

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ  
Издательство "Новые технологии"

## СОДЕРЖАНИЕ

### ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

- Куделя В. Н. Методы перебора на графах и задачи управления телекоммуникационной сетью . . . . . 2  
Жуков О. Д. Система логарифмических вычислений на основе модулярной алгебры . . . . . 7

### ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Мандрикова О. В., Богданов В. В., Полозов Ю. А. Моделирование сигналов со сложной структурой на основе конструкции вейвлет-преобразования . . . . . 12  
Алгулиев Р. М., Оруджов Г. Г., Сабзиев Э. Н., Панахов Н. А., Расулова Н. В., Алиева А. А. Разработка алгоритмов обнаружения и корректировки некоторых характерных ошибок векторной карты рельефа местности . . . . . 19  
Есаков Д. И. Объединение нескольких методов при автоматизированной обработке изображений земной поверхности . . . . . 24

### ПРОГРАММНАЯ ИНЖЕНЕРИЯ

- Петров А. В., Гузев В. Б. МС# — универсальный язык параллельного программирования . . . . . 29  
Синицкий М. Б., Синицкий Б. М. Инновационный подход к разработке кросс-платформенных приложений . . . . . 32

### МОДЕЛИРОВАНИЕ И СТРУКТУРНЫЙ СИНТЕЗ

- Кухаренко Б. Г. Технология спектрального анализа на основе быстрого преобразования Фурье . . . . . 38  
Виленик Л. С., Иванов Ю. В., Трофимов В. П. О математическом моделировании динамики естественных природных и физических систем дискретно-непрерывными системами с запаздывающим аргументом . . . . . 43

### ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ СИСТЕМЫ

- Куликов Г. Э. Автоматизированная система сбора и обработки данных для установок физического профиля . . . . . 50  
Чернышов А. В. Автоматизированная генерация программ контроля бортовой телеметрической аппаратуры . . . . . 56

### ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В СОЦИАЛЬНЫХ И ЭКОЛОГИЧЕСКИХ СИСТЕМАХ

- Соколов Э. М., Панарин В. М., Зуйкова А. А., Бизикин А. В. Автоматизированная система экологического мониторинга атмосферы при выбросах вредных веществ . . . . . 58  
Герман А. С. Процессы обеспечения информационно-телекоммуникационной среды передачи данных и видеоконференцсвязи в органах государственной службы . . . . . 62

### ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

- Пантелеев Е. Р., Суворов В. А. Оперативный анализ действий студента в среде Web-обучения . . . . . 67  
Карданова Е. Ю., Карпинский В. Б. Специальные методы анализа результатов тестирования, основанные на свойстве объективности моделей Раша . . . . . 72  
Углев В. А., Устинов В. А., Добронец Б. С. Системный подход к процессу обучающего компьютерного тестирования . . . . . 81  
Contents . . . . . 88

- Приложение. Волосатова Т. М., Воронов А. В., Чичварин Н. В. Математическое и программное обеспечение решения задач анализа и синтеза для САПР оптоэлектронных систем

Главный редактор  
НОРЕНКОВ И. П.

Зам. гл. редактора  
ФИЛИМОНОВ Н. Б.

Редакционная  
коллегия:

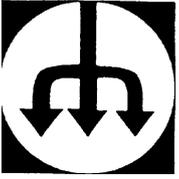
- АВДОШИН С. М.  
АНТОНОВ Б. И.  
БАТИЩЕВ Д. И.  
БАРСКИЙ А. Б.  
БОЖКО А. Н.  
ВАСЕНИН В. А.  
ГАЛУШКИН А. И.  
ГЛОРИОЗОВ Е. Л.  
ГОРБАТОВ В. А.  
ДОМРАЧЕВ В. Г.  
ЗАЛЕЩАНСКИЙ Б. Д.  
ЗАРУБИН В. С.  
ИВАННИКОВ А. Д.  
ИСАЕНКО Р. О.  
КОЛИН К. К.  
КУЛАГИН В. П.  
КУРЕЙЧИК В. М.  
ЛЬВОВИЧ Я. Е.  
МАЛЬЦЕВ П. П.  
МЕДВЕДЕВ Н. В.  
МИХАЙЛОВ Б. М.  
МУХТАРУЛИН В. С.  
НАРИНЬЯНИ А. С.  
НЕЧАЕВ В. В.  
ПАВЛОВ В. В.  
ПУЗАНКОВ Д. В.  
РЯБОВ Г. Г.  
СТЕМПКОВСКИЙ А. Л.  
УСКОВ В. Л.  
ЧЕРМОШЕНЦЕВ С. Ф.  
ШИЛОВ В. В.

Редакция:

- БЕЗМЕНОВА М. Ю.  
ГРИГОРИН-РЯБОВА Е. В.  
ЛЫСЕНКО А. В.  
ЧУГУНОВА А. В.

Аннотации статей размещены на сайте журнала по адресу <http://www.informika.ru/text/magaz/it/> или <http://novtex.ru/IT>.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора наук.



УДК 621.385.6.01:004.7

В. Н. Куделя, д-р техн. наук,  
ОАО "Институт сетевых технологий"

## Методы перебора на графах и задачи управления телекоммуникационной сетью

*Получены верхние оценки вычислительной сложности задач класса управления телекоммуникационной сетью. На основе этих результатов разработан метод трансформированного перебора элементарных путей в графе. Метод позволяет существенно сократить вычислительную сложность задач анализа и оптимизации при управлении телекоммуникационной сетью.*

### Введение

Проблема управления телекоммуникационными сетями (ТС) постоянно находится в центре внимания специалистов и заключается в решении задач выбора вариантов структур вычислительных, телекоммуникационных ресурсов и распределения потоков в них посредством решения классов задач анализа и оптимизации. Под классом задач автор понимает совокупность задач анализа и оптимизации, имеющих аналогичное задание условий, и возможность решения модификациями какого-либо общего метода.

Существующие методы решения задач класса управления ТС не всегда соответствуют требованиям к математическому обеспечению управления<sup>1</sup> по времени и/или точности их решения [1]. В математическом плане ко многим задачам управления ТС, как задачам многовариантного выбора, сводятся NP-полные задачи, не имеющие полиномиальных методов решения.

Исходя из этого, автор выделил класс задач управления структурой и потоками в ТС, к которым сводятся задачи перечисления на графах и решение которых возможно с помощью методов перебора последовательностей вершин (путей, увеличивающих цепей, маршрутов, остовов, вершинных баз и т. д.), и разработал метод оптимизации перебора на графах.

<sup>1</sup> Математическое обеспечение управления — совокупность математических методов, моделей и алгоритмов, разрабатываемых и применяемых для реализации целей и задач управления.

### 1. Общий подход к проблеме математического обеспечения управления телекоммуникационными сетями

Для снижения трудоемкости общая задача управления сетью декомпозируется на ряд частных задач управления ТС, а именно: анализ эффективности сети, управление структурой сети, управление потоками в сети, управление доступом к ресурсам сети. Отмеченные задачи являются, по-видимому, главными, вопрос "функциональной полноты" указанного набора представляет самостоятельный интерес и в данной работе не рассматривается.

Как показывает опыт различных исследователей, декомпозиция общей задачи управления ТС на классы частных задач анализа и оптимизации не уменьшила сложности проблемы, так как время, необходимое для построения оптимального метода решения каждой частной задачи, существенно превосходит время, в течение которого возникают новые, уточненные описания задач в рассматриваемом классе. Примером такого положения может служить класс задач управления структурой и распределением потоков в ТС.

Известно, что процедура маршрутизации в сети ARPA (1969 г.) в вычислительном плане базировалась на выборе кратчайшего пути в графе. Такая процедура приводит к распределению потоков, минимизирующему задержку пакетов между отправителем и адресатом, что дает неоптимальное распределение потоков как по временным характеристикам обмена, так и по пропускной способности сети в целом. В современных технологиях обмена данными, таких как ATM, IP и др., решение проблемы оптимального распределения потоков не нашло своего дальнейшего развития, и протоколы маршрутизации (Open Shortest Path First (OSPF), Intermediate System-to-Intermediate System (IS-IS) и т. д.) также основываются на выборе кратчайшего пути. Следствием такого подхода является и то, что задачи маршрутизации многоадресных и циркулярных сообщений не имеют удовлетворительного решения. Одним из примеров многоадресной передачи является организация услуг видеоконференцсвязи. Однако такие сервисы, использующие интересетевую многоадресную передачу, пока широко не распространены (распространенные используют одноадресную передачу). Как сами прикладные протоколы, использующие многоадресную передачу через

сеть, так и протоколы маршрутизации таких сообщений в сети (PIM-SM (RFC4601), PIM-DM (RFC3973), DVMRP (RFC1075), CBT (RFC2189), OSPF-multicast extensions и др.) находятся в стадии разработки IETF (*Internet Engineering Task Force*).

Такое же состояние проблемы с управлением структурой сети. Управление осуществляется только в автоматизированном режиме, так как точных алгоритмов анализа и оптимизации структуры сети, решающих задачу в реальном масштабе времени, не разработано.

В указанных случаях разработчики ТС, сталкиваясь с целой совокупностью задач управления, как обычно, разрабатывают (выбирают) методы решения для каждой частной задачи в отдельности. Использование большого числа методов в системе приводит к серьезному усложнению математического, программного и информационно-обеспечений. А это естественным образом сказывается на эффективности телекоммуникационной системы в целом.

Очевидно, что постановка проблемы совсем изменится, если задаться целью разработать методологическую основу решения класса задач.

Для реализации выбранной стратегии управления реальная ТС представляется в памяти ЭВМ моделью в виде сети<sup>2</sup>, состоящей из трех компонентов: узловой основы (вершин), сети каналов (дуг, ребер) и технологии действия. Аппарат теории графов, по-видимому, является единственной математической моделью, объединяющей структурное и потоковое содержания задач управления сложными информационно-техническими системами. Исходя из этого и учитывая состояние проблемы математического обеспечения управления ТС, автор пришел к необходимости постановки теоретико-методологической проблемы оптимизации существующих методов перебора путей (маршрутов) в графе, которая позволит на основе модификаций метода решать широкий круг задач анализа и оптимизации на сетях.

Необходимо оговорить, что под оптимизацией метода решения задачи автор понимает доработку метода, которая позволит сократить вычислительную сложность задачи. При этом вычислительная сложность задачи на графах рассматривается как функция от размера условия задачи  $n$  — числа вершин графа, и понимается как число  $O(n)$  элементов множества результатов ее решения. При этом  $O(n)$  состоит из сложности решения задачи  $O_p(n)$  — размер решения, и сложности ответа задачи  $O_o(n)$  — размер ответа [2].

<sup>2</sup> Сеть — граф, у которого некоторые вершины отмечены.

## 2. Исследование вычислительной сложности класса задач управления телекоммуникационными сетями

### 2.1. Исследование вычислительной сложности задач анализа телекоммуникационной сети

Исследование вычислительной сложности задач анализа ТС в рамках цели исследования можно ограничить одной из наиболее трудно-решаемых задач — задачей расчета надежности сети.

**Постановка задачи расчета показателей надежности ТС.** Наибольшие трудности при расчете надежностных характеристик сети сопряжены с учетом способа взаимного соединения ее узлов (структуры ТС), поэтому постановку задачи осуществим для оценки именно надежности сети в целом.

Итак, задана структура некоторой сети, надежность каждого элемента сети известна. Необходимо определить вероятность связности относительно выделенной пары узлов  $s$  и  $t$ . Эта задача известна как *NP*-полная задача о надежной сети.

Одним из методов расчета надежности сетей является метод расчета надежности по совокупности путей. Задача решается перебором всех элементарных путей (путь, у которого отсутствуют циклы) в сети между узлами  $s$  и  $t$ . Искомая надежность определяется надежностью каждого элементарного пути в зависимости от вариантов их пересечений по общим элементам.

**Постановка задачи перечисления элементарных путей в графе.** Известно множество элементарных путей между выделенными вершинами  $s$  и  $t$  в графе  $G[N, A]$ . Требуется перечислить, например записать в файл, весь ответ задачи.

Оценим строго вычислительную сложность записи в файл ответа  $O_o(n)$  задачи перечисления элементарных путей.

Пусть дан полносвязный граф  $G[N, A]$  и  $X(N) = \{X_m\}$  — семейство подмножеств элементарных путей между двумя вершинами этого графа,  $N \rightarrow X(N)$ ,  $X(N) = \bigwedge_{m=1}^{O_o(n)} X_m$  — конъюнк-

тивное множество  $(p + 1)$  — элементарных перестановок ( $s$  и  $t$  зафиксированы) из  $n$  элементов,  $p = 1, \dots, n - 1$ . Определить вычислительную сложность ответа  $O_o(n)$  задачи перечисления элементарных путей (число элементарных путей) между двумя вершинами графа. Здесь и далее  $N$  — множество вершин графа;  $A$  — множество его дуг (ребер), определяемых на множестве натуральных чисел,  $n = |N|$  — число вершин графа (мощность множества  $N$ );  $p$  — длина элементарного пути.

**Теорема 1.**<sup>3</sup> Пусть  $G[N, A]$  — полносвязный граф из  $n$  вершин. Тогда верхняя оценка вычислительной сложности *ответа* задачи перечисления элементарных путей между двумя вершинами графа  $s$  и  $t$  равна

$$O_O(n) = \begin{cases} \sum_{p=1}^{p=n-1} \frac{(n-2)!}{(n-p-1)!} & \text{для } 1 \leq p < n \\ 0 & \text{для } n \leq p \text{ и/или } n < 2. \end{cases}$$

В данной постановке задачи ответ уже известен. Но для получения ответа задачи перечисления путей в графе необходимо исследовать все множество промежуточных результатов<sup>4</sup> решения задачи, так как они либо позволят найти элементарный путь между  $s$  и  $t$ , либо укажут на его отсутствие. Оценим строго размер решения задачи перечисления элементарных путей  $O_P(n)$ .

**Теорема 2.** Пусть  $G[N, A]$  — граф из  $n$  вершин. Тогда верхняя оценка вычислительной сложности решения задачи перечисления элементарных путей между вершинами графа  $s$  и  $t$  равна

$$O_P(n) = \begin{cases} \sum_{p=1}^{p=n-2} \frac{(n-2)!}{(n-p-2)!} & \text{для } 1 \leq p < n \\ 0 & \text{для } n \leq p \text{ и/или } n < 2 \end{cases}$$

Из теорем 1 и 2 окончательно имеем.

**Следствие 1.** Если граф  $G[N, A]$  состоит из  $n$  вершин, то верхняя оценка вычислительной сложности задачи перечисления элементарных путей между вершинами графа  $s$  и  $t$  равна

$$O(n) = \begin{cases} \sum_{p=1}^{p=n-1} \frac{(n-p)(n-2)!}{(n-p-1)!} & \text{для } 1 \leq p < n \\ 0 & \text{для } n \leq p \text{ и/или } n < 2 \end{cases}$$

или

$$O(n+1) = \begin{cases} O(n)(n-1) + n & \text{для } 1 \leq p < n \\ 0 & \text{для } n \leq p \text{ и/или } n < 2. \end{cases}$$

## 2.2. Исследование вычислительной сложности задач управления структурой телекоммуникационной сети

**Постановка задачи управления структурами ТС.** Непредсказуемые в вероятностном смысле выходы из строя элементов ТС и/или превышение нагрузкой на сеть некоторого допустимого значения могут привести к деградации сети, а в худшем случае — к ее блокировке. В этих условиях требуется, прежде всего, перестройка структуры сети, а затем перераспределение как внешних, так и внутренних потоков для обеспечения заданных показателей живучести

<sup>3</sup> Ввиду ограничений на объем статьи здесь и далее доказательства теорем не приводятся, некоторые из представленных результатов опубликованы в работе [3].

<sup>4</sup> Промежуточный результат — это элементарный путь, ведущий из вершины  $s$  в вершину  $i$ , где  $i = 1, \dots, n, i \neq s, t$ .

и надежности сети, временных характеристик информационного обмена, пропускной способности (производительности) сети и параметров качества обслуживания.

Постановка задачи управления структурой сети может быть представлена следующим образом: по заданной матрице избыточности  $\mathbf{R} = r_{ij}$  ( $i, j = 1, \dots, n, r_{ij}$  — целое положительное число) на графе пораженной сети  $G_n[N_n, A_n]$  и графе резерва пораженной сети  $G_{р.п}[N_{р.п}, A_{р.п}]$  построить сеть  $G_{реш}[N, A]$  с минимальным числом ребер  $(ij) \in A$  таким образом, чтобы для каждой пары вершин  $s$  и  $t$ ,  $w_{st} \geq r_{st}$ , где  $w_{st}$  — число вершинно-независимых путей между  $s$  и  $t$ .

Задача в данной постановке известна как  $NP$ -полная задача построения сети с заданной связностью и минимальной стоимостью. Если  $w_{st}$  положить равным двум, то к задаче построения сети с заданной связностью и минимальной стоимостью полиномиально сводится задача распознавания свойств графа "гамильтонов путь"<sup>5</sup>.

**Постановка задачи распознавания свойств графа "гамильтонов путь".** Пусть дан граф  $G[N, A]$ . Вопрос: Верно ли, что  $G[N, A]$  содержит гамильтонов путь из  $s$  в  $t$ . Оценим строго вычислительную сложность  $O(n)$  задачи распознавания свойств графа "гамильтонов путь".

**Следствие 2.** Если граф  $G[N, A]$  состоит из  $n$  вершин, а  $p = (n-1)$  — длина гамильтонова пути, то верхняя оценка вычислительной сложности задачи распознавания свойств графа "гамильтонов путь" между двумя вершинами графа  $G[N, A]$  равна

$$O(n) = \begin{cases} \left( \sum_{p=1}^{p=n-2} \frac{(n-2)!}{(n-p-2)!} \right) + (n-2)! & \text{для } 1 \leq p < n \\ 0 & \text{для } n \leq p \text{ и/или } n < 2, \end{cases}$$

где первое слагаемое — размер решения задачи, а второе — размер ответа для  $p = (n-1)$ .

Таким образом, исследование класса задач управления ТС показало, что общим методом их решения может служить метод перебора путей в графе. Однако существующие методы перебора, несмотря на гарантированно точное решение задачи, вследствие вычислительной сложности решения не могут быть использованы в качестве математического обеспечения управления структурой и потоками в ТС.

## 3. Метод трансформированного перебора путей в графе

Осуществим постановку задачи перечисления элементарных путей в графе следующим образом.

<sup>5</sup> Задача распознавания свойств графа "гамильтонов путь" также полиномиально сводится к задачам маршрутизации многоадресных и циркулярных сообщений.

**Дано:** граф  $G[N, A]$ , начальная  $s$  и конечная  $t$  вершины графа.

**Требуется:** сформировать семейство сочетаний  $C(N) = \{C_p\}$  элементарных путей длины  $p$ , у которых начальная и конечная вершины совпадают, а набор вершин между ними одинаков, как отображения семейства элементарных путей  $X(N) = \{X_m\}$ .

То есть

$$f: X(N) \rightarrow C(N),$$

$$\forall X_m \in X(N) \xrightarrow{(K(p)-1)} C_{pi} = f\left(\bigwedge_{j=1}^{K(p)} X_j\right) =$$

$$= \begin{cases} \bigcup_{j=1}^{K(p)} X_j & \text{— для решения задач анализа,} \\ \bigvee_{j=1}^{K(p)} X_j & \text{— для решения задач оптимизации} \end{cases}$$

при ограничениях:

$$O^{Tp}(n) \ll O(n); \{j\} \subseteq \{m\}, \text{ а } \left\{ \bigwedge_{j=1}^{K(p)} X_j \right\} \subseteq X(N);$$

$$C_{pi} \neq \emptyset, \text{ если } \left\{ \bigwedge_{j=1}^{K(p)} X_j \right\} \neq \emptyset; X_j \subseteq \left\{ \bigwedge_{j=1}^{K(p)} X_j \right\},$$

где  $C_p = \{C_{pi}\}$ ,  $C_{pi}$  — сочетание элементарных путей длины  $p$ , у которых начальная и конечная вершины совпадают, а набор вершин между ними одинаков;  $\cup$  — операция объединения множеств;

$\wedge, \vee$  — логическая операция "и", логическая операция "или";  $O^{Tp}(p)$  — число  $C_{pi} \subseteq C_p$ ;  $K(p)$  — число

$X_j \subseteq \left\{ \bigwedge_{j=1}^{K(p)} X_j \right\}$ ;  $\rightarrow$  — отображение (в нашем

случае разбиение) множества на другое множество;  $\xrightarrow{(K(p)-1)}$  — поставлено в  $(K(p) - 1)$ -соответствие;  $f$  — образ элемента множества,  $f^{-1}$  — прообраз элемента множества;  $\sim$  — отношение эквивалентности.

Метод трансформированного перебора путей в графе представлен блок-схемой алгоритма на рис. 1 (см. третью сторону обложки).

В блок-схеме алгоритма для наглядности процесса перебора исполнительные блоки поставлены последовательно и позволяют получить ответ задачи в режиме ручного счета. Однако в машинных алгоритмах процесс трансформированного перебора выполняется в едином цикле.

Блок № 1 по параметрам  $n$  и  $p$  графа  $G[N, A]$  решает основную задачу — формирует семейство множеств  $X(N) = \{X_m\}$ .

Существует большое число методов перебора элементарных путей в графе: поиск с возвраще-

нием; метод решета; метод перемножения матриц; метод транспонирования матриц<sup>6</sup>.

Блоки № 2, 3, 5—7 — операторы цикла.

Блок № 4 по параметру  $p$  графа  $G[N, A]$  разбивает семейство множеств  $X(N)$  на классы

$Y_p = \bigwedge_{p=1}^{n-1} Y_p$ . Классы  $Y_p$  состоят из подмножеств  $Y_{pr}$ , находящихся во взаимно однозначном соответствии с элементарными путями  $X_r$  по числу и порядку элементов в нем.

Функция  $f_1$  устанавливает  $(1 - 1)$ -соответствие между элементами множеств  $Y(N)$  и  $X(N)$ . Функция  $f_2$  формирует конъюнктивные классы  $Y_p$  элементов  $Y_{pr} = X_r, X_r \subseteq \{X_m\}$  одинаковой длины  $p$ . Функция  $f_3$  формирует семейство классов  $Y(N) = \{Y_p\}$ . Причем  $\{X_m\} =$

$\{Y_p\} = \bigwedge_{p=1}^{n-1} \bigwedge_{r=1}^{O(p)} Y_{pr}$ , где  $O(p)$  — число

элементарных путей длины  $p$ .

Блок № 8: функция  $f_4(f_5)$  разбивает классы  $Y_p$  на сочетания  $C_{pi}$ , объединяя  $f_4$  (выбирая  $f_5$ ) подмножества  $Y_{pr}$ , у которых начальная ( $s$ ) и конечная ( $2, \dots, n$ ) вершины совпадают, а набор вершин между ними одинаков, устанавливая  $(K(p) - 1)$ -соответствие между элементами множеств  $Y(N)$  и  $C(N)$ ; функция  $f_6$  формирует конъюнктивные подмножества  $C_p$  элементов  $C_{pi}$ , а функция  $f_7$  — семейство множеств  $C(N) = \{C_p\}$ .

Таким образом, в процессе решения задачи перечисления элементарных путей происходит трансформация результатов решения задачи как промежуточных, так и ответа, в сочетания  $C_{pi}$ .

На каждой итерации перебора фиксируется число элементарных путей в каждом сочетании, и они объединяются по правилу объединения множеств, либо из них, по определенному критерию<sup>7</sup>, выбирается какой-то один.

Полная карта результатов решения задачи перечисления между двумя вершинами полносвязного графа, состоящего из пяти вершин ( $s = 1, t = 5$ ), методом трансформированного перебора представлена на рис. 2 (см. вторую сторону обложки).

В полученном множестве решений тождество по форме (числу элементов множества и связям вершин в некоторых элементах) отсутствует, т. е. нет взаимно однозначного соответствия между множеством результатов решения задачи перечисления элементарных путей  $X(N)$ , полученного

<sup>6</sup> Метод транспонирования матриц разработан для снятия экспоненциальной зависимости вычислительной сложности решения потоковых задач от числовых параметров ребер (дуг) графа [4]. Однако он эффективен и для решения задач перечисления на графе.

<sup>7</sup> Теорема оптимальности элементарного пути при трансформированном переборе будет рассмотрена далее.

существующими методами, и множеством результатов решения этой же задачи  $C(N)$ , полученного методом трансформированного перебора. Однако сохраняется взаимно однозначное соответствие по числу и набору вершин между элементами множеств результатов решения. Это означает, что метод трансформированного перебора элементарных путей для задач класса управления ТС позволяет получить изоморфные элементам  $X_m$  элементы  $C_{pi} \sim X_m$  гомоморфного множеству  $X(N)$  множества  $C(N)$ , мощность которого  $O^{Tp}(n)$  значительно меньше мощности  $O(n)$  множества  $X(N)$ . Разработанный метод разбивает множество  $X(N)$  на семейство множеств  $C(N)$ , такое, что каждый элемент множества  $X(N)$  поставлен в  $(K(p) - 1)$ -соответствие элементу множества  $C(N)$ , при этом  $f^{-1}(C_{pi}) \sim f(X_j)$ ,  $X_j \subseteq \{X_m\}$ ,  $C_{pi} \subseteq \{C_p\}$ . То есть метод позволяет трансформировать структуру множества решений задачи перечисления, не искажая смысла ее постановки и процесса перебора.

Оценим вычислительную сложность задач управления структурой и потоками ТС при решении их методом трансформированного перебора элементарных путей.

#### 4. Исследование вычислительной сложности класса задач управления телекоммуникационной сетью с применением метода трансформированного перебора

##### 4.1. Исследование вычислительной сложности задач анализа телекоммуникационной сети с применением метода трансформированного перебора

В результате применения к задаче представленной в подразд. 2.1 метода трансформированного перебора имеем следующее.

**Теорема 3.** Если граф  $G[N, A]$  полносвязный, то коэффициент трансформации (сокращения вычислительной сложности) на каждой итерации  $p$  метода трансформированного перебора элементарных путей равен

$$K(p) = (p - 1)!$$

**Теорема 4.** Пусть  $G[N, A]$  — полносвязный граф из  $n$  вершин. Тогда верхняя оценка вычислительной сложности *ответа* задачи перечисления между двумя вершинами графа  $s$  и  $t$  при трансформированном переборе элементарных путей равна

$$O_O^{Tp}(n) = 2^{(n-2)}, n \geq 2.$$

**Теорема 5.** Пусть  $G[N, A]$  — граф из  $n$  вершин. Тогда верхняя оценка вычислительной сложности *решения* задачи перечисления при трансформированном переборе элементарных путей между двумя вершинами графа  $s$  и  $t$  равна

$$O_P^{Tp}(n) = \frac{n-2}{2} 2^{(n-2)}, n \geq 2.$$

**Следствие 3.** Если граф  $G[N, A]$  состоит из  $n$  вершин, то верхняя оценка вычислительной сложности задачи перечисления при трансформированном переборе элементарных путей между вершинами графа  $s$  и  $t$  равна

$$O^{Tp}(n) = \frac{n}{2} 2^{(n-2)}, n \geq 2.$$

##### 4.2. Исследование вычислительной сложности задач управления структурами телекоммуникационной сети с применением метода трансформированного перебора

Постановка задач управления структурами и задачи распознавания свойств графа "гамильтонов путь" в данном подразделе аналогична постановкам в подразд. 2.2, а из теорем 4, 5 вытекает следующее следствие.

**Следствие 4.** Если граф  $G[N, A]$  состоит из  $n$  вершин, то верхняя оценка вычислительной сложности задачи распознавания свойств графа "гамильтонов путь" при трансформированном переборе элементарных путей между двумя вершинами графа равна

$$O^{Tp}(n) = \frac{n-2}{2} 2^{(n-2)} + 1, n \geq 2.$$

Таким образом, метод трансформированного перебора точно решает задачи перечисления на графах, а закон распределения результатов решения задачи представлен на рис. 3.

##### 4.3. Оптимизация пути в графе.

###### Теорема оптимальности пути при трансформированном переборе в графе

Существующие методы решения задач поиска оптимального пути в графе обычно используют принцип оптимальности пути, предложенный Кофманом, который подразумевает разбиение множества вершин (дуг) графа на уровни. В процессе же трансформированного перебора множе-

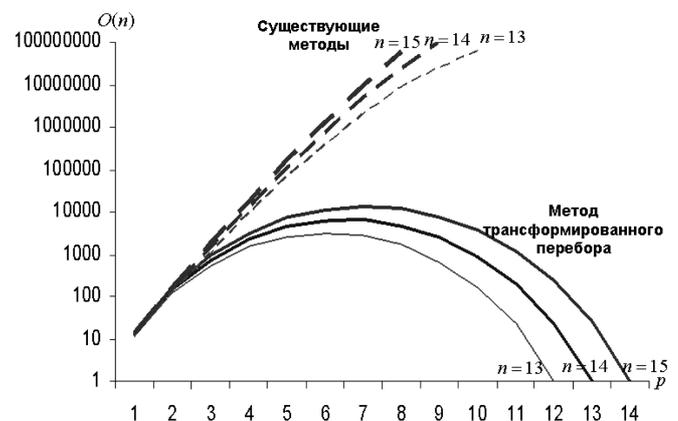


Рис. 3. Закон распределения числа результатов решения задачи по их длине

ство элементарных путей разбивается на сочетания и оказывается, что этот принцип не всегда работает при решении задач поиска оптимального пути предлагаемым методом.

В этой связи разработан принцип оптимальности пути для метода трансформированного перебора путей в графе.

**Теорема 6.** Пусть задан граф  $G[N, A]$  и максимальный (минимальный) элементарный путь  $(X_j)_{st}|_p = a \subseteq (C_{pi})_{st}$  через вершины между вершинами  $s$  и  $t$ , где  $s \neq t$ . Тогда его подпуть  $(X_j)_{sv}|_p = b \subseteq (X_j)_{st}|_p = a$ ,  $(X_j)_{sv}|_p = b \subseteq (C_{pi})_{sv}$  и  $(C_{pi})_{sv}|_p = b \subseteq (C_{pi})_{st}$  между вершинами  $s$  и  $v$ , где  $v \neq s, t$ ,  $b < a$  также максимален (минимален).

### Заключение

Разработанный метод — это метод, реализующий в некотором графе  $G[N, A]$  последовательность вершинно-множественных отображений  $f_k: N \rightarrow X(N) \rightarrow Y(N) \rightarrow C(N)$ , с помощью которых по начальной вершине  $s \in N$  вычисляется последовательность вершин  $C_{pi}$ .

Можно сказать, что метод трансформированного перебора — это своего рода логический шаблон, автоматически выделяющий и группирующий в соответствии с принципом выбора все результаты перебора на графе, объединяя (выбирая) элементарные пути по связям и числу элементов в них. Это позволяет, не снижая точности перебора, сократить число результатов решения за счет объединенного участия элементарных путей (сочетаний) в последующих итерациях перебора.

В научной литературе, посвященной проблематике вычислимости задач на ЭВМ, достаточно

часто говорится о труднорешаемых для ЭВМ задачах, имеющих экспоненциальную функцию вычислительной сложности. Однако каких-либо теоретических исследований по строгой количественной оценке труднорешаемости задач, связанных с перечислением элементарных путей, автор не встречал. Кроме того, доказательство труднорешаемости задач методом полиномиальной сводимости позволяет установить только сам факт труднорешаемости. В то же время, по мнению автора, более интересен для исследования все-таки процесс решения задачи перечисления методом перебора. В этой связи в работе, прежде всего, представлены некоторые результаты исследования труднорешаемости задач, сводимых к частным задачам управления ТС, а затем на основе этого анализа показан наиболее значимый результат, позволяющий существенно сократить (трансформировать) процесс перебора, связанный с перечислением путей (маршрутов) в графе.

### Список литературы

1. Куделя В. Н., Липатников В. А., Привалов А. А. Требования к качеству математического обеспечения управления автоматизированными системами военной связи // Вестник военного регистра. 2003. № 7 (31). М.: ЗАО Оолита, 2003.
2. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 416 с.
3. Куделя В. Н. Информационно-технические системы и методы перебора путей в графах // Труды международной конференции "Устойчивость и процессы управления". Т. 1: секции 1—5, 11: Санкт-Петербург 29 июня—1 июля 2005 года / Под ред. Д. А. Овсянникова, Л. А. Петросяна. СПб: Изд. СПбГУ, НИИ ВМ и ПУ, ООО ВВМ, 2005.
4. Куделя В. Н. Вопросы математического моделирования сетей связи ВМФ в условиях информационных войн // Труды 6-й Всероссийской НТК "Актуальные проблемы защиты и безопасности". СПб.: Изд. НПО Спецматериалы, 2003.

УДК 004.332.3

Жуков О. Д., канд. техн. наук,  
Московский государственный университет им. М. В. Ломоносова

## Система логарифмических вычислений на основе модулярной алгебры

*Рассматривается возможность использования нового метода логарифмического сложения, требующего для выполнения существенно меньшего объема таблиц по сравнению с известными методами. Метод основан на свойствах систем вычетов и конечных полей и колец. Наряду с предельно возможным сокращением объемов таблиц он также характеризуется меньшей вычислительной сложностью.*

### Введение

Известно, что наряду с широким использованием в компьютерных технологиях двоичной и двоично-кодированных систем счисления во многих теоретических исследованиях и некоторых практических применениях обосновывается

возможность внедрения других систем, например, на основе логарифмического представления числовых данных. Очевидно, в подобной системе существенно упрощаются такие массовые процедуры, как умножение, деление, возведение в степень и извлечение корня. Вместе с тем, сложение

и вычитание для данной системы являются не-тривиальными операциями и требуют большого объема таблиц [1]. В работе показана возможность существенного сокращения размерности этих таблиц на основе использования известных из теории чисел систем вычетов (СВ) [2].

В соответствии с Китайской теоремой о вычетах (КТ) СВ определяется набором взаимно простых модулей (целых чисел)  $m_i$ ;  $i = 1, 2, \dots, n$ . Любое целое  $A$  может быть уникально представлено в интервале  $[0, M)$  для  $M = \prod_{i=1}^n m_i$  совокупностью вычетов  $A = (a_1, a_2, \dots, a_n)$ , где  $a_i$  является наименьшим остатком от деления  $A$  на модуль  $m_i$  и определяется конгруэнцией  $a_i = A \bmod m_i \equiv |A|_{m_i}$ . В СВ вычислительные операции (кроме деления нецелых чисел) выполняются параллельно и независимо над всеми вычетами без межрядных переносов [2].

### Свойства конечных полей и колец

Из теории чисел известно, что алгебраическое поле формируется из множества элементов с возможностью выполнения над ними двух операций: сложения и умножения по определенным правилам [3, 4]. Конечные поля (поля Галуа) определяются двумя типами:

- простые поля  $GF(p)$  с  $p$  целочисленными элементами;
- полиномиальные поля  $GF(p^m)$  с  $p^m$  алгебраическими полиномами.

Здесь  $p$  — любое простое число,  $m$  — целое число.

В  $GF(p)$  все целые в интервале  $[0, p - 1]$  изменяются циклически по модулю  $p$  в соответствии с правилами модулярного сложения, а также с правилами модулярного умножения ненулевых операндов. Кроме того, каждое целое  $x$  в мультипликативной группе  $G(p)$  может быть сгенерировано как соответствующая степень определенного примитивного корня  $g$ .  $\forall x \in G(p)$ ,  $\exists g$  и  $i$  суть такие, что  $|g^{\Phi(p)}|_p \equiv 1$  и  $|g^i|_p \equiv x$ , где  $\Phi(p) = p - 1$ . Функция  $\Phi(p)$  равна числу взаимно простых с  $p$  неотрицательными целыми. Эта функция может быть вычислена в соответствии с выражением:

$$\Phi(p) = p \prod_{r|p} (1 - (1/r)),$$

где символ  $\prod_{r|p}$  означает произведение, охватывающее все простые  $r$ , которые делят  $p$ .

В конечных полях  $GF(p^m)$  сами элементы суть алгебраические полиномы. Когда эти элементы трактуются как целые, они не образуют поле, а формируют кольцо, называемое кольцом целых чисел по модулю  $p^m$  и обозначаемое  $Z/(p^m)$ . Эти конечные кольца по модулю  $p^m$  принадлежат двум категориям, а именно:  $Z/(2^m)$  и  $Z/(p^m)$  для четных и нечетных значений  $p$ , а элементы в этих кольцах суть целые числа.

Любое целое  $x \in [1, 2^m - 1]$  может быть уникально представлено в  $Z/(2^m)$  как трехиндексный код  $\langle \alpha, \beta, \gamma \rangle$ , если использовать соотношение  $x = 2^\alpha |5^\beta (-1)^\gamma| 2^m$ , где  $\alpha \in (0, 1, \dots, m - 1)$ ,  $\beta \in (0, 1, \dots, 2^{m-2} - 1)$  и  $\gamma \in (0, 1)$  [5, 6]. Отметим, что такие коды, сформированные для четных целых, не являются уникальными. Подобным образом в кольце  $Z/(p^m)$  для нечетных простых  $p$  может быть показано, что  $\exists g \in Z/(p^m)$ , удовлетворяющее  $|g^{\Phi(p^m)}|_{p^m} \equiv 1$ , где  $g$  есть примитивный корень кольца. Поэтому все ненулевые элементы в  $Z/(p^m)$  могут быть генерированы с использованием соотношения  $|g^\alpha p^\beta|_{p^m}$ , где  $\alpha \in (0, 1, \dots, \Phi(p^m) - 1)$  и  $\beta \in (0, 1, \dots, m - 1)$  [7].

### Определения и свойства мультилогарифмов

Для степенных функций справедливы следующие соотношения:  $x = a^y$  и  $y = \log_a x$ . Данное определение может быть применено непосредственно к вычислениям элементов в  $GF(p)$ , поскольку они выражаются как степени основания  $g$  (примитивного корня) по модулю  $p$ . Вместе с тем, элементы кольца  $Z/(p^m)$  как для четного, так и для нечетного  $p$ , могут быть выражены только как произведение степеней двух и трех элементов соответственно, взятое по модулю  $p^m$ . Таким образом, в обоих случаях для представления элемента в кольце  $Z/(p^m)$  недостаточно использовать единственный логарифм по одному основанию. Здесь мы приходим к необходимости идентификации элементов с помощью групп логарифмов по двум и более основаниям, которые определим как мультилогарифмы.

**Определение 1.** Логарифм с несколькими основаниями, или мультилогарифм (МЛ), от числа  $X$ , обозначаемый как  $\log m_{b_1, b_2, \dots, b_m} X$ , есть  $m$ -элементная экспонента  $(\alpha_1, \alpha_2, \dots, \alpha_m)$ , определяемая как мультистепень (МС), в которую необходимо возвести  $m$  оснований  $b_i$  ( $i = 1, 2, \dots, m$ ) соответственно, чтобы получить  $X = b_1^{\alpha_1} b_2^{\alpha_2} \dots b_m^{\alpha_m}$  ( $m > 1$ ), т. е.

$$(\alpha_1, \alpha_2, \dots, \alpha_m) = \log m_{b_1, b_2, \dots, b_m} X. \quad (1)$$

**Свойства мультилогарифмов.** Все алгебраические свойства обычных логарифмов применимы к мультилогарифмам. Если некоторые два числа  $X$  и  $Y$  удовлетворяют равенству (1), то имеют место следующие соотношения:

$$\begin{aligned} & \log m_{b_1, b_2, \dots, b_m} (XY) = \\ & = \log m_{b_1, b_2, \dots, b_m} X + \log m_{b_1, b_2, \dots, b_m} Y; \\ & \log m_{b_1, b_2, \dots, b_m} (X/Y) = \\ & = \log m_{b_1, b_2, \dots, b_m} X - \log m_{b_1, b_2, \dots, b_m} Y; \end{aligned}$$

$$\log m_{b_1, b_2, \dots, b_m}(X^e) = e \log m_{b_1, b_2, \dots, b_m} X;$$

$$\log m_{b_1, b_2, \dots, b_m}(\sqrt[r]{X}) = (1/r) \log m_{b_1, b_2, \dots, b_m} X.$$

Определение мультилогарифма может быть легко адаптировано для определения дуальных логарифмов с двумя основаниями в случае  $Z/(p^m)$  и логарифмов с тремя основаниями в случае  $Z/(2^m)$ .

**Определение 2.** Для любого ненулевого целого  $X \in Z/(p^m)$ , такого что  $X \equiv |g^\alpha p^\beta|_{p^m}$ , дуальный логарифм от  $X$ , соответствующий основаниям  $(g, p)$ , может быть выражен как  $(\alpha, \beta) = \log m_{g, p} X$ .

**Определение 3.** Для любого ненулевого целого  $X \in Z/(2^m)$ , такого что  $X = 2^\alpha 5^\beta (-1)^\gamma |_{2^m}$ , логарифм от  $X$  с тремя основаниями  $(2, 5, -1)$  может быть выражен как  $(\alpha, \beta, \gamma) = \log m_{2, 5, -1} X$ .

### Логарифмические вычисления на основа СВ

Как было показано выше, логарифмическое умножение двух целых чисел  $Z/(p^m)$  и  $Z/(2^m)$  в  $GF(p)$  выполняется сложением их индексов [6, 7]. Однако сложение и вычитание в логарифмической форме являются далеко нетривиальными и требуют больших объемов таблиц, т. е. табличной памяти. Прямой метод выполнения этих операций основан на использовании полных таблиц [1]. Объем такой таблицы, равный  $2^{2n}$ , оказывается слишком большим для типичных значений  $n$ , где  $n$  — число битов в операнде. В качестве альтернативы в логарифмических процессорах используется другой метод, требующий таблицу объемом не более  $2^n$  [1].

Пусть  $S = X \pm Y$  такое, что  $X = b^{e_x}$ ,  $Y = b^{e_y}$  и  $S = b^{e_s}$ . Принимая  $X$  как общий фактор,  $S$  может быть выражено как  $S = X(1 \pm Y/X)$ . Тогда получим

$$e_s = \log_b |X(1 \pm (Y/X))| = \log_b X + \log_b |1 \pm (Y/X)| = \log_b X + \log_b |1 \pm b^{e_y - e_x}| = e_x + e_u, \quad (2)$$

где  $e_u = \log_b |1 \pm b^{e_y - e_x}|$ .

Значение  $e_u$  должно быть предвычислено и записано в таблице. Хотя данный метод уменьшает размер таблицы, все-таки для операндов большой длины объем таблицы становится также слишком большим. Отсюда возникает необходимость использовать другие методы, требующие существенно меньшего объема таблиц. Такого результата можно добиться, используя свойства рассмотренных выше систем вычетов (СВ) и конечных полей и колец.

**Сложение в  $GF(p)$ .** Из теории чисел следует, что в конечном поле  $GF(p)$  сложение есть циклическая операция по модулю  $p$ . Покажем, что для

любых двух ненулевых целых  $X, Y \in GF(p)$ , определяемых конгруэнциями  $X \equiv |g^{\alpha_x}|_p$  и  $Y \equiv |g^{\alpha_y}|_p$  ( $p$  — простое), индекс  $\alpha$  их суммы по модулю  $p-1$  равен  $|\alpha_x + \alpha_u|_{p-1}$ , где  $\alpha_u = \log |1 + g^{|\alpha_y - \alpha_x|_{p-1}}|_p$ .

Действительно, на основе свойств логарифмов индексы  $X$  и  $Y$  могут быть записаны как  $\alpha_x = \log_g X$  и  $\alpha_y = \log_g Y$ . С учетом свойства циклической аддитивности  $GF(p)$  имеет место конгруэнция  $|X + Y|_p \equiv |g^\alpha|_p$  для некоторого значения  $\alpha \in GF(p)$ . Отсюда

$$\alpha = \log_g |X + Y|_p = \log_g |X(1 + (Y/X))|_p = \log_g X + \log_g |1 + g^{|\alpha_y - \alpha_x|_{p-1}}|_p = |\alpha_x + \alpha_u|_{p-1}, \quad (3)$$

где  $\alpha_u = \log_g |1 + g^{|\alpha_y - \alpha_x|_{p-1}}|_p$ .

Но в  $GF(p)$ , поскольку  $|g^{p-1}|_p \equiv 1$ , справедливо  $\alpha \equiv |\alpha_x + \alpha_u|_{p-1}$ .

**Сложение в конечных кольцах.** Таким же образом рассмотрим сложение в конечных кольцах. Из предыдущего раздела следует, что для них одноиндексное сложение является недостаточным. Здесь требуется поиндексное сложение векторов индексов, и может быть доказано следующее утверждение.

Если  $(\alpha_1, \alpha_2, \dots, \alpha_m) = \log m_{b_1, b_2, \dots, b_m} X$  и  $(\beta_1, \beta_2, \dots, \beta_m) = \log m_{b_1, b_2, \dots, b_m} Y$ , то

$$(\alpha_1, \alpha_2, \dots, \alpha_m) + (\beta_1, \beta_2, \dots, \beta_m) = (\alpha_1 + \beta_1, \alpha_2 + \beta_2, \dots, \alpha_m + \beta_m).$$

Действительно,

$$\begin{aligned} (\alpha_1, \alpha_2, \dots, \alpha_m) + (\beta_1, \beta_2, \dots, \beta_m) &= \log m_{b_1, b_2, \dots, b_m} X + \\ &+ \log m_{b_1, b_2, \dots, b_m} Y = \log m_{b_1, b_2, \dots, b_m} (XY) = \\ &= \log m_{b_1, b_2, \dots, b_m} (b_1^{\alpha_1 + \beta_1} b_2^{\alpha_2 + \beta_2} \dots b_m^{\alpha_m + \beta_m}) = \\ &= (\alpha_1 + \beta_1, \alpha_2 + \beta_2, \dots, \alpha_m + \beta_m). \end{aligned} \quad (5)$$

Сложение в конечных кольцах может быть выполнено на основе введенных выше определений и формулы (5).

**Сложение в  $Z/(p^m)$ .** Следующее утверждение имеет место для логического сложения в  $Z/(p^m)$ . Для любых целых  $X, Y \in Z/(p^m)$ , определяемых конгруэнциями  $X \equiv |g^{\alpha_x} p^{\beta_x}|_{p^m}$  и  $Y \equiv |g^{\alpha_y} p^{\beta_y}|_{p^m}$ , индексная пара  $(\alpha, \beta)$  их суммы характеризуется следующими соотношениями:

для  $\beta_y \geq \beta_x$

$$(\alpha, \beta) = (|\alpha_x + \alpha_u|_{\Phi(p^m)}, (\beta_x + \beta_u)), \quad (6)$$

для  $\beta_y < \beta_x$

$$(\alpha, \beta) = (|\alpha_y + \alpha_u|_{\Phi(p^m)}, (\beta_y + \beta_u)), \quad (7)$$

где

$$(\alpha_u, \beta_u) = \log m_{g,p} |1 + g^{(-1)w(\alpha_y - \alpha_x)}|_{\Phi(p^m)} \times \\ \times p^{(-1)w(\beta_y - \beta_x)}|_{p^m} \quad (8)$$

при

$$w = 0, \text{ если } \beta_y \geq \beta_x; \\ w = 1, \text{ если } \beta_y < \beta_x.$$

Действительно, с учетом свойства аддитивности конечного кольца  $Z/(p^m)$  справедливо, что  $|X + Y|_{p^m} \in Z/(p^m)$ . Поэтому

$$|X + Y|_{p^m} \equiv |g^\alpha p^\beta|_{p^m}.$$

Отсюда

$$(\alpha, \beta) = \log m_{g,p} |X + Y|_{p^m} = \\ = \log m_{g,p} |X(1 + (Y/X))|_{p^m} = \log m_{g,p} X + \\ + \log m_{g,p} |1 + (Y/X)|_{p^m}; \quad (9)$$

$$|Y/X|_{p^m} \equiv |g^{|\alpha_x - \alpha_y|_{\Phi(p^m)}} p^{(\beta_y - \beta_x)}|_{p^m}. \quad (10)$$

Но когда  $\beta_y < \beta_x$ , то  $p^{\beta_y - \beta_x} = p^{-\alpha}$ , где  $\alpha > 0$ . Тогда

$$|X/Y|_{p^m} \equiv |(g^{|\alpha_x - \alpha_y|_{\Phi(p^m)}} / p^\alpha)|_{p^m}.$$

Поскольку  $p^\alpha$  и  $p^m$  суть не взаимно простые,  $|X/Y|_{p^m}$  не может быть вычислено. Поэтому (9) можно переписать как

$$(\alpha, \beta) = \log m_{g,p} |X + Y|_{p^m} = \\ = \log m_{g,p} Y + \log m_{g,p} |1 + (X/Y)|_{p^m} = \\ = \log m_{g,p} Y + \log m_{g,p} |g^{|\alpha_x - \alpha_y|_{\Phi(p^m)}} p^{(\beta_y - \beta_x)}|_{p^m}. \quad (11)$$

Однако, когда  $\beta_y \geq \beta_x$ , (9) принимает вид

$$(\alpha, \beta) = \log m_{g,p} X + \\ + \log m_{g,p} |1 + g^{|\alpha_y - \alpha_x|_{\Phi(p^m)}} p^{\beta_y - \beta_x}|_{p^m}. \quad (12)$$

На основе (11) и (12) получим

$$(\alpha, \beta) = \log m_{g,p} X + \\ + \log m_{g,p} |1 + g^{|\alpha_y - \alpha_x|_{\Phi(p^m)}} p^{\beta_y - \beta_x}|_{p^m} \quad (13)$$

для  $\beta_y \geq \beta_x$  и

$$(\alpha, \beta) = \log m_{g,p} Y + \\ + \log m_{g,p} |1 + g^{|\alpha_x - \alpha_y|_{\Phi(p^m)}} p^{\beta_x - \beta_y}|_{p^m} \quad (14)$$

для  $\beta_y < \beta_x$ .

Отсюда следует

$$(\alpha, \beta) = (\alpha_x, \beta_x) + (\alpha_u, \beta_u)$$

для  $\beta_y \geq \beta_x$  и

$$(\alpha, \beta) = (\alpha_y, \beta_y) + (\alpha_u, \beta_u) \quad (15)$$

для  $\beta_y < \beta_x$ , где

$$(\alpha_u, \beta_u) = \log m_{g,p} |1 + \\ + g^{|\alpha_x - \alpha_y|_{\Phi(p^m)}} p^{(-1)w(\beta_x - \beta_y)}|_{p^m} \quad (16)$$

при

$$w = 0 \text{ для } \beta_y \geq \beta_x; \\ w = 1 \text{ для } \beta_y < \beta_x.$$

С учетом (5) выражения (15) преобразуются соответственно в

$$(\alpha, \beta) = ((\alpha_x + \alpha_u)_{\Phi(p^m)}, (\beta_x + \beta_u)) \text{ для } \beta_y \geq \beta_x,$$

$$(\alpha, \beta) = ((\alpha_y + \alpha_u)_{\Phi(p^m)}, (\beta_y + \beta_u)) \text{ для } \beta_y < \beta_x.$$

Отметим, что, если  $\beta \geq m$ , сумма равна нулю [7].

**Сложение в  $Z/(2^m)$ .** Докажем следующее утверждение. Для любых двух ненулевых целых  $X, Y \in Z/(2^m)$  при  $X = 2^{\alpha_x} |5^{\beta_x} (-1)^{\gamma_x}|_{2^m}$  и  $Y = 2^{\alpha_y} |5^{\beta_y} (-1)^{\gamma_y}|_{2^m}$  тройка индексов  $(\alpha, \beta, \gamma)$  их суммы определяется как

$$(\alpha, \beta, \gamma) = ((\alpha_x + \alpha_u), |\beta_x + \beta_u|_{2^{m-2}}, |\gamma_x + \gamma_u|_2)$$

для  $\alpha_y \geq \alpha_x$ ,

$$(\alpha, \beta, \gamma) = ((\alpha_y + \alpha_u), |\beta_y + \beta_u|_{2^{m-2}}, |\gamma_y + \gamma_u|_2)$$

для  $\alpha_y < \alpha_x$ ,

где

$$(\alpha_u, \beta_u, \gamma_u) = \log m_{2,5,-1} |1 + 2^{(-1)w(\alpha_y - \alpha_x)} \times \\ \times |5^{|\alpha_x - \alpha_y|_{\Phi(2^m)}} p^{(\beta_y - \beta_x)} (-1)^{(-1)w(\gamma_y - \gamma_x)}|_{2^m}|_{2^m} \quad (17)$$

при

$$w = 0 \text{ для } \alpha_y \geq \alpha_x;$$

$$w = 1 \text{ для } \alpha_y < \alpha_x.$$

Действительно, с учетом свойства циклической аддитивности  $Z/(2^m)$  получим  $|X + Y|_{2^m} \in Z/(2^m)$ . Поэтому для некоторых значений  $\alpha, \beta, \gamma$

$$|X + Y|_{2^m} = 2^\alpha |5^\beta (-1)^\gamma|_{2^m}.$$

Отсюда

$$(\alpha, \beta, \gamma) = \log m_{2,5,-1} |X(1 + (Y/X))|_{2^m} = \\ = \log m_{2,5,-1} X + \log m_{2,5,-1} |1 + (Y/X)|_{2^m}. \quad (18)$$

Вместе с тем

$$|Y/X|_{2^m} \equiv |2^{(\alpha_y - \alpha_x)} |5^{|\beta_y - \beta_x|_{2^{m-2}}} (-1)^{|\gamma_y - \gamma_x|_2}|_{2^m}|_{2^m}. \quad (19)$$

Но, когда  $\alpha_y < \alpha_x$ ,  $2^{\alpha_y - \alpha_x} = 2^{-e}$ , где  $e > 0$ . Тогда

$$|Y/X|_{2^m} \equiv (|5^{|\beta_y - \beta_x|} (-1)^{|\gamma_y - \gamma_x|} |_{2^{m-2}}) / 2^e |_{2^m}.$$

Поскольку  $2^e$  и  $2^m$  не взаимно простые,  $|Y/X|_{2^m}$  не может быть вычислено. Поэтому (18) перепишем в виде

$$\begin{aligned} (\alpha, \beta, \gamma) &= \log m_{2, 5, -1} |Y + X|_{2^m} = \\ &= \log m_{2, 5, -1} Y + \log m_{2, 5, -1} |1 + (X/Y)|_{2^m} = \\ &= \log m_{2, 5, -1} Y + \log m_{2, 5, -1} |1 + 2^{(\alpha_x - \alpha_y)}| \times \\ &\times 5^{|\beta_x - \beta_y|} (-1)^{|\gamma_x - \gamma_y|} |_{2^m} |_{2^m}. \end{aligned} \quad (20)$$

Однако, когда  $\alpha_y \geq \alpha_x$ , выражение (18) принимает вид

$$\begin{aligned} (\alpha, \beta, \gamma) &= \log m_{2, 5, -1} X + \log m_{2, 5, -1} |1 + \\ &+ 2^{(\alpha_y - \alpha_x)} |5^{|\beta_y - \beta_x|} (-1)^{|\gamma_y - \gamma_x|} |_{2^m} |_{2^m}. \end{aligned} \quad (21)$$

На основе (20) и (21) получим

$$\begin{aligned} (\alpha, \beta, \gamma) &= \log m_{2, 5, -1} X + \log m_{2, 5, -1} |1 + \\ &+ 2^{(\alpha_y - \alpha_x)} |5^{|\beta_y - \beta_x|} (-1)^{(\gamma_y - \gamma_x)} |_{2^m} |_{2^m} \end{aligned}$$

для  $\alpha_y \geq \alpha_x$  и

$$\begin{aligned} (\alpha, \beta, \gamma) &= \log m_{2, 5, -1} Y + \log m_{2, 5, -1} |1 + \\ &+ 2^{(\alpha_x - \alpha_y)} |5^{|\beta_x - \beta_y|} (-1)^{(\gamma_x - \gamma_y)} |_{2^m} |_{2^m} \end{aligned}$$

для  $\alpha_y < \alpha_x$ .

Тогда

$$(\alpha, \beta, \gamma) = (\alpha_x, \beta_x, \gamma_x) + (\alpha_u, \beta_u, \gamma_u) \quad (22)$$

для  $\alpha_y \geq \alpha_x$  и

$$(\alpha, \beta, \gamma) = (\alpha_y, \beta_y, \gamma_y) + (\alpha_u, \beta_u, \gamma_u) \quad (23)$$

для  $\alpha_y < \alpha_x$ ,

где

$$\begin{aligned} (\alpha_u, \beta_u, \gamma_u) &= \log m_{2, 5, -1} |1 + 2^{(-1)w(\alpha_y - \alpha_x)}| \times \\ &\times 5^{|(-1)w(\beta_y - \beta_x)|} (-1)^{|(-1)w(\gamma_y - \gamma_x)|} |_{2^m} |_{2^m} \end{aligned}$$

при

$$w = 0 \text{ для } \alpha_y \geq \alpha_x;$$

$$w = 1 \text{ для } \alpha_y < \alpha_x.$$

На основе (5) выражения (22) и (23) принимают следующий вид:

$$(\alpha, \beta, \gamma) = ((\alpha_x + \alpha_u), |\beta_x + \beta_u|_{2^{m-2}}, |\gamma_x + \gamma_u|_2)$$

для  $\alpha_y \geq \alpha_x$  и

$$(\alpha, \beta, \gamma) = ((\alpha_y + \alpha_u), |\beta_y + \beta_u|_{2^{m-2}}, |\gamma_y + \gamma_u|_2)$$

для  $\alpha_y < \alpha_x$ .  
Если  $\alpha = m - 1$ , тогда  $\beta$  и  $\gamma$  равны нулю, а когда  $\alpha > m - 1$ , сумма становится равной нулю [7].

### Требования к объемам таблиц

До оценки требуемых объемов таблиц приведем конкретные примеры, которые показывают процедуру выполнения логарифмического сложения в трех различных случаях для  $GF(p)$ ,  $Z/(p^m)$ ,  $Z/(2^m)$ .

**Пример для  $GF(p)$ .** Модуль равен 31 с примитивным корнем  $g = 3$ . Индексное кодирование элементов поля в  $GF(p)$  приведены ниже:

x...1	2	3	4	5	6	7	8	9	10	11	12	13	14			
α...0	24	1	18	20	25	28	12	2	14	23	19	11	22			
	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	21	6	7	26	4	8	29	17	27	13	10	5	3	16	9	15

Пусть 21 и 18 суть индексы двух операндов  $X = 15$  и  $Y = 4$  соответственно. На основе (3) и изложенных выше принципов сложения в  $GF(p)$  получим индекс суммы, равный  $\alpha \equiv |21 + \log_3 |1 + 3^{18-21}|_{30} |_{31}|_{30} \equiv 4$ , который соответствует сумме  $|15 + 4|_{31} \equiv 19$ .

**Пример для  $Z/(p^m)$ .** Модуль равен  $3^3 = 27$  с примитивным корнем  $g = 2$ . Также  $\Phi(3^3) = 18(3 - 1)$ . Индексное кодирование для ненулевых элементов кольца  $Z/(3^3)$  приведено ниже

x...	1	2	3	4	5	6	7	8	9	10	11	12	13
α, β...	0,0	1,0	0,1	2,0	5,0	1,1	16,0	3,0	0,2	6,0	13,0	2,1	8,0
	14	15	16	17	18	19	20	21	22	23	24	25	26
	17,0	5,1	4,0	15,0	1,2	12,0	7,0	4,1	14,0	11,0	3,1	10,0	9,0

Пусть (5,1) и (2,0) суть индексные пары операндов  $X = 15$  и  $Y = 4$  соответственно. Поскольку  $\beta_y < \beta_x$ ,  $w = 1$ . Отсюда  $(\alpha_u, \beta_u) = \log m_{2, 3} |1 + 2^{15-2}|_{18} 3^{(1-0)}|_{3^3} = \log m_{2, 3} 25 = (10,0)$ . На основе (9)–(16) индексная пара, соответствующая сумме операндов, определяется как  $(\alpha, \beta) = (|2 + 10|_{18}, 0 + 0) = (12,0)$ . Это соответствует числу 19, т. е., искомой сумме.

**Пример для  $Z/(2^5)$ .** Модуль равен  $2^5$ . Также  $2^m - 2 = 8$ . Индексное кодирование для ненулевых элементов кольца  $Z/(2^5)$  дано ниже:

x...	1	2	3	4	5	6	7	8	9	10	
α, β, γ...	0, 0, 0	1, 0, 0	0, 3, 1	2, 0, 0	0, 1, 0	1, 3, 1	0, 2, 1	3, 0, 0	0, 6, 0	1, 1, 0	
	11	12	13	14	15	16	17	18	19	20	
	0, 5, 1	2, 3, 1	0, 7, 0	1, 2, 1	0, 4, 1	4, 0, 0	0, 4, 0	1, 6, 0	0, 7, 1	2, 1, 0	
	21	22	23	24	25	26	27	28	29	30	31
	0, 5, 0	1, 5, 1	0, 6, 1	3, 3, 1	0, 2, 0	1, 7, 0	0, 1, 1	2, 2, 1	0, 3, 0	1, 4, 1	0, 0, 1

Пусть (0, 4, 1) и (2, 0, 0) являются индексными тройками операндов  $X = 15$  и  $Y = 4$  соответственно. Из (17)–(23) следует, что при  $\alpha_y \geq \alpha_x$   $w = 0$ . Тогда

$$\begin{aligned} ((\alpha_u, \beta_u, \gamma_u) &= \log m_{2, 5, -1} |1 + 2^{(2-0)}|_8 |5^{0-4}|_8 \times \\ &\times (-1)^{0-1}|_2 |_{2^5} |_{2^5} = \log m_{2, 5, -1} 29 = (0, 3, 0). \end{aligned}$$

му сумма  $(\alpha, \beta, \gamma) = (0 + 0, |4 + 3|_8, |1 + 0|_2) = (0, 7, 1)$ . Это соответствует определяемой сумме 19.

Рассмотренные выше примеры дают наглядное представление о том, какие объемы таблиц требуются для логарифмического сложения элементов конечных полей и колец при использовании систем вычетов с возможными типичными для практики значениями модулей. В [8] было показано, что 36-битный процессор может быть выполнен с использованием сбалансированного набора модулей (17, 19, 23, 25, 27, 29, 31, 32). Здесь для генерации релевантных логарифмов применяются таблицы малого объема. Так, для генерации  $\alpha_u$  для каждого простого модуля требуется таблица размерности  $32 \times 5$ . Для степенных модулей  $25 = 5^2$ ,  $27 = 3^3$  генерация индексных пар  $(\alpha_u, \beta_u)$  обеспечивается таблицами объемом  $64 \times 6$  и  $128 \times 7$  соответственно. Для модуля  $32 = 2^5$  генерация индексной тройки  $(\alpha_u, \beta_u, \gamma_u)$  требует таблицу объемом  $128 \times 7$ . Таким образом, требуемый полный объем таблиц составляет величину меньшую, чем 500 байт (вместо примерно  $2^{36}$  байт при использовании двоичной системы).

### Заключение

Логарифмические системы представления числовых данных существенно упрощают выполненные операции умножения, деления и возведения в

степень. Однако, казалось бы, такие простые операции, как сложение и вычитание, являются для них далеко не тривиальными. Известные методы логарифмического сложения для двоичных систем требуют таблицы слишком большого объема, порядка  $2^n$  бит ( $n$  — разрядность операндов в битах). В статье рассмотрена возможность использования системы вычетов и свойств конечных полей и колец в целях значительного сокращения размерности таблиц до предельно возможного минимума и повышения эффективности выполнения логического сложения. Преимущества данного подхода были продемонстрированы на конкретном примере.

### Список литературы

1. **Koren I.** Computer arithmetic algorithms, Prentice-Hall, New Jersey, 1993.
2. **Акушский И. Я. и Юдицкий Д. И.** Машинная арифметика в системе остаточных классов. М.: Сов. радио. 1968.
3. **Прасолов В. М.** Многочлены. М.: Наука, 2004.
4. **McEliece R. J.** Finite fields for computer scientists and engineers, Kluwer academic publishers, Boston, 1987.
5. **Виноградов И. М.** Теория чисел. М.: Наука, 1952.
6. **Cardarilli G. C., Lojaccond R., Martinalli G., and Salerno M.** Structurally passive filters in residue number systems // IEEE Trans., 1988. CAS-35, (2), P. 149—158.
7. **Radhakrishnan D.** "Modulo multipliers using polynomial rings // IEEE Proc. on circuits, devices, systems. 1998. 145, (6). P. 443—445.
8. **Radhakrishnan D. and Preethy A. P.** A novel 36 bit single fault tolerant multiplier using 5 bit modulo // Proc. of IEEE. 1998. texcon 9. (1). New Delhi. P. 128—130.



## ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

УДК 519

**О. В. Мандрикова**, канд. техн. наук, **В. В. Богданов**, канд. физ.-мат. наук,  
Институт космических исследований и распространения радиоволн ДВО РАН,  
Камчатская обл., с. Паратунка,  
**Ю. А. Полозов**, КамчатГТУ

### Моделирование сигналов со сложной структурой на основе конструкции вейвлет-преобразования

*Предлагается новый метод обработки сигналов, имеющих сложную структуру. Моделирование сигналов осуществлялось на основе совмещения методов вейвлет-преобразования и известного класса моделей — моделей авторегрессионно-проинтегрированного скользящего среднего. Предлагаемый подход рассматривается на примере обработки ионосферных данных. Рассматривается автоматический алгоритм совместной обработки данных критической частоты  $f_0F2$  и трехчасовых индексов геомагнитной активности  $K$ , который позволяет выделить аномальные изменения в ионосферных параметрах, предшествующие сильным землетрясениям на Камчатке.*

### Введение

В статье предлагается новый метод обработки сигналов, имеющих сложную структуру. В основе метода лежит конструкция вейвлет-преобразования, которая дает возможность выявить и деталь-

но проанализировать даже скрытые различными шумовыми факторами закономерности нестационарных временных рядов. Эта конструкция применима для представления сигналов различной природы [4, 5]. При изучении природных явлений

и процессов исследователь ставит перед собой задачу отображения в модели существенных для решения конкретной задачи внутренних взаимосвязей. Сложная структура природных сигналов значительно затрудняет этот процесс. Для поиска устойчивых изменений применяют сглаживание, что автоматически приводит к потере значимых короткопериодных вариаций. На основе конструкции вейвлет-преобразования сигнал представляется в виде разномасштабных образующих его компонентов. Это, с одной стороны, позволяет избежать потери информации, и разработчику предоставляется возможность выбора существенных для задачи компонентов сложного сигнала. С другой стороны, выделенные компоненты имеют более простую структуру, это значительно упрощает процесс построения модели и позволяет расширить спектр методов, традиционно используемых при их моделировании. В работе моделирование сигналов осуществлялось на основе известного класса моделей — моделей авторегрессии-проинтегрированного скользящего среднего (АРПСС), традиционно используемых для изучения закономерностей нестационарных временных рядов [2]. Известно, что описание временных рядов на основе параметрических методов позволяет получить точные оценки статистических характеристик процесса с высоким спектральным разрешением. Но модели данного класса имеют ряд ограничений, и это сужает область их применения [2]. В том случае, когда известна физическая модель процесса, модификация параметрической модели, как правило, не является сложной задачей и может быть решена с применением широкого класса известных нелинейных моделей. Но в большинстве задач зависимость между различными переменными неизвестна. Предлагаемый в статье подход к построению модели позволяет справиться с данными проблемами.

Обработка сложных геофизических сигналов имеет особое значение при решении целого ряда фундаментальных научных задач физики атмосферы, ионосферы, магнитосферы, распространения радиоволн и т. д. В статье для описания предлагаемого метода обработки геофизических параметров используется ряд часовых значений критической частоты  $f_0F2$ , характеризующей максимальную частоту, при которой еще происходит отражение радиопульса от ионосферного слоя  $F2$  при вертикальном его зондировании. Одновременно с регулярными изменениями параметров ионосферы, соответствующими их суточному и сезонному ходу, в сейсмоактивных регионах Земли может наблюдаться также и anomalous поведение распределения характеристик ионосферы, происходящее накануне землетрясения и определяемое механизмами литосферно-ионо-

сферных взаимодействий [1, 3]. Регистрация надежного прогностического признака сейсмического события в вариациях критической частоты сталкивается с серьезными трудностями, которые заключаются в том, что они проявляются на фоне возмущений, определяемых активностью Солнца. На основе совместной обработки и анализа часовых сигналов регистрации критической частоты  $f_0F2$  и трехчасовых  $K$ -индексов, характеризующих степень возмущенности геомагнитного поля, авторами работы были выделены anomalous изменения, предшествующие сильным сейсмическим событиям в Камчатском регионе (рассматривались события энергетического класса с коэффициентом  $k \geq 12,5$ ).

### Многомасштабный анализ сигнала

На основе конструкции вейвлет-преобразования был проведен анализ сигнала критической частоты. В качестве базисных функций при разложении использовались вейвлеты Добеши 3-го порядка [5], которые хорошо согласуются с формой сигнала. На рис. 1 показан результат непрерывного вейвлет-преобразования сигнала регистрации параметра критической частоты за 1986 г. за период 1 января — 15 июля. В верхней части рисунка показан сигнал, оттенками серого цвета показаны абсолютные значения вейвлет-коэффициентов соответствующих масштабных уровней (значения масштабных уровней отмечены на вертикальной оси), более светлым оттенкам серого цвета соответствуют большие значения коэффициентов.

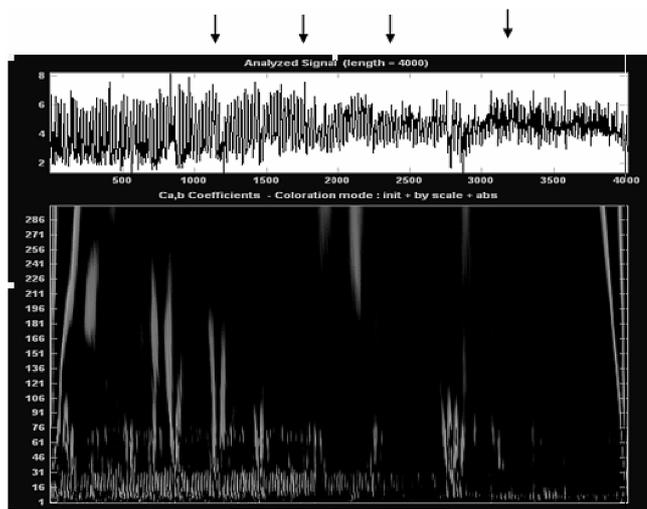


Рис. 1. Результат непрерывного вейвлет-преобразования сигнала критической частоты за период с 1 января по 15 июля 1986 г. В верхней части рисунка показан сигнал, стрелками отмечены моменты происхождения сейсмических событий, оттенками серого цвета показаны абсолютные значения вейвлет-коэффициентов соответствующих масштабных уровней (значения масштабных уровней отмечены на вертикальной оси, горизонтальная ось — ось времени), более светлым оттенкам серого цвета соответствуют большие значения коэффициентов

Данные сейсмического каталога п-ова Камчатка за 1986 г.

Дата	Коэффициент энергетического класса землетрясений $k$
2 марта	12,8
1 апреля	13,6
2 мая	13,7
17 июня	13,5

Анализ результатов преобразования подтверждает сложную структуру сигнала и показывает наличие разномасштабных колебательных составляющих сигнала. Малые масштабные уровни соответствуют высокочастотным компонентам обрабатываемого сигнала. Они отражают динамику процесса (масштабные уровни 1—30): частоту и амплитуду колебаний, их сезонный ход. Низкочастотные компоненты (масштабные уровни 30—300) позволяют выделить переходные процессы ионосферных данных, а также присутствующие в нем локальные аномальные особенности. В первую половину года (до 10 июня) практически непрерывно наблюдается наличие целого ряда ярко выраженных разномасштабных аномальных эффектов, проявляющих себя резким возрастанием вейвлет-коэффициентов в отдельных локальных областях. В весенние месяцы они накладываются на переходные процессы этого периода, что, в свою очередь, затрудняет процесс их выделения. Данные сейсмического каталога за этот год показаны в табл. 1. В верхней части рис. 1 стрелками показаны моменты происхождения сейсмических событий. На рисунке видно, что за три недели (504 отсчета сигнала) до первого сильного землетрясения можно отметить первую ярко выраженную аномалию. Далее подобные аномалии наблюдаются с периодичностью около двух недель до начала июня.

Сопоставление аномалий с данными каталога показывает, что они предшествуют сильным сейсмическим явлениям, произошедшим в этот год на Камчатке. Далее, в целях проведения более детального анализа сигнала из него была выделена аппроксимирующая компонента 4-го масштабного уровня дискретного вейвлет-разложения и дополнительно исследована. Результаты разложения данной компоненты показаны на рис. 2. На рисунке видно, что ярко проявившие себя аномальные эффекты сигнала не перешли при разложении в детализирующие компоненты. Данный факт позволяет в дальнейшем использовать этот компонент сигнала для построения модели сигнала в целях идентификации этих аномалий. Следует отметить, что полученный компонент имеет более простую структуру, что, несомненно, облегчит процесс построения модели.

### Построение модели сигнала

Процесс построения модели включает два основных этапа. Первый этап заключается в применении конструкции вейвлет-преобразования. Второй этап базируется на идентификации и оценке модели АРПСС. Получаемая при этом модель сигнала содержит многоуровневую структуру, объ-

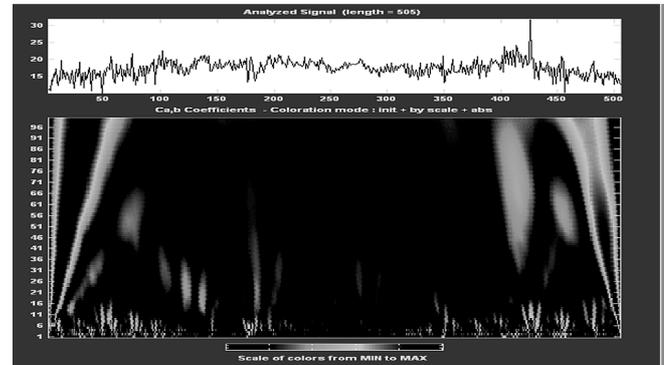


Рис. 2. Результат непрерывного вейвлет-преобразования аппроксимирующей компоненты 4-го масштабного уровня дискретного вейвлет-преобразования сигнала критической частоты за период с 1 января по 15 июля 1986г. В верхней части рисунка показан сигнал, оттенками серого цвета показаны абсолютные значения вейвлет-коэффициентов соответствующих масштабных уровней (значения масштабных уровней отмечены на вертикальной оси, горизонтальная ось — ось времени), более светлым оттенкам серого цвета соответствуют большие значения коэффициентов

единяющую кратномасштабное представление компонент временного ряда и модель АРПСС.

Модель имеет вид

$$z_t = \sum_{j, k \in Z} \langle z_t, \Psi_{b_0; j, k} \rangle \Psi_{b_0}^{j, k} + \sum_{l=1}^p \phi_l \omega_{t-l} + e(t),$$

где  $\omega_t = \nabla^d z_t$ ;  $\nabla$  — оператор взятия разности;  $d$  — порядок разности;  $\Psi_{b_0; j, k}(t) = 2^{j/2} \Psi(2^j t - kb_0)$  — базисный вейвлет;  $b_0$  — норма выборки;  $j, k \in Z$ ;  $\Psi_{b_0}^{j, k}$  — двойственный вейвлет, соответствующий базисному;  $\phi_1 \dots \phi_p$  — коэффициенты авторегрессии;  $e(t)$  — ошибка модели.

В итоге мы получаем нелинейную параметрическую модель нестационарного временного ряда, не выдвигая при этом каких-либо предположений по поводу вида ее нелинейности и учитывая нестационарности лишь определенного типа. В этом заключается основное преимущество данного метода.

При обработке использовались файлы с часовыми данными критической частоты, содержащими результаты измерений ионосферного параметра за период с 1968 по 1986 гг. С учетом се-

зонного характера поведения временного ряда процедура его разложения с последующей идентификацией модели АРПСС проводилась отдельно для каждого времени года. Так как предоставленные данные имеют разрывы, связанные с временной неисправностью аппаратуры, то для уменьшения погрешности получаемых результатов в качестве критерия выбора сезона были взяты сезоны с наименьшим числом пропусков. Анализ данных показал, что таким сезоном является зима. Идентификация модели АРПСС выполнялась для полученных после дискретного вейвлет-преобразования сглаженных компонентов сигнала. В процессе вейвлет-разложения мы, с одной стороны, получаем сглаженный компонент все более простой структуры, что как упрощает процесс идентификации модели АРПСС, так и понижает ее порядок; с другой стороны, часть полезной информации, содержащейся в сигнале, переходит в компоненты разложения. Поэтому выбор уровня разложения является важной задачей, и от него во многом зависят результаты исследования. Эта задача должна быть решена в процессе проведения экспериментов и полностью зависит от особенностей конкретного временного ряда и целей моделирования.

Важным здесь является выбор информативных компонентов исследуемого процесса. Поскольку основной целью построения модели сигнала критической частоты является выявление возможных аномальных эффектов, связанных с повышенной сейсмической активностью, в первую очередь модель была идентифицируема для периодов лет, не содержащих сейсмических событий рассматриваемого класса. Идентификация модели была проведена для аппроксимирующих компонент 3-, 4- и 5-го уровней вейвлет-разложения, в процессе чего было выявлено, что наилучшим уровнем разложения сигнала критической частоты является 4-й уровень. На 3-м уровне компонент сигнала имеет еще достаточно сложную структуру, что затрудняет процесс идентификации модели и показывает плохие оценки полученных моделей при их диагностике. На 5-м уровне разложения для каждого отдельного года была получена модель АРПСС либо 2-, либо 3-, либо 4-го порядка. Модели имели хорошие характеристики, но общий ана-

лиз ионосферного процесса в данном случае провести затруднительно.

Аппроксимирующие компоненты 4-го уровня разложения показали при идентификации удивительно хорошие результаты. Для всех сейсмически спокойных лет были идентифицированы модели 5-го порядка, имеющие хорошие характеристики (остатки являются белым шумом).

Полученные модели имели вид

$$\omega_t = \phi_1 \omega_{t-1} + \phi_2 \omega_{t-2} + \phi_3 \omega_{t-3} + \phi_4 \omega_{t-4} + \phi_5 \omega_{t-5} + e(t),$$

где  $\omega_t = \nabla \tilde{z}_t$ ,  $\tilde{z}_t = z_t - \mu$  — отклонения процесса от некоторого начального уровня.

Параметры полученных моделей оказались достаточно близки между собой. В табл. 2 приведены результаты моделирования сигналов за несколько лет. Особо следует отметить момент яркого проявления в результатах моделирования солнечного двенадцатилетнего цикла (1972—1984 гг.). Модели этих лет имеют наибольшее сходство друг с другом: константы  $\mu$ , характеризующие средний уровень процесса, и параметры модели  $\phi_i$ ,  $i = 1, 2$ . На рис. 3, 4 показаны модели аппроксимирующих компонент сигналов соответственно за 1971 и 1984 гг.

Для выявления возможных аномальных эффектов в сигнале критической частоты, предшествующих наиболее сильным сейсмическим явлениям, воспользуемся данными каталога землетрясений п-ова Камчатка, предоставленными Камчатским филиалом Геофизической службы РАН (г. Петропавловск-Камчатский). Поскольку идентификация проводилась для зимних периодов лет, не содержащих сейсмических событий, землетрясения анализируемого класса наблюдаются только в весенние месяцы, они приведены в табл. 3. На рисунках, содержащих модели компонент, моменты произошедших землетрясений отмечены стрелками. Анализ этих компонент показывает, что во всех случаях накануне сейсмических событий наблюдается увеличение амплитуды колебаний.

Далее был рассмотрен год, имеющий повышенную сейсмическую активность в зимний период. Таким годом явился 1973 г. Для него была сделана попытка идентификации параметров мо-

Таблица 2

Параметры моделей сигналов

Год	$\mu$	$\phi_1$	$\phi_2$	$\phi_3$	$\phi_4$	$\phi_5$
1971	0,03183	-0,6477	-0,7928	-0,2275	-0,539	-0,3564
1972	0,04083	-0,7416	-0,5505	-0,09206	-0,3662	-0,5355
1984	0,04617	-0,7442	-0,6598	-0,2324	-0,4047	-0,4494
1985	0,01554	-0,8846	-0,7797	-0,2702	-0,3374	-0,3738

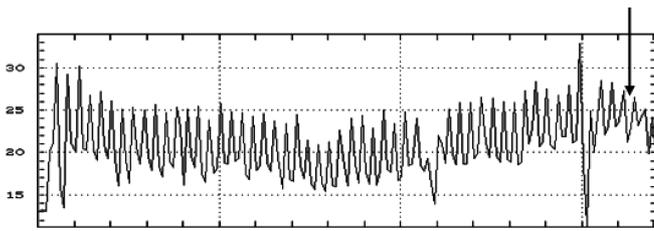


Рис. 3. Сглаженная компонента 4-го уровня дискретного вейвлет-разложения за период 15 ноября — 15 марта 1971 г. (на горизонтальной оси отмечено время, на вертикальной оси — смоделированные значения вейвлет-коэффициентов). Стрелкой отмечен момент возникновения сейсмического события

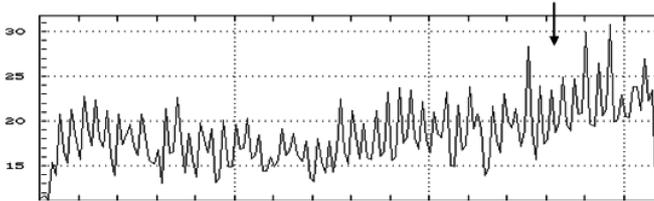


Рис. 4. Модель компонента 4-го уровня дискретного вейвлет-разложения за период 15 ноября — 15 марта 1984 г. (на горизонтальной оси отмечено время, на вертикальной оси — смоделированные значения вейвлет-коэффициентов). Стрелкой отмечен момент возникновения сейсмического события

Таблица 3

Данные сейсмического каталога п-ова Камчатка

Год	Дата	Коэффициент энергетического класса землетрясений $k$
1971	14 марта	12,9
1984	5 марта	13

дели, которая при диагностике показала плохие результаты. Кроме того, модель 1985 г., попадающего в солнечный двенадцатилетний цикл (1973—1985 гг.), имела существенное отличие в параметрах от модели 1973 г. Попытка провести идентификацию модели для 2-, 3-, 4- и 6-го порядков также оказалась неудачной. Данная ситуация, по-видимому, возникла из-за повышенной сейсмической активности данного временного периода, что позволяет сделать вывод о наличии взаимосвязей между моделируемыми ионосферными параметрами и сильными сейсмическими явлениями Камчатского региона.

#### Алгоритм выделения аномальных изменений в ионосферных параметрах на основе совместной обработки сигналов критической частоты $f_0F2$ и трехчасовых индексов геомагнитной активности $K$

Решение проблемы автоматического обнаружения и классификации аномалий позволяет более эффективно решать задачи обработки геофизических данных, обеспечивает автоматический анализ регистрируемых сигналов. В целях решения задачи автоматического обнаружения аномальных изменений в сигналах критической час-

тоты  $f_0F2$  накануне сильных землетрясений п-ова Камчатка был разработан автоматический алгоритм совместной обработки и анализа часовых сигналов регистрации критической частоты  $f_0F2$  и трехчасовых  $K$ -индексов.

Высокочастотные компоненты сигналов критической частоты  $f_0F2$  и  $K$ -индекса обрабатывались следующим образом.

1. Для каждого часа рассчитывались медианные значения сигнала критической частоты  $k_M$  за 30 дней. Затем от текущего часового значения критической частоты  $f_0F2$  этого месяца отнималось соответствующее медианное значение  $k_M$ . Таким образом были получены ряды разностей между текущим часовым значением критической частоты  $f_0F2$  и его медианным значением:

$$\omega_t = \nabla^d z_t = x_{\text{тек}} - k_M.$$

2. Умеренными магнитосферными возмущениями считаются возмущения с  $K_{\text{сумм}} = 18$

( $K_{\text{сумм}} = \sum_{i=1}^8 K_i$ , где  $K_i$  — трехчасовые значения

$K$ -индекса,  $K_{\text{сумм}}$  — суммарное значение  $K$ -индекса за сутки), слабыми — с  $K_{\text{сумм}} < 18$ . Был введен нижний порог  $K_{\text{сумм}} = 20$ , для которого анализировалось превышение значений ряда разностей  $\omega_t$ . Было введено в рассмотрение скользящее временное окно, равное суткам, что соответствует 24 отсчетам сигнала критической частоты  $f_0F2$  и восьми отсчетам сигнала  $K$ -индекса. В пределах данного окна выполнялся расчет суммы  $W_{\text{сумм}}$  положительных значений ряда разностей  $\omega_t$ :

$$W_{\text{сумм}} = \sum_1 \omega_1 : \omega_1 > 0$$

В случае выполнения условия  $\{W_{\text{сумм}} > r_{\text{порог}}, \text{ при } K_{\text{сумм}} > 20\}$ , где  $r_{\text{порог}} = 10$  — пороговое значение, определенное экспериментальным путем, данный временной период отмечается как аномальный.

В основе алгоритма обработки низкочастотных составляющих сигнала лежит конструкция непрерывного вейвлет-преобразования.

Относительно каждого базисного вейвлета  $\Psi$  непрерывное вейвлет-преобразование (НВП) в линейном нормированном пространстве Лебега  $L^2(R)$  определяется формулой

$$(W_{\Psi}f)(b, a) := |a|^{-1/2} \int_{-\infty}^{\infty} f(t) \overline{\Psi\left(\frac{t-b}{a}\right)} dt,$$

где  $f \in L^2(R)$ ,  $\Psi_{b,a}(t) := |a|^{-1/2} \overline{\Psi\left(\frac{t-b}{a}\right)}$  — базисный вейвлет;  $a, b \in R$ ,  $a \neq 0$ ,  $a$  — масштабный параметр,  $b$  — параметр смещения по времени.

В качестве базисных функций при обработке использовались ортогональные вейвлеты Добеши 3-го порядка. Данные базисные функции имеют компактный носитель, что позволяет при обработке разрывных функций получать результаты с наименьшей погрешностью, и, как было отмечено выше, хорошо согласуются с формой сигнала. После НВП сигнала для уровней разложения  $a = \overline{20, 200}$  сохранились коэффициенты, абсолютные значения которых превышают пороговое значение  $r(a)$ ,  $a = \overline{20, 200}$ :

$$(W_{\Psi}f)(b, a) = 0, \\ \text{если } |(W_{\Psi}f)(b, a)| < r(a).$$

Пороговое значение  $r(a)$  определялось экспериментальным путем. В случае обнаружения аномалии  $|(W_{\Psi}f)(b, a)| > r(a)$ ,  $b = \overline{i, j}$ , определялась ее временная протяженность  $l = (j - i)2^a$  и на каждом масштабном уровне  $a$  определялась энергия аномальной окрестности сигнала:

$$E(a) = \int_i^j (W_{\Psi}f)'(b, a) db; \\ (W_{\Psi}f)'(b, a) = \\ |(W_{\Psi}f)(b, a)| / (\max_{b = \overline{i, j}} |(W_{\Psi}f)(b, a)| - \\ - \min_{b = \overline{i, j}} |(W_{\Psi}f)(b, a)|).$$

Затем проводился расчет общей энергии сигнала для уровней разложения  $a = \overline{20, 200}$ :

$$E = \int_{20}^{200} E(a) da.$$

В процессе экспериментов были обработаны сигналы критической частоты  $f_0F2$  и  $K$ -индекса за период с 1975 по 1985 гг. Перед обработкой сигналы были разделены на 40 интервалов по 3,5 мес. Интервалы рассматривались с двухнедельным перекрытием в целях устранения краевого эффекта в начале и в конце каждого изучаемого интервала. За анализируемый период на Камчатке в радиусе  $R \sim 300$  км от П.-Камчатского произошло 72 сейсмических события с  $k \geq 12,5$ , из них 24 землетрясения с  $k \geq 13$ . Накануне 57 землетрясений на основе предложенной методики были выделены аномальные изменения регистрируемых параметров, которые предшествовали сейсмическим явлениям от нескольких дней до 1,5 мес. На рис 5, 6 в каче-

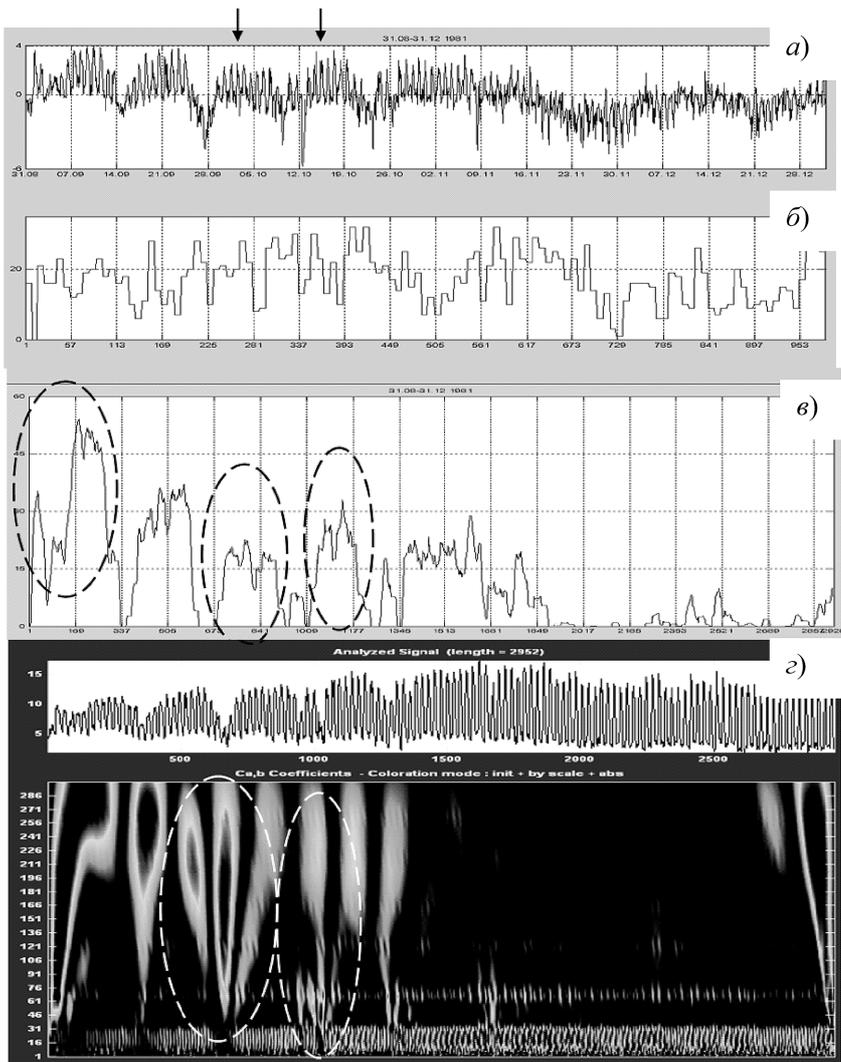


Рис. 5. Ряд разностей  $\omega_r$  текущего значения сигнала критической частоты и соответствующего медианного значения за период 31.08.81–31.12.81 (рис. 5, а); сигнал регистрации  $K$ -индекса за период 31.08.81–31.12.81 (рис. 5, б); суммарные значения  $W_{\text{сумм}}$  положительных значений ряда разностей  $\omega_r$  (рис. 5, в); результаты непрерывного вейвлет-преобразования сигнала критической частоты  $f_0F2$  за период 31.08.81–31.12.81 (рис. 5, г): сверху представлен сам сигнал, ниже результаты его разложения, по горизонтальной оси — время, по вертикальной оси — масштабные уровни, оттенками серого цвета отражены значения вейвлет-коэффициентов соответствующего масштабного уровня. Стрелками показаны моменты возникновения землетрясений энергетического класса с  $k \geq 12,5$  в радиусе  $R \sim 300$  км от П.-Камчатского

стве примера показаны результаты обработки различных интервалов времени. На рис. 5, а, б, а показаны значения ряда разностей между текущим значением критической частоты  $f_0F2$  и медианным значением  $\omega_r$ . На рис. 5, б, б показаны значения  $K$ -индекса за анализируемый период. На рис. 5, в, в показаны суммарные значения  $W_{\text{сумм}}$  положительных значений ряда разностей  $\omega_r$ . Результаты вейвлет-преобразования показаны на рис. 5, г, г: сверху представлен сам сигнал, ниже результаты его разложения, по горизонтальной оси — время, по вертикальной оси — масштабные уровни, оттенками серого цвета отражены значения

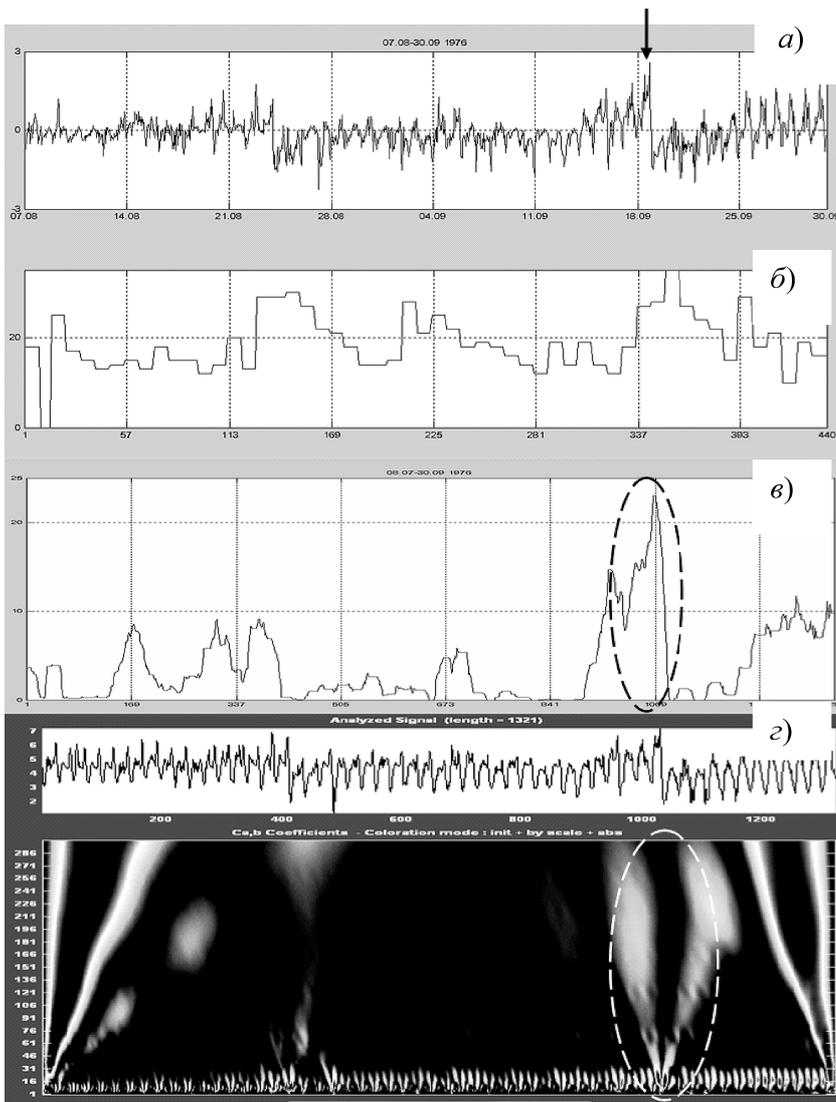


Рис. 6. Ряд разностей  $\omega$ , текущего значения сигнала критической частоты и соответствующего медианного значения за период 07.08.76—30.09.76 (рис. 6, а); сигнал регистрации  $K$ -индекса за период 07.08.76—30.09.76 (рис. 6, б); суммарные значения  $W_{\text{СУММ}}$  положительных значений ряда разностей  $\omega_i$  (рис. 6, в); результаты непрерывного вейвлет-преобразования сигнала критической частоты  $f_0F2$  за период 07.08.76—30.09.76 (рис. 6, г): вверху представлен сам сигнал, ниже — результаты его разложения, по горизонтальной оси — время, по вертикальной оси — масштабные уровни, оттенками серого цвета отражены значения вейвлет-коэффициентов соответствующего масштабного уровня. Стрелками показаны моменты возникновения землетрясений энергетического класса с  $k \geq 12,5$  в радиусе  $R \sim 300$  км от П.-Камчатского

вейвлет-коэффициентов соответствующего масштабного уровня. Стрелками показаны моменты возникновения землетрясений. На основе предлагаемого алгоритма накануне землетрясений были выделены аномальные периоды, которые отмечены на рис. 5, 6 штриховыми линиями.

На рис. 5 показаны результаты обработки осеннего периода времени (31.08.81—01.12.81). Данный период характеризуется наличием в сигнале критической частоты разномасштабных переходных процессов, которые четко прослеживаются на основе анализа результатов непрерывного

вейвлет-преобразования (на рис. 1, 2 отмечаются более светлым цветом). Это затрудняет выделение аномалий, предшествующих сейсмическим событиям. Несмотря на данный эффект рассматриваемая методика и разработанный на ее основе автоматический алгоритм позволили выделить аномальные особенности, предшествующие сильным землетрясениям на Камчатке (отмечены на рис. 5, 6 штриховыми линиями). На рис. 6 показаны результаты обработки летнего периода времени (07.08.76—30.09.76). В данный период времени сигнал критической частоты имеет более простую структуру, о чем свидетельствуют результаты непрерывного вейвлет-преобразования сигнала. На рис. 6 видно, что за анализируемый период можно отметить наличие только одного аномального эффекта, который отмечен штриховой линией и наблюдается за двое суток до сейсмического события.

## Заключение

На основе новой математической конструкции — вейвлет-преобразования — были исследованы флуктуации сигнала критической частоты на Камчатке и выделены основные характерные для него компоненты. Сопоставление обрабатываемых временных периодов с данными сейсмического каталога позволило выявить разномасштабные аномальные эффекты в ионосферном сигнале, которые предшествуют сильным сейсмическим событиям энергетического класса с  $k \geq 12,5$  п-ова Камчатка. На основе предлагаемого подхода к построению моделей сложных природных

сигналов было показано, что для сигналов критической частоты в периоды без повышенной сейсмической активности может быть идентифицирована параметрическая модель. Исследование модели подтвердило наличие взаимосвязей между ионосферными параметрами и сильными сейсмическими явлениями Камчатского региона. Также был разработан автоматический алгоритм, позволяющий выделить аномальные изменения в ионосферных параметрах накануне сильных землетрясений на Камчатке (с  $k \geq 12,5$  в радиусе  $R \sim 300$  км от П.-Камчатского).

## Список литературы

1. Богданов В. В., Геппенер В. В., Мандрикова О. В. Моделирование временных рядов геофизических параметров на основе вейвлет-преобразования. СПб.: Изд-во СПбГЭТУ "ЛЭТИ", 2006.
2. Никифоров И. В. Последовательное обнаружение изменения свойств временных рядов. М.: Наука, 1983.
3. Сорокин В. М., Чмырев В. М., Похотелов О. А., Липеровский В. А. Обзор моделей литосферно-ионосферных связей в периоды подготовки землетрясений // Краткосрочный прогноз катастрофических землетрясений с помощью радиофизических наземно-космических методов. Докл. конф. 2–3 октября 1997 г. М.: ИКИ РАН, 1998. С. 64–87.

4. Фирстов П. П., Филиппов Ю. А., Мандрикова О. В. Предвестниковые аномалии сильных землетрясений в динамике подпочвенного радона на Петропавловск-Камчатском геодинамическом полигоне в 1997–2001 гг. // ДАН. 2003. Т. 389. № 6. С. 810–813.

5. Добеши И. Десять лекций по вейвлетам: Пер. с англ. Ижевск: НИЦ "Регулярная и хаотическая динамика". 2001.

УДК 004(.052.4+.021):912.644.4

Р. М. Алгулиев, д-р техн. наук,  
Г. Г. Оруджов, Э. Н. Сабзиев, Н. А. Панахов,  
Н. В. Расулова, А. А. Алиева  
Институт информационных технологий НАН  
Азербайджана, г. Баку

## Разработка алгоритмов обнаружения и корректировки некоторых характерных ошибок векторной карты рельефа местности

*Целью работы является разработка алгоритмов для определения и устранения ошибок и неточностей векторной карты рельефа местности с помощью математических методов. Разработаны алгоритмы, обеспечивающие обнаружение характерных ошибок векторной карты в автоматическом режиме.*

**Введение.** Одним из составных компонентов ГИС-продукции является векторная карта рельефа местности, которая на отдельном слое содержит высотные изолинии (горизонталы). Подобная карта обычно создается на основе топографических карт на бумажных носителях нанесением объектов типа *line (polyline)* в AutoCAD, Arc View или в других графических редакторах. Однако подобного рода однообразная и трудоемкая операторская работа приводит к многочисленным ошибкам и неточностям, что делает использование этой карты непригодным для решения различных задач, и поэтому подобная карта принимается как первый (грубый) вариант векторной карты местности.

Целью работы является определение, классификация и устранение ошибок и неточностей векторной карты рельефа местности с помощью математических методов обработки.

**Основные понятия и определения.** Прежде всего, приведем основные понятия и определения объектов, используемые в данной работе.

Изолинией, или высотной изолинией, называется один отрезок или одна ломаная линия, соответствующая определенному значению высоты. Очевидно, к одной и той же высоте будут относиться много изолиний. Изолиния называется простой, если она состоит только из одного отрезка.

Изолиния может быть замкнутой, когда у нее нет свободного узла. Изолиния называется простой замкнутой, если в каждом узле соединены два отрезка.

Приведем типичные недостатки первого варианта векторной карты:

- повторение отрезков некоторых изолиний (повторяющиеся лишние отрезки) (рис. 1, а, б);
- зарисовка высотных изолиний в виде последовательности отрезков с несовпадающими узлами без пересечения или с простым пересечением (рис. 2, а, б);
- неправильное указание значений высот изолиний;
- пересечение изолиний, соответствующее разным высотам (рис. 3);

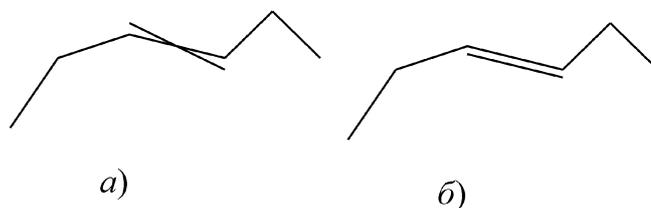


Рис. 1

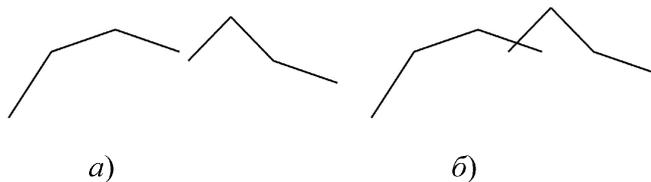


Рис. 2



Рис. 3



Рис. 4

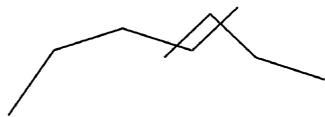


Рис. 5



Рис. 6

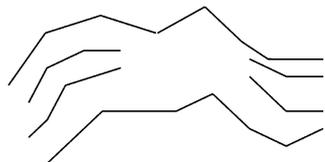


Рис. 7

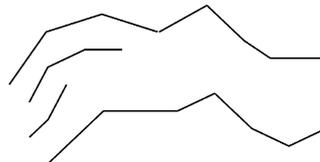


Рис. 8

- более сложные пересечения узловых точек (рис. 4, 5, 6);
- отсутствие части некоторых изолиний (рис. 7, 8).

Как видно, все выявленные недостатки карты, главным образом, сосредоточены вблизи узловых точек (такие точки называются проблемными узловыми точками).

С математической точки зрения изолиния — это последовательность двумерных координат узловых точек, соответствующая заданной высоте. Каждая такая линия, участвующая в обработке, может быть представлена в виде следующего набора данных:

$$N, P, H, x_1, y_1, x_2, y_2, \dots, x_{M-1}, y_{M-1}, x_M, y_M$$

где  $N$  — порядковый номер изолинии на карте;  $P$  — числовой классификатор изолинии;  $H$  — значение высоты данной изолинии;  $M > 1$  — число узловых точек изолинии;  $(x_k, y_k), k = 1, 2, \dots, M$  — координаты узловых точек в порядке расположения вдоль изолинии.

Числовой классификатор  $P$  является одним из управляющих параметров процесса обработки и в ходе процесса обработки принимает одно из следующих значений:

$P = M$  — данная изолиния участвует в обработке карты;

$P = 1$  — данная изолиния незамкнута и не участвует в обработке карты;

$P = 0$  — данная изолиния замкнута и не участвует в обработке карты.

Пусть заданы две узловых точки  $(x', y')$ ,  $(x'', y'')$  и некоторое положительное число  $\varepsilon$ . Будем говорить, что данные узловых точки  $\varepsilon$ -близки, если выполнено неравенство

$$(x' - x'')^2 + (y' - y'')^2 < \varepsilon.$$

Пусть теперь заданы две изолинии

$$1, n_1, H, x_1^1, y_1^1, x_2^1, y_2^1, \dots, x_{m_1-1}^1, y_{m_1-1}^1, x_{m_1}^1, y_{m_1}^1,$$

$$2, n_2, H, x_1^2, y_1^2, x_2^2, y_2^2, \dots, x_{m_2-1}^2, y_{m_2-1}^2, x_{m_2}^2, y_{m_2}^2$$

и некоторое положительное число  $\varepsilon$ . Будем говорить, что данные изолинии  $\varepsilon$ -близки, если  $\varepsilon$ -близки точки одной из следующих пар узловых точек:

$$\{(x_{m_1}^1, y_{m_1}^1), (x_1^2, y_1^2)\}, \{(x_{m_1}^1, y_{m_1}^1), (x_{m_2}^2, y_{m_2}^2)\}, \\ \{(x_1^1, y_1^1), (y_1^1, y_2^1)\}, \{(x_1^1, y_1^1), (x_{m_2}^2, y_{m_2}^2)\}.$$

Объединением  $\varepsilon$ -близких изолиний называется, в зависимости от узлов близости, соответственно, одна из следующих изолиний:

$$1, n_1 + n_2 - 1, H, x_1^1, y_1^1, x_2^1, y_2^1, \dots, x_{m_1-1}^1, y_{m_1-1}^1, x_{m_1}^1, y_{m_1}^1, x_2^2, y_2^2, \dots, x_{m_2-1}^2, y_{m_2-1}^2, x_{m_2}^2, y_{m_2}^2;$$

$$1, n_1 + n_2 - 1, H, x_1^1, y_1^1, x_2^1, y_2^1, \dots, x_{m_1-1}^1, y_{m_1-1}^1, x_{m_1}^1, y_{m_1}^1, x_{m_2-1}^2, y_{m_2-1}^2, \dots, x_2^2, y_2^2, x_1^2, y_1^2;$$

$$1, n_1 + n_2 - 1, H, x_{m_2}^2, y_{m_2}^2, x_{m_2-1}^2, y_{m_2-1}^2, \dots, x_2^2, y_2^2, x_1^1, y_1^1, x_2^1, y_2^1, \dots, x_{m_1-1}^1, y_{m_1-1}^1, x_{m_1}^1, y_{m_1}^1, x_{m_2-1}^2, y_{m_2-1}^2;$$

$$1, n_1 + n_2 - 1, H, x_1^2, y_1^2, x_2^2, y_2^2, \dots, x_{m_2-1}^2, y_{m_2-1}^2, x_1^1, y_1^1, x_2^1, y_2^1, \dots, x_{m_1-1}^1, y_{m_1-1}^1, x_{m_1}^1, y_{m_1}^1.$$

**Основная часть.** В целях выявления проблемных узлов и корректировки изолиний ниже сформулированы несколько задач и приводится алгоритм их решения[1].

**Задача 1.** Обнаружение и устранение не совпадающих узловых точек (см. рис. 2, а, б).

При решении этой задачи осуществляется соединение изолиний с несовпадающими узлами и попутно выявляются возможные неправильно указанные значения высот. Предложен следующий алгоритм, который начинается с первой изо-

линии и выполняется поочередно для всех последующих.

1. Фиксируется базовая изолиния  $j = 1, 2, 3, \dots$

2. Для разумно подобранной  $\varepsilon = \varepsilon_C$  по критерию  $\varepsilon$ -близости конечных узлов точек данной изолинии устанавливается ее замкнутость. Если изолиния замкнута, то:

- ее числовому классификатору  $P_j$  присваивается значение 0;
- координаты одного из этих узлов принимаются за основу, и они отождествляются;
- осуществляется переход к проверке следующей изолинии, и она фиксируется в качестве базовой.

3. Продолжается последовательное рассмотрение изолиний с номерами  $i = j + 1, j + 2, j + 3, \dots$

4. Если для заданного  $\varepsilon = \varepsilon_L$  изолинии  $j$  и  $i$   $\varepsilon$ -близки, но значения высот не совпадают, т. е.  $H_j \neq H_i$ , то:

- информация об этом узле выводится в соответствующий информационный файл (HND.txt) для дальнейшего исследования на предмет правильности написанных значений высот;
- продолжение процесса осуществляется для следующего значения  $i$ .

5. Если для  $\varepsilon = \varepsilon_L$  изолинии  $j$  и  $i$   $\varepsilon$ -близки и значения высот совпадают, т. е.  $H_j = H_i$ , то осуществляется дополнительная проверка взаимного расположения примыкающих отрезков. Если примыкающие к соответствующим узлам отрезки имеют не более одного пересечения (случаи, показанные на рис. 2, а, б), то:

- базовая изолиния заменяется их объединением;
- для числового классификатора  $P_i$  устанавливается значение 1;
- продолжение процесса осуществляется с п. 3.

6. Если примыкающие к соответствующим узлам отрезки имеют более одного пересечения (например, случаи, показанные на рис. 4—6), то информация об этом узле выводится в соответствующий информационный файл (VCN.txt) для дальнейшего визуального анализа и принятия решения по устранению ситуации.

Описанный алгоритм выполняется несколько раз с постепенным увеличением контрольных значений  $\varepsilon_C$  и  $\varepsilon_L$ , при этом обычно процедуру объединения контролируют просмотром информации об этих узлах, предварительно записывая их в соответствующий файл (PVN.txt).

**Задача 2. Обнаружение и классификация свободных узловых точек.** Идеология алгоритма решения данной задачи аналогична идеологии предыдущего алгоритма. Она позволяет оператору рассмотреть и классифицировать узловые точки на экране соответствующего графического редактора. Были рассмотрены следующие варианты:

- узлы, относящиеся к границе карты;

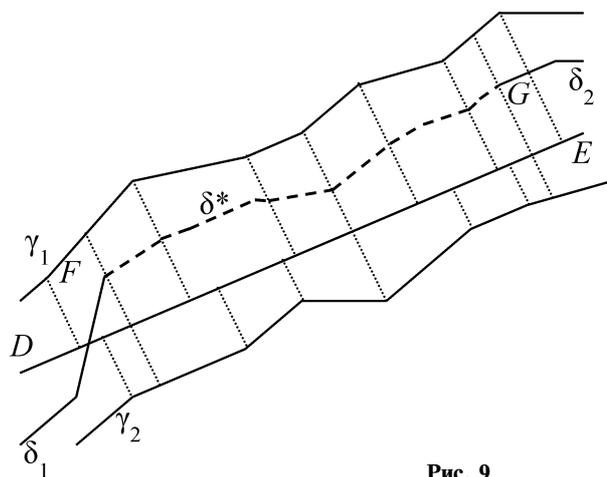


Рис. 9

- узлы, относящиеся к разорванным изолиниям, которые следует восстанавливать (случай, соответствующий рис. 7);
- узлы, для которых дальнейшая обработка требует дополнительной информации, например необходима исходная картографическая информация (случай типа рис. 8).

**Задача 3. Восстановление отсутствующих частей изолиний.**

Исходной информацией для решения этой задачи являются (рис. 9):

- огибающие изолинии ( $\gamma_1, \gamma_2$ );
- некоторое направление  $DE$ , на которое однозначно проецируются изолинии  $\gamma_1, \gamma_2$ ;
- координаты узловых точек  $F$  и  $G$  прерванной изолинии  $\delta_1$  и  $\delta_2$ , относящиеся к одинаковой высоте  $H$ .

Огибающие  $\gamma_1, \gamma_2$  и точки  $D, E$ , которые определяют направление  $DE$ , вводятся редактирующим оператором в интерактивном режиме.

Алгоритм решения задачи предусматривает осуществление следующих процедур.

1. Определение элементов матрицы преобразования

$\mathbf{A} = \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix}$  на основе координат направляющей линии, соединяющей точки  $D(x_D, y_D)$  и  $E(x_E, y_E)$ :

$$\alpha = \frac{x_E - x_D}{\sqrt{(x_E - x_D)^2 + (y_E - y_D)^2}},$$

$$\beta = \frac{y_E - y_D}{\sqrt{(x_E - x_D)^2 + (y_E - y_D)^2}}.$$

(В дальнейшем координаты точек после преобразования  $\mathbf{A}$  будем обозначать соответствующими

заглавными буквами, т. е.  $\mathbf{A} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} X \\ Y \end{pmatrix}$ .)

2. Осуществление преобразования  $\mathbf{A}$  всех узловых точек изолиний  $\gamma_1, \gamma_2$  и узловых точек  $G(x_G, y_G) \in \delta_1$  и  $F(x_F, y_F) \in \delta_2$ .

3. Упорядочение последовательности всех пар координат узловых точек в порядке роста первых координат.

В данной последовательности, перед координатами, узловой точке  $G(X_G, Y_G)$  должна предшествовать как минимум одна узловая точка из каждой огибающей изолинии  $\gamma_1, \gamma_2$ . Эта последовательность также должна замыкаться, как минимум, одной узловой точкой из каждой огибающей  $\gamma_1, \gamma_2$ , эти точки последуют за узловой точкой  $F(X_F, Y_F)$ . В предложенном алгоритме крайние "лишние" узловые точки не участвуют. Но если таковые имеются, они игнорируются и предполагается, что третья точка с начала последовательности соответствует узловой точке  $G$ , а третья точка с конца — точке  $F$ . Для простоты дальнейшего изложения предположим, что полученная последовательность имеет следующий вид:

$$(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), (X_4, Y_4), \dots, (X_{n-2}, Y_{n-2}), (X_{n-1}, Y_{n-1}), (X_n, Y_n).$$

4. Определение коэффициента удаленности  $\lambda(X)$  узловых точек  $G$  и  $F$  от огибающих изолиний  $\gamma_1, \gamma_2$  вдоль оси  $DE$ , как функция аргумента  $X \in [X_F, X_G]$ . Оно осуществляется следующим образом.

Пусть  $\{(X'_1, Y'_1), (X'_2, Y'_2)\}$  и  $\{(X''_1, Y''_1), (X''_2, Y''_2)\}$  — последовательные узловые точки, которые относятся к изолиниям  $\gamma_1$  и  $\gamma_2$ , и при этом выполняются условия  $X'_1 \leq X_F \leq X'_2$  и  $X''_1 \leq X_F \leq X''_2$ .

Вычислим

$$\lambda_1 \equiv \lambda(X_F) = \frac{Y'_1 + \frac{Y'_2 - Y'_1}{X'_2 - X'_1}(X_F - X'_1) - Y_F}{Y_F - Y''_1 - \frac{Y''_2 - Y''_1}{X''_2 - X''_1}(X_F - X''_1)}.$$

Заменяя аналогичным образом  $(X_F, Y_F)$  на  $(X_G, Y_G)$ , вычислим  $\lambda_2 \equiv \lambda(X_G)$ . Теперь на основании этих значений определяется искомый коэффициент:

$$\lambda(X) = \lambda_1 + \frac{\lambda_2 - \lambda_1}{X_G - X_F}(X - X_F), X \in [X_F, X_G].$$

5. Вычисление координат узловых точек изолинии, соединяющей  $\delta_1$  и  $\delta_2$ . Эта процедура осуществляется следующим образом.

Пусть  $(X', Y')$  — произвольный узел  $\gamma_1$ , такой, что  $X' \in [X_F, X_G]$ . Также пусть  $(X''_1, Y''_1), (X''_2, Y''_2)$  — по-

следовательные узлы  $\gamma_2$ , такие, что  $X' \in [X''_1, X''_2]$ . Вычислим

$$Y(X') = \lambda(X') + \frac{\lambda(X') \left( Y''_1 + (X' - X''_1) \frac{Y''_2 - Y''_1}{X''_2 - X''_1} \right)}{1 + \lambda(X')}.$$

Аналогично вычисляются  $Y(X'')$  для всех узлов  $(X''_i, Y''_i) \in \gamma_2$ , таких, что  $X''_i \in [X_F, X_G]$ ,  $(i = 1, 2)$ . Таким образом, получается множество пар  $(X, Y)$ , для которых первая координата принадлежит хотя бы к одной из изолиний  $\gamma_1, \gamma_2$  в интервале  $[X_F, X_G]$ .

Заметим, что в случае, когда узловые точки  $\gamma_1, \gamma_2$  имеют одинаковое значение  $X$ , соответствующие значения  $Y$ , вычисленные по обеим формулам, совпадают. Поэтому, исключая повторяющиеся пары, если таковые имеются, и упорядочивая их в порядке роста первых координат, получаем некоторую последовательность узловых точек  $\{(X_k, Y_k)\}_{k=1, 2, \dots, k_0}$ , где  $k_0$  — общее число найденных узлов.

6. Осуществление обратного преобразования координат найденных узловых точек:

$$\begin{pmatrix} x_k \\ y_k \end{pmatrix} = \mathbf{A}^{-1} \begin{pmatrix} X_k \\ Y_k \end{pmatrix}, k = 1, 2, \dots, k_0.$$

Таким образом определяются координаты узловых точек новой изолинии  $\delta^*$ .

7. Осуществление последовательных объединений изолиний  $\delta_1, \delta^*$  и  $\delta_2$ , что завершает решение задачи 3.

Применение алгоритма задачи 3 создает все предпосылки, чтобы осуществлять выделение всех простых изолиний, для которых  $M = 2$ . Это позволяет идентифицировать изолинии, относящиеся к случаям, изображенным на рис. 1, а, б, и удалять их из карты.

**Задача 4.** Выявление изолиний, имеющих пересечение и относящихся к разным высотам (см. рис. 3).

Для решения этой задачи предложен следующий алгоритм.

1. Упорядочить все изолинии в порядке роста высот  $H_j \leq H_i$  ( $j < i$ ) и заново нумеровать их. Пусть последовательности всех высот, с исключением повторяющихся, удовлетворяет следующее неравенство  $H^1 > H^2 > \dots > H^r$ .

2. Проверить на наличие пересечений каждой изолинии, соответствующей  $H^i$ , с изолиниями, соответствующими  $H^{i+1}$ , начиная с изолиний с наибольшей высотой. Это осуществляется следующим образом:

- последовательно рассматриваются все пары узловых точек изолиний  $H^i$  и  $H^{i+1}$  (соответственно,  $\{(x_1^i, y_1^i), (x_2^i, y_2^i)\}, \{(x_1^{i+1}, y_1^{i+1}), (x_2^{i+1}, y_2^{i+1})\}$ );

- если обнаруживается выполнение одного из ниже перечисленных случаев, констатируется факт пересечения изолиний:

$$(x_1^i = x_2^i = x_1^{i+1} = x_2^{i+1}) \text{ and } ((y_1^{i+1} \in [\min\{y_1^i, y_2^i\}, \max\{y_1^i, y_2^i\}] \text{ or } (y_2^{i+1} \in [\min\{y_1^i, y_2^i\}, \max\{y_1^i, y_2^i\}])));$$

$$(x_1^i = x_2^i) \text{ and } (x_1^{i+1} \neq x_2^{i+1}) \text{ and } (y_2^{i+1} + \frac{y_1^{i+1} - y_2^{i+1}}{x_1^{i+1} - x_2^{i+1}} (x_2^i - x_2^{i+1}) \in [\min\{y_1^i, y_2^i\}, \max\{y_1^i, y_2^i\}]);$$

$$(x_1^i \neq x_2^i) \text{ and } (x_1^{i+1} = x_2^{i+1}) \text{ and } (y_2^i + \frac{y_1^i - y_2^i}{x_1^i - x_2^i} (x_2^{i+1} - x_2^i) \in [\min\{y_1^{i+1}, y_2^{i+1}\}, \max\{y_1^{i+1}, y_2^{i+1}\}]);$$

$$(x_1^i \neq x_2^i) \text{ and } (x_1^{i+1} \neq x_2^{i+1}) \text{ and } \left( \frac{y_2^{i+1} - y_2^i + \frac{y_1^i - y_2^i}{x_1^i - x_2^i} x_2^i - \frac{y_1^{i+1} - y_2^{i+1}}{x_1^{i+1} - x_2^{i+1}} x_2^{i+1}}{\frac{y_1^i - y_2^i}{x_1^i - x_2^i} - \frac{y_1^{i+1} - y_2^{i+1}}{x_1^{i+1} - x_2^{i+1}}} \in [\min\{x_1^i, x_2^i\}, \max\{x_1^i, x_2^i\}] \cap [\min\{x_1^{i+1}, x_2^{i+1}\}, \max\{x_1^{i+1}, x_2^{i+1}\}] \right).$$

3. Визуализация полученной информации для дальнейшей обработки.

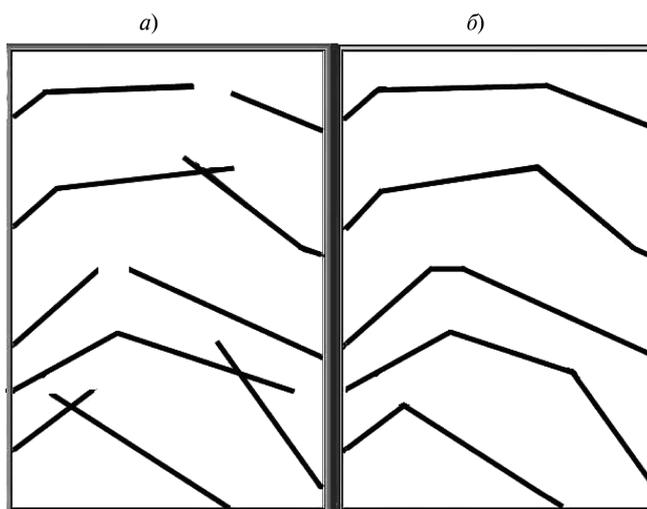


Рис. 10

**Результаты применения.** На основе предложенных алгоритмов разработаны программные модули (скрипты) на алгоритмическом языке Avenue в среде Arc View 3.2. Результаты моделирования приведены на рис. 10, а, — изолинии с ошибками, и на рис. 10, б, — изолинии, скорректированные с помощью предложенных алгоритмов [2, 3, 4].

Разработанные скрипты позволяют обнаруживать в автоматическом режиме допущенные ошибки и неточности, следовательно, заметно повышают эффективность корректировки векторной карты рельефа местности. Следующим шагом в этом направлении может быть полная автоматизация процесса корректировки.

#### Список литературы

1. Беклемишев Д. В. Курс аналитической геометрии и линейной алгебры. М.: Наука, 1987. 320 с.
2. Программа для визуализации, запроса и анализа пространственной информации, [www.esri.com/software/arcgis/arcview/index.html](http://www.esri.com/software/arcgis/arcview/index.html)
3. Using Avenue. Customization and Application Development for Arc View GIS, ESRI, 1996.
4. GeoCommunity — Software, Arc View Scripts and Downloads, <http://software.geocomm.com/scripts/arcview/>.

Д. И. Есаков,

Московский авиационный институт  
(государственный технический университет)

## Объединение нескольких методов при автоматизированной обработке изображений земной поверхности

*Рассмотрено применение сопряжения нескольких методов для многоступенчатого поиска изменений на изображениях; приведены описания методов, реализованных на каждом из этапов обработки; описаны полученные результаты и возможные области применения.*

### 1. Актуальность исследования

При решении значительной части задач дистанционного зондирования используется оперативное наблюдение за состоянием земной поверхности, цель которого — обнаружение резких изменений в структуре наблюдаемых объектов. В ходе этого процесса возникает проблема обработки большого потока поступающей видеоинформации, которая требует интенсивного использования вычислительных, финансовых и операторских ресурсов пользователя, поскольку процесс обработки одного кадра является продолжительным. При этом экспертный метод является медленным, утомляющим и может порождать значительные ошибки [1].

Время обработки и принятия решения возможно сократить в десятки раз, если передавать потребителю не все снимки, а только те, на которых имеются признаки топологических аномалий земной поверхности, а именно изменений по сравнению с некоторым снимком, считающимся эталонным. Причем затраты на анализ могут быть еще более сокращены, если проводить анализ изображений и выделять на них области с изменениями.

Основная трудность в процедуре сравнения между собой космических видеоизображений с использованием традиционных методов заключается в том, что каждое новое поступающее видеоизображение одного и того же участка Земли всегда отличается от предыдущего вследствие изменения условий съемки. До настоящего времени был разработан ряд методов, на основе которых проводят анализ видеоизображений в целях определения изменений на них [2—5]. Эти методы, которые основываются на анализе яркостных и цветовых параметров элементов видеоизображений

земной поверхности, не учитывают связей между отдельными элементами видеоизображения и не используют интегральных характеристик, описывающих земную поверхность как объект в целом. Поэтому такие методы не способны отличать изменения, связанные с вариациями условий съемки, от изменений, вызванных вариациями самой снимаемой поверхности (сцены).

Из сказанного следует, что в качестве информативного параметра, используемого методом обнаружения аномалий на земной поверхности, должна служить характеристика, наиболее чувствительная именно к изменению топологии земной поверхности на космическом видеоизображении и инвариантная к условиям съемки, что позволит максимизировать вероятность обнаружения при прочих равных условиях по сравнению с известными методами.

### 2. Формализованная постановка задачи

Таким образом, формально задача сравнения и анализа изображений может быть записана в следующем виде.

#### *Исходные данные для обработки*

Текущее наблюдение:

$$z_f(x, y) = f(x, y), \text{ где } f(x, y) = S_f(t_{т.н}) + \delta_f(t_{т.н}).$$

Эталонное наблюдение:

$$z_g(x, y) = g(x, y), \text{ где } g(x, y) = S_g(t_{э.н}) + \delta_g(t_{э.н}),$$

где  $z$  — это измеренная функция яркости изображения;  $S_f$ ,  $S_g$ ,  $\delta_f$ ,  $\delta_g$  — совокупности яркостей, представляющие, соответственно, сцены и условия съемок эталонного и текущего изображений;  $x, y$  — двумерные координаты пикселя изображения;  $t_{т.н}$ ,  $t_{э.н}$  — моменты времени, в которые проводились текущее наблюдение и эталонное наблюдение.

#### *Требуется*

Разработать алгоритм  $\Pi_{fg}$ , позволяющий по двум предъявленным для анализа изображениям одной и той же сцены, текущему  $f(x, y)$  и эталонному  $g(x, y)$ , найти изменения сцены изображений вне зависимости от изменений условий съемки на обрабатываемых изображениях:

$$\Pi_{fg} : \{f(x, y); g(x, y)\} \rightarrow \Delta_S \begin{cases} \neq 0 \\ 0 \end{cases}. \quad (1)$$

Здесь  $\Delta_S$  — это формальный параметр, характеризующий только изменения сцены, независимый от условий съемки. В том случае если  $\Delta_S$  равно нулю, изменения сцены отсутствуют. В противном случае имеются изменения между сценами изображений  $f(x, y)$  и  $g(x, y)$ , которые и представляют интерес.

### 3. Выбор критерия поиска

Для анализа изображений, в частности в целях поиска изменений, можно использовать его различные характеристики: яркость пикселя; среднюю яркость изображения; показатель фрактальной размерности (ПФР); форму изображения и др. Однако не все из них применимы при решении задачи оперативного наблюдения в силу указанных выше особенностей изображений, поступающих на обработку. Поэтому в качестве критериев поиска  $K$  были выбраны ПФР и форма изображения, как обладающие свойством инвариантности получаемых результатов при изменении условий съемки, несмотря на существенные вычислительные затраты при их расчете. Эти характеристики являются независимыми от преобразований яркости, т. е. таких преобразований, которые определяются только яркостью пикселя поля зрения  $X$  и не зависят от его расположения на  $X$ , например, изменения значений пикселей в несколько раз.

#### 4. Описание предлагаемого метода поиска

Предлагается использовать комбинацию двух методов: метода анализа показателя фрактальной размерности на предварительном этапе поиска областей и метода морфологического анализа на последующем этапе поиска и уточнения конкретных изменений в найденных на предыдущем этапе областях. Таким образом, показатель фрактальной размерности — это критерий  $K_1$ , а форма изображения — это критерий  $K_2$ .

Применение комбинации двух указанных методов обусловлено следующими причинами:

- необходимостью дополнительных затрат на точное выделение контуров изменений при обработке методом анализа ПФР;
- невозможностью дифференцировать степень изменений при обработке методом морфологического анализа.

Комбинируя эти два метода, мы хотели бы устранить отмеченный недостаток метода анализа ПФР, проводя последующую обработку выделенных областей изображения методом морфологического анализа. В то же время предполагается, что с помощью комбинированного метода будет возможно дифференцировать степень изменений, что обусловлено применением в его составе метода анализа ПФР. При этом полагаем, что созданный комбинированный метод будет нечувствителен к изменениям условий съемки, так как оба применяемых метода обладают указанным свойством.

Таким образом, комбинация этих двух методов позволит компенсировать их недостатки и эффективно применить их достоинства.

### 5. Общая технология поиска

Исходя из сказанного, предлагается следующая схема реализации многоступенчатого комбинированного метода.

1. Осуществляется точное совмещение координат эталонного и текущего изображений для обеспечения их дальнейшей совместной обработки

2. Проводится предварительный поиск областей интереса с целью сузить пространство поиска для сокращения временных и вычислительных затрат. Результатом работы метода предварительного поиска должен быть набор областей текущего изображения (областей интереса), содержащих изменения по сравнению с эталонным изображением. Алгоритм должен осуществлять обработку изображений таким образом, чтобы выявлять лишь изменения топологии изображенной поверхности, игнорируя изменения условий съемки и параметров съемочной аппаратуры. Также пользователю может быть предоставлена возможность устанавливать размеры областей интереса в зависимости от размеров обнаруживаемых аномалий и степени изменений. В рамках рассматриваемого метода при решении данной задачи применяется *метод расчета показателей фрактальной размерности (ПФР) изображений*.

3. Выделяются изменения в найденных областях интереса с целью локализации конкретных изменений в ранее выделенных областях интереса. В результате решения данной задачи оператор должен иметь точную и локализованную информацию о произошедших изменениях. Так же как и на предыдущем шаге, к алгоритму предъявляется требование нечувствительности к изменениям условий съемки и параметров регистрирующей аппаратуры. В рамках рассматриваемого метода при решении данной задачи применяется *метод морфологического анализа изображений*.

#### 6. Краткое описание метода совмещения

В соответствии с предложенным порядком действий при реализации рассматриваемого метода вначале проводится совмещение изображений. Для совмещения выбран известный площадной метод, использующий концепцию опорных точек (ОТ) [6].

Функция преобразования описывает взаимные геометрические соотношения между совмещаемым и базовым изображениями. В качестве такой функции могут использоваться степенные полиномы некоторого порядка, параметры которых оцениваются на основе координат ОТ, например, с использованием МНК.

Оценка координат точки исходного изображения, соответствующей очередному пикселю результирующего изображения, осуществляется с использованием функции трансформирования.

Значение яркости отдельного пикселя результирующего изображения вычисляется на основе значений окружающих найденную точку пикселей исходного изображения с использованием одного из известных методов интерполяции.

## 7. Описание метода анализа фрактальной размерности

Рассмотрим реализацию метода поиска, основанного на вычислении фрактальной размерности. Главной отличительной характеристикой фрактального множества является размерность такого множества. В математике для определения размерности фрактальных образований используют так называемую размерность Хаусдорфа, которая определяется следующим образом:

$$D_f = \lim_{\varepsilon \rightarrow \infty} \frac{\log \Omega_\varepsilon}{\log(1/\varepsilon)}, \quad (2)$$

где  $\varepsilon$  — размер элементов измерения, которыми покрывается объект (например, длина отрезка покрытия, площадь квадрата покрытия, объем куба покрытия, и т. п.);  $\Omega_\varepsilon$  — число элементов измерения размером  $\varepsilon$ , содержащих хотя бы одну точку объекта.

Как оказалось, большое число природных образований — земной рельеф, русла рек, океанские течения, растительный покров — формируются под воздействием сильно нелинейных физических процессов. В результате чего эти природные образования являются фрактальными объектами [7, 8].

Также установлено, что участки поверхности Земли с качественно различной топологией имеют разные фрактальные размерности. Таким образом, изменение оценки ПФР является информативным признаком, свидетельствующем о существенном изменении топологии данной поверхности. В то же время ПФР нечувствителен к аффинным преобразованиям, например к изменению значений пикселей в несколько раз, что обуславливает его инвариантность к изменению условий съемки.

При практической реализации этого метода предлагается использовать в качестве оценки отдельного изображения матрицу ПФР, каждый элемент которой является характеристикой определенного сегмента изображения. При появлении аномалии на изображении ПФР соответствующего сегмента по сравнению с другими показателями изменится существенно в отличие от ПФР всего изображения. Те сегменты, у которых разница ПФР на эталонном и текущем изображениях превышает заданный порог, отбираются для дальнейшей детальной обработки.

Для вычисления фрактальной размерности поверхности предлагается расчет методом вариаций,

предложенный в работе [9]. Данный метод рассматривает изображение как трехмерную поверхность. А именно: пусть  $(x, y)$  — двумерные координаты точки плоскости, а  $z = f(x, y)$  — значение третьей координаты ("высоты" поверхности) в данной точке, которое задано в виде непрерывной функции двух координат. То есть при таком представлении измеренная яркость точки (пикселя) изображения рассматривается как значение высоты трехмерной поверхности. Тогда  $\varepsilon$ -осцилляцией значения  $f$  будет разность наибольшего и наименьшего значений  $f$  в  $\varepsilon$ -окрестности точки  $(x, y)$ :

$$v_f(x, y, \varepsilon) = |f(x_1, y_1) - f(x_2, y_2)|. \quad (3)$$

Очевидно, что данная величина будет нечувствительна к воздействию преобразований яркости, изменяющих яркость всех пикселей изображения, т. е. к изменению условий съемки. После этого вычисляют  $\varepsilon$ -вариацию значения  $f$ :

$$V_f = \int_a^b \int_c^d v_f(x, y, \varepsilon) dx dy. \quad (4)$$

Здесь  $a, b$  — пределы, в которых изменяется переменная  $x$ ;  $c, d$  — пределы, в которых изменяется переменная  $y$ . Фрактальная размерность определяется как

$$D_f = \lim_{\varepsilon \rightarrow \infty} \left( 3 - \frac{\log V_f(\varepsilon)}{\log \varepsilon} \right). \quad (5)$$

Как доказано в работе [9], формула (5) эквивалентна формуле (2).

Практически при использовании данного алгоритма для фрактальной размерности поверхности вычисляется значение  $V_f$  для последовательно уменьшающихся элементов измерения — в данном случае квадратов со стороной  $\varepsilon$ . Графиком функции  $\log V_f(\varepsilon) / \log(\varepsilon)$  в идеальном случае является прямая. Коэффициент наклона этой прямой равен  $(3 - D_f)$ . Для поверхности, как очевидно, значения  $D_f$  лежат в пределах от 2 до 3.

На рис. 1 представлена пара изображений, демонстрирующих результат предварительного поиска областей интереса методом анализа ПФР изображений земной поверхности. Слева показано изображение, сделанное до пожара, а справа — уже в процессе его развития. Светлой рамкой на правом (текущем) изображении выделены те области, в которых зафиксированы изменения по сравнению с левым (эталонным) изображением. Численные значения в рамках являются разностями соответствующих значений ПФР. Как можно видеть, в ходе обработки выделены именно те области изображения, которые содержат признаки деятельности пожара.

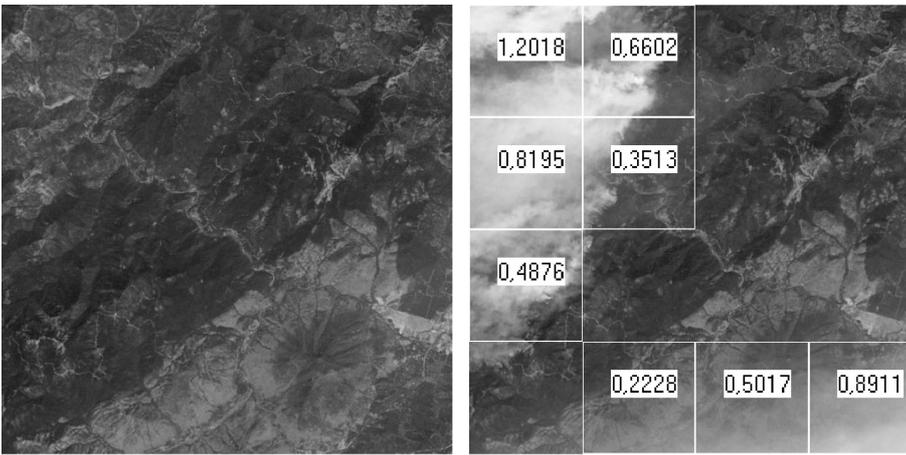


Рис. 1. Изображения территории штата Орегон до и во время пожара

Изображения пожара в штате Орегон были взяты из каталога снимков земной поверхности NASA ([visibleearth.nasa.gov](http://visibleearth.nasa.gov)).

## 8. Описание метода морфологического анализа

Для решения прикладных задач анализа и интерпретации изображений реальных сцен, в которых решение не должно зависеть от условий регистрации изображений, были разработаны методы морфологического анализа изображений, оказавшиеся достаточно эффективными [10, 11].

Изменения параметров устройств формирования изображения или изменившиеся условия освещения и т. п. могут моделироваться путем всевозможных преобразований яркости исходного видеоизображения. Назовем формой ту сущность, которая общая у всех изображений данной сцены, независимо от условий регистрации. Поскольку эта сущность форму сцены полностью не определяет, ее называют формой изображения сцены. Отличия формы одного изображения от формы другого характеризуют свойства сцены, не связанные с условиями регистрации, и могут отражать изменения в содержании изображаемого, например наличие новых объектов или отсутствие старых [10].

Формально каждое изображение можно задать как функцию

$$f(x, y) = \sum_{i=1}^N c_i \chi_i(x, y), \quad (x, y) \in X,$$

определенную на поле зрения  $X$  и принимающую значение  $c_i$  на его подмножестве  $A_i$ . Здесь  $\chi(\cdot)$  — индикаторная функция:

$$\chi_i(x) = \begin{cases} 1, & x \in A_i \\ 0, & x \notin A_i, \quad i = 1 \dots N, \end{cases} \quad (6)$$

причем множества  $A_i$  не пересекаются.

При расчете данным методом мы рассматриваем функцию  $f(x, y)$  как значение яркости пикселя с координатами  $(x, y)$ . Как было сказано выше, мы моделируем условия съемки преобразованием  $F$ :

$$g(x, y) = F(f(x, y)) = \sum_{i=1}^N F(c_i) \chi_i(x, y), \quad (x, y) \in X.$$

Действительно после таких преобразований области поля зрения изображения  $f$ , имеющие одну и ту же яркость  $c$ , на изображении  $g$  будут иметь одинаковую яркость  $F(c)$ , хотя,

возможно, отличную от  $c$ . Тогда зададим форму изображения  $f(x, y)$ , заданного на поле зрения  $X$ , как множество изображений, форма которых не сложнее, чем форма  $f(x, y)$ :

$$V_f = \{f'(x) = F(f(x, y)), \quad (x, y) \in X, F \in \mathbf{F}\} \equiv \{f'(x, y), \quad (x, y) \in X, f'(x, y) < f(x, y)\}.$$

Любое изображение, форма которого не сложнее  $f$ , можно получить подбором яркостей  $c_i$ . Таким образом, построение формы изображения заключается в определении индикаторных функций  $\chi_i$ .

Для построения формы введем оператор проектирования на форму  $P_f$  [12]. Изображение  $P_f \varphi$  — это проекция  $\varphi$  на форму изображения  $f$ . Оно является самым лучшим приближением изображения  $\varphi$  изображениями, форма которых не сложнее, чем форма  $f$ , и вычисляется по следующей формуле:

$$P_f \varphi = \sum_{i=1}^N \frac{(\varphi, \chi_i)}{\|\chi_i\|^2} \chi_i. \quad (7)$$

Таким образом, изображение  $P_f \varphi$  может быть получено усреднением изображения  $\varphi$  по областям  $A_i$  изображения  $f$ . Изображение  $(\varphi - P_f \varphi)$  представляет все то, что отличает  $\varphi$  по форме от  $f$ , т. е. отличия этих двух изображений. При указанном вычитании яркость на областях изображения  $(\varphi - P_f \varphi)$ , не содержащих изменений, будет близка к нулю, а на области с изменениями — отличаться от нуля тем сильнее, чем меньше область изменений и чем больше яркость области изменений отличается от яркостей остальных областей  $A_i$ .

Сравнение областей интереса с помощью морфологической методики позволяет найти и локализовать конкретные изменения в ранее выделенных областях. На рис. 2 представлены результаты дальнейшей обработки вышеупомянутой пары изображений (см. рис. 1) с учетом проведенного

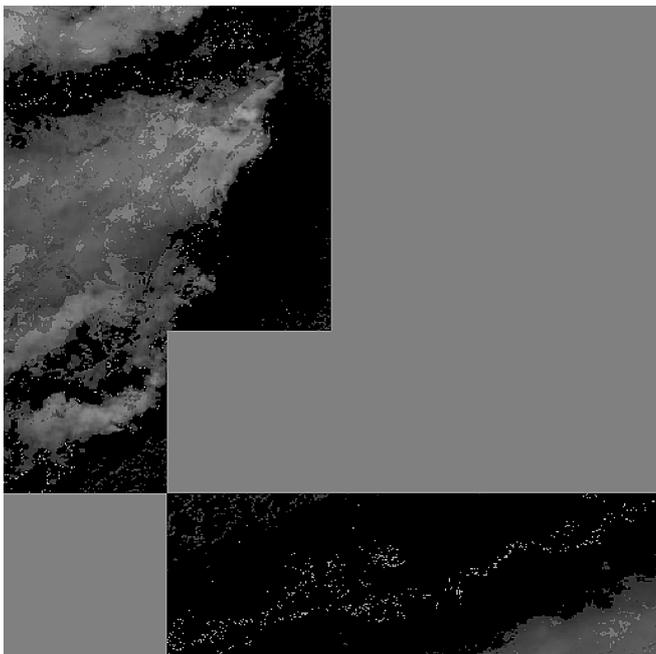


Рис. 2. Результаты дальнейшего поиска и локализации изменений морфологическим алгоритмом

предварительного поиска. Области изображения, признанные неинформативными, закрыты маской. На областях интереса локализованы и выделены произошедшие изменения — признаки деятельности пожара.

## 9. Задача поиска антропогенных ландшафтов

Метод анализа ПФР также можно применять и для выделения областей с антропогенными ландшафтами, используя для анализа всего одно изображение. Антропогенные ландшафты уже не будут фрактальными объектами, так как формируются под воздействием не столь сильно нелинейных процессов, как природные образования. В связи с этим значительная разница между значениями ПФР для областей, содержащих антропогенные ландшафты, и областей, содержащих лишь природные образования, может быть использована при решении задачи обнаружения. Например, необходимость решения такой задачи может возникнуть при анализе кадров, имеющих большое разрешение и содержащих десятки тысяч пикселей.

## 10. Результаты экспериментов

Для оценки эффективности разработанного многоступенчатого комбинированного метода был проведен ряд экспериментов. Наличие достаточно больших вычислительных затрат при реализации этого метода компенсируется высокой эффективностью и устойчивостью результатов

обработки изображений. Это подтверждается методами теории экспертных оценок. Эксперименты по оценке эффективности отбора снимков показывают, что при использовании многоступенчатого комбинированного алгоритма временные и технические затраты, а также затраты экспертных ресурсов на анализ изображений могут быть сокращены более чем в 10 раз.

## Выводы

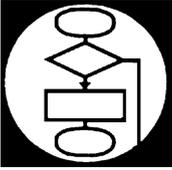
В рамках данной работы:

- проведен анализ задачи поиска областей интереса при оперативной обработке видеоизображений земной поверхности;
- создан комбинированный многоступенчатый метод решения задач поиска изменений и следов антропогенной деятельности;
- проведено тестирование метода на различных типах изображений;
- подтверждена эффективность предлагаемого комбинированного метода методами теории экспертных оценок;
- создано экспериментальное программно-алгоритмическое обеспечение для отработки взаимодействия методов в комплексе и возможного их дальнейшего использования в других приложениях. При разработке данного обеспечения была осуществлена модернизация метода анализа ПФР.

*Работа выполнена при поддержке РФФИ, проект № 05-08-18088-а.*

## Список литературы

1. **Jensen J. R.** Introduction Digital Image Processing. Berlin: Springer-Verlag, 1977. 322 p.
2. **Дуда Л., Харт П.** Распознавание образов и анализ сцен: пер. с англ. М.: Мир, 1984. 382 с.
3. **Прэтт У.** Цифровая обработка изображений: пер. с англ. М.: Мир, 1982. 266 с.
4. **Фукунага К.** Введение в статистическую теорию распознавания образов: пер. с англ. М.: Наука, 1979. 368 с.
5. **Hord R. M.** Remote Sensing: Methods and Applications. N. Y.: Wiley, 1986. 420 p.
6. **Ehlers M.** Rectification and registration / Integration of geographic information systems and remote sensing / Eds. J. L. Star, J. E. Estes, K. C. McGwire. Cambridge: Cambridge University Press, 1997. P. 13–36.
7. **Burrough P. A.** Fractal dimensions of landscapes and other environmental data // Nature. 1981. **294**. N 19. P. 240–242.
8. **Mandelbrot B.** Fractals, Forms, Chance and Pimensions. San Francisco: Freenman, 1977.
9. **Dubuc B., Zucker S. W., Tricot C., Quiniou J. F., Wehbi D.** Evaluating the fractal dimension of surfaces // Proc. R. Soc. Lond. 1989. A 425. P. 113–127.
10. **Пытьев Ю. П.** Морфологический анализ изображений // Докл. АН СССР. 1983. Т. 269. № 5. С. 1061–1064.
11. **Пытьев Ю. П.** Задачи морфологического анализа изображений // Математические методы исследования природных ресурсов Земли из космоса / Под ред. В. Г. Золотухина. М.: Наука, 1984.
12. **Пытьев Ю. П., Чуличков А. И.** ЭВМ анализирует форму изображений. М.: Знание, 1988. 48 с.



УДК 004.432.2

**А. В. Петров,**

Рыбинская государственная авиационная  
технологическая академия,

**В. Б. Гузев,**

Российский Университет дружбы народов,  
г. Москва

## МС# — универсальный язык параллельного программирования

*Рассматриваются основные элементы синтаксиса языка параллельного программирования МС# (Multi-processor C), а также представлены реализации систем исполнения для него под Windows и Linux-платформы*

### Введение

В связи с бурным развитием многоядерных процессоров средства распараллеливания становятся актуальными даже для обычных персональных компьютеров. При этом скоростной потенциал процессоров, как никогда прежде, стал зависеть от качества работы программистов. И никогда раньше успех массового программного обеспечения (ПО) не зависел от его "распараллеливаемости", не говоря уже о ПО, которое должно исполняться на кластерах, метакластерах или GRID-сетях. Стало ясно, что необходимо эффективное расширение синтаксисов языков программирования для встроенной поддержки параллельных конструкций.

Цель данной работы — представить новый язык программирования под названием МС#, расширяющий стандартный синтаксис языка С# для нужд локального и распределенного параллельного программирования.

### Описание языка

Основной целью создания языка МС# является практическое воплощение концепции "Параллельное программирование — это просто!". В связи с этим в качестве базового языка для расширения был выбран язык С#, как один из самых интуитивно понятных. В языке МС# в дополнение к

стандартному синтаксису языка С# были введены лишь четыре новые базовые конструкции:

- порождение нового потока на локальном узле (с помощью асинхронных методов);
- порождение нового потока на удаленном узле (с помощью "перемещаемых" методов);
- обмен объектами между синхронными, асинхронными и перемещаемыми методами (с помощью каналов);
- синхронизация потоков (с помощью связей).

Как оказалось, этих конструкций вполне достаточно для конструирования параллельных распределенных программ любой сложности. Вообще говоря, с помощью данных конструкций можно с легкостью расширить многие другие языки программирования, например, Java, Nemerle и др. Далее мы рассмотрим эти конструкции подробнее.

**Асинхронные методы.** Обычные методы, к которым мы привыкли, программируя на языках, подобных С#, по своей природе синхронны. Это означает, что прежде чем продолжить свои вычисления, вызывающая функция всегда дожидается результата от вызываемой функции. В мире параллельных вычислений ускорение достигается за счет одновременного выполнения нескольких ветвей кода, т. е. неотъемлемым атрибутом любого параллельного языка программирования является возможность запуска функции в отдельном потоке, не дожидаясь возвращения от нее результатов. Это так называемые асинхронные функции.

Для создания асинхронных функций в языке МС# вводится ключевое слово *async*. Для облегчения понимания можно считать, что *async* — это аналог ключевого слова *void*, который в дополнение ко всему придает функции *свойство параллельности*. Ниже приведен пример объявления асинхронной функции с одним параметром на языках С# и МС#:

#### Язык С#:

```
class ObjectWrapper {
    int x;
    public ObjectWrapper(int y)
        {this.x = y; }
    public void afun()
        {Console.WriteLine(x);}
}
...
ObjectWrapper ow = new Object-
Wrapper(5);
Thread t = new Thread
(new ThreadStart(ow.afun));
t.Start();
```

#### Язык МС#:

```
async afun (int x)
{Console.WriteLine(x);}
...
afun(5);
```

**Перемещаемые методы.** Описанные выше асинхронные методы позволяют запускать параллельные ветви вычислений на том же узле, где выполняется поток вызывающей функции. Однако МС# разрабатывался как язык, который в идеале должен позволить программисту полностью абстрагироваться от платформы вычислений — МС#-программы могут выполняться как в локальном режиме, так и в распределенном режиме на кластере, нескольких кластерах (метакластере) или GRID-сетях.

Стоит отметить, что в МС# система исполнения сама решает, какой узел системы лучше всего подходит для выполнения перемещаемой функции. В зависимости от выбранной системы исполнения алгоритмы селекции узлов могут различаться. Для того чтобы система исполнения поняла, что функцию необходимо запускать на другом узле кластера/метакластера/GRID-сети (т. е. чтобы функция стала "перемещаемой"), достаточно пометить ее ключевым словом *movable*. Для облегчения понимания можно считать, что *movable* — это аналог ключевого слова *async*, который в дополнение ко всему придает функции *свойство распределенности*. Отсюда и следствие, что перемещаемые функции не могут возвращать непосредственно никаких значений — взаимодействие между перемещаемыми/асинхронными функциями организуется с помощью так называемых каналов (Channels). Ниже приведен пример объявления и использования перемещаемой функции *mfun*:

```
movable mfun(int x) {
    Console.WriteLine("This code is executed
        on another node! x = " + x);
}
...
mfun (5);
```

При вызове перемещаемой функции переданные объекты в качестве параметров автоматически сериализуются и передаются с одного узла на другой (за исключением каналов, для которых создаются прокси-объекты). Кроме того, копируется и сам оригинальный объект, у которого вызывается перемещаемый метод. При этом все изменения, сделанные перемещаемой функцией на копии объекта на другом узле, не влияют на оригинальный объект.

**Взаимодействие с помощью каналов.** В МС# взаимодействие между параллельными частями программы осуществляется с помощью передачи объектов через специальные каналы (Channels). При этом каналы можно передавать с одного узла на другой сколько угодно раз, а сериализация/десериализация отправляемых объектов (даже сложноустроенных) выполняется в автоматическом режиме. Следующий пример демонстрирует передачу канала *c*, принимающего только строко-

вые значения, в перемещаемую функцию *mfun*, а также передачу значения по каналу с помощью оператора '!':

```
movable mfun(Channel(string) c) {
    c ! ("Hello world!");
}
```

Сами по себе однонаправленные каналы не создаются вручную пользователями — они создаются автоматически при определении связей, о которых будет рассказано ниже. В дополнение к однонаправленным каналам в языке существует класс *BDChannel*, инкапсулирующий логику двунаправленных каналов (bi-directional channels), в которые можно не только удаленно отправлять какие-то значения, но также выполнять и удаленное чтение объектов. В отличие от однонаправленных каналов, двунаправленные каналы не могут участвовать в связках и их нужно инициализировать "вручную".

**Синхронизация потоков вычислений с помощью связей.** Очень часто в параллельных программах необходимо дожидаться результатов работы нескольких параллельно выполняемых функций (возможно перемещаемых) — для этих целей в МС# существуют "связки" (bounds). Связки позволяют объединять в одно целое синхронные методы и каналы с помощью символа '&' и служат средством синхронизации потоков в МС#.

*Общее правило построения связей:* в связке может быть только одно тело связки и в нее может входить не более одного синхронного метода, каналов же может быть несколько.

Тело связки срабатывает тогда и только тогда, когда все методы, входящие в нее, были вызваны (или, соответственно, по всем каналам были отправлены сообщения). Тело связки должно возвращать результат того же типа, что и объявленный в связке синхронный метод. Пример объявления связки:

```
int Get() & Channel result(int x) {
    return x;
}
```

В данном примере в связке находится один канал *result* и один синхронный метод *Get*. Тело связки состоит лишь из одного оператора — возврат целочисленного значения, поступившего по каналу *result*.

При вызове метода *Get* проводится проверка: не приходили ли еще сообщения по каналу *result*. Если сообщения еще не поступали, то текущий поток (вызвавший метод *Get*) блокируется до тех пор, пока в канал *result* не поступит новое сообщение. Когда в канале появится сообщение, срабатывает тело связки (а вместе с тем произойдет и чтение, и извлечение сообщения из канала).

Если же на момент вызова метода *Get* в канале *result* уже находится какое-либо сообщение, то те-

ло связки срабатывает моментально (и проводится чтение и извлечение сообщения из канала).

И наоборот, если при поступлении сообщения по каналу *result* метод *Get* еще не был вызван, то оно ставится в очередь для "последующего использования". Как только метод *Get* будет вызван, сообщение будет братья из этой очереди.

Если же на момент поступления сообщения по каналу *result* метод *Get* уже был вызван, то происходит срабатывание связки (и, соответственно, происходит чтение и извлечение сообщения из канала).

### Примеры программ на языке С#

Рассмотрим задачу перемножения двух матриц *A* и *B* размерности  $N \times N$ . Результирующую матрицу обозначим как *C*. Поскольку вычисления разных элементов матрицы *C* не зависят друг от друга, мы можем осуществить параллельное их вычисление. В нашей программе будем параллельно находить части матрицы *C*, реализуем нахождение части матрицы в процедуре *Multiply*. Результаты вычисления будут пересылаться по каналам в главную процедуру и объединяться в ней для получения конечного результата — матрицы *C*. С предположением, что вышеуказанная процедура должна исполняться на нескольких процессорах, мы можем описать ее как асинхронный метод:

```
public async Multiply(int N, double[,] A, double[,] B,
    double[,] C,
    int from, int to, Channel() sendStop) {
    for (int i = from; i < to; i++)
        for (int j = 0; j < N; j++)
            for (int k = 0; k < N; k++)
                C[i, j] += A[i, k] B[k, j];
    sendStop ! ();
}
```

Главная программа выглядит следующим образом:

```
public class MatrixMultiplierParallel {
    public static void Main(string[] args) {
        int N = System.Convert.ToInt32(args [0]);
        double [,] A = new double [N, N],
        B = new double [N, N],
        C = new double [N, N];
        ReadMatrix (A, B);
        MatrixMultiplierParallel mmp = new MatrixMultiplierParallel();
        int N2 = N/2;
        mmp.Multiply(N, A, B, C, 0, N2, mmp.sendStop);
        mmp.Multiply(N, A, B, C, N2, N, mmp.sendStop);
        for (int i = 0; i < 2; i++)
            mmp.getStop();
        WriteMatrix(C);
    }
    public void getStop() b& Channel sendStop() { return; }
}
```

На рис. 1 изображены графики зависимости времени исполнения от размерности матриц приведенной выше программы и аналогичной программы, реализованной на языке С# (полный исходный текст примера можно найти на сайте проекта [1]).

Рассмотрим теперь решение задачи нахождения числа вхождений слов в текст в рамках распределенной системы. Поиск группы слов будем

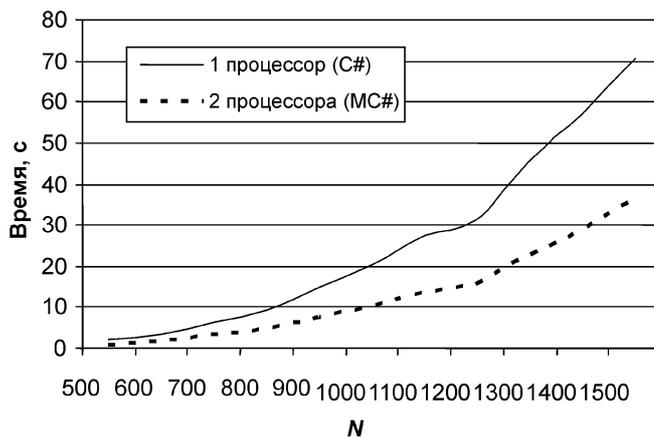


Рис. 1. Перемножение матриц

осуществлять параллельно. В основу решения положим модель организации распределенных вычислений MapReduce [2]. При этом опишем перемещаемый метод Map, в рамках которого будет проводиться считывание текста и поиск заданного слова, и канал Reduce, который будет ответственным за сбор информации от методов Map:

```
public class Program {
    static void Main(string[] args) {
        Program p = new Program();
        //число используемых процессоров в кластере
        int np = CommWorld.Size;
        string[] words = ReadTheWords (args [0]);
        int n = Math.Min(words.Length, np);
        int portion = words.Length / n;
        //отправка порции слов на обработку
        for(int i = 0; i < n; i++) {
            string [] chunkOfWords = new string [portion];
            Array.Copy(words, i*portion, chunkOfWords, 0, portion);
            p. Map(args [1], chunkOfWords, p. Reduce);
        }
        for (int i = 0; i < n; i++)
            p. Pulse();
        PrintTheResult(p. dic);
    }
    movable Map(string fileName, string[] words, Channel(string[], int[]) Reduce) {
        int [] counts = new int [words.Length];
        using (FileStream file =
            new FileStream(fileName, FileMode.Open, FileAccess.Read)) {
            TextReader tr = (TextReader) new StreamReader(file);
            string line = tr. ReadLine();
            while (line != null) {
                for (int j = 0; j < words.Length; j++)
                    counts [j] += Regex.Matches(line, words[j]).Count;
                line = tr. ReadLine();
            }
        }
        Reduce ! (words, counts);
    }
    void Pulse() & Channel Reduce(string[] words, int[] counts){
        for (int i = 0; i < words.Length; i++)
            dic.Add (words [i], counts [i]);
    }
    Hashtable dic = new Hashtable();
}
```

Заметим, что в приведенном примере для его упрощения предполагается, что число обрабатываемых слов и число используемых процессоров кратны степени 2. Необходимо отметить, что модель MapReduce предполагает организацию на

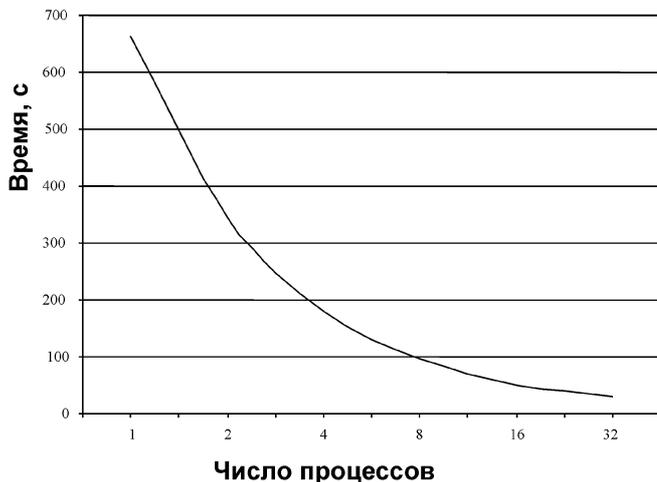


Рис. 2. Поиск 32 слов в тексте объемом 64 Мбайт

кластере распределенной файловой системы, что обеспечивает доступность текста для любого узла кластера и, как следствие, позволяет не пересылать текст "вручную" между главным компьютером кластера и его узлами. На рис. 2 показан график зависимости времени поиска 32 слов в тексте объемом 64 Мбайт от числа используемых процессоров. График получен при тестировании программы на кластере SKIF следующей конфигурации: 16 узлов, каждый из которых содержит по два процессора AMD Athlon(TM) MP 1800+ и по 896 Мб ОЗУ.

## Реализация

На данный момент на сайте проекта доступны две разновидности систем исполнения:

- **MC#.Cluster** — система программирования, позволяющая компилировать и исполнять программы, написанные на языке программирования MC# как в локальном режиме (допустим, на многоядерных машинах), так и в распределенном режиме на кластерах. На данный момент существуют версии для Linux и Windows-кластеров;
- **MC#.MetaCluster** — система программирования, имеющая всю ту же функциональность, что и MC#.Cluster, но в дополнение позволяющая исполнять программы в распределенном режиме одновременно на нескольких кластерах (метакластере). На данный момент существует версия только для метакластеров, основанных на Linux.

## Список литературы

1. **Официальный** сайт проекта MC# — <http://u.pereslavl.ru/~vadim/MCSharp/index.ru.php>
2. **MapReduce**: Simplified Data Processing on Large Clusters, Jeffrey Dean and Sanjay Ghemawat <http://labs.google.com/papers/mapreduce-osdi04.pdf>

УДК 004.4

Синицкий М. Б., Синицкий Б. М.,  
 ЗАО "Телеформ", г. Москва

## Инновационный подход к разработке кросс-платформенных приложений

*Представлен инновационный подход к созданию кросс-платформенных приложений, основанный на использовании специального кодогенератора, позволяющего комплексно нивелировать ограничения традиционных подходов к программированию и повысить эффективность труда программиста путем автоматизации рутинных операций. Верификация подхода проведена на примере создания кросс-платформенного приложения специального назначения.*

Парадигма разработки прикладного программного обеспечения (ПО) с течением времени претерпевает изменения. Современный этап развития подходов и научных взглядов на создание

программных продуктов характеризуется доминированием принципов открытости, масштабируемости, переносимости (кросс-платформенности) и надежности. При этом наряду с большой важностью значительную сложность для разработчиков представляет решение задачи переносимости прикладного ПО. В общем случае кросс-платформенность — это возможность программы работать в различных операционных средах, на различных аппаратных платформах и объединять разные технологические решения. Разработка современного программного обеспечения должна базироваться на принципах и технологиях создания открытых информационных систем, обладающих функциональной гибкостью, реализующих возможности для переноса разработанных приложений на широкий спектр операционных платформ с возможностью совместной работы с другими приложениями на локальных и удаленных системах.

Проблема переносимости ПО приобретает особую актуальность при создании перспективных образцов вооружения и военной техники (ВВТ). Постоянно возрастающие требования вы-

сокой мобильности, живучести, открытости архитектуры систем и отдельных образцов вооружения приводят к необходимости выработки новых подходов к созданию прикладного программного обеспечения специального назначения.

Интерфейс пользователя (то, что в литературе иногда называют "*front-end*") — часть приложения, которая отвечает за взаимодействие с пользователем.

Самым важным принципом разработки кросс-платформенных приложений является четкое разделение интерфейса программы и ее исполняемой части. При этом сохранение возможно большей идентичности интерфейса программы при работе приложения на разных платформах важно как для разработчика в интересах уменьшения трудоемкости его деятельности, так и для конечного пользователя в целях облегчения восприятия визуальной информации.

Традиционно разработка интерфейса пользователя осуществляется программистом без использования средств автоматизации генерации кода или создается в средах программирования с помощью специальных средств — "дизайнеров" [1]. В последнем случае автоматически создается лишь некий каркас кода, который затем придется наполнять необходимым кодом вручную. В любом случае создание соответствующих членов классов и "привязывание" их к элементам пользовательского интерфейса — это кропотливая ручная работа ("ручная вышивка"). В результате этой работы создается код интерфейса пользователя, реализующий соответствующий экранный вид программы и соответствующее ее поведение при воздействии пользователя. Если бы программный код хотя бы только в части обслуживания интерфейса пользователя создавался автоматически, а не вручную, это существенно повысило бы производительность труда программиста и качество создаваемого кода в целом.

Существенным шагом вперед в направлении автоматизации процесса создания интерфейсов является применение кросс-платформенных графических библиотек. Из существующих библиотек наиболее продвинутой считается Qt [2] от фирмы *TrollTech*. Это действительно удобно — один раз написанный код работает на любой платформе и интерфейс во всех случаях выглядит одинаково. Таким образом, существенно облегчается труд программиста. Однако при этом есть серьезные отрицательные моменты. Во-первых, библиотека Qt коммерческая, при создании малобюджетных приложений экономическая эффективность может оказаться крайне низкой, если не отрицательной. Во-вторых, полуручное создание программного кода интерфейса пользователя как было, так и остается. В-третьих, результатом ис-

пользования графической библиотеки является лишь работающий интерфейс пользователя, а о разрешении сопутствующих проблем при этом речи как не было при других подходах, так нет и при этом — "продвинутом". К таким проблемам, условно названным сопутствующими, в частности, относятся:

1. Проблема сериализации классов. Сериализация — средство сохранения содержимого класса в виде образа на диске с возможностью последующего обратного процесса — инициализации создаваемого объекта класса содержимым из этого образа на диске.

2. Проблема представления данных при сохранении их в электронном виде на диске и в твердой копии на бумажных носителях. Необходим способ представления данных, позволяющий удобно работать с этими данными не только программисту, но и предметному специалисту (возможно, человеку, далекому от программирования).

3. Проблема представления данных в файле, позволяющего подключение таких файлов с данными к коду C++ непосредственно, что было бы в ряде случаев весьма удобно при создании программ.

Существующие на сегодняшний день подходы и технологии в области создания программного обеспечения не позволяют решать обозначенные проблемы комплексно. Для решения всех этих проблем авторами разработан инновационный подход создания кросс-платформенных приложений, наследующий достоинства традиционных подходов и нивелирующий их ограничения. Суть подхода заключается в разработке специального кодогенератора, позволяющего по составленному программистом шаблону автоматически генерировать окончательный код интерфейса приложения в текстовом формате. Текстовый формат представления формы является форматом представления данных, кроме того, он служит и для сохранения данных на диске, он же пригоден для оформления бумажных документов и, самое главное, может подключаться непосредственно в код C++. Методы программного кода, которые все это обеспечивают для конкретных классов, генерируются кодогенератором автоматически вместе с кодом, обеспечивающим функционирование интерфейса пользователя.

В качестве демонстрации разработанного авторами инновационного подхода к созданию прикладных программ, обладающих свойством переносимости, был выбран вариант имитационного моделирования траекторных измерений при испытаниях ракетной техники на полигонах МО РФ.

При разработке программы моделирования траекторных измерений (программный комплекс планирования проведения натурального эксперимента "Виртуальный эксперимент", свидетельство о регистрации программ для ЭВМ № 2004611334)

требовалось обеспечить выполнение следующих требований:

- используемый язык программирования — C++;
- исходный текст должен обеспечивать функционирование откомпилированной программы как в среде ОС *Windows*, так и в среде ОС МСВС (*Linux*-подобная система военного назначения);
- разработка каких-либо баз данных не предусматривалась. Формой организации хранения данных выбрана запись в текстовый файл. При этом необходимо обеспечить полную совместимость данных между вариантами программы, работающими на разных платформах.

Важным моментом при выборе варианта решения поставленной задачи явилась сложность создания и отладки программного кода под ОС МСВС из-за несовершенства соответствующих программных средств. С учетом этого было принято решение создавать единый (для обеих платформ) исходный текст на C++ и ориентироваться на то, что основная разработка и отладка программного кода будет производиться под *Windows*, а в среде МСВС будет лишь отлаживаться *Linux* — специфичный код (имеющий отношение только к отображению графики и функционированию интерфейса пользователя). При программировании под *Windows* графическая библиотека Qt не использовалась. Графический интерфейс создавался средствами Win32 API (под МСВС — средствами Qt).

При выборе формата представления данных и выводе их в файл было решено отказаться от применения стандарта XML. Так как предполагалось, что эти файлы должны быть доступны для непосредственного редактирования пользователем, нежелательно нагружать его (пользователя) необходимостью разбираться в тегах гипертекстовой разметки, тем более что использование сложных вложенных структур данных в этих файлах не планировалось. Кроме того, в рамках рассматриваемого проекта разработан и применен особый подход к созданию программного продукта, который позволил найти рациональное решение совокупности взаимообусловленных задач, а именно:

- реализовать идентичность интерфейса пользователя для обеих операционных платформ;
- реализовать удобный для программирования способ организации предметных данных;
- использовать удобный для пользователя (военного оператора) формат хранения наборов данных в файлах;
- осуществлять подключение наборов данных непосредственно в текст программы, написанной на C++;
- обеспечить получение данных в виде текстовых таблиц при выводе их на печать;

- снизить трудоемкость создания и отладки программного кода, который обеспечивал бы решение всех перечисленных выше задач.

Разработанный инновационный подход к эффективному программированию, который позволяет нивелировать проблемные составляющие, обусловленные требованиями кросс-платформенности приложений, и повысить производительность труда программиста, оперирует понятием форма.

Форма — это C++ класс определенной структуры, который реализует конкретный набор возможностей для определенного, логически завершенного набора данных. Используемые при этом наборы данных имеют следующие ограничения по типам и структуре. Данные — члены класса могут быть только следующих типов: **char**, **int**, **double**, **char\***, и являются либо отдельными переменными, либо одномерными массивами элементов перечисленных типов.

Форма может содержать множество массивов различной длины (ограничений на длину массивов нет). Набор массивов одной длины образует таблицу. Форма может содержать до 64 разных таблиц, длина строки каждой из которых может варьироваться от 1 до 64 полей.

Если некий C++ класс не противоречит этим требованиям, то он может быть реализован в виде формы, представленной в исходном тексте на C++ парой файлов — **\*.cpp** и **\*.h**.

При вызове конструктора класса формы в памяти программы создается объект класса формы, представляющий собой хранилище данных, доступных из основной программы по записи и чтению. Кроме этого, **cpp**-файл формы содержит два набора функций: набор методов класса формы и набор глобальных функций, имеющих отношение к этому классу формы. Эти два набора и обеспечивают главную функциональность формы, а именно обеспечивается:

- отображение образа формы в окне при работе приложения (для обозначения образа формы в окне приложения далее будем использовать термин "панель"). Панель состоит из совокупности текстовых таблиц, демонстрирующих данные из объекта формы, преобразованные в текстовый вид. При этом обеспечивается соответствующее цветовое оформление элементов панели, отображаются вертикальные и горизонтальные разделительные линии и иконки. Значения этих таблиц могут функционировать как редактируемые поля (при этом корректность заполнения этих полей пользователем контролируется программно), а также выступать в качестве таких элементов интерфейса, как клавиши, радиокнопки и флажки;
- сохранение данных объекта формы в виде текстового файла (текстовое представление формы).

Получаемый при этом файл можно редактировать с помощью любого текстового редактора, причем в редакторе он визуально представляет собой точную копию соответствующей панели. Привычное расположение полей облегчает чтение и редактирование данных в таком файле;

- загрузка данных объекта формы из соответствующего файла текстового представления формы (с проверкой корректности структуры загружаемого файла). При этом осуществляется также и загрузка иконок, которые должны отображаться на панели формы (из соответствующего набора файлов-иконок для этой формы);
- сохранение данных объекта формы в виде набора текстовых файлов, предназначенных для вывода на печать. При распечатке этих файлов можно получить твердые копии таблиц формы на бумаге формата А4 (эта возможность обеспечивается лишь в случае, если именно для этой цели соответствующим образом подобран формат таблиц данных формы);
- возможность инициализации объекта формы данными из файла текстового представления формы, включенного в состав проекта приложения. Формат текстового представления формы таков, что этот файл может быть подключен

в текст программы на языке С++ непосредственно! Например, реализация этого метода для формы с именем **XX** может выглядеть так:

```
void XX::c_CFinit(...)
{
...
#include "../out/XX_.txt"
...
return;
},
```

где **XX\_.txt** — это файл текстового представления формы **XX**, содержащий вполне конкретный набор данных этой формы. Это дает возможность при необходимости подключить в проект файл — набор данных формы по умолчанию, если требуется, то несколько разных наборов, например, для нескольких различных предусмотренных случаев. При этом все эти наборы данных войдут в состав выполняемого файла приложения.

Продемонстрируем разработанный подход на примере описания экранных форм программного комплекса планирования проведения натурального эксперимента.

Форма "Условия проведения эксперимента" (далее по тексту форма А0) представлена на рис. 1.

А		Условия проведения эксперимента	
	Условия А	Условия проведения эксперимента	
	Измерения В	Полигонный измерительный комплекс	
	Выбор С	Распределение каналов средств ПИК по сопровождаемым целям	
	Районы D	Районы закрытия испытательного полигона	
	Вид E	Настройки режимов отображения	
	Масштаб F	Настройки масштабирования при отображении	
	Синтез G	Настройки генетического алгоритма	
	Данные H	Сохранение варианта схемы на диск и загрузка с диска	
	Трассы A1	Библиотека траекторий	
	Районы A2	Разрешенные / запрещенные для установки средств ПИК	
	Развертка A3	Настройки параметров цветовых разверток траекторий	

Номер	Отображать XZ		Азимут стрельбы XY	Местоположение точки старта				Инф				
	трасса	XY		(град)	(град)	(мин)	(сек)	(град)	(мин)	(сек)	высота (м)	Номер
0	X T_0000	X 5	12.48	55.0	20.0	40.64	37.0	35.0	14.33	0.0	0	T_000
1	X T_0001	X 9	220.75	55.0	36.0	53.08	37.0	47.0	29.58	0.0	1	T_000
2			0.00	0.0	0.0	0.00	0.0	0.0	0.00	0.0	2	
3			0.00	0.0	0.0	0.00	0.0	0.0	0.00	0.0	3	
4			0.00	0.0	0.0	0.00	0.0	0.0	0.00	0.0	4	
5			0.00	0.0	0.0	0.00	0.0	0.0	0.00	0.0	5	
6			0.00	0.0	0.0	0.00	0.0	0.0	0.00	0.0	6	
7			0.00	0.0	0.0	0.00	0.0	0.0	0.00	0.0	7	
8			0.00	0.0	0.0	0.00	0.0	0.0	0.00	0.0	8	
9			0.00	0.0	0.0	0.00	0.0	0.0	0.00	0.0	9	
10			0.00	0.0	0.0	0.00	0.0	0.0	0.00	0.0	10	

Номер	Отображать раз-зону вер-ту		Информация	Длина зоны вдоль оси (м)	Ширина зоны у точки старта (м)	Наклон бок. границ (град)	Смещение азимута (град)	N	ши (град)
	X	Y							
0	X Z_0000	X 5		24060.0	1485.0	4.13	0.00	0	55.0
1	X Z_0001	X 9		16575.0	440.0	6.97	0.00	1	55.0
2	Z_0002			0.0	0.0	0.00	0.00	2	0.0
3	Z_0003			0.0	0.0	0.00	0.00	3	0.0
4	Z_0004			0.0	0.0	0.00	0.00	4	0.0

Рис. 1. Вид панели "Условия проведения эксперимента"

Конкретная форма в проекте может иметь определенные ограничения в функциональности, например, возможны следующие варианты:

- использование иконок при отображении формы в качестве панели не предусмотрено;
- использование формы в качестве панели не предусмотрено;
- использование формы в качестве панели, а также ее сохранение и загрузка из файла не предусмотрены (в этом случае форма используется просто как временное хранилище данных в процессе работы приложения).

Один и тот же файл текстового представления формы может быть использован как в *Windows*-, так и в *MCBC*-реализации программы (с учетом лишь необходимости соответствующего конвертирования ANSI <-> KOI8, обусловленного использованием русского языка). Текстовое представление формы A0 показано на рис. 2 (головная часть панели, представляющая собой меню выбора, в файл текстового представления не выводится).

Два разных проекта, один — под *Windows*, другой — под *MCBC*, организованы так, что они оба "смотрят" на один и тот же (общий) исходный текст, лишь небольшая часть которого, относящаяся к выводу на экран графики и функционированию интерфейса пользователя, имеет две разные реализации — свою для каждой из платформ. Эти фрагменты оформлены директивами препроцессора `#ifdef LINUX` — `#else` — `#endif`, поэтому требуемый вариант сборки всего проекта задается в одном месте и с помощью всего лишь одного `#define`.

В готовом проекте управление программным кодом, реализующим главную функциональность (в нашем случае — обеспечивающим расчеты, связанные с моделированием траекторных измерений и графическую визуализацию результатов

этих расчетов) осуществляется с помощью элементов управления, реализованных в подключенных формах, которые также являются хранилищами данных, доступных для основного вычислительного блока программы.

Написать вручную и отладить программный код форм для рассматриваемого приложения было бы весьма затруднительно (в созданном программном продукте реализованы 23 различные формы и `cpp`-файл для каждой из них содержит около 200 Кбайт программного кода). Тем более что каждое изменение в составе формы в процессе разработки требует внесения многочисленных правок, затрагивающих, как правило, все без исключения реализованные методы формы.

Проблему создания и отладки программного кода удалось решить только благодаря использованию кодогенератора, который был разработан специально для этой цели. При наличии кодогенератора процесс создания программного кода форм выглядит уже следующим образом.

Программист на каждый C++ класс проекта, который намечено реализовать в виде формы, разрабатывает ШАБЛОН — текстовый файл, его имя задает имя класса формы, в котором определяются:

- организация элементов данных в этом классе;
- перечень и количество переменных и массивов — членов класса;
- типы элементов членов класса и способ группировки в таблицах;
- имена элементов данных в тексте программы. Если форма визуализируется в виде панели, то дополнительно определяются:
- цветовое оформление каждого элемента на панели;
- расположение на панели надписи оформления;

```

** * Таблица A000 Точки старта * Таблица A001
** * Отображать Азимут Место положение и точки старта * Таблица A001
** * XZ XY стрельбы широта (град) (мин) (сек) (град) (мин) (сек) высота * И н ф
** *Номер трасса X 5 12.48 55.0 20.0 40.64 37.0 35.0 14.33 0.0 *Номер трасса
** 0 X Z_0000 X 5 12.48 55.0 20.0 40.64 37.0 35.0 14.33 0.0 0 T_0000
** 1 X T_0001 X 9 220.75 55.0 36.0 53.08 37.0 47.0 29.58 0.0 1 T_0001
** 2 0.00 0.0 0.0 0.00 0.0 0.0 0.00 0.0 2
** 3 0.00 0.0 0.0 0.00 0.0 0.0 0.00 0.0 3
** 4 0.00 0.0 0.0 0.00 0.0 0.0 0.00 0.0 4
** 5 0.00 0.0 0.0 0.00 0.0 0.0 0.00 0.0 5
** 6 0.00 0.0 0.0 0.00 0.0 0.0 0.00 0.0 6
** 7 0.00 0.0 0.0 0.00 0.0 0.0 0.00 0.0 7
** 8 0.00 0.0 0.0 0.00 0.0 0.0 0.00 0.0 8
** 9 0.00 0.0 0.0 0.00 0.0 0.0 0.00 0.0 9
** 10 0.00 0.0 0.0 0.00 0.0 0.0 0.00 0.0 10
** ----- *
** F
** * Таблица A010 зоны стрельбы * Таблица A011
** * отображать раз- Длина зоны Ширина зоны у Наклон бок. Смещение *
** * зону вер- вдоль оси точки старта границы азимута *
** *Номер тку информация (м) (м) (град) (град) *
** 0 X Z_0000 X 5 информация 24060.0 1485.0 4.13 0.00 0 55.0
** 1 X Z_0001 X 9 информация 16575.0 440.0 6.97 0.00 1 55.0
** 2 Z_0002 0.0 0.0 0.00 0.00 2 0.0
** 3 Z_0003 0.0 0.0 0.00 0.00 3 0.0
** 4 Z_0004 0.0 0.0 0.00 0.00 4 0.0
** -----
** **

```

Рис. 2. Текстовое представление формы "Условия проведения эксперимента"



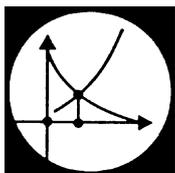
К недостаткам описанного подхода следует отнести то, что реализованный таким образом интерфейс пользователя получается непривычно стилизованным. В рамках данного проекта был реализован лишь самый упрощенный вариант пользовательского интерфейса ввиду естественного стремления избежать сильного усложнения программного кода.

Разработанный инновационный подход к разработке программных продуктов, функционирующих на разных операционных платформах, наиболее эффективен при разработке различных технологических приложений (консольных). При

этом файлы текстового представления форм могут выполнять и функции своеобразного интерфейса пользователя, так как в них сохраняются, а при загрузке файлов и считываются, не только значения элементов данных, но и состояния элементов управления.

#### Список литературы

1. **Хабибуллин И. Ш.** Программирование на языке высокого уровня С/С++. С-Пб. ВHV-Петербург, 2006. 512 с.
2. <http://www.trolltech.com>
3. **Александреску А.** Современное проектирование на С++: Обобщенное программирование и прикладные шаблоны проектирования. Киев. Диалектика, 2004.



## МОДЕЛИРОВАНИЕ И СТРУКТУРНЫЙ СИНТЕЗ

УДК 541.182

**Б. Г. Кухаренко**, канд. физ.-мат. наук,  
Институт машиноведения РАН, г. Москва

### Технология спектрального анализа на основе быстрого преобразования Прони

*Представлена технология быстрого преобразования Прони, использующая матричный алгоритм. В качестве примера анализируются записи нестационарных колебаний с относительно стабильным спектром собственных частот. Показано, что роль информативных параметров, описывающих эволюцию колебания, играет демпфирование на собственных частотах.*

#### Требования к спектральному анализу нестационарных колебаний

Для нестационарных колебаний требуется спектральный анализ, который:

- может без побочных эффектов выполнять спектральное оценивание сегментов временных рядов во временных окнах ограниченной продолжительности;
- использует модель нестационарного временного ряда (например, возрастающего или убывающего во временном окне);
- определяет собственный спектр частот сегмента временного ряда;
- определяет демпфирования на собственных частотах.

С этой точки зрения представляет интерес возможность использования метода Прони (Prony)

для определения зависящих от времени спектров демпфирования нестационарных колебаний [1]. В методе Прони роль априорного знания играет представление о колебательном изменении временного ряда. Однако при идентификации сходства сегментов временного ряда отсутствует ряд ограничений, свойственных Фурье-преобразованию временных рядов. Например, в результате Фурье-преобразования временных рядов, представляющих затухающие синусоиды вида  $Ae^{-\delta t} \times \cos(2\pi ft + \varphi)$ , получают оценки трех параметров:  $A/\delta^{1/2}$  — Фурье-амплитуда,  $\varphi$  — фаза,  $f$  — частота, точность определения которой зависит от величины  $\delta$ . В методе Прони используется разложение сегментов временных рядов, в результате которого для временных зависимостей вида  $Ae^{-\delta t} \times \cos(2\pi ft + \varphi)$  определяются все четыре независимых параметра этих временных зависимостей:  $A$ ,  $\varphi$ ,  $\delta$ ,  $f$ . Так же как и у дискретного преобразования Фурье, главной предпосылкой использования Прони-преобразования временных рядов является линейная модель колебания. Принципиальное отличие состоит в том, что у метода Прони отсутствует непрерывный аналог.

#### Оригинальный метод Прони

Рассмотрим линейную систему, не зависящую от времени

$$\frac{dy(t)}{dt} = Ay(t) + Bu(t), \quad (1)$$

где  $A$  и  $B$  — постоянные матрицы. Предположим, система при некотором импульсном воздействии имеет исходное состояние  $y(t_0) = y_0$ . Если им-

пульсное воздействие прекращается ( $u(t) = 0$ ), то (1) может быть переписано как

$$\frac{dy(t)}{dt} = Ay(t). \quad (2)$$

Пусть матрица  $A$  размерностью  $L \times L$  имеет собственные числа  $\lambda[l]$ , правые собственные векторы  $p[l]$  и левые собственные векторы  $q[l]$ . Решение (2) в виде суммы  $L$  компонент

$$y(t) = \sum_{l=1}^L (q[l]^T x_0) p[l] \exp(\lambda[l]t) \quad (3)$$

представляет сумму затухающих синусоид:

$$y(t) = \sum_{l=1}^M A[l] \exp(\delta[l]t) \cos(2\pi f[l]t + \varphi[l]). \quad (4)$$

В (4) используются следующие обозначения:  $A[l]$  — амплитуда компоненты  $l$ ;  $\delta[l]$  — фактор демпфирования компоненты  $l$ ;  $f[l]$  — частота компоненты  $l$ ;  $\varphi[l]$  — фаза компоненты  $l$ ;  $L = 2M$  — полное число затухающих компонент. Используя формулу Эйлера,  $\cos(2\pi f[l]t + \varphi[l])$  можно представить в виде суммы экспонент:

$$\cos(2\pi f[l]t + \varphi[l]) = \exp(j2\pi f[l]t + \varphi[l]) \times \exp(-j2\pi f[l]t - \varphi[l]) / 2. \quad (5)$$

Подставим (5) в (4), и пусть  $t[k] = k\Delta t$ ,  $\Delta t$  — интервал дискретизации времени. Тогда временной ряд для  $y(t)$ , состоящий из  $N$  образцов  $y[k] = y(t[k])$ ,  $t[k] = k\Delta t$ ,  $k = \overline{0, (N-1)}$ , может быть переписан как

$$y[k] = \sum_{l=1}^L r[l] z[l]^k, \quad (6)$$

где  $r[l] = (A[l]/2) \exp(j\varphi[l])$  и  $z[l] = \exp((\delta[l] + j2\pi f[l])\Delta t)$  называются, соответственно, вычетами и полюсами сегмента временного ряда, поскольку формула (6) похожа на обратное Z-преобразование временного ряда. Действительно, метод Прони связан с Z-преобразованием временных рядов [2].

В оригинальном методе Прони для  $y[\overline{1, N}]$ ,  $N \ll N_0$ , представляющего собой некоторый сегмент полного временного ряда  $y[\overline{1, N_0}]$ , определяется линейная модель предсказания

$$y[k] = -a[1]y[k-1] - a[2]y[k-2] - \dots - a[L]y[k-L].$$

Вычеты  $z[l]$  определяются как корни полинома с коэффициентами  $\{a[l], l = \overline{0, L}\}$

$$\sum_{i=0}^L a[L-i]z^i = \prod_{i=1}^L [z - z[i]], \quad a[0] = 1.$$

Присутствие в алгоритме шага, связанного с определением корней полинома, приводит к

сильной зависимости времени обработки от числа определяемых полюсов и низкой вычислительной устойчивости оригинального метода Прони [1].

### Быстрое преобразование Прони на основе матричного алгоритма

Линейная модель предсказания для сегмента  $y[\overline{1, N}]$ ,  $N \ll N_0$ ,

$$y[k] = -a[1]y[k-1] - a[2] \times y[k-2] - \dots - a[L]y[k-L]$$

в матричной форме имеет вид

$$y = Ya,$$

где

$$Y = \begin{bmatrix} y[L-1] & y[L-2] & \dots & y[0] \\ y[L] & y[L-1] & \dots & y[1] \\ \dots & \dots & \dots & \dots \\ y[N-2] & y[N-3] & \dots & y[N-L-1] \end{bmatrix}.$$

Оказывается, что матрица  $Y$  может быть использована для непосредственного определения полюсов сегмента временного ряда. В матричной версии метода Прони матрица Ганкеля  $Y$  размерностью  $(N-L) \times (L+1)$  формируется как

$$Y[I, \overline{1, (L+1)}] = y[\overline{I, (I+L)}], \quad I = \overline{1, (N-L)}, \quad (7)$$

где  $L$  — параметр декомпозиции Прони. Показано, что  $z = z[l]$ ,  $l = \overline{1, p}$ ,  $p < L < N-p$  понижает ранг матрицы

$$(Y^{(1)} - zY^{(0)}), \quad (8)$$

где

$$Y^{(0)} = Y[\overline{1, (N-L)}, \overline{1, L}], \quad Y^{(1)} = Y[\overline{1, (N-L)}, \overline{2, (L+1)}]. \quad (9)$$

Поскольку число полюсов временного ряда, определяемых на основе свойства понижения ранга матричной формы (8), не может превышать  $N/2$ , то в (9) выбирается значение  $L = N/2$ . Специальный критерий позволяет установить число  $p$  главных полюсов временного ряда. Этот критерий использует сингулярное разложение матриц (9), а именно:

$$Y^{(v)} = U^{(v)} S^{(v)} (V^{(v)})^T, \quad v = \overline{0, 1} \quad (10)$$

где

$$S^{(v)} = \text{diag}\{s^{(v)}[1], s^{(v)}[2], \dots\}, \quad s^{(v)}[1] > s^{(v)}[2] > \dots > 0; \quad (11)$$

$$(U^{(v)})^T U^{(v)} = (V^{(v)})^T V^{(v)} = I, \quad (12)$$

а верхний индекс "т" обозначает транспонирование. Число  $p$  главных полюсов временного ряда

$y[\overline{1}, \overline{N}]$  определяется числом главных сингулярных чисел в (11) по минимуму конечных разностей соседних элементов массивов  $\{s^{(v)}[1], s^{(v)}[2], \dots\}$ ,  $v = \overline{0}, \overline{1}$ , или по минимуму относительных конечных разностей соседних элементов этих массивов, если первый критерий оказывается недостаточно чувствительным.

Определим усеченные матрицы

$$S_{tr}^{(v)} = \text{diag}\{s^{(v)}[1], \dots, s^{(v)}[p], \dots, 0\}, v = \overline{0}, \overline{1}. \quad (13)$$

Использование в (10) матриц  $S_{tr}^{(v)}$ ,  $v = \overline{0}, \overline{1}$ , позволяет определить усеченные матрицы  $Y_{tr}^{(v)}$ ,  $v = \overline{0}, \overline{1}$ . Нахождение полюсов  $z[l]$ ,  $l = \overline{1}, p$ , по формуле (8) с использованием  $Y_{tr}^{(v)}$ ,  $v = \overline{0}, \overline{1}$ , гарантирует вычислительную стабильность определения этих полюсов в пределе временного ряда без шума, поскольку матрицы  $Y_{tr}^{(v)}$ ,  $v = \overline{0}, \overline{1}$ , имеют фиксированный ранг  $p$ . Метод оценки точности определения полюсов по формулам (8), (9) представлен в [3].

#### Определение вычетов по методу наименьших квадратов

После определения полюсов  $z[l]$ ,  $l = \overline{1}, p$ , вычеты  $r[l]$ ,  $l = \overline{1}, p$ , в этих полюсах определяются по методу наименьших квадратов. После определения полюсов  $z[l]$ ,  $l = \overline{1}, p$ , декомпозиция (6) представляет собой систему линейных уравнений для определения их вычетов  $r[l]$ ,  $l = \overline{1}, p$ . Поскольку  $N > 2p$ , система уравнений (6) переопределена, квазирешение этой системы находят по минимуму средней квадратической ошибки декомпозиции. Тогда, используя матрицу  $Z$ ,  $i$ -я строка которой состоит из полюсов  $z[l]$ ,  $l = \overline{1}, p$ , в  $i$ -й степени, квазирешение уравнений (6) можно записать:

$$r = (Z^+ Z)^{-1} Z^+ y, \quad (14)$$

где символ "+" обозначает Эрмитово сопряжение.

Ошибка  $\varepsilon$  замыкания прямого Прони-преобразования временного ряда

$$y[\overline{1}, \overline{N}] \rightarrow \{z[l], r[l], l = \overline{1}, p\}$$

по формулам (8), (9) и (14), и обратного Прони-преобразования

$$\{z[l], r[l], l = \overline{1}, p\} \rightarrow y[\overline{1}, \overline{N}]$$

по формуле (6) определяется выражением

$$\varepsilon = [(y^+ - r^+ Z^+) (y - Zr)]^{1/2}. \quad (15)$$

Полюса  $z[l] = \exp(\delta[l] + j2\pi f[l])$ ,  $l = \overline{1}, p$ , сегмента  $y[\overline{1}, \overline{N}]$  временного ряда определяют дискретные спектры демпфирования и частот  $\{\delta[l], f[l], l = \overline{1}, p\}$ , а вычеты  $r[l] = A[l] \exp(j\phi[l])$ ,  $l = \overline{1}, p$ , в этих полюсах — соответствующие дискретные спектры амплитуд и фаз  $\{A[l], \phi[l], l = \overline{1}, p\}$ , определенные для временного окна  $t \in [0, t_N]$ . Оценка временной зависимости дискретных спектров демпфирования и частот  $\{\delta[l], f[l], l = \overline{1}, p\}$  и соответствующих им дискретных спектров амплитуд и фаз  $\{A[l], \phi[l], l = \overline{1}, p\}$  осуществляется в результате последовательного сдвига временного окна фиксированной длины  $N\Delta t$ . Восстановление колебания по линейной модели с использованием спектральных параметров, полученных по методу Прони, позволяет оценить качество предлагаемой технологии спектрального анализа.

#### Оценка энергии частотных составляющих

Для оценки энергии частотных компонент предполагается, что у всех частотных компонент сегмента  $y[\overline{1}, \overline{N}]$  их временные зависимости затухают симметрично относительно начала отсчета времени [1]. Тогда спектр мощности

$$S(\omega) = \frac{2}{\pi} \left| \sum_{l=1}^M \left\{ \frac{-A[l] \delta[l] \exp(j\phi[l])}{\delta[l]^2 + (\omega[l] - \omega)^2} + c.c. \right\} \right|^2,$$

где  $\omega[l] = 2\pi f[l]$ ,  $v = \overline{0}, \overline{1}$  — круговые частоты полюсов;  $c.c.$  — комплексно сопряженное выражение. Оценка для полной мощности

$$\begin{aligned} & \sum_{k=1}^N y[k]^2 = \\ & = \frac{2}{\pi} \int_{-\infty}^{+\infty} d\omega \left| \sum_{l=1}^M \left\{ \frac{-A[l] \delta[l] \exp(j\phi[l])}{\delta[l]^2 + (\omega[l] - \omega)^2} + c.c. \right\} \right|^2 \approx \\ & \approx \sum_{l=1}^M \left( \frac{A[l]}{\sqrt{\delta[l]}} \right)^2 \frac{2}{\pi} \int_{-\infty}^{+\infty} \frac{\delta[l]^3 d\omega}{[\delta[l]^2 + (\omega[l] - \omega)^2]^2}, \quad (16) \end{aligned}$$

так как фазы различных частотных компонент считаются некоррелированными. В формуле (16)

$$\frac{2}{\pi} \int_{-\infty}^{+\infty} \frac{\delta[l]^3 d\omega}{[\delta[l]^2 + (\omega[l] - \omega)^2]^2} = 1,$$

поэтому аналог теоремы Парсеваля для разложения Прони сегмента  $y[\overline{1}, \overline{N}]$  имеет вид

$$\sum_{k=1}^N y[k]^2 \approx \sum_{l=1}^M \left( \frac{A[l]}{\sqrt{\delta[l]}} \right)^2.$$

## Сравнение преобразования Прони с дискретным преобразованием Фурье

Метод Прони позволяет определить для сегмента временного ряда  $y[1, N]$  его полюса  $z[l]$ ,  $l = \overline{1, p}$ , а на их основе — спектры частот и демпфирования этого сегмента временного ряда

$$f[l] = \frac{1}{2\pi\Delta t} \text{Im}(\log(z[l])) \text{ и } \delta[l] = \frac{1}{\Delta t} \text{Re}(\log(z[l])).$$

Если требуется, то после определения дискретных спектров частот и демпфирования временного ряда по методу Прони можно по формуле (14) определить спектры амплитуд и фаз этого временного ряда. Таким образом, все в точности противоположно тому, в какой последовательности спектральные свойства временного ряда оцениваются по методу Фурье. Дискретное преобразование Фурье сегмента временного ряда  $y[1, N]$  задает  $N/2$  пар комплексно сопряженных формальных частот  $\{f[k], k = \overline{1, N}\} = \{1/N\Delta t, 1/N\Delta t, 1/2\Delta t\}$ , которых определяется результат Фурье преобразования  $\{X[k], k = \overline{1, N}\}$ , и

$$A[k]^{def} = |X[k]|; \varphi[k]^{def} = \arg(X[k]).$$

Спектр собственных частот временного ряда определяется в результате нахождения точек  $k_1, k_2, \dots$  локальных максимумов дискретной функции  $\{A[k], k = \overline{1, N}\}$ , т. е. частоты  $f[k_1], f[k_2], \dots$  считаются собственными частотами временного ряда  $y[1, N]$ .

### Пример

Примером систем, которые совершают нестационарные колебания под воздействием аэроупругих сил, являются турбинные лопатки. Современные центробежные компрессоры имеют монолитную структуру, их лопатки имеют сложную трехмерную геометрию и близко расположены друг к другу, как это показано на рис. 1.

При увеличении числа оборотов компрессора появление флаттера связано с аэроупругими коллективными изгибно-крутильными колебаниями турбинных лопаток.

**Особенности нестационарных колебаний турбинных лопаток.** В настоящей статье анализируются записи изменения напряжений (тензометрические данные) при нестационарных колебаниях лопаток компрессора в процессе наступления флаттера. Как показано на рис. 2, эти колебания характеризуются неустойчивостью. Соответствующая зависимость от времени частоты оборотов турбины компрессора приведена на рис. 3.

Фурье-анализ этих записей показывает, что они характеризуются относительно стабильным спектром частот. Частота изгибной моды, которая доминирует в записи колебаний на рис. 2, а, равна

$\approx 81$  Гц, частота крутильной моды, которая доминирует в записи на рис. 2, б, равна  $\approx 197$  Гц. Однако, как видно на рис. 2, роль этих составляющих колебания меняется со временем, и это приводит к существенно нестационарному характеру колебаний лопатки.

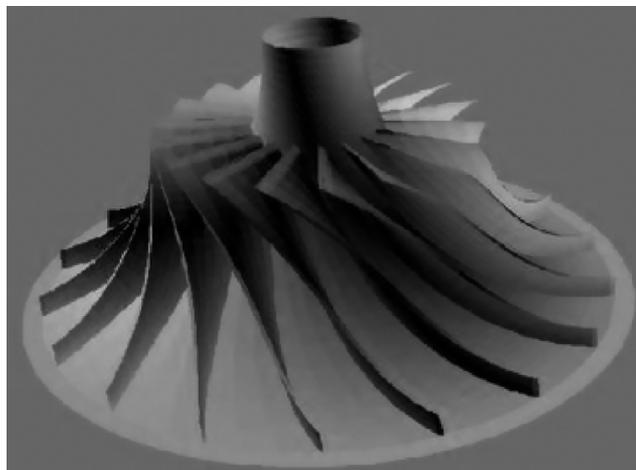


Рис. 1. Турбина центробежного компрессора

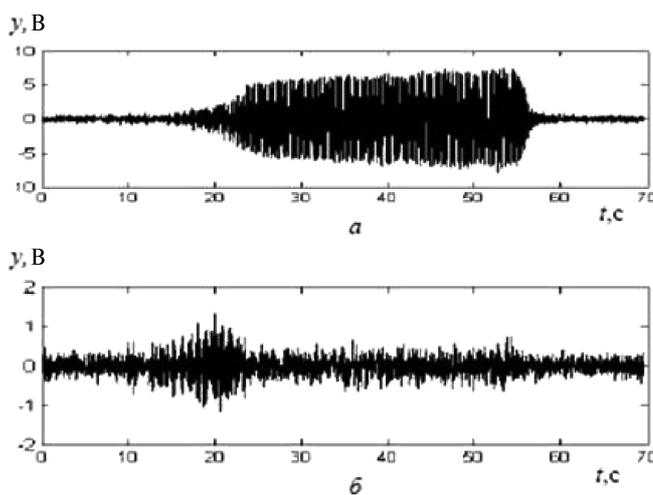


Рис. 2. Записи колебаний в локальных максимумах изгибной (а) и крутильной (б) мод флаттера

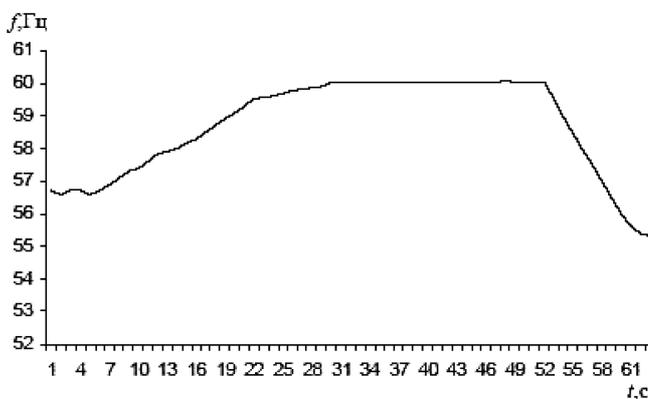


Рис. 3. Зависимость от времени частоты оборотов турбины компрессора

**Постановка задачи.** Задача состоит в определении особенностей временной эволюции флаттера и установлении информативных параметров, определяющих эту эволюцию. Для рассматриваемых нестационарных колебаний эта задача не может быть решена в результате анализа колебания непосредственно во временной области, поскольку характерное изменение огибающей амплитуды колебания, которое хорошо видно на рис. 2, существует только в достаточно крупном масштабе времени. В более мелких масштабах времени эта огибающая амплитуды колебаний испытывает биения и вообще ее изменение оказывается существенно не монотонным. Поэтому для определения особенностей временной эволюции флаттера следует использовать спектральное оценивание записей колебания. В [4] показано, что флаттер турбинных лопаток описывается в рамках линейной модели. Для сегмента временного ряда, представляющего собой сумму затухающих синусоид с произвольным соотношением между их частотами, метод Прони дает точный результат при оценке дискретных спектров частот, демпфирования, фаз и амплитуд. Отмеченная выше неизменность спектра собственных частот является важным критерием постоянства физических свойств и целостности распределенной системы. Для таких распределенных систем важную информацию об изменениях, происходящих в системе, могут предоставлять временные зависимости демпфирования на собственных частотах системы. Собственные частоты нестационарных колебаний и их локальные во времени демпфирования могут быть оценены по методу Прони.

### Результаты

Результаты определения спектров собственных частот и временного изменения демпфирования турбинной лопатки по записям нестационарных колебаний на рис. 2 позволяют охарактеризовать этот многофазный процесс. Временная зависимость  $\delta = \delta(t)$  демпфирования изгибной (*a*) и крутильной (*b*) моды колебания, приведенного на рис. 2, показана на рис. 4.

Как показывает спектральный анализ по методу Прони, вплоть до  $t \approx 25$  с спектр колебаний на рис. 2 содержит только собственные частоты лопатки и гармоники оборотной частоты, т. е. колебания являются линейными. С ростом оборотной частоты турбины (рис. 4) при  $t = 10$  с в переход-

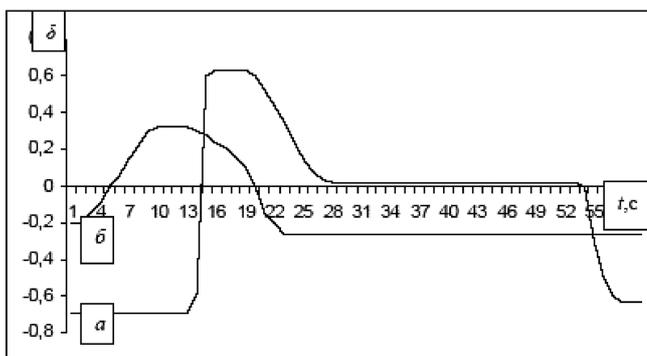


Рис. 4. Временная зависимость демпфирования  $\delta = \delta(t)$  изгибной (*a*) и крутильной (*b*) мод колебаний, приведенных на рис. 2

ном процессе появляется биение с двумя производящими частотами порядка частоты крутильной моды колебаний. При этом демпфирование на более низкой частоте изгибной моды отрицательно до  $t = 15$  с, после чего демпфирование на изгибной частоте становится положительным, и ее амплитуда начинает резко возрастать. Изменение знака и величины демпфирования на собственной частоте означает изменение роли этой частотной составляющей нестационарного колебания и позволяет предсказывать его неустойчивость. При  $t \approx 25$  с в спектре нестационарных колебаний появляются гармоники изгибной частоты. Энергия изгибной составляющей колебания примерно в 300 раз превышает энергию крутильной составляющей. В линейной модели колебаний появление гармоник собственных частот показывает, что формы с этими собственными частотами становятся нелинейными [5]. Таким образом, начиная с  $t \approx 25$  с, флаттер развивается как чисто изгибный и нелинейный.

### Список литературы

1. Марпл С. Л. Цифровой спектральный анализ и его приложения. М.: Мир, 1990. 463 с.
2. Weiss L., McDonogh R. N. Prony's method, Z-transform, and Pade approximation // SIAM Review. 1963. V. 9. N 2. P. 145—149.
3. Kukhareenko B. G. Use of the Prony method for modal identification of slow-evolutionary linear structures // Journal of Structural Control. 2000. V. 7. N 2. P. 203—218.
4. Bendiksen O. O., Friedmann P. P. The effect of bending-torsion coupling on fan and compressor blade flutter // Transactions of the ASME. 1982. V. 104. N 3. P. 617—623.
5. Фролов К. В., Балакшин О. Б., Кухаренко Б. Г., Минаев А. Я. Спектральный критерий и оценка нелинейности колебаний систем // Проблемы машиностроения и надежности машин. 2001. № 6. С. 3—7.

Л. С. Виленчик, д-р техн. наук, проф.,  
Ю. В. Иванов, д-р техн. наук,  
В. П. Трофимов, канд. техн. наук,  
ФГУП МКБ "Электрон"

## О математическом моделировании динамики естественных природных и физических систем дискретно-непрерывными системами с запаздывающим аргументом

*Исследованы динамические режимы автоколебаний цифровой модели системы с нелинейными и параметрическими свойствами. Динамика модели представлена 3D-картами режимов значений спектральной плотности мощности. Динамику топологии карт количественно оценивали по лакунарности и по топотезе. Совпадение полученных величин с подобными оценками топологии изображений природных и физических объектов говорит об эффективности моделирования и выбранного метода оценивания.*

Многие естественные природные и физические процессы со сложной динамикой содержат каналы, входящие в состав некоторой ветвящейся фрактальной структуры. Они описываются уравнениями в дробных производных, причем показатель дробного интеграла соответствует доле каналов — ветвей, открытых для перетока информации. Такие системы называют системами с остаточной памятью [1]. Фрактальная геометрия систем с остаточной памятью и ее корреляции с операторами дробного интегрирования изучались в [2-4].

Физическая интерпретация прямой связи между дробным интегралом и каноническим фрактальным множеством Кантора опирается на два предельных случая (доказывается в [2]).

Если процесс эволюции некоторой физической системы описывается оператором

$$W(t) = \int_0^{\tau} k(t - \tau)\varphi(\tau)d\tau,$$

где  $k(t - \tau)$  — импульсная переходная функция системы;  $\varphi(\tau)$  — начальная функция, зависящая от  $\tau$ , то в первом предельном случае функция памяти системы имеет ступенчатый характер, во втором — система теряет все свои состояния, за исключением одного — эквивалента точки. Второй случай

соответствует марковскому процессу, т. е. процессу без последствия.

Если задаться вопросом: могут ли существовать физические системы, в которых динамика развития занимает промежуточное положение между прямой и точкой, то классическая геометрия ответа на него дать не может, так как в ней не существует промежуточного образа. Фрактальная геометрия отвечает утвердительно — таким образом, например, является канторово множество [5].

Таким образом, можно говорить, что динамика естественных физических систем — фрактальна и, следовательно, может описываться в терминах фрактальной геометрии и анализироваться в единицах фрактальных размерностей.

В настоящее время существует достаточно примеров применения фрактальных размерностей для оценивания состояний параметров окружающей среды, микронеровностей поверхностей при технологических обработках, радиолокационных изображений и многих, многих других [6—8]. Важными особенностями динамики таких систем является свойство масштабной инвариантности, или скейлинга, они обладают памятью, однако это не кратковременная марковская память, кроме того, они персистентны, т. е. сохраняют имеющуюся тенденцию.

Традиционным методом анализа случайных сигналов таких систем является спектрально-корреляционная теория, опирающаяся на фундаментальную теорему Винера—Хинчина. Полное статистическое описание случайных сигналов, если не принимать во внимание гауссовы процессы, требует вычислений моментов высших порядков, что приводит в итоге к возрастанию сложности, росту объема вычислений и потере точности и поэтому не всегда продуктивно. В то же самое время оценки фрактальных размерностей различных геометрических образов и изображений, характеризующих динамику и эволюцию систем, достоверно и адекватно отображают превращения, происходящие в системе.

Таким образом, в настоящее время сложился новый аппарат — аппарат фрактального анализа динамических систем, который, очевидно, может применяться также для оценивания процессов и сигналов их математических интерпретаций при математическом моделировании. Однако остается открытым вопрос: какими (и почему) должны быть математические модели таких систем?

В данной статье рассматривается класс систем таких моделей — системы с запаздывающим аргументом (СЗА), приводится один из обобщающих видов модели — автономной нелинейной параметрической системы с запаздыванием и с линейным полосовым фильтром второго порядка, как один частный случай систем с запаздываю-

шим аргументом, изучаются динамические режимы автоколебаний модели при изменении значений определяющих параметров и применяются фрактальные меры оценивания динамики режимов, представленных 3D-картами режимов. Карты динамических режимов (КДР) рассматриваются как фрактальные геометрические пространственные образы, поэтому применяются такие же алгоритмы их фрактального анализа, как и обычно при фрактальном анализе поверхностей естественных природных и технических систем. Порядки числовых значений оценок сопоставляются.

В теории и практике систем автоматического управления системы с постоянным запаздыванием и с постоянными параметрами описаны подробно. Однако системы с переменными параметрами, в том числе с переменным запаздыванием, — сравнительно недавняя область исследований, предмет которой пока очерчен не полностью. Большинство задач исследования систем с переменным запаздыванием не следует из теории систем с постоянным запаздыванием, а математический аппарат — теория уравнений, совершенно отличен.

Среди относительно небольшого числа работ, касающихся теории систем с переменным запаздыванием, уместно выделить две из них — [9, 10]. В первой наиболее полно отражена проблематика этих систем, а во второй излагаются, главным образом в линейной постановке, общие свойства систем, рассматриваются детерминированные и стохастические процессы и вопросы моделирования на ЭВМ, области их возможного применения, в частности в дальней космической связи.

Системы с запаздыванием обладают рядом особенных свойств, отличающие протекающие в них процессы от обычных представлений. В первую очередь, это вызвано тем, что вид переходной функции состояния системы зависит не от ее начального значения, как это бывает в обычной динамической системе с возмущениями, а от некоторой функции — начальной реакции состояния, которая задается на отрезке времени, предшествующем началу динамического процесса. Таким образом, если задана система с постоянным запаздыванием  $\tau$  с ее переходной функцией состояния  $K(t - \tau)$ , то выходной процесс такой системы определяется не только входным воздействием  $U(t)$ , но и ее некоторой начальной функцией  $\varphi(\tau)$ . Именно необходимость задания последней и определяет принципиальное отличие системы с запаздыванием (с памятью) от обычной динамической системы.

В простейшем случае система с постоянным запаздыванием описывается дифференциальным уравнением

$$\frac{dx}{dt} = f[\omega(t), t, x(t), x(t - \tau)], \quad (1)$$

причем на отрезке  $t_0 - \vartheta \leq t \leq t_0$  выполняется условие  $x(t) = \varphi(t, \vartheta)$ .

Корректные аналитические методы решения уравнения (1) весьма затруднительны из-за необходимости введения дополнительных условий и ограничений. Поэтому наиболее эффективным является путь, основанный на представлении уравнения (1) в форме цифровых или аналоговых моделей, причем параметры моделей могут выбираться и назначаться исходя из подобия реальным физическим или естественным природным процессам. В этом случае решениями уравнения будут временные формы колебаний, параметры которых могут изменяться с течением времени и образовывать ветвления, слипания, провалы решений и даже срывы решений. Отметим, что устойчивые решения обеспечиваются, по меньшей мере, тогда, когда произведение коэффициентов усиления в прямой и обратной цепях больше единицы ( $K_{yc} \cdot K_{oc} > 1$ ), а сумма фаз равна или кратна  $\varphi_{yc} + \varphi_{oc} = 2\pi\omega$ , где  $\omega$  — основная частота.

В данной статье исследовались динамические режимы системы с запаздывающим аргументом, состоящей из нелинейного усилителя, линейного инерционного блока с постоянным запаздыванием, нелинейного блока с вольт-амперной характеристикой туннельного диода, на параметрический вход которого от специального генератора подавался сигнал накачки, кратный основной частоте системы, и линейного идеального полосового фильтра второго порядка (рис. 1).

Идеальный полосовой фильтр второго порядка был реализован в форме цифрового консервативного осциллятора с бесконечно импульсной характеристикой (БИХ) вида  $h(t) = e^{-\beta t} \cos \omega_0 t$ , где  $\beta$  — коэффициент затухания. У такого фильтра выходной сигнал строго соответствует лишь одной гармонической составляющей частоты резонанса полосового фильтра  $\omega_0$ , а полоса частот на уровне полумощности зависит от коэффициента затухания  $\beta \geq 0$ . Достоинством данного решения является то, что основная частота  $\omega_0$  может выби-

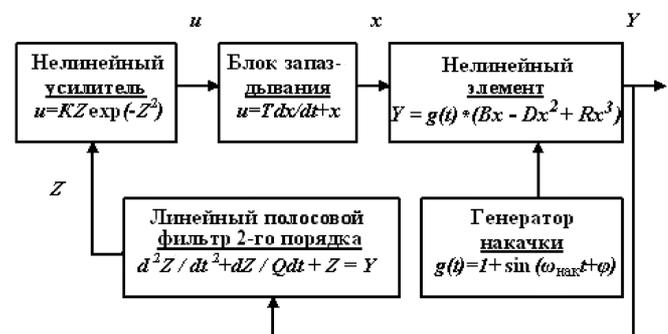


Рис. 1. Структурная схема параметрической системы с запаздывающим аргументом

раться произвольно. Идеальному фильтру соответствует значение  $\beta = 0$ .

Нелинейный элемент с вольт-амперной характеристикой туннельного диода был задан кубической параболой вида

$$F(x) = Bx - Dx^2 + Rx^3,$$

где  $B, D, R$  — коэффициенты. Вольт-амперная характеристика туннельного диода имеет вид  $N$ -образной кривой и содержит падающий участок. Нелинейная характеристика туннельного диода в данной системе является частотно-независимой.

Роль параметрического возмущения в данной автономной системе, т. е. в системе без силового возмущения по входу, заключалась в том, что в результате периодического от времени изменения частоты накачки на входе нелинейного элемента достигался эффект дополнения отображения. Отметим, что трансформациям естественных природных систем свойственна именно параметрическая динамика.

Данная нелинейная динамическая система была реализована в форме цифровой математической модели, которая была настроена на режим автоколебаний. Автоколебания естественных динамических систем являются основной формой их существования и функционирования.

Уравнения динамики параметрической СЗА описываются в форме системы уравнений

$$\left. \begin{aligned} \ddot{Z} + \frac{\dot{Z}}{Q} + Z &= Y; \\ T\dot{x} + x &= KZ \exp(-Z^2); \\ Y &= g(t) \times (Bx - Dx^2 + Rx^3); \\ g(t) &= 1 + \sin(\omega_{\text{нак}} t + \varphi), \end{aligned} \right\}$$

в которой  $x$  — сигнал на выходе инерционного элемента;  $Z$  — сигнал на входе нелинейного усилителя. Параметры  $T$  и  $Q$  соответственно определяют запаздывание инерционного элемента и добротность полосового фильтра, а  $K$  — коэффициенты усиления усилителя. В данной схеме БИХ-фильтра  $\beta = 0$ ,  $Q = 60$ . Параметры  $T$  и  $K$  определяют форму отображения. В данном исследовании параметры  $T$  и  $K$  изменялись в пределах от 1 до 12. Коэффициенты в уравнении нелинейного элемента соответствовали значениям  $B = 8,592$ ;  $D = 22$ ;  $R = 14,408$ . Накачка осуществлялась по гармоническому закону  $g(t) = 1 + \sin(\omega_{\text{нак}} t + \varphi)$  с частотой  $\omega_{\text{нак}} = f(\omega_0)$  и сдвигом по фазе  $\varphi$ , где  $\omega_0$  — резонансная частота БИХ-фильтра.

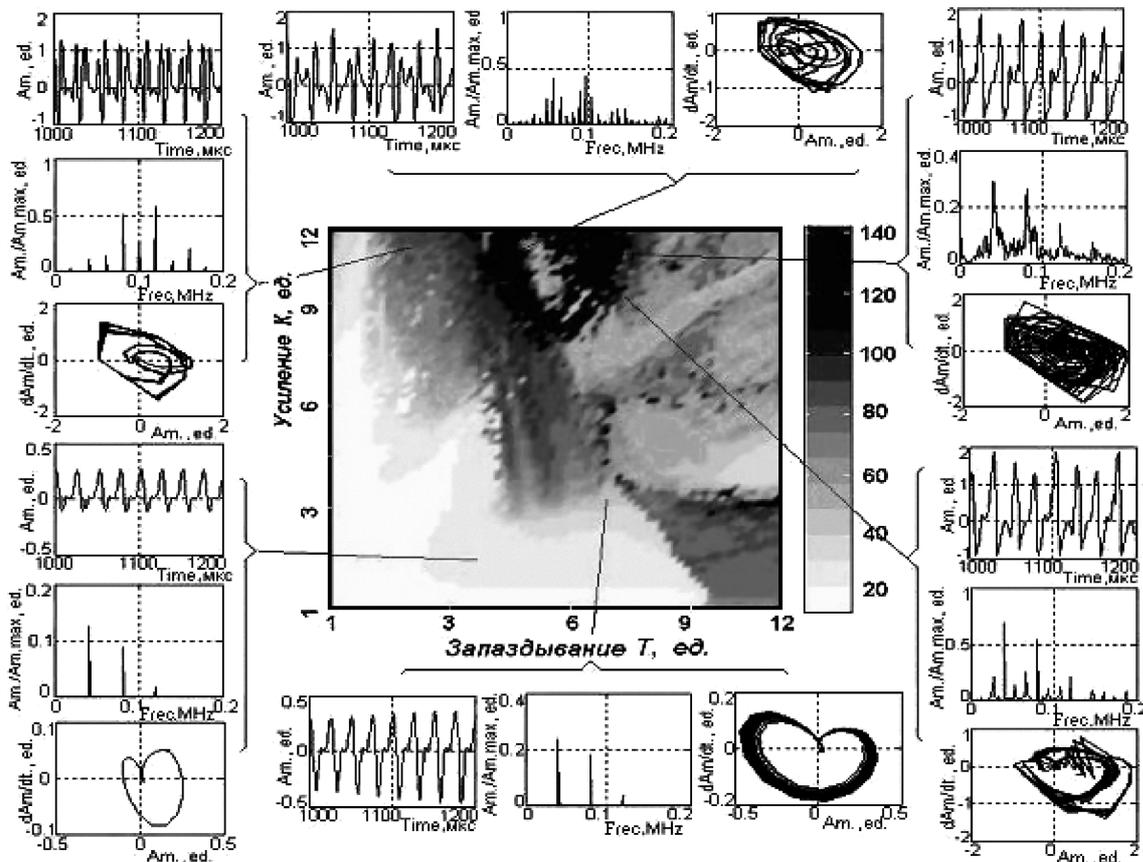


Рис. 2. Пример карты динамических режимов СЗА с накачкой  $\omega_{\text{нак}} = 4\omega_0$  и форм колебаний временных разверток, аттракторов и спектрограмм спектральной плотности мощности

Колебания на выходе СЗА при вариациях параметров  $T$  и  $K$  фиксировались в форме осциллограмм временных разверток сигналов, фазовых портретов аттракторов и оценок спектральной плотности мощности в виде 3D-диаграммы на карте динамических режимов (рис. 2). По оси абсцисс диаграммы отложены значения времени запаздывания  $T$ , а по оси ординат — значения  $K$  коэффициента усиления усилителя.

Третьим параметром диаграммы являлась точечная оценка спектральной плотности мощности, усредненная на интервале времени  $\tau = \eta(10T_0) = \eta\left(10\frac{1}{\omega_0}\right)$  — установления динамических состояний системы, которую вычисляли по формуле

$$S_{T, K} = \frac{1}{\eta} \sum_{j=1}^{\eta} s_j,$$

где  $s_j = \sum_{i=0}^{i=8} \frac{E_{\omega_i}^2}{\omega_i}$  — оценка спектральной плотности мощности (СПМ) в полосе от  $\omega_0$  — частоты резонанса БИХ-фильтра до частоты восьмой гармоники;  $E_{\omega_0}$  — значение коэффициента Фурье, вычисленное на выборке длиной 10 периодов частоты  $\omega_0$ ;  $\eta = 1, 2, 3, \dots$

Длина выборки на 10 периодах обеспечивала достаточное разрешение при наблюдении осциллограмм, спектрограмм и фазовых портретов. Значение частоты  $8\omega_0$  введено для ограничения параметров при оценке  $s_j$  в случае каскада бифуркаций удвоений периода аттрактора, приводящих к хаосу. Как правило, при хаотическом режиме амплитуда спектра на частоте  $\omega_0$  существенно падает, а амплитуды кратных гармоник растут, причем энергия низкочастотных составляющих бывает ниже высокочастотных. Для сглаживания подобных эффектов при оценивании  $s_j$  введены весовые коэффициенты  $1/\omega_i$ , где  $i$  — номер составляющей спектра Фурье представлен положительными числами и соответствует кратности частоты резонанса полосового фильтра  $\omega_0$ .

Фрактальный анализ режимов динамики системы как функций значений частоты накачки относительно собственной частоты системы, полученных в форме карт динамических режимов, осуществлялся посредством оценивания топологии карт методами лакуарности и топотезы.

Фрактальные образования различных динамических систем и различных режимов одной системы могут иметь близкие или даже одинаковые фрактальные размерности, однако резко различающиеся текстуры. Это означает, что фрактальная размерность, например хаусдорфова, не дос-

точно для описания и сегментации естественных текстур. Мандельбротом [11] был введен термин *lacunarity* — заполнение, для оценивания фрактальных поверхностей, которые имеют различающийся внешний вид или текстуру, однако одинаковые фрактальные размерности. Лакуарность хорошо отображает эволюцию динамики систем и масштабную инвариантность — самоподобие заполнения нормированного объема трехмерной КДР. Топотеза [12] поверхности определяет статистические и геометрические свойства случайных пространственных компонентов изотропной поверхности карт для диапазона их длин волн  $\lambda$ .

Вычисление лакуарности и топотез по значениям СПМ-карт динамических режимов с равным числом элементов, значения которых рассчитаны с одинаковой разрешающей способностью, инвариантно в отношении протяженности и расположения границ различных текстур. Такой подход обеспечивает переход от качественных оценок топологии динамики системы к количественным в их одномасштабном представлении.

В данном экспериментальном исследовании динамику системы с запаздыванием с двумерным отображением, вторая мерность которой определялась параметрическим возбуждением, изучали, подавая на параметрический вход системы накачку, значения частоты которой были кратны значениям резонансной частоты полосового фильтра от 0 до 4 с дискретностью 0,5.

На рис. 3 показаны изображения полученных карт динамических режимов.

Оценку лакуарности ( $\Lambda$ ) каждой топологии изображений КДР осуществляли по формуле

$$\Lambda = \frac{\sum_{T=1}^{T=T_{\max}} \sum_{K=1}^{K=K_{\max}} S_{T, K}}{T_{\max} K_{\max}},$$

где  $S_{T, K}$  — значение оценки энергетического спектра в точке КДР на пересечении значений координатных осей  $T$  и  $K$ .

Меру воздействия параметрической накачки  $\omega_{\text{нак}}$  на автоколебания СЗА вычисляли по значению относительного коэффициента лакуарности

$$\mu = \frac{\Lambda_{\omega_{\text{нак}}}}{\Lambda_{\omega_0}},$$

где  $\Lambda_{\omega_{\text{нак}}}$  — лакуарность КДР модели СЗА с накачкой;  $\Lambda_{\omega_0}$  — лакуарность КДР модели СЗА без накачки.

Зависимость относительных значений лакуарности графически показана на рис. 4.

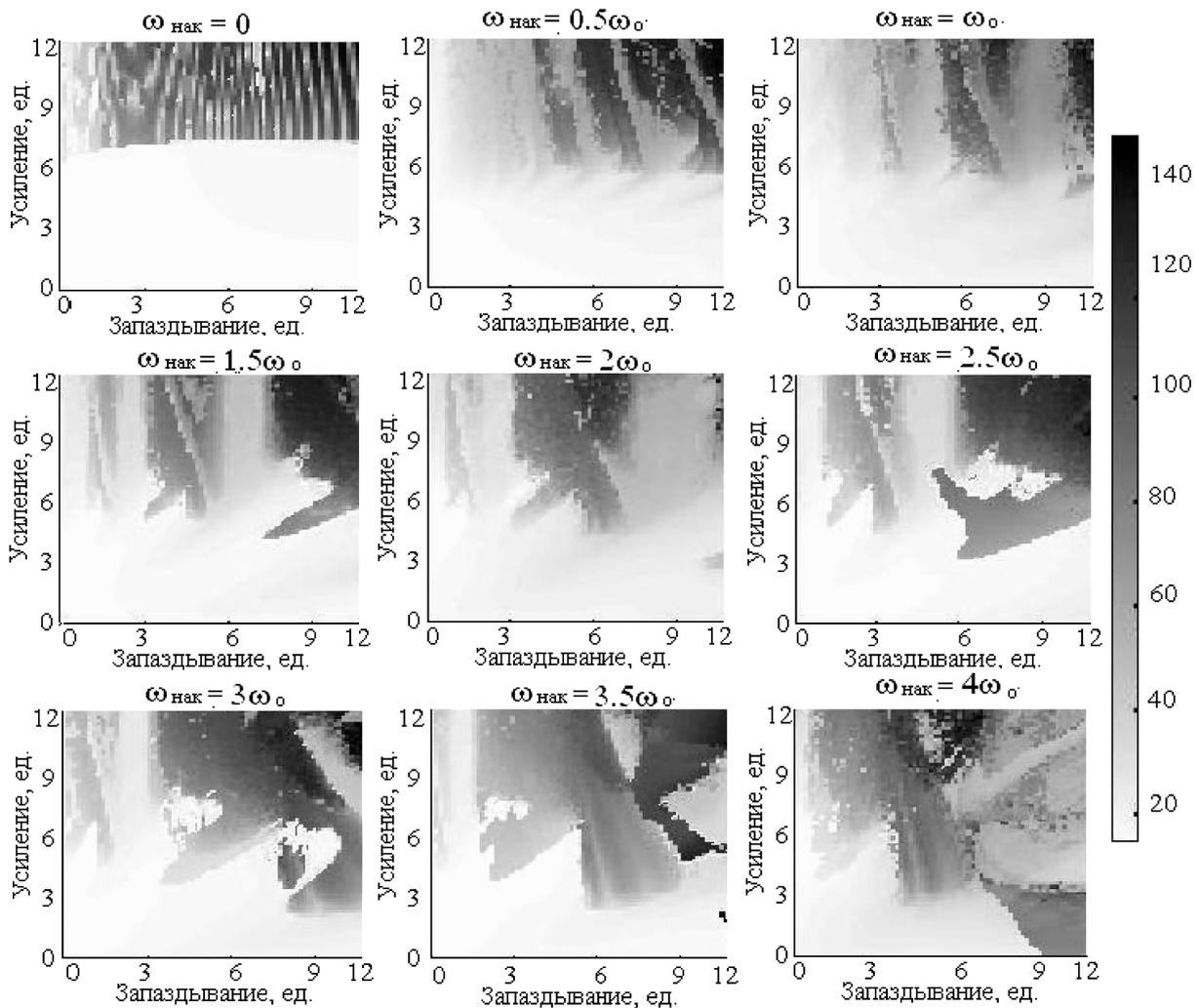


Рис. 3. Карты динамических режимов системы с запаздывающим аргументом с дополнительным двумерным отображением — параметрическим

Топотеза — метод представлений топологии динамики фрактальных естественных природных объектов, таких как поверхности регионов и морей, космических изображений поверхности Земли, получаемых со спутников, радиолокационных изображений, а также поверхностей разнообразных деталей после технологической обработки, основанный на оценивании неровностей пространственных спектров мощности. Природное подобие поверхностей приводит к мнению, что существует общая закономерность, лежащая в основе явления.

В [12] было сделано предположение о том, что пространственный спектр мощности  $G(\Omega)$  изображений обратно пропорционален квадрату угловой пространственной частоты  $\Omega = 2\pi f$  для разных видов 3D-поверхностей. Пространственная частота связана с длиной волны неровностей поверхностей равенством  $f = 1/\lambda$ . Соответственно, физические системы конечной протяженности  $l_0$  имеют минимальную пространственную частоту  $f_0 = \Omega_0/2\pi = 1/l_0$ .

Дисперсия высоты неровностей рельефа поверхности изображения

$$\sigma^2 = \int_{\Omega_0}^{\infty} G(\Omega) d\Omega,$$

как оказалось, удовлетворяет условию

$$\sigma^2 \approx l, \quad (2)$$

где  $l$  — линейная протяженность изображения. Это означает, что наклоны, кривизна и распределение экстремумов рельефов не являются существенными свойствами и поверхностей, и изображений, в том числе изображений КДР.

В спектральной форме соотношение (2) через константу  $L_0$  связано с наибольшей длиной волны  $\lambda_0$  выражением

$$\sigma^2 = L_0/f = \frac{L_0}{\lambda_0} = \frac{2\pi L_0}{\Omega_0},$$

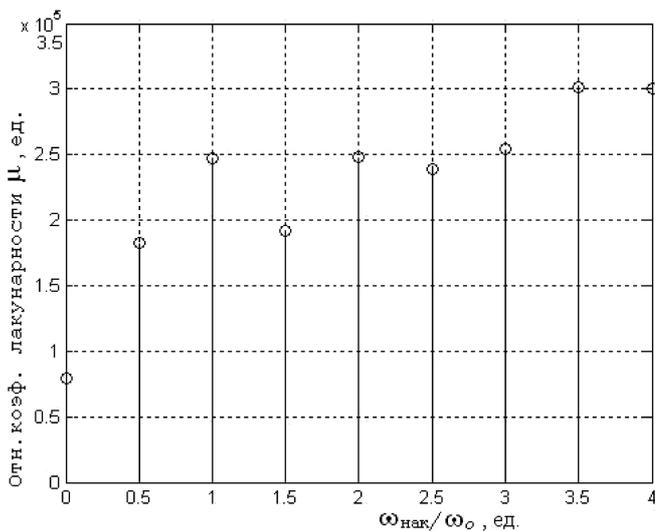


Рис. 4. Зависимость относительных значений лакунарности при различных частотах параметрической накачки

где  $\Omega_0$  — наименьшее значение частоты в спектре дисперсии  $\sigma^2$ .

Величина  $L_0$  была названа топотезой поверхности. Топотезы измеряются в единицах длины. На рис. 5 показаны расчетные графики значений дисперсий столбцов КДР и спектры Фурье их распределений для каждого из параметрических режимов.

На основании полученных распределений дисперсий столбцов карт и их быстрого преобразования Фурье (БПФ) значение нижней частоты было выбрано равным  $\Omega_0 = 0,01$  МГц.

Значения топотез  $L_0$  карт динамических режимов вычислялись по формуле

$$L_0 = \sum_{i=1}^{\varepsilon} \frac{\sigma_i^2 \Omega_0}{2\pi},$$

где  $i$  — номер строки КДР;  $\varepsilon$  — число строк КДР. График топотез показан на рис. 6 как функция значений частоты параметрической накачки.

Полученные числовые значения топотез изображений карт динамических режимов системы с запаздывающим аргументом, как динамической модели естественных природных и технологических фрактальных систем, показанные на рис. 6, сравнивались по порядку величин с известными, заимствованными из [8].

Оказалось, что диапазон значений топотез изображений карт модели без учета масштабного коэффициента по порядку величин лежит в интервале от 2 до 15 и соответствует известным значениям топотез изображений естественных природных систем, таких как поверхность Земли, поток лавы и др., что можно расценивать как показатель фрактальной адекватности модели.

Сопоставляя графические представления значений лакунарности и топотез КДР в исследован-

ной области отображений параметров модели СЗА, следует отметить, что эти меры оценивания изображений динамики фрактальных систем оказались весьма близки по конфигурациям несмотря на различия, положенные в основу алгоритмов вычислений оценок. По-видимому, эту близость также можно считать выражением фрактальности — самоподобия, а используемый метод математического моделирования естественных фрактальных процессов и систем — адекватным их динамике.

Отметим, что вид отображения в структуре СЗА определяется линейным полосовым идеальным резонансным фильтром второго порядка, собственная резонансная частота которого в данном исследовании была выбрана произвольно. Однако модель системы оказалась соответствующей естественной динамике поверхности Земли. Возможно, выбором других значений резонансной частоты фильтра, например более высокой, можно моделировать динамику поверхностей технических устройств после или в процессе их технологической обработки.

В заключение отметим следующее.

Еще в ранних работах [9, 10], посвященных системам с запаздывающим аргументом, было указано на совпадение общей формы записи уравнения движения системы с записью стандартного уравнения диффузии

$$\frac{\partial \rho(r, t)}{\partial t} = \bar{D} \Delta \rho(r, t), \quad \rho(r, 0) = \delta(r),$$

где  $\bar{D}$  — коэффициент диффузии;  $\rho(r, t)$  — распределение частицы с координатами  $r$  в момент времени  $t$ , начавшей свою историю с момента  $t = 0$  в результате воздействия  $\delta(r)$  — дельта-импульса;  $\Delta$  — лапласиан;

и диффузии в самых разнообразных системах с аномальными процессами.

При моделировании диффузионных систем установлено проявление эффектов памяти в результате фрактального скейлинга. Поэтому при описании фрактальных процессов обычные операторы дифференцирования заменяют соответствующими операторами дробного порядка. Такая замена ассоциируется с влиянием распределенных в среде дифференцирования ловушек с бесконечным средним значением времени пребывания в них частиц или решением уравнения движения системы с запаздыванием.

Качественный анализ карт динамических режимов исследуемой системы с запаздыванием показывает наличие таких областей, в которых динамика системы почти не зависит или зависит очень слабо от значений переменных параметров.

Вряд ли возможно получить адекватные аналитические приближения решений уравнений движения систем с запаздывающим аргументом при

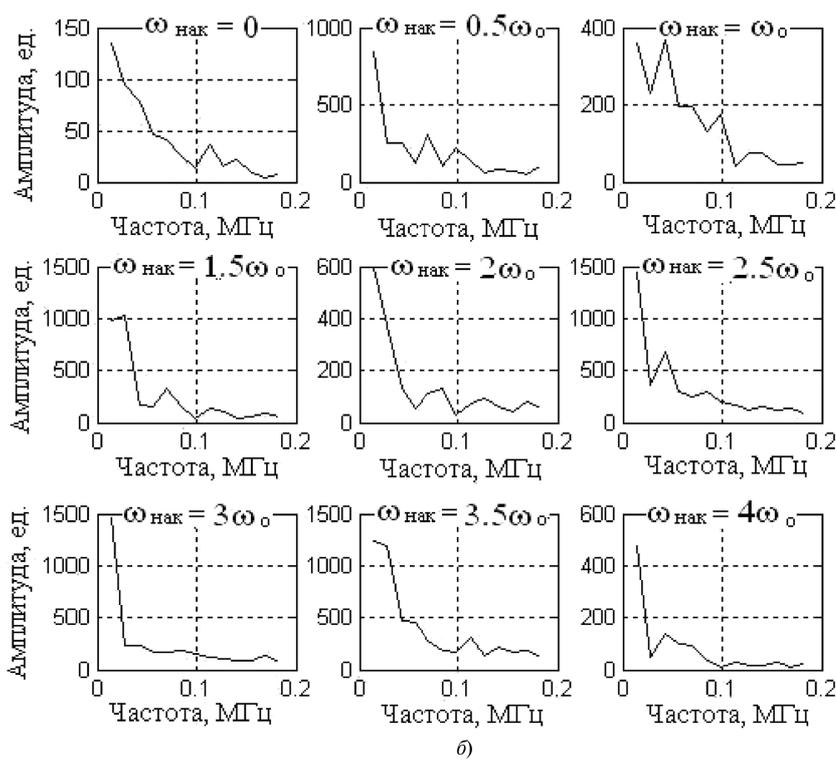
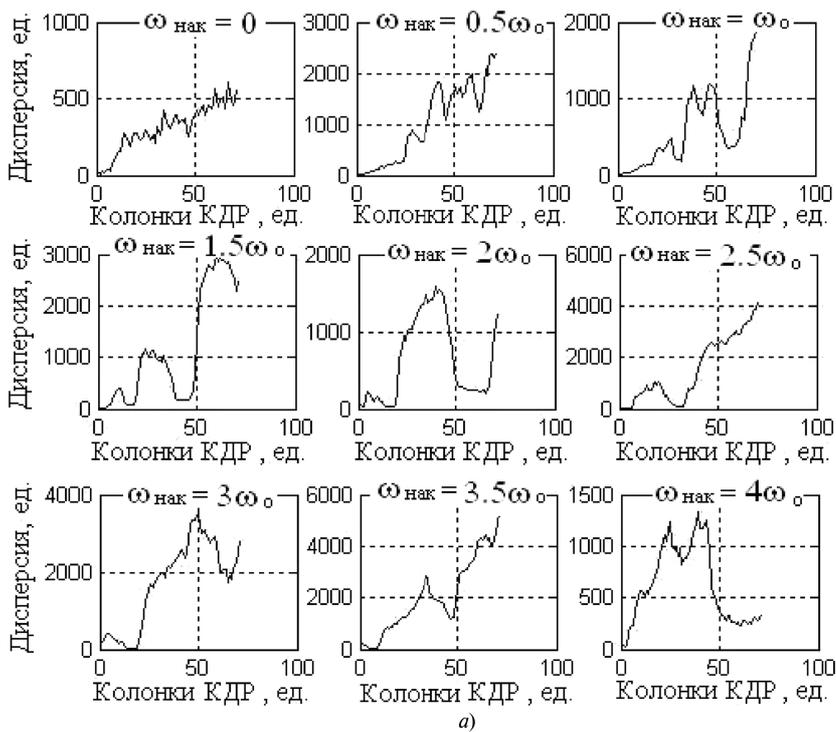


Рис. 5. Дисперсии столбцов КДР (а) и спектры Фурье (б) распределений дисперсий карт параметрических режимов при разных частотах накачки

значениях переменных конкретных систем. Однако спектральный анализ временных представлений решений в различных областях динамики карт режимов показывает наличие сложного текущего Фурье-спектра, особенно сильно изменяющегося в области хаотических режимов. Это позволяет

предположить о принадлежности решений к классу непрерывных, нигде не дифференцируемых функций. В настоящее время математический аппарат таких функций хорошо разработан и широко известен (например, [13]). Одними из простейших по виду примеров нигде не дифференцируемых функций являются функции Римана [14]

$$f(x) = \sum_{n=1}^{\infty} \frac{\sin^2 nx}{n^2}$$

или функции Ханкеля

$$f(x) = \sum_{n=1}^{\infty} \frac{\sin(\pi nx)}{n^2} \sin\left[\frac{1}{\sin(\pi nx)}\right],$$

где  $n$  — натуральные числа. В настоящее время такие функции называют фрактальными функциями, именно для них и создан аппарат фрактального анализа. Все это позволяет говорить о фрактальности системы с запаздывающим аргументом как о модели естественных природных и физических систем и, таким образом, о допустимости использования фрактальных мер оценивания их динамики.

Обобщенная структура системы с запаздывающим аргументом представляет собой замкнутую автоматическую систему с переменной структурой, что обоснованно относит ее к классу систем с переменными параметрами и с многими степенями свободы.

Динамика системы с запаздывающим аргументом в оценках лакунарности и топотезы подобна динамике уравнений с дробными производными или фрактальным канторовым множеством, так как эти оценки совпадают или близки по порядку величин с оценками динамики естественных систем.

Системы с запаздывающим аргументом, таким образом, могут рассматриваться как подобие естественных физических систем, они также описываются уравнениями в дробных производных, содержащих каналы, входящие в состав некоторой ветвящейся структуры, они подобно обобщенному броуновскому движению обладают памятью, причем эта память не кратковременная. Мультифрактальная динамика таких систем подобна динамике диффузных систем, таких как броуновские.

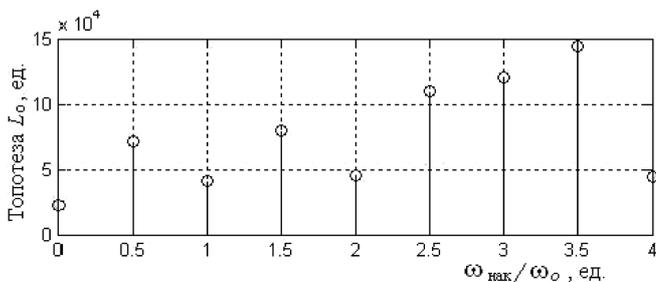
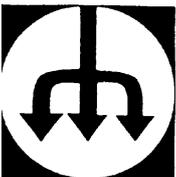


Рис. 6. График значений величин топозеза в зависимости от изменения частоты параметрической накачки

Динамика математических моделей природных и мультифрактальных технических систем, изучаемая с помощью автоматических систем с запаздывающим аргументом, подобно пространственным образам природных систем должна представляться картами динамических режимов в координатах переменных отображений, ответственных за изучаемые физические превращения. Алгоритмы оценивания динамики моделей фрактальными мерами должны отвечать самоаффинным представлениям фрактальности и основываться на пространственных фрактальных размерностях.

Работа выполнена при финансовой поддержке РФФИ, гранты № 06-08-08012, 05-05-08008, 07-08-13503, 07-08-13518.



## ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ СИСТЕМЫ

УДК 621.317.799:004.312.46

Г. Э. Куликов,

Институт оптики атмосферы СО РАН, г. Томск

### Автоматизированная система сбора и обработки данных для установок физического профиля

Предлагается структура инвариантной к условиям эксперимента системы сбора и обработки данных для экспериментальных физических установок. Система может быть реализована как на одном компьютере, так и как распределенная система. Задача корректного разделения ресурсов ЭВМ между времяемкими подпрограммами решена путем использования многотредового программирования. Построение в виде жестко иерархической модели, не зависящей от конкретной аппаратуры, и простая абстрактная модель измерений позволяет использовать работу в качестве основы для построения систем регистрации и обработки данных различных физических установок.

#### Список литературы

1. Самко С. Г., Килбас А. А., Маричев О. И. Интегралы и производные дробного порядка и некоторые их применения. Минск: Наука и техника, 1987.
2. Нигматуллин Р. Р. Дробный интеграл и его физическая интерпретация // Теор. и мат. физика. 1992. Т. 90, № 3. С. 354–368.
3. Климонтович Ю. Л. Введение в физику открытых систем. М.: Янус-К. 2002. 284 с.
4. Кобелев Я. Л., Кобелев Л. Я. Равновесная статистическая физика в фрактальных средах с постоянной и переменной памятью // ДАН. 2003. Т. 391, № 5. С. 614–618.
5. Гуревич В., Волмэн Г. Теория размерности: Пер. с англ. М.: ИЛ, 1948. 232 с. 2-е изд., испр. М.: Едиториал УРСС, 2004. 304 с.
6. Фракталы в физике: Пер. с англ. / Под ред. Я. Г. Синая и И. М. Халатникова. М.: Мир, 1988. 649 с.
7. Иванова В. С., Баланкин А. С., Бунин И. Ж., Оксогоев А. А. Синергетика и фракталы в материаловедении. М.: Наука, 1994. 383 с.
8. Потапов А. А. Фракталы в радиофизике и радиолокации: топология выборки. М.: Университетская книга, 2005. 847 с.
9. Oguztereli M. N. Time-lag control systems. New York and London: Acad. Press. 1966. 287 p.
10. Солодов А. В., Солодова Е. А. Системы с переменным запаздыванием. М.: Наука. Главная редакция физико-математической литературы, 1980. 384 с.
11. Mandelbrot B. B. The Fractal Geometry of Nature. N. Y.: Freeman, 1982. 468 p.
12. Sayles R. S., Thomas T. R. Surface Topography as Nonstationary Random Process // Nature. 1978. V. 271. N 5644. P. 431–434.
13. Keller J. M., Chen S., Grownover R. M. Texture Description and Segmentatin through Fractal Geometry // Comput. Vision, Graph., Image Process. 1989. V. 45. N 2. P. 150–166.
14. Сакс С. Теория интеграла: Пер. с англ. М.: ИЛ, 1949. 496 с.

#### Введение

Современные физические установки характеризуются большим числом каналов регистрации и высокими скоростями потока данных. Уже к началу 80-х годов XX в. исследования как на больших физических установках, так и на компактных экспериментальных стендах достигли такого уровня, при котором наличие автоматизированной системы сбора и обработки данных (ССОД) становится необходимым условием функционирования всего комплекса, а сама ССОД оказывается одной из наиболее сложных частей экспериментальной установки. ССОД предназначены для сбора данных с датчиков регистрирующей аппаратуры и архивации полученной информации для последующего использования в проводимых исследованиях и создаются на базе аппаратно-программных комплексов, включающих активную и измерительную аппаратуру установки, вычислительную технику и программное обеспечение. Существенно, что любая ССОД всегда работает в реальном времени эксперимента и осуществляет стыковку регистрирующей аппаратуры с общей базой данных и системой обработки измерений. Обязательной составляющей современных ССОД считается наличие развитого графического интерфейса с

пользователем, обеспечивающего простое управление и визуализацию хода эксперимента и состояния физической аппаратуры, а также средств протоколирования качества работы системы и возможности автоматического восстановления при сбоях работы по программе эксперимента. В то же время ориентация ССОД на жесткую зависимость деловой логики системы от графического интерфейса нежелательна, так как делает практически невозможным повторное использование кода.

Современный подход к конструированию ССОД экспериментальных физических установок (ЭФУ) предусматривает организацию сбора данных и управления установкой, как совокупности объединенных в единую систему ПЭВМ и/или специализированных микроконтроллеров, каждый из которых отвечает за отдельный узел ЭФУ [1]. На практике известные из публикаций ССОД малых ЭФУ редко реализуются в таком виде. Это обусловлено высокой стоимостью специализированного оборудования и сложностью ПО. Чаще всего ССОД реализуются в виде простых программ, обслуживающих отдельные приборы экспериментальной установки, написанных на каком-либо языке высокого уровня, и последовательно выполняющихся на единственном персональном компьютере. Реже коллектив экспериментаторов создает для проведения одного или группы схожих экспериментов специализированную программную систему.

В этой связи возникает необходимость создания унифицированных ССОД. В лабораторную практику активно внедряются развитые коммерческие программные платформы для создания ССОД, такие как система LabView [2]. Такие платформы предоставляют широкий ассортимент библиотечных элементов, соответствующих типовым задачам ССОД, и дают возможность пользователю строить приложения автоматизации для имеющейся у него экспериментальной техники. Можно было бы ожидать, что построение ЭФУ с использованием коммерчески доступного ПО является рутинной операцией. Тем не менее до сегодняшнего времени активно строятся ССОД, основанные на вновь разрабатываемом ПО. Можно заключить, что, хотя появление такой совершенной системы, как LabView, предоставляет стандартную платформу для создания ПО сбора и обработки данных, все же не в полной мере решает задачу разработки унифицированных методов построения ССОД установок физического профиля: LabView ориентирована на максимальную дружелюбность пользователю-непрограммисту и каждая программа LabView неразрывно связана с представлением пользовательского интерфейса [2, 3] и под каждый новый эксперимент необходимо индивидуально и целиком адаптировать программу на графическом G-языке среды LabView, программы LabView существенно связаны с конкретными типами аппаратуры и требуют переделки при ее замене. По-видимому, сложности с адаптацией G-программ LabView под различные, хотя и схожие, эксперименты вынуждают исследователей создавать собственное ПО ССОД, инвариантное к изменениям условий эксперимента.

Проблема еще более усложняется, если эксперимент ставит повышенные требования к ПО ССОД, такие как малое или гарантированное время реакции, большой объем накапливаемых данных, необходимость работы с

нестандартным (изготовленным в лаборатории) оборудованием.

Представляется целесообразной разработка максимально простой и допускающей адаптацию к конкретным условиям универсальной ССОД, занимающей промежуточную нишу между узкоспециализированными пользовательскими программами и сложными профессиональными платформами типа LabView. Целью настоящей работы была попытка создания такой простой ССОД ЭФУ:

- удовлетворяющей условиям конкретного физического эксперимента;
- предоставляющей основу для реализации программ обслуживания других физических экспериментов, проводимых на этой же или иных физических установках и допускающей модификацию условий проведения эксперимента с минимальными изменениями ПО;
- масштабируемой, пригодной как для простых лабораторных работ, так и для сложных ЭФУ;
- обеспечивающей простоту создания, модификации и повторного использования компонентов ССОД и адаптации их при изменениях условий проведения эксперимента;
- независимой от вида используемой аппаратуры и специфики ЭФУ;
- допускающей жесткое разделение частей ПО на отвечающие за работу со сбором данных и управление ЭФУ, с одной стороны, и интерфейс с пользователем, с другой;
- распределенной, способной выполняться на нескольких стандартных ПЭВМ и/или микроконтроллерах, объединенных сетью TCP/IP;
- в перспективе допускающей выполнение в режиме "жесткого реального времени";
- открытой и построенной на основе программных систем с открытым исходным кодом.

### Основные структурные элементы системы

Требование обеспечения повторного использования кода вызывает появление множества библиотек классов, что часто приводит к их частичной несовместимости между собой. Можно избежать такой ситуации, разбивая задачу на множество небольших подзадач, каждую из которых решает функционально законченный элемент. Предлагается использовать модель, в которой задача решается множеством отдельных модулей — процессов, являющихся генераторами и приемниками данных, взаимодействующими между собой через среду-посредник, доставляющую модулям сигналы и данные. Модули объединяются в цепочки, образующие схему ССОД, причем "схема" собирается динамически (позднее связывание).

В основу разработанной ССОД положены следующие решения.

1. Иерархический модульный принцип построения. Система представляет собой компактный "каркас", в который включаются модули — именованные "схемы", необходимые для проведения текущего эксперимента. Все задачи по сбору и обработке данных осуществляются независимыми и заменяемыми схемами", каждая из которых реализуется изолированным процессом (программой). Сбой в одном из процессов не приводит к краху

всей системы. "Рухнувший" процесс может быть автоматически перезапущен, если только "жив" общий родитель всех процессов системы. Модули-"схемы" оформляются в виде динамически загружаемых библиотек (.so/.dll) и могут включаться в работу без перекомпиляции. Интерфейс с пользователем отделен от "схемы" сбора данных и в любой момент времени может быть выключен или заменен без остановки эксперимента. Предполагается, что основные задачи ССОД выполняются относительно простыми, а значит, содержащими минимум ошибок "схемами", взаимодействующими между собой. Каждая "схема" состоит из совокупности *модулей*, ответственных за взаимодействие с физической аппаратурой и/или обработку данных. В свою очередь, "схема" может быть использована как модуль более сложной "схемы". Каждая схема и модуль схемы следуют одинаковому интерфейсу.

Каждый модуль состоит из нескольких потоков исполнения (тредов) с четко заданными приоритетами [4]. За целевые вычисления отвечает исполнительный тред, сервисные функции сосредоточены в управляющем треде. Исполнительные треды модулей запускаются с дисциплиной планирования реального времени (SCHED\_FIFO | SCHED\_RR). С одной стороны, вычислительные и прочие подпрограммы системы конкурируют за процессор, естественным образом оптимально распределяя имеющиеся ограниченные ресурсы. С другой стороны, модульная и многотредовая архитектуры позволяют, при аккуратном использовании, поддерживать предельную простоту программного кода системы.

Как модули, так и образованные из них "схемы" следуют единому программному интерфейсу: каждый модуль рассматривается как "черный ящик", имеющий доступный для чтения статус и несколько входов и выходов ("портов"), причем выход одного модуля может соединяться со однотипным входом другого, образуя конвейер обработки данных. Определены следующие типы "портов":

**А. Вход команд.** Предназначен для управления состоянием как индивидуального модуля, так и "схемы" в целом. На уровне модуля реализован посредством механизма надежной доставки сигналов, известного как делегаты, или механизм "сигналов-слотов". На уровне "схемы" вход команд реализован в виде системной очереди сообщений SysV [5].

**В. Входы и выходы данных.** Предназначены для обмена данными между модулями и "схемами". Реализованы в виде указателей на программный объект-посредник передачи данных, предоставляющий методы для чтения и записи данных. Модуль может быть как потребителем, так и генератором данных или их комбинацией. Неиспользуемые "порты" и данные игнорируются. Входы и выходы "схемы" всегда именованы, данные размещаются в разделяемой памяти, доступ к которой может получить любой процесс. Данные модуля могут быть как именованными и доступными за пределами "схемы", образуя "контрольные точки", так и приватными.

**С. Входы и выходы управляющих сигналов.** Предназначены для быстрого управления внутренним состоянием модуля и синхронизации модулей в схеме. Реализованы в виде ссылок на событийные семафоры (*conditional variables* [6, 5]), причем используется широковежание. К одному и тому же сигналу могут подключаться

множество модулей. В каждом модуле определено минимум три сигнала: входные сигналы запуск/шаг (GO) и запуск по готовности (RDY); выходные сигналы готовности (READY) и требования на обслуживание (LAM). Кроме того, передача соответствующего сигнала может быть реализована передачей команды. Сигналы влияют на состояние управляющего и исполнительных тредов модуля, осуществляя синхронизацию между модулями. Сигналы модуля всегда приватны (недоступны за пределами "схемы"), сигналы "схемы" доступны для всех связанных с ней других "схем" и программ обработки данных.

**Д.** В случае возникновения ошибочной ситуации модуль может отправить схеме сообщение об ошибке. Для отправки сообщения используется семафор-счетчик, битовый массив и генерация сигнала реального времени SIGRTMIN+13, обработка которого проводится на уровне схемы или любого заинтересованного модуля.

2. Связь с аппаратурой осуществляется через абстракцию "прибора", следующую интерфейсу модуля. Этим достигается независимость "схемы" от типа реально используемой аппаратуры ЭФУ. Благодаря использованию позднего связывания возможна замена аппаратуры без перекомпиляции системы.

3. Модуль не выполняет никаких действий, пока не поступит команда или сигнал запуска (GO). Интерпретация сигнала GO зависит от реализации модуля и может вызывать как однократное, так и серийное срабатывание. Срабатывание происходит только в том случае, если вместе с сигналом GO активен и разрешающий сигнал готовности RDY, таким образом осуществляется синхронизация между модулями в цепочке. Привязка к ходу времени в "схеме" проводится с помощью системных таймеров ядра и обработки сигнала SIGALRM, модуль синхронизации "схемы" получает запросы от других модулей на точный отсчет времени, привязывается к системному таймеру высокого разрешения (/dev/rtc) и транслирует окончание интервала в событие условной переменной.

4. Каждый модуль/"схема" может находиться в трех состояниях: выключен, включен, отключен. В выключенном состоянии модуль не реагирует ни на какие воздействия, кроме команды включения, и не генерирует никаких сигналов. Все сигналы и данные, поступающие в "выключенный" модуль, безвозвратно теряются. Во включенном состоянии исполнительные методы модуля "спят" на своих условных переменных, ожидая возможности выполнения. В отключенном состоянии модуль не производит никаких действий, а все сигналы и данные на входах модуля прозрачно доставляются следующему модулю в цепочке, причем дополнительных временных задержек при этом не вносится. Для поддержания предсказуемого состояния системы переключение модуля из одного состояния в другое выполняется тогда и только тогда, когда все модули одной цепочки находятся в состоянии ожидания (ни один из них не выставил сигнал "готовности").

5. Передача данных между модулями и схемами происходит через полиморфный объект-посредник, выполняющий функции буфера данных. Данные передаются блоками фиксированной длины, выбираемой при реализации модуля. Интерфейс посредника — буфера данных унаследован из интерфейса модуля, расширенного командами записи и чтения блока данных, что позволяет

пользователю уточнять функции и организовывать в посреднике предварительную обработку данных "на лету". Посредник реализован в виде кольцевого буфера в памяти. Управление доступом к буферу осуществляется семафорами-счетчиками и событийными семафорами. Буферы организованы по схеме "много производителей—много потребителей" и допускают обработку данных "на месте": каждый потребитель имеет собственный набор указателей на ячейки буфера, каждая ячейка имеет индивидуальный счетчик клиентов. Это позволяет избежать длительных блокировок буфера. Одновременно объект-посредник служит для выравнивания скоростей генерации и потребления данных, что позволяет бороться с эффектом инверсии приоритетов [7] и использовать медленную и низкоприоритетную обработку и визуализацию данных совместно с "быстрыми" высокоприоритетными процессами сбора. Если при создании объекта-посредника было указано его "имя", то данные размещаются в блоке разделяемой памяти, доступной всем процессам. Это позволяет наладить быстрый обмен данными между "схемами" ССОД и легко реализовать взаимодействие частей ССОД, написанных на разных языках программирования. В целях контроля и визуализации хода эксперимента объект буфера предоставляет метод неблокирующего чтения последней записанной порции данных.

6. "Схемы" сбора и обработки данных могут находиться как на одном компьютере, так и на нескольких ПЭВМ или на микроконтроллерах, объединенных сетью TCP/IP. При разработке метода взаимодействия по сети было решено не использовать стандартных протоколов RPC/CORBA и т. п. в силу относительной "тяжеловесности" [7, 8] последних, что может быть проблемой, учитывая необходимость работы в реальном времени эксперимента. Для обмена сигналами, командами и короткими блоками данных используется протокол UDP. Крупные блоки данных могут передаваться как в последовательных дейтаграммах UDP, так и в потоках TCP. UDP выбран из соображений максимального быстродействия и легкости реализации в микроконтроллерах. Для снижения нагрузки на сеть пакеты сигналов и данных собираются в очереди и отправляются в виде единой дейтаграммы с периодом не меньшим, чем указано при построении "схемы" (обычно 10 мс). "Схемы" не нуждаются в модификации для обеспечения "сетевой прозрачности", вместо этого используется простая программа-посредник, подключающаяся к портам схемы и ретранслирующая сигналы и данные в сеть. Так как важно максимально разгрузить сеть, символические имена для идентификации "схем" не используются. Обращение к "схеме" происходит по IP адресу машины и фиксированному номеру порта, закрепленному за той или иной "схемой". Вызов посредника и создание "схемы", если она не была еще загружена,

осуществляется из стандартного суперсервера xinetd, на него же ложится проверка авторизации.

7. При разработке схемы поддержки физического эксперимента необходимо не только собирать и обрабатывать данные, но и управлять приборами ЭФУ согласно некоторому сценарию эксперимента. Сценарий эксперимента может быть как неинтерактивным, так и допускать вмешательство пользователя. Попытка реализовать обработку сценария в виде дополнительной логики в "схеме" эксперимента приводит к резкому усложнению программирования и нарушает идеологию системы. Вместо этого принято, что сценарий оформляется в виде отдельной программы, способной активно взаимодействовать со "схемой", отправляя ей сообщения. Разработка сценария эксперимента и модулей обработки данных не должна быть трудоемким процессом. Система поддерживает привязку к интерпретируемому объектно-ориентированному языку программирования сверхвысокого уровня *Python* [9], что позволяет очень быстро создавать и отлаживать соответствующие модули. Центральный

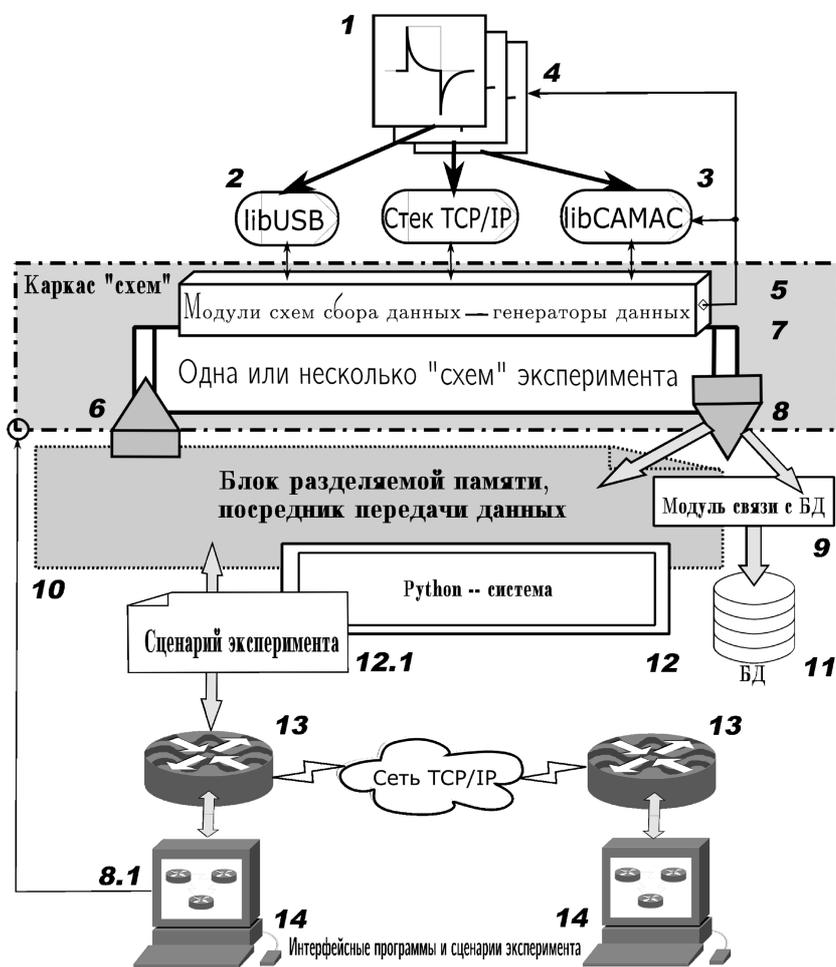


Рис. 1. Общая структура ССОД;

1 — физические приборы стенда; 2 — интерфейсные библиотеки и драйверы; 3 — "сырые" данные эксперимента; 4 — управляющие воздействия; 5 — модули ССОД; 6 — входные порты "схемы" эксперимента; 7 — одна или несколько схем эксперимента; 8 — выходные порты; 8.1 — сигналы и состояние схемы; 9 — модуль архивации данных и связи с СУБД; 10 — полиморфный объект — буфер данных в разделяемой памяти; 11 — СУБД; 12 — интерпретатор *Python*; 12.1 — транслятор событий в пакеты TCP/UDP; 13 — ретранслятор событий в пакеты TCP/UDP; 14 — интерфейсные программы и сценарии эксперимента

модуль разработанной ССОД — это *python*-программа, способная внутри себя интерпретировать другую *python*-программу, представляющую собой пользовательский сценарий эксперимента.

8. Отображение (контроль за ходом эксперимента) осуществляется совершенно независимыми, низкоприоритетными программами, подключенными как "потребитель" к соответствующим буферам передачи данных в разделяемой памяти. Таким образом, минимизируется влияние процессов отображения на ход эксперимента, а возможный сбой или ошибка в программах отображения никак не сказывается на ходе измерений. Более того, для работы с одной и той же "схемой" эксперимента могут использоваться разные программы визуализации, и наоборот, что способствует многократному использованию кода и сокращению времени на программирование. Программы отображения и контроля представляют собой как простые приложения командной строки, так и графические оболочки, написанные на языках C++ и/или *Python* с применением многоплатформенной библиотеки Qt [10–12] и *python*-интерфейса PyQT [13].

Разработанная система содержит небольшой набор заранее разработанных модулей, выполняющих функции взаимодействия с приборами ЭФУ через интерфейсы КАМАК, Ethernet/UDP и USB, а также проводящих некоторую предварительную обработку данных [14–16]. Поддержка конкретного физического эксперимента осуществляется путем сборки "схемы" ССОД из набора этих модулей либо разработанных пользователем. Соединения модулей, образующих "схему", читаются из файла

конфигурации (.ini файла типа секция/параметры) либо блочной диаграммы. Кроме того, "схемой" может быть любая программа на любом языке, следующая соответствующему интерфейсу. Таким образом решается задача сведения программирования к минимуму при сохранении гибкости. Поскольку одновременно работающие "схемы" могут быть совершенно независимыми, решается задача обеспечения работы с аппаратурой стенда независимо от основного эксперимента.

Данные эксперимента необходимо сохранять в форме, пригодной для последующего анализа. Каждый набор экспериментальных данных должен содержать не только информацию измерения, но и метаинформацию, описывающую условия измерений, параметры эксперимента и т. д. Метаинформацию удобно сохранять в реляционной базе данных (РБД), для чего система предоставляет модуль связи с РБД MySQL или PostgreSQL. В то же время основные данные измерений также должны быть сохранены в структурированном виде. Сохранение большого набора данных в виде бинарного объекта РБД нецелесообразно, а при жестких временных требованиях и невозможно. Для сохранения данных непосредственных измерений система предоставляет модуль, осуществляющий предварительную буферизацию и сохранение данных в локальной БД (модель "ключ"—"значение") *Berkeley BD* [17].

Общая структура разработанной ССОД показана на рис. 1; на рис. 2 приведен пример возможного варианта соединения модулей схемы эксперимента.

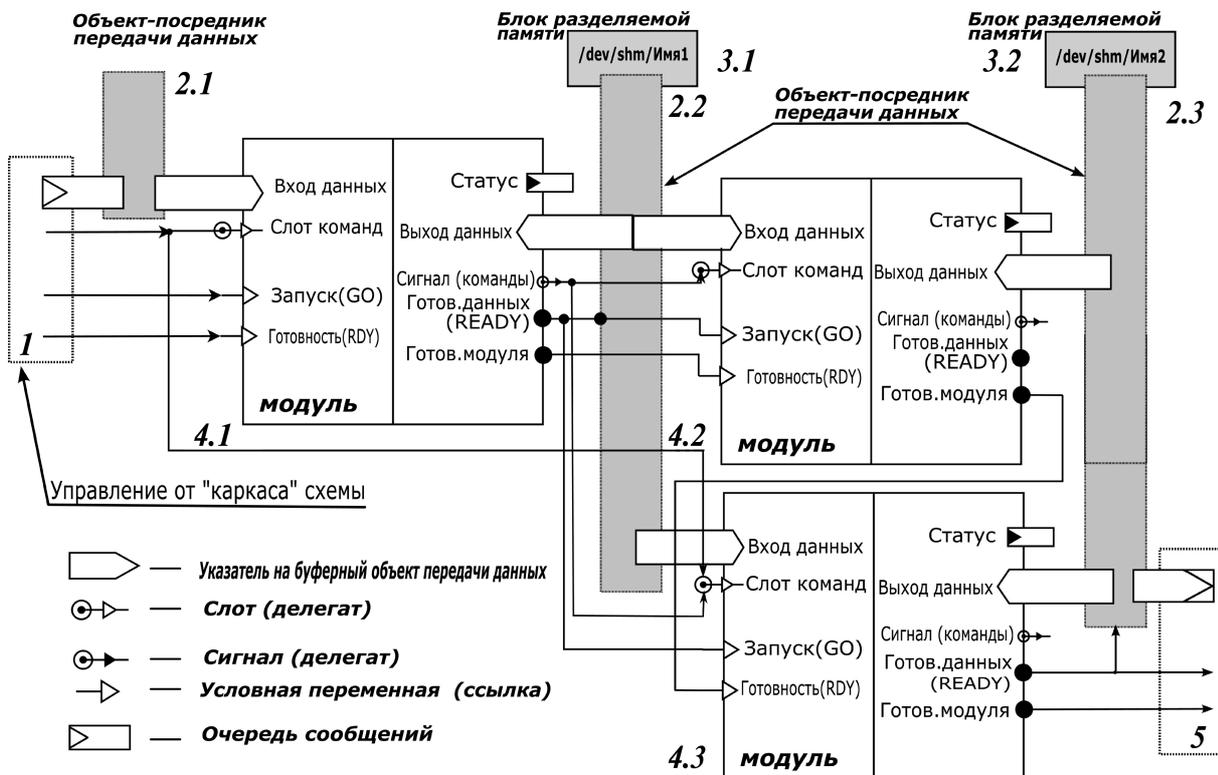


Рис. 2. Возможный вариант компоновки модулей схемы эксперимента:

1 — входные порты схемы; 2.1–2.3 — полиморфные буферы-посредники передачи данных; 3.1–3.2 — данные в разделяемой памяти — контрольные точки схемы эксперимента; 4.1–4.3 — модули — представления физических приборов и обработки данных; 5 — выходные порты схемы

## Программная платформа системы

Предлагаемая система интенсивно использует множество потоков управления во многих процессах. Известно [18], что многотредовая модель применима для решения конкретной проблемы, если время реакции комплекса "операционная система + прикладная программа" на внешнее событие, включая время переключения между потоками, заведомо меньше характерных времен задачи. Для реализации многотредовости при управлении аппаратурой обычно используют специализированные операционные системы (ОС) реального времени, гарантирующие фиксированное время реакции на событие. Вместе с тем в большинстве задач малых ЭФУ особенности работы и характерные времена аппаратуры таковы, что время реакции может без ущерба варьироваться в пределах нескольких миллисекунд. Это позволяет использовать одну из распространенных ОС широкого применения, выбрав ее исходя из общих соображений и по основному критерию — стратегии работы планировщика. Пригодны ОС, планировщик которых переключает процессор строго в соответствии с установленной программистом системой приоритетов и имеет минимальное время переключения между потоками. Желательно также выбрать программную платформу, пригодную не только для решения узкоспециализированных задач, но и для повседневной комфортной работы пользователя.

Описываемая ССОД реализовывалась на платформе открытой, общедоступной ОС *Linux*.

Разработанная ССОД не "привязана" к определенному типу файловой системы. Выбор *Linux* для реализации ССОД позволил рассмотреть различные варианты выбора ФС для хранения данных. Так, в испытывавшемся варианте локальная БД размещается на очень быстрой файловой системе XFS, специально разработанной для хранения больших файлов, мультимедийных данных. На выбор XFS повлияло то, что она практически не подвержена фрагментированию и не требует применения длительных процедур дефрагментации, а также имеет средству работы в "реальном времени" [19].

При реализации концепции "схем" использовался ограниченный набор средств межпроцессного взаимодействия, что позволяет надеяться на успешный запуск разработанной ССОД под расширениями жесткого реального времени [20, 21] или на микроконтроллерных устройствах.

## Результаты

Настоящая система разрабатывалась как составная часть импульсного лазерного спектрометра с бигармоническим излучателем, предназначенного для экспериментальной проверки гипотез [22, 23]. Условия эксперимента накладывают настолько жесткие требования, что необходимо вести длительные (часы) наблюдения, обрабатывать каждый лазерный импульс (период следования  $80 \text{ мс}^{-1}$ ), постоянно контролируя спектральный состав и соотношение интенсивностей компонент возбуждающей бигармоники, отбрасывая "неподходящие" по условию задачи реализации и выдавая извещения экспериментатору. Оптимизация условий в ходе эксперимента требует отображения в реальном времени основной ин-

формации, включая спектрограмму переизлучения. Поток регистрируемых данных, без учета сигналов управления установкой, превышает 100 Кбит/с. Тестовые запуски ССОД показали, что разработанная система обеспечивает требуемые временные характеристики и вполне адекватна по надежности.

Сетевой компонент системы был проверен в промышленных условиях (2 года эксплуатации в г. Ханты-Мансийск в составе аппаратуры оптических гигрометров "Зима").

*Работа выполнена при поддержке РФФИ, грант № 06-08-00751.*

*Автор признателен В. Т. Калайде за ценные критические замечания при подготовке настоящей работы.*

## Список литературы

1. **Олссон Г., Пиани Д.** Цифровые системы автоматизации и управления. В 2 томах. СПб.: Невский Диалект, 2001. 557 с.
2. **Тревис Д.** LabView для всех / Пер. с англ. Клушин Н. А. М.: ДМК Пресс, ПриборКомплект, 2004. 544 с.
3. **Lynn B. N.** Exploring the basic capabilities of labview // Masters Abstracts International. 2000. Vol. 38 N 6.
4. **Куликов Г. Э.** Применение многотредовой модели в программном обеспечении лазерного бигармонического спектрометра // Приборы и техника эксперимента. 1998. № 5. С. 71—78.
5. **Стивенс У.** UNIX: взаимодействие процессов. СПб.: Питер, 2002. 576 с.
6. **Butenhoff D. R.** Programming with POSIX Threads. Addison—Wesley/PUB Coinc., 1997. 382 p.
7. **Цилорик О., Горошко Е.** QNX/UNIX, анатомия параллелизма. СПб.—М.: Символ, 2006. 288 с.
8. **Цимбал А., Аншина М.** Технология создания распределенных систем. СПб.: Питер, 2003. 576 с.
9. **Сузи Р.** Программирование на языке Python. СПб.: БХВ-Петербург, 2002. 528 с.
10. **Qt Reference Documentation (Open Source Edition)**, 2006. <http://doc.trolltech.com/3.3/index.html>.
11. **Шлее М.** Qt: профессиональное программирование на C++. СПб.: БХВ-Петербург, 2005. 544 с.
12. **Циблатев А. В.** Библиотека QT4. Создание прикладных приложений в среде Linux. Киев: Диалектика, 2006. 256 с.
13. **PyQt v4 — Python Bindings for Qt v4 Reference Guide**, 2006. — <http://www.riverbankcomputing.com/Docs/PyQt4/pyqt4ref.html>.
14. **Куликов Г. Э.** О программном обеспечении измерителя длин волн в бихроматическом лазерном спектрометре // Оптика атмосферы и океана. 1999. Т. 12, № 9. С. 854—857.
15. **Куликов Г. Э.** Перестраиваемый заграждающий фильтр лазерного бигармонического спектрометра и процедура его калибровки // Оптика атмосферы и океана. 2003. Т. 16. № 11. С. 1041—1043.
16. **Куликов Г. Э., Калайда В. Т.** Алгоритм высокочастотной фильтрации данных для лазерного спектрометра с ФПЗС — линейкой на основе вейвлет-преобразования с автоматическим выбором коэффициентов // Оптика атмосферы и океана. 2004. Т. 17. № 12. С. 1046—1050.
17. **Олсон М., Бостик К., Зельтцер М.** Berkeley DB // Открытые Системы. 2000. № 11. <http://www.osp.ru/os/2000/11/178318/>.
18. **Lin J.** // Computer. 1991. N 5. P. 58—61.
19. **XFS for Linux**, 2003. — <http://oss.sgi.com/projects/xfs/papers/ukuug2003.pdf>.
20. **RTAI: Real-Time Application Interface**, review, 2006. — <http://www.linuxdevices.com/articles/AT6605918741.html>.
21. **Yodaiken V.** An Introduction to RTLinux, 1997. <http://www.linuxdevices.com/articles/AT3694406595.html>.
22. **Еньшин А., Творогов С.** Возбуждение и эмиссия двухатомных молекул в резонансном бигармоническом поле // ДАН СССР. — 1990. — Т. 341, № 3.
23. **Лопасов В. П.** Возможность активации спин-орбитального взаимодействия в молекуле полем оптической бигармоники // Оптика атмосферы и океана. 1996. Т. 9. № 8. С. 1151.

**А. В. Чернышов**, канд. техн. наук,  
Московский государственный университет леса

## Автоматизированная генерация программ контроля бортовой телеметрической аппаратуры

*Рассмотрена проблема создания программ контроля бортовой телеметрической аппаратуры. Предложено решение рассмотренной проблемы в виде автоматизированного генератора программ контроля. Описана реализация автоматизированного генератора для системы "Орбита-ИВМО".*

### Введение

Современная бортовая радиотелеметрическая аппаратура (БТА) представляет собой сложный технический комплекс, обеспечивающий сбор информации с сотен и даже тысяч датчиков, установленных на борту, и передачу их показаний по единому радиоканалу на наземную аппаратуру. При этом каждый датчик подключается к своему входу аппаратуры, который в нашем случае можно понимать как отдельный измерительный канал в составе БТА.

Задача контроля качества БТА определяется прежде всего как задача контроля корректности передачи сигналов датчиков по измерительным каналам. Наиболее полное решение этой задачи достигается при контроле БТА на земле перед установкой ее на изделие. Традиционно эта задача решается с применением специализированной наземной контрольно-испытательной аппаратуры. При этом обычно на вход каждого измерительного канала подается заранее известное воздействие, и на выходе системы контролируется качество передачи этого воздействия по каналу.

Ранее в системах с относительно небольшим числом каналов такой контроль выполнялся испытателями вручную. Современная БТА может содержать сотни и даже тысячи каналов, поэтому для выполнения контроля в разумное время и с надлежащим качеством процесс необходимо автоматизировать.

Автоматизация процесса контроля достигается включением в состав наземной контрольной аппаратуры ПЭВМ. Но для эффективного применения ПЭВМ необходима разработка соответствующих контролирующих программ. А здесь главной проблемой становится не разработка собственно алгоритмов работы контролирующей программы (которые известны и хорошо проработаны), а необходимость индивидуального описания каждого измерительного канала.

### Постановка задачи

Практически любая современная БТА имеет в своем составе программируемые блоки, которые теоретически могут быть настроены на решение необходимых потребностей функций без привлечения разработчика или изготовителя, причем непосредственно перед установкой на испытываемое изделие либо даже непосредственно на изделие. Эта возможность обеспечивается поставкой в

составе наземных средств БТА соответствующих программаторов и необходимого программного обеспечения (ПО), управляющего программаторами.

Однако на сегодняшний день эти возможности потребителями не востребованы, так как существует проблема проверки корректности подготовленных и зашитых в БТА самими потребителями программ ее работы. Поэтому соответствующая задача либо возлагается на плечи разработчиков (что требует от потребителей и дополнительных финансовых затрат, и дополнительных затрат времени), либо вообще снимается, и потребитель вынужден довольствоваться настройкой БТА, выполненной заводом-изготовителем при производстве и испытании.

Собственно проверка правильности программирования блоков БТА и на заводе-изготовителе, и у потребителя обычно сводится к одной и той же процедуре — установке на входы БТА имитаторов сигналов датчиков, выполненных в виде заглушек, и проверке корректности прохождения сигналов имитаторов через БТА.

Проблематичность этой процедуры состоит именно в проверке корректности прохождения сигналов. Как уже было сказано, современная БТА имеет в своем составе сотни и даже тысячи каналов. Выполнить контроль с надлежащим качеством и в разумное время возможно только с применением современных средств автоматизации. Но если на заводе-изготовителе комплекты БТА, выпускаемые в рамках одной комплектации, проходят проверку по единой программе, описанной в ТУ, и следовательно, для этого единожды выполняется разработка соответствующего контролирующего ПО, то при эксплуатации каждому потребителю обычно требуется свой вариант программирования БТА. Более того, в некоторых случаях одному и тому же потребителю требуется многократно перепрограммировать режимы работы одного и того же комплекта БТА (например, в зависимости от типа испытания или эксперимента, проводимого на изделии), что требует создания контролирующего ПО для каждого варианта программирования БТА.

Практика составления таких программ, сложившаяся сегодня у всех разработчиков и потребителей БТА, сводится к описанию в программе проверки каждого канала непосредственно при составлении контролирующей программы вручную. В силу огромного числа каналов составление каждой такой программы отнимает много времени и сил программистов и черевато ошибками. А при необходимости частой смены программ работы БТА программисты просто не успевают разрабатывать контролирующие программы. Таким образом, весьма актуальным представляется решение задачи автоматизации составления контролирующих программ для БТА.

В данной статье показан вариант решения этой проблемы, реализованный в программном обеспечении рабочего места контроля (РМК) БТА "Орбита-ИВМО" [1], разработанной в ОКБ МЭИ и серийно выпускаемой на Ижевском радиозаводе. В силу конструктивных различий, имеющихся в разных системах БТА, это решение не является универсальным, но может служить базой для решения проблемы автоматизации создания программ контроля и других систем БТА.

### Структура и принцип контроля БТА "Орбита-ИВМО"

Бортовая аппаратура системы "Орбита-ИВМО" представляет собой набор унифицированных, функционально законченных приборов, каждый из которых решает свою задачу, например, коммутатор аналоговых каналов, коммутатор температурных каналов, коммутатор сиг-

нальных каналов, центральный формирователь, усилитель-нормализатор сигналов.

Для построения конкретной БТА приборы объединяются в модули — функционально законченные блоки БТА, каждый из которых способен решать задачи сбора и передачи информации от всех типов датчиков изделия или заданного блока (зоны) изделия. Для небольшого изделия БТА может представлять собой один модуль. Для большого изделия может создаваться несколько различных модулей, объединяемых в систему и работающих совместно.

Для удобства управления системой в большинстве приборов измерительные каналы объединены в группы. Каналы в группе имеют общий групповой адрес опроса и единую частоту опроса со сдвигом по времени. В большинстве приборов имеется четыре группы.

Число каналов во всех группах одного прибора одинаковое. Каждая группа соответствует отдельному разъему на корпусе прибора. В некоторых приборах (например, в коммутаторе температурных датчиков) все каналы объединены в одну группу.

Контроль модуля/системы заключается в том, что на входные разъемы приборов устанавливаются заглушки, имитирующие статические уровни показаний датчиков, а с выхода системы считываются измеренные каждым каналом значения и сравниваются с известными уставками заглушек.

Для полноценной проверки модуля/системы может понадобиться несколько циклов контроля с различными наборами заглушек на входах. И для каждой расстановки заглушек необходимо создание контролирующей программы. При этом алгоритмы работы всех контролирующих программ будут одни и те же. Различие будет заключаться именно в уставках заглушек. Ввод же этих уставок, соответствующих сотням и тысячам каналов, представляет собой объемную и рутинную задачу, автоматизация которой позволяет значительно сократить время на разработку контролирующей программы, снизить вероятность ошибок ввода (а в идеале исключить их), минимизировать время на отладку программы, повысить оперативность контроля и снизить требования к уровню квалификации программиста, разрабатывающего контролирующую программу.

### Возможности автоматизации

Принцип автоматизации генерации контролирующих программ во многом определяется как особенностями самой БТА, так и организацией контролирующей аппаратуры, да и самой технологией контроля, принятой для данной БТА. Поэтому вряд ли можно говорить о каких-либо универсальных решениях, пригодных для всех типов БТА. Но вместе с тем, реализуемый принцип автоматизации должен базироваться на возможно более универсальных решениях.

Так, в качестве базовой системы программирования для генератора контролирующих программ выбрана система ввода, обработки и документирования (СВОД) [2], потенциально пригодная для разработки программ контроля любой БТА.

Для получения рабочей программы на СВОД необходимо написать на специализированном языке [3] задание на обработку, в котором как раз и будут описаны все контролируемые измерительные каналы. В дальнейшем сборка рабочей программы выполняется автоматически при условии однократной настройки процесса сборки на заданный алгоритм контроля, который, как уже было сказано, не зависит от набора входных воздействий.

Таким образом, задача сводится к автоматизации генерации задания на обработку в соответствии со схемой

расстановки заглушек по входным разъемам приборов в модуле/системе. Перечень заглушек, применяемых для контроля, фиксирован. Их уставки заранее известны и не изменяются. Максимальное число заглушек, используемых при испытании, не превышает 40. В большинстве же случаев (для большинства модулей/систем) необходимо 10—20 заглушек.

Описать расстановку такого числа заглушек значительно проще и быстрее, чем описывать сотни и тысячи каналов. Да и проконтролировать корректность сделанного описания значительно проще.

### Реализация автоматизированного генератора рабочих программ

При детальном рассмотрении задачи оказалось, что наиболее удобной формой описания расстановки заглушек является табличная. При этом каждая строка таблицы описывает одну группу каналов одного прибора и включает набор полей, записываемых в строго определенном порядке: имя прибора, принятое в системе; номер прибора в модуле; адрес опроса прибора; номер группы; режим группы; адрес группы в потоке телеметрической информации; имя заглушки; дополнительные параметры, если необходимо. Если для какой-либо группы значение какого-то поля не определено (например, в температурном коммутаторе деления на группы нет, т. е. весь коммутатор — это одна группа), то в соответствующем поле ставится прочерк.

Отдельно была создана база данных уставок заглушек. Она представляет собой набор текстовых файлов. В каждом файле в табличной форме представлена уставка, состоящая из номера канала в группе, номинального значения и допустимых отклонений измеренного значения от заданного. (Допуска реально нужны для программы контроля измерительных каналов, но по ряду причин задавать их оказалось удобнее всего вместе с номиналами заглушек.). Существует также отдельный файл соответствия, описывающий, заглушке с каким именем (обычно русская аббревиатура) какой файл уставок соответствует. Такая структура базы данных позволяет, с одной стороны, легко добавлять в базу данных новые заглушки (достаточно просто описать новую заглушку в единственном файле соответствия), а с другой стороны, избежать проблем с ограничениями файловой системы по заданию имен файлов.

Генерацию задания на обработку по составленной таблице проще всего оказалось выполнить с применением табличного фильтра *awk*, который входит в стандартный набор программного обеспечения ОС *Linux* (существуют свободно распространяемые реализации *awk* и для ОС *Windows*).

Полученное задание на обработку автоматически транслируется средствами СВОД с получением готовой рабочей программы контроля модуля/системы для одного варианта расстановки заглушек.

Поскольку смена заглушек (если это необходимо) выполняется испытателями вручную, автоматизировать процесс получения контролирующей программы более высокого уровня (сразу для нескольких вариантов расстановки заглушек) в настоящий момент нецелесообразно.

В данное время разработана программа, которая получает на вход табличный файл расстановки заглушек, а на выходе генерирует готовую рабочую программу контроля, избавляя пользователя от необходимости оперировать промежуточными файлами.

Поскольку файл расстановки заглушек имеет жестко заданную структуру, нарушение которой может привести к получению некорректной рабочей программы, в некоторых случаях может оказаться полезной разработка формы шаблона (например, на языке Tcl/Tk) для исключения возможности подобных ошибок. Однако в нашем случае такая форма, будучи разработанной, применения не нашла. Дело в том, что строки файла расстановки заглушек во многом повторяют друг друга. Как следствие, при создании такого файла проще скопировать одну правильно созданную строку столько раз, сколько надо, а затем внести несложные изменения в скопированные строки.

В качестве практического примера рассмотрим генерацию рабочей программы контроля для гипотетического модуля, содержащего три прибора:

- МКА2 — коммутатор 64 аналоговых каналов (разделен на четыре группы);
- МКА3 — коммутатор 128 аналоговых каналов (разделен на четыре группы);
- МКТ2 — коммутатор 32 температурных каналов (все каналы в одной группе).

Файл описания расстановки заглушек будет выглядеть так (опущена одна строка, назначение которой не рассматривается в данной статье):

МКА2	1	1	-	1 1010	ИД1
МКА2	1	2	-	1 2010	ИД2
МКА2	1	3	-	1 401012	ИД3
МКА2	1	4	-	1 3010	ИД4
МКА3	1	1	-	1 401022	ИД31
МКА3	1	2	-	1 501010	ИД32
МКА3	1	3	-	1 501020	ИД33
МКА3	1	4	-	1 501030	ИД34
МКТ2	1	-	-	1 501040	ИТ-Т

Как несложно подсчитать, модуль предоставляет 224 канала. С учетом необходимости описания зависимости измерительных каналов от калибровочных (имеющихся

во всех трех рассматриваемых приборах), размер задания на обработку, написанного на языке СВОД традиционным способом (вручную), будет составлять около 600 строк текста. С применением же автоматизированного генератора рабочая программа контроля создается с использованием только показанного файла расстановки заглушек. Время генерации — меньше одной минуты.

### Заключение

Разработанный автоматизированный генератор рабочих программ контроля БТА прошел апробацию и в настоящее время внедрен в технологический процесс разработки программного обеспечения приемо-сдаточных испытаний и входного контроля модулей и систем БТА "Орбита-IVMO". Применение генератора позволило значительно сократить сроки разработки программного обеспечения, исключить ошибки, связанные с вводом уставок, повысить производительность труда программистов.

Практика показала, что, применяя генератор, один программист способен менее чем за неделю подготовить комплект программного обеспечения для проведения испытаний БТА "Орбита-IVMO" в комплектации любой сложности.

### Список литературы

1. Горбатюк Л. Е., Недошивин С. Н., Победоносцев К. А. и др. Бортовая аппаратура "Орбита-IVMO", опыт ее производства и применения для натурной отработки ракет различных классов // Радиотехнические тетради. 2006. № 33.
2. Чернышов А. В. Система ввода, обработки и документирования измерительной информации рабочего места контроля бортовой телеметрической аппаратуры // Информационные технологии. 2007. № 2. С. 58—60.
3. Чернышов А. В. Язык подготовки заданий на обработку телеметрической информации // Информационные технологии. 2006. № 7. С. 64—69.



## ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В СОЦИАЛЬНЫХ И ЭКОЛОГИЧЕСКИХ СИСТЕМАХ

УДК 378.504

Э. М. Соколов, д-р техн. наук, проф., В. М. Панарин, д-р техн. наук, проф.,  
А. А. Зуйкова, канд. техн. наук, А. В. Бизикин,  
Тульский государственный университет

### Автоматизированная система экологического мониторинга атмосферы при выбросах вредных веществ

*Приведены принципы построения системы автоматизированного мониторинга атмосферы промышленно развитых регионов. Составной частью этой системы является система сбора и отображения экологической информации, которая предназначена для повышения эффективности принятия управленческих решений на основе оперативного и более полного представления информации на электронной карте региона.*

Особенности экологической обстановки в субъектах Российской Федерации и возникающих в них экологических проблем в основном обусловлены местными природными условиями и характером воздействия на них

промышленности, транспорта, коммунального и сельского хозяйства, который зависит от специфики размещения предприятий, их мощностей, применяемой технологии и т. д.

Сложная экологическая обстановка во многих субъектах Российской Федерации — не столько результат кризисных явлений в экономике страны, сколько следствие накопленных за многие десятилетия структурных деформаций хозяйства, приведших к доминированию ресурсо- и энергоемких технологий, сырьевой ориентации экспорта, падению технологической дисциплины, а также к чрезмерной концентрации производства в экономически развитых регионах страны. В тех субъектах Российской Федерации, где сосредоточены предприятия нефтегазовой и горнорудной отраслей, черной и цветной металлургии, экологическая обстановка остается неблагоприятной, сохраняется высокий уровень загрязнения окружающей среды.

При явном снижении неудовлетворительных проб воздушной среды в промышленных городах сложилась неблагоприятная ситуация с состоянием воздушного бассейна вблизи автомагистралей.

Как видно из сказанного выше, проблема чистоты воздуха в городе с повышенной техногенной нагрузкой — проблема комплексная. Для выработки действенных мер по сокращению выбросов необходим тщательный анализ всех источников выбросов в атмосферу на данной территории, учет метеорологических условий и особенностей ландшафта. Применение современной вычислительной техники позволяет обеспечить наиболее эффективный анализ экологической обстановки. К основным преимуществам использования информационных технологий можно отнести:

- сокращение времени и трудоемкости выполнения конкретной задачи за счет автоматизации вычислительных процессов, часто повторяющихся операций и другой рутинной работы;

- современная электроника позволяет создавать компактные автономные устройства сбора экологической информации, которые продолжительное время могут работать без вмешательства человека, фиксируя целый ряд параметров, характеризующих состояние окружающей среды;
- современные каналы передачи информации позволяют связать устройства сбора информации в единую сеть. При этом каналы передачи могут быть различными: проводная линия связи, телефонная линия, радиоканал, сеть Интернет.

Размещение компьютерных точек сбора экологической информации по территории промышленно развитого региона обеспечивает возможность непрерывного наблюдения за состоянием атмосферного воздуха.

Специалисты Кафедры аэрологии, охраны труда и окружающей среды Тульского государственного университета разработали одну из таких автоматизированных систем мониторинга, которая называется "ТулГУ: Экомонитор". Она позволяет проводить эффективную оценку загрязнения окружающей среды г. Тулы выбросами промышленных предприятий.

Автоматизированная система экологического мониторинга предназначена для непрерывного контроля содержания загрязняющих веществ в выбросах промышленных предприятий, измерения метеопараметров, формирования и передачи данных в центр сбора, обработки и хранения информации о состоянии атмосферного воздуха. Структура системы представлена на рис. 1.

Автоматизированная система мониторинга состояния атмосферного воздуха промышленного региона позволяет решать следующие задачи:

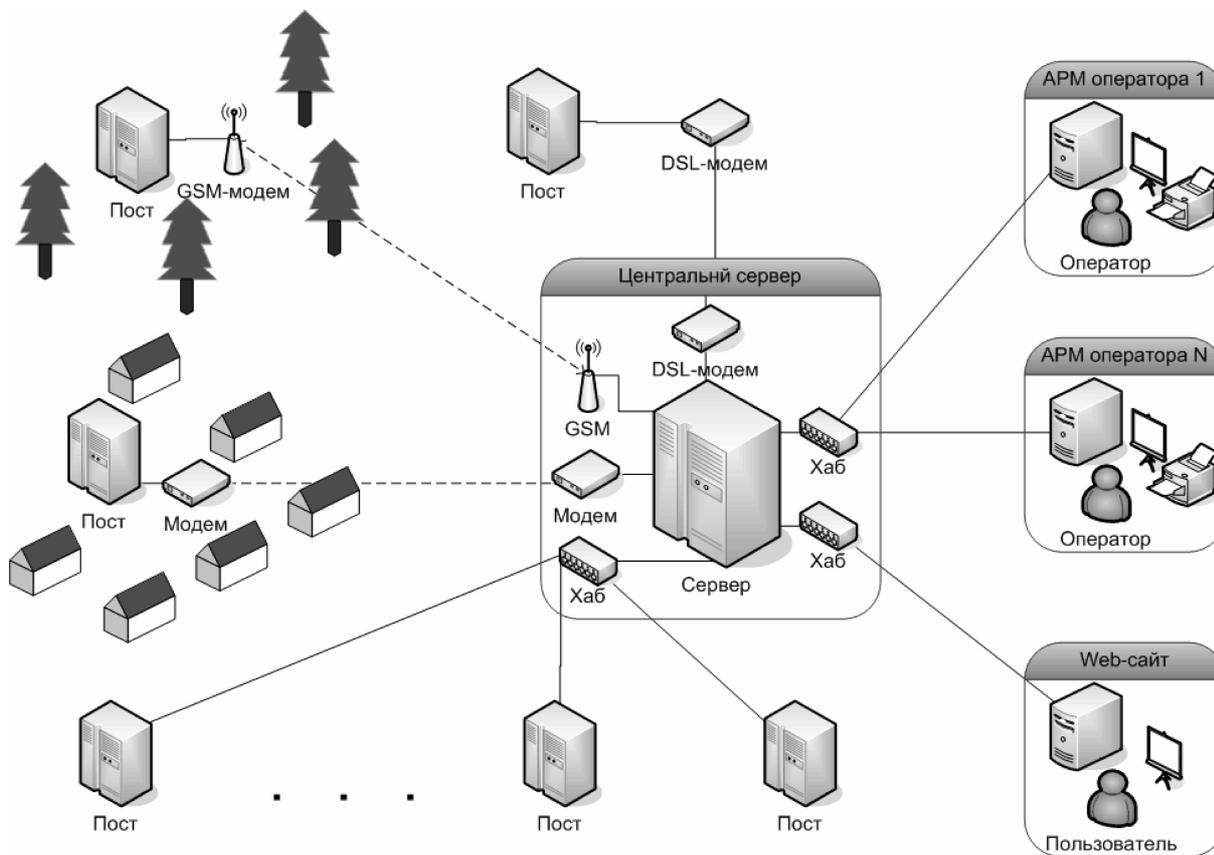


Рис. 1. Структура автоматизированной системы ТулГУ: Экомонитор

- сбор, обработка и хранение информации с датчиков контроля;
- экологический анализ информации;
- выработка управленческих решений по снижению антропогенных воздействий промышленных предприятий на окружающую среду.

В основу построения системы положены следующие основные принципы: *непрерывность* — данные о состоянии атмосферного воздуха в месте расположения датчиков контроля отслеживаются в режиме реального времени, и через небольшие интервалы времени вся информация передается в центр сбора и обработки экологической информации; *стационарность* — датчики контроля состояния атмосферного воздуха выполняются в автономном варианте и после их установки не изменяют своего положения относительно заранее "выбранной системы" координат; *открытость* — обработанная информация о загрязнении воздуха может быть размещена на Интернет-сайте; *модульный принцип* — предполагает, что система включает в себя несколько модулей — модуль контроля загрязнения атмосферного воздуха, модуль сбора и передачи информации (по сети Интернет, по каналу RS-232, по радиоканалу), модуль хранения и обработки информации, центр предоставления информации (рис. 2) [1].

Модульный принцип позволяет увеличивать число модулей контроля загрязнения атмосферного воздуха и распределять их по территории промышленного региона.

Модуль контроля загрязнения атмосферного воздуха содержит датчики содержания веществ в воздухе: оксида углерода; диоксида серы; оксидов азота; пыли. Модуль выполнен таким образом, что возможно использование других датчиков в зависимости от состава веществ, загрязняющих атмосферный воздух в месте расположения модуля. Модуль располагается как вблизи источников выбросов, так и над социально-значимыми объектами региона: детскими садами, школами, больницами, а также на границах санитарно-защитных зон промышленных предприятий.

В качестве первичных средств измерения концентраций вредных выбросов в атмосферном воздухе используются газоанализаторы российского производства типа ДАХ-СО-20, ДАХ-SO<sub>2</sub>-10, газоанализатор ДАХ-NO<sub>2</sub>-20, анализатор пыли МР101М.

Модуль сбора и передачи информации предназначен для сбора информации о степени загрязнения окружающей среды и формирования канала связи для передачи информации на модуль хранения и обработки информации.

После приема информации о степени загрязнения атмосферного воздуха модуль осуществляет передачу данных на модуль хранения и обработки информации по сети Интернет, по каналу связи RS-232 (на расстояние до 1500 м) или по радиоканалу (на расстояние до 3000 м).

Вся информация поступает в центр предоставления информации через модуль хранения и обработки информации. Программное обеспечение центра позволяет в режиме реального времени получать оперативную информацию о степени загрязнения атмосферного воздуха промышленного региона, моделировать процессы загрязнения атмосферы и отображать результаты моделирования на электронной карте региона в виде полей концентраций вредных веществ (мг/м<sup>3</sup>) или долей предельно-допустимых концентраций, а также в виде диаграмм, графиков и таблиц по желанию пользователя.

В центре проводится накопление и архивирование данных измерений и наблюдений, информационный поиск и доступ к архивной информации, анализ и прогноз динамики загрязнений; управление режимами работы автоматизированной системы. Здесь применяется концепция ГИС-технологий для отображения информации о степени загрязнения атмосферного воздуха на электронной карте.

Такое построение системы обеспечивает получение большого объема информации о состоянии атмосферного воздуха. Система позволяет изменять интервал измерения этих параметров от 5 мин до 1 ч. Данные отображаются на терминалах диспетчерских служб и доступны для обработки.

Система сбора и обработки экологической информации состоит из трех основных частей: центр сбора и обработки информации; стационарные экологические посты; устройства связи между экологическими постами и центром сбора и обработки экологической информации [2—4].

Центр сбора и обработки информации представляет собой сервер с соответствующим программным обеспечением и располагается в администрации области. Здесь обрабатывается информация, поступающая со всех ста-

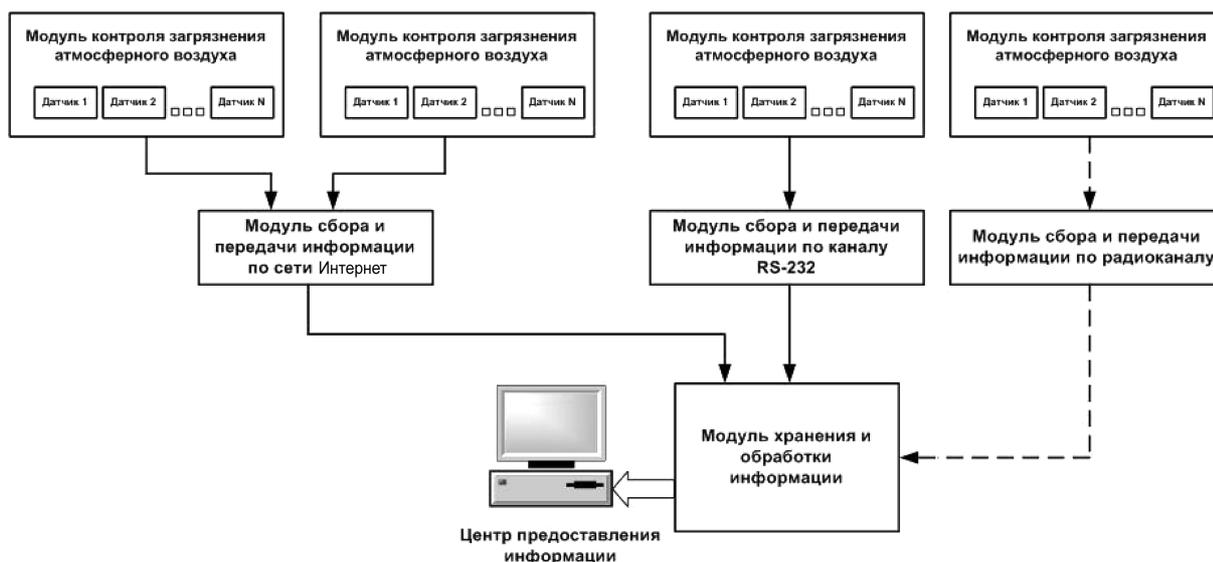


Рис. 2. Модульный принцип построения системы сбора и обработки экологической информации

ционных экологических постов. Поступление информации происходит посредством устройства связи со стационарными экологическими постами.

В администрации области располагается рабочее место оператора, задачей которого является отслеживание информации, поступающей с экологических постов.

Рабочее место оператора представляет собой персональный компьютер, который подключен к центру сбора и обработки экологической информации. Компьютер оператора оснащен специальным программным обеспечением, позволяющим представлять информацию о состоянии атмосферного воздуха в городе в наглядном виде.

На мониторе оператора отображается электронная карта Тульской области. На карте отмечены точки, расположение которых соответствует расположению стационарных экологических постов, находящихся в различных точках города (рис. 3, см. четвертую сторону обложки).

Экологический пост состоит из персонального компьютера с установленным специальным программным обеспечением, блока датчиков, соединенных с контроллером, для измерения различных параметров атмосферного воздуха (концентрации вредных веществ в атмосферном воздухе, температуры окружающей среды, скорости и направления ветра, относительной влажности) и устройства связи с центром сбора и обработки информации.

Датчики формируют специальные сигналы, которые поступают на контроллер, где они преобразовываются и далее передаются на персональный компьютер. Специальное программное обеспечение передает информацию с контроллера на локальный компьютер и записывает в локальную базу данных, затем благодаря устройству связи информация передается в центр сбора и обработки информации.

Система сбора и отображения экологической информации предназначена для повышения эффективности принятия управленческих решений на основе оперативного и более полного представления информации на электронной карте Тульской области.

Система состоит из трех подсистем.

1. Подсистема "*Администрация области*" построена на имеющейся в распоряжении администрации электронной карте Тульской области и компьютерной технике. На электронной карте нанесены промышленные предприятия и другие организации тульского региона. С каждым предприятием будет связана база данных, отражающая информацию о выбросах предприятия, наличии и перемещении опасных и вредных веществ. Также на электронной карте будут отслеживаться перемещения различных объектов (автомобильный и железнодорожный транспорт), перевозящих опасные вещества. Такая база данных в сочетании с электронной картой Тульской области позволит наблюдать на экране монитора зоны экологического загрязнения в районах промышленных предприятий, зоны возможного поражения при аварийных выбросах, оценить последствия и оперативно принять соответствующие управленческие решения, снижающие уровень экологического загрязнения. Подсистема "*Администрация области*" будет функционировать совместно с подсистемой "*Предприятие*".

2. Подсистема "*Предприятие*" устанавливается на вычислительной технике промышленных предприятий региона. Для ее функционирования администрация Тульской области должна будет передать предприятиям региона фрагменты электронной карты Тульской области, где размещены эти предприятия с прилегающими территориями. В подсистеме "*Предприятие*" происходит наполнение баз данных информацией по выбросам конкретного предприятия, отслеживается информация по получению,

переработке и перемещению вредных и опасных веществ. Руководитель предприятия получает возможность визуального наблюдения на электронной карте предприятия и прилегающей территории зон загрязнения, а также возможность компьютерного моделирования различных аварий, которые могут произойти на данном предприятии, что приведет к повышению эффективности взаимодействия сил предприятия и привлеченных сил при реально возникающих аварийных ситуациях. Данная подсистема внедрена на одном из крупнейших российских предприятий, расположенном в Тульской области.

3. Обмен данными между подсистемами "*Администрация области*" и "*Предприятие*" осуществляется по сети Интернет, к которой в настоящее время имеют доступ практически все предприятия Тульского промышленного региона.

Для организации непрерывного контроля ряда экологических параметров, таких как концентрация CO, NO, NO<sub>2</sub>, SO<sub>2</sub>, CL<sub>2</sub>, температура, влажность, направление и скорость движения воздуха и ряда других параметров, разработана подсистема "*Оперативный контроль*", представляющая собой аппаратно-программный комплекс на базе персонального компьютера, связанного с датчиками измерения указанных величин по проводным линиям. Датчики могут располагаться непосредственно на промышленном предприятии, на прилегающих территориях, а также там, где необходим непрерывный контроль, например на территории больниц, парков, музеев. В случае значительного удаления датчиков они будут выполнены в автономном варианте и связаны с базовым компьютером через радиомодем.

При формировании соответствующей сети расположения датчиков на территории города Тулы появляется возможность оперативного контроля загрязняющих выбросов отдельных промышленных предприятий.

Разработанная система автоматизированного мониторинга может быть адаптирована к любому региону и использоваться в экологических службах при администрациях субъектов региона, на промышленных предприятиях для оценки состояния атмосферы, ее загрязнения вредными веществами, а также для более наглядного отображения информации на электронной карте местности и для повышения эффективности принятия необходимых управленческих решений по улучшению экологической обстановки на основе оперативного и более полного представления информации на электронной карте региона.

#### Список литературы

1. **Соколов Э. М.** Автоматизированные системы экологического мониторинга атмосферы промышленно развитых территорий / А. А. Зуйкова [и др.]. — Тула: Изд-во ТулГУ, 2006. — 218 с.
2. **Решетилов А. Н.** Классификация систем мониторинга / А. Н. Решетилов, А. М. Зякун, А. А. Зуйкова // Приоритетные направления развития науки и технологий. Кн. I. Всерос. научно-технич. конф. в 2 кн. / Под общей ред. чл. корр. РАН В. П. Мешалкина. — М.: Тула: Изд-во ТулГУ, 2006. С. 29—30.
3. **Панарин В. М.** Экологический мониторинг — инструмент контроля и охраны окружающей среды / В. М. Панарин [и др.] // Приоритетные направления развития науки и технологий. Кн. I. Всерос. научно-технич. конф. в 2 кн. / Под общей ред. чл. корр. РАН В. П. Мешалкина. — М.: Тула: Изд-во ТулГУ, 2006. — С. 27—28.
4. **Панарин В. М.** Место мониторинга в системе контроля / В. М. Панарин [и др.] // Современные проблемы экологии: Доклады Всероссийской научно-технической конференции. Кн. 2, под общ. ред. чл.-корр. РАН В. П. Мешалкина. — Москва—Тула: Изд-во ТулГУ, 2006. С. 147—148.

**А. С. Герман**, канд. техн. наук,  
Отдел правовой информации  
Верховного Суда РФ

## **Процессы обеспечения информационно- телекоммуникационной среды передачи данных и видеоконференцсвязи в органах государственной службы**

*...Кто инженерства знать не будет,  
тому в чине быть не выше того,  
какой имеет...*

Петр I

*Описаны информационные технологии, которые применяются в судебном делопроизводстве Российской Федерации. В частности, рассматривается применение видеоконференцсвязи в Верховном Суде РФ при рассмотрении кассационных и надзорных жалоб.*

В современных условиях начала XXI века информационный ресурс стал одним из основных факторов, влияющих на функционирование большинства государственных, муниципальных, общественных структур и образований. Предполагается, что информатизация всех сфер жизнедеятельности общества создаст интеллектуальный фундамент для решения широкого круга глобальных проблем. Вместе с тем переход информации в разряд стратегических ресурсов государства и общества остро ставит проблему управления этим ресурсом. Наблюдается устойчивая тенденция увеличения доли автоматизированной генерации, обработки, сбора, хранения, передачи, поиска, представления и воспроизведения информации в процессах управления всех уровней жизнедеятельности, как в деятельности отдельного человека, так и общества в целом [13, 27].

Воздействие возрастающего влияния информационно-телекоммуникационной составляющей было отмечено в Окинавской Хартии глобального информационного общества, принятой лидерами Большой восьмерки 22 июля 2000 года [7].

Уже очевидно, что информационные системы и процессы, телекоммуникационные технологии все более глубоко проникают во все сферы управления государством. Не осталась в стороне и система судебного производства России. Этому способствовал комплекс объективных причин, среди которых основными являются:

- демократизация и все большая открытость внешнему миру российского общества;
- новые экономические условия, сформировавшиеся в Российской Федерации и заставившие руководителей федеральных органов исполнительной, законодательной и судебной власти изыскивать способы эко-

номии средств федерального бюджета, направляемых в их распоряжение.

Объективные причины, показывающие открытость внешнему миру российского общества, способствуют формированию правовых аспектов для использования информационных и телекоммуникационных технологий в судебном производстве<sup>1</sup> Российской Федерации в интересах строгого соблюдения конституционных прав и свобод граждан как внутри государства, так и на международно-народной арене.

В рамках проведения судебной реформы, повышения эффективности деятельности судебной власти в Российской Федерации, оптимального организационно-правового и материально-технического обеспечения судебной системы по поручению Президента РФ была разработана, а затем утверждена Постановлением Правительства РФ от 20 ноября 2001 года № 805 Федеральная целевая программа "Развитие судебной системы России" на 2002—2006 годы, где основной задачей является информационное обеспечение судебной системы. 4 августа 2006 года распоряжением Правительства РФ № 1082-р утвердили Концепцию Федеральной целевой программы "Развитие судебной системы России" на 2007—2011 годы [4, 5].

Следствием указанных причин явилось включение пункта 3 в статью 376 Уголовно-процессуального кодекса<sup>2</sup> Российской Федерации, в соответствии с которым "осужденный, содержащийся под стражей и заявивший о своем желании присутствовать при рассмотрении жалобы или представления на приговор, вправе участвовать в судебном заседании непосредственно либо изложить свою позицию путем использования систем видеоконференцсвязи. Вопрос о форме участия осужденного в судебном заседании решается судом" [2].

Международные правовые аспекты применения технологий представлены на Всемирной встрече на высшем уровне по вопросам информационного общества в Декларации принципов и Плана действий "Построение информационного общества — глобальная задача в новом тысячелетии", проходящей под эгидой ООН и Международного союза электросвязи, первый этап которой состоялся в Женеве в декабре 2003 года, а второй этап — в ноябре 2005 г. в Тунисе [8, 9, 33].

Таким образом, была сформирована правовая база для широкого использования информационно-телекоммуникационной среды передачи данных и видеотехнологии в системе судебного производства, которая, в свою очередь, обеспечила формирование за семь лет в рамках системы судебного производства федеральной системы видеоконференцсвязи (далее — ФВКС).

При этом отметим, что во всем мире судопроизводства видеоконференция применялась только в судопенициарной системе, в основном для заслушивания свидетельских показаний и общения осужденных с адвокатами и родственниками. Именно в России видеоконференция была впервые применена непосредственно для проведения судебных заседаний.

**Информационно-телекоммуникационная среда** — это совокупность аппаратно-программных средств и кана-

<sup>1</sup> Система судебного производства Российской Федерации организационно состоит из двух частей: судебной системы и системы исполнения наказаний, в части, обеспечивающей присутствие заключенных в зале судебного заседания.

<sup>2</sup> Вступил в силу с 1 июля 2003 года.

лов связи, обеспечивающих передачу аудио- и видео-, управляющих и информационных данных между информационными системами, удаленными друг от друга на значительные расстояния.

**Видеоконференция** — это интерактивное аудио- и видеовзаимодействие нескольких удаленных абонентов, при котором между абонентами возможен обмен аудио- и видеoinформацией в реальном масштабе времени.

С помощью технологий было сформировано информационное пространство системы судебного производства, которая связала в единое целое две ее составные части — судебную систему Российской Федерации и исполнительные учреждения Федеральной службы исполнения наказаний России (далее — ФСИН России) Министерства юстиции РФ [2, 18].

Принципиальная схема судебного производства с использованием ФВКС показана на рис. 1.

Формирование ФВКС обеспечивает реализацию конституционных прав наиболее уязвимой части граждан Российской Федерации — осужденных. Эта система позволяет заключенным, не покидая мест лишения свободы, реализовать предоставленное им законом право лично участвовать в судебном заседании с помощью видеоконференцсвязи. Видеотехнология также позволяет государству выполнить взятые им на себя обязательства в части соблюдения сроков рассмотрения кассационных и надзорных жалоб осужденных. Это обстоятельство имеет принципиальное значение, когда речь идет о заключенных, содержащихся в удаленных следственных изоляторах (далее — СИЗО), которых в России большинство [21].

Эксплуатация видеотехнологии за последние семь лет доказала свою эффективность. Использование в судебном производстве видеотехнологии позволило:

- выполнять требования Конституции и федеральных законов Российской Федерации, а также добровольно взятых на себя Российской Федерацией обязательств по международным договорам и соглашениям в части соблюдения сроков рассмотрения кассационных и надзорных жалоб осужденных [1, 7—9];
- избегать межгосударственных инцидентов при этапировании заключенных через территорию Прибалтийских государств из Калининградской области;
- исключить возможность побега заключенных в процессе их этапирования к месту проведения суда и обратно;
- экономить значительную часть бюджетных средств (в масштабе страны миллиарды рублей);
- обеспечить более равномерную загрузку судов и увеличить число рассматриваемых кассационных, надзорных жалоб.

По поводу основной снятой проблемы можно утверждать, что предотвращение побега заключенного из-под стражи сохраняет жизнь не только ему и конвоирам, но и мирным жителям [20, 22].

На развитие и совершенствование системы видеоконференцсвязи выделяются значительные средства федерального бюджета. Однако дальнейшее развитие технологий сдерживается из-за отсутствия недорогих качественных каналов связи как на региональном уровне, так и на общероссийском, а также из-за отсутствия единой государственной политики в области подсистем передачи данных.

Таким образом, необходимость дальнейшего совершенствования и развития федеральной системы видеоконференцсвязи формирует социальный заказ на создание межведомственной телекоммуникационной среды передачи данных для рассмотрения кассационных и надзорных жалоб в режиме удаленного присутствия осужденного. В силу специфики применения видеотехнологий при территориальной разобщенности судов общей юрисдикции и исправительных учреждений Федеральной службы исполнения наказаний России по всей стране возникла необходимость в создании единой подсистемы связи и передачи данных в рамках Государственной автоматизированной системы "Правосудие" (ГАС "Правосудие"), фундаментальной основой которой послужила телекоммуникационная инфраструктура видеоконференцсвязи Верховного Суда РФ и судов общей юрисдикции [4, 5].

Государственная автоматизированная система Российской Федерации "Правосудие" — это территориально распределенная автоматизированная информационная система, предназначенная для формирования единого информационного пространства Верховного Суда РФ, судов общей юрисдикции и системы Судебного департамента при Верховном Суде РФ, обеспечивающая информационную и технологическую поддержку судопроизводства на принципах поддержания требуемого баланса между потребностью граждан, общества и государства в свободном обмене информацией и необходимыми ограничениями на распространение информации.

Сказанное привело к необходимости анализа телекоммуникационного оборудования для совершенствования подсистем видеоконференцсвязи и передачи данных в ГАС "Правосудие". Неотъемлемым компонентом указанной системы является среда передачи аудио-, видео- и информационных потоков данных.

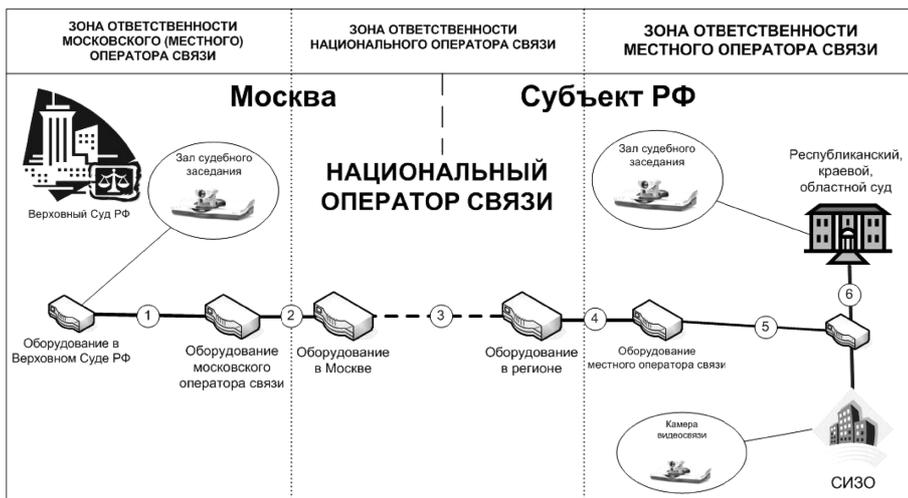
Телекоммуникационная среда передачи данных обеспечивает непрерывность проведения судебного процесса с разными исправительными учреждениями в регионах Российской Федерации круглосуточно.

Однако для полноценной системы необходимо сформировать единую межведомственную телекоммуникационную среду во времени и пространстве (с учетом их территориальной разобщенности). Территориальная разобщенность, с одной стороны, очень влияет на схемы подключения к различным операторам связи региональных судов и исправительных учреждений, которые обеспечивают на всей территории страны одинаковый доступ



Рис. 1. Принципиальная схема судебного производства Российской Федерации с использованием средств видеоконференцсвязи

## Схема подключения регионов для видеоконференции Верховного Суда РФ



**Рис. 2.** Схема подключения субъектов РФ к федеральной системе видеоконференцсвязи Верховного Суда РФ:

1 — канал связи "последняя миля" (не менее 2 Мбит/с) — зона ответственности московского (местного) оператора связи; 2 — городское продление (стык) между московским и национальным оператором связи — зона ответственности национального или московского (местного) оператора связи; 3 — канал связи между Москвой и субъектами РФ (не менее 1 Мбит/с) — зона ответственности национального оператора связи; 4 — городское продление (стык) между национальным и местным операторами связи — зона ответственности национального или местного оператора связи; 5 — канал связи "последняя миля" не менее 1 Мбит/с) — зона ответственности местного оператора связи; 6 — канал связи (не менее 1 Мбит/с) — зона ответственности местного оператора связи (суда общей юрисдикции)

для реализации прав осужденным, с другой стороны, она объективно обусловила временной интервал проведения судебных процессов [30, 31, 34]. Основная схема подключения субъектов Российской Федерации к ФВКС представлена на рис. 2.

Реализация социального заказа на новые информационные технологии в судебной системе потребовала решения на концептуальном уровне вопросов, связанных как с анализом и выбором специального оборудования и решением этих задач на каждом этапе работы, так и с организацией среды передачи данных общепринятыми методами на основе коммуникационной инфраструктуры конкретного региона [23, 24, 26, 35].

В связи с этим следует отметить ряд противоречий, которые должны быть разрешены при организации единой межведомственной телекоммуникационной среды передачи данных. Это противоречия между:

- потребностью в соблюдении добровольно взятых на себя государством обязательств по международным договорам и соглашениям в части соблюдения сроков рассмотрения кассационных жалоб осужденных;
- наличием разнородных технических средств видеотехнологий и единой политикой обеспечения оборудованием;
- необходимостью реализации конституционных прав наиболее уязвимой части граждан России — осужденных — и территориальной разобщенностью;
- потребностью судов общей юрисдикции в рассмотрении кассационных и надзорных жалоб и отсутствием качественных каналов связи до региональных исполнительных учреждений.

Эти противоречия возникли при практической эксплуатации технологии видеоконференцсвязи. Обратимся

к истории зарождения и развития федеральной системы видеоконференцсвязи Верховного Суда РФ, которые были вызваны объективной необходимостью обеспечить конституционные права и свободы граждан Российской Федерации, выполнение требований законов Российской Федерации в части ведения судопроизводства. Закон предусматривает обязательное участие в кассационных и надзорных судебных заседаниях всех осужденных, заявивших об этом ходатайство [2]. В мае 1999 года Челябинский областной суд приступил к созданию первой в стране пилотной видеоконференцсети для дистанционного участия осужденных в кассационных судебных заседаниях. Принятая схема организации канала видеоконференцсвязи соответствовала мировым требованиям и в полном объеме решала необходимую задачу.

Сеансы связи между залом суда и следственным изолятором проводились в соответствии с протоколом Н.323 (видеоконференция в сетях с коммутацией пакетов) на скорости 512 кбит/с. Как окончное

оборудование видеоконференцсвязи так и Ethernet-мосты позволяли еще более увеличить полосу для сеанса видеоконференцсвязи (до 1,5 Мбит/с), однако предварительные тесты показали, что соединение уже на скорости 512 кбит/с обеспечивает хорошую динамику и качество аудио- и видеопотоков.

18 ноября 1999 года впервые в России состоялся суд с использованием технологии видеоконференцсвязи (рис. 3). Осужденный Муртазин стал первым, чью кассационную жалобу в режиме видеоконференции рассмотрел Челябинский областной суд. Идея применения технологии для судебного рассмотрения кассационных



**Рис. 3.** Первое судебное заседание с использованием видеоконференцсвязи 18 ноября 1999 года

жалоб заключенных принадлежит Председателю Челябинского областного суда. ОАО "Связьинформ" г. Челябинска обеспечила канал связи между залом судебного заседания областного суда и камерой следственного изолятора. Подбор и наладка оборудования видеоконференций были выполнены специалистами компании "СТЭЛ — Компьютерные Системы".

В связи с полученным в короткие сроки положительным результатом от использования этой технологии с 2000 года началось формирование федеральной системы видеоконференцсвязи в Верховном Суде РФ. Первое слушание дел в Верховном Суде РФ в режиме удаленного присутствия осужденных было проведено 19 апреля 2000 года. Непрерывный сеанс связи Верховный Суд РФ—СИЗО-77/3 Главного управления исполнения наказаний (далее — ГУИН<sup>3</sup>), г. Москва, продолжался около трех часов. Судебной инстанцией было рассмотрено три уголовных дела. Система видеоконференцсвязи продемонстрировала высокое качество и стабильность работы. Никаких срывов связи или заметных артефактов на изображении отмечено не было.

16 мая 2001 года состоялось первое в России судебное заседание с использованием видеоконференцсвязи между Верховным Судом Российской Федерации и следственным изолятором ГУИН Министерства юстиции в городе Челябинске (т. е. между субъектами России) при кассационном рассмотрении уголовного дела.

В судебном заседании участвовал осужденный, находившийся в это время на значительном расстоянии от зала суда в исправительном учреждении в г. Челябинске, т. е. за 1763 километра. Судебная коллегия заседала в Верховном Суде РФ, подсудимый следил за процессом из Челябинска. Он имел возможность непосредственно заявлять свои ходатайства, судьи видели и слышали осужденного, задавали ему вопросы. Это было достигнуто посредством системы видеоконференцсвязи, налаженной в Верховном Суде РФ и в Челябинском областном суде по совместной инициативе Председателя Верховного Суда РФ В. М. Лебедева и Председателя Челябинского областного суда Ф. М. Вяткина (рис. 4).

Представители судов Урала и судов соседствующих с Уралом областей и автономных округов, а также американские юристы, приехавшие изучить опыт российского правосудия, наблюдали за ходом заседания с третьей точки видеоконференции — из зала Челябинского областного суда.

В сентябре—октябре 2002 года были проведены первые судебные заседания по рассмотрению уголовных дел со следственными изоляторами г. Тюмени (более 2000 км) и г. Хабаровска (8533 км) с использованием цифровых те-

<sup>3</sup> С 1 января 2005 года ГУИН Министерства юстиции РФ переименовано в Федеральную службу исполнения наказаний России (ФСИН России).



Рис. 4. Кадры из репортажа телепрограммы "Вести" РТР 16 мая 2001 года



Рис. 5. Кадры из репортажа телепрограммы ОРТ и НТВ 24 апреля 2003 года

лефонных каналов связи ISDN Сети делового общения "Искра-2". 5 марта 2003 года к федеральной системе видеоконференцсвязи подключилась Кемеровская область.

В феврале 2003 года Верховный Суд РФ ввел в действие еще два зала видеоконференцсвязи: зал Президиума и зал Военной коллегии. С начала 2003 года в федеральную систему подключен Московский областной суд.

24 апреля 2003 года между Калининградским следственным изолятором ГУИН Министерства юстиции РФ и Верховным Судом РФ прошел первый сеанс видеоконференцсвязи (рис. 5). В Калининграде проблему этапирования в Москву ощущают особенно остро: литовские власти ввели запрет на проезд вооруженных караулов по территории страны, а отправлять заключенных пассажирскими самолетами — удовольствие недешевое.

28 мая 2003 года проведен первый пробный сеанс многоточечном режиме с регионами Российской Федерации. Данная технология позволяет выводить на один экран сразу несколько московских и региональных следственных изоляторов. Сотрудники СИЗО ГУИН МЮ РФ Калининградской и Кемеровской областей участвовали в подготовке сеанса и настройке аппаратуры для проведения судебных заседаний. При участии в уголовном деле нескольких человек, содержащихся в разных следственных изоляторах Министерства юстиции РФ, теперь нет необходимости их этапировать в один следственный изолятор [14, 17].

В настоящее время в России использование информационных и коммуникационных технологий для обеспечения участия в судебном заседании осужденных и подследственных является единственным средством соблюдения конституционных прав данной категории граждан страны, содержащихся в местах лишения свободы,

в части соблюдения сроков рассмотрения судом их кассационных и надзорных жалоб.

Система видеоконференцсвязи была представлена Президенту РФ Владимиру Владимировичу Путину 22 июня 2006 года, который оценил эффективность и одобрил применение информационных технологий в судебной власти.

Видеоконференция и телекоммуникационная среда передачи данных внедрены в Верховном Суде РФ, Федеральной службе исполнения наказаний России Министерства юстиции РФ, более чем в 90 судах общей юрисдикции субъектов Российской Федерации, в образовательных и медицинских учреждениях, в силовых структурах России, в Министерстве юстиции Финляндии, а также практические основы видеоконференцсвязи приняты и внедрены в других государствах.

#### Список литературы

1. Конституция Российской Федерации от 12 декабря 1993 года // Российская газета. 25.12.1993. № 237.
2. Уголовно-процессуальный кодекс Российской Федерации от 18 декабря 2001 года // Российская газета. 22.12.2001. № 249.
3. Указ Президента РФ № 611 от 12 мая 2004 года "О мерах по обеспечению информационной безопасности Российской Федерации в сфере международного информационного обмена // Собрание законодательства РФ. 17.05.2004. № 20. С. 1938.
4. Постановление Правительства Российской Федерации от 20 ноября 2001 года № 805 "О федеральной целевой программе "Развитие судебной системы России" на 2002—2006 годы // Собрание законодательства РФ. 03.12.2001. № 49. С. 4623.
5. Распоряжение Правительства РФ от 4 августа 2006 года № 1082-р. О концепции Федеральной целевой программы "Развитие судебной системы России" на 2007—2011 годы. Собрание законодательства РФ. 04.08.2006 года. № 33. С. 3652.
6. Закон "О государственной тайне" от 21 июля 1993 года № 5485-1 // Собрание законодательства РФ. 13.10.1997 года. № 41. С. 8220—8235.
7. Окинавская Хартия глобального информационного общества от 22 июля 2000 года. Хартия стран Большой восьмерки (G8). Япония, о. Окинава // Дипломатический вестник. 2000. № 8. С. 51—56.
8. Международный документ WSIS-03/GENEVA/DOC/4-R. Декларация принципов. Построение информационного общества — глобальная задача в новом тысячелетии. Всемирная встреча на высшем уровне по вопросам глобального информационного общества. — Женева, 2003 г. — Тунис, 2005 г. — 24 с.
9. Международный документ WSIS-03/GENEVA/DOC/5-R. План действий. Всемирная встреча на высшем уровне по вопросам глобального информационного общества. — Женева, 2003 г., Тунис, 2005 г. 32 с.
10. Белов В. Ю., Герман А. С., Морозов В. Л. и др. Информационные технологии и социальные институты // Сборник статей. М.: Изд-во РАГС, 2002. 70 с.
11. Бутлицкий И. М. Кабели местных телефонных сетей как среда передачи для технологии DSL. М.: Альварес Паблшинг, 2005. 96 с.
12. Виноградов М. В., Шуа Б. Р. Терминалы видеоконференцсвязи: Руководство пользователя. Stel pv25/stel, pvl50/stel, pv384/stel, gv8000, 2001, 2002 VCON Inc. All Rights Reserved. М.: Стэл — Компьютерные Системы, 2003. 138 с.
13. Герман А. С. Информационное противоборство: условия функционирования современных информационных систем управления // Сб. тр. Н. Новгород: Изд-во ВГИПА, 2002. Вып. 5. Ч. 2. 277 с.
14. Герман А. С. Корпоративная система дополнительного образования — необходимые условия эффективной работы системы ВКС Верховного Суда РФ // Сб. тр. XXII Межведомственной научно-технической конференции. Том 3. Серпухов: Изд. Министерство обороны РФ, 2003. 374 с.
15. Герман А. С. Формирование требований к корпоративной системе дополнительного образования в области информационных и коммуникационных технологий Верховного Суда РФ // Сб. тр. XXII Межведомственной научно-технической конференции. Том 3. Серпухов: Министерство обороны РФ, 2003. — 374 с.
16. Герман А. С., Гречишников А. В. Видеоконференцсвязь в органах государственной службы и коммерческих структурах // Сб. ст. М.: Изд-во РАГС, 2004. С. 91—103.
17. Герман А. С., Крюков С. А. Пять лет на службе правосудия // Адвокатские вестн. 2005. № 4 (54). С. 20—22.
18. Герман А. С. Видеотехнологии: пять лет на службе правосудия // Технологии и средства связи. 2005. № 2 (47). С. 68—69.
19. Герман А. С. Руководство пользователя. Операторам видеоконференцсвязи судов общей юрисдикции и исправительных учреждений Федеральной службы исполнения наказаний России: Межведомственная инструкция. М.: Верховного Суда РФ — Министерства юстиции РФ, 2005. Вып. 2. 34 с.
20. Герман А. С. Мультимедийные технологии на службе правосудия // Телемультимедиа. 2005. С. 3—5.
21. Герман А. С. Телекоммуникационные технологии на службе правосудия // Государственная служба. 2005. № 4 (36). С. 42—44.
22. Герман А. С. Информационно-аналитические средства поддержки принятия решения и ситуационные центры // Материалы научно-практической конференции. М.: Изд-во РАГС, 2006. С. 297—300.
23. Данчул А. Н., Домрачев С. А. Процессно-ориентированные методы анализа систем: Учебно-методическое пособие. М.: Изд-во РАГС. 1997. 49 с.
24. Данчул А. Н., Аскеров Т. М., Бородько В. П., Гречишников А. В. и др. Информатика: Словарь. Термины и определения по информатике. М.: Термика, 2004. 123 с.
25. Данчул А. Н. Информатика: Учебник для слушателей вузов. М.: Изд-во РАГС, 2004. 525 с.
26. Журавлев Е. Е., Надчий А. И. Стандартизация обеспечения информационных систем: Учебное пособие. М.: Изд-во Российской экономической академии им. Г. В. Плеханова, 2000. 95 с.
27. Киселев В. Д., Евсиков О. В., Кислицин А. С. Защита информации в современных системах ее передачи и обработки: Учебное пособие. М.: СОЛИД, 2000. 20 с.
28. Мирошниченко В. М. Организация управления и обеспечение национальной безопасности Российской Федерации: Учебное пособие М.: Экзамен, 2002. 256 с.
29. Никитов В. А., Орлов Е. И., Старовойтов А. В., Савин Г. И. Информационное обеспечение государственного управления. М.: Славянский диалог, 2000. 415 с.
30. Петров А. В., Тихомиров М. М., Федулов Ю. Г. Применение ситуационных центров в региональном управлении: Учебно-методическое пособие. М.: Изд-во РАГС, 1999. 130 с.
31. Попов И. И. Автоматизированные информационные системы (по областям применения): Учебное пособие. М.: Изд-во Российской экономической академии им. Г. В. Плеханова, 1998. 103 с.
32. Райков А. Н. Интеллектуальные информационные технологии: Учебное пособие. М.: Изд-во МИРЭА, 2000. 96 с.
33. Смирнов А. И. Информационная глобализация и Россия: вызовы и возможности. М.: ИД Парад, 2005. 392 с.
34. Соколов Н. А. Принципы построения телекоммуникационных систем. М.: Альварес Паблшинг, 2003. 128 с.
35. Тихомиров М. М. Системы информационной и интеллектуальной поддержки управленческой деятельности в структурах государственной службы. М.: Изд-во РАГС, 1995. 185 с.



# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

УДК 004.31

**Е. Р. Пантелеев**, д-р техн. наук, **В. А. Суворов**,  
Ивановский государственный  
энергетический университет

## Оперативный анализ действий студента в среде Web-обучения

*Рассматриваются вопросы разработки комплекса методов и средств оперативной аналитической обработки действий студента в среде Web-обучения. Представлены архитектура комплекса, обоснована модель хранения данных, описан программный интерфейс заполнения хранилища и методы аналитической обработки. Приведены примеры применения комплекса для анализа результатов подготовки студентов.*

В течение последних пятнадцати лет в сообществе разработчиков интеллектуальных обучающих систем (ИОС) получил широкое распространение конструктивистский подход к организации и использованию учебно-контролирующего материала [1, 2]. Конструктивизм — это теория, которая утверждает, что студент активно конструирует предметные знания в процессе решения прикладных задач, а не выступает в роли пассивного приемника и хранилища готовых знаний, поставляемых преподавателем. В рамках этой парадигмы среда компьютерного обучения выступает не как поставщик готовых решений, а как тренер. Ее основная функция состоит в создании проблемных ситуаций, в которых студент должен применить свои знания для получения решений, и контрпримеров, побуждающих к переосмыслению этих решений. Собственно, именно поддержка многошагового процесса решения задач, которая выражается в обеспечении контекстной обратной связи на границах шагов решения, и является главным отличительным признаком ИОС [2]. Новый взгляд на функции среды компьютерного обучения требует ее радикальной интеллектуализации, основанной на формальном представлении предметных знаний и управлении процессом обучения в функции действий студента.

Эволюция подходов к интеллектуализации компьютерного обучения повлекла за собой появление систем, использующих онтологии (машинно-интерпретируемые формулировки основных понятий предметной области и отношений между ними) и оверлейные модели пользователя в качестве информационной основы адаптивного управления процессами решения задач [4, 5]. Использование онтологий в системах Web-обучения для деклари-

рования явных допущений о структуре предметных знаний и оверлейных моделей для содержательной интерпретации действий студента стало общеупотребительным [2—6]. Однако построение адаптивных механизмов только на базе оперативной транзакционной обработки персональных действий в контексте онтологической модели ограничивает возможности управления обучением. Данная информационная основа не позволяет выявить такие важные закономерности, как, например, наличие и сила вероятностной связи между концептами предметных знаний в модели студента, типизация ошибок и стилей поведения студентов. Это существенно затрудняет решение задач компоновки информационных ресурсов обучения и контроля, обеспечивающих возможности для достижения эталонного уровня подготовки, с учетом оценок эффективности управления процессами обучения и контроля во внешнем контуре схемы "планирование—реализация плана качества" [8].

Адекватную информационную основу для решения этих задач дает использование хранилищ данных и технологий их оперативной аналитической обработки (*online analytical processing, OLAP*) [9]. Примерами немногочисленных пока попыток использования OLAP-технологий для анализа эффективности являются работы [6, 7]. Авторами [7], например, предложен подход к аналитической обработке данных о действиях студентов в процессе решения задач медицинской диагностики (ИОС SlideTutor). В частности, методами интеллектуального анализа данных были идентифицированы следующие типовые варианты поведения студентов:

- обучение на собственных ошибках (*failure-driven behavior*);
- предварительное консультирование с ИОС после обдумывания задания (*hint-driven behavior*);
- "игровое" поведение (*gaming behavior*), когда студент провоцирует систему на подсказки путем необдуманных случайных ответов или использует сообщения ИОС о частичной правильности ответа для "подгонки" методом перебора вариантов, что, как правило, не способствует обучению.

Полученные результаты авторы предполагают использовать для корректировки набора задач (*case-set*) и интеллектуального выделения информационных ресурсов SlideTutor на основе оперативно выявляемых типов поведения.

Недостатком предложенного в [7] решения является его интеграция в конкретную ИОС и вытекающая отсюда невозможность повторного использования в других системах. Преимущество повторного использования достигается применением клиент-серверных архитектур хранения и обработки данных об активности пользователя. Их характерные особенности [10]:

- информация о действиях пользователя размещена в интегрированном хранилище и может быть одновременно предоставлена нескольким приложениям;
- информация, размещенная в хранилище одним приложением, может быть предоставлена другому приложению;
- информация о действиях пользователя хранится в избыточном и непротиворечивом формате;
- сервер предоставляет пользователю средства защиты данных и контроля доступа.

Пример реализации клиент-серверной архитектуры хранилища данных в сети Интернет (*User Modeling Server CUMULATE*) рассматривается в [11] (см. также [http://www.sis.pitt.edu/~paws/system\\_cumulate.htm](http://www.sis.pitt.edu/~paws/system_cumulate.htm)). Данная реализация предоставляет пользователю программный Web-интерфейс наполнения хранилища. Однако интерфейсы аналитической обработки в рамках этого проекта не реализованы. Между тем потребность в разработке методов и средств Web-OLAP, ориентированных на решение задач анализа процессов и результатов Web-обучения в их взаимосвязи, безусловно, существует. Практический интерес представляют ответы на такие, например, вопросы.

- Как распределены дефекты подготовки на оверлейной модели студента (группы студентов)?
- Какова дискриминативность и надежность процедур контроля?
- Как соотносятся результаты подготовки студентов по различным программам (или результаты по одной и той же программе по годам и/или группам)?
- Какова эффективность процедур тренажа, выраженная отношением процента правильных ответов (с возможной фильтрацией по концептам онтологической модели) в режимах тренажа и контроля?
- Какова эффективность управления навигацией, выраженная отношением числа активированных ссылок к числу правильных ответов?

С учетом содержательного аспекта Web-OLAP можно считать актуальной задачей разработки хранилища и средств OLAP, которые удовлетворяют следующим требованиям [13]:

- наличие программного интерфейса заполнения хранилища;
- наличие пользовательского интерфейса для построения нерегламентированных запросов к хранилищу данных;
- оперативное выполнение запросов;
- открытые спецификации на интерфейсы, сервисы и поддерживаемые форматы данных;
- минимальный трафик "клиент—сервер";
- нулевое администрирование клиента OLAP, простота освоения инструментов анализа;
- наглядное представление результатов выполнения запросов средствами деловой графики.

Вопросам разработки клиент-серверной архитектуры хранилища, программных интерфейсов его заполне-

ния, а также Web-интерфейсов аналитической обработки, ориентированных на пользователя с отсутствием навыков работы с OLAP-приложениями и разнообразными потребностями в анализе активности студентов в среде Web-обучения (авторы курсов, кураторы программ обучения, администраторы), и посвящена настоящая статья.

Взаимодействие компонентов комплекса оперативной аналитической обработки со средой Web-обучения показано на рис. 1. Собственно говоря, взаимодействие двух подсистем имеет место только на этапе сбора информации, когда информация о действиях студента, представленная в формате событий, поступает на OLAP-сервер (сервер аналитики), который размещает ее в хранилище OLAP. На этапе аналитической обработки данных подсистема обучения не связана с подсистемой аналитики ни программно, ни через разделяемую информацию. Данные о событиях могут быть извлечены из хранилища и представлены в удобной для анализа форме с помощью Web-интерфейса клиента аналитики. Отсутствие взаимодействия системы Web-обучения с сервером аналитики на этапе обработки данных оправдано самим характером решений, принимаемых на основе OLAP. Это, как правило, решения контура планирования стандартов PDCA [8], связанные с пересмотром приоритетов целей обучения и соответствующим перераспределением информационных ресурсов, которые принимаются в режиме "оффлайн".

Рассмотрим модели и методы решения задач сбора и аналитической обработки данных, реализованные в рамках предлагаемого подхода, более подробно.

Информационную основу решения задачи сбора и обработки данных о действиях студента в среде Web-обучения составляет модель его активности, представленная в виде упорядоченной во времени последовательности дискретных событий. Под *событием*  $E_i$  понимается акт взаимодействия студента с интерактивным элементом учебного или контролирующего материала учебника, который инициирует передачу параметров этого взаимодействия на сервер. К таким взаимодействиям относятся активация навигационной ссылки на учебный материал или задание контроля, а также нажатие кнопки на форме контроля знаний. Поэтому целесообразно различать следующие типы событий (в скобках указаны их атрибуты  $A_i$  и измерения  $D_i$ ):

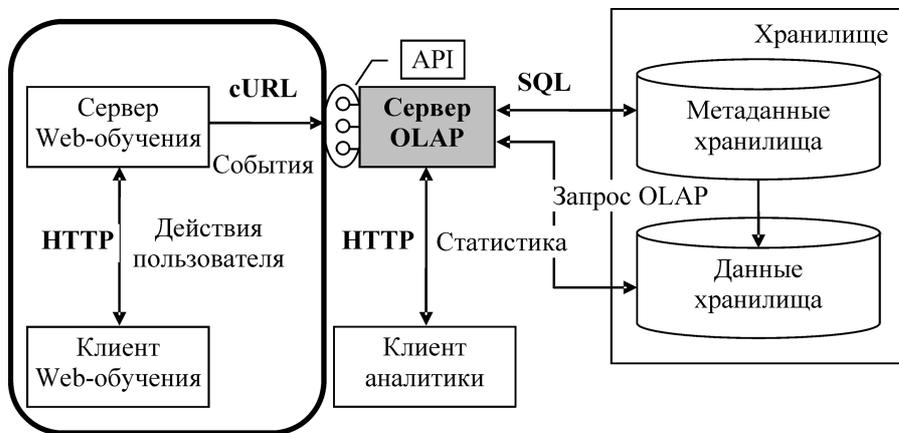


Рис. 1. Взаимодействие средств OLAP со средой Web-обучения

- активизация ссылки (время, пользователь, группа, программа, тип источника ссылки, объект-адресат ссылки);
- ответ на вопрос (время, вопрос, пользователь, группа, программа, модуль, оцениваемый элемент знаний, мера правильности ответа);
- инициализация/завершение процедуры контроля (время, флаг своевременности, тип события, пользователь, группа, программа, модуль).

Поддерживаемые сервером аналитики типы дискретных событий и их измерения представлены на рис. 2<sup>1</sup>.

Программный интерфейс сбора статистики представлен на сервере аналитики сценарием обработки событий. Передача этому сценарию данных, предварительно преобразованных к формату URL, осуществляется по протоколу HTTP (методы POST или GET). Вот так, например, выглядит заголовок сообщения,

включающий список параметров события "Ответ на вопрос" (`type_msg = 0`), которые направляются локальному серверу статистики:

```
http://127.0.0.1/statistics/addition/interface.php?type_mse = 0
&user_name = "Суворов%20Виталий"&group_name = "5-42"&program_name = "Методы%20сортировки%20и%20поиска"&module_name = "Тест_сортировки"&concept_name = "Сортировка%20Шелла"&question_name = "Вопрос1"&validity = 1.
```

При отсутствии хотя бы одного необходимого параметра или при недопустимом значении любого переданного параметра запись события в базу не проводится.

Как следует из приведенных выше определений типов событий, каждое событие  $E_i$  определено в своем базе измерений  $D_j$ :  $E_i = f(D_j)$ , причем  $\forall i, j: D_i \cap D_j \neq \emptyset$ . Это обстоятельство определило выбор поликубической модели хранилища в виде нескольких гиперкубов с различной размерностью и с различными измерениями в качестве их граней (рис. 2). Такое решение более экономично по сравнению с "монолитным" гиперкубом, так как позволяет сократить число "пустых" ячеек.

Web-интерфейс аналитической обработки (показан в средней части рис. 3) включает в себя следующие основные элементы:

- панель управления для выбора типа события, относительно которого выполняется запрос, и настройки параметров построителя запросов (размеры диаграммы, глубина трехмерной прорисовки, цвет заголовка, цвета фона картинки, цвета фона диаграммы, цвета линий сетки, цвет бордюра, шрифт заголовка, число рядов подписей и угол их наклона);

<sup>1</sup> Кроме явно определяемых в формате события атрибутов и измерений, сервер аналитики в качестве "скрытого" измерения фиксирует IP-адрес клиента, что позволяет при аналитической обработке фильтровать события по их источнику в предположении, что IP-адрес однозначно идентифицирует сервер Web-обучения.

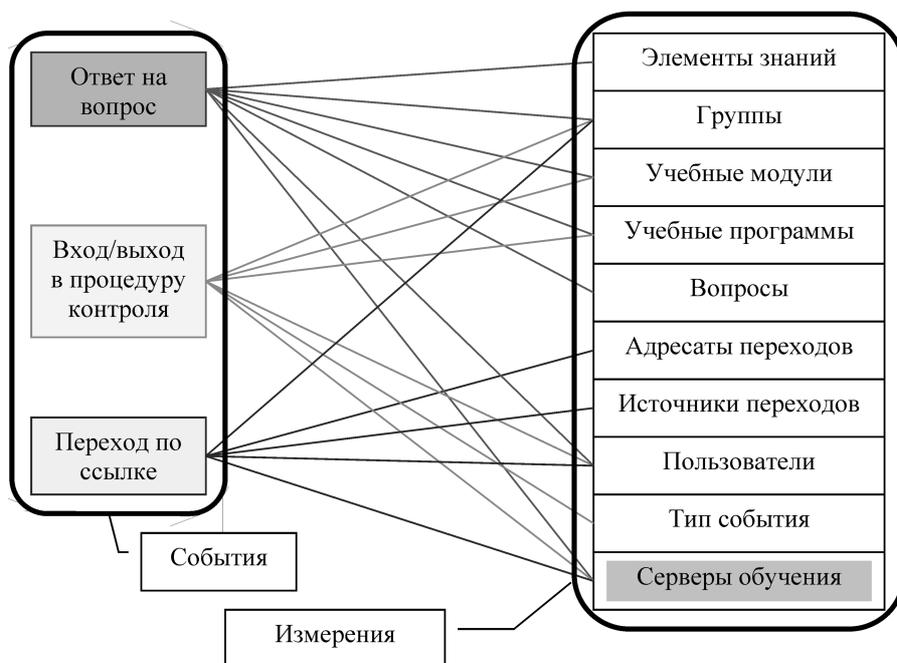


Рис. 2. Виды активности студента и их атрибуты

- панель запросов для выбранного типа события, включающая три закладки:
  - "Мастер запросов";
  - "SQL-консоль";
  - "Сохраненные выборки".

Наибольший интерес представляет закладка "Мастер запросов". Запрос пользователя формируется путем заполнения стандартного SQL-шаблона `SELECT... FROM... WHERE... GROUP BY...` информацией об анализируемых событиях, их измерениях и способах агрегирования атрибутов. Информационную базу решения этой задачи составляет метаописание взаимосвязи событий  $E$  и измерений  $D$  поликуба. Метаописание реализовано в виде явно определенного отношения  $R: (E \times D) \rightarrow \{ \}$ , *утверждение\_SQL*, областью значений которого являются фрагменты SQL-утверждений, используемые для заполнения шаблона. Реляционная модель метаописания представлена в верхней части рис. 3. Ниже приведена общая схема заполнения шаблона с использованием метainформации.

#### 1. SELECT

- а) наименование вычисляемого атрибута события;
- б) наименования координатных осей (измерений), в которых вычисляется атрибут.

#### 2. FROM

- а) названия таблиц измерений (поле "Название в базе" таблицы "Измерения");
- б) название таблиц, участвующих в фильтрах (поле "Название в базе" таблицы "Измерения");
- с) название таблицы, содержащей вычисляемый атрибут (поле "Название в базе" таблицы "События").

#### 3. WHERE

- а) описание связи таблицы событий с таблицами измерений, участвующими в запросе (поле "Текст SQL" таблицы "Измерения событий");

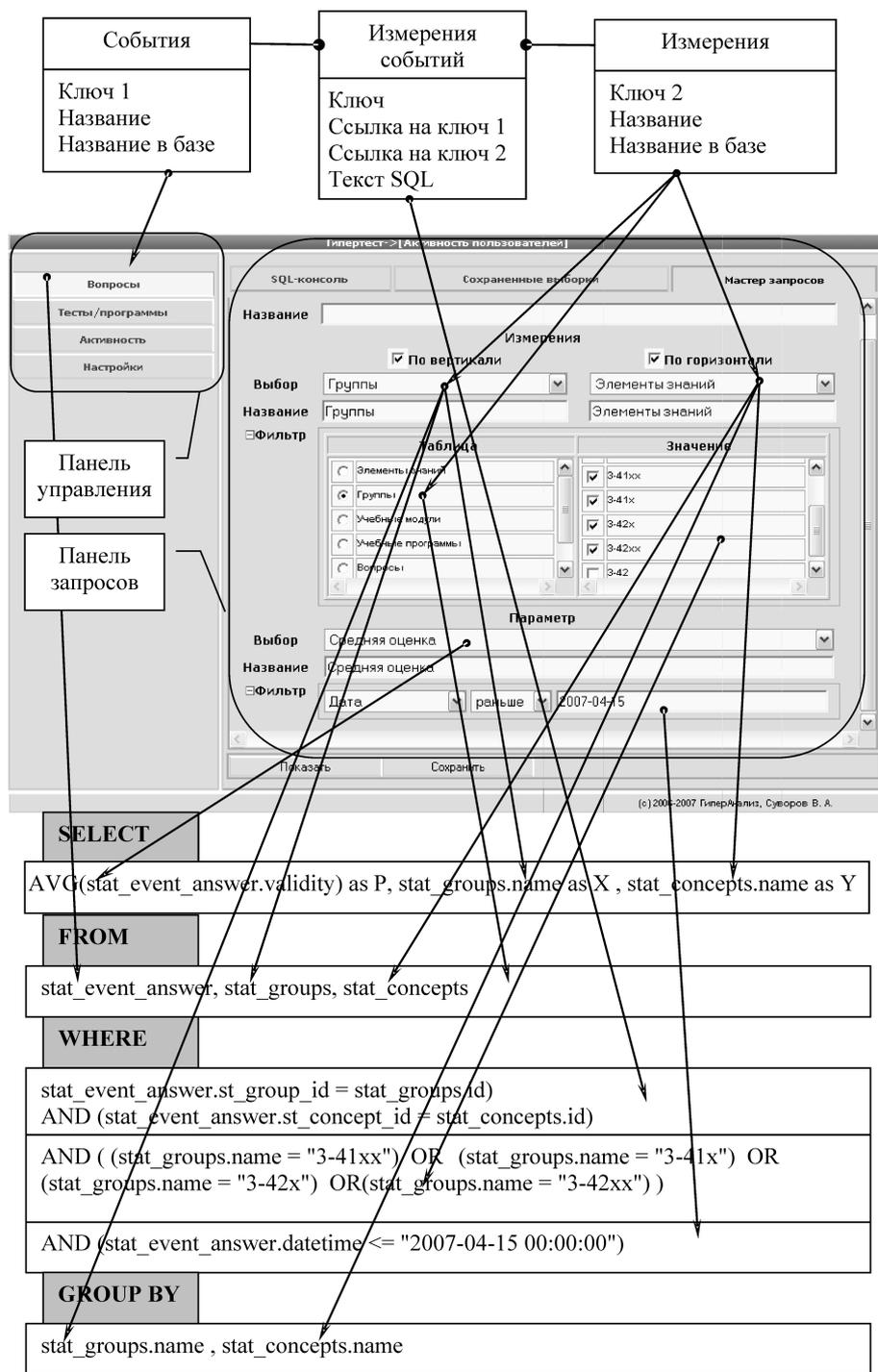


Рис. 3. Использование метаданных для построения запроса к серверу аналитики

б) фильтры по измерениям (поле "Название в базе" таблицы "Измерения" и выбранные значения);

с) фильтр по атрибуту (поле "Название в базе" таблицы "События" и введенное значение).

#### 4. GROUP BY

а) названия полей измерений (поле "Название в базе" таблицы "Измерения").

Интерфейс построителя запросов предоставляет пользователю возможности "разрезания" гиперкуба поперек одного или двух измерений и агрегирования атри-

бутов, которые характеризуют попавшие в сечения события. Например, для того чтобы построить сечные гиперкуба поперек измерений "группы" и "элементы знаний"<sup>2</sup>, надо установить флаги "по горизонтали" и "по вертикали" (если сечение разрезает только одну ось, устанавливается любой из этих флагов) и выбрать названия этих измерений из соответствующих списков. Чтобы сделать предметом анализа среднюю оценку, надо выбрать соответствующий способ агрегирования (другие варианты агрегирования — максимальная или минимальная оценка). Если требуется выполнить усреднение оценки по конкретным группам, надо настроить соответствующий фильтр установкой флага "Группы" и установкой флагов, соответствующих именам групп, задать фильтр по группам. Дополнительно может быть определен фильтр по времени наступления событий. Схема взаимодействия метаописания с шаблоном запроса в процессе такой параметризации, а также соответствующие настройки в интерфейсе построителя запросов показаны на рис. 3. Подготовленный пользователем запрос может быть исполнен (кнопка "показать") и/или сохранен (кнопка "сохранить"). Результаты исполнения запроса, возвращенные Web-клиенту в виде штабельной диаграммы (*stacked bar chart*), приведены на рис. 4, а (см. четвертую сторону обложки). Эти результаты были получены на основании статистики, собранной в процессе обучения студентов ИГЭУ специальности 230105 "Программное обеспечение вычислительной техники и автоматизированных систем" по дисциплине "Структуры и алгоритмы обработки данных" (раздел "Методы сортировки"). Их анализ, в частности, позволил установить, что самые низкие результаты соответствуют элементу профиля "сортировка естественным

слиянием". Так как оценка знаний формируется по результатам ответов на контрольные вопросы, далее был выявлен источник обнаруженного дефекта, т. е. вопрос, вызвавший наибольшие трудности. Анализ среднего балла для каждого из вопросов, оценивающих элемент

<sup>2</sup> В терминологии ГИПЕРТЕСТ элемент знаний — это значимая и независимо оцениваемая процедурами контроля единица учебного материала.

профиля "сортировка естественным слиянием", показал, что источником дефекта является вопрос № 36 (рис. 4, б, см. четвертую сторону обложки).

SQL-консоль позволяет формулировать любые запросы на выборку данных из хранилища. Составление запроса на языке SQL требует от пользователя знания этого языка и структуры хранилища. Например, чтобы оценить эффективность управления навигацией, выраженную отношением процента активации рекомендованных ссылок к оценке результатов подготовки, необходим запрос-агрегат (рис. 5). Он считает отношения параметров по каждому элементу знаний на выборках, построенных с помощью двух вложенных запросов, которые вычисляют сами значения параметров. Первый вложенный запрос вычисляет число правильных ответов по каждому элементу знаний. В запросе участвует таблица фактов (stat\_event\_answer) и таблица измерений (stat\_concepts). Этот запрос возвращает таблицу вида "название элемента знаний"—"количество правильных ответов". Второй вложенный запрос осуществляет расчет числа переходов по ссылкам на каждый из элементов знаний. В запросе участвует таблица фактов (stat\_event\_link) и таблица измерений (stat\_targets). Этот запрос возвращает таблицу вида "название элемента знаний"—"количество переходов по ссылке". Внешний запрос вычисляет отношения параметров при соответствующих элементах знаний. Результат возвращается в виде таблицы "название элемента знаний (X)"—"значение отношения (P)".

Как запросы, созданные с помощью Мастера, так и запросы, написанные непосредственно на SQL, могут быть сохранены для повторного использования. Для сохранения запроса на закладках "SQL-консоль" и "Мастер запросов" имеются соответствующие кнопки. Сохраненные запросы доступны на одноименной закладке.

Рассмотренные примеры позволяют утверждать, что результатом реализации предложенного подхода является клиент-серверная архитектура средств оперативной аналитической обработки действий студента в среде Web-обучения, отвечающая заявленным функциональным требованиям (наличие программного интерфейса заполнения и Web-интерфейса обработки и представления данных, пригодного для пользователя с отсутствием навыков работы с OLAP-приложениями). Далее будут рассмотрены технические решения, обеспечивающие открытость сервера аналитики и минимизацию трафика "клиент—сервер".

Под *открытой системой* принято понимать систему, которая обеспечивает возможность совместной работы с локальными и удаленными приложениями благодаря наличию открытых спецификаций на интерфейсы, сервисы и поддерживаемые форматы данных.

Совместная работа сервера аналитики с системами Web-обучения требует от последних поддержки про-

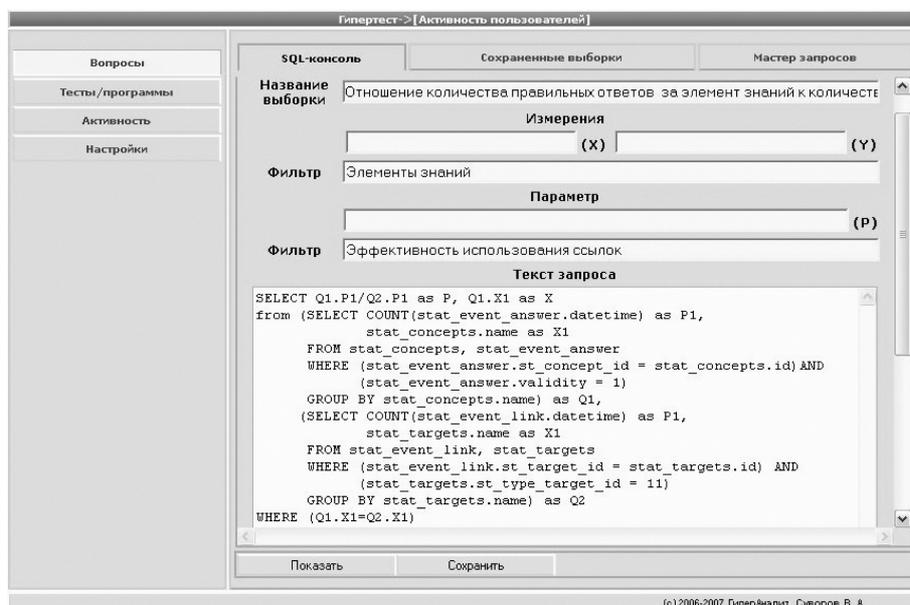


Рис. 5. Редактирование запроса в SQL-консоли

граммного интерфейса, построенного на основе библиотеки cURL (сокращение от "Client URLs", [14]). Библиотека cURL предназначена для организации такого взаимодействия серверов, при котором каждый из них может выступать в роли HTTP-клиента. Такое решение позволяет использовать сервис аналитики любому серверу Web-обучения, который поддерживает данный интерфейс. В процессе сбора статистики сервер Web-обучения с помощью набора функций cURL передает серверу аналитики однопоточный поток данных, характеризующих действия студента. Сценарий передачи данных включает в себя следующие действия:

- инициализация сеанса связи с сервером аналитики;
- настройка опций сеанса (адрес сервера, метод передачи, поля передаваемых данных);
- передача информации о событии удаленному серверу аналитики;
- завершение сеанса взаимодействия.

Минимизация трафика "клиент—сервер" в системе Web-OLAP обеспечивается за счет применения технологии Ajax [15] и серверного компонента деловой графики ChartDirector (<http://www.advsofteng.com/>).

Принципиальное отличие технологии Ajax (Asynchronous JavaScript and XML) от традиционного обмена Html-кодом между клиентом и Web-сервером заключается в перераспределении функций обработки информации в пользу клиента [15]. Это достигается за счет размещения на странице Java-скриптов (Ajax-машины), которые включают в себя XMLHttpRequest-объект для взаимодействия с сервером в формате XML и набор функций-обработчиков событий этого объекта (инициализация загрузки данных, соединение с сервером, начало загрузки, завершение загрузки и другие). Посредничество Ajax-машины позволяет существенно уменьшить объем трафика за счет обработки части событий "на месте", а также отправки клиенту только тех данных, которые нужны для обновления страницы. Применение технологии Ajax для создания Web-интерфейса аналитики позволило загружать с сервера только

те данные, с которыми в данный момент работает пользователь. Например, информация о сохраненных запросах загружается только после активизации соответствующей закладки.

Компонент ChartDirector предназначен для представления результатов выполнения запросов к хранилищу на странице клиента в формате деловой графики (круговые и столбчатые диаграммы, тренды, радарные схемы и т. п.). Обработка результатов аналитического запроса на стороне сервера также способствует уменьшению трафика "сервер—клиент".

Таким образом, в результате выполненных исследований разработаны клиент-серверная архитектура и базовые компоненты хранилища данных о действиях студента в среде Web-обучения. Хранилище имеет программный интерфейс, который позволяет системам Web-обучения, поддерживающим одноранговый обмен данными в стандарте сURL, заносить в хранилище данные о действиях студента. Хранилище также предлагает Web-интерфейс для построения и оперативного выполнения запросов к хранилищу данных (в том числе — нерегламентированных). Результаты выполнения запросов могут быть представлены в табличном или в графическом формате. Для представления деловой графики в интерфейсе Web-клиента использован серверный компонент ChartDirector. Хранилище аналитических данных апробировано на примере организации его взаимодействия с системой Web-обучения ГИПЕРТЕСТ 2.0. Полученные данные были использованы для анализа результатов контроля знания студентов специальности "Программное обеспечение вычислительной техники и автоматизированных систем" и выявления причин дефектов подготовки.

### Список литературы

1. **Ben-Ari M.** Constructivism in computer science education // Journal of Computers in Mathematics and Science Teaching 20 (1). 2001. P. 45—73.

2. **Henze N., Nejd W.** Constructivism in computer science education: evaluating a teleteaching environment for project oriented learning. (<http://citeseer.istpsu.edu/259179.html>)

3. **VanLehn K.** Intelligent tutoring systems for continuous, embedded assessment. ETS Invitational Conference "The Future of Assessment: Shaping Teaching and Learning". New York, October 10—11, 2005. 39 p.

4. **Murray R. C., VanLehn K.** Dynamic Approach for Optimal Selection of Tutorial Actions // In G. Gauthier, C. Frasson, & K. VanLehn (Ed.), Intelligent Tutoring Systems, 5th International Conference, ITS. New York: Springer. 12000. P. 153—162.

5. **Conati C., VanLehn K.** Providing Adaptive Support to the Understanding of Instructional Material. IUI'01, January 14—17, Santa Fe, New Mexico. 2001.

6. **Baker R. S. J. d., Corbett A. T., Koedinger K. R., Roll I.** Generalizing Detection of Gaming the System Across a Tutoring Curriculum // In Proceedings of the 8th International Conference on Intelligent Tutoring Systems. 2006. P. 402—411.

7. **Mining Student Learning Data to Develop High Level Pedagogic Strategy in a Medical ITS / M. V. Yudelso, O. Medvedeva, E. Legowski, M. Castine, D. Jukic, R. S. Crowley // In: Proceedings of Workshop on Educational Data Mining at The Twenty-First National Conference on Artificial Intelligence (AAAI 2006), Boston, MA. 2006.**

8. **Пантелеев Е. П.** Средства поддержки жизненного цикла Web-обучения в инструментальном комплексе ГИПЕРТЕСТ 2.0 // Информационные технологии. 2007. № 2. С. 39—45.

9. **Codd E. N., Codd S. B., Salley C. T.** Providing OLAP (On-Line Analytical Processing) to User-Analysts: An IT Mandate — E. F. Cold & Associates. 1993.

10. **Cobsa A.** Generic User Modeling Systems // User Modeling and User-Adapted Interaction. 2001. N 11. P. 49—63.

11. **Brusilovsky P., Sosnovsky S. A., & Shcherbinina O.** User Modeling in a Distributed E-Learning Architecture // Paper presented at the 10th International Conference on User Modeling (UM 2005), Edinburgh, Scotland, UK, July 24—29. 2005.

12. **OLAP-средства и Web-технологии.** По материалам зарубежных сайтов. ([http://www.citforum.ru/cfin/articles/olap\\_web/](http://www.citforum.ru/cfin/articles/olap_web/))

13. **Обзор OLAP-продуктов для Web.** По материалам зарубежных сайтов. ([http://www.citforum.ru/cfin/articles/olap\\_web\\_market/](http://www.citforum.ru/cfin/articles/olap_web_market/))

14. **Curl website** (<http://curl.haxx.se/>)

15. **Кейн Х. AJAX:** что скрывается за этими буквами? (<http://www.seonews.ru/article/.publication/161/>)

УДК 004.942:37

**Е. Ю. Карданова**, канд. физ.-мат. наук, доц.

**В. Б. Карпинский**,

Новгородский государственный университет  
имени Ярослава Мудрого, г. Великий Новгород

## Специальные методы анализа результатов тестирования, основанные на свойстве объективности моделей Раша

*Предлагаются специальные методы обработки результатов тестирования, основанные на свойстве объективности математических моделей Г. Раша и дающие дополнительные возможности исследования качества тестовых заданий и измерения испытуемых.*

### Введение

В современном образовательном процессе активно используется тестирование, что в ряде случаев вызывает нарекания педагогов и общественности. И действительно, тестирование имеет свои недостатки. Однако большая часть этих недостатков определяется не природой тестирования как такового, а методами извлечения из результатов тестирования полезной информации. Поэтому актуальны исследования, целью которых является разработка технологии обработки тестовой информации.

Важнейшим достижением современной тестологии является возможность подойти к оцениванию знаний как к процессу объективного измерения, что подразумевает получение инвариантных

относительно друг друга оценок мер испытуемых и параметров заданий теста, расположенных на единой метрической шкале и сопровождаемых характеристиками точности оценивания. Сделать это возможно только в рамках определенных математических моделей, а именно моделей семейства Г. Раша [1, 2]. Модели этого семейства позволяют преодолеть недостатки классической теории тестирования (КТТ), в рамках которой первичный балл участника тестирования по данному конкретному тесту принимается за оценку его уровня подготовленности. Вместе с тем, модели семейства Раша обладают целым рядом преимуществ по сравнению с многочисленными другими математическими моделями современной теории тестирования — теории моделирования и параметризации тестов (ТМПТ) [3].

Главным преимуществом моделей Раша является объективность измерений, что подразумевает следующее:

- оценки уровней подготовленности участников тестирования инвариантны относительно тестовых заданий, по результатам выполнения которых эти оценки получены;
- оценки трудностей тестовых заданий инвариантны относительно контингента испытуемых, по результатам тестирования которых эти оценки получены.

Другим важным преимуществом моделей Раша является возможность получения оценок уровней подготовленности испытуемых и трудностей заданий на единой метрической шкале. Это позволяет, в частности, сопоставлять результаты по различным вариантам теста (и даже полученные в разные годы) [4].

В настоящей статье приводится обоснование объективности измерений в рамках моделей Раша, а также указываются некоторые дополнительные возможности, обеспечиваемые наличием этого свойства. В частности, к этим возможностям относятся обнаружение искажений в измерении, исследование (возможно, различного) функционирования заданий по отношению к различным группам испытуемых и, наоборот, поведения испытуемых при работе над разными группами заданий. В статье представлены и продемонстрированы на примерах методы обработки информации, реализующие указанные дополнительные возможности.

### Обоснование объективности измерений в рамках моделей Раша

Пусть  $N$  испытуемых работают над тестом, содержащим  $I$  заданий. За выполнение каждого задания испытуемый может получить от 0 до  $m$  баллов в зависимости от числа правильно выполненных

шагов: за правильное выполнение каждого шага он получает 1 балл; общий балл за выполнение всего задания равен сумме правильно выполненных шагов. Если  $m = 1$ , задание называется дихотомическим; если  $m > 1$  — политомическим.

В результате тестирования мы получаем матрицу индикаторов ответов  $\mathbf{A} = (a_{ni})$  размерности  $N \times I$ :  $n$ -я строка этой матрицы ( $n = 1, 2, \dots, N$ ) содержит баллы испытуемого  $n$  по всем заданиям теста;  $i$ -й столбец матрицы ( $i = 1, 2, \dots, I$ ) содержит баллы всех испытуемых по  $i$ -му заданию теста.

Сумма элементов матрицы  $\mathbf{A}$ , стоящих в  $n$ -й строке, называется первичным баллом испытуемого  $n$ :

$$b_n = \sum_{i=1}^I a_{ni}, \quad n = 1, 2, \dots, N.$$

Сумма элементов матрицы  $\mathbf{A}$ , стоящих в  $i$ -м столбце, называется первичным баллом  $i$ -го задания:

$$c_i = \sum_{n=1}^N a_{ni}, \quad i = 1, 2, \dots, I.$$

Рассмотрим модель Раша с произвольными промежуточными категориями выполнения заданий [2]. Согласно этой модели, вероятность  $P_{nik}$  того, что испытуемый  $n$  с уровнем подготовленности  $\theta_n$  получит  $k$  баллов за выполнение  $i$ -го задания (т. е. выполнит  $k$  шагов в этом задании),  $k = 0, 1, \dots, m_i$ , определяется формулой

$$P_{nik} = \frac{\exp\left(k\theta_n - \sum_{j=0}^k \delta_{ij}\right)}{\sum_{l=0}^{m_i} \exp\left(l\theta_n - \sum_{j=0}^l \delta_{ij}\right)}, \quad (1)$$

где  $\delta_{ij}$  — трудность выполнения  $j$ -го шага задания  $i$ ;  $\delta_{i0} = 0$ ;  $m_i$  — число категорий оценивания в задании  $i$ . Модель (1) будем называть политомической моделью Раша. Величины  $\theta_n$ ,  $n = 1, \dots, N$ , и  $\delta_{ij}$ ,  $i = 1, \dots, I$ ;  $j = 1, \dots, m_i$ , являются параметрами модели и подлежат оцениванию.

В частном случае, если задание  $i$  — дихотомическое,  $m_i = 1$ ,  $k = 1$ , и модель (1) принимает вид

$$P_{ni} = \frac{\exp(\theta_n - \delta_i)}{1 + \exp(\theta_n - \delta_i)}. \quad (2)$$

Здесь  $P_{ni}$  — вероятность правильного выполнения испытуемым  $n$  задания  $i$ ;  $\delta_i$  — трудность выполнения этого задания. Модель (2) будем кратко называть дихотомической моделью Раша. Для простоты все рассуждения проведем для дихото-

мической модели (для случая политомической модели см., например, [5]).

Покажем, что в рамках дихотомической модели Раша справедливы следующие утверждения [3, 6]:

- первичный балл испытуемого  $b_n$  является достаточной статистикой относительно параметра  $\theta_n$  — уровня подготовленности испытуемого  $n$ ; первичный балл задания  $c_i$  является достаточной статистикой относительно параметра  $\delta_i$  — уровня трудности задания  $i$ ;
- уровни подготовленности испытуемых  $\theta_n$ ,  $n = 1, \dots, N$ , могут быть оценены независимо от трудностей заданий конкретного теста; уровни трудности заданий теста  $\delta_i$ ,  $i = 1, \dots, I$ , могут быть оценены независимо от того, какой контингент испытуемых выполнял эти задания.

Обозначим  $P(a_{ni}/\theta_n, \delta_i)$  — вероятность ответа  $a_{ni}$  испытуемого  $n$  с уровнем подготовленности  $\theta_n$  на  $i$ -е задание трудности  $\delta_i$ . Случайная величина  $a_{ni}$  может принимать значения 1 или 0 в зависимости от того, правильно или нет выполнено задание. Тогда с учетом (2) имеем:

$$P(a_{ni}/\theta_n, \delta_i) = \frac{\exp[a_{ni}(\theta_n - \delta_i)]}{1 + \exp(\theta_n - \delta_i)}.$$

Далее обозначим через  $\mathbf{a}_n = (a_{n1}, a_{n2}, \dots, a_{nI})$  строку ответов испытуемого  $n$  на задания теста (т. е.  $n$ -ю строку матрицы ответов  $\mathbf{A}$ ). Тогда в предположении независимости тестовых заданий вероятность этого события определяется как

$$\begin{aligned} P(\mathbf{a}_n/\theta_n, (\delta_i)) &= \prod_{i=1}^I \frac{\exp[a_{ni}(\theta_n - \delta_i)]}{1 + \exp(\theta_n - \delta_i)} = \\ &= \frac{1}{\psi_n} \exp \sum_{i=1}^I a_{ni}(\theta_n - \delta_i), \end{aligned} \quad (3)$$

где  $\psi_n = \prod_{i=1}^I (1 + \exp(\theta_n - \delta_i))$ ;  $\delta_i$  обозначает множество трудностей всех заданий. Преобразуем второй сомножитель формулы (3):

$$\begin{aligned} \exp \sum_{i=1}^I a_{ni}(\theta_n - \delta_i) &= \exp \left( \sum_{i=1}^I a_{ni}\theta_n - \sum_{i=1}^I a_{ni}\delta_i \right) = \\ &= \exp \left( \theta_n b_n - \sum_{i=1}^I a_{ni}\delta_i \right) = \\ &= \exp(\theta_n b_n) \exp \left( - \sum_{i=1}^I a_{ni}\delta_i \right). \end{aligned}$$

Таким образом,

$$P(\mathbf{a}_n/\theta_n, (\delta_i)) = \frac{\exp(\theta_n b_n)}{\psi_n} \exp \left( - \sum_{i=1}^I a_{ni}\delta_i \right).$$

Далее можно вычислить вероятность всей матрицы ответов  $\mathbf{A}$ . В силу независимости ответов испытуемых имеем:

$$\begin{aligned} P(\mathbf{A}/(\theta_n, \delta_i)) &= \prod_{n=1}^N P(\mathbf{a}_n/(\theta_n, \delta_i)) = \\ &= \frac{1}{\Psi} \prod_{n=1}^N \left[ \exp(\theta_n b_n) \exp \left( - \sum_{i=1}^I a_{ni}\delta_i \right) \right] = \\ &= \frac{1}{\Psi} \exp \left( \sum_{n=1}^N \theta_n b_n - \sum_{i=1}^I c_i \delta_i \right) = \\ &= \frac{1}{\Psi} \exp \left( \sum_{n=1}^N \theta_n b_n \right) \exp \left( - \sum_{i=1}^I c_i \delta_i \right), \end{aligned} \quad (4)$$

где  $\Psi = \prod_{n=1}^N \psi_n = \prod_{n=1}^N \prod_{i=1}^I (1 + \exp(\theta_n - \delta_i))$ ;  $\theta_n$  обозначает множество уровней подготовленности всех испытуемых.

В правую часть уравнения (4) входят только первичные баллы заданий и испытуемых, а не сами элементы матрицы  $\mathbf{A}$ . Следовательно, вся информация об уровне подготовленности  $\theta_n$  испытуемого  $n$  сосредоточена только в одном числе — первичном балле  $b_n$  этого испытуемого. Аналогично вся информация о трудности  $\delta_i$  задания  $i$  сосредоточена в одном числе — первичном балле  $c_i$  этого задания. Это означает, что первичные баллы  $b_n$  и  $c_i$  являются достаточными статистиками относительно соответствующих параметров  $\theta_n$  и  $\delta_i$ . Отметим, что указанная редукция матрицы ответов  $\mathbf{A}$  к маргинальным суммам  $b_n$  и  $c_i$  по строкам и столбцам матрицы возможна лишь в рамках определенных математических моделей, в частности моделей Раша.

Далее, разделение  $\theta_n b_n$  и  $c_i \delta_i$  в равенстве (4) позволяет оценить уровни подготовленности испытуемых вне зависимости от трудностей заданий теста, а трудности заданий — вне зависимости от уровней подготовленности испытуемых. Это свойство называется сепарабельностью оценок трудностей заданий и уровней подготовленности испытуемых. Именно сепарабельность позволяет обеспечить инвариантность этих оценок — необходимое условие объективности. Для оценивания параметров  $\theta_n$  и  $\delta_i$  модели Раша могут быть использованы различные методы — метод наибольшего правдоподобия, метод моментов и др. [1, 6]. Полученные оценки будут расположены на единой метрической шкале и сопровождаются характеристиками точности оценивания.

### Методы обнаружения искажений в измерении испытуемых

Главная цель педагогического тестирования — измерение уровня подготовленности учащихся в

соответствующей области знаний. Под искажением в процессе измерения будем понимать любое отклонение от достижения этой цели. Источниками искажений могут быть любые факторы, которые влияют на процесс измерений, в частности, различные нарушения процедуры тестирования — списывание, подсказки и т. д. Искажения могут иметь место как на индивидуальном, так и на групповом уровне. Далее в работе рассматриваются методы обнаружения искажений, основанные на свойстве объективности моделей Раша. Возможность применения таких методов обнаружения искажений является новой, недостижимой средствами КТТ.

**Исследование поведения испытуемых по отношению к различным группам заданий.** Один из предлагаемых методов обнаружения искажений основан на исследовании поведения испытуемых по отношению к различным группам заданий. Если результат исследования для конкретного испытуемого существенно отличается от общей нормы, он может свидетельствовать о наличии искажений, и результаты тестирования этого испытуемого должны быть особо проанализированы или пересмотрены.

Предположим, что испытуемый самостоятельно выполняет относительно легкие задания, а затем списывает трудные для него задания с задания хорошо подготовленного учащегося (или ему сообщаются ответы на более трудные задания). Это приводит к завышению оценки уровня подготовленности испытуемого по сравнению с его истинным значением.

Предлагается следующая технология обработки информации о результатах тестирования для обнаружения случаев списывания/подлога:

- калибровка всего теста и получение оценок мер испытуемых и параметров заданий;
- разбиение теста на отдельные группы заданий (субтесты) и их калибровка;
- выравнивание показателей по отдельным субтестам;
- сравнение мер испытуемых, полученных по результатам выполнения отдельных субтестов, в целях исследования их устойчивости относительно различных групп заданий.

Как показано выше, в условиях моделей Раша оценки уровней подготовленности испытуемых инвариантны относительно наборов заданий, по результатам выполнения которых они получены. Рассматриваемый вид искажений может быть выделен, если уровень подготовленности испытуемого, полученный по результатам выполнения более трудных заданий, статистически значимо выше уровня подготовленности этого испытуемого, полученного по результатам выполнения более легких заданий. Применительно к ЕГЭ в ка-

честве первого субтеста можно рассмотреть относительно легкую часть А теста, в качестве второго субтеста — части В и С.

Рассмотрим эту технологию на примере теста по географии, имеющего структуру, аналогичную КИМ ЕГЭ. Тест содержит 50 заданий, которые разделены на три части. Часть А содержит 32 задания (А1—А32) с выбором одного ответа из четырех предложенных. Часть В содержит 12 заданий (В1—В12) с кратким ответом в виде числа или последовательности букв. Часть С включает 6 заданий (С1—С6), в которых требуется записать полный ответ на поставленный вопрос. Все задания части А оценивались дихотомически, часть заданий частей В и С — политомически. Общее число испытуемых, выполнявших данный тест, равно 655.

Разобьем тест на два субтеста. В первый субтест включим задания А1—А32 части А. Во второй субтест включим задания А27—А32 части А и задания частей В и С. Максимальный первичный балл за выполнение каждого субтеста равен 32. Далее каждый субтест отдельно откалибруем. Окончательно получим оценки трудностей заданий теста и уровней подготовленности испытуемых, но результаты каждой калибровки будут находиться на своей шкале. Для возможности сравнения уровней подготовленности испытуемых, полученных по результатам выполнения различных субтестов, необходимо провести специальную процедуру, позволяющую установить связь между уровнями подготовленности из различных калибровок и поместить их на одну общую шкалу. Такая процедура называется выравниванием. Для проведения процедуры выравнивания в каждый субтест были включены общие задания А27—А32. В результате отдельных калибровок были получены две оценки трудностей для каждого из этих заданий. Метричность шкал позволяет, оценив разность между значениями трудностей общих заданий, перевести все значения параметров испытуемых и заданий со шкалы одной калибровки на шкалу другой [4].

Таким образом, после выполнения процедуры выравнивания все параметры отдельных калибровок будут находиться на одной шкале, например, шкале первого субтеста. Это позволит непосредственно сравнить оценки уровней подготовленности испытуемых, полученные по результатам выполнения различных субтестов в целях исследования их устойчивости относительно различных групп заданий. Для этого для каждого испытуемого вычислим статистику:

$$t_n = \frac{\theta_n^{(2)} - \theta_n^{(1)}}{\sqrt{(\sigma_n^{(1)})^2 + (\sigma_n^{(2)})^2}},$$

где  $\theta_n^{(1)}$ ,  $\theta_n^{(2)}$  — оценки уровня подготовленности испытуемого  $n$  по результатам выполнения субтеста 1 и субтеста 2 соответственно;  $\sigma_n^{(1)}$ ,  $\sigma_n^{(2)}$  — средние квадратические ошибки измерения  $\theta_n^{(1)}$  и  $\theta_n^{(2)}$ . Статистика  $t_n$  имеет асимптотически нормальное распределение с нулевым математическим ожиданием и единичной дисперсией [1]. Значения этой статистики, по модулю большие 2, указывают на слишком большие отклонения оценок мер испытуемых. Это, в свою очередь, означает, что оценки уровня подготовленности испытуемых зависят от того, на основе каких заданий они получены.

В нашем случае абсолютное большинство испытуемых демонстрируют инвариантность мер относительно различных групп заданий: только 25 испытуемых (менее 4 % от общего числа) имеют статистически значимо различные оценки уровня подготовленности. При этом у 20 испытуемых значения статистики  $t_n > 2$  и только у пяти испытуемых  $t_n < -2$ . С точки зрения выявления случаев списывания/подлога нас интересуют испытуемые, для которых  $t_n > 2$ . Это означает, что оценка уровня подготовленности по результатам выполнения второго субтеста значимо выше оценки уровня подготовленности по результатам первого субтеста.

В табл. 1 представлены данные по двум испытуемым, имеющим самое большое положительное и самое маленькое отрицательное значения статистики  $t_n$ . (Все данные представлены на шкале логитов [1]. Стандартным линейным преобразованием шкала логитов может быть преобразована в любую удобную шкалу — например стобалльную.) В табл. 2 представлены профили ответов испытуемых из табл. 1.

Рассмотрим испытуемого № 38. Его тестовый балл довольно высок:  $\theta = 0,82$ . Проанализируем

профиль его ответов. Этот испытуемый выполнил немногим более половины заданий части А, но при этом не выполнил ряд очень легких заданий. За выполнение заданий части С он получил максимально возможное число баллов. Возможно, этот испытуемый много сил отдал на выполнение заданий части С, и ему не хватило времени на задания частей А и В. Но, возможно, ему была оказана помощь при выполнении заданий части С, и в этом случае его истинный тестовый балл ниже того, что он получил. В любом случае, его знания имеют существенные пробелы, раз он не выполнил несколько заданий части А, очень легких для остальных участников тестирования. То, что у этого испытуемого уровень подготовленности по результатам выполнения трудного субтеста 2 существенно превышает уровень подготовленности по результатам легкого субтеста 1, свидетельствует о вероятных искажениях в его оценивании: у нас есть все основания считать, что тестовый балл этого испытуемого завышен по сравнению с его истинным значением.

Рассмотрим испытуемого № 307, имеющего тестовый балл  $\theta = 0,29$ . Он выполнил правильно почти все задания части А (28 из 32) и получил только 2 балла за выполнение заданий части В и 1 балл — за выполнение заданий части С. Задания части А, которые данный испытуемый не выполнил, имеют более высокий уровень трудности, чем остальные задания этой части. Задания частей В и С — намного более трудные, чем задания части А. Таким образом, профиль ответов этого испытуемого очень правильный, близок к каноническому: легкие задания выполнены, трудные — нет. Уровень подготовленности этого испытуемого по результатам выполнения легкого субтеста 1 существенно превышает уровень подготовленности по результатам трудного субтеста 2, что также

Таблица 1

Номер испытуемого	1-й субтест (задания 1—32)		1-й субтест (задания 27—50)		Статистика $t_n$
	Оценка уровня подготовленности	Ошибка измерения	Оценка уровня подготовленности	Ошибка измерения	
38	0,57	0,38	2,43	0,42	3,28
307	2,12	0,55	0,44	0,41	-2,45

Таблица 2

Номер испытуемого	Балл	Ряд ответов		
		Часть А	Часть В	Часть С
38	0,82	10111 01010 10111 01111 01000 00111 11	010 00002 1100	122223
307	0,29	11111 11010 11111 11111 11011 01111 11	010 00000 0100	100000

может свидетельствовать об искажениях в его оценивании. Наиболее вероятная причина — чрезмерная осторожность этого испытуемого, но возможны и другие причины. В результате тестовый балл данного испытуемого, по всей вероятности, занижен по сравнению с его истинным значением.

Отметим, что наличие проблем в оценивании испытуемых может быть диагностировано и статистическими методами с помощью различных статистик согласия [6, 7]. Однако это не входит в рамки данной статьи.

**Исследование функционирования заданий по отношению к различным группам испытуемых.** Искажения при измерении испытуемых могут порождаться также различным функционированием заданий теста по отношению к разным группам испытуемых.

Предлагается следующая технология обработки информации о результатах тестирования для исследования функционирования заданий по отношению к различным группам испытуемых.

- Калибровка всего теста и получение оценок мер испытуемых и параметров заданий.
- Разбиение всей выборки испытуемых на две части и отдельная их калибровка.
- Выравнивание показателей по отдельным калибровкам.
- Сравнение трудностей заданий, полученных по результатам отдельных калибровок в целях исследования их устойчивости относительно различных групп испытуемых.

В зависимости от того, как происходит разбиение исходной выборки испытуемых, данная процедура может применяться для решения различных задач.

В условиях моделей Раша оценки параметров заданий должны быть инвариантны относительно групп испытуемых, по результатам тестирования которых эти оценки получены. Невыполнение этого свойства может быть следствием ошибок в разработке заданий или нарушений процедуры тестирования на групповом уровне.

Рассмотрим эту технологию на примере того же теста по географии, который анализировался в предыдущем разделе. Разобьем всех испытуемых, выполнявших данный вариант теста, на две группы — сильную и слабую. В сильную группу отнесем испытуемых, первичный балл которых не менее 29 (29 — медиана распределения первичных баллов), а в слабую — испытуемых, первичный балл которых не более 29. В обе выборки включенным 21 испытуемый с первичным баллом, равным 29, для проведения процедуры выравнивания [4]. Далее каждую выборку отдельно откалибруем. После выполнения процедуры выравнивания все параметры отдельных калибровок будут находить-

ся на одной шкале, что позволит непосредственно сравнить оценки трудностей заданий, полученные по результатам выполнения теста различными группами испытуемых в целях исследования их устойчивости относительно различных групп испытуемых. Для этого для каждого задания вычислим статистику:

$$t_i = \frac{\delta_i^{(1)} - \delta_i^{(2)}}{\sqrt{(\sigma_i^{(1)})^2 + (\sigma_i^{(2)})^2}},$$

где  $\delta_i^{(1)}$  — оценка трудности  $i$ -го задания теста, полученная по результатам выполнения теста сильной группой испытуемых;  $\delta_i^{(2)}$  — оценка трудности  $i$ -го задания теста, полученная по результатам выполнения теста слабой группой испытуемых;  $\sigma_i^{(1)}$ ,  $\sigma_i^{(2)}$  — средние квадратические ошибки измерения величин  $\delta_i^{(1)}$  и  $\delta_i^{(2)}$ .

Статистика  $t_i$  имеет асимптотически нормальное распределение с нулевым математическим ожиданием и единичной дисперсией [1]. Значения этой статистики, по модулю большие 2, указывают на слишком большие отклонения оценок трудностей заданий. Это, в свою очередь, указывает на то, что оценки трудностей заданий зависят от того, на основе тестирования каких испытуемых они получены.

Если какое-нибудь задание имеет статистически различные уровни трудности, то это свидетельствует о различном функционировании этого задания по отношению к различным группам испытуемых. Такие задания должны быть исследованы для выяснения причин. Для анализируемого теста корреляция между оценками трудностей заданий, полученными по результатам выполнения теста сильной и слабой группами испытуемых, равна 0,93, что говорит о том, что большинство заданий демонстрируют инвариантность мер относительно различных выборок испытуемых. В табл. 3 представлены данные по трем заданиям теста — со значением статистики  $t_i$  внутри и снаружи критической области.

Рассмотрим задание № 5, имеющее значение статистики  $t_i = -0,40$ . Это задание демонстрирует устойчивость оценок трудности относительно различных выборок испытуемых. На рис. 1 представлены характеристические кривые задания № 5, соответствующие каждой из групп испытуемых. Ось абсцисс представляет собой ось измеряемой переменной, по оси ординат откладывается математическое ожидание балла за выполнение данного задания. В дихотомическом случае  $M(a_{ni}) = P_{ni}$ , т. е. математическое ожидание балла совпадает с вероятностью правильного ответа (2).

Номер задания	Сильная группа			Слабая группа			Статистика $t_i$
	Трудность задания	Ошибка измерения	Ранг	Трудность задания	Ошибка измерения	Ранг	
5	-0,79	0,16	14	-0,71	0,12	16	-0,40
14	-1,18	0,19	11	-0,37	0,12	22	-3,60
26	0,28	0,12	28	-0,86	0,12	15	6,72

Таким образом, характеристическая кривая задания представляет собой график теоретической вероятности успеха.

На рис. 1 видно, что характеристические кривые задания № 5 для различных групп испытуемых практически совпадают. Далее, для каждой группы испытуемых вычислим среднее значение  $\bar{\theta}_i$ ,  $i = 1, 2$ , уровней подготовленности испытуемых данной группы и эмпирическую вероятность  $\hat{p}_i$ ,  $i = 1, 2$ , правильного ответа на рассматриваемое задание теста — относительную частоту правильного ответа. На рисунке отмечены точки с координатами  $(\bar{\theta}_i, \hat{p}_i)$ ,  $i = 1, 2$ . Две точки в верхней части рисунка соответствуют сильной выборке испытуемых, две точки в нижней части — слабой. Такое расположение точек означает, что частота правильного ответа на данное задание выше для испытуемых сильной группы, чем для испытуемых слабой группы. Но трудность задания инвариантна относительно этих двух групп испытуемых. Такое задание функционирует в полном соответствии с требованиями модели Раша.

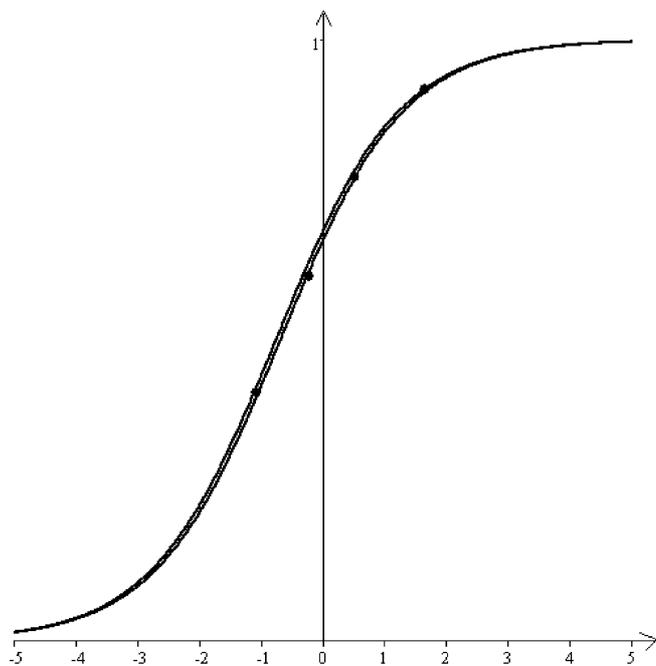


Рис. 1. Характеристические кривые задания № 5

Рассмотрим задание № 26, имеющее значение статистики  $t_i > 2$ . Это задание оказалось существенно более легким для испытуемых из слабой группы, чем из сильной: оно занимает 28-е место по уровню трудности для сильной выборки и 15-е место — для слабой группы (задания были ранжированы от самого легкого к самому трудному относительно каждой группы испытуемых). Анализ ответов испытуемых на это задание показал, что 43 испытуемых (7 % от общего числа) выполнили это задание неправильно, хотя вероятность того, что они выполнят его правильно, превышала 0,8. Таким образом, это задание, по всей вероятности, вводит в заблуждение сильных учащихся, которые в результате выполняют его неправильно. Следовательно, это задание не дифференцирует испытуемых, внося дезинформацию в их оценивание.

На рис. 2 представлены характеристические кривые задания № 26, соответствующие каждой из групп испытуемых. В отличие от рис. 1, характеристические кривые одного и того же задания, соответствующие различным выборкам испытуемых, теперь не совпадают: оценки уровня трудности статистически различны. Более того, это задание плохо согласуется с моделью измерения, о чем свидетельствует расположение эмпирических точек.

Для иллюстрации функционирования задания № 26 на рис. 3 представлена характеристическая кривая этого задания для полной выборки испытуемых. Структура рис. 3 та же, что и рис. 2, с той лишь разницей, что для большей наглядности эмпирических точек теперь три в соответствии с разбиением всей выборки испытуемых на три части — с высоким, средним и низким уровнем подготовленности. На рис. 3 видно, что дифференцирующая сила этого задания ниже, чем у других заданий теста. В результате эмпирическая характеристическая кривая задания будет отличаться от теоретической кривой: наклон эмпирической кривой в точке перегиба меньше, чем у других заданий. Тем самым для данного задания нарушается одно из основных требований моделей Раша: дифференцирующая сила всех заданий предполагается одинаковой [2, 6]. Именно поэтому данное

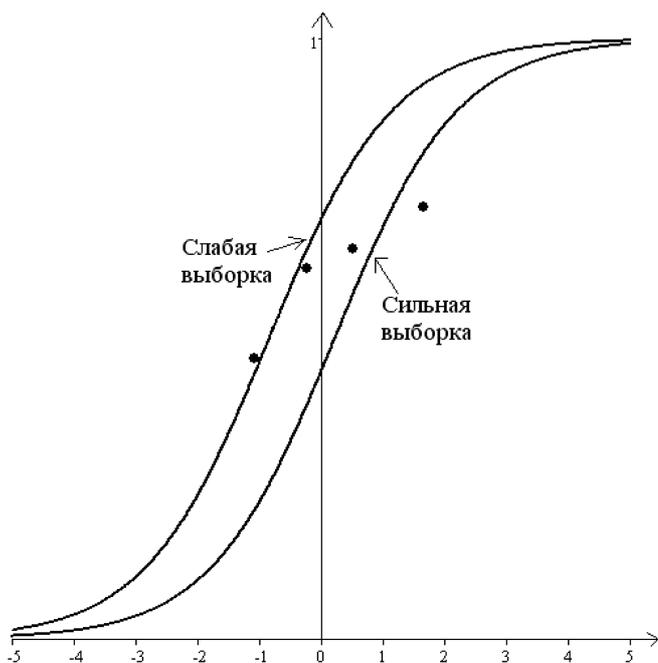


Рис. 2. Характеристические кривые задания № 26 (для различных выборок)

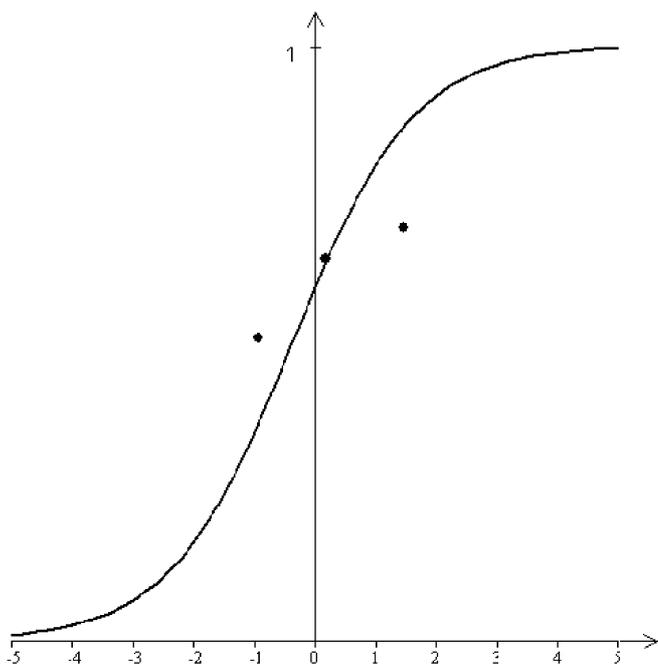


Рис. 3. Характеристическая кривая задания № 26 (для полной выборки)

задание демонстрируют плохое согласие с моделью, с которой хорошо согласуются большинство других заданий теста.

Рассмотрим задание № 14, имеющее значение статистики  $t_i < -2$ . Это задание оказалось существенно более трудным для испытуемых из слабой группы, чем из сильной: оно занимает 11-е место по уровню трудности для сильной выборки и 22-е место — для слабой группы. Следовательно, это

задание слишком хорошо дифференцирует испытуемых, его дифференцирующая сила выше, чем у других заданий. Эмпирическая характеристическая кривая этого задания также будет отличаться от теоретической кривой: наклон эмпирической кривой в точке перегиба будет больше, чем у других заданий. Следовательно, это задание также находится в плохом согласии с моделью измерения. Среди возможных причин наиболее вероятной является нарушение принципа локальной независимости [7]. Такие задания не вносят самостоятельного, независимого вклада в измерение уровня подготовленности испытуемых.

Таким образом, исследование устойчивости параметров заданий относительно групп испытуемых с различным уровнем подготовки позволяет выявить задания, находящиеся в плохом согласии с моделью измерения, и более того, проанализировать причины.

Отметим, что наличие проблем в функционировании указанных заданий может быть диагностировано и статистическими методами с помощью различных статистик согласия [1, 6, 7]. Однако это не входит в рамки данной статьи.

Иногда встречаются задания, которые не демонстрируют инвариантности оценок их трудности относительно групп испытуемых одинакового уровня подготовленности. Эти группы могут быть сформированы по различным основаниям. Например, в качестве такого основания может быть выбран пол участника тестирования, его этнические характеристики, место проживания, тип учебного заведения и т. д. Эти задания, как правило, находятся в хорошем согласии с моделью измерения и на стадии исследования согласия экспериментальных данных с моделью не выявляются.

Пример такого задания показан на рис. 4. Характеристические кривые одного и того же задания, соответствующие различным выборкам испытуемых, не совпадают: оценки уровня трудности статистически различны. При этом для каждой выборки испытуемых наблюдается хорошее согласие экспериментальных данных с моделью измерения, о чем свидетельствует близость точек экспериментальной характеристической кривой к теоретической.

В этом случае говорят о различном функционировании задания относительно различных групп испытуемых (*item bias* или *differential item functioning* в англоязычной литературе [7]). Причины различного функционирования могут быть двух типов. Во-первых, причины могут быть связаны с самим заданием. Например, задание может содержать слова, которые хорошо известны и понятны жителям одного региона и неизвестны (а потому и непонятны) жителям другого. В ре-

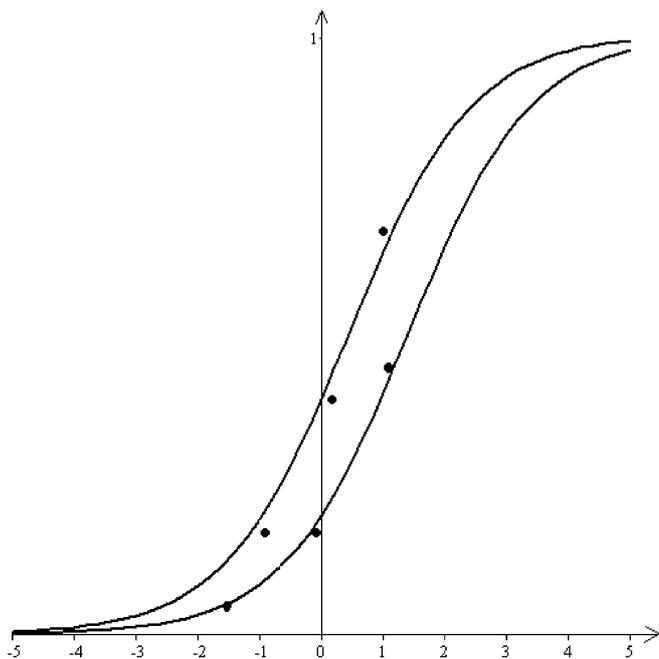


Рис. 4. Характеристические кривые задания, демонстрирующего смещение оценок

зультате такое задание окажется существенно труднее для учащихся второго региона, чем для учащихся первого. Если в тесте таких заданий несколько, то это повлияет на первичные, а следовательно, и тестовые баллы участников тестирования из второго региона: они окажутся существенно ниже. При этом причина более низких баллов кроется не в более низком уровне подготовленности по тестируемому предмету, а в непонимании самих заданий. (Естественно, все это справедливо только в том случае, если речь не идет о тесте, целью которого как раз и является проверка знания именно этих слов.)

Во-вторых, причины смещения оценок могут быть связаны с нарушениями процедуры тестирования. Например, предположим, что при выполнении части заданий теста учащимся некоторого региона  $X$  была оказана помощь. Тогда эти задания окажутся существенно легче для учащихся этого региона, чем для учащихся из других регионов. Применительно к ЕГЭ помощь может быть оказана по заданиям части  $C$ , которые требуют развернутого решения. В результате тестовые баллы учащихся из региона  $X$  окажутся несправедливо завышенными. Следовательно, будут иметь место искажения в измерении испытуемых, при-

чем на групповом уровне. Таким образом, наличие заданий, не демонстрирующих инвариантности оценок трудности относительно различных групп испытуемых одинакового уровня подготовленности, может служить свидетельством возможных нарушений процедуры тестирования.

### Заключение

Свойство объективности моделей Раша является их главным преимуществом по сравнению с другими математическими моделями современной теории тестирования ТМПТ. В статье рассмотрены некоторые дополнительные возможности, которые дает это свойство для исследования качества тестовых заданий и измерения испытуемых, и предложены специальные методы обработки результатов тестирования, реализующие эти возможности. Предложенные методы апробированы как при исследовании реальных результатов тестирования по разным предметам разными выборками испытуемых, так и на модельных данных, случаи искажений в которых заведомо известны. Для этого разработано соответствующее программное обеспечение.

Использование моделей Раша при анализе результатов тестирования позволяет применять различные статистические методы, использование которых также в ряде случаев позволяет реализовать соответствующие дополнительные возможности. Особенно эффективно совместное использование описанных выше специальных методов и методов математической статистики, но рассмотрение этого вопроса выходит за рамки настоящей статьи.

### Список литературы

1. Нейман Ю. М., Хлебников В. А. Введение в теорию моделирования и параметризации педагогических тестов. М.: Прометей, 2000. 169 с.
2. Карданова Е. Ю., Нейман Ю. М. Основные модели современной теории тестирования // Вопросы тестирования в образовании. 2003. № 7. С. 26.
3. Карданова Е. Ю. Преимущества современной теории тестирования по сравнению с классической теорией тестирования // Вопросы тестирования в образовании. 2004. № 10. С. 28.
4. Карданова Е. Ю., Нейман Ю. М. Проблема выравнивания в современной теории тестирования // Вопросы тестирования в образовании. 2003. № 8. С. 20.
5. Нейман Ю. М., Хлебников В. А. Как оценивается уровень подготовленности учащихся по результатам единого государственного экзамена. М.: Центр Тестирования МО РФ, 2003. 44 с.
6. Wright B. D., Stone M. N. Best Test Design. Rasch Measurement Chicago. Mesa Pres. 1979. 223 p.
7. Smith R. M. Person and Item Analysis. Mesa Press. Chicago, 1992.

**В. А. Углев,**

**В. А. Устинов,** канд. тех. наук, проф.,

**Б. С. Добронев,** д-р физ.-мат, наук, проф.,

Сибирский федеральный университет,  
г. Красноярск

## Системный подход к процессу обучающего компьютерного тестирования

*Вводится понятие обучающего компьютерного тестирования как инструмента реализации обратной связи электронных учебных курсов и обучаемого, которое осуществляется комплексным оцениванием и адаптацией. С учетом приоритетов обучения проведен системный анализ обучающих возможностей компьютерного тестирования, предложен подход к программной организации среды обучающего тестирования с использованием механизма экспертных систем.*

### Введение

Повышение интенсивности применения классических компьютерных тестов в составе учебных курсов ставит вопрос об эффективности тестового контроля как доминирующего метода определения уровня успеваемости учащихся. Существуют различные мнения об обоснованности данного метода контроля, так как эта предметная область сложна и не поддается исчерпывающей формализации. На наш взгляд, рассмотрение теста только как инструмента вычисления количественной оценки несколько сужает проблему. Это является одним из тех факторов, которые препятствуют эффективному применению компьютерных тестов для целей обучения и самообучения.

Рассмотрим компьютерное тестирование (КТ) как инструмент обратной связи электронных учебных курсов (ЭУК) и пользователя с целью выявить более эффективную форму автоматизированного обучения, которая порождает качественно новые требования к реализации процесса компьютерного тестирования и соответствующего программного обеспечения.

### Обучение как цель компьютерного тестирования

Исследуя предметную область, нами был обследован рынок программного обеспечения систем КТ<sup>1</sup> (более 40 пакетов). В результате было замечено, что эти системы целенаправленно игно-

рируют сам процесс обучения, т. е. их целью<sup>2</sup> является исключительно контроль. Назовем такой подход *классическим*, под которым будем понимать тестирование, характеризующееся следующими признаками:

- активное участие преподавателя на всех этапах процесса тестирования как контролирующей и организующей стороны;
- минимальная инициатива тестируемого;
- приуроченность процесса тестирования к мероприятиям коллективной оценки всего контингента обучаемых;
- изолированность тестовых заданий от обучающего материала, характеризующих финишный этап его изучения;
- направленность всего процесса исключительно на контроль знаний с последующим выставлением оценки;
- расчет итоговой оценки исключительно по числу верно решенных тестовых заданий.

Все это характерно для тестирования в рамках аудиторных занятий при учебном заведении. Здесь обособленность процесса контроля компенсируется всеобщим участием учителя (тьютора). Примерами этих систем будут являться программные пакеты ОРОКС, WebCT, LearningSpace, Sun-Rav TestOfficePro, Mirax Test и многие другие [1, 2]. Парадоксально то, что базовый (классический) подход к организации, проведению и оценке результатов КТ при переходе к дистанционному обучению фактически не претерпел никаких изменений.

Для выяснения особенностей процесса КТ представим его в виде сложного объекта (черного ящика, рис. 1). Традиционно можно выделить про-

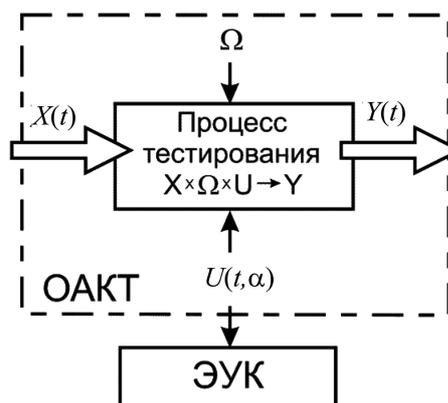


Рис. 1. Процесс тестирования в виде модели черного ящика

<sup>1</sup> В статье рассматриваются педагогические тесты на выявление уровня знаний. Психологическое тестирование имеет другие цели и методики, поэтому далее не рассматривается.

<sup>2</sup> Программы автоматизации КТ принято разделять по следующим целям: предварительный контроль; текущий контроль; тематический контроль; рубежный и итоговый контроль.

странство входных воздействий  $X(t)$ , влияние окружающей среды  $\Omega$  и механизм управления  $U(t)$  [3], согласованное действие которых в результате работы системы КТ позволяет оценить знания обучаемого.

Можно отметить наличие очевидных закономерностей, которые дают возможность получить приближенную оценку истинных знаний (число верно решенных вопросов, время прохождения теста, сложность тестовой выборки<sup>3</sup> и пр.). Оценка обычно выставляется на основании простого подсчета числа верно решенных тестовых заданий и не учитывает управляющего и внешнего воздействия на процесс тестирования — налицо явная идеализация. Это подтверждается экспериментами [4] и множеством критических публикаций в адрес эффективности метода КТ.

Работы, посвященные совершенствованию обучающего эффекта систем КТ, проводились многими исследователями. Так, например, в пакете АСТ осуществляется динамическое формирование тестовой выборки [5], в работе [4] предложена методика расчета поправки оценок, а в [6] применен метод нечеткой логики. В подобных системах тесты остались ориентированными на классический подход. Данная проблематика приводит к необходимости пересмотра целей организации КТ в учебном процессе, а также к выработке новой методики тестирования и совершенствованию соответствующего программного обеспечения.

Выделим в дереве целей основную цель учебного процесса — обучение, и сделаем ее доминантой при рассмотрении вопросов автоматизации. Тогда целью КТ будет повышение эффективности и управляемости учебного процесса. Упоминание обучающей роли тестов можно встретить в ряде исследований [например, 2, 8], но там функция обучения сводится исключительно к выводу подсказок либо переходу к учебному материалу, т. е. тестирование (контроль) остается доминантой (например, программные пакеты Прометей и ГИПЕРТЕСТ).

Исходя из того, что контроль и оценка знаний являются подпроцессами системы обучения, введем понятие "обучающее компьютерное тестирование" (ОКТ). Под ним будем понимать такую систему КТ, которая ориентирована на достижение оптимальных показателей обучения для отдельного пользователя посредством функций оценки знаний, адаптации, а также интеграции с ЭУК [8]. В сравнении с классическим КТ обучающее компьютерное тестирование можно характеризовать следующим образом:

<sup>3</sup> Тестовая выборка — множество предъявляемых пользователю вопросов на этапе решения теста.

- активное участие преподавателя на этапе разработки тестов и ЭУК: их создание и интеграция, выработка модели учебной дисциплины и параметров алгоритмов оценивания и адаптации, формирование базы знаний;
- ведущая роль обучаемого как при тестировании (самодиагностике), так и при обращении к учебному материалу (самообучении);
- процессы тестирования и обучения не ограничены по последовательности и интенсивности обращения;
- интеграция в рамках ЭУК учебно-практического материала, тестов и справочного материала, которые управляются моделью учебного курса;
- доминантой процесса тестирования является обучение, но с поддержкой возможности осуществления эффективного итогового контроля;
- системный (комплексный) подход к оценке результатов ОКТ и формированию обратной связи с ЭУК в соответствии с целями обучения.

На наш взгляд, при ОКТ ведущую роль должен играть ЭУК, содержащий следующие объекты: рабочую программу учебного курса, теоретический блок курса, лабораторный практикум с циклом исследовательских задач, обучающие компьютерные тесты, базу знаний. Данные объекты объединяются программой, реализующей функции управления, анализа данных и адаптации тестовой выборки посредством набора настраиваемых моделей учебных дисциплин (базы знаний) и базы данных (БД). Само обучающее тестирование можно дополнить следующими признаками: разделены режимы обучения<sup>4</sup> и контроля; присутствует банк тестовых заданий большой мощности; тестовая выборка генерируется случайным образом в соответствии с балансом тематических разделов курса; осуществляется индивидуализация тестов посредством механизма адаптации [8].

Для оптимизации педагогического эффекта при ОКТ на этапе формирования тестовой выборки используется функция адаптации; затем регистрируется набор факторов, помогающих оценить сам процесс тестирования; и, наконец, на этапе оценивания осуществляются анализ и принятие решения о значении выставляемой оценки и сохранение параметров подстройки теста в памяти системы ОКТ. Модуль обучающего адаптивного компьютерного тестирования (ОАКТ) будем рассматривать как сложную систему.

<sup>4</sup> Режим обучения включает работу с подсказками, правильными ответами, литературой, электронными учебниками и может не ограничиваться по времени.

## Системный анализ процесса обучающего тестирования

При рассмотрении системы ОАКТ как объекта системного анализа выдержим все формальные этапы метода обследования [3, 9]. Согласно [3] систему ОАКТ можно отнести к классу сложных, так как она обладает свойствами робастности (модульная слабосвязная архитектура), неоднородности (сочетание программных, модельных и учебных компонентов), эмергентности (формирование замкнутой образовательной среды) и нуждается в целенаправленном управлении. Так как результаты тестирования есть точечная оценка во времени, то систему причислим к разряду дискретных. Что касается наличия неоднозначности процесса оценивания и учета многофакторности ситуации тестирования, то исследуемая система будет характеризоваться как открытая и сложная [10].

Отметим наличие ряда основных *свойств* ОАКТ, которые принято выделять при анализе систем:

- целостность и относительная обособленность от окружающей среды;
- наличие связей с внешней средой;
- делимость;
- взаимосвязь подсистем;
- эмерджентность;
- вложенность (надсистемой будет ЭУК);
- обусловленность целью (повышение эффекта обучение) и динамичность системы.

Первым приближением на пути детализации системы ОАКТ будет *модель черного ящика* (рис. 1). Если рассмотреть потоки данных, участвующие в процессе обучающего тестирования, то их, как уже было отмечено, можно разделить на три части: входные  $X(t_i)$ , управляющие  $U(t_i)$  и внешние  $\Omega_i$ , где  $i$  — индекс очередного по времени сеанса тестирования.

Поток данных  $X(t_i)$  представляет собой вектор, состоящий из адаптированной тестовой выборки, данных статистики успехов тестируемого, модели курса (базы знаний), начальных условий очередного теста (ограничение по времени, цели тестирования, качество и валидность тестового задания). Поток данных  $\Omega_i$  также можно представить вектором, включающим следующие вероятностные факторы: проявление в ходе тестирования пользователем знания (число верно решенных тестовых заданий), регистрация параметров процесса тестирования (скорость ответов, использование подсказок, переход к ЭУК), неконтролируемые воздействия окружающей среды (готовность обучающегося, наличие внешних подсказок и пр.). Набор факторов, формирующих вектор управ-

ляющей информации  $U(t_i, \alpha)$ , где  $\alpha$  — параметр адаптации, помогает осуществить ситуационную оценку процесса тестирования, т. е. подстройку параметров теста под индивидуальный уровень знаний обучающегося, вызов связей с учебным материалом в режиме обучающего тестирования посредством модели курса, структурную адаптацию тестовой выборки. Конкретизируем понятия "модель курса" и "адаптация".

Задача *модели курса и базы знаний* заключается в подстройке баланса параметров системы посредством анализа совокупности обрабатываемой информации и последующей адаптации. В модель курса включаются: структура учебной дисциплины (предметная область); модель отображения структуры дисциплины на читаемый курс с учетом цели курса<sup>5</sup> (по учебному плану); параметры процесса тестирования (эталонные баллы, время решения теста); набор баз знаний для комплексного анализа рассматриваемых факторов; система связей (ссылок) различных компонентов ЭУК с учебным материалом и заданиями ОКТ.

Под процессом *адаптации* понимается такое изменение структуры тестовой выборки, при котором обучающее воздействие будет носить оптимальный характер: эталонное состояние знаний изучаемой дисциплины сравнивается с текущим и, в случае отклонения, осуществляется перераспределение тестового задания в выборке в целях повышения сложности всего теста. Мы вернемся к этой теме при рассмотрении управляемости системы ОАКТ.

Выходной информацией системы ОАКТ станет оценка знаний  $Y(t_i)$ , предъявляемая пользователю по окончании обучающего теста. Оценка выводится на основании комплексного анализа доступных факторов с учетом данных из предыдущей статистики успеваемости [11, 12].

Осуществим декомпозицию тестирующей подсистемы<sup>6</sup> ОАКТ по функциональному признаку, построив *модель состава*. Опираясь на необходимость усиления обучающего эффекта ОКТ, выделим следующие структурные подсистемы:

- генерирования тестовой выборки;
- тестирования (предъявление вопросов и текущий мониторинг);

<sup>5</sup> Необходимо учитывать особенности читаемого курса, так как, например, содержание дисциплины "теория экономических систем" для студентов-экономистов и студентов-информатиков будет различаться по степени детализации отдельных ее разделов.

<sup>6</sup> Здесь упускаются подсистемы группового анализа статистического материала, генерации сводной отчетности, шифрования, конструирования обучающих тестов и пр., как вспомогательные.

- получения помощи и перехода к учебному материалу (обучение);
- статистического и экспертного анализа полученной оценки с осуществлением последующих поправок балла (параметр  $\alpha$ );
- отображения балла на конкретную оценочную шкалу и предъявление его пользователю.

Можно заметить, что каждый блок характеризует отдельные этапы жизненного цикла процесса тестирования и тесно взаимосвязан с моделью учебного курса и хранимой в ЭУК метаинформацией. Под метаинформацией здесь понимается совокупность данных из базы данных (БД), содержащих сведения о динамике успехов обучаемого, отдельных взаимосвязях модели курса с компонентами ЭУК, а также индивидуальные подстройки среды обучения (профили).

Проведем синтез модели черного ящика и модели состава для получения *модели структуры* тестирующей подсистемы ОАКТ (рис. 2). Сплошными линиями обозначены основные информационные потоки, а штриховыми — вспомогательные, которые призваны координировать выполнение надлежащей функциональности системы ОАКТ.

Центральную роль при формировании тестовой выборки играет связь блоков "Генератор тестовой выборки"—"Модель курса", так как от нее в основном зависит сложность теста, его структура и, как следствие, последующий обучающий эффект  $U(t, \alpha)$ . Другая важная двунаправленная связь соединяет блоки "Тестирование"—"Обучение"; она характерна тем, что позволяет переходить от тестирования к обучению по инициативе пользователя, максимально сокращая путь от непонятого тестового задания к конкретному элементу учебного материала. Следует подчеркнуть, что основная нагрузка ложится на алгоритмы обработки информации, которые позволяют принять решение об адекватном отклике системы ОАКТ на действия пользователя (оценка начальных условий, отбор и анализ данных из БД, реализация перехода по связям тестовых заданий с учебными элементами ЭУК). Агрегирование трех рассмотренных моделей дает структурную схему процесса обучающего тестирования на этапе его непосредственного выполнения.

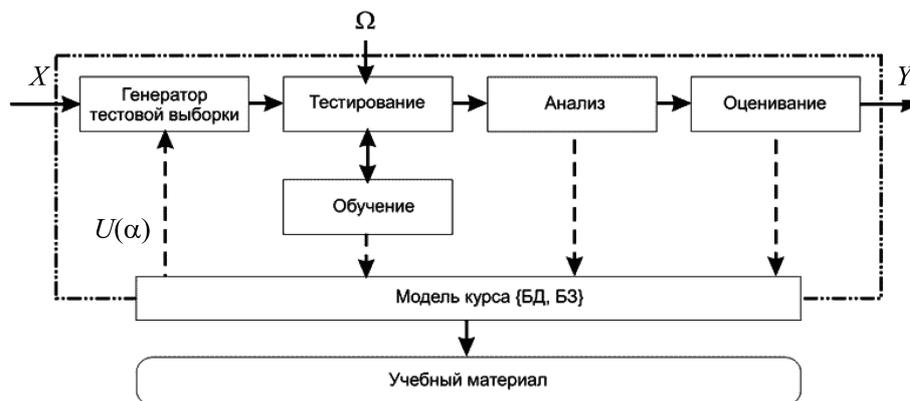


Рис. 2. Структурная схема процесса обучающего тестирования

*Динамику* системы можно рассмотреть с двух уровней приближения. На макроуровне система ОАКТ не является статичным дополнением ЭУК и подвержена расширению и настройке: для каждой реализации учебного курса ОКТ будет иметь собственное наполнение, которое определяется составителем ЭУК в соответствии с целями проведения курса и его моделью. На микроуровне динамика проявляется в процессе обучающего тестирования, когда функция адаптации подстраивает содержимое тестовой выборки так, чтобы она позволила максимально эффективно выполнять обучение и контроль (адаптация с памятью).

Раскрывая динамику системы ОАКТ, нельзя не упомянуть проблему *управляемости* процесса тестирования. В отличие от классического тестирования, в обучающем тестировании предусмотрена специальная функция адаптации тестовой выборки. Традиционно под адаптацией системы КТ понимается такое комплектование тестовой выборки заданиями, при котором эффективность контроля будет высокой. Для ОАКТ адаптация направлена на выявление наиболее слабо подготовленных разделов учебного курса с возможностью последующего перехода непосредственно к учебному материалу и корректировки следующей тестовой выборки в сторону усложнения для конкретного тестируемого [12].

Мы расширяем общее соотношение теста как подмножества банка тестовых заданий (БЗТ) в виде  $D \subset \text{БЗТ}$  с помощью введения дополнительных параметров, позволяющих включать механизмы управления структурой теста, где  $D$  есть реализация тестовой выборки. Перечислим настраиваемые параметры, с помощью которых можно моделировать и управлять структурой тестовой выборки ( $m$ -выборки из БЗТ). Пусть  $m$  — размер выборки из БЗТ или число тестовых заданий в тестовой выборке;  $n$  — объем банка тестовых за-

даний;  $k$  — число разделов предметной области, т. е.  $n = n_1 + n_2 + \dots + n_k$ . Так как нас интересует надежность тестирования и его управляемость, то мы должны моделировать<sup>7</sup>:

- возможные различные неупорядоченные  $m$ -выборки из множества  $n$  элементов  $N(n, m)$  — число сочетаний без учета разбиения  $n$ -множества на разделы;
- возможные различные неупорядоченные  $m$ -выборки из множества  $n$  элементов  $N(n, m, k)$  — число сочетаний с учетом разбиения  $n$ -множества на разделы;
- процесс тестирования через задание априорного распределения вероятностей на множестве разделов предметной области  $P = (p_1, p_2, \dots, p_k)$  и затем, после оценки результатов тестирования, генерирования нового распределения  $P' = (p_1, p_2, \dots, p_k)'$  и, следовательно, получение нового теста  $D'$  как результат адаптации структуры теста к результатам тестирования. Каждая реализация  $D$  при формировании использует алгоритм случайного выбора тестового задания для каждого  $n_k$ . Тогда процесс адаптации можно записать в виде

$$D' = f(X(D), U(\alpha), \Omega, t_i), \quad (1)$$

где  $D'$  — адаптированная выборка. Таким образом, после каждого следующего тестирования в режиме обучения тестовая выборка будет подстраиваться к знаниям обучаемого. При этом упор делается на непоследовательный процесс тестирования, где необходимость получения учебной информации кратчайшим путем важнее, чем очередная промежуточная оценка.

**Критерием эффективности** функционирования системы ОАКТ будет результат освоения учебного материала пользователем относительно того стандарта (цели), который лежал в основе реализации каждого ЭУК; он определяется в подсистеме тестирования на основе текущих баллов и динамики успеваемости с учетом системы поправок. В связи с этим на этапах адаптации и оценивания автоматическое принятие решений сопряжено с рядом объективных трудностей. В первую оче-

<sup>7</sup> Вопрос моделирования структуры случайной тестовой выборки из БТЗ мощности  $n$  подробно рассмотрен в [13].

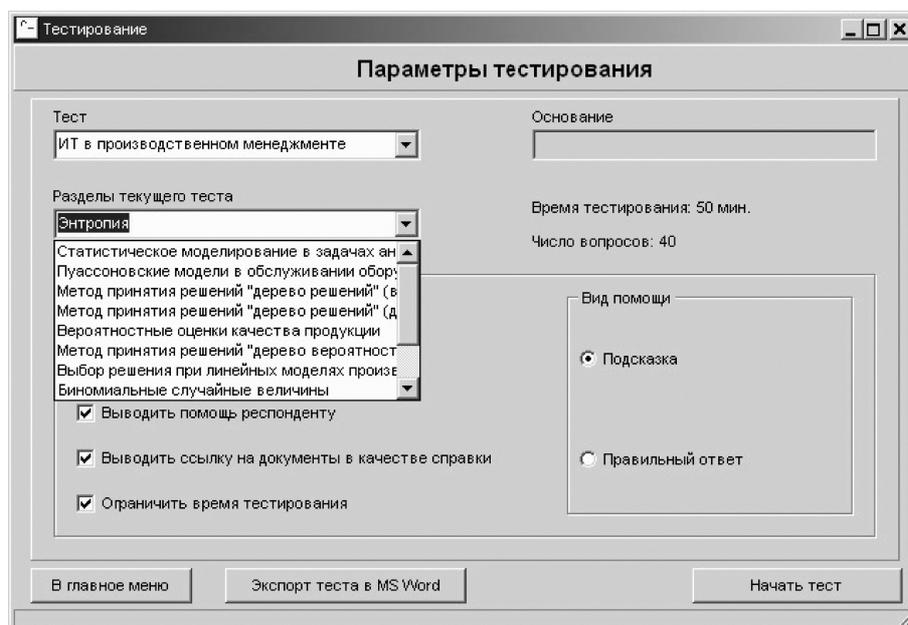


Рис. 3. Окно начальных параметров модуля "Тестирование" программы СиТест

редь, это неоднозначность параметров ситуации тестирования  $\Omega$ . Поэтому необходимо учитывать следующие виды неопределенностей [3]:

- стохастическую (*randomness*) — вероятность угадывания верных ответов;
- расплывчатую (*fuzziness*) — неоднозначность перевода в различные шкалы оценивания, учет важности отдельных факторов;
- частотно-временную (*time-and-frequency uncertainty*) — анализ динамики успеваемости обучаемого по отдельным результатам из данных статистики.

Следовательно, все неопределенности должны комплексно анализироваться на множестве возможных для измерения и учета факторов, позволяющих наиболее адекватно оценить результаты прохождения теста [14, 15]. Сюда включается как итоговый балл, так и вспомогательные факторы процесса тестирования (время обдумывания ответов, распределения верных ответов и пр.), а также динамика успеваемости:

$$Y = p(D', U(\alpha), \Omega, h, R, t_i), \quad (2)$$

где  $h$  — число верно решенных тестовых заданий;  $R$  — динамика успеваемости.

На наш взгляд, для повышения помехоустойчивости и оптимизации управляющего воздействия на систему рационально использовать сочетание методов статистики и искусственного интеллекта.

### Результаты исследования

Авторами разрабатывается программный пакет обучающего адаптивного компьютерного тести-

рования СиТест (рис. 3) [16], опирающийся на полученную структурную схему ОАКТ. В его основу были положены такие формы обучающего воздействия, как индивидуализация преподавания учебно-контрольного материала, мотивация обучения за счет механизмов подсказок, адаптации структуры ОКТ, целенаправленного перехода к отдельным элементам ЭУК.

Результаты проделанного анализа позволили выявить, что эффективно повлиять на процесс обучения с помощью ОКТ можно на следующих этапах:

- формирование тестовой выборки (процесс адаптации) — экспертно-аналитическая обработка профиля пользователя в совокупности с данными из модели курса и БД, позволяющая осуществить адаптацию тестовой выборки к индивидуальным особенностям обучаемого по формуле (1);
- прохождение теста (процесс обучения) — обеспечение возможности непосредственного перехода к учебному материалу ЭУК;
- оценивание — экспертно-аналитическая обработка результатов прохождения теста, осуществляющая поправку выставляемой оценки и сохранение в БД информации для последующей адаптации тестовой выборки по формуле (2).

Если работа второго этапа очевидна (ссылочная навигация по учебному материалу с помощью модели курса), то для реализации оставшихся этапов требуется серьезный анализ. Было принято решение использовать механизм экспертных систем (ЭС) как наиболее адекватный при анализе большого массива разнородной информации [17]. Для сглаживания и обобщения  $X(t)$  и  $\Omega$  используются методы статистики и нечеткой логики.

При осуществлении агрегации блоков структурной схемы (см. рис. 2) возникла необходимость интеграции в рамках одного приложения нескольких ЭС с возможностью их дальнейшей гибкой настройки профессионалами (экспертами-педагогами и инженерами по знаниям) независимо от работы программистов. Для этого мы предлагаем использовать модульную архитектуру: откомпилированная исполняемая программа связывается с внешними модулями ЭС посредством специальных компонентов [18]. Был разработан конструктор экспертных систем FLM\_builder и специальные компоненты для среды программирования — FLM\_modul. Это позволило выработать соответствующую структуру базы знаний ЭС и такой алгоритм преобразования данных в модуле тестирования системы ОАКТ, при котором ЭС воздействуют на информацию о тесте как на этапе адаптации ( $D \rightarrow D'$ ), так и на этапе оценивания

( $h \rightarrow Y$ ) [19]. Данный подход позволяет комплексно подойти к проблеме обучения посредством автоматизированных систем и сделать управляющие воздействия более эффективными.

## Выводы

На наш взгляд, осуществление автоматизированного педагогического контроля в изоляции от среды обучения малоэффективно. Для разрешения этой проблемы предлагается использовать обучающее тестирование [7]. Авторами разрабатывается соответствующее программное обеспечение (СиТест, FLM\_builder, FLM\_modul), которое позволит интегрировать электронные учебные курсы и компьютерное тестирование, а также осуществлять комплексное оценивание результатов прохождения тестов и обратную связь с учебным материалом. Эффективность применения обучающего компьютерного тестирования достигается за счет комплексного подхода к обучению и оценке в условиях дефицита квалифицированного педагогического состава, что особенно важно при осуществлении самостоятельного и дистанционного образовательного процесса.

## Список литературы

1. Орел Е. Создание тестов и компьютерное тестирование / Е. Орел, А. Прохоров // Компьютер-Пресс, № 7. 2006. — С. 94–99.
2. Пантелеев Е. Р. Средства поддержки жизненного цикла Web-обучения в инструментальном комплексе ГИПЕРТЕСТ 2.0 / Е. Р. Пантелеев // Информационные технологии, № 2. 2007. — С. 39–45.
3. Перегудов Ф. И. Основы системного анализа: Учеб. 2-е изд., доп. / Ф. И. Перегудов, Ф. П. Тарасенко — Томск.: НТЛ, 1997. — 396 с.
4. Половко А. М. Компьютерные технологии оценки знаний методами тестирования / А. М. Половко // Информационные технологии, № 8. 2004. — С. 46–51.
5. Тягунова Т. Н. Философия компьютерного тестирования / Т. Н. Тягунова — М.: МГУП, 2003. — 246 с.
6. Данилова С. Д. Адаптивная, нечеткая модель оценивания результатов автоматизированного тестирования с разделением заданий по уровням усвоения: дис. ... канд. тех. наук: 05.13.01 / С. Д. Данилова — М., 2005. — 153 с.
7. Аванесов В. С. Методические и теоретические основы тестового педагогического контроля: дис. ... док. пед. наук: 13.00.01 / В. С. Аванесов — М., 1994. — 326 с.
8. Углев В. А. Обучающее компьютерное тестирование / В. А. Углев // Теоретические и прикладные вопросы современных информационных технологий: Материалы VIII Всероссийской научно-технической конференции. — Улан-Удэ: ВСГТУ, 2007. — С. 312–316.
9. Анфилатов В. С. Системный анализ в управлении: Учеб. Пособие / В. С. Анфилатов, А. А. Емельянов, А. А. Кукушкин — М.: Финансы и статистика, 2003. — 368 с.
10. Месарович М. Общая теория систем: математические основы: Пер. с англ. / М. Месарович, Я. Такахара — М.: Мир, 1978. — 312 с.

11. Рудинский И. Д. Модель статистического оценивания знаний / И. Д. Рудинский, С. В. Грушецкий // Информационные технологии. 2004. — № 12. — С. 48—54.

12. Углев В. А. Модель адаптации при обучающем адаптивном компьютерном тестировании / В. А. Углев // Теоретические и прикладные вопросы современных информационных технологий: Материалы Всероссийской научно-технической конференции. — Улан-Удэ: ВСГТУ, 2006. — С. 238—242.

13. Углев В. А. Оценка возможностей случайных тестовых выборок / В. А. Углев, В. А. Устинов // Вестник Хакасского технического института — филиала КГТУ, № 21. Абакан, 2006. — С. 260—269.

14. Углев В. А. К вопросу об эффективности анализа факторов при обучающем компьютерном тестировании / В. А. Углев, В. А. Устинов // Информационные технологии в образовании и науке: Материалы Всероссийской научно-практической конференции. — М.: МФА, 2007. Ч. 1. — С. 203—208.

15. Углев В. А. Модель оценивания при обучающем адаптивном компьютерном тестировании / В. А. Углев // Информационно-вычислительные технологии и их приложения: сборник статей V Международной научно-технической конференции. — Пенза: РИО ПГСХА, 2006. — С. 312—315.

16. Углев В. А. Система обучающего адаптивного компьютерного тестирования. — М.: ОФАП, 2006. — номер государственной регистрации (ЕСПД) — 03524577.01414—01.

17. Углев В. А. Обучающее адаптивное тестирование с применением экспертных систем / В. А. Углев // Информационные технологии в образовании и науке: Материалы Всероссийской научно-практической конференции: В 3 ч. Ч. 3. — М.: МФА, 2006. — С. 606—611.

18. Углев В. А. Разработка экспертных систем с применением внешних модулей / В. А. Углев, Б. С. Добронец // Молодежь и наука: начало XXI века: Материалы Всероссийской научно-технической конференции студентов, аспирантов и молодых ученых: В 3 ч. Ч. 1. — Красноярск: ИПЦ КГТУ, 2006. — С. 305—306.

19. Углев В. А. Подход к интеграции модульных экспертных систем с пользовательскими приложениями на примере создания проекта для адаптивного тестирования / В. А. Углев // Современная техника и технологии: Материалы XIII Международной научной конференции. В 3 т. Т. 2. — Томск: ТПУ, 2007. — С. 454—456.

*Новости от IBM*

## **В Уфимском государственном авиационном техническом университете запущен самый мощный в России суперкомпьютер IBM**

Уфимский государственный авиационный технический университет (УГАТУ), IBM, компания АйТи и корпорация Intel объявили о запуске самого мощного суперкомпьютера в России, построенного на основе blade-серверов IBM.

Суперкомпьютер будет использоваться в инновационной программе обучения "Компьютерное моделирование с применением суперкомпьютерных технологий", а также в преподавании других дисциплин, связанных с информационными технологиями, параллельным программированием и компьютерным моделированием.

Основу суперкомпьютера УГАТУ составляют 271 blade-сервер IBM на базе 542 четырехъядерных процессоров Intel® Xeon® серии 5345. Расчетная пиковая производительность суперкомпьютера составляет 20 Тфлопс (триллионов операций с плавающей запятой в секунду). Запущенный суперкомпьютер является на сегодня самым производительным вычислительным кластером IBM на территории России и СНГ.

Уникальными особенностями суперкомпьютера УГАТУ являются его компактность (шесть монтажных шкафов, в которых размещены шасси

с блейд-серверами, управляющие узлы кластера и сеть хранения данных), низкое энергопотребление (потребляемая мощность оборудования кластера — порядка 85 кВт).

Объединение blade-серверов в единую систему выполнено с помощью высокоскоростной технологии Infiniband на базе 288-портового коммутатора Cisco с неблокируемой коммутационной матрицей.

Для отвода теплоты от оборудования в закрытом пространстве серверных стоек используется модульная система воздушно-водяных теплообменников Rittal LCP.

В суперкомпьютере УГАТУ использованы операционные системы Red Hat Enterprise Linux 4 AS/WS и специализированное ПО управления кластерными системами IBM Cluster Systems Management, IBM Tivoli Workload Scheduler, General Parallel File System (GPFS), а также ПО для обеспечения информационной безопасности Tivoli Access Manager for Enterprise Single Sign-On (TAM SSO) с дополнительными модулями Provisioning Adapter, Desktop Password Reset Adapter и Authentication Adapter.

# CONTENTS

<b>Kudelya V. N.</b> <i>The Optimization of Sorting Method on Graph and the Problems of Control by Telecommunication Network</i> . . . . .	2
<b>Zhukov O. D.</b> <i>Logarithmic System on the Base of Modular Algebra</i> . . . . .	7
<b>Mandrikova O. V., Bogdanov V. V., Polozov Yu. A.</b> <i>Modeling of Complicated Structure Signals on the Basis of Wavelet-Transform Constructiop</i> . . . . .	12
<b>Alguliyev R. M., Orujov G. H., Sabziev E. N., Panahov N. A., Rasulova N. V., Aliyeva A. A.</b> <i>Development of Detecting and Updating Algorithms of Some Characteristic Errors for a Vector Map of a Lay of Land</i> . . . . .	19
<b>Esakov D. I.</b> <i>Union of Several Methods at the Automated Processing of Earth Remote Sensing Images</i> . . . . .	24
<b>Petrov A. V., Guzev V. B.</b> <i>MC# — General Purpose Parallel Programming Language</i> . . . . .	29
<b>Sinitskij M. B., Sinitskij B. M.</b> <i>The Innovative Approach to Development Cross-Country-Platform of Appendices</i> . . . . .	32
<b>Kukhareenko B. G.</b> <i>Spectral Analysis Technology Based on Fast Prony Transform</i> . . . . .	38
<b>Vilenchik L. S., Ivanov Yu. V., Trofimov V. P.</b> <i>About Mathematical Modelling of Dynamics of Natural and Physical Systems Duscretely-Continuous by Systems with Delaying Argument</i> . . . . .	43
<b>Kulikov G. E.</b> <i>Automated Data Processing and Acquisition System For Experimental Physical Adjustments</i> . . . . .	50
<b>Chernyshov A. V.</b> <i>Automation generation of test programs for onboard telemetry equipment</i> . . . . .	56
<b>Sokolov E. M., Panarin V. M., Zuykova A. A., Bizikin A. V.</b> <i>The Automated System Ecological Monitoring of the Atmosphere at Emissions of Harmful Substances</i> . . . . .	58
<b>German A. S.</b> <i>Process of Securing of Information and Telecommunication Medium for Data Transmission and Video Conferencing in Public Services Bodies</i> . . . . .	62
<b>Panteleev E. R., Suvorov V. A.</b> <i>Online Analytical Processing of Student Activities in a Web-Learning Environment</i> . . . . .	67
<b>Kardanova E. Yu., Karpinsky V. B.</b> <i>Special Methods of the Analysis of Testing Results Based on the Objectivity Property of Rush's Models</i> . . . . .	72
<b>Ughev V. A., Ustinov V. A., Dobronets B. S.</b> <i>The System Analysis of the Process of Learning Computer Testing</i> . . . . .	81

---

---

## Адрес редакции:

107076, Москва, Стромьинский пер., 4/1

Телефон редакции журнала **(495) 269-5510**

E-mail: [it@novtex.ru](mailto:it@novtex.ru)

Дизайнер *Т.Н. Погорелова*. Художник *В.Н. Погорелов*.  
Технический редактор *О. А. Ефремова*. Корректор *О. А. Шаповалова*

Сдано в набор 08.02.2008. Подписано в печать 18.03.2008. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.

Усл. печ. л. 10,78. Уч.-изд. л. 11,84. Заказ 300. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Отпечатано в ООО "Подольская Периодика"  
142110, Московская обл., г. Подольск, ул. Кирова, 15