Выходной слой сопоставляет преобразованное входное пространство с вектором выходных значений.

Было разработано ПО, позволяющее классифицировать аэрокосмические изображения с использованием радиальных базисных нейронных сетей. Обучение сети осуществляется методом обратного распространения ошибки, который заключается в том, что при каждой итерации обучения сети по входному значению происходит вычисление отклонения полученного значения от эталонного с последующей корректировкой значений сети в обратном порядке.

Разработанное ПО имеет возможность загрузки и сохранения аэрокосмического снимка и нейронной сети. Присутствует механизм выбора спектральных каналов снимка в директории



в реальном времени, а так же возможность кэширования. Программа имеет три режима работы, связанные с обучением, классификацией и проверкой качества классификации эталонными выборками. ПО позволяет настраивать параметры классов результирующего изображения, а так же предоставляет возможность расширенной настройки параметров сети (рис. 3).

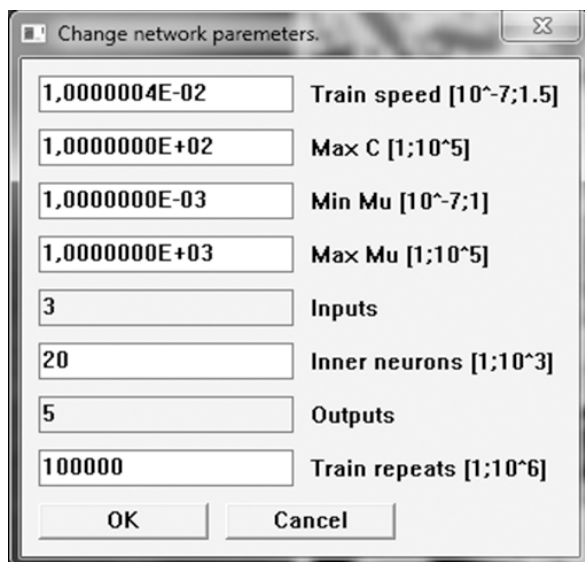


Рис. 3. Окно настройки параметров сети

Входными данными для обучения сети является список набора пикселей – примеров для каждого класса. Результатом классификации снимка Санкт-Петербурга в комбинации спектральных каналов спутника Landsat 8 является новое изображение с выделенными на нем различными классами объектов (рис. 4).

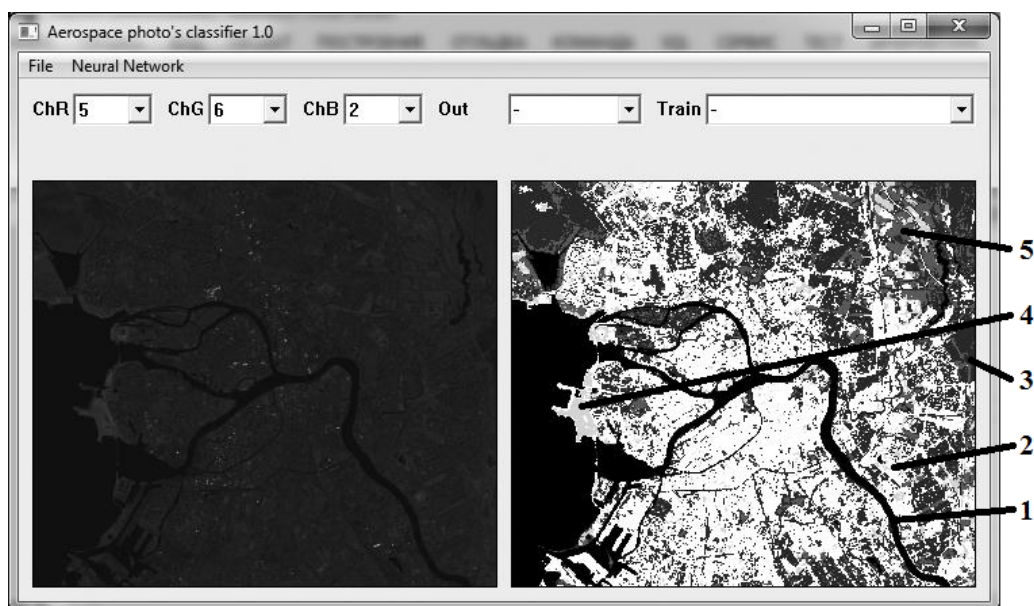


Рис. 4. Результат классификации снимка спутника Landsat 8 в комбинации 5, 6, 2.  
 1 – вода, 2 – антропогенные объекты, 3 – густая растительность,  
 4 – песчаные поверхности, 5 – редкая растительность

В результате было получено изображение, сохраненное в исходном формате без потери метаинформации. Качество классификации было проверено с помощью эталонных выборок. Для каждого класса объектов на изображении было проведено по 10 эталонных выборок.

Водные объекты были классифицированы наиболее точно. 10 из 10 выборок совпали, а наименьшая точность успешной классификации была равна 82.93%.

Густая растительность успешно классифицирована в 10 из 10 случаев. Наименьшая точность успешной классификации равна 76.42%.

Антропогенные объекты и песчаные поверхности были успешно классифицированы в 9 из 10 эталонных пикселях. Наименьшая успешная точность классификации находится на уровне 60%.

Редкая растительность была определена хуже всего. Совпало 10 из 10 значений, однако наименьшая успешная точность классификации равнялась 45.30%.

Поскольку классификация по цветовым компонентам заведомо не может быть абсолютно успешной, из-за спорных значений «граничных» значений пикселей у соседних классов, классификация аэрокосмических изображений с использованием радиальных базисных нейронных сетей имела неточности и небольшие ошибки.

Метод оказался весьма успешным и позволил достаточно точно классифицировать изображение на пять классов. Важно отметить, что точность классификации подобным методом напрямую зависит от качества подбора списка пикселей, использованных для обучения нейронной сети. Так же, результат может отличаться при разных попытках классификации изображения при одних и тех же параметрах сети и с одинаковым списком обучения, поэтому время, затрачиваемое на обучение системы, может варьироваться.

#### **Библиографический список**

1. Капралов Е. Г. Основы геоинформатики: в 2 книгах: учеб. пособие для студентов вузов / Е. Г. Капралов, А. В. Кошкарёв, В. С. Тикунов [и др.]; под ред. В. С. Тикунова. М.: Академия, 2004. – Кн. 1. – 352 с.
2. Рогачев С. А. Радиальные базисные нейронные сети в задачах классификации аэрокосмических изображений // Сборник трудов XXIII Международного научно-практического семинара "Современные технологии в задачах управления, автоматизации и обработки информации", г. Алушта, с. 82–83, М.: ИКД "Зерцало-М", 2014.

УДК 004.021

**А. С. Топчий**

студент кафедры вычислительных систем и сетей

**А. В. Аксёнов**

научный руководитель

### **ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ ДЛЯ РАЗРАБОТКИ АДАПТИВНЫХ ПРОТИВНИКОВ**

Генетические алгоритмы – это методы оптимизации, основанные на принципах естественного отбора Ч. Дарвина. С их помощью можно найти значение экстремума целевой функции, хотя при этом не всегда можно быть уверенным, что получено значение глобального экстремума, а не локального. Генетические алгоритмы довольно просты для реализации, могут быть использованы в задачах с изменяющейся средой и не требуют никакой информации о поведении целевой функции. Цель этой работы – исследовать возможность применения генетических алгоритмов для подстройки противников под поведение и тактику игрока. Поскольку заранее неизвестны навыки игрока, его стиль игры и его слабости, а также невозможно предусмотреть все варианты его поведения, генетический алгоритм – хороший кандидат на роль алгоритма адаптации противников.

Для демонстрации решения этой задачи с помощью генетического алгоритма была написана пошаговая игра (см. рис.), в которой на обычной квадратной карте расположены одинаковые существа, одним из которых игрок может управлять. Другие существа – его противники. Задача игрока – уничтожить всех противников и тем самым перейти на следующий уровень. При этом противники сами пытаются уничтожить игрока, приближаясь к нему на расстояние выстрела и атакуя. Поскольку у противников численное преимущество, игрок всегда ходит первым. Существа наносят урон огненной или ледяной стихией, для каждого из которых используется оружие ближнего и дальнего действия. Кроме того, у существ есть показатели защиты от каждого из типов оружия и урона, очки здоровья и очки действий. Все эти параметры ограничены как максимальным и минимальным значениями, так и очками ДНК, являющимися в этой игре аналогом уровня существа.

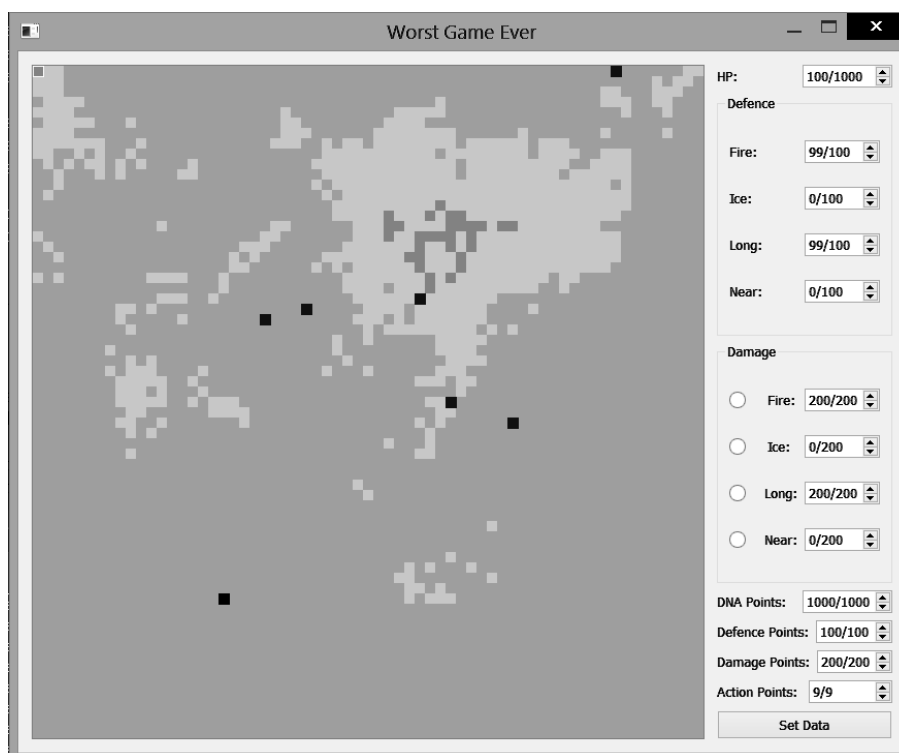


Рис. Интерфейс игры. Темные ячейки – существа, светлые – различные участки карты. Игрок управляет левым нижним существом. Справа задаются параметры игрока

Информация об этих параметрах хранится в массиве, который играет роль ДНК. Для изменения ДНК написаны специальные функции. Мутация прибавляет случайное значение к случайному параметру (гену) ДНК. В качестве мутации можно было бы использовать задание случайного значения гена или инверсию случайных битов, но эти методы не позволяют управлять силой мутации [1]. Кроссинговер делит две ДНК на части в произвольной точке между генами и меняет местами части разных ДНК. Он делит две ДНК строго между генами для того, чтобы избежать случайных больших мутаций при размножении [2]. Также можно случайным образом задать значения всех генов или задать конкретное значение конкретному гену.

В классическом генетическом алгоритме создается множество особей со своими ДНК, пригодность которых оценивается фитнес-функцией. В игре эта функция вычисляет, сколько урона каждый противник нанес игроку. Затем, в зависимости от значения урона, для каждого противника определяется, сможет ли он перейти на следующий уровень («выжить» в терминах теории эволюции). Противники, которые не смогли выжить, скрещиваются с выжившими особями (то есть происходит кроссинговер их ДНК), и случайный из двух потомков, пройдя через мутации,

занимает место неудачливого предка. Таким образом происходит естественный отбор [3]. К сожалению, такой алгоритм работает очень медленно: могут смениться многие поколения противников, прежде чем они адаптируются к игроку. Решением этой проблемы является использование гибридного алгоритма, который сочетает классический генетический алгоритм с неким другим способом решения задачи [4]. В каждом поколении все потомки оптимизируются этим способом и только потом заносятся в новую популяцию. Характер подстройки зависит от типа игры и имеющихся в распоряжении у алгоритма параметров. В этой игре оптимизационная функция перераспределяет очки защиты и урона противников в соответствии с очками урона и защиты игрока. Например, если у игрока в прошлом уровне были высокие показатели защиты от огня и низкие от льда, то у новых противников будут высокие показатели атаки льдом и низкие – огнем. При этом подстройка противников позволяет оптимально распределить очки атаки и защиты при заданных ограничениях для каждого потомка, а генетический алгоритм ищет оптимальные ограничения. Такая оптимизация позволяет уменьшить время реакции популяции на перераспределение очков игрока до 2–3 поколений.

Таким образом, генетические алгоритмы действительно можно использовать для подстройки противников под конкретного игрока, но при этом желательно использовать гибридные алгоритмы для большего быстродействия. Представленный в статье алгоритм можно улучшать и дальше. Например, установить для каждого гена вероятность его мутации в соответствии с его смыслом или желаемым изменением параметров противника, или добавить разделение генов на доминантные и рецессивные.

#### Библиографический список

1. Матвеев А. С. ТАУ-Дарвинизм – <http://habrahabr.ru/post/111075/>
2. Матвеев А. С. Концепции практического использования генетических алгоритмов – <http://habrahabr.ru/post/110419/>
3. Генетический алгоритм. Просто о сложном – <http://habrahabr.ru/post/128704/>
4. Панченко Т. В. Генетические алгоритмы / Под ред. Ю. Ю. Тарасевича Астрахань: Издательский дом «Астраханский университет», 2007. 87 с.

УДК 681.3.06

**А. В. Устинов**

студент кафедры компьютерной математики и программирования

**М. Ю. Охтилев**

д-р техн. наук, проф. – научный руководитель

### ПОДСИСТЕМА ВЗАИМОДЕЙСТВИЯ С ПОЛЬЗОВАТЕЛЕМ В СОСТАВЕ ПРОГРАММНОГО КОМПЛЕКСА ОБРАБОТКИ АНАЛИЗА ИНФОРМАЦИИ В АСУ СЛОЖНЫМИ ТЕХНИЧЕСКИМИ ОБЪЕКТАМИ

В современном мире для эффективного контроля и управления сложными техническими объектами используются автоматизированные системы управления технологическими процессами (АСУ ТП). Современные АСУ ТП – это сплав «умного железа» и интеллектуальной программной составляющей [1]. Программная составляющая играет важную роль и обеспечивает обработку и анализ данных, а также взаимодействие пользователя и системы. При разработке программной составляющей важной и неотъемлемой частью является создание графического пользовательского интерфейса (graphical user interface – GUI).

Для унификации элементов GUI и возможности расширения функционала в процесс проектирования была введена сущность графического элемента интерфейса – Shell (Оболочка). Shell представляет собой независимый строительный блок, элемент графического интерфейса, который содержит свойства, характеризующие его и может быть модифицирован: изменять свойства, состояния и др.

Итогом работы является разработанная подсистема и комплект документации по проектированию и реализации новых элементов графического интерфейса.

Программный комплекс обработки и анализа информации  
в АСУ сложными техническими объектами

Программный комплекс представляет собой набор взаимодействующих между собой подсистем, обеспечивающих функциональность комплекса. Ключевыми подсистемами комплекса являются:

- подсистема автоматизированной подготовки исходных данных и знаний (АПИД) – отвечает за преобразование исходных данных и знаний в формальное описание программной составляющей и алгоритмы решения задач. Формальное описание необходимо для дальнейшего преобразования в «Исполнительной системе»;
- подсистема AnServer – отвечает за обработку и анализ данных, является серверной частью «Исполнительной системы»;
- подсистема AnClient – отвечает за обмен данными между пользователем и подсистемой AnServer, является клиентской частью «Исполнительной системы».

Структурная схема программного комплекса представлена на рис. 1.

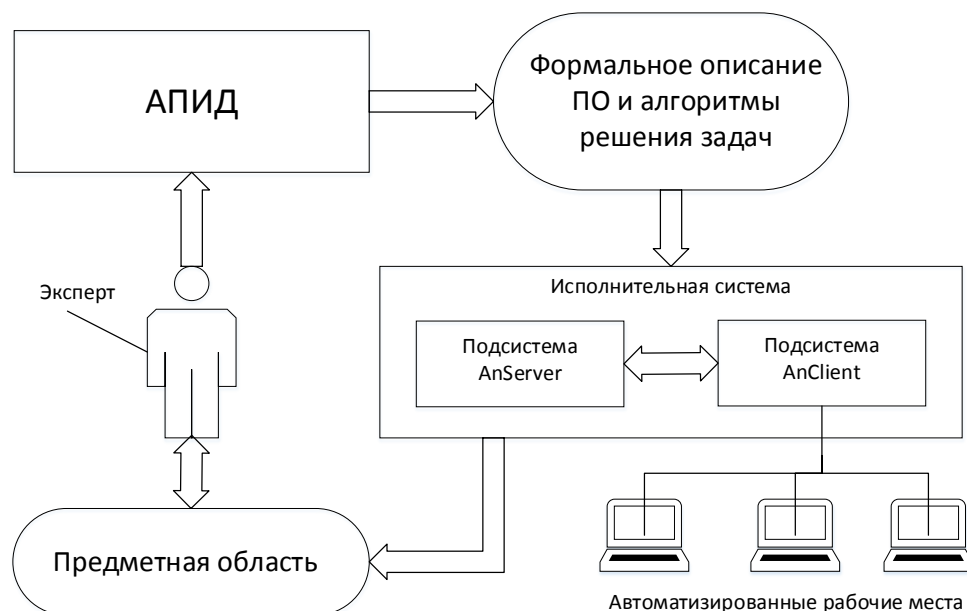


Рис. 1. Структурная схема программного комплекса

Подсистема «AnClient»

AnClient является программой-клиентом. Базируется на платформе JavaFX (The Rich Client Platform by Oracle) [2]. Выбор данной платформы был не случайным. JavaFX является кроссплатформенным инструментом разработки для создания функционально насыщенных приложений. Поддерживает работу в браузерах и непосредственный запуск из-под операционной системы.

Поддерживаются такие операционные системы, как OS Windows и OS Linux.

AnClient можно представить в виде двух взаимодействующих компонентов:

- подсистема «Взаимодействия с сервером» – ClientCore;
- подсистема «Взаимодействия с пользователем» – ClientGUI.

Взаимодействие между компонентами осуществляется посредством интерфейса обмена данными. Схема обмена данными внутри подсистемы AnClient представлена на рис. 2.

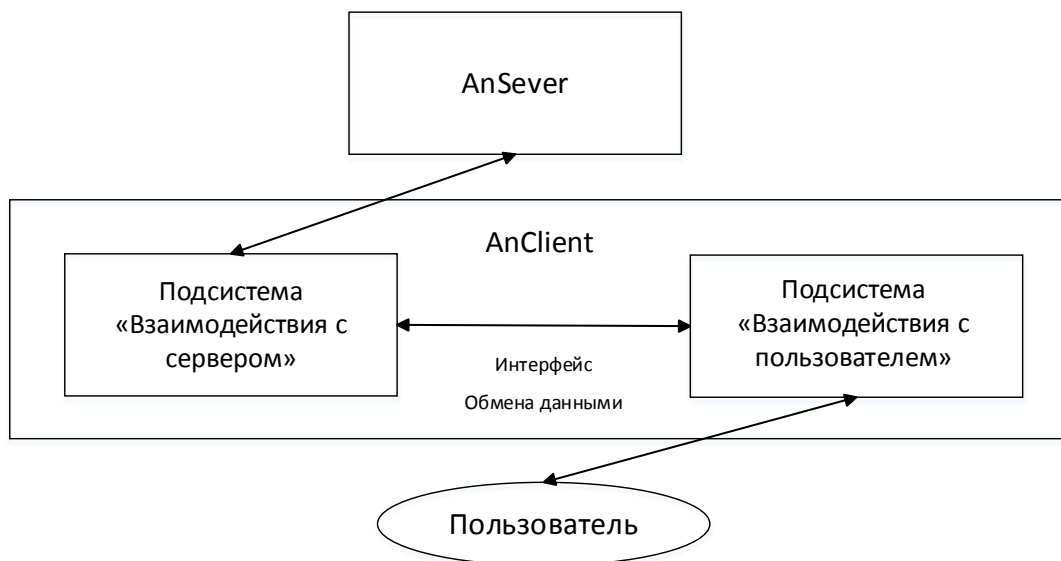


Рис. 2. Схема обмена данными между компонентами программы AnClient

#### Подсистема «Взаимодействия с пользователем»

Построение пользовательского интерфейса основано на преобразовании XML-документа, содержащего описание элементов, спроектированных в подсистеме АПИД, в GUI. Для унификации элементов GUI и возможности расширения функционала в процесс проектирования была введена сущность графического элемента – Shell (Оболочка). Shell представляет собой независимый строительный блок, элемент графического интерфейса, который содержит свойства, характеризующие его и может быть модифицирован: менять свойства, состояния и др.

#### Классификация и описание Shell

Shell является функционально расширенным примитивом графического интерфейса пользователя (виджета) [3], поэтому для анализа следует обратиться к описаниям существующих элементов интерфейса, предоставляемых используемой платформой JavaFX. Информация об элементах представлена в документации разработчика JavaFX API Documentation [4], в которой предлагается использовать такие элементы, как Контейнеры, Контроллеры (элементы управления, ввода и вывода информации), Графические примитивы (линия, круг и т. д.) и др. Часть этих элементов используется как базовые, для подсистемы ClientGUI. Каждый Shell имеет набор свойств, которые формируют внешний вид. Для большинства простых Shell, определяются такие параметры, как идентификатор, наличие или отсутствие графической составляющей, координата X, координата Y, размер по оси X, размер по оси Y, цвет, текст надписи.

Для специализированных Shell, нет необходимости иметь многие из перечисленных свойств, но для них требуется создать свои собственные свойства, которые делают уникальными такие элементы графического интерфейса. Таким образом, первый аспект классификации Shell – это состав свойств. Второй аспект – это поведение Shell – взаимодействие с другими элементами графического интерфейса, смена состояний, отправка внешней системе информации о событиях, произошедших в Shell. Третий аспект – это роль Shell. Shell может быть инструментальным (является элементом GUI) или сервисным, невидимым пользователю. Классификация элементов Shell приведена в табл. 1.

Классификация элементов Shell

Характеристика	Вид Shell
Категория	контейнеры контроллеры специальные элементы
Набор параметров	простой средний сложный
Модифицируемость	модифицируемый не модифицируемый
Роль	сервисный инструментальный

Каждый вид Shell из классификации описан в табл. 2.

Краткое описание Shell

Вид Shell	Описание
Контейнер	Группирует несколько других элементов интерфейса
Контроллер	Предоставляет элементы управления, ввода и вывода информации
Специальный элемент	Элемент не имеющий аналога в платформе, специально разработанный для конкретной задачи
Простой	Shell, имеющий менее 25 свойств
Средний	Shell, имеющий от 25 до 45 свойств
Сложный	Shell, имеющий более 45 свойств
Модифицируемый	Shell, имеющий параметры, которые могут быть модифицированы во время работы
Не модифицируемый	Shell, не имеющий параметры, которые могут быть модифицированы во время работы
Сервисный	Не имеет графической составляющей, однако имеет специфическую функциональность. В зависимости от каждого конкретного элемента набор свойств разный
Инструментальный	Стандартный Shell, имеющая графическую составляющую
Стандартный	Shell, основанный на элементе платформы JavaFX
Кастомизированный	Shell, вновь разработанный для решения определенных задач

### Реализация Shell

Внутренняя структура Shell имеет минимальный набор параметров, свойств, событий и алгоритмов обработки данных, предоставляя большие возможности модернизации элементов. На приведенной ниже иерархии классов (рис. 3), мы можем наблюдать набор базовых классов для разработки элементов GUI. Главным базовым классом является BaseShell, реализующий интерфейс IShell. Для реализации нового графического элемента интерфейса разработчику необходимо наследовать один из подклассов BaseShell: ToolBase (Инструментальный) или ServiceBase (Сервисный).

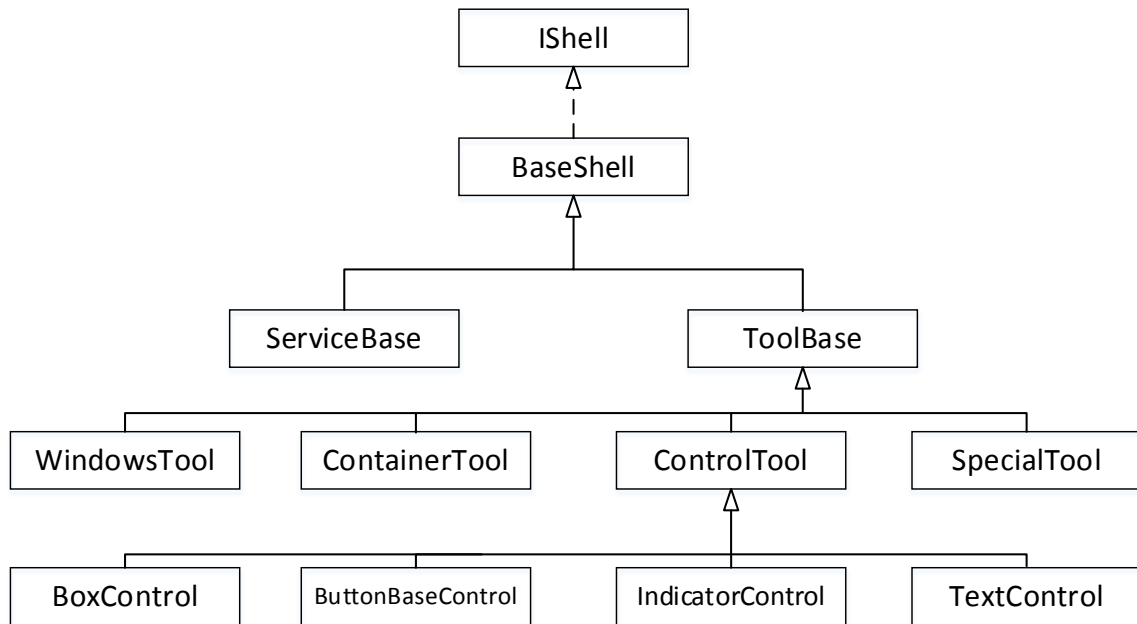


Рис. 3. Иерархия базовых классов Shell

Пример внешнего вида стандартных и кастомизированных Shell приведены на рис. 4. Кнопка, подпись и панель являются стандартными элементами платформы JavaFX, а панель с рамкой и контейнер состояний являются кастомизированными элементами интерфейса.

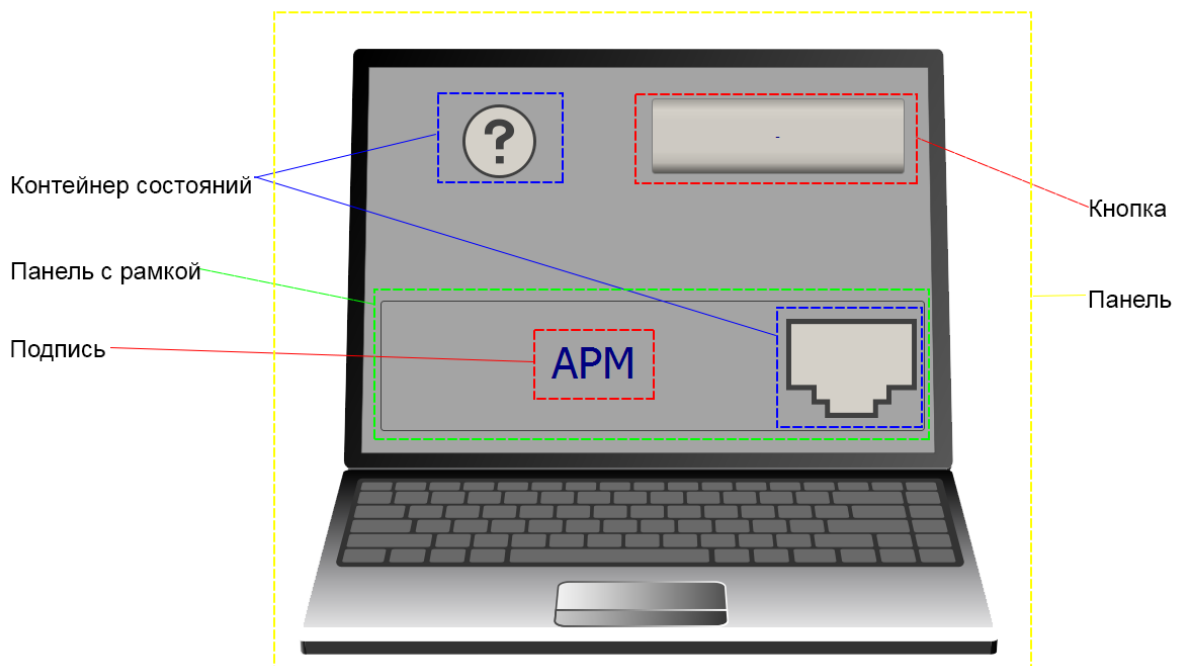


Рис. 4. Внешний вид Shell

Помимо графической части в составе подсистемы «Взаимодействия с пользователем» можно выделить логическую, которая отвечает за получение запросов на изменение параметров и отправку обратно события, произошедшего вследствие действий пользователя. Примером работы может служить следующая последовательность действий.



В некоторый момент времени AnServer инициирует отправку пакета с данными в AnClient, где через ClientCore передается непосредственно в ClientGUI. В этом пакете передается структура установки свойств для Shell, стандартизованная по протоколу взаимодействия. ClientGUI принимает этот пакет и в соответствии с запросом на установку свойства выполняет изменение значения свойства в указанном Shell. Через некоторое время пользователь-оператор нажимает на кнопку, инициировав тем самым событие, стандартизованная структура которого отправляется в пакете к ClientCore и далее в AnServer. Приняв и распознав данные, AnServer, в соответствии со своей логикой функционирования, отправляет в AnClient другой запрос на установку иного свойства. AnClient принимает пакет и модифицирует соответствующую структуру данных указанного элемента. Обмен пакетами между AnServer и AnClient продолжается. При этом текущие состояния элементов интерфейса (Shell) отображаются на экране.

### Заключение

В рамках данной статьи были рассмотрены основные аспекты подсистемы «Взаимодействия с пользователем». Все идеи, описанные в данной статье, были реализованы в рамках подсистемы ClientGUI, программного комплекса обработки и анализа информации в АСУ сложными техническими объектами, над которым работает СКБ «Орион» [1]. Проведенная унификация структур GUI и стандартизация разработки позволила повысить эффективность проектирования и реализации графического интерфейса. Разработка GUI на основе реализованной подсистемы, в будущем позволит повысить эффективность использования графических интерфейсов и расширить спектр применения программного комплекса.

### Библиографический список

1. Автоматизированные системы управления технологическими процессами СКБ «Орион» – URL: <http://www.skborion.ru>.
2. JavaFX - The Rich Client Platform by Oracle – URL: <http://www.oracle.com/technetwork/java/javase/overview/javafx-overview-2158620.html>.
3. Элемент интерфейса – URL: <https://ru.wikipedia.org>
4. JavaFX API Documentation – URL: <http://docs.oracle.com/javase/8/javafx/api/toc.htm>.

УДК 004.932

**А. Ю. Фролова**

студент кафедры проблемно-ориентированных вычислительных комплексов

**Г. С. Евсеев**

канд. техн. наук, доц. – научный руководитель

## ПРЕДОБРАБОТКА ИЗОБРАЖЕНИЯ ДЛЯ РАСПОЗНАВАТЕЛЯ ТЕКСТА

Два фактора определяют качество изображения – совершенство используемого печатного оборудования и качество оригинала. На изображении появляются лишние линии (вследствие изломов бумаги). Из-за дефектов принтера выпадают цветные точки, цвет и толщина шрифта непостоянны.

Главная задача обработки – повысить качество изображения: цвет фона и текста сделать постоянными, убрать искажения «соль и перец», устранить мелкие разрывы в линиях [1].

Можно выделить следующие этапы предобработки изображения:

- перевод в оттенки серого цвета;
- увеличение контрастности;
- фильтр для удаления искажения типа «соль и перец»;
- преобразование яркостной компоненты.

Перевод в оттенки серого цвета

Полноцветное изображение представляется массивом пикселей, каждый из которых описывается значениями красной, синей и зеленой составляющей.

Полутоновое изображение представляется массивом яркостей пикселей (яркость принимает значения от 0 до 255) [2].

При переводе полноцветного изображения в полутоновое убирается информация о цвете, которая является несущественной для распознавания текста, но усложняет работу с изображением. Указанный перевод (RGB → Y) выполняется по стандартной формуле  $Y = 0,299R + 0,587G + 0,114B$ , где Y – яркостная компонента.

Увеличение контрастности

Контрастность изображения определяется соотношением между наиболее яркими и темными участками изображения. На рис. 1 динамический диапазон яркостей от 30 до 200. Изображение в этом случае выглядит ненасыщенным, блеклым, неудобным в использовании [2].

При увеличении контрастности максимально расширяется диапазон яркостей и улучшаются условия для распознавателя (рис. 2). Повышение контрастности реализуется преобразованием яркостей по формуле

$$y = (x - P_{\min}) \frac{255}{(P_{\max} - P_{\min})},$$

где  $x$  – яркость пикселя в исходном изображении;  $y$  – яркость пикселя в полученном изображении;  $P_{\min}$  – минимальная яркость пикселя в исходном изображении;  $P_{\max}$  – максимальная яркость пикселя в исходном изображении.

Итог. рейтинг	Оценка	Дата
—	не эл.	18.01.14
—	не эл.	18.01.14
80	хорошо	18.01.14
90	отлично	18.01.14
65	удобн.	18.01.14
90	отлично	18.01.14
70	хорошо	18.01.14

Рис. 1. Исходное изображение

Итог. рейтинг	Оценка	Дата
—	не эл.	18.01.14
—	не эл.	18.01.14
80	хорошо	18.01.14
90	отлично	18.01.14
65	удобн.	18.01.14
90	отлично	18.01.14
70	хорошо	18.01.14

Рис. 2. Обработанное изображение

Применение Медианного фильтра

Процедуру медианной фильтрации можно пояснить на следующем примере (рис. 3). Имеется элемент Y, яркость которого существенно искажена. Фильтр создает выборку, включающую яркость текущего пикселя и яркости его соседей, после чего полагает восстановленную яркость пикселя равной медианному значению выборки (в данном случае это x4).

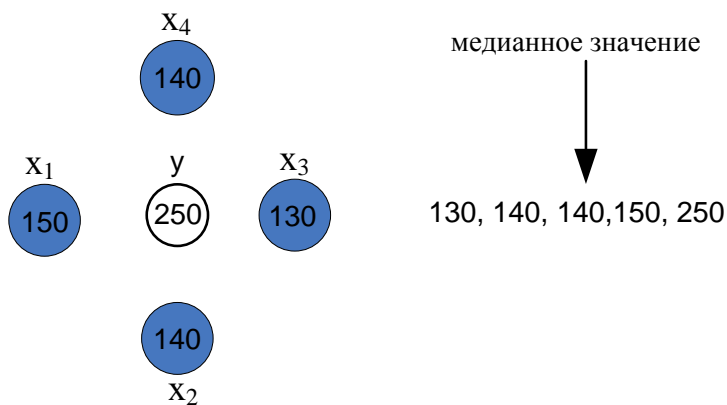


Рис. 3. Медианный фильтр

Когда искажения типа «соль и перец» затрагивают один или два пикселя, то по значению яркости соседей медианный фильтр всегда восстанавливает его яркость [2].

После увеличения контрастности в изображении ведомости появились лишние точки типа «перец», большая часть которых убирается медианным фильтром (см. рис. 4, рис. 5).

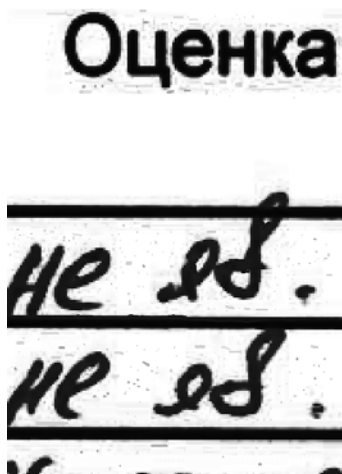


Рис. 4. Исходное изображение

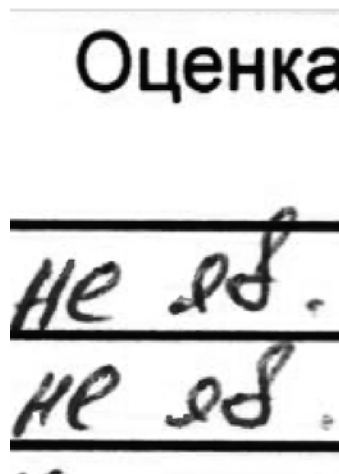


Рис. 5. Обработанное изображение

#### Преобразование яркостной компоненты

В полутоновом изображении яркость пикселей принимает значения от 0 до 255. Эта информация является лишней для распознавателя, ему только необходимо знать, к какой из трех областей относится пиксель: фон, линия бланка, вписанный текст. Поэтому достаточно иметь три уровня яркости, а не 256.

При выполнении указанного преобразования яркость каждого пикселя сравнивается с двумя порогами. Параметры  $P_1$  и  $P_2$  выбирались экспериментально.

1.  $A \leq P_1 \Rightarrow A = 0$
2.  $P_1 < A \leq P_2 \Rightarrow A = 100$ , где  $P_1 = 70$ ,  $P_2 = 200$ .
3.  $A > P_2 \Rightarrow A = 255$

После преобразования яркостной компоненты бланк стал абсолютно чистым, это упростило алгоритм распознавания, т.к. до преобразования яркость черных пикселей на линиях изменя-

лась от 0 до 70, яркость белых пикселей – от 200 до 255, яркость пикселей рукописного текста – от 70 до 200 (рис. 6, рис. 7).

Иг	Оценка	Дата
	не эф.	18.01.
	не эф.	18.01.
	Хорошо	18.01.
	Отлично	18.01.

Рис. 6. Исходное изображение

г	Оценка	Дата
	не эф.	18.01.
	не эф.	18.01.
	Хорошо	18.01.
	Отлично	18.01.

Рис. 7. Обработанное изображение

#### Заключение

Изображение на рис. 9 является входным для распознавателя текста.

Итог. рейтинг	Оценка	Дата
—	не эф.	18.01.
—	не эф.	18.01.
80	Хорошо	18.01.
90	Отлично	18.01.
65	удобн.	18.01.
90	Отлично	18.01.
90	Отлично	18.01.
90	Отлично	18.01.

Рис. 8. Исходное изображение

ИТОГ. рейтинг	Оценка	Дата
—	не эф.	18.01.
—	не эф.	18.01.
80	Хорошо	18.01.
90	Отлично	18.01.
65	удобн.	18.01.
90	Отлично	18.01.

Рис. 9. Обработанное изображение

В экспериментах по оценке эффективности предобработки было установлено, что на необработанных ведомостях распознаватель ошибался в несколько раз чаще, более того ему даже не всегда удавалось выделить область, где написана оценка.

#### Библиографический список

1. Горский Н., Анисимов В., Горская Л. Распознавание рукописного текста: от теории к практике. – СПб.: Политехника, 2009. – 126 с.: ил.
2. Шульга В. И. Основы обработки изображения, 2010 г.

УДК 681.3.06

**В. Г. Шалумов**

студент кафедры компьютерной математики и программирования

**М. Ю. Охтилев**

д-р техн. наук, проф. – научный руководитель

### ПРОГРАММНЫЙ КОМПЛЕКС ДИАГНОСТИРОВАНИЯ ЭЛЕМЕНТОВ ОБОРУДОВАНИЯ АСУ СТО

В настоящее время большое число различных предприятий используют дорогостоящее оборудование, которое в случае поломки может привести к большим финансовым убыткам. Как минимизировать в этой ситуации экономические потери из-за брака, простоев оборудования, устранения последствий аварийных отказов и, наконец, сократить расходы на текущие и плановые ремонты, объем которых часто неоправданно завышается?

Лучший выход – ввести в регламенты по обслуживанию оборудования его техническую диагностику. Но на большинстве предприятий, которые разворачивают деятельность в этом направлении, нет специалистов, готовых ее организовать при минимуме затрат и максимуме экономической эффективности.

Практическую помощь предприятия могут получить, обратившись к специалистам нового направления в науке и технике, получившего на западе название «Мониторинг состояния и диагностика машин и оборудования», занимающихся вопросами поддержания работоспособного состояния эксплуатируемых машин и оборудования между крупными ремонтами.

К типовым задачам по контролю состояния и диагностике машин и оборудования, решаемых отечественными предприятиями, а также к задачам по снижению затрат на их обслуживание и ремонт, как правило относятся:

- защита основного оборудования от аварий (предаварийная сигнализация);
- контроль состояния оборудования после обслуживания (ремонта);
- периодический контроль и прогноз состояния вспомогательного оборудования;
- постоянный контроль и периодический прогноз состояния основного оборудования.

В рамках диагностирования технического состояния элементов оборудования АСУ ПП (автоматизированной системы управления подготовкой и пуском) на унифицированном техническом комплексе (УТК) 14П83 требовалось разработать программный комплекс, позволяющий проводить следующие виды диагностики:

- диагностика сети (контроль подключения аппаратуры по сети);
- физические сигналы стойки;
- диагностика состояния источника бесперебойного питания;
- диагностика состояния переключателя фаз;
- диагностика питания (контроль наличия питания 220 В на фидерах питания).

Особое значение АСКД (автоматизированные системы контроля, диагностики, прогнозирования и мониторинга) имеют в жизненном цикле наукоемкой продукции, где многие происходящие процессы за пределами физиологических возможностей человека – аэрокосмонавтика, атомная энергетика, радиоэлектроника и другие.

Применение АСКД позволяет улучшить такие характеристики сложных технических изделий, как надежность, готовность к применению, способность выполнить задачу.

Эффективность использования АСКД, неразрывно связана с повышением научно-технического уровня их проектирования. На ранних этапах создания АСКД в основном базировалось на опыте и интуиции разработчиков, что приводило к аппаратной и программной избыточности систем, снижению других качественных показателей. Созданные на сегодня средства проектирования, элементная база, математические методы обработки информации, разработка мо-

делей, методов и алгоритмов позволяют создавать средства контроля и диагностики на самом высоком научно-техническом уровне и строить на их основе системы управления качеством, иметь мониторинг каждого изделия и электронный паспорт на него. Следует отметить, что существующие средства и методы контроля и диагностики далеко не всегда соответствуют современным требованиям пользователя и возможностям современной науки, техники и технологии и не всегда оказывают существенное влияние на качество изделий и услуг (человеческий фактор) [4].

На современном рынке присутствуют различные программные продукты для автоматизации процессов мониторинга, которые обладают собственными особенностями и преимуществами, например:

Scada система PcVue (рис. 1) – полнофункциональный продукт, который включает все последние достижения в области программного обеспечения класса SCADA для задач распределенного мониторинга и управления. Он обеспечивает гибкое решение для контроля промышленных процессов, энергоносителей и инфраструктуры. Он отвечает промышленным стандартам надежности и производительности, сохраняя при этом удобство для пользователей. Он охватывает требования от однопользовательских приложений до сложных клиент-серверных проектов с резервированием [5].



Рис. 1. Пример работы программы PcVue Solutions

АПИД (рис. 2) – эта автоматизированная система подготовки исходных данных, которая предназначена для занесения данных об измерительной информации произвольной природы, алгоритмах, моделях и т.п. В базу данных, просмотра и корректировки данных, содержащихся в БД и описывающих объект анализа, в качестве которого могут рассматриваться отдельные системы и агрегаты в целом. Основной функцией программы АПИД (программного компонента «Операционная среда») является подготовка исходных данных (ИД) для проведения автоматизированного анализа измерительной информации (ИИ).

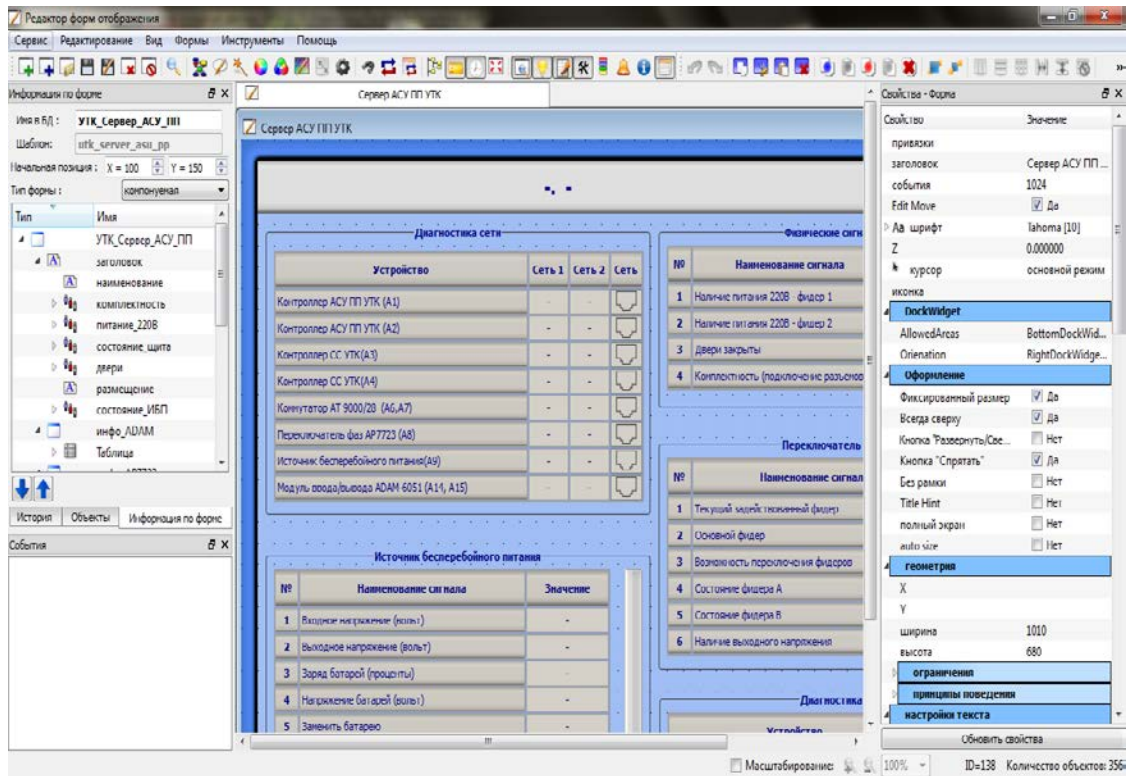


Рис. 2. Пример работы программы АПИД

Предлагаемая ИТ (информационная технология) имеет реализацию в виде программного комплекса состоящего из программы Автоматизированной Подготовки Исходных Данных и Знаний (АПИДЗ) и Исполнительной системы. При этом пользователю обеспечивается работа в рамках рассматриваемого интеллектуального интерфейса в ходе реализации нескольких этапов, перечень которых представлен на рис. 3.



Рис. 3. Этапы работы пользователя с системой АПИДЗ

Работа пользователя при этом выполняется в следующей последовательности:

- проектирование структур и типов данных. На данном этапе описываются параметры, переменные, таблицы текстов и другие структуры данных, используемые при мониторинге;
- визуальное проектирование форм. Создаются мнемосхемы, диалоговые окна с которыми пользователь будет работать на этапе анализа;
- логическое проектирование программы мониторинга – наиболее важный этап проектирования системы мониторинга.

Пользователь, используя предлагаемую МПЗ (модель представления знаний), описывает ОА (объект анализа) с его подсистемами и элементами на языке предметной области, – до необходимого уровня детализации.

Для каждого этапа работы пользователя имеется специализированный редактор, который наилучшим образом помогает решить поставленную задачу. Редактор мнемосхем обладает обширным и постоянно пополняемым набором примитивов, которые позволяют создать мнемосхему практически для любой предметной области и любой сложности. Сервисные функции редактора помогают сделать процесс разработки быстрым и удобным. Использование когнитивной графики в примитивах повышает информативность мнемосхемы и качество восприятия информации.

Редактор БЗ (базы знаний) позволяет сформировать описание ОА и его состояний в соответствии с МПЗ на языке предметной области. Такое описание облегчает понимание и сопровождение созданных алгоритмов другими специалистами в этой предметной области, не требует привлечения специалистов по программированию на традиционных языках программирования. Тем самым устраняется фактор недопонимания между специалистом в конкретной предметной области и программистом, так как обе эти функции может выполнить сам технолог (анализатор) [1–3].

Разработка ПК диагностики элементов оборудования АСУ ПП в рамках УТК 14П83 проводилась с помощью автоматизированной системы подготовки исходных данных. На рис. 4 представлен главный экран ПК, где отображены все объекты мониторинга, а также степень работоспособности элементов оборудования.

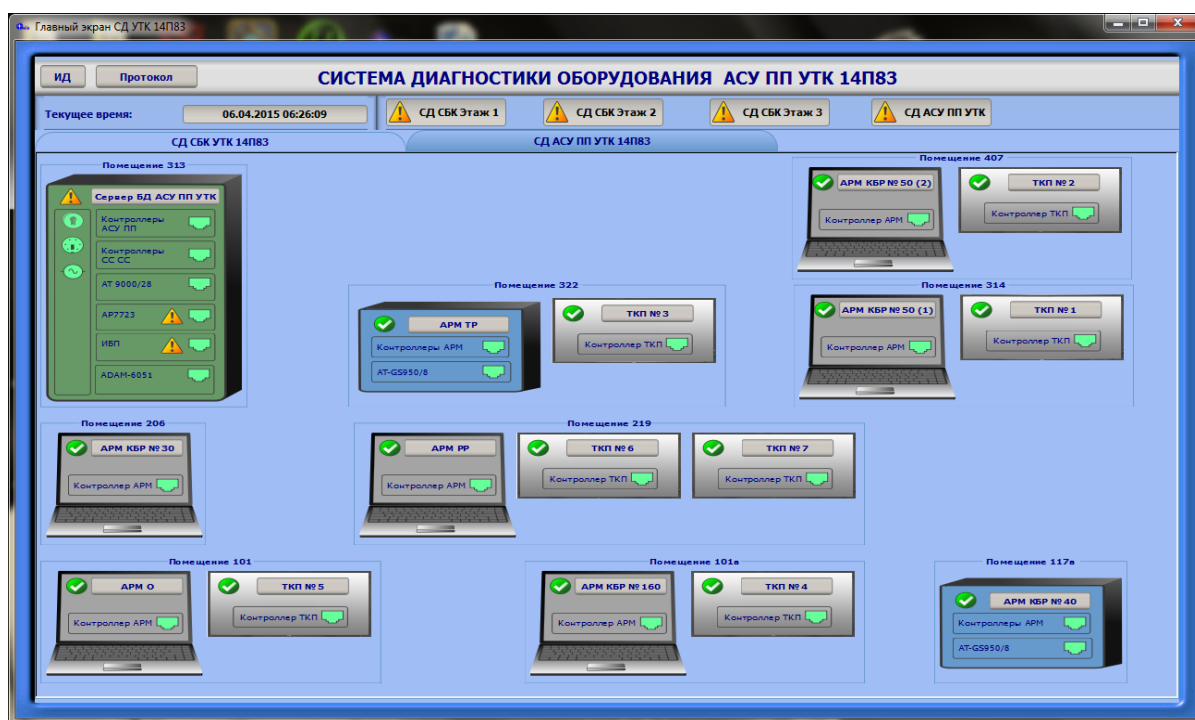


Рис. 4. Главный экран ПК диагностики элементов оборудования АСУ ПП



Для каждого объекта мониторинга предусмотрена возможность детализированного просмотра технического состояния его элементов, как представлено на рис. 5. Также в таблице «Диагностика сети» при наведении курсора на сеть элементов оборудования появляется всплывающая подсказка с номером их IP-адреса, взятого из таблицы исходных данных (рис. 6).

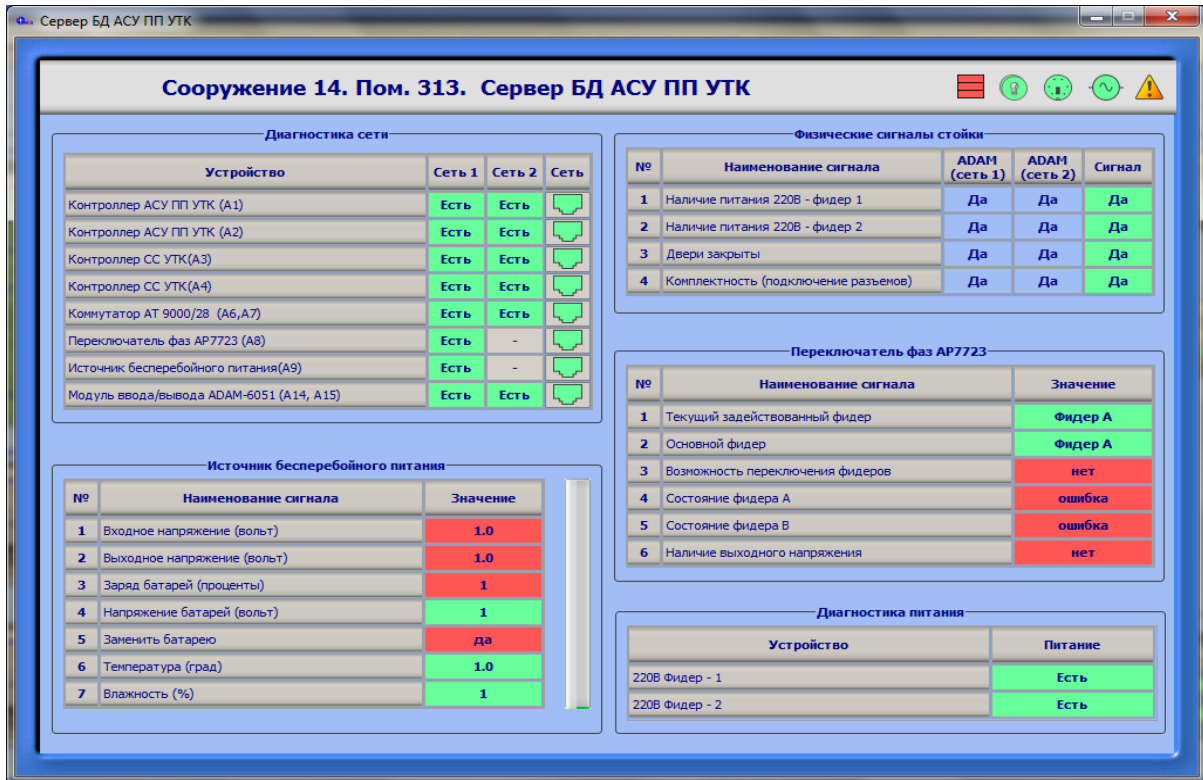


Рис. 5. Детализация технического состояния элементов объекта мониторинга

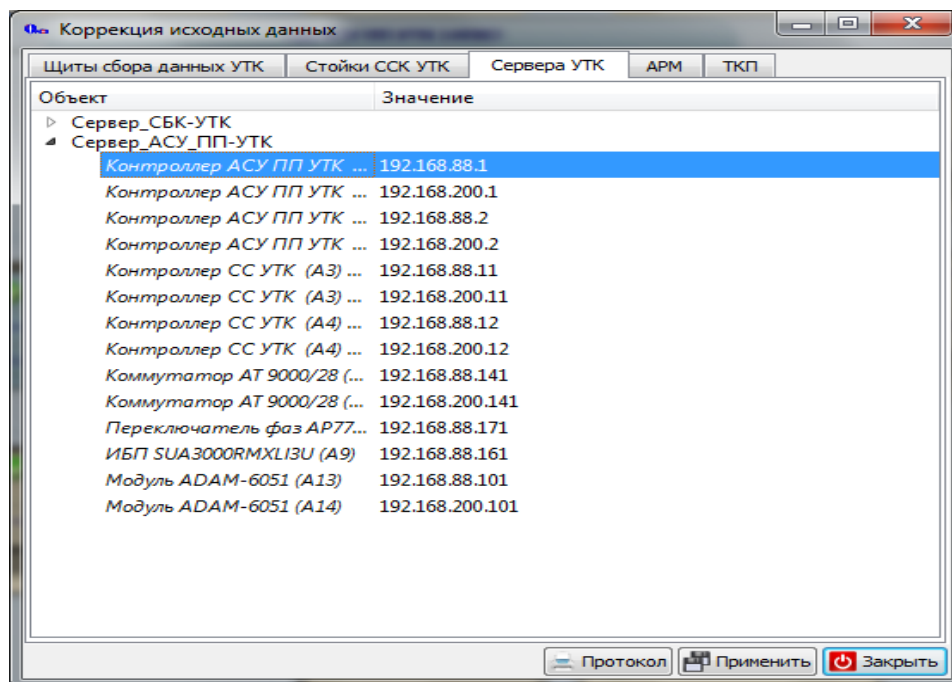


Рис. 6. Исходные данные

Информация обо всех произошедших событиях заносится в протокол (рис. 7).

Время	Событие
2015/04/06 06:25:27.675	Сооружение 14. Пом. 322. АРМ ТР № 0002. Коммутатор АТ-GS950/8 (A2) в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 322. АРМ ТР № 0002. Коммутатор АТ-GS950/8 (A3) в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 117а. АРМ КБР №40 № 0001. Коммутатор АТ-GS950/8 (A2) в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 117а. АРМ КБР №40 № 0001. Коммутатор АТ-GS950/8 (A3) в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 313. Сервер БД АСУ ПП УТК № 0001. Коммутатор АТ 9000/28 (A6) в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 313. Сервер БД АСУ ПП УТК № 0001. Коммутатор АТ 9000/28 (A7) в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 313. Сервер БД АСУ ПП УТК № 0001. Переключатель фаз AP7723 (A8) в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 313. Сервер БД АСУ ПП УТК № 0001. ИБП SUA3000RMLBU (A9) в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 313. Сервер БД АСУ ПП УТК № 0001. Модуль ADAM-6051 (A14) в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 313. Сервер БД АСУ ПП УТК № 0001. Модуль ADAM-6051 (A15) в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 219. АРМ РР № 0001. в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 101. АРМ О № 0001. в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 322. АРМ ТР № 0002. Контроллер (A10) в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 322. АРМ ТР № 0002. Контроллер (A11) в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 206. АРМ КБР №30 № 0001. в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 117а. АРМ КБР №40 № 0001. Контроллер (A10) в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 117а. АРМ КБР №40 № 0001. Контроллер (A11) в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 314. АРМ КБР №50 (1) № 0001. в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 407. АРМ КБР №50 (2) № 0002. в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 101а. АРМ КБР №160 № 0001. в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 314. ТКП №1 № 0001. в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 407. ТКП №2 № 0002. в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 322. ТКП №3 № 0003. в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 101а. ТКП №4 № 0004. в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 101. ТКП №5 № 0005. в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 219. ТКП №6 № 0006. в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 219. ТКП №7 № 0007. в сети
2015/04/06 06:25:27.675	Сооружение 14. Пом. 313. Сервер БД АСУ ПП УТК № 0001. Двери закрыты
2015/04/06 06:25:27.675	Сооружение 14. Пом. 313. Сервер БД АСУ ПП УТК № 0001. Есть комплектность

Рис. 7. Протокол произошедших событий

Программный комплекс диагностирования элементов АСУ ПП УТК 14П83 был успешно реализован. Благодаря ему пользователь сможет следить за состоянием оборудования в удобной для него форме и сможет моментально определить поломку какого-либо элемента. В итоге, это сэкономит время на починку, а также сократит финансовые убытки предприятия в связи с выходом неисправного элемента из строя. Поскольку подобные системы очень удобны и экономически выгодны для диагностирования элементов оборудования большинства фирм, их разработка становится одним из важнейших направлений автоматизации управления производством.

### Библиографический список

1. Охтилев М. Ю., Соколов Б. В., Юсупов Р. М. Интеллектуальные технологии мониторинга и управления структурной динамикой сложных технических объектов. М.: Наука, 2006. 410 с.
2. Майданович О. В., Соколов Б. В., Охтилев М. Ю. Новый подход к созданию интеллектуальных информационных технологий проектирования систем мониторинга состояния сложных объектов // XI международная научно-техническая конференция 12–14 мая 2010 г. «Кибернетика и высокие технологии XXI века»: Сборник докладов в 2-х томах. Воронеж: НПФ «Саквое», 2010. Т. 2. С. 601–608.
3. Охтилев М. Ю., Соколов Б. В. Теоретические и прикладные проблемы разработки и применения автоматизированных систем мониторинга состояния сложных технических объектов // Тр. СПИИРАН. Отделение информационных технологий и вычислительных систем. СПб., 2002. Вып. 1. Т. 1. С. 167–180.
4. Посупонько Н. В. Автоматизированные системы контроля, диагностики, прогнозирования. Ростов-на-Дону, 2008. 79 с.
5. Колтунцев А. В., Золотарёв С. В., Построение распределённых систем управления с помощью SCADA-пакета PcVue // ИСУП, 2008 г., № 3.

**Р. Ю. Шеин**

студент кафедры компьютерной математики и программирования

**М. Д. Поляк**

научный руководитель

### ОБРАБОТКА ИЗОБРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ СТОХАСТИЧЕСКОЙ НЕЙРОННОЙ СЕТИ

Основной задачей данной работы является определение эффективности использования искусственных нейронных сетей типа «машина Больцмана с ограничениями» (МБО), а также построенной на основе МБО «Глубокой сети доверия» (ГСД) при решении задачи классификации изображений. В дополнение необходимо определить, какие преимущества дает использование особенностей архитектуры ГСД, по сравнению с одиночной МБО.

«Машина Больцмана с ограничениями» является модифицированным вариантом сети «машина Больцмана» (МБ). В отличие от изначальной конфигурации, обладавшей полносвязной структурой, затруднявшей обучение, структура МБО позволяет существенно снизить вычислительные и временные затраты на обучение, и более приспособлена для решения практических задач. Как и МБ, МБО является стохастической рекуррентной нейронной сетью. В классическом варианте все нейроны сети являются бинарными и сгруппированы в два слоя. Нейроны разных слоев полностью связаны друг с другом, однако связи между нейронами в пределах одного слоя отсутствуют. На рис. 1 схематично отображена структура классической МБО.

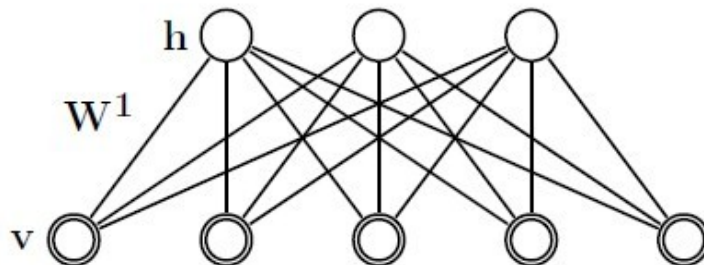


Рис. 1. Структура МБО

В виду отсутствия нейронных связей внутри слоя можно сформировать относительно простое правило обновления состояний нейронов скрытого слоя на основе состояний видимого [1]:

$$p(h_j = 1|v) = \sigma(b_j + \sum v_i w_{ij}), \quad (1)$$

где  $\sigma$  – логистическая функция.

Аналогично выглядит и правило обновления состояний нейронов видимого слоя, на основе скрытого:

$$p(v_j = 1|h) = \sigma(a_j + \sum h_i w_{ij}). \quad (2)$$

Правило обновления весовых коэффициентов в соответствии с [1] выглядит следующим образом:

$$\Delta w_{ij} = \varepsilon (\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}). \quad (3)$$

Процесс обучения МБО сводится последовательному обновлению состояний нейронов скрытого слоя в соответствии с выражением (1) с последующим обновлением видимого слоя в соответствии с выражением (2) вплоть до достижения ситуации равновесия. Данный процесс выполняется для всех примеров обучающей выборки и носит название «Марковская цепь». После завершения указанного процесса происходит обновление весовых коэффициентов сети в соответствии с (3), на основании полученных распределений  $\langle v_i h_j \rangle_{data}$  и  $\langle v_i h_j \rangle_{model}$ . Алгоритм обучения, реализующий описанную выше последовательность действий, называется «алгоритм сравнительного расхождения» (АСР).

Глубокая сеть доверия – система, включающая в себя некоторое количество предварительно обученных МБО соединенных последовательно в виде стека. Входные данные ГСД поступают на видимый слой самой нижней МБО и последовательно передаются к скрытому слою либо SoftMax-группе нейронов самой верхней МБО. Подобная конструкция позволяет обрабатывать значительно более сложные данные, для которых одиночные МБО не в состоянии продемонстрировать высокие результаты. Схематично структура ГСД приведена на рис. 2.

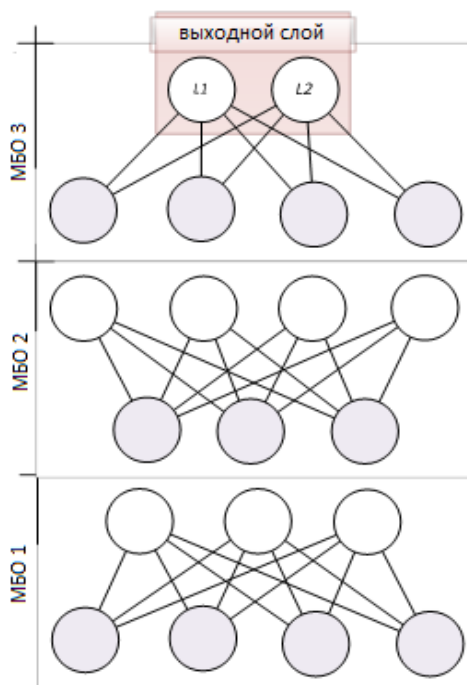


Рис. 2. Структура ГСД

ГСД может быть представлена в виде многослойной нейронной сети прямого распространения и, следовательно, может быть обучена с использованием алгоритма обратного распространения ошибки (ОРО). При обучении ГСД используются как АСР, для пред-обучения отдельных МБО, входящих в состав ГСД, так и алгоритм обратного распространения ошибки, для пост-обучения ГСД в целом. Данный комбинированный подход позволяет получить более высокие результаты при меньших временных затратах на обучение.

Для решения поставленной задачи было проведено несколько серий экспериментов с последующим сравнением и анализом полученных результатов. Для получения более объективных результатов использовались несколько наборов изображений, отличающихся по структуре и сложности:

- MNISTSMALL – база данных изображений рукописных цифровых символов белым цветом на черном фоне;

– Caltech 101 – база данных изображений реальных объектов, представленных в виде контуров на белом фоне;

– CIFAR 10 – база данных реальных фотореалистичных изображений без ограничений на угол поворота, размер и положение объектов относительно центра изображений. Изначально все изображения представлены в трехцветном формате, однако в целях уменьшения требований в вычислительным и временным ресурсам были переведены в формат оттенков серого, путем усреднения по трем компонентам.

Рассматривался только стандартный способ обработки, поэтому изображения для представления в виде входных данных раскладываются в вектор и подаются на вход сети целиком. Для каждого набора изображений была проведена серия из нескольких экспериментов с постепенным увеличением количества МБО в составе используемой ГСД и подбором параметров обучения. Обучение сетей проходило в два этапа:

- обучение каждой МБО в отдельности с использованием АСР;
- составление из обученных МБО единой ГСД и последующее ее обучение при помощи алгоритма ОРО.

По итогам всех серий экспериментов представлены результаты для баз MNIST (табл. 1), Caltech 101 (табл. 2) и CIFAR 10 (табл. 3). Значения, приведенные в таблицах – ошибка классификации соответствующей сети после завершения обучения. В скобках указано количество МБО в составе используемой ГСД.

Таблица 1

Результаты для базы данных MNIST

	ГСД № 1 (1 МБО)	ГСД № 2 (2 МБО)	ГСД № 3 (3 МБО)
Ошибка До ОРО	6.1%.	5.91	5.82
Ошибка После ОРО	4.1%	3.92	3.51

Таблица 2

Результаты для базы данных Caltech 101

	ГСД № 1 (1 МБО)	ГСД № 2 (2 МБО)	ГСД № 3 (3 МБО)
Ошибка До ОРО	42.5%.	35.56%	36.97%
Ошибка После ОРО	41.7%	28.03%	35.52%

Таблица 3

Результаты для базы данных CIFAR 10

	ГСД № 1 (1 МБО)	ГСД № 2 (2 МБО)
Ошибка До ОРО	69.49%.	79.87%
Ошибка После ОРО	60.76%	65.05%

Согласно представленным выше результатам серии экспериментов на изображениях базы данных MNIST можно заключить, что МБО и ГСД на основе МБО можно применять в качестве классификатора изображений с относительно высокими результатами при условии, что изображения обладают простой структурой и малым количеством признаков и категорий. При этом увеличение слоев ГСД позволяет добиться повышения эффективности, давая ГСД преимущество перед одиночными МБО.

Значительное увеличение категорий, как показала серия экспериментов с изображениями базы Caltech 101, приведет к заметному увеличению ошибки сети, что ставит возможность ее использования в подобных условиях под сомнение.

Серия экспериментов, проведенная на изображениях базы данных CIFAR, демонстрирует неспособность МБО и ГСД при использовании стандартного подхода корректно выделять и использовать для классификации существенные признаки изображений с достаточно сложной структурой. Основная причина – преобразование двумерного изображения в формат одномерного массива или вектора. При выполнении подобного преобразования происходит потеря значительного количества информации о взаимном расположении пикселей, которая, в случаях с изображениями, играет существенную роль. Кроме того, преобразование изображений в формат оттенков серого также привело к снижению информативности и отрицательно сказалось на результате. В целом МБО, при использовании стандартного подхода, является не самым эффективным инструментом при решении задач классификации достаточно сложных изображений.

Наименьшая ошибка классификации с использованием ГСД на цельных изображениях базы CIFAR 10 была получена А. Крижевски с использованием сети с очень большим количеством нейронов и составляет 35.26% [1]. Изображения при этом прошли значительную предварительную обработку, позволяющую выделить и усилить существенные признаки, облегчив тем самым задачу для сети. Данный результат был улучшен им же (21%) при помощи ГСД, использующей некоторые принципы построения сверточных нейронных сетей [2]. Используемый им подход отличается повышенной сложностью реализации, однако позволил акцентировать внимание отдельных нейронов сети на конкретных частях изображения, повысив тем самым показатели сети. Минимальная ошибка классификации, полученная на сегодняшний день на наборе CIFAR 10 среди всех типов нейронных сетей, составляет всего 6% и достигнута с использованием сложной нейронной сети сверточной архитектуры [3].

Таким образом можно заключить, что МБО и ГСД способны решать задачи классификации изображений, однако показатели их эффективности в данной области ниже аналогичных показателей, специально разработанных для решения данной задачи аналогов.

### **Библиографический список**

1. Geoffrey Hinton. A Practical Guide to Training Restricted Boltzmann Machines (Version 1). Department of Computer Science, University of Toronto
2. Krizhevsky A. Convolutional Deep Belief Networks on CIFAR-10.
3. The MNIST Database of handwritten digits by Yann LeCun <http://yann.lecun.com/exdb/mnist/>

УДК 004.75

***М. Н. Шелест***

студент кафедры проблемно-ориентированных вычислительных комплексов

***Е. А. Бакин***

канд. техн. наук, доц. – научный руководитель

## **МОДЕЛИРОВАНИЕ СЕТЕЙ МАССОВОГО ОБСЛУЖИВАНИЯ С РЕЗЕРВИРОВАНИЕМ РЕСУРСОВ**

Зачастую на практике перед разработчиком вычислительных и телекоммуникационных систем возникает задача их предварительного анализа с целью выявления критических мест и оценки потенциальных характеристик. Такой анализ может быть затруднителен для сетей, в которых применяются сложные механизмы работы, отличающиеся от тех, которые традиционно рассматриваются в литературе [1, 2]. Примерами таких систем могут служить системы

с резервированием ресурсов (например, системы с коммутацией канала [3] и системы обработки транзакций [4]).

Типовой моделью вычислительных сетей являются сети массового обслуживания. Такая сеть состоит из множества элементарных систем массового обслуживания (СМО). Каждое из этих СМО представляет собой пару – обслуживающее устройство (ОУ) и буфер. Поступление заявок в такую сеть и их обслуживание моделирует процесс функционирования системы. Процесс обслуживания заявки соответствует ее прохождению по маршруту из нескольких ОУ (рис. 1). Так, например, в случае систем с коммутацией канала заявка соответствует сообщению, роль ОУ выполняет АТМ-маршрутизатор. В случае же систем обработки транзакций, заявке соответствует транзакция, а ОУ – это объект информационной системы, который задействован в процессе выполнения транзакции (например, таблица базы данных или отдельная запись).

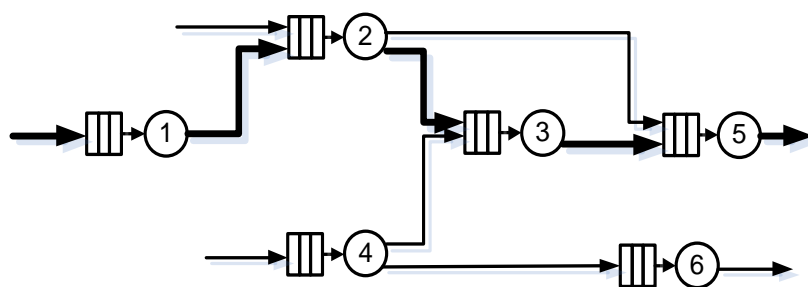


Рис. 1. Сеть СМО с выделенным маршрутом: 1-2-3-5

В приведенных выше примерах зачастую используется следующий механизм резервирования ресурсов: поступаая на обслуживание, заявка блокирует весь набор ОУ, входящих в ее маршрут, и снимет с них блокировку только тогда, когда обслуживание заявки завершится, и она покинет систему.

Так как теоретический анализ таких систем является затруднительным даже в простейших случаях, для оценок их характеристик функционирования, как правило, применяется метод имитационного моделирования. Наиболее эффективным методом моделирования сетей СМО является моделирование по событиям [5, 6]. События в системе наступают в случайные моменты времени. Состояние системы при этом может меняться только в моменты наступления событий, и полный набор состояний системы может быть получен путем продвижения имитационного системного времени от одного события к другому, что позволяет значительно сократить время моделирования. Таким образом, работа имитационной модели системы представляет собой хронологическую последовательность событий перехода системы из одного состояния в другое.

Целью настоящей разработки являлось создание имитационной модели системы с резервированием ресурсов, работающей по принципам моделирования по событиям. Характеристики предлагаемой модели приведены в таблице 1.

Таблица 1

Характеристики модели	
Входные параметры	<ul style="list-style-type: none"> <li>– Количество элементарных СМО (ОУ и буфер);</li> <li>– Интенсивность поступления заявок;</li> <li>– Интенсивности обслуживания для каждого ОУ;</li> <li>– Набор маршрутов;</li> <li>– Вероятности выбора заявкой маршрута</li> </ul>

## ТЕХНИЧЕСКИЕ НАУКИ

Выходные параметры	<ul style="list-style-type: none"> <li>– Количество заявок в системе;</li> <li>– Среднее время обслуживания заявки;</li> <li>– Время пребывания заявки в системе;</li> <li>– Среднее время работы ОУ</li> </ul>
Механизм резервирования	– Резервирование набора ОУ всего маршрута на время обслуживания заявки в системе
Поток заявок	– Пуассоновский
Обслуживание заявок	– В порядке поступления (первый пришел – первый ушел, FIFO)

В разработанной модели реализованы следующие события:

- приход заявки;
- завершение обслуживания в ОУ;
- переход заявки из буфера в ОУ;
- переход заявки в следующее ОУ на маршруте;
- блокировка набора ОУ;
- снятие блокировки с набора ОУ.

В табл. 2 приведена реакция модели на возникновение того или иного события.

Таблица 2

### События

Событие	Следствие наступления события
Приход заявки	<ul style="list-style-type: none"> <li>– Блокировка ОУ на всем маршруте;</li> <li>– Вычисление времени завершения обслуживания заявки в ОУ;</li> <li>– Вычисление времени прихода следующей заявки</li> </ul>
Приход заявки	<ul style="list-style-type: none"> <li>– Поступление заявки в буфер;</li> <li>– Вычисление времени прихода следующей заявки;</li> </ul>
Завершение обслуживания заявки в ОУ	– Переход заявки в следующее ОУ на маршруте;
Переход заявки в следующее ОУ на маршруте	– Вычисление времени завершения обслуживания заявки в ОУ;
Завершение обслуживания заявки в ОУ (последнем на маршруте)	– Разблокировка ОУ на всем маршруте;
Разблокировка ОУ на всем маршруте	– Переход заявки из буфера в ОУ;
Переход заявки из буфера в ОУ	<ul style="list-style-type: none"> <li>– Блокировка ОУ на всем маршруте;</li> <li>– Вычисление времени завершения обслуживания заявки в ОУ;</li> <li>– Вычисление времени прихода следующей заявки;</li> </ul>

На этапе верификации программного средства был подготовлен пакет тестовых сценариев, для которых возможен аналитический расчет. Пример одного из таких сценариев представлен на рис. 2.



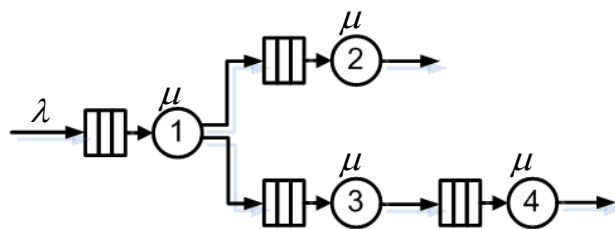


Рис. 2. Пример тестовой сети

Рассматриваемая сеть СМО состоит из четырех элементарных СМО. Заявка случайным образом выбирает один из двух маршрутов: ( $\rightarrow 1 \rightarrow 2 \rightarrow$ ) или ( $\rightarrow 1 \rightarrow 3 \rightarrow 4 \rightarrow$ ). Очевидно, что в такой системе не может обслуживаться более одной заявки одновременно. Можно показать, что данную сеть можно свести к однолинейной СМО типа M/G/1. Тогда среднее время пребывания заявки в системе определяется формулой Поллачека-Хинчина [1]:

$$\bar{T} = \frac{1}{\lambda} \left( \frac{\rho^2 (1 + v_{\Sigma}^2)}{2(1 - \rho)} + \rho \right).$$

Здесь:

$$v_{\Sigma} = \frac{\sqrt{6\rho_1 + 12\rho_2 - (2\rho_1 - 3\rho_2)^2}}{2\rho_1 + 3\rho_2},$$

$$\rho = \frac{\lambda}{\mu} (2\rho_1 + 3\rho_2),$$

где  $\lambda$  – интенсивность поступления заявок в систему;  $\mu$  – интенсивность обслуживания каждого ОУ;  $\rho_i$  – вероятность выбора  $i$ -го маршрута;  $v_{\Sigma}$  – результирующий коэффициент вариации сети;  $\rho$  – коэффициент загрузки сети.

Сравнение результатов моделирования с теоретическим расчетом представлено на рис. 3.

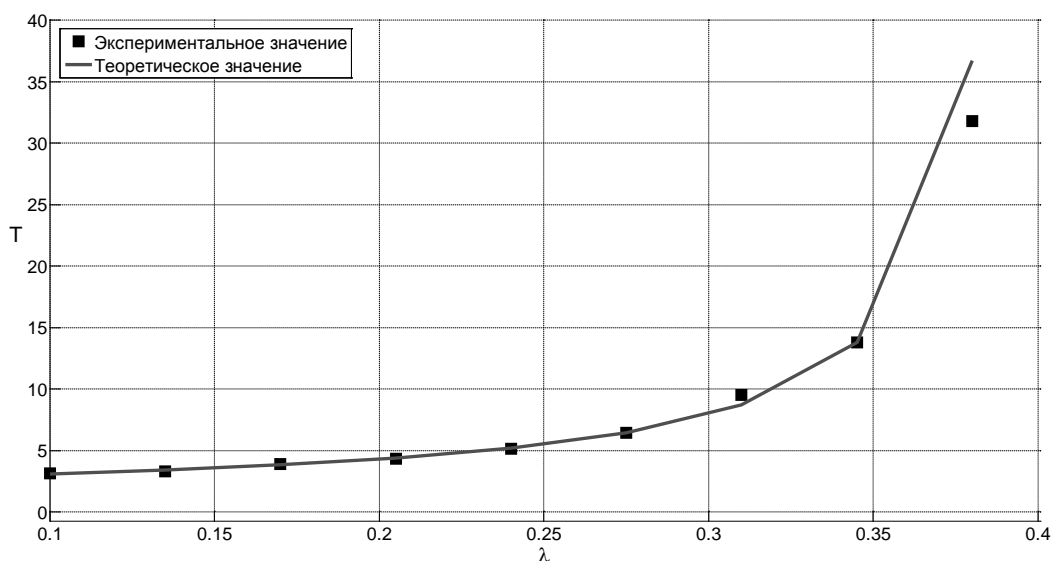


Рис. 3. Графики зависимости среднего времени обслуживания заявки от интенсивности поступления заявок

В результате работы над проектом было создано и отлажено программное средство, позволяющее методами имитационного моделирования находить такие характеристики как среднее время пребывания заявки в системе и среднее количество заявок в системе. В рамках развития проекта предлагается рассмотреть другие дисциплины обслуживания и механизмы блокировки ресурсов. Также представляет интерес анализ быстродействия работы модели в зависимости от изменения параметров системы.

### Библиографический список

1. Клейнрок Л. Теория массового обслуживания: пер. с англ. / пер. И. И. Грушко; ред. В. И. Нейман – М.: Машиностроение, 1979. – 432 с.
2. Клейнрок Л. Вычислительные системы с очередями: пер. с англ. / под. ред. Б. С. Цыбакова. – М.: Мир, 1979. – 597 с.
3. Уолрэнд Дж. Телекоммуникационные и компьютерные сети. Вводный курс: пер. с англ. М.: Постмаркет, 2001. – 480 с.
4. Пушников А. Ю. Введение в системы управления базами данных. Часть 2. Нормальные формы отношений и транзакции: учеб. пособие / Изд-е Башкирского ун-та. – Уфа, 1999. – 138 с.
5. Бусленко Н. П. Моделирование сложных систем. М.: Наука, 1978.
6. Наместников А. М. Разработка имитационных моделей в среде MATLAB: Методические указания для студентов / УлГТУ. Ульяновск, 2004. 72 с.

УДК 004.42

**Е. А. Шинкаренко**

студент кафедры компьютерной математики и программирования

**С. А. Рогачев**

научный руководитель

## АВТОМАТИЗИРОВАННАЯ СИСТЕМА УДАЛЕННОЙ ПРОВЕРКИ ЛАБОРАТОРНЫХ РАБОТ

На дисциплинах, связанных с программированием, для закрепления преподаваемого материала, предусматривают практические занятия. Перед студентом встает задача освоения и закрепления пройденного материала, все это в совокупности определяет лабораторную работу. В свою очередь, защита лабораторной работы разбивается на два этапа: выполнение практической части и ее защита преподавателю. Защита лабораторной работы, в общем случае, включает в себя проверку общей информации по лабораторной работе; условия задачи по варианту студента; требований, предъявляемых к данной работе; ожидаемых результатов выполнения программы [1].

Основная проблема, которая может возникнуть у студента – лабораторная работа не выполнена полностью по требованиям, предъявляемым к ней, итогом чего будет служить отказ преподавателя допустить к прохождению этапа защиты. В ином же случае, сам преподаватель может допустить неправильную программу (не все входные тестовые данные были предусмотрены при проверке работы). Необходимо контролировать процесс выполнения лабораторной работы. Именно для решения этой задачи и предназначена данная автоматизированная система удаленной проверки лабораторных работ. Основные функции, возлагаемые на данную систему: предоставление студенту информации о лабораторных работах; выявление корректности предоставленного решения; конечное оценивание результатов работы; ведение электронного журнала. Их реализация способствует к упрощению проверки лабораторных работ, так как программа проходит предварительное тестирование. Студент, в свою очередь, после получения положительных результатов работы, будет уверен, что допущен к этапу защиты.

В основе работы системы заложена клиент-серверная архитектура (рис. 1) [2]. Выбранную архитектуру можно аргументировать тем, что все вычисления, а также обработка всех данных и их непосредственная выдача, должны осуществляться на стороне сервера, благодаря чему происходит полный контроль работы системы.

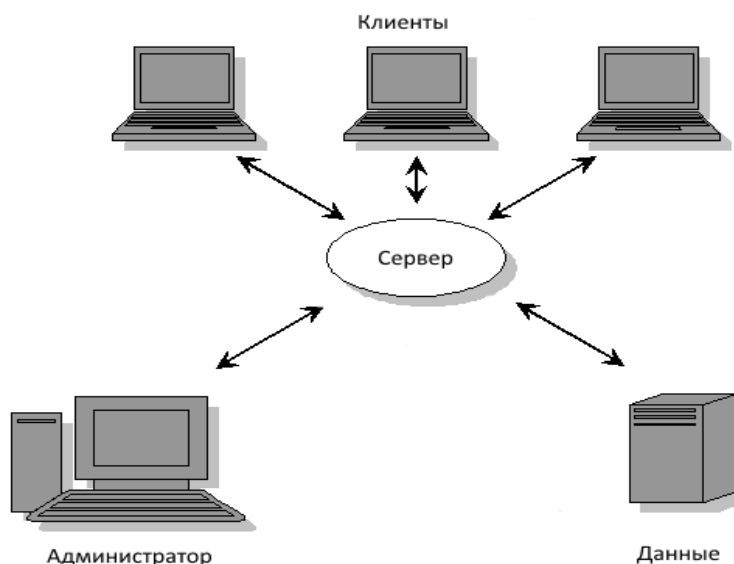


Рис. 1. Клиент-серверная архитектура системы

В роли сервера здесь выступает удаленная от клиента вычислительная машина, которая решает следующие задачи: интерпретация входящих запросов; обеспечение хранения данных; осуществление безопасности в работе всей системы. Также, благодаря вынесению всех вычислений на серверную составляющую, клиент полностью высвобождается от не нужной для него нагрузки, в то время как, сервер должен сохранять, по возможности, полную работоспособность в момент пиковых нагрузок со стороны студентов. В роли клиентов в данной системе выступают компьютеры студентов. Краткое описание взаимодействия студента с системой: студент получает всю информацию о лабораторных работах, которые закреплены за его именем; от студента ожидается поступление варианта своего решения одной из лабораторных работ; система обрабатывает полученную работу и высылает результаты проверки. Под администратором понимается человек, который следит за работоспособностью системы. В функции администратора входят: контроль за данными; внесение изменений в конфигурацию работы сервера; управление правами доступа к системе. Объект «Данные» на рис. 1. отвечает за хранение и доступ к данным.

Как на серверную, так и клиентскую составляющие, был определен объемный список требований и, дабы сократить по времени конечный объем работ, необходимо было определиться с соответствующим инструментарием разработки. Для клиентской части было решено реализовать отдельное независимое приложение и воспользоваться кроссплатформенным фреймворком «Qt» [3]. Инструмент предоставляет удобный набор библиотек для постройки нужного графического интерфейса пользователя, и что не менее важно для клиентской части, готовый программный код можно скомпилировать под нужную среду, список которых со временем становится все больше и больше. Что же касается разработки серверной части, здесь «Qt» позволяет не опускаться в тонкости конкретной платформы, что ускоряет весь процесс разработки.

На рис. 2. представлен общий графический интерфейс клиентского приложения.

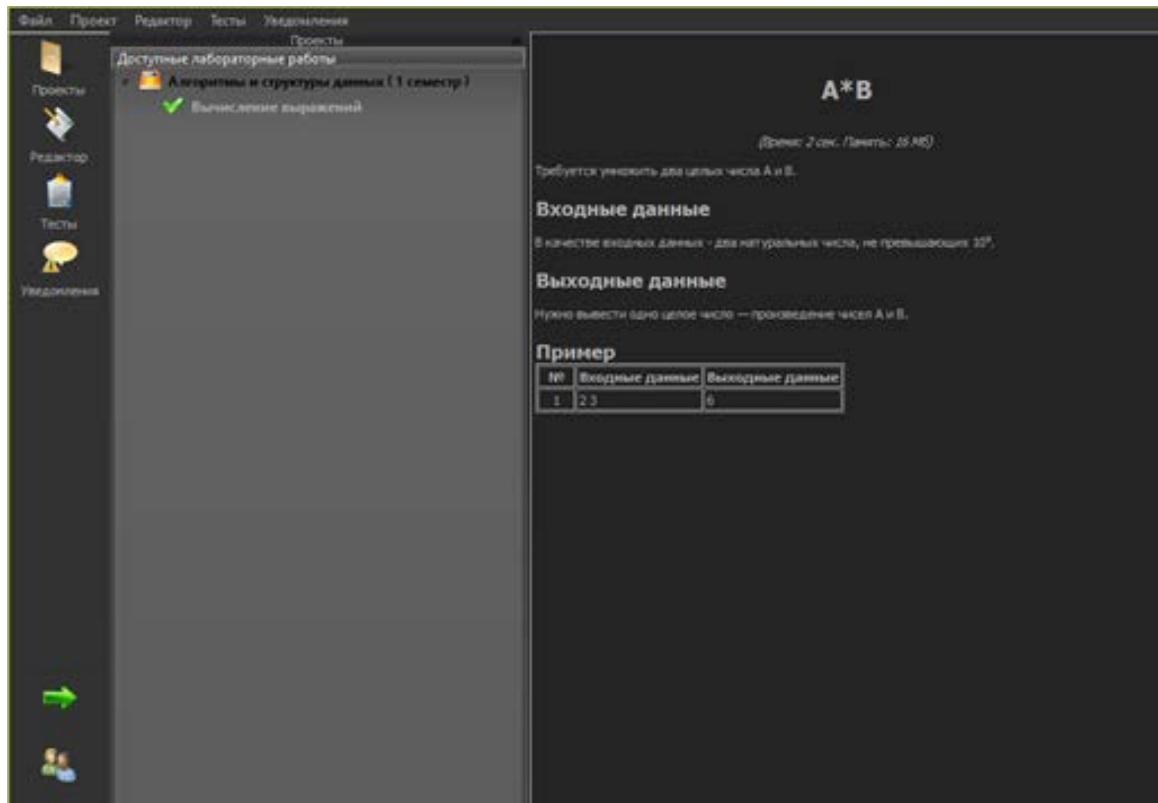


Рис. 2. Общий вид графического интерфейса клиентского приложения

Как видно из рис. 2, клиенту предоставлены: вся информация о доступных для него лабораторных работах; редактор для управления исходным программным кодом по лабораторной работе; возможность просматривать историю тестирования своих работ; получать уведомления, касающиеся не только самой системы, но и учебного плана в целом.

На данном этапе имеет смысл и дальнейшее развитие проекта. Существует множество направлений: добавление анализатора на корректность написанного программного кода; внедрение в систему интерактивных уроков для повторения пройденного материала; решение дополнительных заданий по курсу; добавление новых, не только связанных с программированием, дисциплин. Не исключены плюсы и для самого студента: упрощение сдачи работы путем возможности защиты этапа с программным кодом в домашних условиях; внедрение процесса автоматизированного тестирования конечной программы, заставляет студента анализировать свой исходный программный код, перед отправкой, для его полного (частичного) прохождения по тестам; быстрый и удобный доступ к данным по лабораторным работам, которые со временем могут изменяться, поэтому студент никогда не упустит из виду обновленный материал.

### Библиографический список

1. Ключарев А. А., Матяш В. А., Щекин С. В. Структуры и алгоритмы обработки данных: учеб. пособие / СПбГУАП. СПб., 2003. 172 с.: ил. ISBN 5-8088-0120-6.
2. Википедия. Свободная энциклопедия [Электронный ресурс]: <https://ru.wikipedia.org/wiki/Клиент-Сервер> – Режим доступа: свободный
3. Макс Шлее. Qt 4.8. Профессиональное программирование на C++. БВХ-Петербург, 2012. 894 с.

# Информационные системы и защита информации

УДК 004.923

**В. С. Бобров**

студент кафедры инфокоммуникационных систем

**А. И. Веселов**

научный руководитель

## РАЗРАБОТКА И ИССЛЕДОВАНИЕ АЛГОРИТМА ВЫЧИСЛЕНИЯ КАРТЫ ГЛУБИНЫ ПО СТЕРЕИОИЗОБРАЖЕНИЮ

В настоящее время в области инфокоммуникационных технологий наблюдается существенное развитие представления данных в трехмерном пространстве. Данными могут являться как изображения, так и видеосигналы. Для представления пользователю в интерактивном режиме трехмерных сцен в современных технологиях используются стереоизображения. Одна из важных задач при обработке стереоизображений – это процесс преобразования двух плоских изображений (стереопары) в трехмерную сцену с восстановлением информации о глубине (карты глубины) каждой точки плоского изображения (расстоянии от стереокамеры до соответствующей точки реальной сцены) [1]. Данной проблеме посвящено большое количество работ и по сей день она остается актуальной. Причина этого в том, что пока еще не найден надежный и быстрый алгоритм для решения этой задачи, так как в процессе вычисления глубины возникают различные дефекты (области с неизвестными значениями) и искажения (неверная информация о глубине). Для решения задачи восстановления глубины существует целый ряд алгоритмов, которые можно разделить на две категории: локальные и глобальные. Локальные подходы, как правило, отличаются меньшей вычислительной сложностью, однако результат их работы может содержать большое число ошибок. Глобальные подходы, как правило, дают очень хорошие результаты по сравнению с локальными, но использовать их в реальном времени, на сегодняшний день, практически невозможно. В представленной работе будет рассмотрен известный локальный подход [2] для нахождения информации о глубине сцены по стереопаре, и проанализированы результаты его работы. Таким образом, целью данной работы является выделение особенностей и проблем для карты глубины, полученной с использованием данного алгоритма.

Стереоизображение или стереопара (рис. 1) – это два изображения (левое и правое) одной и той же сцены, сделанные под разными ракурсами съемки. Левое изображение относительно центра сцены объекта сдвинуто немного вправо, а правое изображение сдвинуто немного влево.



а)



б)

Рис. 1. Стереопара tsukuba

а) левое изображение стереопары; б) правое изображение стереопары

Рассмотрим локальный подход для вычисления карты глубины по стереопаре (рис. 2). Здесь и далее будем называть его локальный алгоритм.

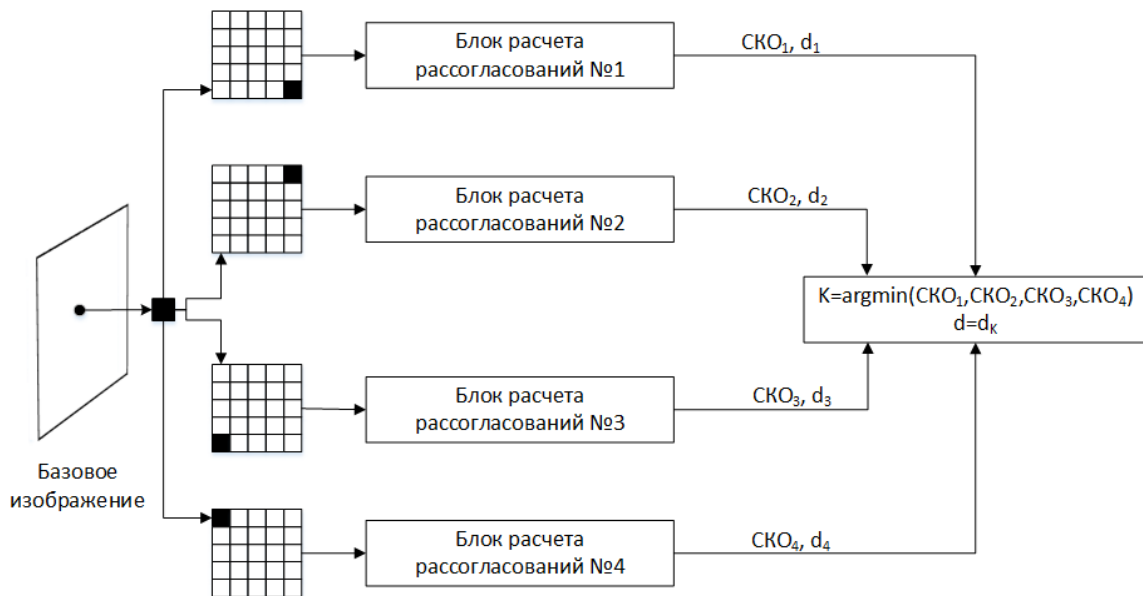


Рис. 2. Структурная схема локального алгоритма

На вход данного алгоритма подается пиксель базового изображения. Для данного пикселя строится четыре окна начальных размеров  $5 \times 5$  с максимальным увеличением до  $30 \times 30$ . В дальнейшем каждое окно будет обрабатываться отдельно независимо друг от друга. При обработке каждого из четырех окон действия алгоритма повторяются и выделены в структурной схеме, как блок расчета рассогласований (рис. 3).

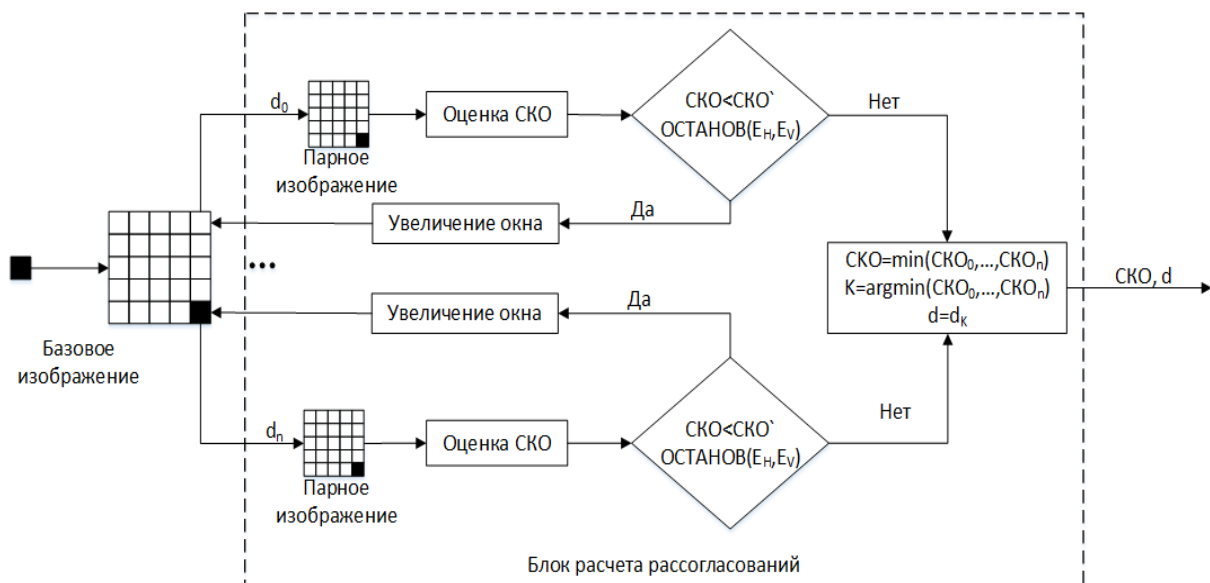


Рис. 3. Схема расчета рассогласований

Опишем работу блока расчета рассогласований, если входными данными является окно с исследуемым пикселем в правом-нижнем углу (правое). Так как заранее известны максимальное и минимальное рассогласования, то можно получить область пикселей на парном изображении, один из которых будет являться сопряженной точкой. Для каждого пикселя из этой области строится левое верхнее окно размером 5×5. После этого для каждой возможной пары окон базового и парного изображения будут происходить независимо следующие шаги.

1. Расчет между парами окон среднеквадратичное отклонение (СКО). Среднеквадратичное отклонение можно подсчитать по следующей формуле:

$$C = \frac{1}{(W_H W_V)^2} \sum_{m=-\frac{W_H}{2}}^{\frac{W_H}{2}} \sum_{n=-\frac{W_V}{2}}^{\frac{W_V}{2}} [I_L(i+m, j+n) - I_R(i+m-d, j+n)]^2 \quad (1)$$

где  $W_H$  – размер окна по высоте,  $W_V$  – размер окна по ширине,  $I_L$  – окно с базового изображения стереопары,  $I_R$  – окно с парного изображения стереопары.

2. Увеличение размеров пары окон на единицу по высоте или по ширине.

3. Если на текущем шаге СКО оказалось меньше чем на предыдущем, то переходим к пункту 1 и записываем значение  $СКО_{\min} = СКО$ .

4. Если СКО оказалось на текущем шаге больше или равно, чем на предыдущем шаге, то переходим к рассмотрению следующих пар окон.

Помимо того, что увеличение СКО на текущем шаге является критерием остановки роста окна, существуют еще два критерия: горизонтальное-среднее отклонение (Horizontalaveragedeviation) и вертикальное-среднее отклонение (Verticalaveragedeviation). Последние два критерия подсчитываются только для окна в базовом изображении.

Горизонтальное среднее отклонение высчитывается по следующей формуле:

$$E_H = \frac{1}{W_H W_V} \sum_{m=-\frac{W_H}{2}}^{\frac{W_H}{2}} \sum_{n=-\frac{W_V}{2}}^{\frac{W_V}{2}} |I_L(i+m, j+n) - \mu_{j+n}|, \quad (2)$$

где  $\mu_{j+n}$  – среднее значение по строке, которое можно подсчитать по формуле:

$$\mu_{j+n} = \frac{1}{W_H} \sum_{m=-\frac{W_H}{2}}^{\frac{W_H}{2}} I_L(i+m, j+n). \quad (3)$$

Вертикальное среднее отклонение подсчитывается по следующей формуле:

$$E_V = \frac{1}{W_H W_V} \sum_{m=-\frac{W_H}{2}}^{\frac{W_H}{2}} \sum_{n=-\frac{W_V}{2}}^{\frac{W_V}{2}} |I_L(i+m, j+n) - \mu_{i+m}| \quad (4)$$

где  $\mu_{i+m}$  – среднее значение по столбцу, которое можно подсчитать по формуле:

$$\mu_{i+m} = \frac{1}{W_H} \sum_{m=-\frac{W_V}{2}}^{\frac{W_V}{2}} I_L(i+m, j+n) \quad (5)$$

Если при расширении окна  $E_H > \frac{T}{W_H W_V}$  или  $E_V > \frac{T}{W_H W_V}$ , где  $T = 1440$ , то расширение окна прекращается и берется результат СКО на предыдущем шаге. После работы блока для каждой пары окон подсчитано и соответствующее ему рассогласование. На выходе блока расчета рассогласований необходимо выбрать  $CKO = \min(CKO_0, \dots, CKO_n)$ ,  $K = \operatorname{argmin}(CKO_0, \dots, CKO_n)$ ,  $d = d_k$ .

После того как получен результат работы вычислительного блока для левого-верхнего окна, аналогичные действия производятся с правым-верхним окном, левым-нижним окном и правым-нижним окном. В завершении работы алгоритма для рассматриваемого пикселя базового изображения выбирается из выходных данных вычислительных блоков  $K = \operatorname{argmin}(CKO_1, CKO_2, CKO_3, CKO_4)$ ,  $d = d_k$ .

После применения алгоритма ко всем пикселям базового изображения, получится карта рассогласования. Для того, чтобы вычислить карту глубины, необходимо каждый пиксель карты рассогласования умножить на шкалирующий коэффициент. На этом поиск предварительной карты глубины завершен. После нахождения предварительной карты глубины начинается ее постобработка. Для улучшения карты глубины в данном алгоритме применяется медианная фильтрация с размером окна  $7 \times 7$ . В данной работе будет приведен результат работы локального алгоритма для стереопары tsukuba (рис. 4, а) [3].

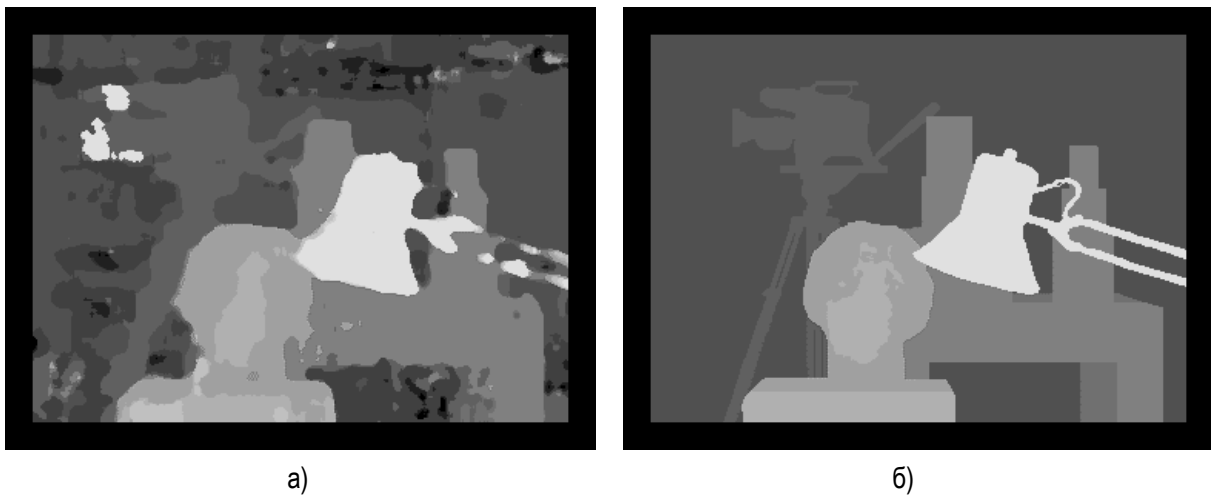


Рис. 4. Карты глубины для стереопары tsukuba  
а) карта глубины полученная с помощью локального алгоритма; б) GroundTruth

Для оценки полученных результатов применяются следующие критерия сравнения [4]: визуальное «сравнение», процент плохих пикселей, RMS (rootmeansquared) (табл. 1).



Оценка полученных результатов локального алгоритма для различных стереопар

Стереопара	RMS	Количество «плохих» пикселей, %
Tsukuba	23,6	28
Venus	9,5	59
Teddy	38,2	57
Cones	27	59

В первую очередь, в результате визуального анализа результатов выявлены ключевые проблемы локального алгоритма: обработка на краях объектах, обработка гладких областей, обработка маленьких объектов, обработка на краях изображений.

В дальнейшем планируется повышение эффективности вычисления глубины в локальном алгоритме и создание гибридного метода поиска карты глубины по стереопаре, при котором сочетается качество глобальных алгоритмов со скоростью локальных подходов.

#### Библиографический список

1. Стокман Д., Шапиро Л. Компьютерное зрение. М.: «БИНОМ. Лаборатория знаний», 2006.
2. Demoulin C., Van Droogenbroeck M. Disparity map determination by multiple adaptive windows, 2005.
3. The Middlebury Computer Vision Pages: stereo ([www.vision.middlebury.edu/stereo/](http://www.vision.middlebury.edu/stereo/)).
4. Scharstein D., Szeliski R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms // International Journal of Computer Vision, 47(1/2/3):7-42, April-June 2002. Microsoft Research Technical Report MSR-TR-2001-81, November, 2001.

УДК 004.056

**М. В. Васильев, А. С. Пруданов, А. В. Горбунов**

студенты кафедры технологий защиты информации

**К. А. Жиданов**

научный руководитель

### ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ АЛГОРИТМА ВСТРАИВАНИЯ ЦИФРОВЫХ ВОДЯНЫХ ЗНАКОВ С ПОМОЩЬЮ КОДА ХЭММИНГА В ИЗОБРАЖЕНИЯ ФОРМАТА BMP

В наш век развитых информационных технологий одной из самых актуальных задач является решение вопроса о защите авторских прав на объект интеллектуальной собственности в цифровой форме. К подобным объектам можно отнести цифровую живопись, фото-, аудио-, видеоматериалы, чертежи, цифровые книги и прочее. Существует ряд методов защиты авторских прав в цифровой среде, одним из таких методов является встраивание скрытых цифровых водяных знаков в объекты защиты.

Цифровые водяные знаки (ЦВЗ) представляют собой специальные метки, встраиваемые в цифровые файлы или сигналы с целью выявления случаев незаконного копирования и использования чужих цифровых произведений.

Водяные знаки бывают двух видов: невидимые и видимые. Последние достаточно просто удалить с помощью соответствующих редакторов, поэтому использование видимых ЦВЗ нецелесообразно в случаях, когда требуется обеспечить надежную защиту цифрового объекта. Для удаления скрытого водяного знака потребуется затратить больше усилий, но из этого не следует, что его удаление не представляется возможным. Сложность этого процесса зависит

от используемого метода встраивания и от свойств, которыми обладает данный водяной знак. При этом задача ставится таким образом, чтобы после удаления ЦВЗ объект претерпел сильные искажения, а после его внедрения видимых изменений наблюдаться не должно, т.е. было сохранено визуальное качество [1].

В данном исследовании в качестве объектов защиты определены цифровые изображения, в качестве водяного знака будут выступать кодовые конструкции кода Хэмминга [2]. Обобщенная схема процедуры встраивания и извлечения водяного знака представлена на рис. 1.

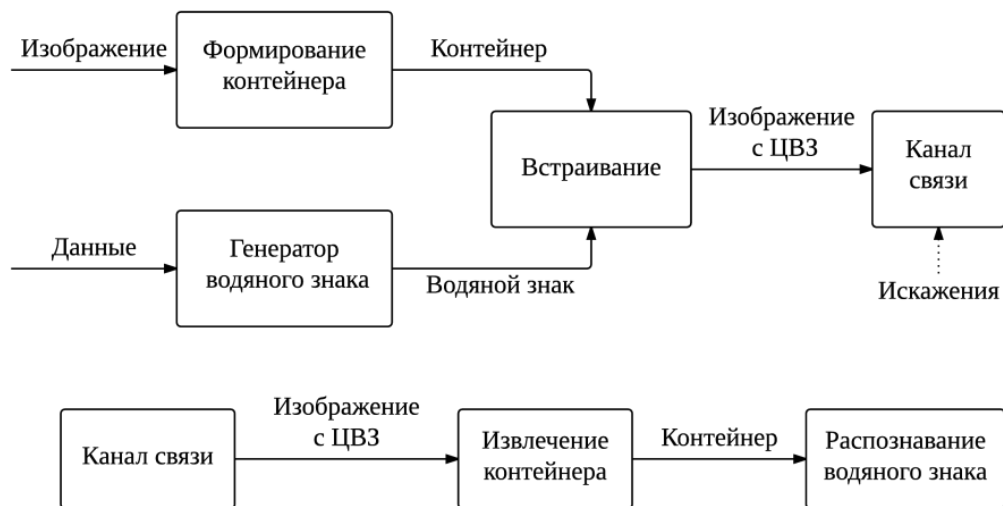


Рис. 1. Схема встраивания и извлечения ЦВЗ

Одним из широко известных методов встраивания ЦВЗ в объект является метод LSB (Less Significant Bit) [4], согласно которому контейнер формируется из младших бит каждого байта изображения. Преимуществом этого подхода является возможность встраивания водяного знака без видимых искажений изображения и предоставления большого объема контейнера.

Целью настоящего исследования является сравнение эффективности двух вариантов LSB-метода: классического и его модификации с использованием кода Хэмминга с параметрами (7, 4, 3).

Оба метода реализуют разные подходы к встраиванию знака в контейнер: стандартный LSB заносит в изображение в качестве ЦВЗ явное информационное сообщение, в то время как его модификация с кодом Хэмминга придает изображению определенный признак – наличие кодовых слов, по которому определяется, что изображение содержит водяной знак.

В модифицированной версии LSB контейнер делится на отрезки, равные длине кодового слова ( $n$ ), с последующим приведением полученных слов ( $b$ ) к кодовым словам кода Хэмминга ( $c$ , см. рис. 2). Любая последовательность из  $n$  символов является кодовым словом с возможным наличием ошибки, следовательно, из любого такого кодового слова можно получить слово кода Хэмминга, исправив ошибку путем сложения исходного слова с соответствующим вектором ошибки [2]. При таком подходе заведомо известна верхняя граница количества бит, подвергающихся изменению в процессе постановки водяного знака, и оно равно минимальному расстоянию кода Хэмминга ( $d$ ). В обычном LSB могут измениться все биты контейнера.

Таким образом, в обычном LSB контейнер заполняется текстом сообщения независимо от исходного содержимого младших бит изображения, в то время как в модифицированной версии метода содержимое контейнера формируется на основе исходных данных цифрового файла.

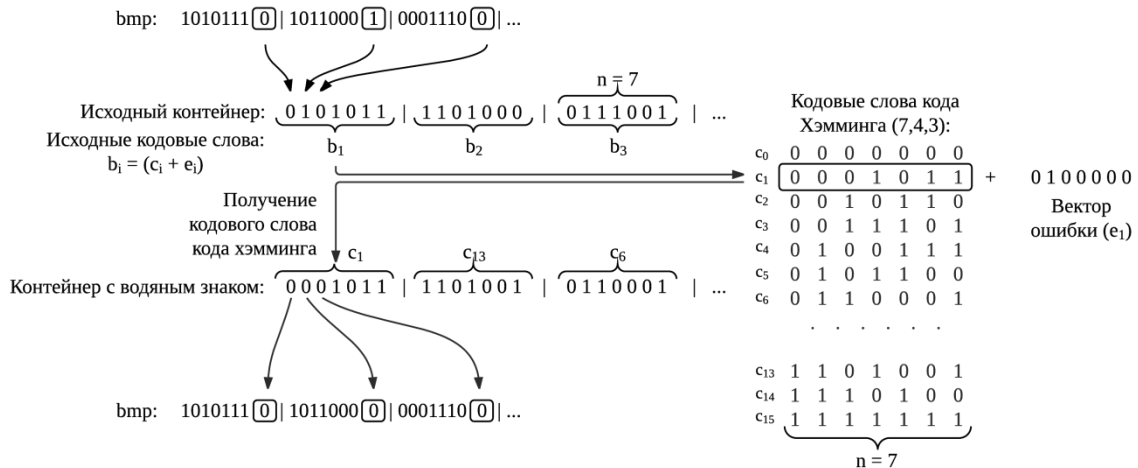


Рис. 2. Процесс встраивания водяного знака в контейнер

Для проведения оценки эффективности описываемых методов постановки ЦВЗ было взято несколько известных тестовых изображений (lena.bmp, boy.bmp, sails.bmp и др.). Качество встраивания водяного знака измерялось с помощью показателя PSNR (пиковое отношение сигнал/шум), который отражает степень искажения изображения относительно исходного после внесения в него некоторых изменений [3]. Известно, что человеческий глаз реагирует в большей степени на изменение яркостной составляющей, поэтому расчет PSNR производился только для яркости.

Данный критерий определяется выражением:

$$PSNR = \frac{20 \log_{10} \square (\max \times m \times n)}{\sqrt{\sum_{i=1, j=1}^{m, n} (x_{i,j} - y_{i,j})^2}}$$

где  $x$ ,  $y$  – значения яркости одного пикселя исходного и измененного изображений соответственно;  $\max$  – максимальное значение яркости пикселя изображения;  $m$ ,  $n$  – размерности.

Существует граничное значение, относительно которого определяется степень искажения изображения – 40 дБ. Если PSNR принимает значение от 40 дБ и выше, считается, что качество изображения приемлемо, в противном случае полагают, что изображение сильно изменилось; в этом случае искажения становятся заметны невооруженным глазом [1].

В ходе экспериментов было выявлено, что модифицированный LSB по сравнению со стандартным обладает большей устойчивостью к вносимым искажениям, что подтверждается полученным показателем PSNR (см. рис. 3, рис. 4).

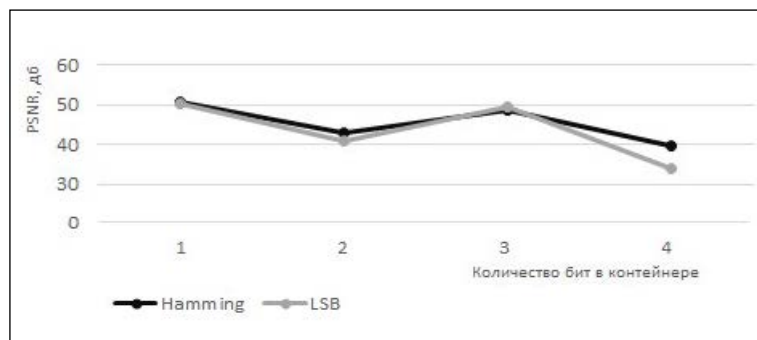


Рис. 3. PSNR изображения после встраивания ЦВЗ

На графике показано, как меняется качество изображения при встраивании водяного знака в контейнер, состоящий из разного количества младших бит каждого пикселя цифрового изображения. Для установления представленной зависимости было взято одно из тестовых изображений, из которого по заданному количеству младших бит файла составлялся контейнер. Далее в контейнер одним из методов встраивался водяной знак, после чего происходил расчет показателя PSNR для яркостной составляющей. Аналогичные действия повторялись для того же изображения, но с водяным знаком, встроенным другим способом. Эксперимент повторялся с последовательным увеличением числа младших бит до тех пор, пока не был превышен допустимый порог искажения изображения в 40 дБ.

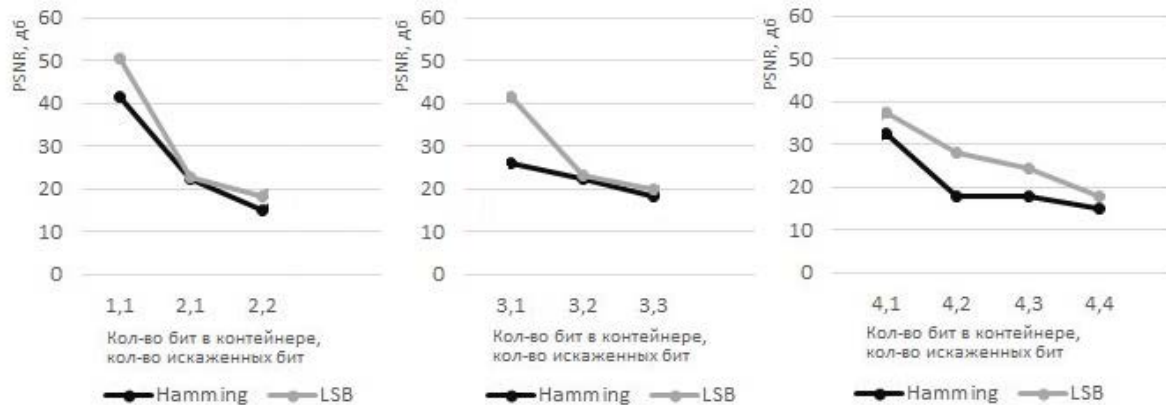


Рис. 4. PSNR изображения с водяным знаком, после воздействия искажения

В ходе данного эксперимента последовательно производился расчет PSNR для изображений с водяными знаками, которые были сгенерированы на основе контейнеров, составленных из разного количества младших бит (от одного до четырех). Для каждого такого изображения был проведен ряд тестов с увеличением числа искаженных бит. По результатам тестов было обнаружено, что изображения с водяными знаками, поставленными с использованием кодовых конструкций, изменяются сильнее под воздействием вносимых искажений по сравнению с изображениями, в которые цифровой знак был внедрен стандартным способом. Таким образом, использование кодовых конструкций повышает надежность цифрового водяного знака, так как при попытке его удаления качество изображения заметно ухудшается.

Исследование показало, что использование примитивного алгоритма встраивания водяного знака LSB и кодовых конструкций в качестве содержимого контейнера позволяет добиться повышения надежности водяного знака. Применение кодовых конструкций наряду с более сложными методами формирования контейнера позволит добиться увеличения устойчивости ЦВЗ к искажениям. В дальнейшем планируется прибегнуть к более совершенному способу встраивания водяных знаков с использованием синдромного кодирования, что даст возможность заносить в контейнер осмысленные сообщения, сохранив при этом высокую стойкость водяного знака к искажениям.

### Библиографический список

1. Аграновский А. В., Балакин А. В., Грибунин В. Г., Саложников С. А. Стеганография, цифровые водяные знаки и стеганоанализ. М.: Вузовская книга, 2009. – 220 с.
2. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. Пер. с англ. М.: Техносфера, 2005. – 320 с.
3. Статья «Пиковое отношение сигнала к шуму» на сайте wikipedia.org
4. Завьялов С. В., Ветров Ю. В. Стеганографические методы защиты информации: учеб. пособие. – СПб.: Изд-во Политехн. ун-та, 2012. – 190 с.

**Д. Н. Иванов**

студент кафедры инфокоммуникационных сетей

**А. И. Веселов**

научный руководитель

## ИССЛЕДОВАНИЕ АЛГОРИТМОВ АВТОМАТИЧЕСКОГО РАСКРАШИВАНИЯ ИЗОБРАЖЕНИЙ В ГРАДАЦИЯХ СЕРОГО

Алгоритмы автоматического раскрашивания можно применять в разных сферах. Например, как для сжатия разноцветных изображений или фильмов, так и для раскрашивания черно-белых изображений. На раскрашивание старых фильмов уходит много времени, сил и средств. Алгоритмы автоматического раскрашивания призваны облегчить этот процесс. В представленной статье проведен анализ алгоритмов автоматического раскрашивания изображений в градациях серого для выделения ключевых проблем с целью их устранения.

Работа [2] была одной из первых работ по автоматическому раскрашиванию. В этой статье было показано, что можно раскрасить черно-белое изображение методом переноса цвета с изображения-источника цвета. Для этого достаточно собрать некоторое количество образцов цвета и рассчитать вероятность их появления в той или иной области. Для описания цвета была использована оценка математического ожидания и оценка среднеквадратичного отклонения. В этой статье изображение-источник цвета уже дано, такое допущение допустимо в некоторых ситуациях, однако, в общем случае, необходимо вести поиск похожего изображения в некоторой заранее подготовленной базе.



Рис. 1. Схема процесса автоматического раскрашивания

Процесс автоматического раскрашивания состоит из нескольких этапов (рис. 1). Рассмотрим основные подходы, используемые на каждом этапе, а также определим базовые проблемы и сформулируем возможные способы их решения.

### 1. Подготовка базы данных.

В работе [4], используется сравнение сигнатур яркостной компоненты для автоматического поиска похожих изображений. Этот метод подходит тогда, когда надо выбрать наиболее похожее изображение из множества уже выбранных, похожих изображений. Однако в общем случае этот метод очень часто приводит к ошибкам из-за того, что яркостная компонента не несет никакой

информации о цвете, поэтому в базе данных алгоритм может найти два совершенно разных изображения с похожими гистограммами. Один из вариантов создания базы данных представлен на рис. 2.

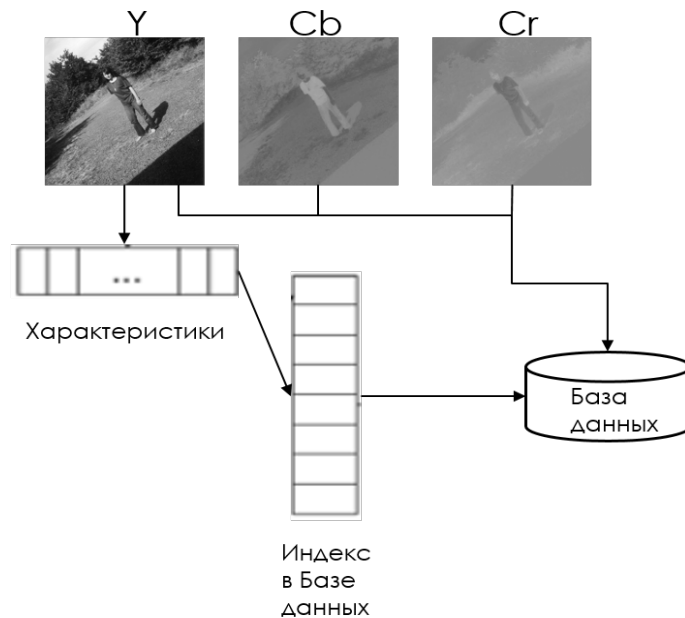


Рис. 2. Подготовка базы данных

Для того чтобы решить эту проблему, нужно определить, что следует хранить в базе данных и по каким критериям следует вести поиск. Наиболее подходящим решением будет провести поиск похожих объектов на изображениях.

### 2. Поиск похожего изображения.

Основная задача – какой выбрать критерий для схожести изображений. Нужно искать такое изображение, которое было бы схоже по цветам с черно-белой картинкой. Для этого следует проанализировать снимок и искать объекты схожие с объектами на цветной фотографии. Этот вид поиска основывается на технологии компьютерного зрения. Набор метаданных, характеризующих изображение, ограничен, а компьютерное зрение позволяет значительно расширить количество атрибутов, которые учитываются при поиске картинок.

Можно упростить задачу поиска используя классификаторы. Например, если на изображении есть лицо, то нужно искать похожее изображение среди фотографий с изображениями людей.

### 3. Перенос цвета с изображения-источника цвета на изображение в градациях серого.

В работе [2] говорится, что для удачного раскрашивания черно-белого изображения достаточно взять около 200 цветов. В общем случае, цветная картинка переводится из формата RGB в формат YCbCr, далее компонента Y разбивается на NxN блоков, из каждого блока случайным образом выбирается пиксель и сохраняются значения цветоразностных компонент этого пикселя, далее для этого пикселя рассчитываются оценка математического ожидания и оценка среднеквадратичного отклонения для области с радиусом R вокруг этого пикселя, таким образом появляется массив с образцами цвета. Далее на черно-белой картинке, для каждого пикселя строится область радиусом R и рассчитывается оценка математического ожидания и оценка среднеквадратичного отклонения [1]. После этого результат сравнивается с образцами из массива и выбирается тот образец цвета, для которого рассчитанные характеристики наиболее близки к характеристикам пикселя черно-белого изображения.

Проблема в том, что пиксели в цветном изображении выбираются случайно и поэтому возможно, что в массиве не окажется цветов некоторых предметов [1–4]. Эту проблему можно решить, например, с помощью детектора границ, и выбирать пиксели в пределах этих границ.

Детектор границ поможет решить сразу две проблемы. Во-первых, каждый объект на изображении можно будет раскрашивать своим цветом. А во-вторых, цвета, на полученном изображении, можно будет сгладить с помощью усредняющей фильтрации.

Таким образом, раскрашенное изображение, скорее всего, будет выглядеть более четким и насыщенным по сравнению с первоначальным вариантом.

Так же оценка среднеквадратичного отклонения и оценка математического ожидания не являются лучшими характеристиками для описания цвета-образца. Необходимо найти такие характеристики, которые позволяют однозначно описывать пиксель.

Как правило, при переносе цвета осуществляется жесткий выбор цвета, т. е. соседние пиксели никак не анализируются, что может приводить к ошибкам. В этом случае следует выбирать некоторое количество кандидатов цвета, и из них далее выбирать оптимальный.

#### 4. Оценка полученных результатов.

На сегодняшний день не существует однозначных методов оценки качества раскрашивания. Однако существует возможность оценить изображение как единое целое. К методам, с помощью которых можно оценить полученное раскрашенное изображение, относятся следующие:

- экспертная оценка. Человек сидит перед компьютером и делает вывод, устраивает его результат или нет. Этот вариант требует вмешательство человека, также у разных людей возможны разные мнения по одному и тому же изображению.

- PSNR. Показывает разницу между исходным изображением и восстановленным. Для этой оценки требуется иметь оригинал, а в случае с автоматическим раскрашиванием одно и то же изображение может быть раскрашено разными цветами и при этом выглядеть вполне достойно.

Для оценки полученного результата необходимо выбрать цветовую модель. Наиболее удобной представляется модель RGB по нескольким причинам:

- эта модель достаточно проста как для понимания, так и для математического описания;
- она применяется во многих технических устройствах и, при необходимости, преобразуется в другие цветовые модели;

- она близка к представлениям о природе чувствительности к цвету человеческого глаза.

Требования, предъявляемые к критериям оценки качества изображений:

- показатель оценки качества, для сравнения с критериями, должен вычисляться, то есть представлять численную оценку;

- критерий оценки качества должен опираться на характеристики, которые можно вычислить;

- значения критериев должны иметь относительный характер (не зависеть от диапазона яркости RGB);

- критерии должны быть понятны и наглядны для человека.

В данной работе были рассмотрены алгоритмы автоматического раскрашивания и представлены ключевые проблемы, встречающиеся на каждом этапе работы алгоритма. Было показано, что для работы алгоритма автоматического раскрашивания в общем случае необходимо иметь базу данных, в которой ведется поиск похожего изображения вручную, либо с помощью заранее рассчитанных признаков. После того как изображение-источник цвета будет найдено, цвета с цветного изображения необходимо перенести на черно-белое изображение. Важно подобрать такое количество цветов, которое сможет достоверно раскрасить черно-белое изображение. Далее полученный результат необходимо оценить. На данный момент не существует способа однозначно оценить качество переноса цвета. В некоторых случаях можно использовать существующие критерии оценки искажений в изображениях, например PSNR и экспертную оценку.

Разработка автоматических методов оценки окрашивания изображений, а также развитие данных подходов для обработки видео являются предметом дальнейших исследований.

### Библиографический список

1. Окрашивание изображений. 2013 г. (<http://habrhabr.ru/post/183226>)
2. Welsh T., Ashikhmin M., Mueller K. Transferring color to greyscale images // ACM SIGGRAPH, 2002. С. 277–280.
3. Aurelie Bege, Vinh-Thongau Ta. Patch-based Image colorization // International Conference on Pattern Recognition, Tsukuba, Japan, 2012.
4. Viera, Fully automatic coloring of grayscale images // Image and Vision Computing 25 (2007). С. 50–60, L.F.M. et al, 2007.

УДК 004.415.53, 004.056

**В. И. Ильин**

студент кафедры технологий защиты информации

**Р. Ф. Жаринов**

научный руководитель

## ИССЛЕДОВАНИЕ ИНСТРУМЕНТОВ ЭМУЛЯЦИИ НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ ВЕБ-ПРИЛОЖЕНИЙ

Современная ИТ-система должна обеспечивать необходимый уровень производительности. Любого рода задержки, отказы и сбои могут служить причиной колоссальных финансовых потерь, потери репутации, а также действующих и потенциальных клиентов. Чтобы избежать потерь и защитить вашу систему от сбоев, необходимо проводить нагрузочное тестирование.

Нагрузочное тестирование – подвид тестирования производительности, при котором осуществляется нагрузка на сайт, путем моделирования поведения большого количества пользователей одновременно. Нагрузочное тестирование, прежде всего, направлено на проверку качественной и бесперебойной работы системы под нагрузками. Своевременное проведение нагрузочного тестирования – обязательный этап в обеспечении стабильной работы любого современного Веб-приложения. Моделирование нагрузки происходит с применением определенных продуктов и техник, выбор которых определяется индивидуально.

### Этапы нагрузочного тестирования

Главной задачей нагрузочного тестирования является наблюдение за показателями производительности системы при определенной нагрузке. Отчет о результате тестирования включает в себя информацию о максимальной производительности системы, результатах мониторинга таких параметров производительности, как: времена отклика пользовательских операций и загрузка системных ресурсов.

Типовая структура проведения нагрузочного тестирования состоит из:

- создания методики нагрузочного тестирования (формулировка сценариев нагрузки – выбор вида нагрузки: линейная, постоянная, ступенчатая, комбинированная; определение набора значимых с точки зрения нагрузки, операций: формирование отчета, регистрация, отображение представления, сложные запросы к базе данных, и частоты их выполнения – интенсивности);
- создания скриптов, для каждой операции (описание действий для эмулируемых пользователей – зайти на страницу затем заполнить форму после чего отправить запрос на генерацию отчета);
- проведения нагрузочных тестов (повышение нагрузки, с добавлением новых пользователей через определенный интервал времени);
- обработки результатов тестирования (результаты тестирования сохраняются в обычных текстовых журналах, сгруппированных по директориям для отдельных тестов, во время теста



специальный модуль организует вывод результатов в консольный интерфейс в виде таблиц, одновременно запускается локальный веб-сервер, позволяющий видеть те же самые результаты на информативных графиках. По окончании теста возможно автоматическое сохранение результатов на сервисе Loadosophia.org. Также имеется модуль загрузки результатов в хранилище Graphite) [1].

Каждый из этих этапов рассматривается для Веб-приложения индивидуально, исходя из его архитектуры. В зависимости от типа частых запросов к серверу, используемой базы данных (реляционная или не реляционная) и внутренней логики приложения, нагрузочное тестирование будет иметь свои характерные особенности.

#### Анализ существующих утилит

Существует множество причин, способных вызвать сокращение производительности Веб-сервиса, или же вовсе его полную недоступность. Это может быть связано с нехваткой серверных ресурсов, неправильным построением архитектуры, плохой масштабируемостью. Все эти причины могут быть исправлены, оптимизированы или временно перераспределены, в то время как неготовность к DDoS-атакам может остановить работу дата-центра на время проведения атаки. Для сравнения, по результатам исследования компании Ponemon Institute, одна минута простоя дата-центра выливается в значительную сумму – 7900 долл. США [2], кроме того, цена простоя увеличивается из года в год.

Одним из наиболее популярных среди атак типа «отказ в обслуживании» является отправка большого числа HTTP пакетов. Принцип этой атаки заключается в отправке огромного количества запросов к серверу, которые он не способен обработать. Имитация данной атаки и происходит при нагрузочном тестировании, и на текущий момент существует множество инструментов, способных эмулировать поведение большого числа пользователей. Инструмент для тестирования должен иметь хорошую документацию, простой язык для написания скриптов и стабильно работать при проведении длительных тестов со сложными сценариями нагрузки. Для сравнения уже существующих решений были выбраны наиболее популярные и функциональные инструменты, сравнение которых представлено в таблице.

Таблица

Сравнение инструментов нагрузочного тестирования

Утилита\Метрика	Apache JMeter	Tsung	Яндекс-Танк	Joe Dog Siege
Open Source	+	+	+	+
Кроссплатформенность	+	*nix	Linux	*nix
Графическое отображение	+	+	+	–
Сценарии нагрузки	+	+	+	только простые скрипты
Ресурсоемкость	ЦПУ: 6 Гц ОЗУ: 8 Гб	ЦПУ: 1,8 Гц ОЗУ: 4 Гб	ЦПУ: 1,8 Гц ОЗУ: 4 Гб	ЦПУ: 8 Гц ОЗУ: 12 Гб
Стабильная работа длительных тестов (часы)	5,5	> 15	> 15	4,5
Модульность	нет	нет	да	нет
Язык программирования	Java	Erlang	Python	Perl
Документированность	+/- неактуальность	–	+	–
Язык для правил	GUI	Xml	ini ключ/значение	запрос

В результате, были выявлены два лидера – это Tsung и Яндекс.Танк. Для окончательного выбора утилиты было проведено дополнительное сравнение двух инструментов, по результатам которого был выбран Яндекс.Танк. Его главным преимуществом является возможность использования сетевого стека Linux на 100%, что значительно увеличивает максимально возможное количество запросов в секунду. Не менее важное преимущество – модульная архитектура Яндекс.Танка, позволяющая использовать различные генераторы нагрузок. Яндекс.Танк – это скорее обертка над генераторами нагрузки, в которую можно подключить свои модули для получения любых нужных функции [3].

#### Практическое применение

Яндекс.Танк был установлен на тестовый сервер с частотой процессора 4 ГГц, оперативной памятью 4 Гб, в качестве операционной системы использовалась Ubuntu 14.04. Для проведения нагрузочного тестирования на локальный сервер, в качестве front-end был установлен Nginx, в качестве back-end: apache2, php и mysql. В случае применения схемы этой схемы, Nginx, предназначен для раздачи статического контента, большому количеству клиентов, при незначительном потреблении системных ресурсов. После того как приходит запрос клиента, Nginx передает запрос Apache и получает ответ. А в случае со статическим контентом (html, картинки, файлы промежуточного буфера и пр.) Nginx самостоятельно сформирует ответ, не используя ресурсы Apache. В качестве системы управления контентом были выбраны Drupal 7 и OpenCart, благодаря своей популярности. Все эти средства имели стандартную конфигурацию из коробки. После их установки была подана нагрузка с линейным увеличением количества запросов от 1 до 1000 в течение 15 мин. При нагрузке в 516 запросов в секунду на тестируемом сервере начали появляться ошибки, с кодом ответа 502 Bad Gateway «ошибочный шлюз» [4] и 500 Internal Server Error «внутренняя ошибка сервера» [4], рост числа которых привел к дальнейшему отказу сервера. В результате, тестируемый сайт, как и сервер в целом, перестал отвечать уже через 7 мин.

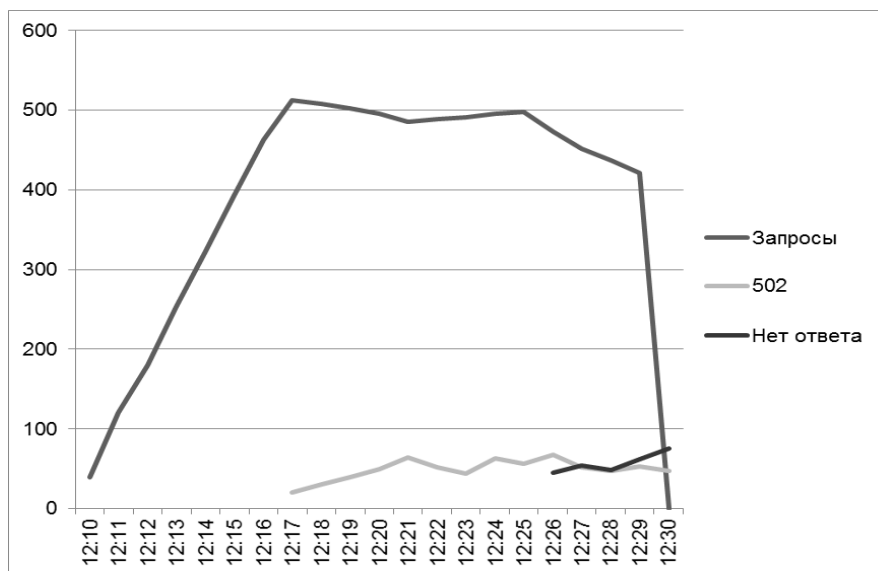


Рис. 1. Нагрузка на тестируемый сервер

#### Методы защиты

Для повышения производительности сервера были исследованы уже разработанные методы:

– фильтрация и блокировка трафика – возможна реализация с помощью модуля для Nginx – Naxsi, который является фаерволом для Веб-приложений (WAF);

– построение распределенной системы – разные уровни приложения будут находиться на разных серверах;

– Alternative PHP Cache, использование которого увеличит производительность интерпретатора благодаря кешированию байт-кода PHP сценариев.

Оптимизация базы данных заключается в использовании сервисов кеширования, которые хранят данные в оперативной памяти, таких как Memcached и Redis, а также настройке репликации, горизонтального и/или вертикального шардинга.

В результате исследования была смоделирована архитектура, включающая в себя сочетание различных методов оптимизации. На рис. 2 приведена схема данной архитектуры.

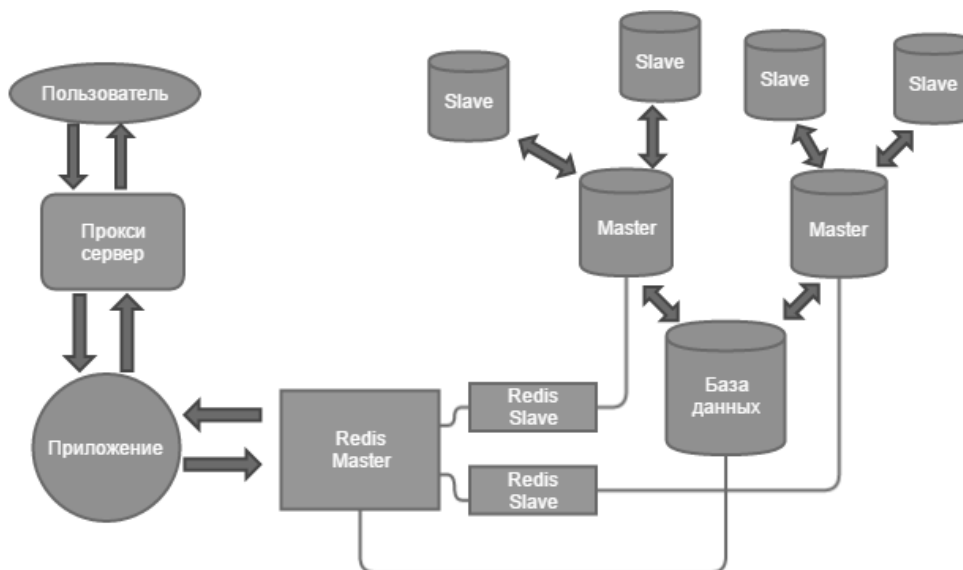


Рис. 2. Схема архитектуры высоконагруженного Веб-приложения

Пользователь посылает запросы к приложению через прокси сервер, главная задача которого – раздача статического контента. Помимо этого, при существовании других приложений прокси-сервер будет выполнять функцию балансировки нагрузки. Приложение осуществляет доступ к базе данных через реплицированный Redis. К БД применен вертикальный шардинг – таблицы, к которым приложение обращается чаще всего, вынесены на различные сервера и репликация – каждая вынесенная таблица дублируется на отдельном сервере, запросы на чтение и запись будут обращаться к разным серверам.

В результате, мы сможем обеспечить стабильную работу приложения в периоды регулярной и пиковой нагрузки, а также минимизировать риски при внедрении новых релизов систем.

### Заключение

Таким образом, своевременное проведение нагрузочного тестирования любого Веб-приложения поможет избежать последствий DoS атаки: простоя, ошибок, и медленной работы сервиса, которые несут прямые убытки и потерю клиентов. Для проведения тестирования лучше всего подойдет Яндекс.Танк, ввиду возможности стабильной эмуляции поведения реальных пользователей на сервер с помощью различных генераторов нагрузки на протяжении длительного времени. DoS-атака, с применением этого инструмента, смогла на время вывести наш тестируемый сервер из строя, который не смог выдержать нагрузки выше 516 запросов в секунду. Для увеличения производительности и отказоустойчивости Веб-сервера была смоделирована архитектура высоконагруженного проекта, включающая в себя сочетание различных методов

оптимизации базы данных и приложения в целом. В качестве дальнейшего исследования данная архитектура будет применена к популярным системам управления контентом, на которых написано большинство интернет-магазинов. После чего будет создан нагрузочный кластер для проведения DDoS-атаки с применением облачных ресурсов для обработки отчетов, что сделает работу с кластером значительно легче.

### Библиографический список

1. Гильмуллин Т. «Яндекс.Танк и автоматизация нагрузочного тестирования», [http://www.securitylab.ru/analytics/448244.php], дата обращения: 26-05-2015
2. Компания King Servers, «Результаты одного исследования: минута простоя среднестатистического дата-центра стоит 7900 долларов США», [http://habrahabr.ru/company/kingservers/blog/205082/], дата обращения: 26-05-2015.
3. Компания Яндекс, «Официальная документация Яндекс.Танк», [https://tech.yandex.ru/tank], дата обращения: 26-05-2015.
4. Компания Яндекс «Справочник кодов ошибок», [https://help.yandex.ru/webmaster/error-dictionary/http-codes.xml], дата обращения: 26-05-2015

УДК 004.627

**Н. В. Устюжанин**

студент кафедры инфокоммуникационных систем

**М. Р. Гильмутдинов**

канд. техн. наук, доц. – научный руководитель

## АНАЛИЗ АЛГОРИТМОВ МОДЕЛИРОВАНИЯ ПЕРЕДНЕГО ПЛАНА И ФОНА В ЗАДАЧЕ ДЕТЕКТИРОВАНИЯ ДВИЖЕНИЯ

В повседневной жизни активно используются системы видеонаблюдения с большим количеством камер (например, в торговых центрах с большими площадями, на многоуровневых парковках, на улицах). Для автоматизации работы оператора в подобные многокамерные системы в последнее время активно внедряются алгоритмы видеоаналитики, которые позволяют обнаружить появление нежелательных объектов (с помощью алгоритма Детектирования Движения), осуществлять их идентификацию и сопровождение (трекинг). В данной работе рассматривается алгоритм моделирования фона, применяемый в задаче детектирования движения. Данный класс алгоритмов используется в современных стандартах сжатия видеопоследовательностей. Например, в стандарте IEEE 1857 [1]. Также моделирование фона может использоваться в задаче интерполяции видеокладов [2]. Среди множества готовых решений можно выделить свободно распространяемые (OpenCV [3]) и коммерческие (ViBe [4]).

Моделирование фонового кадра  $B_t$  производится по некоторой предыстории:

$$B_t = F(I_t, I_{t-1}, \dots, I_0) \quad (1)$$

Для рассмотрения были взяты два класса алгоритмов моделирования фона:

- статистический, основанный на накоплении статистики;
- вероятностный, на базе смеси нормальных распределений.

Статистический метод основан на накоплении статистики из  $N$  предыдущих кадров и ее анализе. Используя накопленные данные, моделируется фон. В данной работе использовались два способа моделирования по накопленной статистике:

1) усреднение:

$$B_t(x, y) = \frac{1}{N} \sum_{i=1}^N I_{t-i}(x, y) \quad (2)$$

где  $I_t$  – некоторый кадр видеопоследовательности,  $B_t$  – модель фона.

2) нахождение медианы – элемент с индексом  $\frac{N}{2}$  упорядоченной выборки.

Усреднение является вычислительно более простой операцией по сравнению с медианой, однако дает погрешности при постоянно движущихся объектах в кадре и случайных изменениях яркости. Медиана не обладает данными недостатками. На иллюстрации ниже приведено сравнение моделей фона, полученных с помощью приведенных методов по одинаковым выборкам.



Рис. 1. Сравнение моделей фона, полученных с помощью статистического алгоритма при использовании разных способов моделирования: а) усреднение, б) медиана

Для алгоритма на базе смеси нормальных распределений следует ввести ряд допущений:

Допущение 1:  $Y_t(x, y) \approx N(m_i, \sigma_i^2)$ , где  $i = 1 \dots K$ ,  $K$  – количество процессов.

Допущение 2: каждый из процессов описывается нормальным распределением  $N(m, \sigma^2)$  с некоторыми параметрами математического ожидания  $m$  и дисперсии  $\sigma^2$  и весом  $\omega_i$ . Вес отражает частоту использования процесса в смеси.

Допущение 3: сумма весов процессов, которыми описывается пиксель, равна единице:

$$\sum_{i=1}^K \omega_i = 1 \quad (3)$$

Типовая схема работы алгоритма представлена на рис. 2. Описание подходов приведено в статьях [5, 6].



Рис. 2. Схема алгоритма на базе смеси нормальных распределений

Чем больше кадров обрабатывается, тем точнее оценки математического ожидания  $m$  и дисперсии  $\sigma^2$ . Базовый метод моделирования фона выражается следующей формулой:

$$B_t(x, y) = m_R,$$

где  $R = \operatorname{argmax}\{\omega_i\}(x, y)$ ,  $m_R$  – оценка математического ожидания процесса с наибольшим весом для пикселя с координатами  $x, y$ .

Результатом работы вероятностного алгоритма является бинарная карта  $BM_t$ , в которой значению 0 соответствует пиксель фона, а 1 – переднего плана. На основании данной информации была предложена модификация метода:

$$B_t(x, y) = \begin{cases} I_t(x, y), & BM_t(x, y) = 0 \\ m_R, & BM_t(x, y) = 1 \end{cases} \quad (5)$$

При таком способе моделирования фона уменьшается ошибка, вносимая среднеквадратичным отклонением процесса с наибольшим весом, так как значения копируются из исходного кадра, а не из модели. Эффективность моделирования фона оценивается с помощью двух критериев:

1) PSNR между эталоном, выбранным вручную на видеопоследовательности, (ground truth –  $GT$ ) и фоном, сформированным с помощью модели;

2)  $EP$  – количество пикселей, разность которых больше порога  $T$  между эталонным фоном и моделью.



а)



б)

Рис. 3. Сравнение моделей фона, полученных с помощью алгоритма на базе смеси нормальных распределений при использовании разных способов моделирования:

а) базовый метод (PSNR = 41.15 дБ, EP = 708), б) предложенный метод (PSNR = 43.36 дБ, EP = 66)

Для получения переднего плана для статистического алгоритма сначала производится межкадровая разность:

$$D_t(x, y) = |I_t(x, y) - B_t(x, y)|, \quad (6)$$

где  $D_t$  – межкадровая разность.

Затем формируется бинарная карта по порогу:

$$BM_t(x, y) = \begin{cases} 1, & D_t(x, y) > T \\ 0, & D_t(x, y) \leq T \end{cases} \quad (7)$$

где  $T$  – порог оценки межкадровой разности.

После этого передний план  $F_t$  формируется по следующей формуле:

$$F_t(x, y) = I_t(x, y) BM_t(x, y). \quad (8)$$

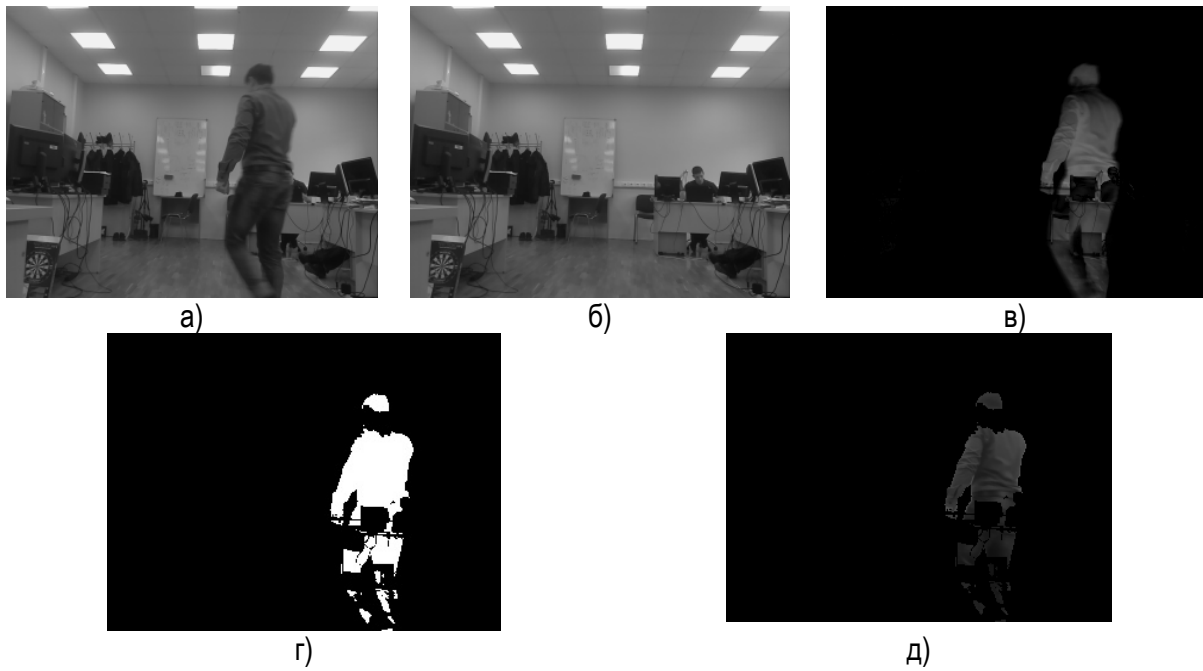


Рис. 4. Процесс получения переднего плана для статистического алгоритма:  
а)  $I_t$ , б)  $B_t$ , в)  $D_t$ , г)  $BM_t$ , д)  $F_t$

Так как для вероятностного алгоритма результатом работы является бинарная карта, передний план можно восстановить без использования кадра фона. Однако на данной бинарной карте присутствуют артефакты, которые нужно устранить. Для их устранения применяются алгоритм размывания [7] и медианный фильтр. Результатом фильтрации является кадр  $BMF_t$ .

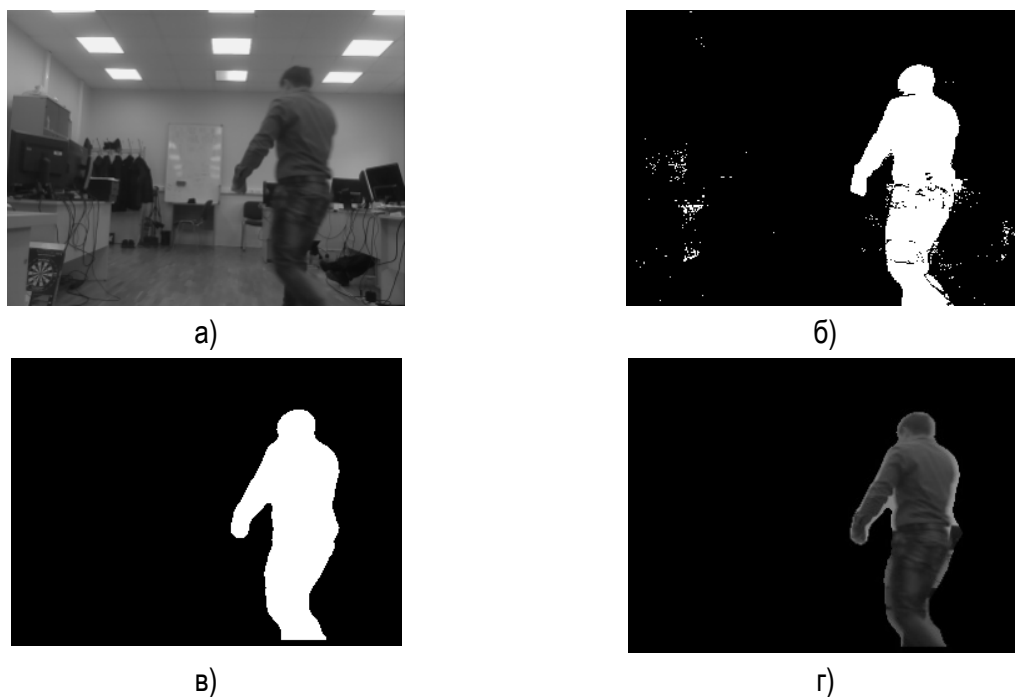


Рис. 5. Процесс получения переднего плана для вероятностного алгоритма:  
а)  $I_t$ , б)  $BM_t$ , в)  $BMF_t$ , г)  $F_t$

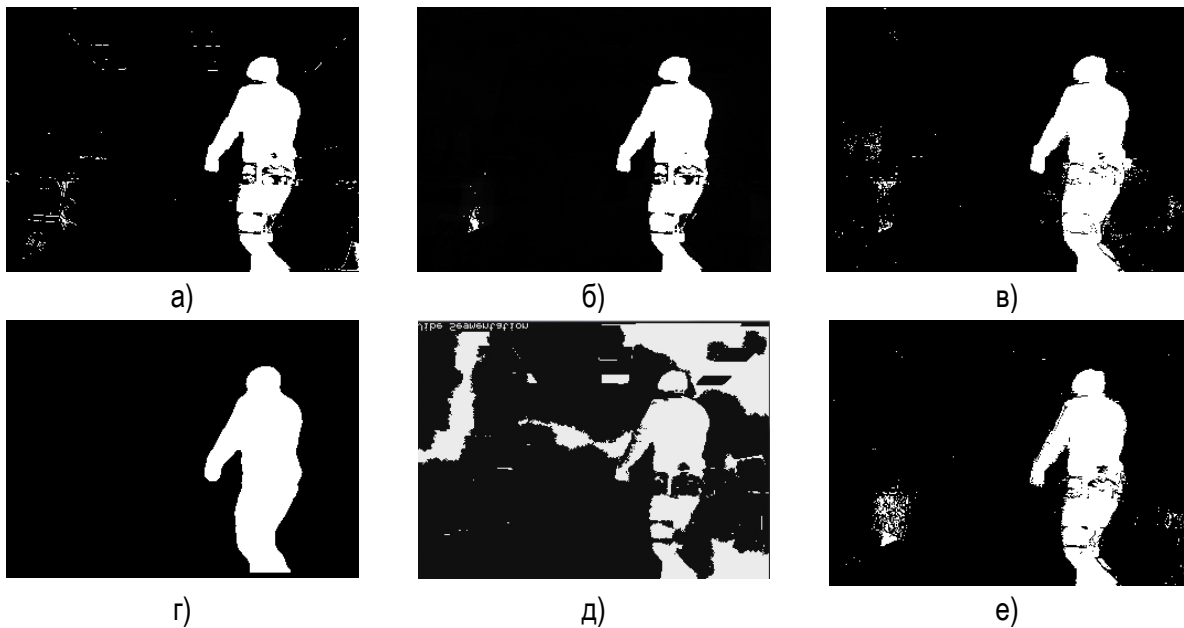


Рис. 6. Сравнение карт передних планов, полученных разными алгоритмами:  
 а) статистический метод (усреднение), б) статистический метод (медиана),  
 в) алгоритм на базе смеси нормальных распределений (без фильтрации),  
 г) алгоритм на базе смеси нормальных распределений (с фильтрацией), д) ViBe [2],  
 е) алгоритм на базе смеси нормальных распределений (mog2) из пакета OpenCV [3]

Для оценки качества переднего плана был использован метод сравнения с эталонной картой ( $BGT$ ) переднего плана, сделанной вручную по следующей формуле:

$$BEM_t(x,y) = BGT_t(x,y) \oplus F_t(x,y), \quad (9)$$

где  $BEM$  – бинарная карта ошибок.

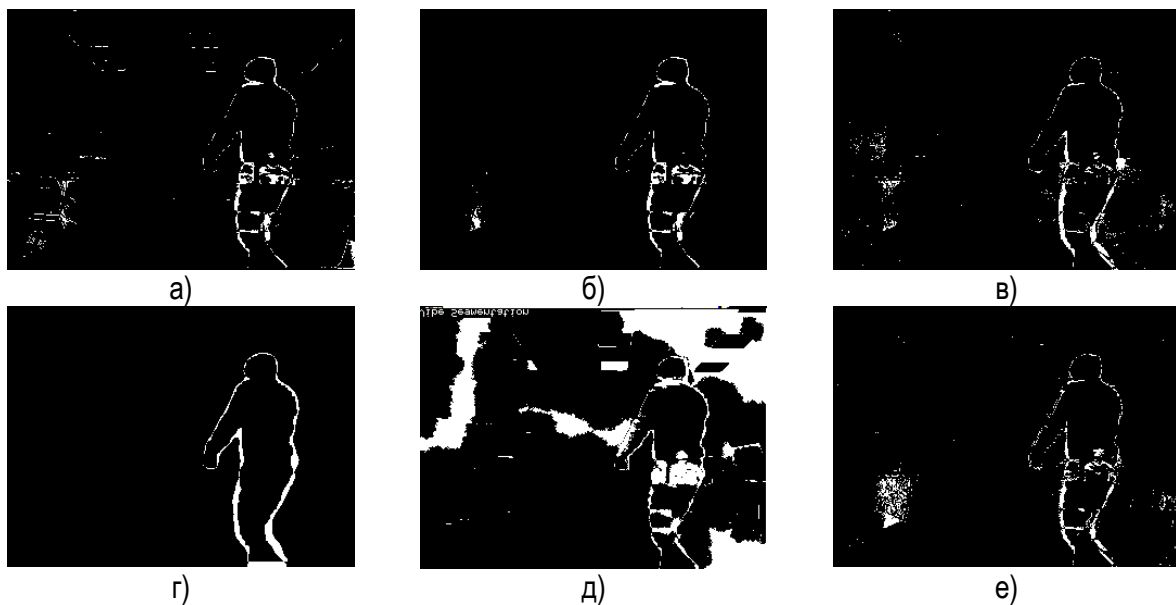


Рис. 7. Сравнение ошибок моделирования фона для разных алгоритмов:  
 а) статистический метод (усреднение), б) статистический метод (медиана),  
 в) алгоритм на базе смеси нормальных распределений (без фильтрации),  
 г) алгоритм на базе смеси нормальных распределений (с фильтрацией), д) ViBe [2],  
 е) алгоритм на базе смеси нормальных распределений (mog2) из пакета OpenCV [3]



Белые пиксели на карте ошибок отображают ложные срабатывания алгоритмов. Прежде всего, это связано с разной чувствительностью алгоритмов, спецификой классификации пикселей и постобработкой карт переднего плана (например, как показано на рис. 5). Особое внимание стоит уделить ошибочным областям, которые не относятся к объектам.

Таблица 1

Сравнение скорости работы реализованных методов

Алгоритм	Статистический		Вероятностный	
	усреднение	медиана	без фильтрации	с фильтрацией
Скорость работы (кадров/сек)	150	70	150	120

В данной работе рассмотрены два алгоритма моделирования фона и выделения переднего плана для задачи детектирования движения, а также приведены оценки эффективности количественных и качественных критериев, и приведено сравнение используемых методов с аналогами.

### Библиографический список

1. Yu L, Chen S., Wang J. Overview of AVS-video coding standards. Signal Process. Image Commun. 24, 247–262 (2009).
2. Устюжанин Н. В. «Анализ методов оценки движения в задаче интерполяции видеок кадров», СПб., 67 Международная Студенческая Научная Конференция ГУАП. Ч. 1, 2014. С. 216–219.
3. Olivier Barnich and Marc Van Droogenbroeck «ViBe: A universal background subtraction algorithm for video sequences», Member, IEEE, 2011.
4. The OpenCV Reference Manual, June 25, 2014.
5. Stauffer C., Grimson W. «Adaptive background mixture models for real – time tracking», Proc IEEE Conf on Comp Vision and Patt Recog (CVPR 1999)1999. P. 246–252.
6. Jain V, Kimia B, Mundy J. «Background modelling based on subpixel edges», ICIP 2007, San Antonio, USA, September 2007. P. 321–324.
7. Гонсалес Р., Вудс Р. «Цифровая обработка изображений», Москва: Техносфера, 2005. С. 759–765.

УДК: 004.056.5

**С. С. Шевелев**

студент кафедры безопасности информационных систем

**Р. Ф. Жаринов**

научный руководитель

### ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ДЛЯ МОБИЛЬНОЙ АУТЕНТИФИКАЦИИ НА ВЕБ-СЕРВИСАХ

В среднем пользователь посещает десятки Веб-сервисов в день [1], используя при этом свой персональный компьютер, ноутбук или мобильное устройство. И, как правило, для того чтобы войти в систему под своим аккаунтом, нужно ввести свой логин и пароль, т. к. на большинстве сервисов используется простая парольная защита.

Регулярное использование Веб-сервисов сталкивает пользователей с проблемой запоминания паролей, которую большинство из них решают двумя способами:

- использование одних и тех же легких паролей;
- создание «памятки» с паролями.

К примеру, можно привести статистику [2] – 25 самых популярных паролей 2014 г.

Статистика популярных паролей

Место	Пароль	В сравнении с 2013 г.
1	123456	Не изменялся
2	password	Не изменялся
3	12345	Поднялся на 17 позиций
4	12345678	Опустился на 1 позицию
5	qwerty	Опустился на 1 позицию
6	123456789	Не изменялся
7	1234	Поднялся на 9 позиций
8	baseball	Новый
9	dragon	Новый
10	football	Новый

Использование таких легких паролей ставит под угрозу защиту личных данных, т. к. подбор пароля не составляет особой сложности [3]. В свою очередь использование различных сложных паролей затруднительно и неудобно. Особенно это касается мобильных устройств, на которых одновременный ввод букв разного регистра, цифр и специальных символов затруднен.

#### Исследование существующих решений

Для того чтобы отказаться от необходимости создания, набора и запоминания длинных и сложных паролей, на сегодняшний день существуют множества решений. Вот некоторые из них:

- для физических лиц:
  - платформа Digits [4];
  - «Яндекс Ключ» [5];
  - аутентификация посредством одноразового кода, переданного через SMS;
  - комплекс Google Authenticator [6];
  - технология True Key от Intel Security [7];
- для юридических лиц:
  - USB-ключи и смарт-карты eToken компании Алладин [8];
  - системы генерации одноразовых паролей eToken PASS и eToken NG-OTP компании Алладин [8].

Ниже представлено сравнение этих технологий. В первой таблице стоит обратить внимание на то, что не каждое из решений может идентифицировать пользователя [3]. Также не каждое из них обладает мобильностью, т. е. позволяет авторизоваться как на компьютере, так и на мобильном устройстве.

Таблица 2

Сравнение существующих решений

	Одноразовый код через SMS	USB-ключи и смарт-карты eToken	Системы генерации одноразовых паролей eToken PASS (NG-OTP)	Google Authenticator	Яндекс ключ
Только программная реализация	Да	Нет	Нет	Да	Да

## ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ЗАЩИТА ИНФОРМАЦИИ

Стоимость	1.1 руб. за SMS (для поставщика)	От 1500 руб.	От 2500 руб.	Бесплатно	Бесплатно
Идентификация субъекта	Нет	Да	Нет	Нет	Да
Мобильность	Да	Нет	Да	Нет	Нет

Из табл. 3 видно, что только в решении «USB-ключи и смарт-карты eToken» есть поддержка электронной подписи [9]. Также стоит обратить внимание на немаловажный критерий – время авторизации, т. к. для большинства простых пользователей удобство использования стоит на первом месте.

Таблица 3

Техническое сравнение существующих решений

	Одноразовый код через SMS	USB-ключи и смарт-карты eToken	Системы генерации одноразовых паролей eToken PASS (NG-OTP)	Google Authenticator	Яндекс ключ
Количество памяти	420 символов	72 кб	72 кб	5 символов	5 символов
Количество циклов перезаписи	–	Не менее 500 000	Не менее 500 000	–	–
Использование ЭП	–	ГОСТ Р 34.10-2001	–	–	–
Время авторизации	~ 45 сек	От 1 сек	~ 10 сек	~ 15 сек	~ 15 сек

Данные решения избавляют пользователя от необходимости запоминать сложные пароли и позволяют увеличить безопасность за счет использования нескольких факторов (многофакторная аутентификация) [10]. Но у всех этих решений есть одна проблема – мобильность. Каждая из этих систем идентифицирует пользователя на определенных устройствах. Например, «Яндекс Ключ» позволяет идентифицировать пользователя на персональном компьютере с помощью мобильного устройства. Авторизоваться на самом мобильном устройстве с помощью этой технологии уже не получится. Другой пример, платформа Digist. Данная система позволяет авторизоваться только на мобильных устройствах, с персональными компьютерами она вообще никак не взаимодействует.

Разрабатываемый программно-аппаратный комплекс

Для решения этой проблемы был разработан программно-аппаратный комплекс для мобильной аутентификации на Веб-сервисах, где в качестве средства аутентификации выступает бесконтактная смарт-карта [11], которая взаимодействует с внешними устройствами по бесконтактной технологии RFID (ISO 14443) [12]. Принцип взаимодействия представлен на рис. 1.

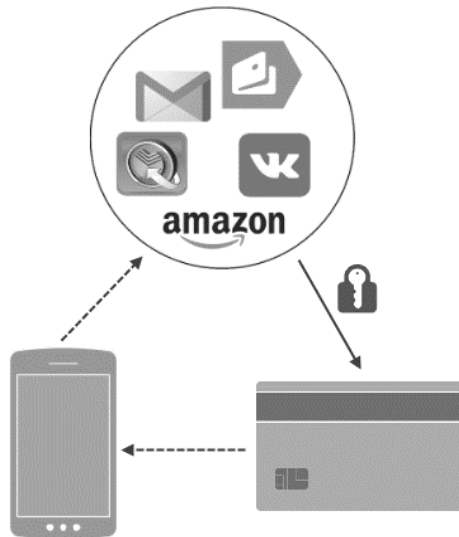


Рис. 1. Принцип взаимодействия

Идентифицирующая информация о пользователе хранится на его смарт-карте в защищенной памяти [13] и на Веб-сервисах. Для авторизации достаточно поднести смарт-карту к считывающему устройству (компьютера или мобильного устройства) и ввести ПИН код.

В качестве протокола регистрации и авторизации использовалась модифицированная версия протокола OPACITY [14], который был разработан для бесконтактных платежей. В настоящее время данный протокол официально зарегистрирован в качестве протокола аутентификации ISO/IEC 24727-6 [15]. В его основе лежит криптография на эллиптических кривых, где используется гораздо меньшая длина ключа по сравнению с классической асимметричной криптографией. В условиях ограниченной вычислительной мощности смарт-карты и небольшого объема памяти, это играет большую роль. В качестве примера приведена таблица времени генерации ключей EC и RSA на одной и той же смарт-карте.

Таблица 4

Время генерации ключей на смарт-карте

	Длина ключа, бит	Время генерации, мс
RSA	512	537,87
RSA	1024	2050,05
RSA	2048	5110,37
EC	160	649,80
EC	192	691,12

### Заключение

Использование смарт-карты с бесконтактным интерфейсом связи позволяет ее использовать как на мобильных устройствах, так и на персональных компьютерах. Тем самым решается проблема мобильности. Для авторизации на Веб-сервисе достаточно поднести смарт-карту к считывателю, с которым установка связи и обмен данными происходит менее чем за одну секунду. В результате данное решение позволяет также быстро и удобно авторизоваться на Веб-сервисе, как использование легкого пароля, и также безопасно, как использование выше перечисленных технологий двухфакторной аутентификации.

**Библиографический список**

1. Identity Theft Resource Center Breach Report Hits Record High in 2014. <http://www.idtheftcenter.org/ITRC-Surveys-Studies/2014databreaches.html/>.
2. The 25 Most Popular Passwords of 2014: We're All Doomed. <http://gizmodo.com/the-25-most-popular-passwords-of-2014-were-all-doomed-1680596951/>.
3. Фергюсон Н., Шнайер Б. Практическая криптография. 2005 год. 416 с.
4. Digits latform. <https://get.fabric.io/?locale=ru/>.
5. Двухфакторная аутентификация на Яндексе. <https://yandex.ru/promo/2fa/>.
6. Google Authenticator. <https://support.google.com/accounts/answer/1066447?hl=ru/>.
7. True Key. <https://www.truekey.com/>.
8. Компания "Аладдин Р.Д.". <http://www.aladdin-rd.ru/>.
9. Рябко Б. Я., Фионов А. Н. Основы современной криптографии. – «Научный Мир», 2004. – 173 с. – ISBN 5-89176-233-1.
10. Ричард Э. Смит. Аутентификация: от паролей до открытых ключей = Authentication: From Passwords to Public Keys First Edition. – М.: Вильямс, 2002. – С. 432. – ISBN 0-201-61599-1.
11. Rankl W., Effing W.: Smart card handbook. John Wiley & Sons, West Sussex, England (2003)
12. ISO/IEC 14443. [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=28728/](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=28728/).
13. Угрюмов Е. П. Цифровая схемотехника БХВ-Петербург (2005)
14. About Smart Cards : Industry Technical Contributions: OPACITY. <http://www.smartcardalliance.org/smart-cards-contributions-opacity/>.
15. ISO/IEC 24727-6:2010. [http://www.iso.org/iso/ru/catalogue\\_detail?csnumber=51572](http://www.iso.org/iso/ru/catalogue_detail?csnumber=51572).

## СОДЕРЖАНИЕ

### ТЕХНИЧЕСКИЕ НАУКИ

АЭРОКОСМИЧЕСКИЕ ПРИБОРЫ И СИСТЕМЫ.....	3
<i>Асенчик А. Ю.</i> Исследование программного обеспечения для решения задач имитационного моделирования транспортных систем.....	3
<i>Барбашова Л. И.</i> Исследование сингулярно-спектральных алгоритмов фильтрации сигналов.....	6
<i>Васильев Д. В.</i> Исследование алгоритмов одноклассовой классификации для мониторинга состояния технических объектов.....	12
<i>Виноградов А. Ю.</i> Анализ характеристик моста Ethernet–SpaceWire.....	17
<i>Гоголев К. В.</i> Сравнительный анализ прикладных пакетов программ для планирования грузовых перевозок на автомобильном транспорте.....	20
<i>Григорьев А. П., Чернелевский А. О.</i> Сценарный метод контроля знаний при реализации компетентностного подхода на теоретическом этапе обучения авиационных специалистов.....	23
<i>Донхан Ким</i> Методы построения тренажеров для пилотов вертолетов.....	26
<i>Зайченко В. С.</i> Система обеспечения безопасности движения транспорта на аэродроме.....	30
<i>Колокольцова М. С.</i> Принципы формирования логистической цепи передвижения в мегаполисе.....	33
<i>Кононенко А. В., Мухина Е. К.</i> Исследование цифровой фильтрации сигналов микромеханических датчиков.....	36
<i>Курбанов Л. И.</i> Проектирование буферного пространства для встроенных сетей–на–кристалле и систем–в–корпусе.....	41
<i>Миненко В. И.</i> Обеспечение отказоустойчивости бортовых инфокоммуникационных сетей.....	45
<i>Паламарчук М. В.</i> Обзор определений джиттера при передаче аудиоинформации и методов борьбы с ним.....	49
<i>Пахарев С. М.</i> Динамические характеристики сетевых систем.....	53
<i>Подшивалов Г. А.</i> Исследование алгоритмов центрирования легких мотопланеров в воздушных восходящих потоках.....	57
<i>Рагинис Д. М.</i> Диагностирование дефектов в системах управления.....	61
<i>Синёв Н. И., Коробков И. Л.</i> Тестирование протокола СТП-ИСС на модели бортовой сети космического аппарата.....	65
<i>Соловяненко К. В., Авагян Р. Ю.</i> Методика расчета элементов систем аварийного спасения космонавтов.....	69
<i>Степков Н. В.</i> Сравнение ABC и XYZ методов анализа широкой номенклатуры запасов в логистике.....	73
<i>Степанов В. Е.</i> Состояние гонки при моделировании на SDL.....	76
<i>Фортышев Е. Ю.</i> Алгоритм поддержки сети spacewire при децентрализованном алгоритме Plug-and-Play.....	80
<i>Ягубов Д. О.</i> Применение искусственной нейронной сети в задаче управления легким мотопланером.....	84
РАДИОЭЛЕКТРОНИКА, ЭЛЕКТРОНИКА И СВЯЗ.....	91
<i>Афанасьева К. Р.</i> Анализ направлений развития MID-технологий.....	91
<i>Булякулов Р. Р., Голубев С. С.</i> Портативное устройство для лечения и профилактики заболеваний опорно-двигательной системы человека.....	95
<i>Козырев Е. В.</i> Выходные фазокомпенсированные зеркала для CO <sub>2</sub> лазеров со ступенчатым профилем отражения.....	97
<i>Комарова А. М.</i> Исследование оптических констант пленок ZnS по спектрам пропускания в диапазоне 0,2–1,1 мкм.....	100
<i>Котлова М. А.</i> Исследование реализации связи в медицине катастроф.....	103

<i>Прокофьев П. А.</i> Анализ скорости передачи информации в системах с многочастотной модуляцией и многопозиционной частотной модуляцией.....	105
<i>Разумова Н. В.</i> Методы совмещения изображений в задачах дистанционного зондирования.....	108
<i>Семенцов М. А.</i> Определение оптических констант пленок фторида бария по спектрам отражения в диапазоне 200–1100 нм.....	110
<i>Соколов М. Н.</i> Сравнительный анализ дискретных преобразований в задачах распознавания изображений.....	113
<i>Суркин А. С.</i> Обзор и анализ методов обнаружения номерных знаков.....	115
<i>Хомутов Я. С.</i> Система территориального взаимодействия группы подвижных абонентов.....	119
<i>Хухунин С. А.</i> Защита персональных данных в телемедицине при использовании коротких волн.....	122
<i>Шатохин А. В.</i> Канал связи волоконно-оптического теплового пожарного извещателя.....	123
<b>Вычислительные системы и программирование.....</b>	<b>126</b>
<i>Алексеев В. А.</i> Основы программной реализации базовых методов регрессионного анализа.....	126
<i>Апанасенко Н. В.</i> Модель системы передачи данных с позиционно импульсной модуляцией при воздействии импульсной помехи.....	131
<i>Бабанов К. В.</i> Выделение объектов социальной сети по заданным информационным признакам.....	135
<i>Бабанов К. В.</i> Моделирование аксонального наведения.....	137
<i>Балян А. А.</i> Преобразование сжатого цветного изображения в сжатое изображение с градацией серого в микроконтроллерных системах.....	140
<i>Бахмут А. Д.</i> Программный комплекс вторичной обработки и анализа информации при оценке состояний сложных технических объектов в составе автоматизированной системы поддержки принятия решений.....	144
<i>Беляевский А. О.</i> Алгоритм вычисления траектории движения волномерного буя.....	148
<i>Богданов В. В.</i> Совместное применение статического и эвристического анализаторов безопасности кода.....	150
<i>Виноградов В. В.</i> Длинная арифметика.....	156
<i>Виноградов В. В.</i> Оптимизация вычислительных процессов при трехмерном моделировании.....	158
<i>Виноградов В. В.</i> Проект стратегической игры. Интеллектуализация Геймплея.....	160
<i>Гнутова Ю. Д.</i> Проектирование автоматизированной информационной системы создания рабочих программ дисциплин на кафедре.....	163
<i>Головин Е. Н.</i> Протокол информационного взаимодействия прикладного уровня и его применение в АСУ космическими объектами.....	167
<i>Гончаров М. Е.</i> Разработка интерактивного электронного руководства по сборке квадрокоптера.....	172
<i>Долгов Н. С.</i> Программный комплекс диагностирования элементов оборудования системы безопасности и контроля.....	178
<i>Дубков И. А., Лозач Г. К.</i> Разработка трехмерной модели персонажа для игрового движка.....	184
<i>Жукова С. А.</i> Анализ экспериментальных результатов по исследованиям океанологических процессов.....	190
<i>Зянчурин А. Э.</i> Разработка пользовательского интерактивного интерфейса 3D-мнемосхем в составе АСУ подготовки и пуска ракет космического назначения.....	196
<i>Иванов И. С.</i> Анализ алгоритмов подавления боковых лепестков фазоманипулированных последовательностей.....	199
<i>Ильинская Д. М.</i> Оценка эффективности нормализации таблиц в СУБД MySQL.....	203
<i>Капитонов В. С.</i> Применение распределения мел-частотных кепстральных коэффициентов для голосовой идентификации личности диктора.....	206
<i>Кривец Н. В.</i> Система поддержки принятия решений в лабораторной диагностике энтеробактерий.....	212
<i>Кузнецова Ю. А.</i> Тренажер для операторов по вводу информации.....	216
<i>Лесникова О. С.</i> Неявные блокировки транзакций в СУБД MySQL.....	219

<i>Логвинчук Е. А.</i> Проект переноса базы данных конфигурации колл-центра в систему управления базами данных кассандра .....	223
<i>Максимова О. Н.</i> Использование вариации теста тьюринга для выявления «машинных» паттернов человеческого поведения.....	226
<i>Матусова И. Е., Суменсан М. А.</i> 3D моделирование в исследовании зооморфных образов скифо-сибирского звериного стиля.....	228
<i>Меликян А. Ю., Исаев А. Н.</i> Представление элементов управления в интерактивных приложениях на примере реконструкции образов скифо-сибирского звериного стиля .....	233
<i>Мулеронко Р. А.</i> Особенности создания трехмерного приложения в Blender.....	237
<i>Нагорный Т. В.</i> Генерация текстуры 3D мнемосхем в АСУ космическими средствами на основе малой картинки для открытых ландшафтов .....	241
<i>Нестерук И. В.</i> Модель акторов.....	244
<i>Никифорова А. А.</i> Веб-ориентированные медиатехнологии в музыкальном образовании .....	247
<i>Овчаров С. Ю.</i> Методика компьютерного прогнозирования спроса на основе первичных данных .....	251
<i>Охтилев П. А.</i> Программный комплекс первичной обработки информации при оценивании состояний сложных технических объектов в составе автоматизированной системы поддержки принятия решений .....	255
<i>Печенин Д. В., Дамизов Р. И.</i> Проектирование программного комплекса для обучения работе с микроконтроллерами с использованием case-технологий.....	259
<i>Полех О. В.</i> Сравнительный анализ систем электронного документооборота.....	262
<i>Прокуратов Р. С.</i> Моделирование и оценка эффективности размещения сил и средств при нанесении противником группового ракетного удара .....	266
<i>Рожко Д. С.</i> Использование гибридных вычислительных систем в помехоустойчивом кодировании. Код Рида-Соломона .....	271
<i>Смолиенко Д. А.</i> Трансляция визуальных языков.....	275
<i>Сытин А. Р.</i> Классификация аэрокосмических изображений с использованием радиальных базисных нейронных сетей.....	279
<i>Топчий А. С.</i> Применение генетических алгоритмов для разработки адаптивных противников .....	282
<i>Устинов А. В.</i> Подсистема взаимодействия с пользователем в составе программного комплекса обработки анализа информации в АСУ сложными техническими объектами.....	284
<i>Фролова А. Ю.</i> Предобработка изображения для распознавателя текста.....	289
<i>Шалумов В. Г.</i> Программный комплекс диагностирования элементов оборудования АСУ СТО .....	293
<i>Шеин Р. Ю.</i> Обработка изображений с использованием стохастической нейронной сети .....	299
<i>Шелест М. Н.</i> Моделирование сетей массового обслуживания с резервированием ресурсов .....	302
<i>Шинкаренко Е. А.</i> Автоматизированная система удаленной проверки лабораторных работ .....	306
<b>ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ЗАЩИТА ИНФОРМАЦИИ.....</b>	<b>309</b>
<i>Бобров В. С.</i> Разработка и исследование алгоритма вычисления карты глубины по стереоизображению .....	309
<i>Васильев М. В., Пруданов А. С., Горбунов А. В.</i> Исследование эффективности алгоритма встраивания цифровых водяных знаков с помощью кода хэмминга в изображения формата BMP .....	313
<i>Иванов Д. Н.</i> Исследование алгоритмов автоматического раскрашивания изображений в градациях серого .....	317
<i>Ильин В. И.</i> Исследование инструментов эмуляции нагрузочного тестирования Веб-приложений.....	320
<i>Устюжанин Н. В.</i> Анализ алгоритмов моделирования переднего плана и фона в задаче детектирования движения .....	324
<i>Шевелев С. С.</i> Программно-аппаратный комплекс для мобильной аутентификации на Веб-сервисах.....	329



Научное издание

ШЕСТЬДЕСЯТ ВОСЬМАЯ  
МЕЖДУНАРОДНАЯ СТУДЕНЧЕСКАЯ  
НАУЧНАЯ КОНФЕРЕНЦИЯ ГУАП

Часть I  
Технические науки

Сборник докладов

Ответственные за выпуск:  
*А. В. Бобович, В. Н. Соколова*

Материалы публикуются в авторской редакции  
Компьютерная верстка *И. Н. Мороз*

---

Подписано к печати 06.11.15. Формат 60x84 1/16. Бумага офсетная.  
Печать офсетная. Усл. печ. л. 19,48. Уч.-изд. л. 20,94. Тираж 150 экз.  
Заказ № 425

---

Редакционно-издательский центр ГУАП  
190000, Санкт-Петербург, ул. Б. Морская, д. 67