

# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

7(191)  
2012

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ  
Издательство "Новые технологии"

## СОДЕРЖАНИЕ

### МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Кухаренко Б. Г., Пономарев Д. И. Аппроксимация смесью Гауссовых распределений в модели переключающегося фильтра Калмана для идентификации режимов колебаний временных рядов . . . . . 2
- Андреев Д. В. Универсальный логический модуль для обработки многозначных данных . 7
- Сандуляну Л. Н., Стрижов В. В. Выбор признаков в авторегрессионных задачах прогнозирования . . . . . 11
- Чепурко В. А., Чепурко С. В. Непараметрическая оценка коэффициента деградации геометрических процессов . . . . . 16
- Оцоков Ш. А. Метод проверки необходимости округления при организации высокоточных вычислений в модулярной арифметике . . . . . 21

### ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Косяков С. В., Гадалов А. Б., Садыков А. М. Моделирование пространственных данных при решении задач дискретной оптимизации в среде ГИС . . . . . 27
- Астафуров В. Г., Скороходов А. В. Нейросетевой классификатор облачности по спутниковым данным . . . . . 32

### СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Кривошеин Д. Ю., Марченко А. М. Инкрементальный алгоритм поиска кратчайших путей в графе . . . . . 38

### АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ

- Мартынов П. Н., Ковшов Е. Е. Разработка средств автоматизации тестирования интерфейсов пользователя в человеко-машинных системах управления . . . . . 42
- Бражник С. А., Малафеев С. И. Синхронизация в промышленных контроллерах с операционной системой Linux . . . . . 47

### ДИСКУССИОННЫЙ КЛУБ

- Вяткин В. Б. Информационно-квантовые характеристики и отраженные образы конечных множеств . . . . . 50

### ОБМЕН ОПЫТОМ

- Федорова Е. В., Гетьман М. А., Савельева Е. В. Использование облачных технологий Google Apps Education Edition для создания саморазвивающейся информационной платформы вуза . . . . . 57

### Журнал в журнале НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- Скрибцов П. В., Казанцев П. А., Червоненкис М. А. Применение искусственных нейронных сетей для решения обратных задач в гидрологии . . . . . 62
- Борисов В. В., Мисник А. Е. Комбинированный нейросетевой способ моделирования для оперативного управления сложными системами . . . . . 69
- Степанов С. Ю., Кабак И. С. Алгоритм фрагментации больших нейронных сетей и исследование его сходимости . . . . . 73
- Contents . . . . . 78
- Приложение Карпенко А. П. Популяционные алгоритмы глобальной поисковой оптимизации. Обзор новых и малоизвестных алгоритмов.

Главный редактор  
НОРЕНКОВ И. П.

Зам. гл. редактора  
ФИЛИМОНОВ Н. Б.

Редакционная  
коллегия:

- АВДОШИН С. М.  
АНТОНОВ Б. И.  
БАРСКИЙ А. Б.  
БОЖКО А. Н.  
ВАСЕНИН В. А.  
ГАЛУШКИН А. И.  
ГЛОРИОЗОВ Е. Л.  
ДОМРАЧЕВ В. Г.  
ЗАГИДУЛЛИН Р. Ш.  
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.  
НЕЧАЕВ В. В.  
ПАВЛОВ В. В.  
ПУЗАНКОВ Д. В.  
РЯБОВ Г. Г.  
СОКОЛОВ Б. В.  
СТЕМПКОВСКИЙ А. Л.  
УСКОВ В. Л.  
ФОМИЧЕВ В. А.  
ЧЕРМОШЕНЦЕВ С. Ф.  
ШИЛОВ В. В.
- Редакция:  
БЕЗМЕНОВА М. Ю.  
ГРИГОРИН-РЯБОВА Е. В.  
ЛЫСЕНКО А. В.  
ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.  
Журнал включен в систему Российского индекса научного цитирования.  
Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

УДК 519.246

**Б. Г. Кухаренко**, канд. физ.-мат. наук,  
ст. науч. сотр., вед. науч. сотр.,  
Институт машиноведения РАН, г. Москва,  
e-mail: kukharenko@imash.ru

**Д. И. Пономарев**, аспирант, Московский  
физико-технический институт (ГУ),  
e-mail: ponomarev-102@mail.ru

## Аппроксимация смесью Гауссовых распределений в модели переключающегося фильтра Калмана для идентификации режимов колебаний временных рядов

*Рассматриваются временные ряды со сменой режимов колебаний. Авторегрессионные модели метода Прони в пространстве состояний используются как динамические модели режимов колебаний временного ряда в переключающемся фильтре Калмана. Вероятности скрытых состояний аппроксимируются смесью Гауссовых распределений. Демонстрируется обнаружение последовательностей повторяющихся паттернов (жестов оператора) в управляющем сигнале дистанционного манипулятора.*

**Ключевые слова:** временные ряды, режим колебаний, метод Прони, авторегрессионная модель, переключающийся фильтр Калмана, модель смеси Гауссовых распределений, паттерн, дистанционный манипулятор

Для временного ряда присутствие паттернов проявляется в приблизительно повторяющихся последовательностях элементов этого временного ряда. В настоящей работе рассматриваются временные ряды, у которых эти приблизительно повторяющиеся последовательности элементов расположены подряд, т. е. сами образуют повторяющиеся серии. Исследуется, какие динамические модели могут использоваться для идентификации смены различных серий паттернов временного ряда.

### Почти совпадающие последовательности, паттерны и режимы колебаний временных рядов

Обнаружение паттернов (задача разведывания данных — Data Mining) временных рядов состоит в поиске почти совпадающих последовательностей

элементов во временном ряду  $x[\overline{1, N_0}]$ ,  $N_0 \gg 1$ . Для заданного временного ряда  $x[\overline{1, N_0}]$  длиной  $N_0$  последовательность  $S$  — это выборка длиной  $M \ll N_0$ , являющаяся непрерывной частью  $x[\overline{1, N_0}]$ . Все последовательности  $S_k[\overline{1, M}]$ ,  $k = \overline{1, (N_0 - M + 1)}$ , получаются из временного ряда  $x[\overline{1, N_0}]$  сдвигом временного окна длины  $M$ . Подобие двух последовательностей  $S_k, S_l$  с заданными  $k, l = \overline{1, (N_0 - M + 1)}$  определяется величиной некоторой меры расстояния  $\text{dist}(S_k, S_l)$ . Последовательности  $S_k, S_l$  считаются почти совпадающими, т. е. они представляют паттерн, если  $\text{dist}(S_k, S_l) < R$ , где  $R$  — значение порога [1]. Как правило, для последовательностей элементов из временных рядов используется Евклидова мера расстояния в пространстве размерности  $M$ , равной длине этих последовательностей. Когда временной масштаб  $M \ll N_0$  для обнаружения паттернов временного ряда  $x[\overline{1, N_0}]$  не известен, их поиск требует варьирования длины  $M$  временной последовательности [2, 3]. Это многократно увеличивает фактические временные затраты по сравнению с оценками временных затрат при фиксированной длине последовательностей.

Для временного ряда присутствие паттернов, представленных приблизительно повторяющимися последовательностями элементов этого временного ряда, предполагается его колебательное изменение. То есть существует некоторый характерный средний период колебательных изменений временного ряда, определяющий временной масштаб, в котором обнаруживаются паттерны. Для оценки этого характерного периода (временного масштаба) временного ряда могут использоваться методы определения спектров частот, несоизмеримых с частотой дискретизации временного ряда [4]. Например, одночастотная аппроксимация временного ряда по методу Прони оценивает его характерный период колебаний — временной масштаб для поиска его почти совпадающих последовательностей [5]. После определения временного масштаба почти совпадающие последовательности идентифицируются с использованием алгоритма обнаружения паттернов. Если конкретные паттерны временного ряда повторяются без промежутков, то у временного ряда наблюдаются режимы колебаний.

### Авторегрессия сегмента временного ряда как модель режима колебания

Процедура локального (во времени) спектрального анализа записи колебаний лопатки (временного ряда)  $x[\overline{1, N_0}]$  (определенного с шагом дискретизации времени  $\Delta t$ ) использует его сегментирование посредством сдвига временного окна фиксированной длины  $N \ll N_0$  ( $N\Delta t$  определяет временной масштаб для локального анализа этого нестационарного временного ряда). Для локально переходного временного ряда с колебательным изменением  $x[\overline{1, N_0}]$  имеется временной масштаб  $N\Delta t$ , такой, что в пределах последовательных сегментов

$$x[\overline{n_j, (n_j + N - 1)}], \quad n_j = NJ, \quad J = [0 : \text{round}(N_0/N)],$$

огнивающая амплитуды колебаний меняется монотонно. Такие сегменты (так же, как и сам временной ряд, локально во времени) характеризуется переходным колебательным изменением. Результаты спектрального анализа, соответствующего модели временного ряда, являются информативными (т. е. имеющими физическую интерпретацию). Поэтому для локально переходного временного ряда  $x[\overline{1, N_0}]$  с колебательным изменением используется декомпозиция Прони последовательных сегментов  $x[\overline{1, N}]$ , которая имеет вид

$$x[k] = \sum_{l=1}^L r[l](z[l])^{k-1}, \quad k = \overline{1, N}, \quad (1)$$

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;  $A[l]$ ,  $\varphi[l]$ ,  $f[l]$  и  $\delta[l]$  — соответственно амплитуда, фаза, частота и фактор демпфирования (логарифмический декремент с обратным знаком) спектральной составляющей [6]. В классическом алгоритме метода Прони полюса  $z[l]$  определяются как корни полинома

$$\sum_{i=0}^L a[L-i+1]z^i = \prod_{i=1}^L (z - z[i]), \quad a[0] = 1, \quad (2)$$

коэффициенты которого  $\{a[l], l = \overline{1, L}\}$  являются решением уравнений авторегрессии (autoregression)

$$\sum_{l=1}^L x[k-l]a[l] = -x[k], \quad k = L, L+1, \dots, N. \quad (3)$$

Для решения системы уравнений (3) она представляется в матричной форме

$$\mathbf{x} = -\mathbf{Y} \cdot \mathbf{a}, \quad (4)$$

где

$$\mathbf{x} = \begin{bmatrix} x[L] \\ x[L+1] \\ \dots \\ x[N] \end{bmatrix}; \quad \mathbf{Y} = \begin{bmatrix} x[L-1] & x[L-2] & \dots & x[1] \\ x[L-2] & x[L-3] & \dots & x[2] \\ \dots & \dots & \dots & \dots \\ x[N-1] & x[N-2] & \dots & x[N-L] \end{bmatrix};$$

$$\mathbf{a} = \begin{bmatrix} a[1] \\ a[2] \\ \dots \\ a[L] \end{bmatrix}.$$

Оценка вектора коэффициентов  $\mathbf{a}$  авторегрессии (линейной модели предсказания) (3) посредством решения переопределенной задачи (4) с  $N > 2L$  по методу наименьших квадратов имеет вид

$$\tilde{\mathbf{a}} = -(\mathbf{Y}^T \mathbf{Y})^{-1} \mathbf{Y}^T \mathbf{x} \quad (5)$$

и эквивалентна использованию псевдообращения матрицы данных (размерности  $L \times (N - L + 1)$ ) при решении (4). Квадрат ошибки авторегрессии (4)—(5) обозначается

$$\|\mathbf{x} + \mathbf{Y} \cdot \tilde{\mathbf{a}}\|_2 \stackrel{\text{def}}{=} (N - L + 1)\sigma^2, \quad (6)$$

где  $\|\cdot\|_2$  — Евклидова норма вектора.

Задача вычисления корней полинома (2) является принципиально неустойчивой с вычислительной точки зрения. Поэтому для спектрального оценивания записей колебаний используется матричный алгоритм сегментированного метода Прони [7—9]. Полюса  $z[l]$ ,  $l = \overline{1, L}$ , определяются непосредственно как обобщенные собственные числа двух редуцированных матриц

$$\mathbf{Y}^{(0)} = \mathbf{Y}[\overline{1, (N-L+1)}, \overline{1, (L-1)}],$$

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

Длина  $N$  последовательного сегмента  $x[\overline{1, N}]$  может быть любой, но лучшие результаты спектрального оценивания получаются, когда размер скользящего временного окна  $N\Delta t$  примерно равен двум средним периодам временного ряда  $x[\overline{1, N_0}]$ . Более плотная выборка значений спектральных параметров временного ряда получается за счет использования дробных сдвигов временного окна. При минимальном числе полюсов  $L = 2$  аппроксимация по методу Прони последовательных сегментов (гладкого) временного ряда  $x[\overline{1, N_0}]$  позволяет оценить временную зависимость частоты этого временного ряда. Однако определяемый в методе Прони вектор коэффициентов авторегрессии (5) имеет самостоятельное значение, поскольку (3)—(4) используются как модель режима колебания сегмента  $x[\overline{1, N}]$

временного ряда в переключающемся фильтре Калмана (switching Kalman filter model), который с вычислительной точки зрения является очень эффективным [10–13].

### Переключающийся фильтр Калмана (Switching Kalman filter)

Фильтр Калмана — это модель единственного режима колебаний со скрытым состоянием  $\mathbf{x}[k]$ . Эволюция скрытого состояния описывается вероятностью

$$p(\mathbf{x}[k]|\mathbf{x}[k-1]) \propto N(\mathbf{x}[k]|\mathbf{A} \cdot \mathbf{x}[k-1], \mathbf{Q}). \quad (7)$$

Наблюдения генерируются линейным Гауссовым пространством состояний (linear Gaussian state space), т. е. вероятность

$$p(\mathbf{y}[k]|\mathbf{x}[k]) \propto N(\mathbf{y}[k]|\mathbf{C} \cdot \mathbf{x}[k], \mathbf{R}), \quad (8)$$

где  $N(\mathbf{y}|\mathbf{x}, \mathbf{R})$  обозначает Гауссово (нормальное) распределение случайной переменной  $\mathbf{y}$  с параметрами:  $\mathbf{x}$  — средним и  $\mathbf{R}$  — ковариационной матрицей. В формулах (7) и (8)  $\mathbf{A}$  — квадратная матрица модели режима колебания,  $\mathbf{C}$  — матрица наблюдений, а  $\mathbf{Q}$  и  $\mathbf{R}$  — матрицы ковариации шума. При заданных матрицах  $\mathbf{A}$ ,  $\mathbf{Q}$ ,  $\mathbf{C}$ ,  $\mathbf{R}$  из (7)–(8) и наблюдениях  $\mathbf{y}[1, k]$  уравнения фильтра Калмана дают оценки скрытого состояния  $\mathbf{x}[k]$  и ковариации ошибки  $\mathbf{P}[k]$ . Для последовательных временных шагов эти оценки получаются рекурсивно и в два этапа: предсказание и коррекция. Динамическая модель фильтра Калмана получается из авторегрессии (3) (с коэффициентами, оцененными посредством (5)), когда она представляется в  $L$ -мерном пространстве состояний с векторами из элементов временного ряда в дискретные времена с запаздыванием

$$\mathbf{x}[k] = \mathbf{A} \cdot \mathbf{x}[k-1], \quad (9)$$

где

$$\mathbf{x}[k] = \begin{bmatrix} x[k] \\ x[k-1] \\ \dots \\ x[k-L] \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} a[1] & a[2] & \dots & a[L-1] & a[L] \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

$$\text{и } \mathbf{Q} = \begin{bmatrix} \sigma^2 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

с  $\sigma^2$  из (6). Фильтр Калмана предполагает исходные оценки  $\mathbf{x}[1]$  и  $\mathbf{P}[1]$  и вычисляет оценки  $\{\mathbf{x}[k], \mathbf{P}[k]\}$  на основе оценок и наблюдений  $\{\mathbf{x}[k-1], \mathbf{P}[k-1], \mathbf{y}[1, k]\}$ .

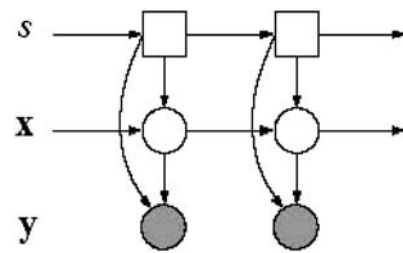


Рис. 1. Графическое представление модели переключающегося фильтра Калмана

Графическое представление модели переключающегося фильтра Калмана показано на рис. 1 ( $s$  — режим) [10–13].

Ниже представлен алгоритм переключающегося фильтра Калмана с  $K$  состояниями для вывода в переключающейся линейной динамической модели (9) с аппроксимацией вероятности скрытых состояний  $p(\mathbf{x}[k]|s[k], \mathbf{y}[1, k])$ ,  $s[k] \in \overline{1, K}$  смесью  $K$  Гауссовых распределений [14]. Для каждого переключения режима  $s[k-1] = i, s[k] = j, (i, j) \in \overline{1, K}$ , выполняются следующие шаги.

1. Посредством фильтра Калмана вычисляются наиболее правдоподобные оценки для временного шага  $k$  на основе оценок для предыдущего временного шага и известных параметров модели  $\mathbf{A}$  и  $\mathbf{Q}$ :

$$\mathbf{x}^-[k; i, j] = \mathbf{A}^{\{j\}} \cdot \mathbf{x}[k-1; i]; \quad (10)$$

$$\mathbf{P}^-[k; i, j] = \mathbf{A}^{\{j\}} \cdot \mathbf{P}[k-1; i] \cdot (\mathbf{A}^{\{j\}})^T + \mathbf{Q}^{\{j\}},$$

где  $\mathbf{x}^-[k]$  и  $\mathbf{P}^-[k]$  обозначают оценки в момент времени  $k$ , сделанные без учета наблюдения  $\mathbf{y}[k]$ . Наблюдение  $\mathbf{y}[k]$  используется для улучшения этих оценок

$$\begin{aligned} \tilde{\mathbf{y}}[k; i, j] &= \mathbf{y}[k] - \mathbf{C}^{\{j\}} \cdot \mathbf{x}^-[k; i, j]; \\ \mathbf{S}[k; i, j] &= \mathbf{C}^{\{j\}} \cdot \mathbf{P}^-[k; i, j] \cdot (\mathbf{C}^{\{j\}})^T + \mathbf{R}^{\{j\}}; \\ \mathbf{K}[k; i, j] &= \mathbf{P}^-[k; i, j] \cdot (\mathbf{C}^{\{j\}})^T \cdot (\mathbf{S}[k; i, j])^{-1}; \\ \hat{\mathbf{x}}[k; i, j] &= \mathbf{x}^-[k; i, j] + \mathbf{K}[k; i, j] \cdot \tilde{\mathbf{y}}[k]; \\ \hat{\mathbf{P}}[k; i, j] &= (\mathbf{I} - \mathbf{K}[k; i, j] \cdot \mathbf{C}^{\{j\}}) \mathbf{P}^-[k; i, j], \end{aligned} \quad (11)$$

в которых  $\tilde{\mathbf{y}}[k]$  обозначает инновацию (разницу между предсказываемыми и фактическим наблюдениями);  $\mathbf{S}[k]$  — ковариация инновации;  $\mathbf{K}[k]$  — матрица усиления Калмана, определяющая оптимальную корректировку оценки  $\mathbf{x}^-[k]$  при заданном наблюдении  $\mathbf{y}[k]$ . Отметим, что матрицы  $\mathbf{K}$ ,  $\mathbf{P}$  и  $\mathbf{S}$  (11) не зависят от наблюдений. Также отметим, что эффект этого шага предсказания всегда состоит в увеличении неопределенности оценки  $\mathbf{x}[k]$ . Акт наблюдения всегда снижает вариацию этой оценки (этого можно было бы ожидать).

2. При каждом переключении режима ( $i \rightarrow j$ ) вычисляется правдоподобие наблюдений  $\mathbf{y}[\overline{1}, \overline{N}]$

$$p(\mathbf{y}[k]|s[k-1]=i, s[k]=j) \propto N(\mathbf{y}[k]|\tilde{\mathbf{y}}[k; i, j], \mathbf{S}[k; i, j]) \quad (12)$$

и маргинальная вероятность

$$\begin{aligned} p(s[k-1]=i, s[k]=j|\mathbf{y}[\overline{1}, \overline{k}]) &\equiv \\ &\equiv p(s[k-1]=i, s[k]=j|\mathbf{y}[k], \mathbf{y}[\overline{1}, (k-1)]) \propto \\ &\propto p(\mathbf{y}[k]|s[k-1]=i, s[k]=j)p(s[k-1]=i, s[k]= \\ &= j|\mathbf{y}[\overline{1}, (k-1)]) = p(\mathbf{y}[k]|s[k-1]=i, s[k]=j)p(s[k] = \\ &= j|s[k-1]=i, \mathbf{y}[\overline{1}, (k-1)])p(s[k-1]= \\ &= i|\mathbf{y}[\overline{1}, (k-1)]), \end{aligned} \quad (13)$$

где  $p(s[k]=j|\mathbf{y}[\overline{1}, \overline{k}]) \equiv \sum_{i=1}^K p(s[k-1]=i, s[k]=j|\mathbf{y}[\overline{1}, \overline{k}])$ ,  $j = \overline{1}, \overline{N}$ , — вероятности, оцененные для идентификации установки режима в момент времени  $k$ . Предполагается, что вероятности переключения режима  $p(s[k]=j|s[k-1]=i, \mathbf{y}[\overline{1}, (k-1)])$ ,  $i = \overline{1}, \overline{K}$ ,  $j = \overline{1}, \overline{K}$ , независимы от наблюдений  $\mathbf{y}[\overline{1}, \overline{N}]$  и известны априори (дополнительные параметры переключающегося фильтра Калмана).

Поскольку на временном шаге  $k-1$  вероятность  $p(\mathbf{x}[k-1]|s[k-1], \mathbf{y}[\overline{1}, (k-1)])$  аппроксимируется смесью  $K$  Гауссовых распределений, то время, необходимое для вычисления вероятности  $p(s[\overline{1}, \overline{k}], \mathbf{x}[\overline{1}, \overline{k}], \mathbf{y}[\overline{1}, \overline{k}])$  в переключающемся фильтре Калмана растёт экспоненциально с  $k$  и делает вывод неуправляемым. Действительно, при вычислении апостериорной вероятности  $p(\mathbf{x}[k]|s[k], \mathbf{y}[\overline{1}, \overline{k}])$  в момент времени  $k$  учитываются вероятности переходов между каждой возможной комбинацией режимов в моменты времени  $k-1$  и  $k$ . Поэтому число Гауссовых распределений, необходимых для точного представления этой апостериорной вероятности, на каждом временном шаге  $k$  возрастает в  $K$  раз (число возможных переключений режима). То есть вычисление оценок и правдоподобий по Калману для каждой возможной установки режима  $s[k]$  приводит к апостериорной вероятности  $p(\mathbf{x}[\overline{1}, \overline{k}]|s[\overline{1}, \overline{k}], \mathbf{y}[\overline{1}, \overline{k}])$ , представляемой смесью  $K^2$  Гауссовых компонент. Описываемый алгоритм делает вывод управляемым посредством сокращения числа  $K^2$  компонент смеси Гауссовых распределений до  $K$  компонент на каждом временном шаге (дискретном моменте времени)  $k$  посредством процедуры совмещения средних и вариаций этого распределения на шаге 3 и 4 этого алгоритма, как описано в [11].

3. Вычисляются веса для каждого переключения режима:

$$w[i, j] = \frac{p(s[k-1]=i, s[k]=j|\mathbf{y}[\overline{1}, \overline{k}])}{\sum_{i=1}^K p(s[k-1]=i, s[k]=j|\mathbf{y}[\overline{1}, \overline{k}])}. \quad (14)$$

4. Вычисляются безусловные моменты (эта процедура называется совмещением моментов)

$$\mathbf{x}[k; j] = \sum_{i=1}^K w[i, j] \cdot \hat{\mathbf{x}}[k; i, j]; \quad (15)$$

$$\mathbf{P}[k; j] = \sum_{i=1}^K w[i, j] (\hat{\mathbf{P}}[k; i, j] +$$

$$+ (\hat{\mathbf{x}}[k; i, j] - \mathbf{x}[k; j])(\hat{\mathbf{x}}[k; i, j] - \mathbf{x}[k; j])^T).$$

В работах [15–16] показано, что смесь Гауссовых распределений с моментами (15) является наиболее близкой по мере Кульбака—Лейблера (Kullback—Leibler distance) к исходной смеси Гауссовых распределений (см. [16], Part 6 "Model for mixed data", P.162, формулы (6.7)—(6.9)). Переход к моментам (15) сокращает число  $K^2$  Гауссовых компонент вероятности  $p(\mathbf{x}[\overline{1}, \overline{k}]|s[\overline{1}, \overline{k}], \mathbf{y}[\overline{1}, \overline{k}])$  до  $K$  компонент [11].

#### Паттерны и режимы колебаний в записях управляющих сигналов манипулятора

В последнее время широкое распространение получают дистанционные манипуляторы, передающие движения руки оператора в трехмерном пространстве. В настоящей работе исследуются управляющие сигналы разрабатываемого авторами манипулятора (рис. 2), чувствительным элементом которого является прецизионный трехосевой MEMS-акселерометр (*Micro-Electro-Mechanical System*) [17]. Акселерометр обладает чувствительностью к земной гравитации, поэтому изменение наклона манипулятора относительно направления ускорения  $g$  свободного падения приводит к изме-

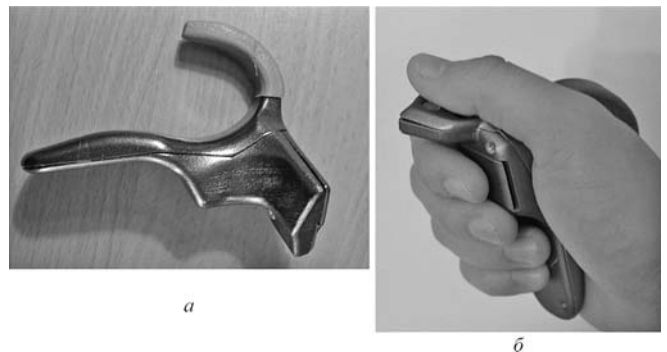


Рис. 2. Дистанционный манипулятор с прецизионным акселерометром: а — внешний вид; б — манипулятор в руке оператора

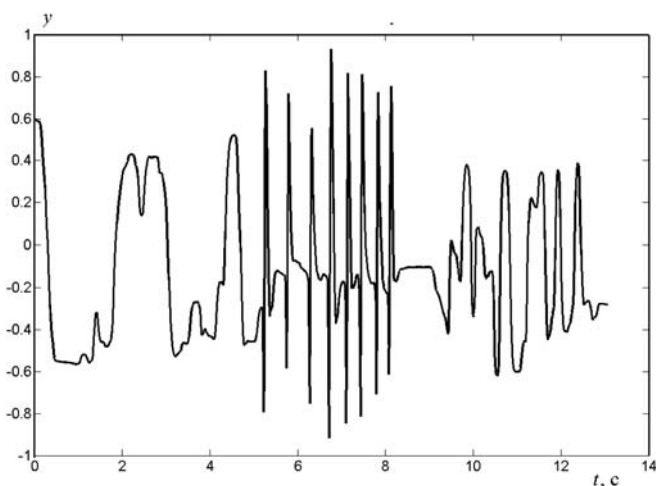


Рис. 3. Сигнал MEMS-акселерометра с тремя динамическими состояниями, представляющими три типа жестов

нению значений проекций этого ускорения на чувствительные  $x$ ,  $y$ ,  $z$ -оси акселерометра. Значения этих проекций используются как управляющие сигналы для позиционирования курсора (стандартно управляемого компьютерной мышью) на экране дисплея компьютера. Записи (временные ряды) управляющих сигналов дистанционного манипулятора используются для обнаружения паттернов. Паттерны записи управляющих сигналов манипулятора представляют произвольные движения — жесты руки оператора. Когда конкретные паттерны временного ряда повторяются без промежутков, у временного ряда наблюдаются режимы колебаний, поскольку последовательности одинаковых жестов руки оператора определяются особенностями построения этих движений [18—19].

Для обработки используется запись с чувствительной  $x$ -оси MEMS-акселерометра для трех последовательностей однотипных жестов руки оператора (рис. 3). Сигнал  $y = y(t)$  на рис. 3 представлен в безразмерных единицах относительно (гравитационного) ускорения свободного падения  $9,8 \text{ м/с}^2$ . Интервал дискретизации времени  $15 \text{ мс}$  (это соответствует частоте дискретизации  $1/0,015 = 66,6 \text{ Гц}$ ). Первый тип жестов соответствует медленным наклонным движениям кисти относительно направления силы тяжести. Такие жесты характерны для работы в операционной системе, где требуется точное наведение курсора на мелкие элементы графического интерфейса. Второй тип жестов представляет собой резкие горизонтальные движения руки слева направо с возвратом в исходное положение. Такие жесты используются для "перелистывания" страниц в графическом интерфейсе, например, для перелистывания фотографий. Третий тип жестов, так же как и первый, представляет собой наклонные движения, но более быстрые. Такие жесты характерны при использовании дистанционного манипулятора в компьютерных играх.

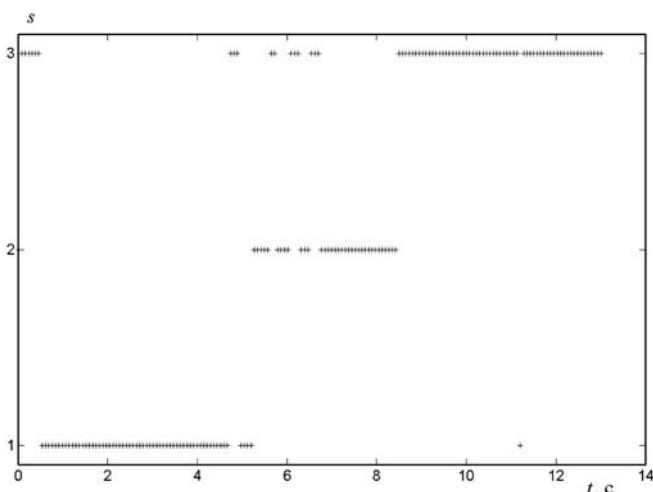


Рис. 4. Динамические состояния сигнала MEMS-акселерометра на рис. 3

Три типа жестов руки оператора соответствуют трем динамическим состояниям (режимам колебаний) сигнала MEMS-акселерометра, приведенного на рис. 3. На рис. 4 символом "+" показано наиболее вероятное состояние сигнала MEMS-акселерометра (см. рис. 3) на каждом временном шаге, выведенное посредством переключающегося фильтра Калмана (10)—(15).

На каждом временном шаге сигнал MEMS-акселерометра на рис. 3 проверяется на сходство с одним из режимов  $s = 1, 2, 3$  колебаний (в целом). Как видно на рис. 4, оказывается, что пологие участки сигнала на рис. 3 в начале последовательности второго типа жестов (для  $t \in [4,35; 6,53]$  с) интерпретируются посредством переключающегося фильтра Калмана как режим  $s = 3$  колебаний этого сигнала.

#### Список литературы

1. Chiu B., Keogh E., Lonardi S. Probabilistic discovery of time series motifs // Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. August 24—27, 2003. DC, Washington: ACM. 2003. P. 493—498.
2. Mueen A., Keogh E., Zhu Q., Cash S., Westover B. Exact discovery of time series motifs // Proceedings of the SIAM International Conference on Data Mining (SDM 2009). American Statistical Association (ASA). 2009. P. 473—484.
3. Mueen A., Keogh E. J. Online discovery and maintenance of time series motifs // Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2010), July 25—28, 2010. Washington, DC: ACM. 2010. P. 1089—1098.
4. Бернштейн Н. А. К анализу непериодических колебательных сумм с переменными спектрами по методу взвешенных решеток // Биофизика. 1962. Т. VII. Вып. 4. С. 377—381.
5. Кухаренко Б. Г., Пономарев Д. И. Использование метода Прони для оценки временного масштаба при обнаружении паттернов во временных рядах // Информационные технологии. 2012. № 1. С. 37—42.
6. Weiss L., McDonogh R. N. Prony's method, Z-transform, and Pade approximation // SIAM Review. 1963. V. 9. N 2. P. 145—149.
7. Sarkar T. K., Pereira O. Using the matrix pencil method to estimate the parameters of a sum of complex exponentials // IEEE Antenna and Propagation Magazine. 1995. V. 37. N 1. P. 48—55.
8. Kухarenko 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.

9. Кухаренко Б. Г. Технология спектрального анализа на основе быстрого преобразования Фурье // Информационные технологии. 2008. № 4. С. 38–42.

10. Ghahramani Z., Hinton G. E. Variational learning for switching state-space models // Neural Computation. 1998. V. 12. N 4. P. 963–996.

11. Murphy K. P. Switching Kalman Filter // Technical Report 98-10. UC Berkeley: Compaq Cambridge Research Laboratory. 1998.

12. Williams C. K. I., Quinn J. A., McIntosh N. Factorial switching Kalman filters for condition monitoring / Y. Weiss, B. Scholkopf, J. Platt, eds. Advances in Neural Information Processing Systems. V. 18. Cambridge, Massachusetts: The MIT Press. 2006. P. 1513–1520.

13. Quinn J. A., Williams C. K. I. Known unknowns: Novelty detection in condition monitoring // IbPRIA'07 Proceedings of the 3rd Iberian Conference on Pattern Recognition and Image Analysis. Berlin, Heidelberg: Springer-Verlag. 2007. Part I. P. 1–6.

14. Alspach D. L., Sorenson H. W. Nonlinear Bayesian Estimation Using Gaussian Sum Approximations // IEEE Transactions on Automatic Control. 1972. Vol. AC-17. N 4. P. 439–448.

15. Lauritzen S. L. Propagation of probabilities, means, and variances in mixed graphical association models // Journal of the American Statistical Association. 1992. V. 87. N 420. P. 1098–1108.

16. Lauritzen S. L. Graphical Models. Oxford Statistical Science Series. V. 17. Oxford: Clarendon Press. 1996.

17. Kukharencov B. G., Ponomarev D. I. Bayesian filtering of control signal of telerobotic manipulator with precise accelerometer // Проблемы машиностроения и автоматизации. 2011. № 1. С. 72–76.

18. Бернштейн Н. А. О построении движений. М.: Медгиз. 1947.

19. Бернштейн Н. А. Очерки по физиологии движений и физиологии активности. М.: Медицина. 1966.

УДК 004.3.047

Д. В. Андреев, д-р техн. наук, проф.,  
e-mail: andreev@ulstu.ru  
Ульяновский государственный  
технический университет

## Универсальный логический модуль для обработки многозначных данных

*Рассмотрены математическая модель, элементный базис и схемное решение нового аналогового модуля, универсального в классе всех  $n$ -арных функций  $k$ -значной логики. Представлен способ минимизации указанной модели при соответствующем ограничении класса воспроизводимых функций.*

**Ключевые слова:** многозначная логика, многоуровневые сигналы, универсальные модули

### Введение

При решении обширного круга задач обработки (преобразования) информации используются логические устройства, представляющие собой в общем случае  $(n, m)$ -полусетку, который каждому заданному набору значений  $n$  входных сигналов сопоставляет соответствующий набор значений  $m$  выходных сигналов. В условиях научно-технического прогресса структурная организация указанных устройств эволюционирует с постоянным повышением уровня элементной интеграции от индивидуальных (специализированных) структур к универсальным структурам широкого применения. Эффективность такой эволюции неразрывно связана с решением проблемы обеспечения универсальности (функциональной гибкости) логических устройств.

Усилиями многих исследователей создана теория конечных автоматов, служащая основой синтеза

произвольного логического устройства (модуля), входные и выходные сигналы которого являются двоичными. Алгоритм функционирования и внутренняя структура такого модуля описываются функциями двузначной логики (ДЛ). Видное место в среде конечных автоматов занимают так называемые перестраиваемые конечные автоматы или универсальные логические модули. Указанные модули за счет дополнительных к информационным настроечных входов имеют возможность настройки на реализацию любой из нескольких функций ДЛ.

Логические модули с двоичными входами и выходами широко применяют в системах автоматики, вычислительной техники и управления. Однако, если состояние объекта управления (контроля) характеризуется и регулируется соответственно  $n$  и  $m$  многозначными переменными, то для управления таким объектом необходим логический  $(n, m)$ -полусетку, выполняющий обработку многозначных данных. Очевидно, что алгоритм функционирования последнего должен определяться функциями многозначной логики.

В работе [1] рассмотрены схемные решения аналоговых модулей (селекторов напряжений), универсальных в классе всех  $k$ -значных  $n$ -арных логических функций. Недостатком этих модулей является зависимость структуры их базисных элементов от значности воспроизводимых функций. Селекторы напряжений, рассмотренные в работе [2], свободны от указанного недостатка, но воспроизводят более узкий класс логических функций.

Здесь предлагаются принципы построения универсального в классе всех  $n$ -арных функций  $k$ -значной логики аналогового модуля (селектора напряжений), структура базисных элементов которого не зависит от значности воспроизводимых функций. Отметим, что такие элементы наиболее подходят для аппаратного воплощения в однокристалльном исполнении.

**Новая каноническая форма представления  
многозначных логических функций  
и основанные на ней схемные решения**

Введем выражение следующего вида:

$$f(x_1, \dots, x_n) = \bigvee_{i=1}^N f(\alpha_{i1}, \dots, \alpha_{in}) \cdot \bigwedge_{j=1}^n \psi_{\alpha_{ij}}(x_j), \quad (1)$$

где  $N = k^n$ ;  $\wedge, \vee$  и  $\cdot$  — символы, обозначающие ДЛ-конъюнкцию, ДЛ-дизъюнкцию и алгебраическое умножение;  $f(\alpha_{i1}, \dots, \alpha_{in}) \in \{0, \dots, k-1\}$  — значение  $k$ -значной логической функции  $f(x_1, \dots, x_n)$  на  $i$ -м неповторяющемся наборе значений  $\alpha_{i1}, \dots, \alpha_{in} \in \{0, \dots, k-1\}$  ее аргументов  $x_1, \dots, x_n$ ;

$$\psi_{\alpha_{ij}}(x_j) = \begin{cases} 1 & \text{при } x_j = \alpha_{ij} \\ 0 & \text{при } x_j \neq \alpha_{ij} \end{cases} \quad (2)$$

— нормированная характеристическая функция;

$$\bigwedge_{j=1}^n \psi_{\alpha_{1j}}(x_j) + \dots + \bigwedge_{j=1}^n \psi_{\alpha_{Nj}}(x_j) = 1.$$

**Утверждение 1.** Выражение (1) задает любую  $k$ -значную логическую функцию  $n$  аргументов.

Для доказательства этого утверждения достаточно отметить следующее. ДЛ-дизъюнкция в выражении (1) содержит  $k^n$  произведений — по числу всех возможных неповторяющихся комбинаций из  $n$  значений  $n$   $k$ -значных переменных. Причем левый

сомножитель  $i$ -го ( $i = 1, k^n$  произведений определяет значение реализуемой функции на  $i$ -м наборе значений  $\alpha_{i1}, \dots, \alpha_{in}$  ее аргументов  $x_1, \dots, x_n$ , а правый сомножитель равен единице (нулю), если выполняется (не выполняется) условие  $x_1 = \alpha_{i1}, \dots, x_n = \alpha_{in}$ .

Таким образом, на любом из  $k^n$  наборов значений  $n$   $k$ -значных переменных выражение (1) будет давать значение реализуемой  $k$ -значной функции. Утверждение 1 доказано.

**Пример 1.** Для функции, заданной в таблице, в соответствии с выражением (1) получим

$$f(x_1, x_2) = 2(\psi_0(x_1) \wedge \psi_0(x_2)) \vee 2(\psi_0(x_1) \wedge \psi_1(x_2)) \vee 2(\psi_0(x_1) \wedge \psi_2(x_2)) \vee 2(\psi_2(x_1) \wedge \psi_1(x_2)) \vee 1(\psi_1(x_1) \wedge \psi_0(x_2)) \vee 1(\psi_2(x_1) \wedge \psi_0(x_2)) \vee 1(\psi_2(x_1) \wedge \psi_2(x_2)).$$

В последнем соотношении отсутствуют те произведения, в которых  $f(x_1, x_2) = 0$ .

**Трехзначная логическая функция  $f(x_1, x_2)$**

$x_1$	$x_2$	$f(x_1, x_2)$	$x_1$	$x_2$	$f(x_1, x_2)$	$x_1$	$x_2$	$f(x_1, x_2)$
0	0	2	1	0	1	2	0	1
0	1	2	1	1	0	2	1	2
0	2	2	1	2	0	2	2	1

Как уже отмечалось, выражение (1) содержит  $k^n$  произведений — по числу всех возможных наборов значений аргументов реализуемой  $k$ -значной логической функции  $f(x_1, \dots, x_n)$ . При этом количество возможных значений функции  $f(x_1, \dots, x_n)$  равно  $k < k^n$ . То есть среди указанных произведений найдутся произведения с одинаковыми левыми сомножителями. Пусть  $m$  — максимальное число таких сомножителей. Тогда выражение (1) можно записать в виде

$$f(x_1, \dots, x_n) = \bigvee_{i=1}^N f(\alpha_{i1}, \dots, \alpha_{in}) \cdot \bigwedge_{j=1}^n \psi_{\alpha_{ij}}(x_j) \vee \frac{N}{m} \bigvee_{i=1}^m f(\alpha_{(N+1)i}, \dots, \alpha_{(N+1)n}) \cdot \bigwedge_{j=1}^n \psi_{\alpha_{ij}}(x_j), \quad (3)$$

где  $N = k^n - m$ ;  $f(\alpha_{(N+1)1}, \dots, \alpha_{(N+1)n}) = f(\alpha_{(N+2)1}, \dots, \alpha_{(N+2)n}) = \dots = f(\alpha_{k^n 1}, \dots, \alpha_{k^n n})$ .

Наглядная трактовка соотношения (3) может быть выполнена с помощью продукционных правил "если..., то..., иначе..." следующим образом:

если  $x_1 = \alpha_{11}$  и ... и  $x_n = \alpha_{1n}$ ,  
то  $f(x_1, \dots, x_n) = f(\alpha_{11}, \dots, \alpha_{1n})$ ;  
.....  
если  $x_1 = \alpha_{N1}$  и ... и  $x_n = \alpha_{Nn}$ ,  
то  $f(x_1, \dots, x_n) = f(\alpha_{N1}, \dots, \alpha_{Nn})$ ;  
иначе  $f(x_1, \dots, x_n) = f(\alpha_{(N+1)1}, \dots, \alpha_{(N+1)n})$ .

Применив к соотношению (3) законы де Моргана, получим

$$f(x_1, \dots, x_n) = \bigvee_{i=1}^N f(\alpha_{i1}, \dots, \alpha_{in}) \cdot \bigwedge_{j=1}^n \psi_{\alpha_{ij}}(x_j) \vee \bigvee_{i=1}^m f(\alpha_{(N+1)i}, \dots, \alpha_{(N+1)n}) \cdot \bigwedge_{j=1}^n \bar{\psi}_{\alpha_{ij}}(x_j). \quad (4)$$

**Пример 2.** При  $n = 2, k = 3, m = 4, f(0, 0) = f(1, 1) = f(2, 1) = f(2, 2)$  согласно (4) имеем

$$f(x_1, x_2) = f(0, 1) \cdot (\psi_0(x_1) \wedge \psi_1(x_2)) \vee f(0, 2) \cdot (\psi_0(x_1) \wedge \psi_2(x_2)) \vee f(1, 0) \cdot (\psi_1(x_1) \wedge \psi_0(x_2)) \vee f(1, 2) \cdot (\psi_1(x_1) \wedge \psi_2(x_2)) \vee f(2, 0) \cdot (\psi_2(x_1) \wedge \psi_0(x_2)) \vee f(0, 0) \cdot ((\bar{\psi}_0(x_1) \vee \bar{\psi}_1(x_2)) \wedge (\bar{\psi}_0(x_1) \vee \bar{\psi}_2(x_2)) \wedge (\bar{\psi}_1(x_1) \vee \bar{\psi}_0(x_2)) \wedge (\bar{\psi}_1(x_1) \vee \bar{\psi}_2(x_2)) \wedge (\bar{\psi}_2(x_1) \vee \bar{\psi}_0(x_2))).$$

В качестве базисного элемента для аппаратной реализации выражения (4) удобно использовать аналоговое устройство, предложенное в работе [3]. Схема указанного устройства приведена на рис. 1 и содержит компараторы, логический элемент ИСКЛЮЧАЮЩЕЕ ИЛИ, замыкающий и размыкающий аналоговые ключи.



Устройство (рис. 1) воспроизводит операцию

$$Z = v_1 \cdot P(x) \vee v_2 \cdot \bar{P}(x),$$

где  $\vee$  — символ ДЛ-дизъюнкции;

$$P(x) = \begin{cases} 1 & \text{при } x \in y = [y_1, y_2]; \\ 0 & \text{при } x \notin y, \end{cases}$$

$x, v_1, v_2, y_1, y_2$  — аналоговые сигналы (напряжения).

Пусть  $y \in \{a^{(0)}, \dots, a^{(k-1)}\}$ , где  $a^{(r)} = [a_r, a_{r+1}]$  ( $r \in \{0, \dots, k-1\}$ );  $x, v_1, v_2 \in \{u_0, \dots, u_{k-1}\}$ , где  $u_r = 0,5(a_r, a_{r+1})$ ; действует система аналогий:  $u_0$  — логический "0",  $u_1$  — логическая "1", ...,  $u_{k-1}$  — логическая " $k-1$ ". Тогда, если  $x \in a^{(r)}$ , то это означает, что  $x = r$ . Таким образом, можно записать

$$P(x) = \psi_r(x) = \begin{cases} 1 & \text{при } x = r \\ 0 & \text{при } x \neq r, \end{cases}$$

где  $\psi_r(x)$  — нормированная характеристическая функция вида (2).

Приняв  $r = \alpha_1, v_1 = f(\alpha_1), v_2 = f(\alpha_2) = \dots = f(\alpha_k)$  ( $f(\alpha_1), \dots, f(\alpha_k)$  — значения  $k$ -значной логической функции  $f(x)$  при соответствующих значениях  $\alpha_1 \neq \dots \neq \alpha_k \in \{0, \dots, k-1\}$  ее аргумента  $x$ ), получим

$$Z = f(x) = f(\alpha_1) \cdot \psi_{\alpha_1}(x) \vee f(\alpha_2) \cdot \bar{\psi}_{\alpha_1}(x). \quad (5)$$

Нетрудно усмотреть, что выражение (4) при  $n = 1, m = k-1$  и выражение (5) совпадают. Следовательно, устройство (рис. 1) является универсальным логическим модулем (УЛМ) в классе  $k$ -значных логических функций  $f(x)$ , для которых  $m = k-1$ . Настройка указанного модуля осуществляется с помощью сигналов  $y_1, y_2$ , задающих порядок  $r = \alpha_1$  характеристической функции  $\psi_r(x)$  ( $k$ -значную переменную  $r$ ), и сигналов  $v_1, v_2$ , определяющих значения  $f(\alpha_1), f(\alpha_2)$  реализуемой функции  $f(x)$ .

На рис. 2 представлена схема модуля  $F(n)$ , построенная на основе выражения (4) в элементном базисе УЛМ  $F(1)$  по рис.1. Отметим, что у базисных УЛМ  $F(1)$  выходы ключей разведены. Модуль  $F(n)$  является универсальным в классе  $k$ -значных логических функций  $f(x_1, \dots, x_n)$ , каждая из которых принимает одинаковые значения на любых  $m$  наборах значений своих аргументов, а при  $m = k^n - 1$  указанный модуль универсален в классе всех  $k$ -значных логических функций  $f(x_1, \dots, x_n)$ . Настройка УЛМ  $F(n)$  на заданную функцию выполняется с помощью действующих на входах ( $ij$ -го модуля  $F(1)$ ) сигналов  $y_1, y_2$ , которые задают значение  $\alpha_{ij}$  переменной  $x_j$  ( $y_{ij} \rightarrow \alpha_{ij}$ ), и фиксации значений реализуемой функции на соответствующих переключаемых входах крайних слева модулей  $F(1)$ .

Аппаратурная сложность, схемная глубина и число настроечных входов УЛМ  $F(n)$  составляют  $L(F(n)) = 5n(k^n - m)$  компараторов, элементов ИСКЛЮЧАЮЩЕЕ ИЛИ, аналоговых ключей,  $H(F(n)) = 3$  и  $G(F(n)) = (k^n - m)(2n + 1) + 1$ . Приведенное значение глубины обусловлено тем, что максимальное время задержки распространения сигнала от входа к выходу в схеме (рис. 2) определяется выражением  $\tau = \tau_1 + \tau_2 + \tau_3$ , где  $\tau_1, \tau_2$  и  $\tau_3$  — время задержки компаратора, элемента ИСКЛЮЧАЮЩЕЕ ИЛИ и ключа.

Добиться существенного сокращения числа настроечных входов можно, если  $\alpha_{ij} = \text{const}$ . В этом случае  $G(F(n)) = k^n - m + 1$ . Однако указанное сокращение ведет к ограничению класса воспроизводимых функций. В нем останутся лишь функции  $f(x_1, \dots, x_n)$ , каждая из которых принимает одинаковые значения на одних и тех же  $m$  наборах значений своих аргументов.

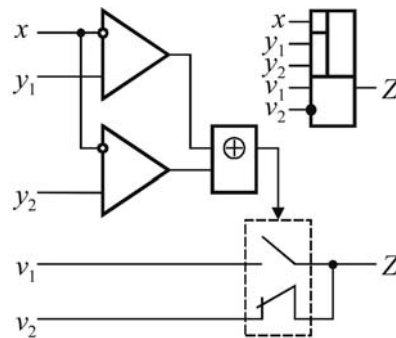


Рис. 1. Схема устройства с двумя регулируемыми порогами срабатывания

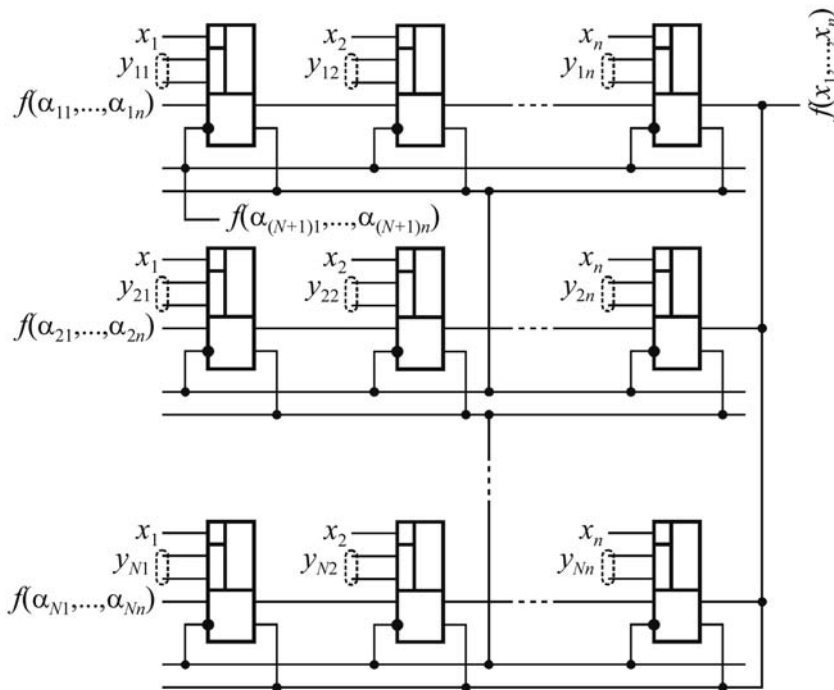


Рис. 2. Схема универсального логического модуля  $F(n)$

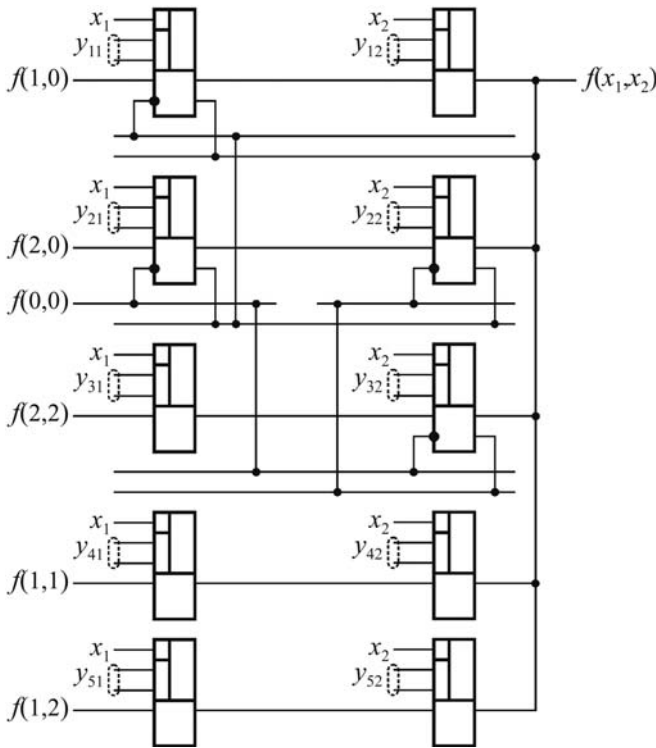


Рис. 3. Схемная реализация выражения (7)

Сложность модуля  $F(n)$ , воспроизводящего такой ограниченный класс  $k$ -значных логических функций, может быть уменьшена за счет упрощения выражения (4) с помощью законов двузначной логики и следующего утверждения.

**Утверждение 2.** Для функций  $\psi(x_j)$  и  $\bar{\psi}(x_j)$  из (4) справедливо равенство

$$\bar{\psi}_{\alpha_1}(x_j) \wedge \dots \wedge \bar{\psi}_{\alpha_p}(x_j) = \psi_{\alpha_{p+1}}(x_j) \vee \dots \vee \psi_{\alpha_k}(x_j), (6)$$

где  $\vee$  и  $\wedge$  — символы ДЛ-дизъюнкции и ДЛ-конъюнкции;  $\alpha_1 \neq \dots \neq \alpha_k \in \{0, \dots, k-1\}$ ;  $p \in \{1, \dots, k-1\}$ .

Справедливость равенства (6) вытекает с учетом (2) из очевидной справедливости условного предложения: если  $x_j \neq \alpha_1$  и  $\dots$  и  $x_j \neq \alpha_p$ , то  $x_j = \alpha_{p+1}$  или  $\dots$  или  $x_j = \alpha_k$ . Отметим, что в указанном предложении верно и обратное.

Смысл применения утверждения 2 заключается в уменьшении количества функций  $\bar{\psi}_{\alpha_{ij}}(x_j)$  при сохранении числа функций  $\psi_{\alpha_{ij}}(x_j)$ .

**Пример 3.** Пусть  $n = 2$ ,  $k = 3$ ,  $m = 4$  и

$$\begin{aligned} f(x_1, x_2) = & f(1, 0) \cdot (\psi_1(x_1) \wedge \psi_0(x_2)) \vee \\ & \vee f(2, 0) \cdot (\psi_2(x_1) \wedge \psi_0(x_2)) \vee f(2, 2) \cdot (\psi_2(x_1) \wedge \\ & \wedge \psi_2(x_2)) \vee f(1, 1) \cdot (\psi_1(x_1) \wedge \psi_1(x_2)) \vee \\ & \vee f(1, 2) \cdot (\psi_1(x_1) \wedge \psi_2(x_2)) \vee f(0, 0) \cdot ((\bar{\psi}_1(x_1) \vee \\ & \vee \bar{\psi}_0(x_2)) \wedge (\bar{\psi}_2(x_1) \vee \bar{\psi}_0(x_2)) \wedge (\bar{\psi}_2(x_1) \vee \bar{\psi}_2(x_2))) \wedge \\ & \wedge (\bar{\psi}_1(x_1) \vee \bar{\psi}_1(x_2)) \wedge (\bar{\psi}_1(x_1) \vee \bar{\psi}_2(x_2))). \end{aligned}$$

Тогда, с учетом (6) имеем

$$\begin{aligned} f(x_1, x_2) = & f(1, 0) \cdot (\psi_1(x_1) \wedge \psi_0(x_2)) \vee \\ & \vee f(2, 0) \cdot (\psi_2(x_1) \wedge \psi_0(x_2)) \vee f(2, 2) \cdot (\psi_2(x_1) \wedge \\ & \wedge \psi_2(x_2)) \vee f(1, 1) \cdot (\psi_1(x_1) \wedge \psi_1(x_2)) \vee \\ & \vee f(1, 2) \cdot (\psi_1(x_1) \wedge \psi_2(x_2)) \vee f(0, 0) \cdot ((\bar{\psi}_1(x_1) \wedge \\ & \wedge (\bar{\psi}_2(x_1) \vee (\bar{\psi}_0(x_2) \wedge \bar{\psi}_2(x_2))))). \end{aligned} (7)$$

Схема УЛМ  $F(n)$ , воспроизводящего выражение (7), изображена на рис.3. Здесь  $y_{11} \rightarrow 1$ ,  $y_{12} \rightarrow 0$ ,  $y_{21} \rightarrow 2$ ,  $y_{22} \rightarrow 0$ ,  $y_{31} \rightarrow 2$ ,  $y_{32} \rightarrow 2$ ,  $y_{41} \rightarrow 1$ ,  $y_{42} \rightarrow 1$ ,  $y_{51} \rightarrow 1$ ,  $y_{52} \rightarrow 2$  и  $L(F(n)) = 44 < 5n(k^n - m)$ ,  $H(F(n)) = 3$  и  $G(F(n)) = k^n - m + 1 = 6$ .

### Заключение

Предложенные автором новые формы представления многозначных логических функций удобно использовать для формализации алгоритма логического управления, описанного продукционными конструкциями "если..., то...", "если..., то..., иначе..." импликационного языка и учитывающего многозначный характер данных от источников информации и множество устойчивых состояний исполнительных механизмов системы управления. На основе математической модели произвольной продукционной конструкции "если..., то..., иначе...", оперирующей многозначными переменными, получено схемное решение аналогового модуля (селектора напряжений), структура базисных элементов которого не зависит от значности воспроизводимых функций.

Этот модуль является универсальным аналоговым средством логической обработки многоуровневых сигналов, поскольку может быть настроен на воспроизведение произвольных функций многозначной логики. Перспективность указанных средств обуславливается прежде всего тем, что они образуют элементный базис СБИС нового класса — аналоговых ПЛИС, которые способны снять присущую цифровым ПЛИС остроту проблемы значительного роста числа внутрикристалльных межсоединений и числа внешних выводов.

Необходимо отметить возможность обработки многозначных данных, представленных многоуровневыми сигналами, в стандартной системе "аналого-цифровой преобразователь  $\rightarrow$  цифровой процессор  $\rightarrow$  цифро-аналоговый преобразователь". Однако выполняемое в последней системе промежуточное преобразование многозначных данных в цифровую форму, отсутствующее в предложенном аналоговом модуле, снижает производительность обработки этих данных.

### Список литературы

1. **Специализированные** многозначные анализаторы / Под ред. М. А. Ракова. Киев: Наукова думка, 1977. 169 с.
2. **Андреев Д. В.** Реляторные комбинаторные сети со структурно-независимой настройкой и их комплексная оценка // Приборы и системы. Управление, контроль, диагностика. 2007. № 1. С. 18–22.
3. **Реляторный** идентификатор / Андреев Д. В., Волгин Л. И.: пат. 2117329 РФ. 1998.

Л. Н. Сандуляну, студент,  
 Московский физико-технический институт,  
 В. В. Стрижов, канд. физ.-мат. наук, науч. сотр.,  
 Вычислительный центр РАН,  
 e-mail: strijov@ccas.ru

## Выбор признаков в авторегрессионных задачах прогнозирования<sup>1</sup>

*Исследуется проблема выбора модели оптимальной сложности при авторегрессионном прогнозировании. Задача состоит в отыскании наименее обусловленного набора признаков, доставляющего при этом заданное значение функции ошибки. Для выбора этого набора используется модифицированный алгоритм последовательного добавления и удаления признаков. В работе предложен метод поиска оптимальной модели прогнозирования временных рядов. В вычислительном эксперименте проведено сравнение прогнозов рядов почасовых цен на электроэнергию.*

**Ключевые слова:** отбор признаков, мультиколлинеарность, шаговая регрессия, метод Белсли, прогнозирование временных рядов

### Введение

Решается задача восстановления линейной регрессии при наличии большого числа мультиколлинеарных признаков. Термин "мультиколлинеарность" введен Р. Фишером при рассмотрении линейных зависимостей между признаками [1]. Проблема состоит в том, что число признаков значительно превосходит число зависимых переменных, т. е. мы имеем дело с переопределенной матрицей. Для решения этой задачи необходимо исключить наиболее малоинформативные признаки. Для отбора признаков предлагается использовать модифицированный метод шаговой регрессии.

Ранее для решения подобных задач использовали следующие методы: метод наименьших углов LARS [2], метод Лассо [3], ступенчатая регрессия [4], последовательное добавление признаков с ортогонализацией FOS [5, 6], шаговая регрессия [4, 7, 8] и другие [9].

В работе предложен модифицированный метод шаговой регрессии. Шаговыми методами называются методы, заключающиеся в последовательном удалении или добавлении признаков согласно определенному критерию. Метод включает два основных этапа: последовательное добавление признаков и последовательное удаление признаков. Добавле-

ние признаков выполняется с помощью метода быстрого ортогонального спуска FOS [5, 6]. Данный метод последовательно добавляет признаки, которые максимально коррелируют с вектором регрессионных остатков. Удаление признаков в нашей работе выполняется с помощью метода Белсли [10]. Он позволяет выявить мультиколлинеарность признаков, используя сингулярное разложение матрицы признаков. Для нахождения алгоритма, который доставляет одновременно точную и устойчивую, в смысле минимизации числа признаков, модель, предложен новый метод останова этапов добавления и удаления признаков, а также останова всего алгоритма.

Основной причиной использования метода шаговой регрессии в качестве алгоритма выбора модели послужило то, что метод Белсли невозможно использовать при вырожденной авторегрессионной матрице, а метод ортогонального добавления признаков, в свою очередь, выбирает мультикоррелирующий набор признаков. Пошаговый алгоритм позволяет избежать этих недостатков и получить хорошо обусловленный набор признаков, позволяющий сделать достаточно точный прогноз.

Предложенный метод выбора модели проиллюстрирован задачей прогнозирования почасовых цен на электроэнергию на сутки вперед. Ранее эта задача решалась с помощью гребневой регрессии [11], метода наименьших углов, построения локальных регрессионных моделей [12, 13].

Для получения прогноза в работе предлагается использовать авторегрессионный алгоритм. В основе этого алгоритма лежит построение авторегрессионной матрицы, в которую построчно укладывается временной ряд, причем длина строки (ширина матрицы авторегрессии) равна периоду — 24 ч. Таким образом, каждый столбец содержит цену в некоторый час по всем суткам, и рассматривается как признак в задаче регрессии, а каждая строка является элементом выборки. Строятся 24 регрессионные модели — для прогнозирования цен на каждый час следующих суток. При использовании линейных моделей задача может быть решена методом наименьших квадратов.

В вычислительном эксперименте проведено сравнение предлагаемого алгоритма с базовым методом SSA [14].

Работа состоит из трех основных частей. Первая часть посвящена прогнозированию с помощью авторегрессионной матрицы. Во второй части описан выбор признаков при прогнозировании, здесь же можно найти описание метода Белсли. В последнем разделе приведены результаты вычислительного эксперимента, проведенного на основе данных почасовых цен на электроэнергию.

<sup>1</sup> Работа выполнена при поддержке РФФИ, проект 10-07-00422.

# 1. Задача прогнозирования с помощью авторегрессионной матрицы

Даны временной ряд  $\mathbf{s}_1 = \{x_i\}_{i=1}^T$ , будем называть его целевым рядом, и матрица признаков, столбцами которой являются временные ряды  $\mathbf{s}_2, \mathbf{s}_3, \dots, \mathbf{s}_p$ . Необходимо спрогнозировать следующие  $\tau$  значений ряда  $\mathbf{s}_1$ . Предполагается, что

- отсчеты  $x_i$  сделаны через равные промежутки времени;
- ряд  $\mathbf{s}$  имеет периодическую составляющую  $\tau$ ;
- ряд  $\mathbf{s}$  не имеет пропущенных значений;
- длина ряда  $\mathbf{s}$  кратна периоду  $\tau$ .

Для нахождения оптимальной модели предлагается построить алгоритм прогноза, позволяющий решать задачи прогнозирования периодических рядов. С помощью этого алгоритма строится прогноз по выбранному набору признаков. В нашей работе предлагается использовать метод авторегрессии.

Сначала опишем как строится прогноз методом авторегрессии без учета вспомогательных рядов  $\mathbf{s}_2, \mathbf{s}_3, \dots, \mathbf{s}_p$ , а затем обобщим метод на случай многомерного ряда (метод многомерной авторегрессии).

Пусть длина временного ряда  $T = m\tau$ . Построим авторегрессионную матрицу  $\mathbf{X}^*$ :

$$\mathbf{X}^* = \begin{pmatrix} x_1 & x_2 & \dots & x_{\tau-1} & x_{\tau} \\ \dots & \dots & \dots & \dots & \dots \\ x_{j\tau+1} & x_{j\tau+2} & \dots & x_{(j+1)\tau-1} & x_{(j+1)\tau} \\ \dots & \dots & \dots & \dots & \dots \\ x_{(m-2)\tau+1} & x_{(m-2)\tau+2} & \dots & x_{(m-1)\tau-1} & x_{(m-1)\tau} \\ x_{T-\tau+1} & x_{T-\tau+2} & \dots & x_{T-1} & x_T \end{pmatrix}$$

Введем обозначения:

$$\mathbf{X}^* = \begin{pmatrix} \mathbf{X} & \mathbf{y} \\ \mathbf{x}_m & x_T \end{pmatrix}$$

Необходимо построить линейную регрессию:

$$\mathbf{y} = \mathbf{X}\mathbf{w}, \tag{1}$$

где  $\mathbf{w}$  — вектор параметров. Тогда получим  $x_T = \langle \mathbf{x}_m, \mathbf{w} \rangle$ .

Требуется решить задачу минимизации евклидовой нормы вектора регрессионных остатков:

$$\|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 \rightarrow \min.$$

Вектор параметров  $\mathbf{w}$  отыскивается с помощью метода наименьших квадратов:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{y}).$$

Однако зависимость  $\mathbf{y} = \mathbf{f}(\mathbf{w}, \mathbf{X})$  может быть существенно нелинейной относительно свободных

переменных, и для построения линейной модели удовлетворительного качества необходимо расширить множество признаков с помощью функциональных преобразований исходных признаков.

В случае многомерного ряда при построении авторегрессионной матрицы необходимо учитывать временные ряды  $\mathbf{s}_2, \mathbf{s}_3, \dots, \mathbf{s}_p$ . Сначала строится авторегрессионная матрица  $\mathbf{X}_1$  для ряда  $\mathbf{s}_1$ . Для каждого следующего временного ряда  $\mathbf{s}_j$ , где  $j = 1, \dots, p$ , строится авторегрессионная матрица  $\mathbf{X}_j$ . Но для рядов  $\mathbf{s}_2, \mathbf{s}_3, \dots, \mathbf{s}_p$  не вводятся вектора  $\mathbf{y}$  в отличие от ряда  $\mathbf{s}_1$ . То есть для этих рядов авторегрессионная матрица будет содержать  $\tau$  столбцов, а не  $\tau - 1$ , как для целевого ряда. Присоединив авторегрессионные матрицы всех временных рядов, получим матрицу

$$\mathbf{X} = [\mathbf{X}_1 | \mathbf{X}_2, \dots, \mathbf{X}_p],$$

откуда получим

$$\mathbf{X} = \begin{pmatrix} x_1 & x_2 & \dots & x_{\tau-1} & x_{\tau} & s_2^1 & \dots & s_p^1 \\ x_{\tau+1} & x_{\tau+2} & \dots & x_{2\tau-1} & x_{2\tau} & s_2^2 & \dots & s_p^2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{T-\tau+1} & x_{T-\tau+2} & \dots & x_{T-1} & x_T & s_2^m & \dots & s_p^m \end{pmatrix},$$

где  $s_i^j$  —  $j$ -е значение ряда  $\mathbf{s}_j$ .

Опишем, в чем состоит задача выбора оптимальной модели. Задана выборка  $D = (\{x_i, y_i\})$ ,  $i \in \mathcal{J}$ , где множество свободных переменных — вектор  $\mathbf{x} = [x_1, \dots, x_j, \dots, x_n]$ , проиндексировано  $j \in \mathcal{J} = \{1, \dots, n\}$ . Задано разбиение множества индексов элементов выборки  $\mathcal{J} = \mathcal{L} \sqcup \mathcal{C}$ . Также задан класс регрессионных моделей  $f(\mathbf{w}, \mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$  — параметрических функций, линейных относительно параметров. Функция ошибки задана следующим образом:

$$S = \sum_{i \in \mathcal{X}} (y_i - f(\mathbf{w}, \mathbf{x}_i))^2, \tag{2}$$

где  $\mathcal{X} \subseteq \mathcal{J}$  — некоторое множество индексов. Требуется найти такое подмножество индексов  $\mathcal{A} \subseteq \mathcal{J}$ , которое бы доставляло минимум функции

$$\mathcal{A}^* = \arg \min_{\mathcal{A} \subseteq \mathcal{J}} S(f_{\mathcal{A}} | \mathbf{w}^*, \mathcal{D}_{\mathcal{C}}) \tag{3}$$

на множестве индексов  $\mathcal{C}$ . При этом параметры  $\mathbf{w}^*$  модели должны доставлять минимум функции

$$\mathbf{w}^* = \arg \min_{\mathbf{w} \in \mathcal{W}} S(\mathbf{w} | \mathcal{D}_{\mathcal{L}}, f_{\mathcal{A}}) \tag{4}$$

на множестве индексов  $\mathcal{L}$ . Здесь  $f_{\mathcal{A}}$  обозначает модель  $f$ , включающую только столбцы матрицы  $\mathbf{X}$  с индексами из множества  $\mathcal{A}$ , а обозначение вида  $S(\mathbf{w} | \mathcal{D})$  означает, что переменная  $\mathcal{D}$  фиксирована, а переменная  $\mathbf{w}$  изменяется.

## 2. Выбор признаков при прогнозировании

Предположим, что мы имеем данные о цене на электроэнергию за год. Тогда матрица  $\mathbf{X}$  имеет размерность  $(\tau - 1) \times (n - 1)$ , где  $\tau = 24$ , а  $n = 365$ . То есть  $\mathbf{X}$  является матрицей  $23 \times 364$ . Как видно, авторегрессионная матрица является переопределенной, поэтому необходим отбор признаков. Для отбора признаков предлагается модифицировать метод шаговой регрессии.

### 2.1. Процедура выбора оптимального набора признаков

Опишем два этапа алгоритма *Add* и *Del*. На первом этапе последовательно добавляются признаки согласно (4), доставляющие минимум  $S$  на обучающей выборке, заданной множеством индексов  $\mathcal{L}$ . На втором этапе происходит последовательное удаление признаков согласно методу Белсли. Пусть на  $k$ -м шаге алгоритма имеется активный набор признаков  $\mathcal{A}_k \in \mathcal{J}$ . На нулевом шаге  $\mathcal{A}_0$  пуст. Опишем этапы *Add* и *Del*.

*Этап Add.* Находим признак, доставляющий минимум  $S$  на обучающей выборке:

$$j^* = \arg \min_{j \in \mathcal{J} \setminus \mathcal{A}_{k-1}} S(\mathbf{w} | \mathcal{D}_{\mathcal{L}}, f_{\mathcal{A}_{k-1} \cup \{j\}}).$$

Затем добавляем новый признак  $j^*$  к текущему активному набору:

$$\mathcal{A}_k = \mathcal{A}_{k-1} \cup \{j^*\},$$

и повторяем эту процедуру до тех пор, пока  $S(f_{\mathcal{A}_k} | \mathbf{w}^*, \mathcal{D})$  превосходит свое минимальное значение на данном этапе не более чем на некоторое заданное значение  $\Delta S_1$ .

*Этап Del.* Находим индексы обусловленности и долевые коэффициенты для текущего набора признаков  $\mathcal{A}_{k-1}$  согласно методу Белсли, описание которого приведено ниже. Далее находим максимальный индекс обусловленности

$$i^* = \arg \max_{i \in \mathcal{A}_{k-1}} \eta_i. \quad (5)$$

Затем ищем максимальный долевого коэффициент, соответствующий найденному индексу обусловленности  $\eta_{i^*}$ ,

$$j^* = \arg \max_{i \in \mathcal{A}_{k-1}} q_{i^*j}. \quad (6)$$

Удаляем  $j^*$ -й признак из текущего набора

$$\mathcal{A}_k = \mathcal{A}_{k-1} \setminus j^*,$$

и повторяем эту процедуру до тех пор пока  $S(f_{\mathcal{A}_k} | \mathbf{w}^*, \mathcal{D})$  превосходит свое минимальное значение на данном этапе не более чем на некоторое заданное значение  $\Delta S_2$ .

Повторение этапов *Add* и *Del* осуществляется до тех пор, пока значение  $S(f_{\mathcal{A}_k} | \mathbf{w}^*, \mathcal{D})$  не стабилизируется.

### 2.2. Метод Белсли для удаления признаков

Рассмотрим матрицу признаков  $\mathbf{X}$ . Она имеет размерность  $m \times n$ . Выполним ее сингулярное разложение:

$$\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T,$$

где  $\mathbf{U}$ ,  $\mathbf{V}$  — ортогональные матрицы размерностью соответственно  $m \times m$  и  $n \times n$ ;  $\mathbf{\Lambda}$  — диагональная матрица с элементами (сингулярными числами) на диагонали, такими что

$$\lambda_1 > \lambda_2 > \dots > \lambda_r,$$

где  $r$  — ранг матрицы  $\mathbf{X}$ . Заметим, что в нашем случае  $r = n$ . Это связано с тем, что в алгоритме шагового выбора на каждом шаге мы имеем мультиколлинеарный, но невырожденный набор признаков.

Столбцы матрицы  $\mathbf{V}$  являются собственными векторами, а квадраты сингулярных чисел — собственными значениями корреляционной матрицы  $\mathbf{X}^T\mathbf{X}$ :

$$\mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{\Lambda}^T\mathbf{U}^T\mathbf{U}\mathbf{\Lambda}\mathbf{V}^T = \mathbf{V}\mathbf{\Lambda}^2\mathbf{V}^T,$$

$$\mathbf{X}^T\mathbf{X}\mathbf{V} = \mathbf{V}\mathbf{\Lambda}^2.$$

Отношение максимального сингулярного числа к  $j$ -му сингулярному числу назовем индексом обусловленности с номером  $j$ :

$$\eta_j = \frac{\lambda_{\max}}{\lambda_j}.$$

Если матрица  $\mathbf{X}$  неполноранговая, то значительная часть индексов обусловленности неопределено. Однако в нашем случае, как упоминалось выше, матрица признаков  $\mathbf{X}$  является матрицей полного ранга.

Используя сингулярное разложение, дисперсию параметров, найденных методом наименьших квадратов  $\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ , где  $\mathbf{w}$  — вектор параметров модели, можно записать как

$$\mathbf{var}(\mathbf{w}) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} = \sigma^2(\mathbf{V}^T)^{-1}\mathbf{\Lambda}^{-2}\mathbf{V}^{-1} = \sigma^2\mathbf{V}\mathbf{D}^{-2}\mathbf{V}^T,$$

где  $\sigma^2$  — это дисперсия регрессионных остатков. Таким образом, дисперсия  $j$ -го регрессионного коэффициента — это  $j$ -й диагональный элемент матрицы  $\mathbf{var}(\mathbf{w})$ .

Для обнаружения мультиколлинеарности признаков построим таблицу, в которой каждому индексу обусловленности  $\eta_j$  соответствуют значения  $q_{ij}$  — долевые коэффициенты. Сумма долевых коэффициентов по индексу  $j$  равна единице,

$$\sigma^{-2}\mathbf{var}(w_j) = \sum_{i=1}^n \frac{v_{ij}^2}{\lambda_j^2} = (q_{i1} + q_{i2} + \dots + q_{in}) \sum_{j=1}^n \frac{v_{ij}^2}{\lambda_j^2},$$

где  $q_{ij}$  — отношение соответствующего слагаемого в разложении вектора  $\sigma^{-2}\mathbf{var}(w_j)$  ко всей сумме, а  $\mathbf{V} = (v_{ij})$ .

Таблица 1

Разложение  $\text{var}(w_j)$ 

Индекс обусловленности	$\text{var}(w_1)$	$\text{var}(w_2)$	...	$\text{var}(w_n)$
$\eta_1$	$q_{11}$	$q_{21}$	...	$q_{n1}$
$\eta_2$	$q_{12}$	$q_{22}$	...	$q_{n2}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\eta_n$	$q_{1n}$	$q_{2n}$	...	$q_{nn}$

Мультиколлинеарность определяется с помощью табл. 1: большие значения  $\eta_j$  означают, что возможно есть зависимость между признаками. Большие значения  $q_{ij}$  в соответствующих строках относятся к признакам, между которыми эта зависимость существует. Малые значения  $\eta_j$  также исследуются: между признаками, соответствующими большим значениям  $q_{ij}$ , зависимости не существует. Для нахождения мультиколлинеарных признаков решаются задачи (5) и (6).

### 3. Описание базового алгоритма прогноза

В вычислительном эксперименте проведено сравнение предлагаемого алгоритма с базовым методом SSA, русское название метода — "Гусеница". Приведем его краткое описание.

Для последующего разложения ряда  $\mathbf{s}_1 = \{x_i\}_{i=1}^T$  по главным компонентам преобразуем ряд в траекторную матрицу (матрицу Ганкеля)  $\mathbf{Y}$ , которую строим следующим образом:

$$\mathbf{Y} = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ x_2 & x_3 & \dots & x_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_m & x_{m+1} & \dots & x_T \end{pmatrix}, \quad (7)$$

где величина  $n = T - m + 1$  называется временем жизни гусеницы. Матрицу (7) будем называть нецентрированной траекторной матрицей, порожденной гусеницей длины  $n$ . В дальнейшем анализ главных компонент можно проводить как по централизованной, так и по нецентрированной выборкам. Для упрощения выкладок рассмотрим простейший нецентрированный вариант.

Построим ковариационную матрицу следующим образом:

$$\mathbf{C} = \frac{1}{n} \mathbf{Y}^T \mathbf{Y}.$$

Так как матрица Ганкеля  $\mathbf{Y}$  невырождена, то матрица  $\mathbf{C}$  является полноранговой, т. е. ее ранг равен  $n$ . Выполним ее сингулярное разложение:

$$\mathbf{C} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T,$$

где  $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$  — диагональная матрица собственных чисел;  $\mathbf{V} = [v^1, \dots, v^n]$  — ортогональная матрица собственных векторов-столбцов. При этом будем предполагать, что собственные векторы упорядочены по убыванию соответствующих собственных чисел, т. е.  $\lambda_1 > \lambda_2 > \dots > \lambda_n$ .

Перейдем к прогнозированию временных рядов методом гусеницы. Рассмотрим систему уравнений:

$$\begin{cases} \sum_{j=1}^n h_j v_1^j = x_{m+1}, \\ \sum_{j=1}^n h_j v_{n-1}^j = x_T. \end{cases} \quad (8)$$

Пусть  $\mathbf{h}^* = [h_1^*, \dots, h_n^*]$  — решение системы (8), тогда для продолжения ряда получим

$$x_{T+1} = \sum_{j=1}^n h_j^* v_n^j.$$

Выбираем  $r$  главных компонент из матрицы  $\mathbf{V}$ :

$$\mathbf{V}^* = \begin{pmatrix} v_1^{i_1} & v_1^{i_2} & \dots & v_1^{i_r} \\ v_2^{i_1} & v_2^{i_2} & \dots & v_2^{i_r} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n-1}^{i_1} & v_{n-1}^{i_2} & \dots & v_{n-1}^{i_r} \end{pmatrix},$$

где  $i_1, \dots, i_r$  — номера выбранных главных компонент. Введем следующие обозначения  $\mathbf{v} = (v_n^{i_1}, v_n^{i_2}, \dots, v_n^{i_r})$ ,  $\mathbf{q} = (x_{m+1}, \dots, x_n)^T$  и  $\tilde{\mathbf{h}} = (h_{i_1}, \dots, h_{i_r})^T$ .

В этих обозначениях система (8) запишется как

$$\mathbf{V}^* \tilde{\mathbf{h}} = \mathbf{q}. \quad (9)$$

Учитывая (9), можно записать для прогнозируемого значения  $x_{T+1}$  следующую формулу:

$$x_{T+1} = \mathbf{v}((\mathbf{V}^*)^T \mathbf{V}^*)^{-1} (\mathbf{V}^*)^T \mathbf{q}.$$

Таким образом мы построили прогноз — следующее по времени значение  $x_{T+1}$  временного ряда  $\mathbf{s}_1 = \{x_i\}$ .

### 4. Вычислительный эксперимент

С целью сравнить предложенный в настоящей работе подход с базовым алгоритмом SSA была проведена серия экспериментов по краткосрочному прогнозированию временных рядов. Данные — почасовые цены на электроэнергию в Германии за период с 1 января 2003 г. по 10 июля 2009 г., опубликованы в работе [15]. Мы прогнозируем почасовые цены на ближайшие сутки по предыдущей истории.

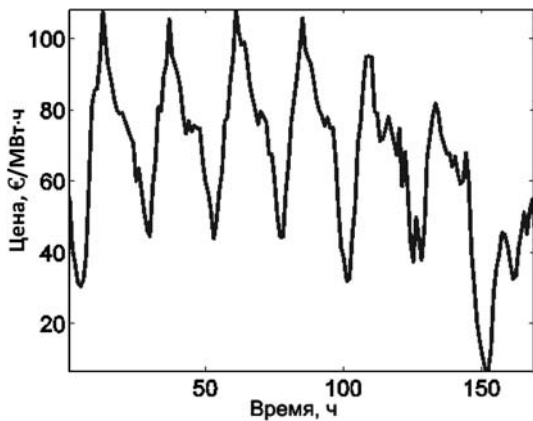


Рис. 1. Диаграмма зависимости цен от времени суток и дня недели

На рис. 1 приведен график, отражающий колебание цен на электроэнергию в течение недели. По оси абсцисс отложено время в часах от начала недели, а по оси ординат — цена в евро за мегаватт-час. На рисунке виден спад цен в ночное время и в выходные дни. Диаграмма на рис. 2 (см. третью сторону обложки) показывает изменение цен за четыре недели в зависимости от времени суток. На рис. 3 (см. третью сторону обложки) приведены два графика. Первый отражает зависимость реальных данных о ценах от времени, а второй — зависимость спрогнозированных, с помощью предложенного в работе алгоритма, на сутки цен от времени. На рис. 4 (см. третью сторону обложки) показано, как изменяется квадратичная ошибка  $S$ , определенная по (2), на тестовой выборке  $C$ , от итерации к итерации.

Сравним результаты работы предложенного алгоритма с базовым алгоритмом SSA. Результаты экспериментов показаны в табл. 2. Для каждого алгоритма вычислялись ошибки MSE (среднее квадратичное отклонение) и MAPE (средняя абсолютная процентная ошибка):

$$MSE = \frac{1}{\tau} \sum_{i=T+1}^{T+\tau} (\tilde{x}_i - x_i)^2,$$

$$MAPE = \frac{1}{\tau} \sum_{i=T+1}^{T+\tau} 100 \frac{|\tilde{x}_i - x_i|}{|x_i|},$$

Таблица 2

Результаты работы алгоритмов

Алгоритмы выбора признаков	MSE	MAPE в рабочие дни	MAPE в выходные дни	AIC	BIC	Число признаков
Предложенный алгоритм	8,18	6,17	10,33	96,44	123,53	23
Метод SSA	13,25	16,16	29,01	130,74	149,59	16

где  $\tilde{x}_i$  — спрогнозированное значение целевого ряда в точке  $i$ ,  $x_i$  — фактическое значение этого ряда в точке  $i$ . Также сравнивались значения информационных критериев Акаике (AIC) и Байеса (BIC):

$$AIC = \tau \left( \ln \frac{S}{\tau} \right) + 2|A|, \quad BIC = \tau \left( \ln \frac{S}{\tau} \right) + |A| \ln \tau,$$

где  $S$  — средняя квадратичная ошибка, вычисленная по набору активных признаков  $A$ . В табл. 2 также указано число  $n$  признаков, входящих в модель.

## Заключение

Предложенный подход значительно эффективнее базового алгоритма, особенно в случае, когда выборка содержит большое число мультиколлинеарных признаков. Вычислительный эксперимент показал, что увеличение числа признаков, из которых проводится выбор, позволяет добиться улучшения качества модели. Однако при этом требуется введение дополнительных условий, позволяющих избежать появления мультиколлинеарных признаков. Предлагаемый алгоритм включает процедуру анализа мультиколлинеарности и позволяет выбирать хорошо обусловленные наборы признаков.

## Список литературы

1. **Frisch R.** Statistical confluence analysis by means of complete regression systems. Universitetets Okonomiske Institute, 1934.
2. **Efron B., Hastie T., Johnstone I., Tibshirani R.** Least angle regression // The Annals of Statistics. 2004. Vol. 32, no. 3. P. 407—499.
3. **Tibshirani R.** Regression shrinkage and selection via the lasso // Journal of the Royal Statistical Society. 1996. Vol. 32, no. 1. P. 267—288.
4. **Draper N. R., Smith H.** Applied regression analysis. New York: John Wiley and Sons, 1998.
5. **Chen Y. W., Billings C. A., Luo W.** Orthogonal least squares methods and their application to non-linear system identification // International Journal of Control. 1989. Vol. 2, no. 50. P. 873—896.
6. **Chen S., Cowan C. F. N., Grant P. M.** Orthogonal least squares learning algorithm for radial basis function network // Transaction on neural network. 1991. Vol. 2, no. 2. P. 302—309.
7. **Efroymson M. A.** Multiple regression analysis. New York: Ralston, Wiley, 1960.
8. **Rawlings J. O., Pantula S. G., Dickey D. A.** Applied regression analysis: a research tool. New York: Springer-Verlag, 1998.
9. **Стризов В. В., Крымова Е. А.** Выбор моделей в линейном регрессионном анализе // Информационные технологии. 2011. № 10. С. 21—26.
10. **Belsley D. A.** Conditioning Diagnostics: Collinearity and weak data in regression. New York: John Wiley and Sons, 1991.
11. **Tarantola A.** Inverse problem theory. Philadelphia: SIAM, 2005.
12. **Johnstone I., Tibshirani R., Efron B., Hastie T.** Least angle regression. New York: Springer, 2004.
13. **McNames J.** Innovations in local modeling for time series prediction, 1999.
14. **Солнцев В. Н., Данилов Д. Л., Жигляевский А. А.** Главные компоненты временных рядов: Метод "Гусеница". С.-Пб.: Изд-во СПбГУ, 1997.
15. **Strijov V.** Lectures on Data Mining and Business Analytics, Time Series Set. URL: <https://dmba.svn.sourceforge.net/svnroot/dmba/Data/GermanSpotPrice.csv>, 2011.

**В. А. Чепурко**, канд. физ.-мат. наук, доц.,  
**С. В. Чепурко**, аспирант,  
 Обнинский институт атомной энергетики  
 НИЯУ МИФИ,  
 e-mail: chepurko@iate.obninsk.ru

## Непараметрическая оценка коэффициента деградации геометрических процессов

*Рассмотрено построение точечной и интервальной непараметрической оценки знаменателя геометрического процесса  $\gamma$ . На основании интервальной оценки возможна проверка гипотезы о том, что наблюдаемый процесс восстановления является геометрическим. Исследуемая модель геометрического процесса линеаризуется логарифмированием. Предложенный метод основан на непараметрической оценке Тейла углового коэффициента простой линейной регрессии.*

**Ключевые слова:** геометрический процесс, процесс восстановления, знаменатель процесса, угловой коэффициент, непараметрическая оценка Тейла, статистика Кендалла

### Введение

В процессе работы техническое оборудование, как известно, проходит несколько стадий эксплуатации. В зависимости от стадии эксплуатации изменяются показатели надежности оборудования, а также способы их расчета. До недавних пор наибольшее внимание уделялось периоду нормальной эксплуатации, на котором параметр (интенсивность) потока отказов является приблизительно постоянной величиной. Но при расчете показателей надежности необходимо учитывать и два других периода: приработки и интенсивного изнашивания, когда параметр потока отказов в одном случае убывает, а в другом возрастает со временем. В общем случае могут иметь место и другие, более сложные зависимости от времени. Одной из достаточно простых моделей неоднородных (во времени) процессов восстановления является модель геометрических процессов. Эти процессы появились сравнительно недавно [1, 2] и пока еще не приобрели такой популярности, как обычные процессы восстановления. Это, в первую очередь, вызвано тем, что до сих пор не исследованы многие теоретические вопросы, связанные со свойствами этих процессов. В частности, до сих пор не существует критерия, на основании которого можно принять гипотезу о применимости модели геометрического процесса. Построению доверительного интервала и, следовательно, критерия проверки гипотезы согласия с моделью геометрических процессов посвящена данная статья.

### Геометрические процессы

Название процесса напрямую связано с понятием геометрической прогрессии. Геометрические процессы являются обобщением процессов восстановления. В отличие от обычного процесса восстановления, который моделирует идеальный ремонт, геометрические процессы могут быть полезны для моделирования, например, несовершенного ремонта, когда получающиеся длительности циклов процесса неодинаково распределены. Тем не менее по сравнению с другими неоднородными процессами модель достаточно бедна, поскольку длительности цикла "регулируются" одним и тем же параметром. Геометрические процессы (в контексте теории надежности) были определены в работах [2–5].

**Определение.** Случайная величина  $\xi$  равна случайной величине  $\eta$  по распределению, если равны их функции распределения —  $F_\xi(x) = F_\eta(x)$ . Равенство по распределению обозначается следующим образом:

$$\xi \stackrel{d}{=} \eta.$$

**Определение.** Последовательность неотрицательных (типа времени жизни) независимых случайных величин  $\{\Delta_k; k = 1, 2, \dots\}$  образует геометрический процесс, если выполняется следующее равенство по распределению:

$$\Delta_{k+1} \stackrel{d}{=} \gamma \Delta_k, \quad k = 1, 2, \dots, \quad (1)$$

где  $\gamma > 0$  — вещественная константа, называемая знаменателем геометрического процесса. При значениях, меньших 1, будем называть знаменатель  $\gamma$  коэффициентом деградации.

### Исходная информация

Будем предполагать, что под наблюдением находится  $m$  однотипных мгновенно восстанавливаемых объектов, каждый из которых имеет свою реализацию наработок между отказами. При этом реализации могут иметь, вообще говоря, различные объемы:  $n_1, n_2, \dots, n_m$ . При этом

$$n_1 + n_2 + \dots + n_m = n. \quad (2)$$

Кроме этого, предположим, что в общем случае в таблице данных возможны пропуски. К примеру, у 3-го объекта отсутствует информация о 4-й и 5-й наработках, а далее 6-я и т. д. присутствуют.

Транспонируем таблицу данных, т. е. сгруппируем информацию согласно номеру наработки. В этом случае массив  $\Delta_{1,1}, \dots, \Delta_{m,1}$  будет массивом реализаций наработки до первого отказа —  $\Delta_1$ ;  $\Delta_{1,2}, \dots, \Delta_{m,2}$  — наработки между первым и вторым отказами —  $\Delta_2$  и т. д. Пусть самая последняя наблюдаемая наработка будет иметь номер  $l$ . Таким



образом исходную информацию представим в следующем виде:

$\Delta_{1,1}, \dots, \Delta_{m_1,1}$  — первые наработки (объем  $m_1$ );

$\Delta_{1,2}, \dots, \Delta_{m_2,2}$  — вторые наработки (объем  $m_2$ ), ...,

$\Delta_{1,l}, \dots, \Delta_{m_l,l}$  —  $l$ -е наработки (объем  $m_l$ ).

При этом так же как и в формуле (2)

$$m_1 + m_2 + \dots + m_l = n.$$

В силу соотношения (1) будет справедливо следующее равенство по распределению:

$$\Delta_{k+1} \stackrel{d}{=} \gamma^k \Delta_1, \quad k = 0, 1, \dots, \text{ или}$$

$$\ln \Delta_{k+1} \stackrel{d}{=} k \ln \gamma + \ln \Delta_1, \quad k = 0, 1, \dots \quad (3)$$

Уравнение (3) является моделью простой линейной регрессии зависимой переменной  $\ln \Delta$  от неслучайной переменной  $k$ . При этом  $\ln \Delta_1^{(i,j)}$  будет играть роль случайного шума, присутствующего для  $i$ -го наблюдения  $j$ -й наработки между отказами:

$$\ln \Delta_{i,k} = (k-1) \ln \gamma + \ln \Delta_1^{(i,k)}, \quad i = 1, \dots, m_k, \quad k = 1, \dots, l.$$

Будем предполагать, что случайный шум представляет собой массив независимых одинаково распределенных случайных величин  $\ln \Delta_1^{(i,j)}$ , распределенных так же, как и  $\ln \Delta_1$ . Обозначив

$$Y_{i,j} = \ln \Delta_{i,j}, \quad \beta = \ln \gamma \quad \text{и} \quad \varepsilon_{ij} = \ln \Delta_1^{(i,j)}, \quad (4)$$

получим уравнение

$$Y_{i,k} = \beta(k-1) + \varepsilon_{i,k}, \quad i = 1, \dots, m_k, \quad k = 1, \dots, l.$$

Таким образом, задача оценивания неизвестного параметра  $\gamma$  сводится к оцениванию углового коэффициента  $\beta$  простой линейной регрессии  $Y$  на  $k$ . При этом для каждого  $k$  проводятся повторные наблюдения величины  $Y_k$  объема  $m_k$  —  $Y_{i,k}$ .

### Непараметрическая оценка коэффициента наклона

В настоящее время для точечного оценивания коэффициентов регрессии широко используется метод наименьших квадратов (МНК). При этом для более тонких и глубоких выводов при решении таких задач, как построение доверительных интервалов, проверка адекватности модели приходится предполагать, что ошибки  $\varepsilon_{ij}$  независимы и одинаково распределены по нормальному закону. На этой основе за многие годы выросла обширная и развитая система статистической обработки регрессионных и факторных экспериментов. На практике эти методы приходится применять и в тех случаях, когда результаты наблюдения имеют другой, отличный от нормального закон распределения.

В этом случае формулируемые выводы будут приближенными. Было проведено немалое количество исследований, чтобы выяснить, как те или иные отступления от основных предположений о случайных ошибках влияют на конечные результаты. Выяснилось, что иногда нарушения, кажущиеся незначительными, могут существенно исказить выводы: привести к смещению оценок, доверительных границ и коэффициентов доверия.

При явном отличии распределения от гауссовского проверка корректности многих выводов, полученных в рамках нормальной модели, перерастает в сложную проблему. В таких случаях разумно вообще отказаться от стандартной гауссовской модели и применять иные, *непараметрические методы*. Непараметрическими называют такие методы, которые не предназначены специально для какого-нибудь параметрического семейства распределений и не используют его свойства.

Возьмем за основу построения наших оценок непараметрический метод точечного и интервального оценивания углового коэффициента  $\beta$ , принадлежащий Тейлу [6—8] и обобщенный Сенном [9] на случай совпадения отдельных значений независимой переменной. Кроме этого, рассмотрим задачу проверки нулевой гипотезы:

$$H_0 : \gamma = \gamma_0, \quad (5)$$

решение которой также предложено Тейлом.

### Алгоритм нахождения точечной и интервальной оценки коэффициента деградации и проверки гипотезы о его значимости

Для нахождения точечной оценки углового коэффициента  $\beta$  необходимо проделать следующее [10, 11]. Коэффициент  $\gamma$  будем параллельно определять согласно уравнениям связи (4).

1. Положить

$$N = \sum_{i < j} m_i m_j = C_n^2 - \sum_{i=1}^l C_{m_i}^2.$$

В случае, если  $m_i = 1$  (наблюдается один объект)  $N = C_n^2$  — числу возможных пар экспериментальных точек  $(i, Y_{i+1})$ .

2. Определить  $N$  значений углового коэффициента  $\beta$ :

$$\beta_{i,j}^{(k,l)} = \frac{Y_{i,k} - Y_{j,l}}{k-l}, \quad k > l.$$

Возвращаясь к исходным переменным, можно сразу (без нахождения  $\beta_{i,j}^{(k,l)}$ ) определить  $N$  значений коэффициента  $\gamma$ :

$$\gamma_{i,j}^{(k,l)} = \left( \frac{\Delta_{i,k}}{\Delta_{j,l}} \right)^{\frac{1}{k-l}}, \quad k > l. \quad (6)$$

Если наблюдается один объект, то рассчитываются  $\beta^{(k, l)}$  и  $\gamma^{(k, l)}$ .

3. Упорядочить массив  $\beta_{i,j}^{(k, l)}$ , т. е. построить вариационный ряд

$$\beta_{(1)} \leq \beta_{(2)} \leq \dots \leq \beta_{(N)}.$$

Аналогично упорядочивается массив значений  $\gamma_{i,j}^{(k, l)}$ :

$$\gamma_{(1)} \leq \gamma_{(2)} \leq \dots \leq \gamma_{(N)}. \quad (7)$$

4. В качестве оценки углового коэффициента взять выборочную медиану

$$\hat{\beta} = \text{med}\{\beta_{(1)}, \dots, \beta_{(N)}\} = \begin{cases} \beta_{(k+1)}, & N = 2k + 1; \\ \frac{\beta_{(k)} + \beta_{(k+1)}}{2}, & N = 2k. \end{cases}$$

Соответственно оценивается коэффициент  $\gamma$ :

$$\hat{\gamma} = \text{med}\{\gamma_{(1)}, \dots, \gamma_{(N)}\} = \begin{cases} \gamma_{(k+1)}, & N = 2k + 1; \\ \frac{\gamma_{(k)} + \gamma_{(k+1)}}{2}, & N = 2k. \end{cases} \quad (8)$$

5. Для построения доверительного интервала для коэффициентов  $\beta$  и  $\gamma$  надежности  $1 - \alpha$  положить

$$M_1 = \left[ \frac{N - K_{1-\alpha/2}}{2} \right]; M_2 = \left[ \frac{N + K_{1-\alpha/2}}{2} \right] + 1, \quad (9)$$

где  $[x]$  — целая часть числа  $x$ ;  $K_{1-\alpha/2}$  — квантиль распределения  $\tau$ -статистики Кендалла. Если объем наблюдений  $n$  достаточно велик, то справедлива следующая аппроксимация квантилем нормально-го закона  $u_{1-\alpha/2}$ :

$$K_{1-\alpha/2} = \sqrt{D(K)} u_{1-\alpha/2};$$

$$D(K) = \frac{\left\{ n(n-1)(2n+5) - \sum_{i=1}^l m_i(m_i-1)(2m_i+5) - \sum_{j=1}^h u_j(u_j-1)(2u_j+5) \right\}}{18} + \frac{\left\{ \sum_{i=1}^l m_i(m_i-1)(m_i-2) \right\} \cdot \left\{ \sum_{j=1}^h u_j(u_j-1)(u_j-2) \right\}}{9n(n-1)(n-2)} + \frac{\left\{ \sum_{i=1}^l m_i(m_i-1) \right\} \cdot \left\{ \sum_{j=1}^h u_j(u_j-1) \right\}}{2n(n-1)},$$

где  $l$  — номер максимальной наработки между отказами (число групп совпадающих наблюдений абсциссы  $X$ );  $m_i$  — объем  $i$ -й группы наблюдений  $X$ ;  $h$  — число групп совпадающих наблюдений  $Y$  (или  $\Delta$ );  $u_j$  — объем  $j$ -й группы наблюдений  $Y$  (или  $\Delta$ ). Причем несвязанные наблюдения рассматриваются как группа совпадающих наблюдений объема 1.

Если среди  $\Delta_{i,j}$  нет одинаковых (связанных) значений, то дисперсия статистики Кендалла упрощается до следующего выражения:

$$D(K) = \frac{n(n-1)(2n+5) - \sum_{i=1}^l m_i(m_i-1)(2m_i+5)}{18}. \quad (10)$$

Если кроме этого наблюдается один объект, то

$$D(K) = \frac{n(n-1)(2n+5)}{18}.$$

6. В качестве доверительного интервала для иско- мых коэффициентов взять промежутки  $(\beta_{(M_1)}, \beta_{(M_2)})$  и  $(\gamma_{(M_1)}, \gamma_{(M_2)})$  соответственно.

7. С уровнем значимости  $\alpha$  принять гипотезу (5) в случае, если  $\gamma_0 \in (\gamma_{(M_1)}, \gamma_{(M_2)})$ , и отвергнуть ее в противном случае.

### Параметрический аналог оценки коэффициентов регрессии

Для точечной оценки неизвестного углового ко- эффициента широко используется следующая оценка по методу наименьших квадратов (МНК) [12–14]:

$$\hat{\beta} = \frac{\sum_{k=1}^l m_k (X_k - \bar{X})(\bar{Y}_{\bullet k} - \bar{Y})}{\sum_{k=1}^l m_k (X_k - \bar{X})^2}, \quad (11)$$

где  $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i = \frac{1}{n} \sum_{j=1}^l m_j X_j$  — среднее арифметиче-

ское  $X_j$ ;  $\bar{Y}_{\bullet k} = \frac{1}{m_k} \sum_{i=1}^{m_k} Y_{i,k}$  — средняя арифметическая

$k$ -я наработка между отказами;  $\bar{Y} = \frac{1}{n} \sum_{k=1}^l m_k \bar{Y}_{\bullet k} =$

$= \frac{1}{n} \sum_{k=1}^l \sum_{i=1}^{m_k} Y_{i,k}$  — общая средняя наработка. При

этом, поскольку  $X_j = j - 1$ , то  $\bar{X} = \frac{1}{n} \sum_{j=1}^l m_j (j - 1)$

и (11) приводится к виду

$$\hat{\beta} = \left( \frac{\frac{1}{n} \sum_{k=1}^l m_k k Y_{\bullet k} - \bar{Y} \sum_{k=1}^l m_k k}{S_x^2} \right),$$

где  $S_x^2 = \frac{1}{n} \sum_{k=1}^l m_k k^2 - \left( \frac{1}{n} \sum_{k=1}^l m_k k \right)^2$ .

Оценка по МНК параметра  $\gamma$  в силу (4) будет очевидно равна

$$\hat{\gamma} = \exp \hat{\beta} = \exp \left( \frac{\frac{1}{n} \sum_{k=1}^l m_k k \bar{Y}_{\bullet k} - \bar{Y} \sum_{k=1}^l m_k k}{S_x^2} \right).$$

Если  $\Delta_1$  имеет логнормальное распределение с параметрами  $m$  и  $\sigma^2$  (в общем случае неизвестными), то  $\ln \Delta_1$  будет иметь нормальное распределение с математическим ожиданием  $m$  и дисперсией  $\sigma^2$ . В этом случае можно построить следующий параметрический доверительный интервал как для углового коэффициента, так и для знаменателя геометрического процесса:

$$\beta = \hat{\beta} \pm \frac{\sigma}{\sqrt{n} S_x} u_{1-\alpha/2}, \quad \gamma = \exp \left( \hat{\beta} \pm \frac{\sigma}{\sqrt{n} S_x} u_{1-\alpha/2} \right). \quad (12)$$

Если дисперсия  $\sigma^2$  неизвестна, то она заменяется остаточной дисперсией

$$S^2 = \frac{n}{n-2} (S_y^2 - \hat{\beta}^2 S_x^2),$$

где  $S_y^2 = \frac{1}{n} \sum_{k=1}^l m_k \bar{Y}_{\bullet k}^2 - \bar{Y}^2$ , а квантиль нормального

закона  $u_{1-\alpha/2}$  — на квантиль распределения Стьюдента того же уровня и с  $n-2$  степенями свободы.

Нулевую гипотезу (5) можно проверить с помощью доверительного множества (12). Кроме этого, в параметрическом МНК принято проверять адекватность регрессионной модели критерием Фишера, основанном на следующей статистике:

$$F = S_2^2 / S_1^2, \quad (13)$$

где  $S_1^2 = \frac{1}{n-l} \sum_{k=1}^l \sum_{i=1}^{m_i} (Y_{i,k} - \bar{Y}_{\bullet k})^2$  — дисперсия вос-

производимости;  $S_2^2 = \frac{1}{l-2} \sum_{k=1}^l (\bar{Y}_{\bullet k} - \hat{\beta}(k-1 - \bar{X}) -$

$\bar{Y})^2$  — дисперсия адекватности. При расчете дисперсии воспроизводимости не применяется функция регрессии. На ее значение влияет только ошибка эксперимента. Дисперсия неадекватности, напротив, предназначена для измерения различия параметрической функции регрессии и выборочного среднего, являющегося, по сути, непараметрической оценкой регрессии в точке наблюдения. Чем больше это отличие, тем более модель неадекватна. Отсюда правило проверки адекватности модели. Принять нулевую гипотезу о том, что модель адекватна с уровнем значимости  $\alpha$  в случае, если статистика (13) окажется меньше квантиля  $F_{1-\alpha}(l-2, n-l)$  — распределения Фишера—Снедекора. Понятно, что

в случае принятия нулевой гипотезы об адекватности простой линейной регрессии, можно с тем же уровнем значимости утверждать — модель геометрического процесса адекватна. Однако это утверждение должно сопровождаться проверкой согласия остатков  $\ln \Delta_1^{(i,j)}$  с нормальным распределением.

Рассмотрим несколько примеров расчета знаменателя геометрического процесса и проверки гипотезы его значимости, а также практический расчет по имеющимся наработкам до отказа некоторых систем АЭС в ВВЭР-1000 Нововоронежской АЭС, представленных в работе [15].

**Практический пример 1.** В этом примере разберем построение точечной и интервальной оценок коэффициента  $\gamma$  в случае "стареющей" системы. При "старении" наработки до отказа будут стохастически уменьшаться, т. е.  $\gamma < 1$ . Нарботки между отказами моделировали по закону Вейбулла с параметром масштаба  $\lambda = 1$  и параметром формы  $\alpha = 2$ . Параметр  $\gamma$  полагался равным 0,7. Под наблюдением предположительно находилось три объекта, от каждого известно 3, 4 и 5 наработок между отказами (табл. 1, рис. 1).

Число точек  $n = 12$ . Число всевозможных пар точек будет равно  $N = C_{12}^2 - 3C_3^2 - C_2^2 = 56$ . Результаты расчетов по формулам (6), (7) сведены в табл. 2.

Таблица 1

Исходные данные для расчетов

Объект	1-я наработка	2-я наработка	3-я наработка	4-я наработка	5-я наработка
1	6,69	1,59	2,24	0,50	0,35
2	5,51	2,04	0,40	1,49	
3	2,30	0,63	1,07		

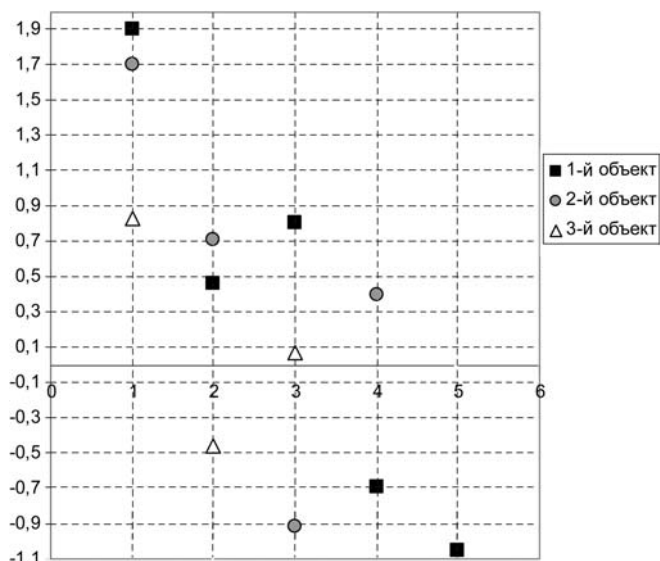


Рис. 1. Логарифмические наработки между отказами "стареющего" объекта

Вариационный ряд

$k$	1	2	3	4	5	6	7	8	9	10
$\gamma_{(k)}$	0,094	0,114	0,198	0,223	0,233	0,238	0,246	0,254	0,271	0,272
$k$	11	12	13	14	15	16	17	18	19	20
$\gamma_{(k)}$	0,289	0,306	0,371	0,394	0,400	0,419	0,421	0,440	0,449	0,467
$k$	21	22	23	24	25	26	27	28	29	30
$\gamma_{(k)}$	0,477	0,494	0,501	0,522	0,554	0,560	0,570	0,578	0,601	0,602
$k$	31	32	33	34	35	36	37	38	39	40
$\gamma_{(k)}$	0,606	0,623	0,637	0,645	0,647	0,667	0,670	0,681	0,692	0,696
$k$	41	42	43	44	45	46	47	48	49	50
$\gamma_{(k)}$	0,822	0,854	0,865	0,888	0,893	0,927	0,968	0,985	1,093	1,236
$k$	51	52	53	54	55	56				
$\gamma_{(k)}$	1,396	1,403	1,543	1,705	3,569	3,692				

В качестве точечной оценки параметра  $\hat{\gamma}$  берем выборочную медиану, которая (согласно формуле (8)) является полусуммой 28-го и 29-го элементов вариационного ряда и равна  $\hat{\gamma} = 0,589$ .

Посчитаем дисперсию статистики Кендалла по формуле (10):

$$D(K) = \frac{12 \cdot 11 \cdot 29 - 3 \cdot 3 \cdot 2 \cdot 11 - 2 \cdot 1 \cdot 9}{18} = 200,67.$$

Пусть уровень значимости  $\alpha = 0,05$ , тогда  $K_{1-\alpha/2} = \sqrt{200,67} \cdot 1,96 = 27,77$ . По формулам (9) получаем  $M_1 = 14$  и  $M_2 = 42$ , т. е. искомый доверительный интервал  $\gamma \in (0,394; 0,854)$  с надежностью 95 %. При этом, поскольку данный интервал не накрывает точку 1, имеет смысл считать точечную оценку  $\hat{\gamma} = 0,589$  значимой и отвергнуть нулевую гипотезу (5)  $H_0: \gamma = 1$  с выбранным уровнем значимости  $\alpha$ . Таким образом, можно считать, что данные наработки принадлежат системе "старееющего" типа.

Для сравнения приведем параметрическую оценку (12) параметра деградации:  $\hat{\gamma} = 0,576$ ,  $\gamma \in (0,491; 0,675)$ .

Как и следовало ожидать, доверительный интервал получился существенно меньше. Однако для статистического обоснования таких выводов необходима тщательная проверка остатков  $\ln \Delta_1$  на принадлежность нормальному распределению. В разрабатываемом авторами программном комплексе такую проверку планируется проводить по ряду критериев.

Хотелось бы отметить, что моделируемое значение 0,7 не попало в доверительный интервал. С одной стороны, это можно объяснить случайностью. С другой стороны, предостерегаясь от возможных ошибок, можно ожидать, что построенное доверительное множество обладает чрезмерной оптимистичностью, т. е. истинный уровень надежности несколько ниже 95 %, что вызвано неверным предположением о распределении остатков.

По критерию Фишера модель линейной регрессии при всех остальных выводах получилась адекватной. Посчитанное  $p$ -значение равняется 0,379.

**Практический пример 2.** Проведем расчет параметров представленной модели на реальных данных. Для этого воспользуемся информацией об отказах одной из систем АЭС 5-го блока Нововоронежской АЭС, представленной в работе [15] (рис. 2).

Анализируя рисунок, можно ожидать, что значимой тенденции скорее всего найдено не будет. Число точек  $n = 26$ . Число всевозможных пар точек будет велико:  $N = C_{26}^2 - 6C_3^2 - C_2^2 = 306$ , поэтому приведем окончательные результаты (табл. 3).

Поскольку интервал  $\gamma \in (0,845; 1,102)$  накрывает точку 1 с уровнем значимости 0,1, а значит и со всеми общепринятыми уровнями значимости, имеет смысл считать точечную оценку  $\hat{\gamma} = 0,957$  незначимой и принять нулевую гипотезу (12)  $H_0: \gamma = 1$ , т. е. можно считать, что гипотезу о деградации в рамках модели геометрических процессов для системы подпитки и борного регулирования АЭС имеет смысл отвергнуть.

Для сравнения приведем параметрическую оценку параметра деградации  $\hat{\gamma} = 0,973$  и в табл. 4 параметрические доверительные интервалы.

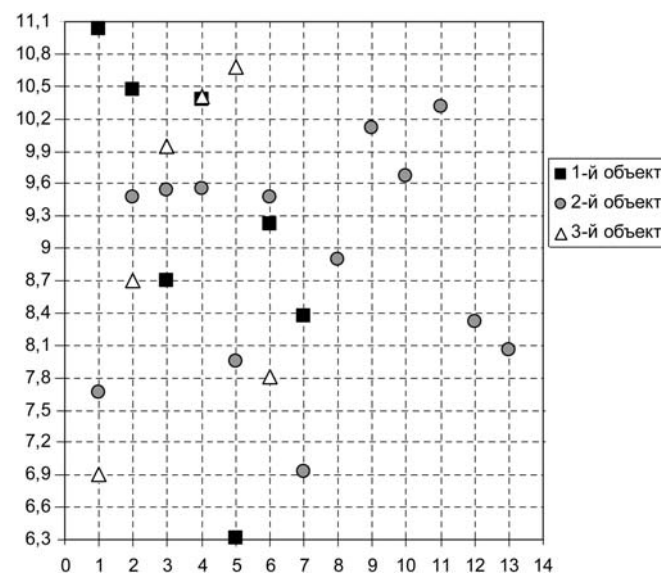


Рис. 2. Логарифмические наработки между отказами системы 5ПН

Таблица 3

Непараметрические доверительные интервалы		
$1 - \alpha$	$\underline{\gamma}$	$\bar{\gamma}$
0,90	0,845	1,102
0,95	0,809	1,136
0,99	0,770	1,243

Таблица 4

Параметрические доверительные интервалы		
$1 - \alpha$	$\underline{\gamma}$	$\bar{\gamma}$
0,90	0,898	1,053
0,95	0,884	1,071
0,99	0,854	1,108

Как и следовало ожидать, доверительный интервал получился существенно меньше. Но все равно даже по параметрическому критерию деградация отсутствует. Интересно отметить, что по критерию Фишера модель линейной регрессии при всех остальных выводах получилась адекватной. Посчитанное  $p$ -значение высоко и равняется 0,918.

### Заключение

В статье приведена модель геометрических процессов. В рамках этой модели исследуется вопрос оценивания знаменателя этого процесса. Предложена непараметрическая и параметрическая оценки, а также доверительные интервалы для коэффициента деградации. Разобран ряд примеров, включая реальные данные об отказах некоторых систем АЭС 5-го блока Нововоронежской АЭС. Статистически показано, что имеющаяся информация позволяет утверждать, что в рамках модели геометрических процессов старения у этой системы не наблюдается.

### Список литературы

1. Саенко Н. Б. Учет неполноты восстановления элементов при расчете надежности систем // Изв. Вузов. Приборостроение. 1994. № 11–12. С. 76–79.
2. Lam Y. A note on the optimal replacement problem // Advances in Applied Probability. 1988. N 20. P. 479–482.
3. Lam Y. Geometric processes and replacement problem // Acta Mathematicae Applicatae Sinica: english series. 1988. N 4 (4). P. 366–377.
4. Lam Y. Some limit theorems in geometric processes // Acta Mathematicae Applicatae Sinica: english series. 2003. N 3 (19). P. 405–416.
5. Finkelstein M. Failure rate modelling for reliability and risk. Springer, 2008. 290 p.
6. Theil H. A rank-invariant method of linear and polynomial regression analysis, I // Proc. Kon. Ned. Akad. v. Wetensch. A. 1950. Vol. 53. P. 386–392.
7. Theil H. A rank-invariant method of linear and polynomial regression analysis, II // Proc. Kon. Ned. Akad. v. Wetensch. A. 1950. Vol. 53. P. 521–525.
8. Theil H. A rank-invariant method of linear and polynomial regression analysis, III // Proc. Kon. Ned. Akad. v. Wetensch. A. 1950. Vol. 53. P. 1397–1412.
9. Sen P. K. Estimates of the regression coefficient based on Kendall's tau // Journal of the American Statistical Association. 1968. Vol. 63. P. 1379–1389.
10. Холлендер М., Вулф Д. Непараметрические методы статистики. М.: Финансы и статистика, 1983. 518 с.
11. Тюрин Ю. Н., Макаров А. А. Статистический анализ данных на компьютере. М.: ИНФРА-М, 1998. 528 с.
12. Ивченко Г. И., Медведев Ю. И. Математическая статистика. М.: Высшая школа, 1984. 248 с.
13. Афифи А., Эйзен С. Статистический анализ. Подход с использованием ЭВМ. М.: Мир, 1982. 488 с.
14. Чернова Н. И. Математическая статистика: учеб. пособие. Новосибирск: Изд-во Новосиб. гос. ун-та, 2007. 148 с.
15. Антонов А. В., Зюляева Н. Г., Таратунин В. В. и др. Оценка показателей надежности и оптимизация объема запасных элементов насосных агрегатов ЦН 60-180 ядерными и корневыми методами. Технический отчет. М.: ВНИИАЭС. 2008. 47 с.

УДК 004.3

Ш. А. Оцоков, д-р техн. наук, доц.,  
Московский энергетический институт  
(технический университет)

## Метод проверки необходимости округления при организации высокоточных вычислений в модулярной арифметике<sup>1</sup>

*Предложен метод проверки необходимости округления в процессе высокоточных вычислений с двоичными дробями в модулярной арифметике. Получены экспериментальные оценки эффективности высокоточных вычислений в модулярной арифметике на примере нахождения скалярного произведения векторов.*

**Ключевые слова:** модулярная арифметика, двоичные дроби, высокоточные вычисления

<sup>1</sup> Работа выполнена при поддержке гранта президента для молодых докторов наук МД-5302.2012.9.

### Введение

В последнее время при решении ряда задач математического моделирования в нанoeлектронике и других областях возрастает потребность в высокоточных вычислениях, т. е. выполнения арифметических операций с плавающей точкой с точностью, более высокой, чем поддерживаемой современными ЭВМ. В настоящее время существует множество библиотек, поддерживающих высокоточные вычисления, например, ZREAL (Россия), MPARITH (Германия), GMP (США) и др. [1]. Основной проблемой этих библиотек является сильная зависимость времени выполнения арифметических операций от точности, что может приводить к резкому (неприемлемому) росту времени решения вычислительных задач на суперкомпьютерах.

Исследование модулярной системы счисления показало возможность ускорения высокоточных вычислений при решении ряда частных задач в модулярной арифметике, таких как: решение дифференциальных уравнений методами Рунге—Кутты,

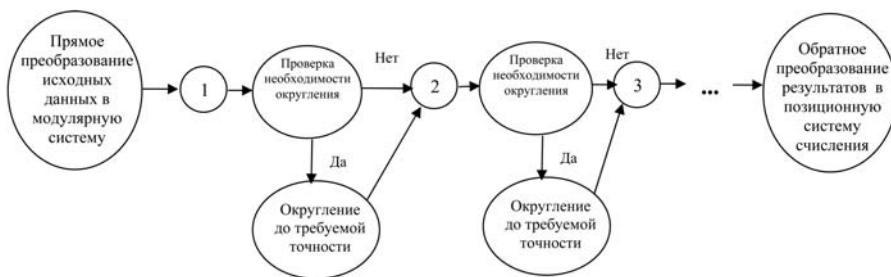


Рис. 1. Схема высокоточных вычислений с отложенным округлением

Эйлера, систем линейных уравнений методами Гаусса—Зейделя, релаксации и др. [2].

Известно, что арифметические операции в модулярной системе счисления (МСС) осуществляются не с исходными числами, а с остатками от их деления на заранее выбранные простые числа (модули), причем в процессе таких операций по каждому модулю отсутствуют переносы. В результате этого арифметические операции с числами большой длины сводятся к множеству таких же операций с числами меньшей длины, которые могут выполняться параллельно и независимо друг от друга.

Благодаря этому модулярную арифметику широко применяют для ускорения вычислений в цифровой обработке сигналов, криптографии и других областях [3].

Недостатком этой системы счисления является сложность выполнения таких операций как: деление, сравнение, преобразование чисел из модулярной системы в позиционную, округление и др.

Рассмотрим схему высокоточных вычислений в МСС, представленную на рис. 1. В соответствии с этой схемой вычисления проводятся следующим образом:

- 1) прямое преобразование исходных данных в МСС;
- 2) выполнение первой арифметической операции в МСС;
- 3) проверка необходимости округления. Если оно требуется, проводится округление, если нет — переход на п. 4;
- 4) выполнение второй арифметической операции в МСС;
- 5) проверка необходимости округления. Если оно требуется, проводится округление, если нет — переход на п. 6;
- 6) выполнение третьей арифметической операции и т. д.;
- 7) обратное преобразование полученных результатов из МСС в позиционную систему.

Высокоточные вычисления по предложенной схеме проводятся с двоичными дробями следующего вида:

$$A = \frac{K}{2^{k_f}}, \quad (1)$$

где  $K$  — целое число такое, что  $|K| \leq 2^{n_f + k_f} - 1$ ;  $n_f$  — длина целой части числа  $A$  в формате с фиксированной точкой;  $k_f$  — длина дробной части числа  $A$  в формате с фиксированной точкой.

Цель настоящей работы состоит в разработке метода проверки необходимости округления при вычислениях в модулярной арифметике с двоичными дробями вида (1) по схеме, представленной на рис. 1.

Рассмотрим последовательно каждую из операций:

- преобразование числа в МСС;
- арифметические операции;
- обратное преобразование из МСС;
- округление;
- метод проверки необходимости округления.

### 1. Преобразование двоичного числа в МСС

Пусть  $p_1, p_2, \dots, p_n$  — модули МСС, такие что  $p_1 < p_2 < \dots < p_n$ .

Формулы для преобразования чисел (1) в МСС имеют вид

$$\alpha_i = |A|_{p_i} = |K \cdot 2^{-k_f}|_{p_i}, \quad i = 1, \dots, n,$$

и число  $A$  в МСС записывается в виде

$$A = [(\alpha_1, \alpha_2, \dots, \alpha_i, \dots, \alpha_n), t],$$

где  $\alpha_i = |A|_{p_i} = |K \cdot 2^{-k_f}|_{p_i}$  — мантисса;  $|A|_y$  — остаток от деления числа  $A$  на  $y$ .

### 2. Правила выполнения арифметических операций в МСС

Пусть даны числа

$$A_1 = [(\alpha_1, \alpha_2, \dots, \alpha_i, \dots, \alpha_n), t_1, r_1];$$

$$A_2 = [(\beta_1, \beta_2, \dots, \beta_i, \dots, \beta_n), t_2, r_2].$$

Результатом выполнения арифметической операции сложения, вычитания или умножения является число

$$A_3 = [(\gamma_1, \gamma_2, \dots, \gamma_i, \dots, \gamma_n), t_3, r_3].$$

#### Сложение

Шаг № 1. Вычисляем:  $\gamma_i = |\alpha_i + \beta_i|_{p_i}$ ,  $i = 1, \dots, n$ .

Шаг № 2. Порядок результата  $t_3$  равен большему порядку чисел  $A_1$  и  $A_2$ .

#### Вычитание

Шаг № 1. Вычисляем:  $\gamma_i = |\alpha_i - \beta_i|_{p_i}$ ,  $i = 1, \dots, n$ .

Шаг № 2. Порядок результата  $t_3$  равен большему порядку чисел  $A_1$  и  $A_2$ .

### Умножение

Шаг № 1. Вычисляем:  $\gamma_i = |\alpha_i \cdot \beta_i|_{p_i}, i = 1, \dots, n$ .

Шаг № 2. Порядок результата  $t_3$  равен сумме порядков чисел  $A_1$  и  $A_2$ .

### 3. Обратное преобразование из МСС

Пусть в результате вычислений с числами вида (1) получен результат

$$A = [(\alpha_1, \alpha_2, \dots, \alpha_p, \dots, \alpha_n), t].$$

Операция обратного преобразования из МСС сводится к восстановлению числителя  $K$  в МСС, к проверке его знака и вычислению отношения числителя к двойке в степени порядка  $t$ .

1. Для восстановления числителя в МСС  $K = (\beta_1, \beta_2, \dots, \beta_p, \dots, \beta_n)$  воспользуемся формулой

$$\beta_i = \alpha_i(2^t) \bmod p_i, i = 1, \dots, n.$$

2. Для проверки знака  $K$  используем китайскую теорему об остатках. В МСС отрицательные числа представляются как дополнение до модуля или до произведения модулей, если их несколько.

Пусть

$$P_{(1/2)} = \frac{1}{2} P, \text{ тогда}$$

$[0, \dots, P_{(1/2)})$  — диапазон представления положительных чисел;

$[P_{(1/2)}, \dots, P)$  — диапазон представления отрицательных чисел.

В соответствии с китайской теоремой об остатках

$$K = \sum_{i=1}^n B_i \beta_i - \text{rank} P, \quad (2)$$

где  $B_i$  — ортогональные базисы;  $\text{rank}$  — ранг (число, которое подбирается таким образом, чтобы выполнялось неравенство  $K < P$ ).

Ортогональные базисы  $B_i$  являются константами для МСС с заданными модулями и определяются по формулам

$$B_i = m_i \frac{P}{p_i};$$

$$m_i = \left| \frac{P}{p_i} \right|^{-1}. \quad (3)$$

Рассмотрим способ определения числа  $\text{rank}$ .

Максимально возможное значение  $\text{rank}$  определяется из следующих неравенств:

$$\sum_{i=1}^n B_i \beta_i \leq \sum_{i=1}^n B_i (p_i - 1) < P \left( \sum_{i=1}^n p_i - n \right).$$

Отсюда видно, что

$$\text{rank} < \left( \sum_{i=1}^n p_i - n \right).$$

Выберем дополнительный модуль МСС  $p^*$  таким, чтобы удовлетворялось условие

$$p^* > \left( \sum_{i=1}^n p_i - n \right).$$

Тогда

$$|K|_{p^*} = \left| \sum_{i=1}^n B_i \beta_i - \text{rank} P \right|_{p^*}$$

или

$$|\text{rank}|_{p^*} = \left| \left( \sum_{i=1}^n B_i \beta_i - K \right) P^{-1} \right|_{p^*}.$$

Но так как  $\text{rank} < p^*$ , то

$$\text{rank} = \left| \left( \sum_{i=1}^n B_i \beta_i - K \right) P^{-1} \right|_{p^*}. \quad (4)$$

На рис. 2 приведена схема для вычисления  $K$  по формуле (2).

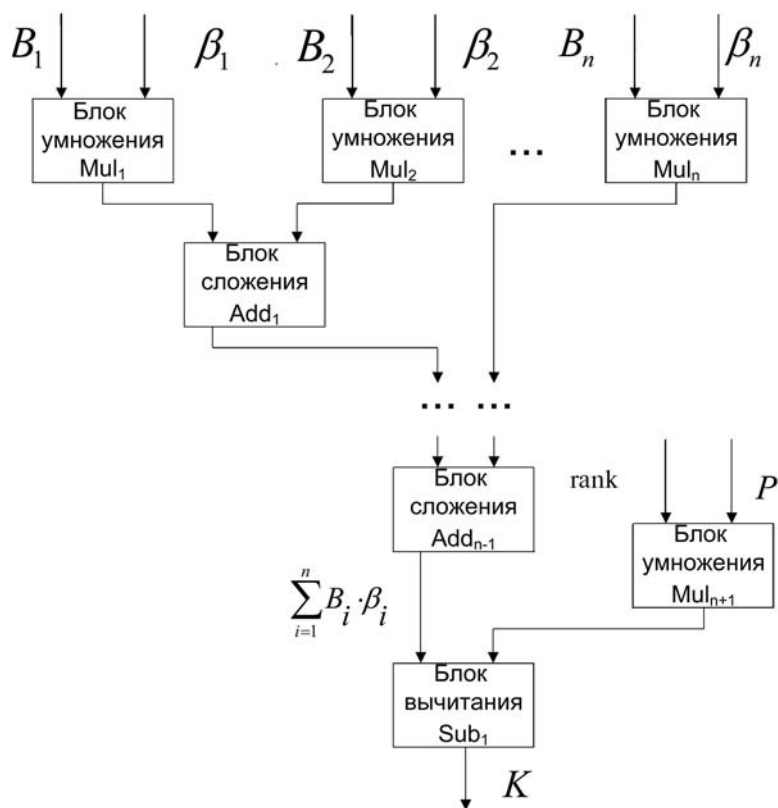


Рис. 2. Схема для быстрого вычисления  $K$  по формуле (2)

В соответствии с приведенной выше схемой  $K$  может быть вычислено по формуле (2) за  $\lceil \log_2 n \rceil + 1$  шагов при известном значении  $\text{rank}$ .

Операция  $\lceil x \rceil$  определяется следующим образом:

$$\min(n | n \geq x, n - \text{целое}).$$

Если значение  $\text{rank}$  неизвестно, то его можно вычислить по формуле (4). Выражения  $\left| \sum_{i=1}^n B_i \beta_i \right|_{p^*}$ ,  $\sum_{i=1}^n B_i \beta_i$  в формулах (4) и (2) могут быть вычислены параллельно и независимо друг от друга за  $\lceil \log_2 n \rceil$  шагов.

Следует отметить, что время выполнения одного шага при вычислении по формулам (2) и (4) различное, так как в формуле (4) все вычисления проводятся с числами разрядностью  $\lceil \log_2 p^* \rceil$ , а в формуле (2) — разрядностью  $\lceil \log_2 P \rceil$ .

Сравниваем  $K$  с  $P_{(1/2)}$ . Если  $K$  больше, то  $K$  положительное, если меньше, то отрицательное.

3. Если  $K$  положительное, то вычисляем отношение  $K$  к двойке в степени порядка  $t$  в формате с плавающей точкой.

4. Если  $K$  отрицательное, то находим дополнение  $K$  до модулей  $K^- = (p_1 - \beta_1, p_2 - \beta_2, \dots, p_i - \beta_i, \dots, p_n - \beta_n)$ .

Восстанавливаем  $K^-$  по китайской теореме об остатках в соответствии со схемой, представленной на рис. 2, находим отношение  $(-1)K^-$  к двойке в степени порядка  $t$ .

Преобразование чисел

$$K^- = (p_1 - \beta_1, p_2 - \beta_2, \dots, p_i - \beta_i, \dots, p_n - \beta_n),$$

$$K = (\beta_1, \beta_2, \dots, \beta_i, \dots, \beta_n)$$

в соответствии со схемой, представленной на рис. 2, можно проводить параллельно и независимо друг от друга.

В этом случае для обратного преобразования из МСС потребуется  $\lceil \log_2 P \rceil + 2$  шагов. Обратное преобразование является более сложной операцией, чем прямое преобразование. Однако прямое и обратное преобразования выполняются всего лишь один раз в начале вычислений в МСС и один раз — в конце.

Операция округления выполняется тогда, когда результат выходит за пределы допустимого диапазона, определяемого произведением выбранных модулей. Для обнаружения этого выхода необходима оценка числителя и знаменателя результата арифметических операций с двоичными дробями. В следующем пункте представлены формулы для таких оценок.

#### 4. Оценка сверху результатов арифметических операций с двоичными дробями

$$\text{Пусть } A_1 = \frac{K_1}{2^{t_1}}, A_2 = \frac{K_2}{2^{t_2}}$$

и

$$r_1 = \lceil \log_2 K_1 \rceil, r_2 = \lceil \log_2 K_2 \rceil,$$

где  $r_1, r_2$  — порядки числителей  $A_1, A_2$ ;  $\lceil d \rceil$  — результат округления числа  $d$  до ближайшего целого в большую сторону.

Пусть

$$A_3 = A_1 A_2 = \frac{K_1 K_2}{2^{t_1 + t_2}}, r_3 \text{ — порядок числителя } A_3.$$

Тогда

$$\begin{aligned} \lceil \log_2 K_1 K_2 \rceil &= \\ &= \lceil \log_2 K_1 + \log_2 K_2 \rceil \leq \lceil \log_2 K_1 \rceil + \lceil \log_2 K_2 \rceil \end{aligned}$$

или

$$r_3 \leq r_1 + r_2. \quad (5)$$

Для суммы  $A_1, A_2$  верно неравенство

$$\begin{aligned} A_3 &= A_1 + A_2 = \frac{K_1}{2^{t_1}} + \frac{K_2}{2^{t_2}} = \\ &= \frac{K_1 \cdot 2^{\max(t_1, t_2) - t_1} + K_2 \cdot 2^{\max(t_1, t_2) - t_2}}{2^{\max(t_1, t_2)}} \leq \\ &\leq \frac{2 \cdot \max(K_1, K_2) 2^{\max(\max(t_1, t_2) - t_1, \max(t_1, t_2) - t_2)}}{2^{\max(t_1, t_2)}} \leq \\ &\leq \frac{2 \cdot 2^{\max(r_1, r_2)} 2^{\max(\max(t_1, t_2) - t_1, \max(t_1, t_2) - t_2)}}{2^{\max(t_1, t_2)}} \leq \\ &\leq \frac{2^{\max(r_1, r_2) + \max(\max(t_1, t_2) - t_1, \max(t_1, t_2) - t_2) + 1}}{2^{\max(t_1, t_2)}} \end{aligned}$$

или

$$\begin{aligned} r_3 &\leq \max(r_1, r_2) + \\ &+ \max(\max(t_1, t_2) - t_1, \max(t_1, t_2) - t_2) + 1. \quad (6) \end{aligned}$$

Пусть

$P = p_1 \cdot p_2 \cdot \dots \cdot p_n$  — произведение модулей

$$n_{(1/2)} = \left\lfloor \frac{\log_2 P}{2} \right\rfloor, n_{(1)} = \lfloor \log_2 P \rfloor,$$

где  $\lfloor d \rfloor$  — округление числа  $d$  до ближайшего целого в меньшую сторону.

Округление проводится тогда, когда

$$r_3 > n_{(1/2)}. \quad (7)$$



Так как, если  $r_1 > n_{(1/2)}$  и  $r_2 > n_{(1/2)}$ , то при умножении чисел  $A_1, A_2, r_3 > n_{(1)}$  возникает ошибка переполнения, которая не обнаруживается при вычислениях только в модулярной арифметике.

Поэтому в случае, если  $r_3 > n_{(1/2)}$ , необходимо округление.

### 5. Округление чисел в МСС

Пусть при выполнении арифметических операций в МСС получен результат, для которого условие (7) выполняется. В этом случае требуется его округление. Округление проводится путем преобразования результата в позиционную систему счисления в соответствии с п. 3, отбрасывания лишних разрядов дробной части результата (оставляется только лишь  $k_f$  цифр после запятой) и обратного преобразования полученного результата в МСС. Метод проверки необходимости округления в процессе выполнения арифметических операций представлен в п. 6.

### 6. Метод проверки необходимости округления при организации высокоточных вычислений в модулярной арифметике

Пусть даны числа

$$A_1 = [(\alpha_1, \alpha_2, \dots, \alpha_i, \dots, \alpha_n), t_1, r_1];$$

$$A_2 = [(\beta_1, \beta_2, \dots, \beta_i, \dots, \beta_n), t_2, r_2].$$

Результатом выполнения арифметической операции сложения, вычитания или умножения является число

$$A_3 = [(\gamma_1, \gamma_2, \dots, \gamma_i, \dots, \gamma_n), t_3, r_3].$$

Метод проверки необходимости округления при организации высокоточных вычислений в модулярной арифметике заключается в вычислении порядка результата  $r_3$  и проверки условия (7) после каждой арифметической операции сложения, вычитания и умножения.

#### Сложение

Шаг 1. Вычисляем:

$$\gamma_i = |\alpha_i + \beta_i|_{p_i}, i = 1, \dots, n.$$

Шаг 2. Порядок результата  $t_3$  равен большему порядку чисел  $A_1$  и  $A_2$ .

Шаг 3. Вычисляем

$$r_3 = \max(r_1, r_2) + \max(\max(t_1, t_2) - t_1, \max(t_1, t_2) - t_2) + 1.$$

Шаг 4. Проверяем условие (7), если оно выполняется, то выполняем округление.

#### Вычитание

Шаг 1. Вычисляем:  $\gamma_i = |\alpha_i - \beta_i|_{p_i}, i = 1, \dots, n.$

Шаг 2. Порядок результата  $t_3$  равен большему порядку чисел  $A_1$  и  $A_2$ .

Шаг 3. Вычисляем  $r_3 = \max(r_1, r_2) + \max(\max(t_1, t_2) - t_1, \max(t_1, t_2) - t_2) + 1.$

Шаг 4. Проверяем условие (7), если оно выполняется, то проводим округление.

#### Умножение

Шаг 1. Вычисляем:  $\gamma_i = |\alpha_i \cdot \beta_i|_{p_i}, i = 1, \dots, n.$

Шаг 2. Порядок результата  $t_3$  равен сумме порядков чисел  $A_1$  и  $A_2$ .

Шаг 3. Вычисляем  $r_3 = r_1 + r_2.$

Шаг 4. Проверяем условие (7), если оно выполняется, то проводим округление.

### 7. Численные эксперименты

Для оценки числа округлений проводились эксперименты по сложению случайно выбранных чисел в МСС с числителем от 1 до 1000, порядком знаменателя, изменяемом от 1 до 4. Число модулей было равным 10. Длина целой части числа  $n_f = 9$ , дробной части числа  $k_f = 5$ , коэффициент  $n_{(1/2)} = 29$ . На рис. 3 представлена зависимость порядка числителя результата вычисленного ( $r$ ) и реального от числа операций.

На рис. 3 видно, что когда порядок числителя достигает значения  $n_{(1/2)} = 29$ , происходит округление, в результате округления порядок числителя уменьшается и вычисления продолжают. Всего за 1000 операций произошло 13 округлений.

Эффективность высокоточных вычислений в модулярной арифметике оценивалась на примере решения модельной задачи — нахождения скалярного произведения векторов со случайно заданными координатами и числом координат, равным 10 000.

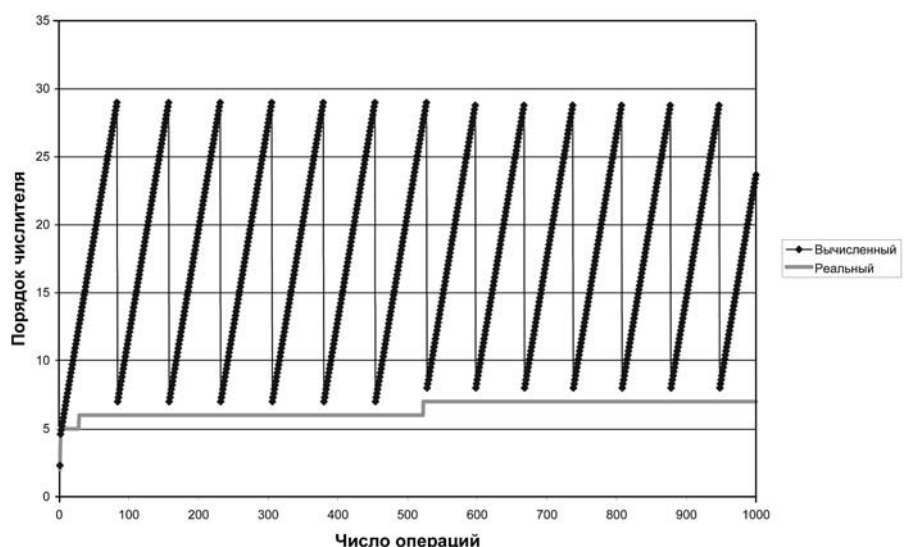


Рис. 3. Зависимость порядка числителя результата от числа операций при  $n_{(1/2)} = 29$

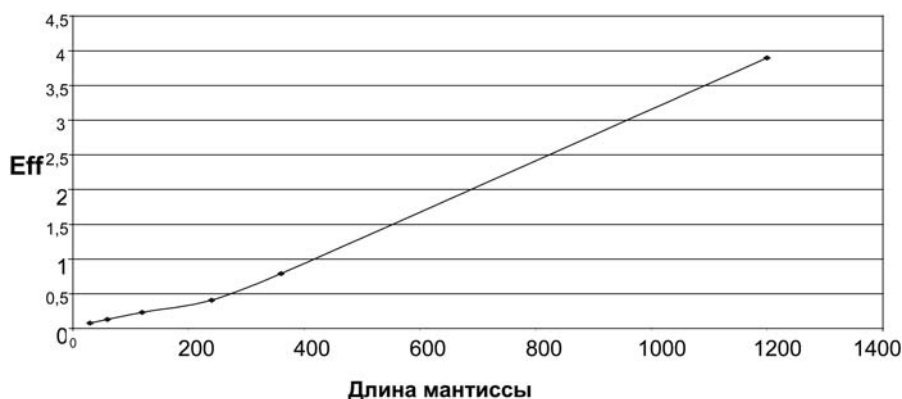


Рис. 4. Зависимость коэффициента абсолютного ускорения от длины мантиссы

Скалярное произведение вычислялось с использованием существующей библиотеки высокоточных вычислений MPArith с различной точностью и на основе модулярной арифметики при той же точности. Вычисления в модулярной арифметике проводились на многоядерном графическом ускорителе NVIDIA параллельно и независимо по каждому модулю.

Зависимость коэффициента абсолютного ускорения от длины мантиссы представлена на рис. 4. Коэффициент абсолютного ускорения Eff определяется следующим образом:

$$\text{Eff} = \frac{T_1}{T_2},$$

где  $T_1$  — время вычислений с использованием библиотеки MPArith;  $T_2$  — время вычислений в модулярной арифметике при той же точности.

На рис. 4 видно, что, начиная с длины мантиссы 420, происходит ускорение высокоточных вычислений в модулярной арифметике. Для длины ман-

тиссы меньше чем 500, Eff меньше единицы. Это объясняется накладными расходами, присущими вычислениям в модулярной системе счисления, связанными с прямым и обратным преобразованием исходных данных и результатов в модулярную систему счисления.

### Заключение

Как было указано выше, округление в модулярной системе счисления является сложной операцией. Чем меньше в процессе вычислений было проведено округлений, тем выше ускорение вычислений. В схеме высокоточных вычислений в модулярной системе счисления округление выполняется по мере необходимости, т. е. когда результат выходит за пределы допустимого диапазона, и эта ситуация обнаруживается с помощью представленного в настоящей работе метода. Данный метод может быть использован при решении дифференциальных уравнений методами Рунге—Кутты, Эйлера, систем линейных уравнений методами Гаусса—Зейделя, релаксации, нахождении скалярного произведения векторов, дискретного преобразования Фурье и др.

### Список литературы

1. Bailey D. H. High-Precision Computation and Mathematical Physics // Lawrence Berkeley National Laboratory. 2009.
2. Дзегеленок И. И., Оцоков Ш. А. Алгебраизация числовых представлений в обеспечении высокоточных суперкомпьютерных вычислений // Вестник МЭИ. 2010. № 3. С. 107—116.
3. Sonderstrand M. A., Jendins W. K., Junllien G. A., and Taylor F. J. Residue Number System Arithmetic: Modern Applications in Digital Signal Processing // IEEE Press, New York. 1986.

## Информация

### Конференция "Прагматик.РМ"

**Дата и место проведения:** Москва 25 августа.

**Формат мероприятия:** 1,5-часовые мастер-классы ведущих специалистов в 4—5 потоков.

**Аудитория:** ИТ-специалисты, разработчики, тестировщики ПО, системные и бизнес-аналитики, менеджеры, руководители подразделений и директора компаний.

#### Основные направления:

- Разработка ПО
- Тестирование ПО
- Управление проектами
- Системный анализ
- Soft Skills

**Контакты: e-mail: edu@stratoplan.ru**

**+7 921 931 78 07 (Россия)**

**+380 97 354 17 76 (Украина) +380 50 381 75 93 (Украина)**

УДК 004.415.2

**С. В. Косяков**, д-р техн. наук, проф., зав. каф.,  
**А. Б. Гадалов**, ст. преподаватель, нач. отд.,  
**А. М. Садыков**, аспирант,  
ФГБОУ ВПО "Ивановский государственный  
энергетический университет им. В. И. Ленина",  
e-mail: ksv@igt.ispu.ru

## Моделирование пространственных данных при решении задач дискретной оптимизации в среде ГИС

*Рассматриваются проблемы и методы расчета и поддержания в актуальном состоянии в среде ГИС матрицы исходных данных для решения транспортных задач и задач оптимального размещения предприятий. Приводятся описание метода сокращения времени перерасчета полной матрицы маршрутов транспортной задачи при изменении данных в графе дорожного движения и структура базы данных, позволяющая автоматизировать процесс обновления этой матрицы в среде ГИС.*

**Ключевые слова:** пространственные данные, матрица маршрутов, геоинформационная система, транспортная задача

### Введение

В настоящее время в практику работы предприятий начинают внедряться информационные системы, построенные на базе сочетания средств ГИС и методов решения вычислительно сложных задач в оптимизационной постановке. Одним из наиболее известных видов таких систем являются системы транспортной логистики, в состав которых входят программно реализованные алгоритмы решения динамических транспортных задач. Близкими к ним по математической постановке и методам реализации являются информационные системы для решения задач размещения предприятий. При реализации указанных видов систем возникают две существенные проблемы: 1) выбор и реализация алгоритмов оптимизации, обеспечивающих достаточную точность расчетов в приемлемое время; 2) выбор методов преобразования исходных данных о реальных объектах в формализованные исходные данные математической модели оптимизаци-

онной задачи. В данной статье рассматриваются вопросы, касающиеся решения второй проблемы, которая связана с реализацией программных средств подготовки и ведения исходных данных для решения задач оптимального планирования в ГИС. В научных публикациях по теории дискретной оптимизации эта проблема обычно не рассматривается. Однако при практической реализации информационных систем она может оказаться не менее сложной и важной, чем разработка алгоритмов оптимизации.

### 1. Постановка задачи

Математическая постановка транспортной задачи широко известна. Воспользовавшись формулировками из работ [1] и [2], будем рассматривать граф  $G = \{N, A\}$ , где  $N$  — множество вершин, которые соответствуют набору клиентов и обозначаются  $1, 2, \dots, n$ , и вершины  $0$  и  $n + 1$  соответствуют источнику (складу), из которого начинают и в котором заканчивают свой маршрут все автомобили;  $A$  — набор дуг, соединяющих соответствующие вершины графа (соответствующих клиентов); если  $i$  — один клиент, а  $j$  — другой, то дуга, их соединяющая, обозначается  $(i, j) \in A$ . Обозначим  $C$  — множество клиентов  $|C| = n$ . Каждой дуге соответствует время  $t_{ij}$  — время перемещения от клиента  $i, i \in C$ , к клиенту  $j, j \in C$ , и  $c_{ij}$  — стоимость пути автомобиля из  $i$  в  $j$ , которая выражается через  $d_{ij}$  — длину этого пути. Верхним индексом  $k$  обозначим соответствующий автомобиль,  $k \in V, V$  — множество автомобилей. Переменные  $X_{ij}^k$  принимают значения  $\{0, 1\}$ ,  $1$  означает, что автомобиль движется от вершины  $i$  к вершине  $j$ ,  $0$  — обратное. Математическая формулировка транспортной задачи записывается в виде:

$$\begin{aligned} \min \sum_{k \in V} \sum_{(i,j) \in A} c_{ij} X_{ij}^k, & \quad (1) \\ \sum_{k \in V} \sum_{(i,j) \in A} X_{ij}^k = 1, \quad \forall i \in C, \\ X_{ij}^k \in \{0, 1\} \quad \forall i, j \in A, \quad \forall k \in V. \end{aligned}$$

В зависимости от разновидности задачи в постановку вводятся различные дополнительные ограничения: на загрузку автомобилей, на время прибытия автомобилей к клиентам и т. д. Однако в любом случае исходные данные для оптимизаци-

онной задачи включают матрицу  $D$ , элементы которой  $d_{ij}$  соответствуют длине кратчайшего маршрута между пунктами  $i$  и  $j$ . В общем случае используемая в критерии (1) стоимость перемещения автомобиля из  $i$  в  $j$  задается некоторой функцией

$$c_{ij} = F(d_{ij}, t_{ij}, k). \quad (2)$$

Далее для упрощения изложения будем говорить только о расчете  $d_{ij}$ , поскольку  $t_{ij}$  может рассчитываться одновременно с длиной  $d_{ij}$  и храниться с ней в одной структуре данных. Будем называть эту структуру данных полной матрицей маршрутов (ПММ), имея в виду, что в действительности она включает несколько одинаковых по размеру матриц с характеристиками маршрутов множества  $A$ .

Аналогичную математическую постановку имеют задачи размещения предприятий, которые рассмотрены, например, в работе [3]. В этих задачах граф  $G$  определяется через  $N$  — множество вершин, в которых могут размещаться предприятия  $i$ , и  $A$  — набор дуг, соединяющих соответствующие вершины графа с точками размещения потребителей  $j$ . В отличие от транспортной задачи, где ПММ квадратная, в задачах размещения предприятий число строк и столбцов данной матрицы обычно различаются. Однако с точки зрения формирования матриц в среде ГИС эти различия несущественны. Далее для простоты будем рассматривать только транспортную задачу. Однако рассмотренные методы моделирования данных применимы и к задаче размещения.

Научные исследования в рассматриваемой области направлены в основном на преодоление вычислительной сложности алгоритмов решения комбинаторных задач, которые являются  $NP$ -сложными. При этом процесс получения ПММ не рассматривается, а элементы этой матрицы считаются заранее известными и постоянными. При реализации программных продуктов расчет ПММ выделяют в отдельный этап моделирования, который часто реализуется средствами сетевого анализа на топологической модели (графе) дорожной сети территории средствами ГИС. Кроме того, средства ГИС в прикладных информационных системах используются для визуализации исходных данных и результатов моделирования.

Как уже отмечалось, в реальной практике применения информационных систем вопросы эффективности расчета ПММ играют не менее важную роль, нежели сам оптимизационный расчет. При больших размерностях задачи построение этой матрицы также связано со значительными вычислительными затратами, а точность определения ее элементов существенно отражается на качестве решения задачи в целом. Поэтому для практического применения средств решения транспортных задач и задач размещения вопросы трудоемкости постро-

ения и поддержания в актуальном состоянии ПММ являются важными.

## 2. Построение ПММ в ГИС

Построение ПММ в ГИС может осуществляться в автоматическом режиме для заданного на карте множества клиентов  $S$  при наличии в базе пространственных данных транспортного графа территории  $S = \{P, L\}$ , где  $P$  — множество вершин, которые соответствуют перекресткам или точкам изменения условий движения;  $L$  — набор участков дорог между указанными точками. Каждому участку соответствуют характеристики условий движения по нему. Пользователь ГИС может редактировать на карте территории указанный граф, изменять характеристики движения на любой из его дуг (участков дорог), а также добавлять и удалять клиентов.

Для построения каждого элемента  $d_{ij}$  ПММ используется специальная процедура, с помощью которой строится оптимальный маршрут между парой точек на карте. Критерием оптимальности маршрута может являться время, расстояние или их комбинация. В дальнейшем будем называть этот маршрут кратчайшим в смысле использованного критерия. Процедура автоматического построения каждого кратчайшего маршрута обычно реализуется на основе алгоритма Дейкстры [4]. При этом найденный маршрут определяется как подмножество  $L^{ij} \subset L$ , включающее участки дорог общего транспортного графа  $S$ , вошедшие в маршрут.

На рис. 1 показан фрагмент графа  $S$ , на котором с помощью процедуры построения кратчайшего маршрута построены (выделены) три маршрута проезда между точками 0 — склад, 1 — первый клиент, 2 — второй клиент (здесь считаем, что кратчайшие маршруты в прямом и обратном направлении движения совпадают, но в общем случае они могут отличаться). Показанные толстыми линиями маршру-

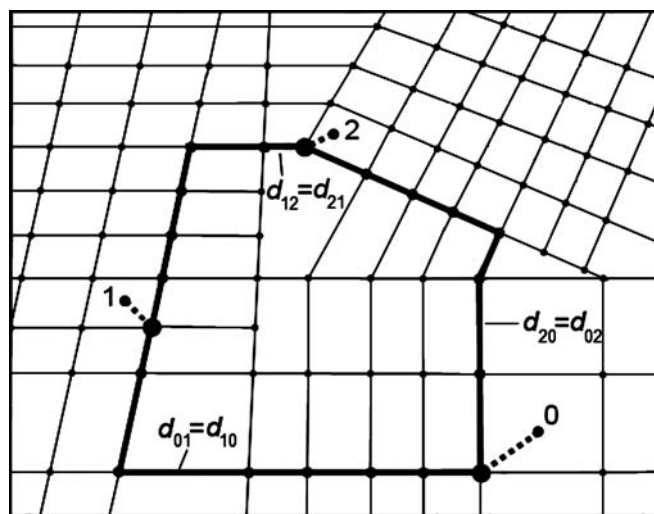


Рис. 1. Пример построения маршрутов на транспортном графе

руты образуют рассмотренное в постановке задачи множество  $A$  дуг графа  $G$ . Длины этих дуг соответствуют элементам  $d_{01}, d_{02}, d_{10}, d_{12}, d_{20}, d_{21}$  ПММ, имеющей размерность  $3 \times 3$  (диагональные элементы матрицы всегда нулевые).

Точки размещения объектов, между которыми осуществляется движение, географически могут не совпадать с узлами графа  $S$ . Поэтому при построении маршрутов могут использоваться специальные процедуры построения дополнительных элементов "вывода" маршрута на ближайшие точки существующего транспортного графа. На рис. 1 эти элементы показаны упрощенно пунктирными линиями к существующим перекресткам. Поскольку эти построения не меняют сути рассматриваемой проблемы, такие элементы для упрощения в дальнейшем изложении учитывать не будем.

Задача построения ПММ имеет полиномиальную сложность. При этом время расчета матрицы зависит от размерности графа  $S$ . При построении ПММ процедура построения кратчайшего маршрута запускается в цикле для всех комбинаций точек  $i$  и  $j$ . Число вызовов процедуры построения оптимального маршрута не превышает  $n^2 - n$  раз. Тем не менее при больших значениях  $n$  и большой мощности  $L$  процесс построения ПММ занимает значительное время. Известны алгоритмы, строящие одновременно маршруты между всеми клиентами, например алгоритм Флойда—Уоршолла [4], но их вычислительная сложность соизмерима с последовательным поиском маршрутов и их применение не устраняет рассматриваемую проблему.

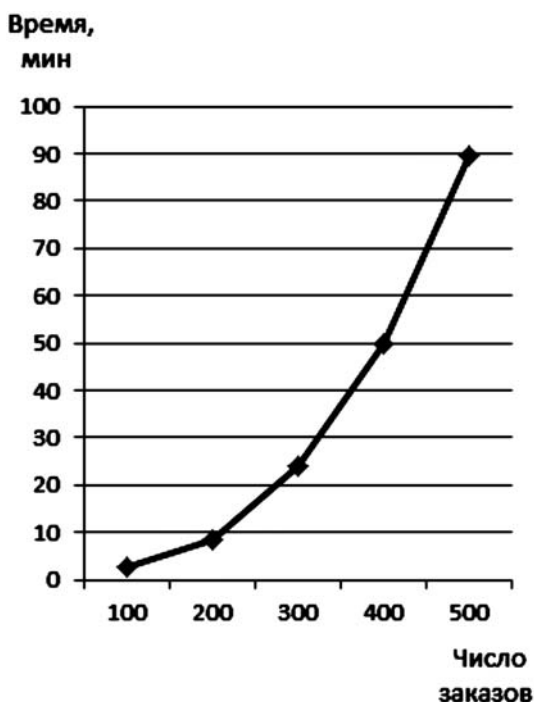


Рис. 2. Зависимость времени расчета ПММ от числа клиентов

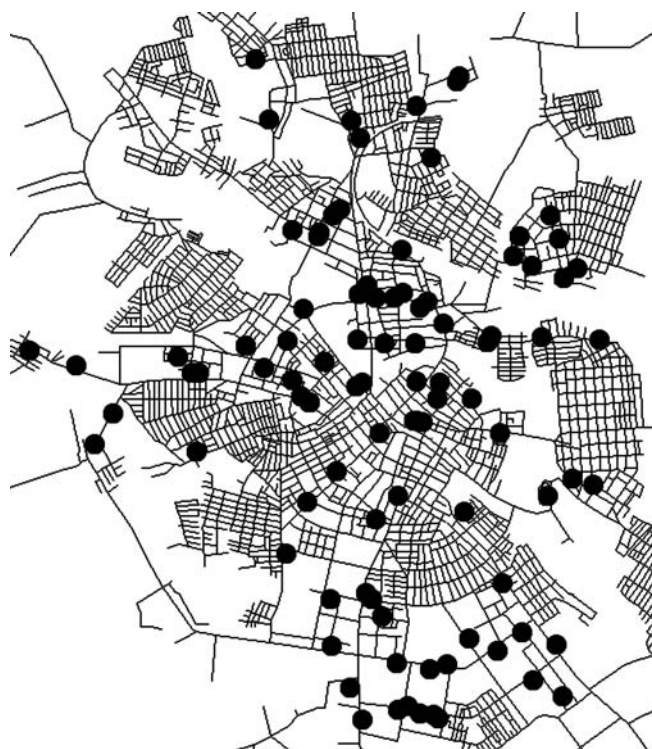


Рис. 3. Представление в ГИС модели дорожной сети города с указанием местоположения клиентов

Задача построения ПММ в реальных условиях работы приложений на рядовых компьютерах, на которых обычно и эксплуатируются подобные приложения, может занимать несколько часов. В рамках реализации прикладных систем оптимизации грузоперевозок в городе Иванове [5] авторами проведены эксперименты по определению времени расчета матрицы  $D$  для разной мощности множества клиентов, реализующих продукты питания в городе. На рис. 2 приведен график зависимости времени расчета ПММ от числа клиентов в модели. Транспортный граф города и прилегающих территорий включал 6313 участков дорожной сети. Вид использованной при расчете модели, включающей слой графа дорожной сети (линии) и слой клиентов (кружки), приведен на рис. 3. Измерения проводили на компьютере с процессором Intel Core2 Duo 2,33 ГГц и объемом ОЗУ 4 Гбайт. Результаты расчетов сохранялись в таблицах базы данных средствами СУБД MS SQL Server Compact Edition.

Как видно из приведенного графика, при малом числе элементов время построения матрицы несущественно, и ее можно полностью рассчитывать перед каждым построением плана грузоперевозок. При числе клиентов более 200 время построения матрицы значительно возрастет. При первоначальном создании матрицы такие затраты неизбежны и обоснованы. Однако, если в процессе эксплуатации программы в условия задачи необходимо внести оперативные изменения, связанные, например, с временным выводом участков дорожной сети в

ремонт, то полный пересчет матрицы может стать неприемлемым по причине слишком больших затрат времени на эту процедуру. В этом случае приходится либо отказываться от учета таких ситуаций при планировании, что приводит к снижению качества рассчитанных планов грузоперевозок, либо прибегать к специальным методам частичного обновления элементов матрицы с учетом локального характера правки графа  $S$ .

### 3. Метод динамической актуализации ПММ в ГИС

Ввиду жестких экономических ограничений на использование приложений рассматриваемого класса, применение многопроцессорной техники и других дорогостоящих технических решений для работы программного обеспечения подобных систем чаще всего оказывается неоправданным. В этих условиях для сокращения времени обновления ПММ можно использовать пересчет ограниченных частей этой матрицы. Такая возможность обусловлена тем, что изменения условий движения чаще всего имеют локальный характер. Предлагаемый авторами метод частичного обновления матрицы основан на ведении единой модели данных для транспортной сети города, матрицы транспортных затрат и клиентов. Этот метод легко реализуется в среде объектно-ориентированных ГИС, использующих парадигму баз геоданных.

Метод опирается на согласованное представление в единой базе геоданных графа транспортной сети и ПММ. Основанием для этого является представление элементов ПММ в виде подмножеств множества  $L$ . На рис. 4 приведена структура данных, которая позволяет решать задачи согласования данных в графе  $S$  и ПММ в процессе редактирования карты в среде ГИС. На основе этой модели поддерживается процесс автоматического обновления отдельных элементов матрицы  $D$  в информационной системе планирования грузоперевозок при редактировании графа дорожной сети.

Пользователь информационной системы, используя ГИС в качестве интерфейса, может редактировать исходные данные задачи, изменяя условия движения и добавляя или удаляя клиентов на карте. При этом он выполняет ограниченное число элементарных процедур редактирования (транзакций). Эти изменения в свою очередь приводят к необходимости внесения изменений в ПММ. Анализ влияния изменения исходных данных на карте на содержание ПММ представлен в таблице.

Задача актуализации ПММ сводится к внесению в нее изменений, состав которых определяется по данной таблице в процессе или после окончания пользователем сеанса редактирования транспортного графа территории. Данная модель позволяет однозначно выбирать элементы матрицы, которые необходимо изменить при выполнении пользова-

телем действий 1–7. Изменения для действий 1, 2 осуществляются практически мгновенно. Для действия 3 приходится пересчитать характеристики  $2n - 2$  маршрутов. Число маршрутов, которые требуют пересчета при действиях 4–7, зависит от топологии графа  $G$  и мест внесения правок. Это число находится в диапазоне от 0 до  $N$ , где  $N$  может быть эмпирически определено для конкретного состояния набора исходных данных и линейно зависит от  $|L|$  и  $n$ . Для перечисленных видов действий процедуры построения маршрутов и изменения матрицы запускаются автоматически после наступления соответствующих событий. Возможно также накопление заданий по синхронизации для выполнения их пакетом после окончания редактирования по команде пользователя.

При выполнении действий 8 или 9 однозначно определить, какие маршруты изменятся довольно трудно. При этом наиболее надежным вариантом является пере-

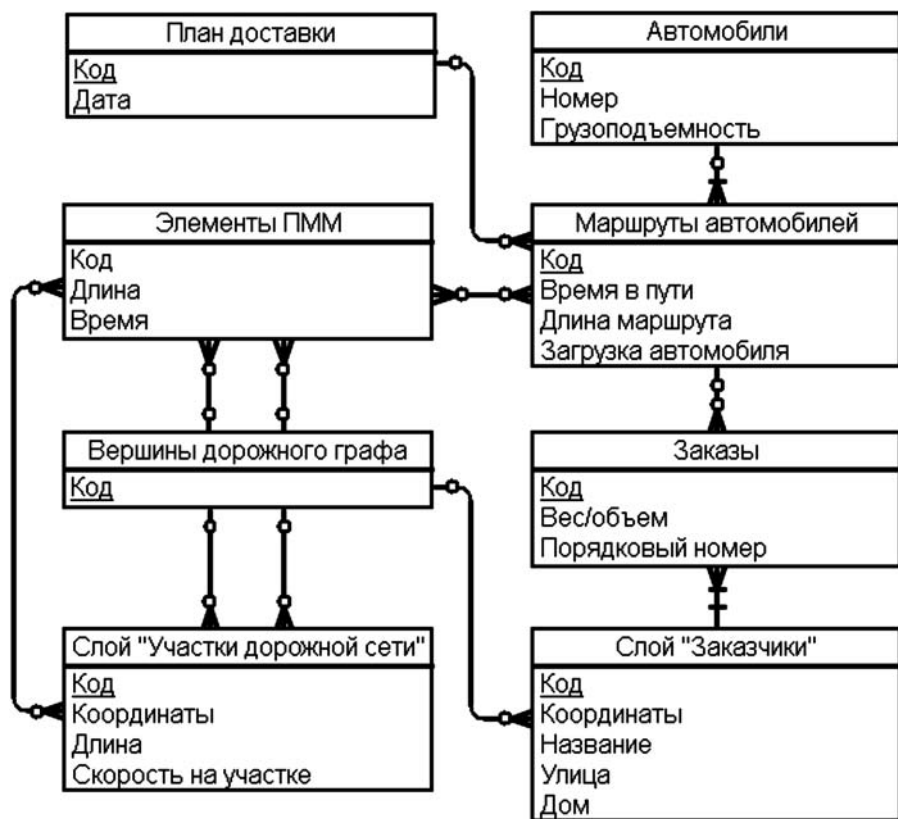


Рис. 4. Структура пространственных данных согласованного ведения модели дорожной сети и ПММ

№ действия	Действия пользователя в среде ГИС	Необходимые изменения в ПММ	Влияние отказа от учета изменений на расчет плана
1	Удаление клиента	Удаление строки и столбца матрицы	Не влияет
2	Разбивка участка дорожной сети на части	Не требуются	Не влияет
3	Добавление нового клиента	Добавление строки и столбца в матрицу	Невозможно выполнить расчет
4	Уменьшение скорости движения по участку дорожной сети	Пересчет маршрутов, проходящих по этому участку, и изменение соответствующих элементов матрицы	Доставка по рассчитанному плану будет осуществляться с опозданием по времени
5	Удаление участка дорожной сети		
6	Разрыв связи в графе дорожной сети		
7	Добавление терминального участка дорожной сети	Пересчет строк и столбцов в матрице для клиентов, к которым новый участок будет ближайшим	Доставка по рассчитанному плану может осуществляться с опережением по времени
8	Увеличение скорости движения по участку дорожной сети	Пересчет маршрутов, которые могут пройти по этому участку, и изменение соответствующих элементов матрицы	Доставка по рассчитанному плану будет осуществляться с опережением по времени
9	Добавление нового участка дорожной сети между двумя перекрестками		

счет всей ПММ. Однако последствия не учета этого действия не приводят к срыву рассчитанного плана, поскольку характеристики маршрутов при таких изменениях не могут ухудшиться. Поэтому изменения ПММ в этом случае можно отложить и выполнять при простое процессора компьютера в фоновом режиме, заменив матрицу по мере готовности ее новой версии.

Таким образом, использование модели данных, в которой элементы графа дорожной сети и матрицы транспортных затрат связаны друг с другом, позволяет автоматически определять степень несоответствия исходных данных оптимизационного расчета и данных об условиях движения, представленных на динамически обновляемой карте территории, и управлять процессом синхронизации этих данных. При этом изменение входных данных для расчетной модели в наиболее часто встречающихся на практике ситуациях, связанных с закрытием или ограничением движения на отдельных участках дорог, реализуется автоматически в режиме реального времени при задании новых характеристик движения на участках дорожной сети в среде ГИС.

### Заключение

Рассмотренный подход к представлению пространственных данных при решении задач транспортной логистики и размещения предприятий

может использоваться в объектно-ориентированных ГИС при хранении пространственных данных средствами СУБД. Он позволяет автоматизировать процесс синхронизации модели исходных данных, необходимой для решения задач дискретной оптимизации, и модели данных дорожной сети, представленной в виде слоев цифровой карты в среде ГИС. При этом обеспечивается рациональный режим загрузки компьютера с освобождением дополнительного времени для работы вычислительно сложных алгоритмов решения оптимизационных задач. Метод является универсальным для широкого класса задач дискретной оптимизации, связанных с использованием матриц транспортных затрат.

### Список литературы

1. **Babb T.** Pickup and Delivery Problem with Time Windows, Coordinated Transportation Systems: The State of the Art. Transportation, Florida, 2005. Vol. 54. Is. 1. P. 7—22.
2. **Potvin J.-Y., Xu Y., Benyahia I.** Vehicle routing and scheduling with dynamic travel times // Computer & Operations Research. 2006. N 33. P. 1129—1137.
3. **Береснев В. Л.** Дискретные задачи размещения и полиномы от булевых переменных. Новосибирск: Изд-во Ин-та математики, 2005. 408 с.
4. **Кормен Т. Х., Лейзерсон Ч. И., Ривест Р. Л., Штайн К.** Алгоритмы: построение и анализ. 2-е изд. М.: Вильямс, 2006. 1296 с.
5. **Косяков С. В., Гадалов А. Б., Жидовинов К. А.** Разработка информационной системы оптимального планирования грузоперевозок на базе ГИС-технологий // Вестник ИГЭУ. 2010. № 4. С. 101—105.

**В. Г. Астафуров**<sup>1, 2</sup>, д-р физ.-мат. наук, проф.,  
**А. В. Скороходов**<sup>1</sup>, аспирант,

<sup>1</sup> Учреждение Российской академии наук  
 Институт оптики атмосферы им. В. Е. Зуева  
 Сибирского отделения РАН, г. Томск,  
 e-mail: astafurov@iao.ru

<sup>2</sup> Томский государственный университет  
 систем управления и радиоэлектроники

## Нейросетевой классификатор облачности по спутниковым данным

*Предложен алгоритм классификации облачности по типам на основе трехслойного персептрона с использованием информации о текстуре спутниковых снимков. На основе экспертной оценки выделено 14 уникальных текстур различных типов облачности, которые могут быть определены предложенным классификатором. Для описания текстуры изображений облаков используется подход Gray-Level Co-occurrences Matrix, в основе которого лежит вычисление матриц смежности градаций яркости. Представлено описание архитектуры сети и алгоритма ее обучения. Обсуждаются результаты классификации облачности.*

**Ключевые слова:** классификация, облака, нейронные сети, текстурные признаки

### Введение

Данные дистанционного зондирования Земли из космоса используются для решения широкого круга задач, связанных с метеорологией, климатологией, переносом загрязнений, сельским и лесным хозяйствами. Для решения некоторых из них необходимы сведения о типах облачности, согласно принятому в метеорологии стандарту [1], на основании которого выделяют 10 основных типов облаков, а также их подтипы и сочетания (всего 27 фиксируемых на метеостанциях). Наиболее точные результаты по классификации облачности получены с применением многослойного персептрона [2, 3], вероятностной нейронной сети (НС) [4, 5] и алгоритма ближайшего соседа [6, 7]. Пороговые алгоритмы [8], статистические классификаторы [9] и самоорганизующиеся карты Кохонена [5] уступают им по точности.

Вероятностная НС обучается значительно быстрее многослойного персептропа, однако требует большего объема памяти ЭВМ, так как хранит ин-

формацию обо всей обучающей выборке [10]. По скорости выполнения процедуры классификации алгоритм ближайшего соседа превосходит многослойный персептрон и вероятностную сеть, так как в его основе используется простейшее решающее правило. Однако для получения сравнимых по точности результатов классификации облачности с помощью этого алгоритма необходимо иметь обучающую выборку из нескольких тысяч образцов [11]. Наиболее полные результаты по числу выделенных типов облаков получены в работах [4, 6, 7], где облачность разделялась по девяти основным типам и одному подтипу (мощные кучевые облака). Во всех упомянутых работах для оценки точности классификации использовали отношение числа правильно проклассифицированных тестовых фрагментов изображения, классификация которых была выполнена на основе экспертной оценки, к общему числу фрагментов.

Для классификации облаков на спутниковых снимках необходимо выбрать набор эталонных признаков. В работах [2, 5] для описания облачности применен текстурный подход, в работе [8] использован спектральный подход, а в некоторых из указанных выше работ — совместно текстурные признаки (ТП) и спектральные признаки. Цель данной работы — получение более полной классификации облачности по типам (включая их подтипы и различные сочетания) с помощью многослойного персептрона на основе информации о текстуре спутниковых снимков, определяемых по Gray-Level Co-occurrences Matrix (GLCM).

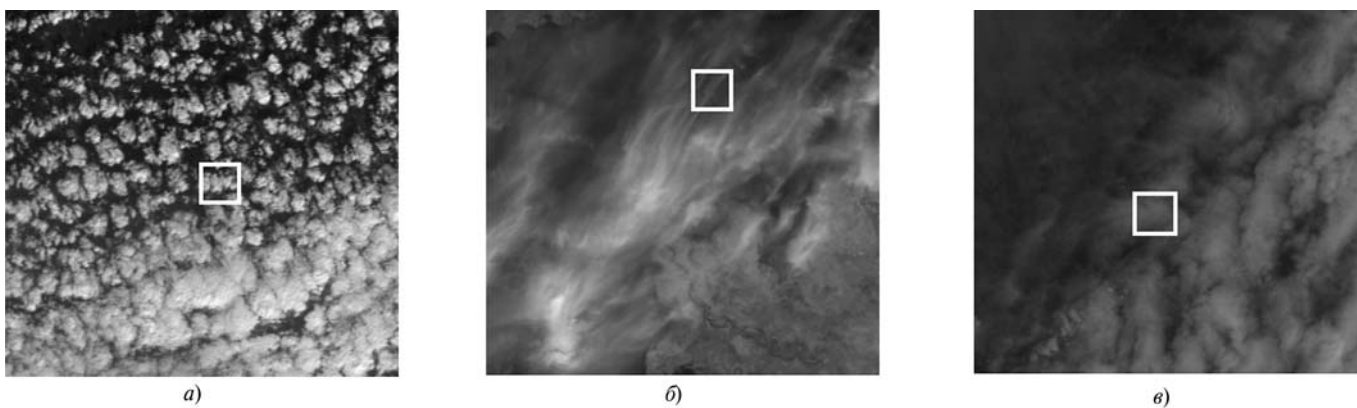
### Исходные данные

Для решения задачи классификации необходимы эталонные наборы значений ТП различных типов облачности. Поиск их характерных текстур осуществлялся путем сопоставления архивных данных метеостанций со снимками, выполненными системой MODIS. При этом использовалась информация об облаках с сервера "Погода России" (<http://meteo.infospace.ru>). Для интересующих нас данных метеостанций подбирались соответствующие снимки MODIS из архива на сервере Goddard Space Flight Center (<http://ladsweb.nascom.nasa.gov>). Интервал времени между фиксацией данных на метеостанции и спутниковой съемкой не превышает 10 мин. Выбор данных MODIS основан на их доступности, покрытии снимками всей территории Земли, продолжительности функционирования системы, начиная с 1999 года. Система MODIS сканирует земную поверхность с разрешением 250, 500 и 1000 м в различных спектральных интервалах (<http://modis.gsfc.nasa.gov>).

В результате проведенных исследований были сформированы три набора характерных изображе-

<sup>1</sup> Работа выполнена при поддержке гранта президента для молодых докторов наук МД-5302.2012.9.





**Рис. 1. Эталонные изображения облачности:**

*a* — слоисто-кучевые облака; *b* — перистые облака; *в* — высококучевые облака

ний текстур для 25 типов облачности, включая их подтипы и сочетания [1] по данным 1-го спектрального канала с разрешением 250, 500 и 1000 м. Согласно спецификации MODIS, первый канал используется для исследования облаков и земных объектов. Для каждого типа облачности найдено от 2 до 6 фрагментов снимков размером  $250 \times 250$  пикселей. Общее число фрагментов в одном наборе при фиксированном разрешении составляет 88. Примеры фрагментов изображений различных типов облачности показаны на рис. 1, где рамкой выделена область, в центре которой расположена метеостанция.

Вывод о степени сходства текстур различных типов облачности при трех пространственных разрешениях делался на основе экспертной оценки. С ее помощью было определено, что при 250 м можно выделить 14 типов облачности с отличающейся текстурой, при 500 м — 8 типов, при 1000 м — 6 типов облачности. Для дальнейших исследований использовались изображения с разрешением 250 м.

### Система текстурных признаков

Для формирования системы ТП использовался подход GLCM, основанный на вычислении матриц смежности  $P$  [12], которые характеризуют пространственное распределение и пространственную взаимосвязь значений яркости локальной области изображения. Элементы матрицы  $P_{i,j}$  рассчитываются по фрагменту изображения, выделенному скользящим окном, с размерами  $n_x$  и  $n_y$  пикселей по вертикали и горизонтали, соответственно. Подробное описание алгоритма формирования матриц смежности и определение различных ТП приведено в работе [12]. Матрица смежности зависит от определяемого парой соседних пикселей углового направления, от расстояния между ними и числа уровней квантования яркости. В работе [12] рассматриваются матрицы смежности для четырех угловых направлений  $\alpha = 0^\circ, 45^\circ, 90^\circ, 135^\circ$ , что позволяет учитывать различные ориентации текстуры.

Для определения набора информативных ТП использовали предложенную нами в работе [13] методику, с помощью которой были исследованы выбранные фрагменты изображений различных типов облачности.

1. На каждом фрагменте выбирали участок размером  $100 \times 100$  пикселей.

2. Участок обрабатывали скользящим окном размером  $20 \times 20$  пикселей и вычисляли набор средних значений ТП: энергия, энтропия, максимальная вероятность, однородность, контраст, дисперсия, момент обратной разности, суммарное среднее, суммарная дисперсия, суммарная энтропия, дифференциальная дисперсия, дифференциальная энтропия и корреляция, их рассчитывали по матрицам смежности для каждого из четырех угловых направлений; первый начальный момент и вариацию вычисляли по яркости пикселей для заданного положения скользящего окна.

3. Для каждого ТП определяли степень его относительной изменчивости и вычисляли оценки коэффициентов корреляции различных пар ТП. Информативным считался такой признак, который наиболее однороден и меньшее число раз "значимо" коррелирован с другими ТП.

4. После обработки всех эталонных фрагментов изображений по указанной в п. 3 методике был сформирован набор из 26 эффективных ТП: контраст, максимальная вероятность, дисперсия, суммарная дисперсия, дифференциальная дисперсия, дифференциальная энтропия (для четырех направлений), а также первый начальный момент и вариация.

Для классификации облачности использовали скользящее окно размером  $20 \times 20$  пикселей. Это обусловлено тем, что при увеличении окна от  $20 \times 20$  до  $80 \times 80$  пикселей значения ТП изменяются не более чем на 5–10% [14], а при малых размерах  $3 \times 3$  пикселей или  $5 \times 5$  пикселей ТП действуют как обнаружители перепадов яркости, а не как измерители текстуры. Аналогичный результат отмечен в работе [13], где показано, что при увеличении размера скользящего окна от  $20 \times 20$  до  $40 \times 40$  пикселей значения ТП практически не изменяются.

## Архитектура нейронной сети

Для классификации облачности используется трехслойный персептрон. Сложность областей в пространстве решений, которые могут быть реализованы трехслойной НС (два скрытых слоя и выходной слой), в принципе является неограниченной [11]. Число нейронов в выходном слое равняется числу типов облачности, по образцам которых обучается сеть. Число скрытых слоев и число нейронов в каждом скрытом слое задаются экспериментально, но существуют некоторые рекомендации. Максимальное число нейронов в каждом скрытом слое, согласно следствию из теоремы Колмогорова—Арнольда, равняется  $2Q + 1$ , где  $Q$  — размерность входного вектора признаков  $\mathbf{X}$ . При этом результаты классификации не всегда будут наилучшими, если число нейронов в скрытых слоях будет максимальным.

На рис. 2 показана архитектура НС, которая состоит из двух скрытых и одного выходного слоев. На основании численных экспериментов и анализа полученных результатов классификации облачности были выбраны следующие параметры сети: число нейронов в первом скрытом слое  $F = 2Q + 1 = 53$ , число нейронов выходного слоя  $L = 14$  равно числу типов облачности, а второй скрытый слой содержит  $S = (F + L)/2 \approx 34$  нейрона. На вход НС подается вектор признаков  $\mathbf{X} = (\hat{T}_1, \hat{T}_2, \hat{T}_3, \dots, \hat{T}_{26})^T$ ,

где  $\hat{T}_1, \hat{T}_2, \hat{T}_3, \dots, \hat{T}_{26}$  — масштабированные значения ТП. Применение процедуры масштабирования связано с тем, что значения ТП изменяются в различных диапазонах и их необходимо выравнять. Используется процедура линейного масштабирования [15]:

$$\hat{T}_i = \frac{T_i - T_i^{\min}}{T_i^{\max} - T_i^{\min}},$$

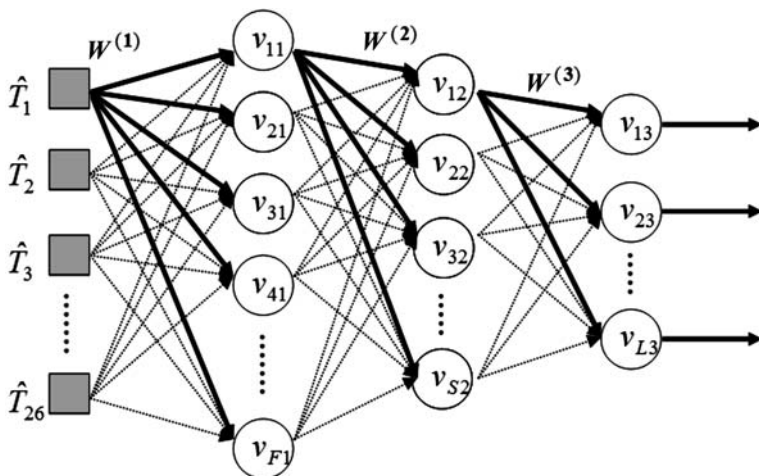


Рис. 2. Архитектура трехслойного персептрона

где  $T_i$  — текущее значение ТП;  $T_i^{\min}$  — минимальное значение  $i$ -го ТП;  $T_i^{\max}$  — максимальное значение  $i$ -го признака.

Функция активации имеет вид гиперболического тангенса:

$$G(v_{ij}) = \tanh(\lambda v_{ij}),$$

где  $v_{ij}$  — суммарный сигнал, поступающий на вход  $i$ -го нейрона в  $j$ -м слое при  $j = 1, 2, 3$ ;  $\lambda$  — коэффициент, определяющий пологость функции.

Выходом НС являются сведения о принадлежности обучающего или классифицируемого фрагмента изображения к какому-либо типу облачности. Учитывая, что функция гиперболического тангенса принимает значения из интервала  $(-1; 1)$ , эти сведения кодируются в следующем виде [10, 15]:

$(1, -1, -1, \dots, -1)^T \rightarrow$  тип облачности № 1

$(-1, 1, -1, \dots, -1)^T \rightarrow$  тип облачности № 2

$(-1, -1, 1, \dots, -1)^T \rightarrow$  тип облачности № 3

.....

$(-1, -1, -1, \dots, 1)^T \rightarrow$  тип облачности № L.

Нейронная сеть обучается алгоритмом обратного распространения ошибки, который минимизирует целевую функцию

$$E(\mathbf{W}) = \frac{1}{2} \sum_{k=1}^L (y_k - d_k)^2,$$

где  $\mathbf{y} = [y_1, y_2, \dots, y_L]^T$  — текущий отклик сети на входной сигнал;  $y_i = G(v_{i3})$ ,  $i = \overline{1, L}$ ,  $\mathbf{d} = [d_1, d_2, \dots, d_L]^T$  — вектор ожидаемых выходных сигналов сети.

После предъявления каждого образца происходит коррекция весовых коэффициентов нейронов всех слоев, начиная с выходного, по формуле

$$\mathbf{W}(l+1) = \mathbf{W}(l) + \Delta\mathbf{W}(l),$$

где  $\Delta\mathbf{W}(l) = \eta(l)\mathbf{p}(\mathbf{W})$  — значение коррекции весовых коэффициентов;  $\eta(l)$  — коэффициент обучения на шаге  $l$ ;  $\mathbf{p}(\mathbf{W})$  — направление поиска минимума функции  $E(\mathbf{W})$  в многомерном пространстве  $\mathbf{W}$ . Направление поиска минимума  $E(\mathbf{W})$  определяется выражением

$$\mathbf{p}(\mathbf{W}) = -\nabla E(\mathbf{W}).$$

Скорость обучения НС зависит от выбора коэффициентов  $\eta$  и  $\lambda$ . Коэффициент скорости обучения  $\eta$  можно задавать следующими способами [10, 16]:

1. Подбирать  $\eta_f$  эмпирически и фиксировать на протяжении всего периода обучения для всех слоев.

2. В процессе обучения уменьшать от заданного начального значения  $\eta_0$  до  $\eta_{\min}$  с постоянным шагом  $\Delta\eta$  для всех слоев.

3. Подбирать эмпирически и фиксировать на протяжении всего периода обучения для каждого слоя  $\eta_f^{(1)}$ ,  $\eta_f^{(2)}$ ,  $\eta_f^{(3)}$  соответственно.

4. В процессе обучения уменьшать от заданных значений  $\eta_0^{(1)}$ ,  $\eta_0^{(2)}$ ,  $\eta_0^{(3)}$  до  $\eta_{\min}^{(1)}$ ,  $\eta_{\min}^{(2)}$ ,  $\eta_{\min}^{(3)}$  с постоянным шагом  $\Delta\eta^{(1)}$ ,  $\Delta\eta^{(2)}$ ,  $\Delta\eta^{(3)}$  для каждого слоя соответственно.

5. Задавать обратно пропорционально квадратному корню из суммы числа синаптических связей нейрона.

Выбор наиболее эффективного способа задания коэффициента скорости обучения рассмотрен на примере трехслойной НС с параметрами:  $F = 53$ ,  $S = 27$  и  $L = 2$ . Сеть обучалась двум типам текстуры облачности (кучевой и слоистой) по 20 образцам для каждой. Скорость обучения зависит от выбора начальной точки поиска минимума. Учитывая, что эта точка выбирается случайно, алгоритм обучения НС запускался по 20 раз при каждом способе задания коэффициента  $\eta$ . В таблице приведено усредненное время обучения НС.

Качество результатов классификации при различных способах задания  $\eta$  оценивалось по следующим критериям:

- совпадение результатов классификации с данными метеостанций;
- однородность выделенных областей облачности;
- отсутствие разрывов внутри выделенных областей.

Было определено, что наилучшие результаты классификации получаются при способах задания 1, 3 и 4 из таблицы. В результате был выбран четвертый способ задания  $\eta$ , при котором НС имеет наибольшую скорость обучения. Следуя рекомендациям, приведенным в работе [10], выбрано значение коэффициента пологости  $\lambda = \frac{2}{3}$ .

Время обучения НС при различных способах задания  $\eta$

№ способа	Способ задания $\eta$	Среднее время обучения, с
1	$\eta_f = 0,01$	246
2	$\eta_0 = 0,5$ , $\eta_{\min} = 0,01$ , $\Delta\eta = 0,001$	10
3	$\eta_f^{(1)} = 0,03$ , $\eta_f^{(2)} = 0,02$ , $\eta_f^{(3)} = 0,01$	201
4	$\eta_0^{(1)} = 0,3$ , $\eta_{\min}^{(1)} = 0,03$ , $\Delta\eta^{(1)} = 0,001$ $\eta_0^{(2)} = 0,2$ , $\eta_{\min}^{(2)} = 0,02$ , $\Delta\eta^{(2)} = 0,001$ $\eta_0^{(3)} = 0,1$ , $\eta_{\min}^{(3)} = 0,01$ , $\Delta\eta^{(3)} = 0,001$	37
5	$\eta_f^{(1)} = 1/\sqrt{53}$ , $\eta_f^{(2)} = 1/\sqrt{27}$ , $\eta_f^{(3)} = 1/\sqrt{3}$	25

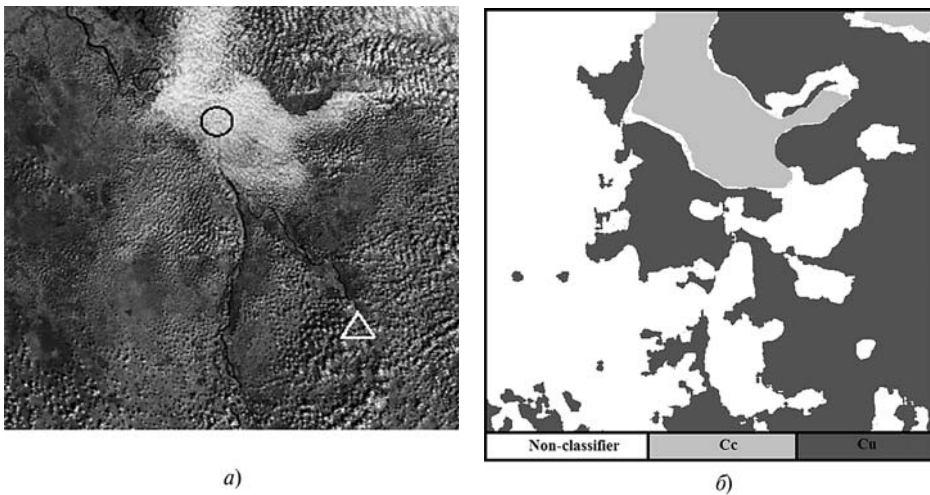
Начальные веса НС случайным образом инициализируются из интервала  $(-0,3; 0,3)$  [10, 15]. Обучение происходит по эпохам с помощью модифицированного алгоритма обратного распространения ошибки. Эпохой считается предъявление сети числа образцов, равного общему числу образцов всех типов облачности. В процессе обучения образцы предъявляются сети в случайном порядке (стохастическое обучение). После предъявления каждого образца облачности происходит подстройка весовых коэффициентов от выходного слоя к входу, что соответствует последовательному режиму обучения. Обучение считается законченным, если в рамках всей эпохи после предъявления каждого образца на соответствующем его типу облачности выходе значение отклика будет находиться в интервале  $(0,9; 1,0)$ , а на других выходах значения отклика будут в интервале  $(-1; -0,9)$ .

Веса настроенной НС можно сохранить для последующей классификации спутниковых изображений. Процедура классификации аналогична процедуре обучения. После предъявления классифицируемого фрагмента определяется его принадлежность к одному из типов облачности в соответствии с откликом сети. В итоге центральный пиксель классифицируемого фрагмента перекрашивается в заранее определенный цвет в соответствии с выбранной легендой. Если НС не может определить принадлежность классифицируемого фрагмента к какому-либо типу облаков, то цвет пикселя изменится на белый.

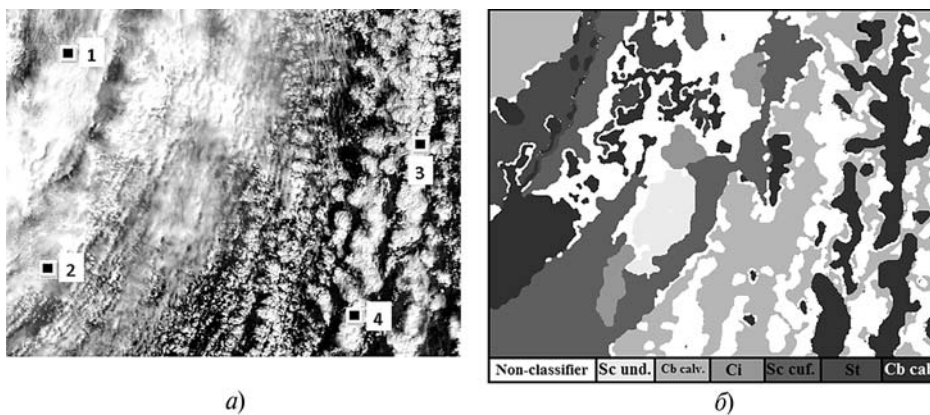
При построении НС были проведены исследования по выбору числа уровней квантования исходной матрицы яркости. НС обучалась по характерным образцам облачности при различных числах уровней квантования (4, 8, 16, 32, 64, 128, 255). Далее проводилась классификация типов облачности на одном и том же снимке. Выбранное изображение охватывает территорию с несколькими метеостанциями, над которыми известен тип облачности. По результатам исследования можно сделать вывод, что наилучший результат классификации соответствовал числу уровней квантования, находящемуся в интервале 16...32. В дальнейшем для классификации облачности использовалось число уровней квантования, равное 20.

### Результаты классификации и их обсуждение

На рис. 3, *а* представлен фрагмент снимка Томской области размером  $158 \times 147$  км от 4 августа 2003 г. НС была обучена 14 типам облачности. Результатом классификации является изображение на рис. 3, *б*. Согласно данным метеостанций Томска (выделено треугольником на рис. 3, *а*) и Красного Яра (выделено кругом), в этих районах фиксировалась кучевая и перисто-кучевая облачность. Белым цветом обозначена непроклассифицированная территория. В результате визуальной оценки снимка



**Рис. 3. Результат классификации облаков по типам:**  
*a* — исходное изображение; *б* — проклассифицированное изображение



**Рис. 4. Результат классификации облаков по типам:**  
*a* — снимок территории Томской области; *б* — проклассифицированное изображение

можно заметить, что проклассифицированные области имеют однородную структуру с незначительным числом разрывов. Наиболее однородно (без разрывов) была выделена перисто-кучевая облачность. К непроклассифицированным областям относятся земля, реки и границы разделов между разными типами облачности.

На рис. 4, *a* показан фрагмент снимка Томской области размером  $280 \times 214$  км от 30 апреля 2010 г. Черными метками на снимке обозначены метеостанции региона: 1 — Колпашево, 2 — Бакчар, 3 — Первомайское, 4 — Томск. По данным метеостанций в этот момент времени в Бакчаре, Первомайском и Томске наблюдались кучево-дождевые волосатые облака, а в Колпашево кучево-дождевые лысые. На рис. 4, *б* показан результат классификации облачности по типам. Участки изображения рядом с метками 2, 3 и 4 были отнесены к кучево-дождевым волосатым облакам, о чем свидетельствует выделение этих областей одним цветом.

Фрагмент снимка рядом с меткой 1 был правильно проклассифицирован, так как результат совпадает с данными наземных наблюдений. Другие выделенные участки облачности относятся к перистой, слоистой, слоисто-кучевой (кучевообразной и волнистообразной). Однако ввиду отсутствия метеостанций в этих областях результаты классификации не могут быть проверены. Наличие множества разрывов в выделенных областях на рис. 4, *б* связано с неоднородностью поля облачности, показанного на рис. 4, *a*.

Оценка точности результатов классификации облачности выполнялась на основе методики их сопоставления с данными наземных метеостанций. В качестве тестового участка была выбрана территория Томской области, ограниченная четырьмя метеостанциями из предыдущего примера. Далее были отображены изображения за период 2008—2011 гг. с мая по сентябрь, когда одновременно на трех или четырех метеостанциях фиксировалась только однослойная облачность. Оценка точности, которая была определена процентом совпадений результатов класси-

фикации облачности на основе выбранных снимков с данными метеостанций, составила 91%.

## Заключение

Результаты исследований подтверждают эффективность использования трехслойного перцептрона и информации о текстуре изображений для классификации облачности по типам на основе спутниковых данных. В ходе исследований было определено 14 типов уникальных текстур различных типов облачности, которые могут быть идентифицированы по снимкам MODIS с разрешением 250 м. С помощью изображений облачности разрешением 250 м, размера скользящего окна  $20 \times 20$  пикселей и числа уровней квантования яркости изображения, равного 20, был сформирован набор из 26 информативных ТП, который включает: максимальную вероятность, первый начальный момент, вариацию, контраст, дисперсию, суммарную дисперсию,

дифференциальную дисперсию и дифференциальную энтропию.

На основе набора информативных ТП и определенного количества уникальных текстур изображений облачности с разрешением 250 м была сформирована структура НС с числом нейронов в первом скрытом слое  $F = 53$ , во втором слое  $S = 34$  и в выходном слое  $L = 14$ . Достоверность алгоритма классификации облачности по 10 основным типам облаков, а также подтипам кучево-дождевой (лысье и волосатые облака), слоисто-кучевой (кучевообразные и волнистообразные) и высококучевой (кучевообразные и волнистообразные) облачности и сочетанию кучевых и слоисто-кучевых облаков, наблюдаемых одновременно, составила 91%. Полученные результаты классификации превосходят достигнутые в работах [3, 6, 7].

*Работа выполнена при частичной финансовой поддержке Минобрнауки РФ (госконтракт № 02.740.11.0674).*

#### Список литературы

1. **Облака** и облачная атмосфера: справочник / Под ред. Мазина И. П., Хргиана А. Х. Л.: Гидрометеиздат, 1989. 647 с.
2. **Lee J., Weger R. C., Sengupta S. K., Welch R. M.** A neural-network approach to cloud classification // IEEE Trans. Geosci. Remote Sensing, 1990. Vol. 28, no. 5. P. 846—855.
3. **Bankert R. L., Rabindra P., Sengupta S. K.** A probabilistic neural network approach to cloud classification: Miss // Tech. Note of the Naval Oceanographic and Atmospheric Research Laboratory 39529—5004, 1992. 35 p.
4. **Miller S. W., Emery W. J.** An automatic neural-network cloud classifier for use over land and ocean surface // J. Appl. Meteorol., 1997. Vol. 36. P. 1346—1362.
5. **Tian B., Shaikh M. A., Azimi-Sadjadi M. R., Vonder Haar T. H., Reinke D. L.** A study of cloud classification with neural networks using spectral and texture features // IEEE Trans. Neural Networks, 1999. Vol. 10. P. 138—151.
6. **Bankert R. L., Wade R. H.** Optimization of an instance-based GOES cloud classification algorithm // J. Appl. Meteor. Climatol. 2007. Vol. 46. P. 36—49.
7. **Bankert R. L., Mitrescu C., Miller S. W., Wade R. H.** Comparison of GOES cloud classification algorithms employing explicit and implicit physics // J. Appl. Meteor. Climatol. 2009. Vol. 48. P. 1411—1421.
8. **Волкова Е. В., Успенский А. Б.** Детектирование облачности и определение ее параметров по спутниковым данным в светлое время суток // Метеорология и гидрология. 2007. №12. С. 5—20.
9. **Uddstrom M. J., Gray W. R.** Satellite cloud classification and rain-rate estimation using multispectral radiances and measures of spatial texture // J. Appl. Meteor. 1996. Vol. 35. P. 839—858.
10. **Хайкин С.** Нейронные сети: Полный курс / Пер. с англ. Н. Н. Кузсуль, А. Ю. Шелестовой. М.: Вильямс, 2008. 1103 с.
11. **Гонсалес Р., Вудс Р.** Цифровая обработка изображений. М.: Техносфера, 2005. 1072 с.
12. **Haralick R. M., Bosley R.** Texture features for image classification // Proc. Third ERTS Symp. 1973. P. 1929—1969.
13. **Астафуров В. Г., Скороходов А. В.** Сегментация спутниковых снимков облачности по текстурным признакам на основе нейросетевых технологий // Исследование Земли из космоса. 2011. №6. С. 10—20.
14. **Потапов А. А.** Фракталы в радиофизике и радиолокации: Топология выборки. М.: Университетская книга, 2005. 848 с.
15. **Каллан Р.** Основные концепции нейронных сетей / Пер. с англ. М.: Вильямс, 2001. 287 с.
16. **Осовский С.** Нейронные сети для обработки информации / Пер. с польс. И. Д. Рудинского. М.: Финансы и статистика, 2002. 344 с.

## Информация

### 10<sup>th</sup> IEEE EAST-WEST DESIGN & TEST SYMPOSIUM (EWDTS 2012)

Харьков, Украина, 14—17 сентября 2012

Симпозиум проводится Харьковским национальным университетом радиозлектроники и Академией наук прикладной радиозлектроники <http://anpre.org.ua/>

Цель симпозиума **IEEE East-West Design & Test Symposium (EWDTS)** — расширение международного сотрудничества и обмен опытом между ведущими учеными Западной и Восточной Европы, Северной Америки и других стран в области автоматизации проектирования, тестирования и верификации электронных компонентов и систем. Симпозиум проводится, как правило, в странах бассейнов Черного и Балтийского морей, Центральной Азии.

Оргкомитет приглашает ученых, аспирантов и студентов принять участие в работе международного симпозиума EWDTS'12

**Регистрация докладов:** <http://www.ewdtest.com/conf>

**Адрес оргкомитета:** Проф. Владимир Хаханов, кафедра Автоматизации проектирования вычислительной техники Харьковского национального университета радиозлектроники, пр. Ленина 14, Харьков, 61166, Украина.

**Тел.: +380-57-702-13-26, E-mail: [hahanov@kture.kharkov.ua](mailto:hahanov@kture.kharkov.ua), [www.ewdtest.com/conf/](http://www.ewdtest.com/conf/)**

УДК 519.178

Д. Ю. Кривошеин, аспирант,  
А. М. Марченко, д-р техн. наук, проф.,  
МГУ им. М. В. Ломоносова

## Инкрементальный алгоритм поиска кратчайших путей в графе

*Описан алгоритм решения задачи поиска кратчайших путей в динамически изменяющемся графе. Для статических графов известен ряд алгоритмов, например, алгоритм Флойда—Уоршелла или алгоритм Дейкстры. Рассматривается случай, когда вес какого-либо ребра в графе изменяется, и ставится задача быстро пересчитать все кратчайшие пути. Такая задача может возникнуть, например, на этапе трассировки сверхбольших интегральных схем (СБИС), где графы могут иметь большую размерность и классические алгоритмы малоэффективны. Приведен инкрементальный алгоритм нахождения всех кратчайших путей в случае, если вес ребра увеличивается. Также рассмотрен случай графа с малым числом ребер, при этом оценка сложности инкрементального алгоритма составила  $O(n^{4/3})$ .*

**Ключевые слова:** динамически изменяющийся граф, поиск кратчайших путей, инкрементальный алгоритм, алгоритм Флойда—Уоршелла, трассировка СБИС

### Введение

Поиск кратчайших путей на графе является хорошо известной задачей, для решения которой существует несколько алгоритмов. Наиболее часто используется алгоритм Флойда—Уоршелла для нахождения кратчайших путей между всеми парами вершин. Этот алгоритм имеет сложность  $O(V^3)$ , где  $V$  — число вершин графа. Алгоритм Дейкстры находит кратчайшие пути от заданной вершины до всех остальных и имеет сложность  $O(V^2)$  в наихудшем случае. Для разреженных графов существуют модификации этих алгоритмов сложности  $O(V^2 \log V)$  и  $O(V \log V)$  соответственно [1].

Указанные алгоритмы могут быть эффективно использованы для решения прикладных задач, если граф не изменяется во времени. Тогда достаточно один раз вычислить матрицу кратчайших путей. Для задач, в которых граф изменяется, например меняются веса ребер, каждое локальное изменение одного ребра может привести к изменению некоторых путей в данном графе. В таких задачах необходимо пересчитывать кратчайшие пути многократно, что делает применение алгоритма Флойда—

Уоршелла практически невозможным ввиду слишком большого времени работы (например, на графе с 5000 вершин время его работы доходит до 8 ч).

Таким образом, задача разработки алгоритма пересчета кратчайших путей при динамическом изменении исходного графа является актуальной задачей, для которой к настоящему времени не предложено более эффективных алгоритмов, чем алгоритм Флойда—Уоршелла.

### 1. Обзор существующих алгоритмов

В работе [2] рассматривается задача планирования маршрутов движения самолетов над Европой. Авторами был предложен новый алгоритм, который является инкрементальной модификацией алгоритма Флойда—Уоршелла. Алгоритм был протестирован на случайных планарных графах и показал лучшие результаты, чем базовый алгоритм, в случае большого количества изменений в графе. Однако при этом алгоритм использует в  $|V|$  раз больше памяти, чем алгоритм Флойда—Уоршелла, вследствие хранения большого числа матриц.

Также был проведен эксперимент на реальной задаче планирования маршрутов самолетов, и в этом случае время работы нового алгоритма оказалось на 20 % больше, чем у алгоритма Флойда—Уоршелла, вследствие большого числа обращений к памяти, которые фактически и занимали основное время.

В работе [3] рассмотрена проблема динамического контроля простых временных сетей. Задача сводится к задаче нахождения кратчайших путей в графе, при этом веса ребер графа динамически изменяются.

Основная сложность алгоритма приходится на преобразование временной сети в граф, при этом используется алгоритм сложности  $O(V^3)$ , а также на начальное вычисление кратчайших путей между всеми вершинами в графе с помощью алгоритма Беллмана—Форда сложности  $O(VE)$ , где  $E$  — число ребер. После построения графа задачи и подсчета начальных значений кратчайших путей при ситуации, когда вес ребра изменяется, пересчет кратчайших путей выполняется с помощью особой процедуры, однако в статье для нее не приводятся теоретические оценки сложности. Таким образом, эффективность применения предложенного алгоритма не превосходит известные.

Задача нахождения кратчайших путей также возникает на этапе трассировки СБИС. В настоящее время для пересчета путей в графе трассиров-

ки используется алгоритм Дейкстры, например в системе MARS [4]. Этот алгоритм необходимо запускать многократно, что приводит к большим временным затратам.

В итоге можно сделать вывод, что задача инкрементального поиска кратчайших путей в графе изучена недостаточно, а разработанные алгоритмы имеют достаточно высокую сложность и не дают выигрыша по сравнению с классическими алгоритмами.

## 2. Инкрементальный алгоритм вычисления кратчайших путей

Пусть дан связный ориентированный граф  $G = (V, E)$ , каждому ребру  $(u, v) \in E$  приписан вес  $w(u, v) \geq 0$ . Предполагается, что уже известны расстояния между всеми вершинами (например, вычислены по алгоритму Флойда—Уоршелла), и вес некоторого ребра  $(u, v)$  в графе изменяется на  $\delta > 0$ .

**Определение 2.1.** Пусть кратчайший путь  $P$  между вершинами  $v'$  и  $v''$  в графе  $G$  проходит через ребро  $(u, v)$  и имеет вид  $v'P_1(u, v)P_2v''$ , где  $P_1, P_2$  — некоторые пути. Тогда вершины  $v'$  и  $v''$  называются соответственно *входящей* и *выходящей* вершинами для ребра  $(u, v)$ .

Пусть  $Inc$  и  $Out$  — множества входящих и выходящих вершин соответственно.

**Утверждение 2.1.** При пересчете кратчайших путей в графе достаточно рассмотреть только пути между вершинами из множества  $Inc \cup Out$ .

**Доказательство.** Пусть кратчайший путь  $P$  в графе  $G$  между вершинами  $v', v'' \notin Inc \cup Out$  имеет длину  $L$ . При добавлении ребра  $(u, v)$  в путь  $P$  его длина увеличится как минимум на  $\delta$  и станет больше минимальной, поэтому на вершины, не принадлежащие множеству  $Inc \cup Out$ , увеличение веса ребра  $(u, v)$  не влияет.

Пусть теперь  $P'$  — путь между вершинами  $v_i \in Inc, v_o \in Out$  длины  $L'$ . При увеличении веса ребра  $(u, v)$  длина пути  $P'$ , проходящего через это ребро, станет равной  $L' + \delta$ , но при этом в графе  $G$  может оказаться путь меньшей длины, поэтому требуется пересчитать кратчайший путь между вершинами  $v_i$  и  $v_o$ . Утверждение доказано. ■

Инкрементальный алгоритм  $IncrAPSP$  вычисления кратчайших путей в графе, основанный на утверждении 2.1, состоит из следующих шагов:

1) из вершины  $u$  поиском в ширину находятся все вершины, принадлежащие множеству  $Inc$ ; аналогично из вершины  $v$  находятся все вершины, принадлежащие множеству  $Out$ ;

2) для каждой вершины из множества  $Inc$  с помощью алгоритма Дейкстры вычисляются значения кратчайших путей до множества  $Out$ ;

3) каждой вершине из множества  $Out$  приписывается значение длины кратчайшего пути до каждой вершины из множества  $Inc$ , полученное на шаге 2.

В результате вычисляются новые значения кратчайших путей между всеми вершинами множества  $Inc \cup Out$ , остальные пути не требуют пересчета, согласно утверждению 2.1. Таким образом, алгоритм корректно пересчитывает все кратчайшие пути в графе.

**Утверждение 2.2.** Сложность инкрементального алгоритма поиска кратчайших путей в графе не превосходит  $O(|Inc| \cdot L_D)$ , где  $L_D$  — сложность алгоритма Дейкстры.

**Доказательство.** Достаточно оценить сложность каждого шага описанного алгоритма:

1) построение множеств  $Inc$  и  $Out$  осуществляется с помощью алгоритма поиска в ширину, который в худшем случае имеет сложность  $\Theta(|V| + |E|)$  [1];

2) поиск кратчайших путей выполняется  $|Inc|$  раз по алгоритму Дейкстры, при простейшей реализации  $L_D = O(|V|^2)$  [1];

3) изменения значений кратчайших путей для вершин из множества  $Out$ , сложность шага  $O(|Inc| \cdot |Out|)$ .

Общая оценка для сложности инкрементального алгоритма имеет вид:

$$\begin{aligned} L(IncrAPSP) &\leq \\ &\leq \Theta(|V| + |E|) + |Inc| \cdot O(|V|^2) + O(|Inc| \cdot |Out|), \\ L(IncrAPSP) &\leq |Inc| \cdot O(|V|^2). \end{aligned}$$

Утверждение доказано. ■

**Замечание 2.1.** В случае, если граф разреженный, а именно, если  $|E| = o(|V|^2/\log|V|)$ , алгоритм Дейкстры может быть модифицирован таким образом, что его сложность составит  $L_D = O(|V|\log|V| + |E|)$  [1].

**Замечание 2.2.** В худшем случае сложность инкрементального алгоритма будет составлять  $O(|V|^3)$ , что возможно, если мощность множества входящих вершин асимптотически равна мощности множества всех вершин, и граф при этом не удовлетворяет условию разреженности из замечания 2.1.

## 3. Оценка сложности алгоритма для разреженных графов

В этом разделе будут рассматриваться графы с малым числом ребер. В частности, при решении задачи трассировки СБИС возникают графы решетчатой структуры, такие, что из каждой вершины выходит не более 12 ребер во всех направлениях (по два ребра на каждое, так как граф ориентированный). Используя структуру графа, в этом случае можно получить лучшую оценку, чем в общем случае.

**Замечание.** Граф решетчатой структуры является разреженным.

Пусть  $S$  — ортогональная сетка и точка  $c \in S$  является вершиной  $u$  изменившегося ребра. Тогда верно следующее утверждение о мощности множества  $Inc \cup Out$ .

**Утверждение 3.1.** Для ортогональной сетки  $S$  верно, что

$$|Inc \cup Out| \leq \sqrt{Crit(S, c)},$$

где  $Crit(S, c)$  — число кратчайших путей на сетке  $S$ , проходящих через точку  $c$ .

**Доказательство.** Пусть на сетке расположено  $m$  вершин. Если рассмотреть пути между всеми этими вершинами как элементы матрицы размера  $m \times m$ , то их число будет составлять  $m^2$ . Таким образом, можно сделать вывод, что число вершин, кратчайшие пути между которыми проходят через точку  $c$ , выражается как корень квадратный из числа путей между этими точками. Так как определение множеств  $Inc$  и  $Out$  вводилось для ребра, а не для точки, то из всех путей нужно оставить только пути, проходящие также и через вершину  $v$ , которая на сетке отображается некоторой точкой  $d$ . Утверждение доказано. ■

Пусть  $P_c(a, b)$  — вероятность того, что кратчайший путь между двумя точками сетки  $a$  и  $b$  пройдет через случайную точку  $c$ ,  $P_c(a, b) \leq 1, \forall a, b, c \in S$ .

Очевидно, что  $Crit(S, c) = \sum_{i \in S} \sum_{j \in S} P_c(i, j)$ .

Пусть  $G = (V, E)$  — разреженный граф,  $|V| = n$ , тогда  $|E| \leq 6n$ . Не ограничивая общности, такой граф можно представить в виде параллелепипеда  $\Pi$  с размерами  $L \times H \times D$ , тогда  $|V| = LHD = n$ . Аналогично случаю на плоскости оценка числа кратчайших путей в графе  $G$ , проходящих через вершину  $u$ , имеет вид:

$$Crit(G, u) = \sum_{i \in \Pi} \sum_{j \in \Pi} \sum_{k \in \Pi} P_u(i, j, k).$$

**Утверждение 3.2.** Для графа  $G$  верно, что

$$|Inc \cup Out| \leq \sqrt[3]{Crit(G, u)}.$$

Доказательство проводится аналогично утверждению 3.1, но пути рассматриваются как элементы трехмерной матрицы, что дает в результате корень третьей степени. ■

**Утверждение 3.3.** Для разреженных графов верна следующая оценка сложности инкрементального алгоритма:

$$L(IncrAPSP) \leq O(n^{4/3}).$$

**Доказательство.** Очевидно, что

$$\begin{aligned} Crit(G, u) &= \sum_{i \in \Pi} \sum_{j \in \Pi} \sum_{k \in \Pi} P_u(i, j, k) \leq \\ &\leq \sum_{i \in \Pi} \sum_{j \in \Pi} \sum_{k \in \Pi} 1 = LHD = n. \end{aligned}$$

Следовательно, мощность множества  $Inc$  выражается через число вершин в графе следующим образом:

$$|Inc| \leq |Inc \cup Out| \leq \sqrt[3]{Crit(G, u)} \leq \sqrt[3]{n}$$

(аналогичная оценка верна и для множества  $Out$ ).

Используя утверждение 2.2, а также замечание 2.1 о сложности алгоритма Дейкстры для разреженных графов, можно получить:

$$\begin{aligned} L(IncrAPSP) &\leq |Inc| \cdot L_D \leq \sqrt[3]{n} (|Out| \log |Out| + |E|) \leq \\ &\leq \sqrt[3]{n} (\sqrt[3]{n} \log \sqrt[3]{n} + 6n) = \\ &= \frac{2}{3} n^{2/3} \log n + 6n^{4/3} = O(n^{4/3}). \end{aligned}$$

Утверждение доказано. ■

Таким образом, сложность инкрементального алгоритма пересчета кратчайших путей в разреженном графе равна  $O(n^{4/3})$ .

#### 4. Экспериментальные результаты

Инкрементальный алгоритм был реализован на языке C++, проведено тестирование на случайных графах и сравнение с алгоритмом Флойда—Уоршелла поиска всех кратчайших путей. Временная сложность алгоритмов представлена на рис. 1. Аналогичные эксперименты были проведены на разреженных графах (рис. 2). Измерялось среднее и максимальное время выполнения на множестве случайных графов. Результаты подтвердили теоретические оценки.

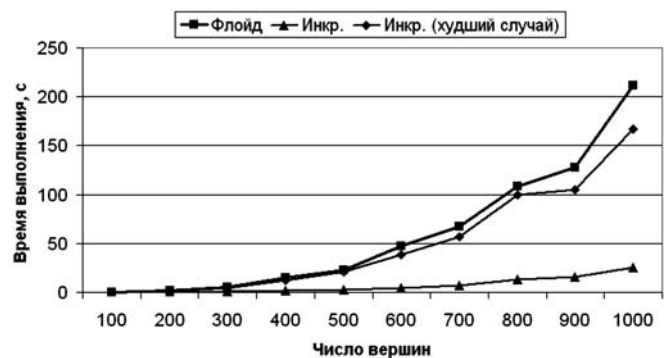


Рис. 1. Время работы алгоритмов на случайных графах

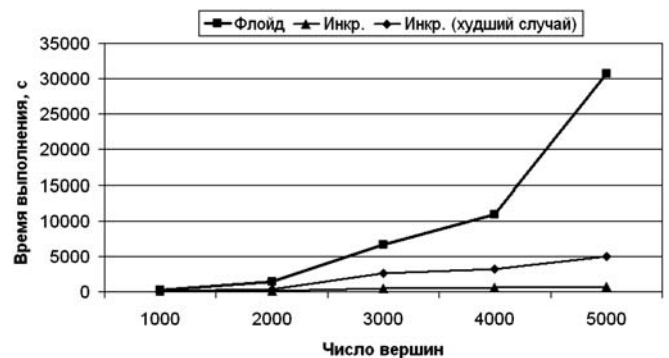


Рис. 2. Время работы алгоритмов на случайных разреженных графах



## Заключение

В статье описан алгоритм пересчета всех кратчайших путей в графе, если вес какого-либо ребра увеличивается. Показано, что сложность зависит от структуры графа, в худшем случае алгоритм имеет сложность  $O(n^3)$ . Для разреженных графов была получена оценка сложности  $O(n^{4/3})$ . Алгоритм был программно реализован, проведены эксперименты на случайных графах (в том числе разреженных), которые подтвердили теоретическую оценку, а также проведено сравнение с известными алгоритмами. Предложенный алгоритм может быть использован для решения различных прикладных за-

дач, например при трассировке сверхбольших интегральных схем.

## Список литературы

1. Cormen T. H., Leiserson C. L., Rivest R. L., and Stein C. Introduction to Algorithms. Cambridge, Massachusetts: The MIT Press, 2001.
2. Rivière T., Brisset P. Shortest path in planar graph and air route network // Proc. of 4th EUROCONTROL Innovative Research Workshop & Exhibition. Brétigny-sur-Orge, 2005. P. 306–315.
3. Stedl J., Williams B. A Fast Incremental Dynamic Controlability Algorithm // Proc. of the ICAPS Workshop on Plan Execution. Monterey, CA, 2005.
4. Cong J., Fang J., Xie M., and Zhang Y. MARS — A Multi-level Full-Chip Gridless Routing System // Proc. of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. Los Angeles, CA. 2005. Vol. 24, N 3. P. 382–394.

---

---

## Информация

19–21 ноября 2012 года в ИПУ РАН (г. Москва), состоится

### 7-я международная научно-практическая конференция "УПРАВЛЕНИЕ ИННОВАЦИЯМИ — 2012"

На конференции предполагается обсудить теоретические основы  
и практические проблемы управления инновациями:

- Инновации и новое качество экономического роста.
- Научно-техническая информация как хозяйственный ресурс и как фактор производства.
- Управление технологической структурой производства на предприятии.
- Макроэкономические предпосылки инновационных процессов.
- Инновации и цикличность экономической динамики.
- Человеческий капитал, его формирование и использование.
- Институциональные аспекты стимулирования инновационных процессов.
- Национальные инновационные системы.
- Инновационные процессы в экономике регионов.
- Моделирование и прогнозирование инновационных процессов.
- Инновации и макроэкономическая политика.
- Стратегия инновационного развития России.

В рамках конференции пройдут Тринадцатые Друкеровские чтения  
"Современные стратегии инновационного развития".

Все контакты между Оргкомитетом и участниками осуществляются  
по электронной почте: [InnovConf@mail.ru](mailto:InnovConf@mail.ru).

Подробную информацию о конференции см. на сайте:  
<http://www.ipu.ru/node/15205>

# АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ

УДК 004.5

**П. Н. Мартынов**, инж.,  
**Е. Е. Ковшов**, д-р техн. наук, проф.,  
Государственный технологический университет  
"СТАНКИН", г. Москва,  
e-mail: oxumo@mail.ru

## Разработка средств автоматизации тестирования интерфейсов пользователя в человеко-машинных системах управления

*Рассматриваются методы исследования качества интерфейсов программных систем и их применимость в области человеко-машинных систем управления. В качестве альтернативы существующим методикам тестирования интерфейсов предлагается использовать авторский подход и разработанный на его основе программный комплекс.*

**Ключевые слова:** человеко-машинное взаимодействие, графический интерфейс, юзабилити-тестирование

### Введение

С тех самых пор, как человечество стало применять различные машины и приспособления, позволяющие автоматизировать производство, возникла необходимость в управлении этими машинами, человеко-машинном взаимодействии и, соответственно, в человеко-машинных интерфейсах. Изначально применяли механические элементы управления, но с повсеместным появлением электроники стало возможным создание более сложных в функциональном плане и в то же время компактных производственных объектов.

В связи с ростом производственных возможностей и усложнением технологических процессов происходило изменение интерфейсов человеко-машинных систем управления (ЧМСУ).

Сегодня промышленные системы управления — это в основном программные приложения, наглядно и интерактивно отображающие прохождение технологического процесса и позволяющие его контролировать.

Вместе с тем, подобные возможности систем управления порождают значительный информаци-

онный поток между человеком и машиной. И при таком взаимодействии оператор системы управления технологическим оборудованием является "узким местом", так как его возможности по оперативной обработке данных психологически и физически лимитированы. Таким образом, создание интерфейсов систем управления, ориентированных, в первую очередь, на возможности и ограничения человека, является особо важным фактором обеспечения безопасности и эффективности работы производства в целом.

Отдельной статистики о влиянии интерфейсов автоматизированных систем управления технологическими процессами (АСУТП) на число аварий не ведется, однако при расследовании происшествий [1] в десятку причин во многих случаях попадает формулировка "ошибочные действия и несвоевременное реагирование оперативного и диспетчерского персонала", появление которой в том числе зависит и от несовершенства интерфейсов ЧМСУ. При этом, к сожалению, в списках мероприятий по предупреждению последующих аварийных ситуаций редко содержатся работы над АСУТП. На рис. 1 (см. четвертую сторону обложки) изображены примеры интерфейсов оператора, перегруженные информацией.

Удобный и продуманный интерфейс ЧМСУ не просто несет в себе конкурентное преимущество для разработчика компьютерных систем как в случае прикладного программного обеспечения из других областей, а является жизненно важным элементом системы управления, который должен обеспечивать:

- получение оператором только необходимой информации о ходе процесса;
- использование минимального числа элементов интерфейса для осуществления оперативного управления;
- максимальное привлечение внимания оператора при возникновении аварийной ситуации, особенно в условиях монотонного труда;
- минимизацию действий и помощь оператору ЧМСУ при попытке устранения аварийной ситуации.

Таким образом, задача создания качественных интерфейсов, ориентированных на возможности человека, является немногим менее важной, чем создание работоспособной и отказоустойчивой ЧМСУ в целом.

## Качество интерфейсов АСУТП и методики тестирования

Создание интерфейса практически любого программного приложения — процесс, состоящий из большого числа итераций, которые включают разработку, прототипирование, исследование, анализ и доработку версий интерфейса. При этом уровень качества интерфейса можно определить только на основании исследований или тестов. В настоящее время оценка качества программных систем в целом регламентируется стандартом ГОСТ Р ИСО/МЭК 9126—93 [2], который выделяет шесть основных характеристик. Из них к уровню качества интерфейсов компьютерных приложений непосредственно относятся только две характеристики:

- *практичность (usability)* — оценивается через набор таких подхарактеристик, как понятность программного обеспечения, легкость в освоении и простота использования;
- *эффективность (efficiencies)* — оценивается через набор подхарактеристик, относящихся ко времени отклика, скорости выполнения функций, объему и продолжительности использования аппаратных ресурсов.

Для определения данных характеристик качества в настоящее время регламентированы и используются следующие виды исследований программных интерфейсов [3]:

**Прототипирование.** Заключается в создании нескольких вариантов интерфейса и последующей сравнительной оценки вариантов, в том числе посредством других методов исследования. Так как метод находится в тесной взаимосвязи с другими видами тестирования интерфейсов (*GOMS*, фокус-группы, экспертная оценка), его чаще относят к этапам разработки.

**Анализ задач.** Как видно из названия, исследование строится исходя, во-первых, из задач, которые пытается выполнять пользователь посредством тестируемого интерфейса и, во-вторых, из того, насколько эффективно пользователь выполняет задачу, поставленную перед ним руководителем тестирования. Тестирование проводится с участием нескольких респондентов (потенциальных пользователей системы), и основными его инструментами являются анкетирование и измерение временных характеристик выполнения заданий в исследуемом интерфейсе. На основе полученных данных анализируется продуктивность работы пользователей, проводится оценка общей удовлетворенности интерфейсом и составляется список приоритетных проблем.

**Фокус-группы** [4]. Суть процесса такого тестирования заключается в наблюдении за действиями пользователей во время их взаимодействия с интерактивной компьютерной системой при выполнении ими реальных задач. Во время исследования

испытываемые пользователи под присмотром специалистов выполняют определенные тестовые сценарии в исследуемой компьютерной системе. Все действия, а также мимика респондентов, и в зависимости от типа тестирования голосовые комментарии, фиксируются на видео. Затем специалисты проводят анализ графической, аудио- и видеoinформации и на этой основе выносят решение об уровне практичности (*usability*) продукта.

**Метод GOMS (Goals, Operators, Methods and Selection rules) (Цели, Операторы, Методы и Правила выбора)** [5]. Данная методика позволяет моделировать выполнение задачи пользователем и на основе полученной модели оценить время выполнения, характеризующее скорость работы пользователя, и, соответственно, такие параметры, как эффективность и практичность интерфейса. Основная идея метода заключается в том, что любые задачи, выполняемые пользователем на человеко-машинном интерфейсе, можно представить в виде последовательностей типовых составляющих действий, для которых можно получить среднестатистические значения времени выполнения и определить время, которое в среднем будет затрачено на выполнение заданий пользователем. Минимальное время, затраченное на выполнение задачи, является основным критерием выбора того или иного варианта интерфейса пользователя.

**Экспертная оценка.** Как следует из названия метода, оценку программного интерфейса проводит группа экспертов. Специалисты проводят исследование, опираясь на собственный профессионализм, опыт разработки и проектирования, а также на наборы существующих эвристических правил и рекомендаций, сформированных за годы изучения психологических, физических и технических аспектов человеко-машинного взаимодействия. По результатам тестирования создается список мероприятий, нацеленных на улучшение интерфейса, или заключение о соответствии интерфейса существующим нормам и правилам.

Рассмотренные способы исследования интерфейсов применимы для любых компьютерных систем, однако наиболее широко их используют компании, создающие прикладные программные приложения для непроизводственных отраслей человеческой деятельности. Это обусловлено тем, что АСУТП имеют ряд отличий от привычных и повсеместных пользовательских приложений: системы управления находятся в динамике, взаимодействие с оператором и предоставление информации происходит в реальном времени, они могут сами вызывать события во время прохождения технологического процесса, а не только ожидать действия оператора. Поэтому и исследования интерфейсов ЧМСУ нельзя проводить без изменений и доработки технологий тестирования.

## Формирование критериев тестирования для ЧМСУ

Помимо отличий ЧМСУ от компьютерных приложений для персональных ЭВМ, существуют и другие факторы, ограничивающие возможности применения описанных выше методик в промышленной отрасли. Большинство парадигм исследований и оценки интерфейсов программных приложений оперируют качественными, а не количественными параметрами. То есть по итогам тестирования получается набор данных, которые не могут дать жесткую характеристику объекта исследования, и по которым нельзя определить наилучший по совокупности параметров вариант интерфейса. Полученные данные не позволяют проводить моделирование интерфейса (за исключением метода *GOMS*), а могут быть лишь интерпретированы и проанализированы специалистами по человеко-машинному взаимодействию.

Таким образом, по результатам тестов формируется субъективная оценка интерфейса и субъективные рекомендации, правильность которых в значительной степени зависит от компетентности и опыта специалиста по проектированию интерфейсов. А так как отрасль информационных систем для АСУТП в мире представлена значительно меньшим числом компаний, чем отрасль прикладных программных приложений в целом, то и специалистов, имеющих навыки и опыт исследования интерфейсов ЧМСУ, найти сложнее.

Качественные результаты исследования порождают и другое слабое место существующих методик. Визуальная и аудиоинформация, произвольные анкетные данные и экспертный анализ — это информационный материал, сбор и обработку которого в данном контексте практически невозможно автоматизировать. То есть помимо трудоемкого анализа результатов исследования, специалистам необходимо выполнять также все рутинные этапы, что не лучшим образом сказывается на производительности, сроках и стоимости тестирования.

В связи с вышеизложенными факторами подходы к оценке качества интерфейсов были тщательно изучены. В результате анализа преимуществ и недостатков современных методик исследования и специфики предметной области были определены критерии оптимальной парадигмы тестирования качества интерфейсов оператора ЧМСУ.

- Для того чтобы избежать субъективной интерпретации данных исследований, необходимо выполнять фиксацию количественных характеристик интерфейса.
- При взаимодействии оператора и системы управления важно, чтобы оператор выполнял свои действия за кратчайшее время (особенно в условиях аварийной ситуации), поэтому при тестировании интерфейса ЧМСУ значимыми показателями являются время выполнения задания и скорость работы оператора.

- Исследования интерфейса необходимо проводить с участием потенциальных пользователей (например, надо учесть, что метод *GOMS* и экспертная оценка применяются без участия респондентов), так как именно человеко-машинные взаимодействия операторов АСУТП в конечном итоге определяют эффективность системы.
- Необходим сравнительный анализ и количественная оценка цветовых параметров интерфейсов ЧМСУ, так как в условиях монотонной работы выбор цветового обозначения различных состояний системы влияет на скорость реакции и усталость оператора.
- Желательна оптимизация и автоматизация этапов сбора, обработки и систематизации данных тестирований в целях уменьшения нагрузки специалистов на рутинных этапах исследования. Помимо описанных критериев предпочтительной является возможность моделирования интерфейса и взаимодействий на основе данных исследования, а также возможность математического обоснования ошибок интерфейса и путей их устранения.

### Описание выбранной методики тестирования

Исходя из составленных критериев, применение существующих подходов к исследованию интерфейсов программных систем для тестирования интерфейсов ЧМСУ без комбинирования и модификаций нежелательно. Поэтому была сформулирована и описана оригинальная методика исследования интерфейсов компьютерных информационных систем, соответствующая большей части приведенных критериев [6].

В основе предлагаемого подхода лежит применение программной системы, осуществляющей фиксацию и сохранение данных о действиях оператора в тестируемом интерфейсе АСУТП. Система регистрирует количественные характеристики взаимодействий человека и машины (временные и координатные данные, а также число активаций оператором каждого элемента экранной формы) и затем преобразует их в понятную специалистам визуальную модель.

Научно-практической основой методики является, фактически, синтез преимуществ методов фокус-групп и *GOMS* с возможностью проведения удаленного тестирования. Задачи оператора, которые он решает с помощью программы, разбиваются на составляющие действия, но в отличие от метода *GOMS* разбиение проводится не аналитически, а в процессе непосредственного взаимодействия с интерфейсом исследуемой системы, за счет фиксации всех атомарных действий человека.

В рамках предложенной методики принимается, что оператор может совершать только два типа элементарных действий: непосредственную активацию элемента управления (нажатие клавиши

## Разработка программного комплекса

мыши или клавиатуры, нажатие на сенсорную панель в области действия элемента и т. п.) и перемещение между различными элементами управления (перемещение указателя мыши, смена фокуса посредством клавиатуры и т. п.). Так как в процессе разработки графических интерфейсов расположение любых элементов задается в локальной системе координат экранной формы, то каждый элемент интерфейса содержит информацию о своем расположении на плоскости  $\{x, y\}$  и размерные характеристики  $\{l, h\}$ , что позволяет определять точные координаты каждого элемента интерфейса, активированного оператором системы. Фиксация действий человека-оператора подразумевает также запись временных  $\Delta t$  параметров каждого зарегистрированного взаимодействия. Перемещения пользователя по экранной форме характеризуются не координатами, а расстоянием между последовательно задействованными элементами управления, т. е. "путем"  $\Delta s$ . На основании этого вычисляются скоростные характеристики действий оператора:  $\Delta v = \Delta s / \Delta t$ .

Представление взаимодействий человека и АСУТП двумя различными типами позволяет использовать в предложенной методике математическую модель ориентированного взвешенного графа для отображения количественных характеристик действий пользователей на исследуемом интерфейсе. Взаимодействия, характеризующие перемещения между элементами, представляются в виде дуг, а элементы экранной формы — вершинами. Такая модель, во-первых, позволяет проводить более наглядный анализ, а, во-вторых, применять для обработки данных исследования существующий математический аппарат алгоритмов на графах.

Таким образом, в предложенной методике тестирование интерфейса аналогично методу фокус-групп и проводится людьми, непосредственно работающими с системой, что позволяет получать в результате исследования наиболее достоверные данные. В рамках методики проводится сбор количественных параметров человеко-машинных взаимодействий, характеризующих оперативность и удобство работы оператора в интерфейсе исследуемой программной системы. Количественная информация и графическое представление обеспечивают возможность автоматизации этапов сбора экспериментальных данных, их обработки, построения математической модели и частичного анализа графа. Кроме того, возможно дополнение и комбинирование предложенной методики с другими методиками, существующими для проведения разностороннего исследования. А для оценки цветовых характеристик интерфейсов АСУТП на мнемосхемах возможно применение алгоритмов, предлагаемых другими исследователями человеко-машинных интерфейсов [7].

На основе выбранной методики исследования была выполнена разработка программного комплекса, позволяющего частично автоматизировать процесс тестирования интерфейсов ЧМСУ. Целевой программно-аппаратной платформой разработки была выбрана платформа *Wintel*, как наиболее распространенная на сегодняшний день, в том числе для промышленных *SCADA* и *ERP*-систем.

Архитектура созданной программной системы представляет собой комбинацию самостоятельных функциональных блоков (программных агентов), соответствующих этапам сбора, подготовки, хранения информации, построения математической модели и предварительного анализа модели. Как представлено на рис. 2 (см. четвертую сторону обложки), блок сбора данных функционирует на рабочих местах пользователей исследуемой ЧМСУ и пересылает структурированные отчеты о взаимодействиях на сервер разработчиков интерфейса.

Блок обработки информации, сервер реляционной базы данных запущены на серверах разработчика в круглосуточном режиме. Приложение, осуществляющее визуализацию модели графа, функционирует на рабочих местах специалистов по проектированию интерфейсов и получает параметры моделирования из базы данных на сервере разработчика. Благодаря такому выбору распределенной архитектуры, программный комплекс легко внедряется на предприятии (рабочие места операторов) и не предъявляет особых требований к своей эксплуатации. Модуль сбора данных фиксирует взаимодействия оператора с программой, сохраняя качественные (тип действия, объект) и количественные (время, координаты) параметры в структурированный текстовый файл, и осуществляет передачу накопленных данных модулю подготовки информации. Исполнение данного модуля может быть в виде двух вариантов: модуль встраивается в тестируемое программное приложение (добавление должно проводиться на этапе разработки исследуемой информационной системы, что не всегда является возможным), или модуль функционирует на тестируемом рабочем месте в качестве отдельного программного приложения.

В первом варианте сбор данных осуществляется через обработку событий, связанных с элементами управления основной программы. Во втором случае модуль отслеживает сообщения на уровне операционной системы, вызванные аппаратными устройствами ввода.

Блок обработки данных исследования выполняет систематизацию информации и передает ее блоку хранения, который добавляет полученную информацию в сетевую реляционную базу данных. Информация из базы данных по запросу поступает в блок построения графа. Этот модуль осуществляет визуализацию математической модели и является

рабочим инструментом специалиста-аналитика, проводящего тестирование интерфейса. С помощью блока визуализации возможно просматривать модель взаимодействий за определенные промежутки времени, процесс формирования графа в динамике и перемещения пользователя по тестируемой экранной форме (аналог видеоданных, полученных в результате проведения фокус-групп). Модуль предварительного анализа модели входит в состав блока построения и позволяет выполнять автоматизированную оценку графа. С помощью генетических алгоритмов модуль выявляет наборы действий, состоящие из максимального числа составляющих и имеющие наибольшие веса. На основе этих данных составляется список предварительных рекомендаций по исправлению интерфейса [8].

Распределенная архитектура системы позволяет организовывать очень гибкое асинхронное взаимодействие между модулями. Программная реализация методики выполнена в соответствии с современными тенденциями и стандартами разработки программного обеспечения. Схема работы исследовательской системы предусматривает круглосуточное функционирование и сбор данных тестирования в любое время и на любом рабочем месте, оснащенном доступом в глобальную сеть Интернет.

### Заключение

Предложено применение методик исследования интерфейсов ЧМСУ в качестве мероприятий, позволяющих оценить программные системы с точки зрения удобства и эффективности взаимодействия с ними оператора, как фактора обеспечения техносферной безопасности. Сформулированы критерии применения подобных методик в области промышленных компьютерных систем, и предложен оригинальный подход к исследованию человеко-машинных взаимодействий. На данном этапе развития созданный подход позволяет снизить влияние человеческого фактора в процессе

исследования и повысить объективность результатов исследования.

На основе предложенной методики разработан программный комплекс, автоматизирующий работу с данными исследования и построение модели взаимодействий. Визуализация взаимодействий оператора и ЧМСУ в виде ориентированного графа предусматривает представление данных для специалиста как в статике, так и в динамике. Количественные характеристики модели позволяют проводить предварительный автоматизированный анализ и статистическую обработку данных исследования интерфейса.

### Список литературы

1. **Аварии:** причины и следствия [Электронный ресурс]. — Электрон. дан. — [М.]: Safeprom, cop. 2006—2011. — URL: <http://www.safeprom.ru/articles/history.php>
2. **ГОСТ Р ИСО/МЭК 9126—93.** Оценка программной продукции. Характеристики качества и руководства по их применению. М.: Госстандарт России: Изд-во стандартов, 1994. — II, 9 с.
3. **Пономарев И. А.** Методы оценки качества пользовательского интерфейса // Интеллектуальные технологии и системы. Сб. учебно-методических работ и статей аспирантов и студентов. Вып. 6 / Сост. и ред. Ю. Н. Филипповича. — М.: Эликс+, 2004. 334 с.
4. **Юзабилити:** правила, психология, термины. Электронные данные. — [М.]: Design For Masters, cop. 2007—2008. — URL: <http://designformasters.info/posts/usability-rules-psychology-terms/>
5. **GOMS.** Electronic data. Wikimedia Foundation, cop. 2011. — URL: <http://en.wikipedia.org/wiki/GOMS>
6. **Мартынов П. Н.** Оценка удобства работы в системе "человек—ЭВМ" на основе статистических моделей // Прикладная информатика и математическое моделирование: межвузовский сб. науч. трудов. — М.: МГУП, 2009. — С. 164—168.
7. **Сухов А. А.** Обзор методов создания и оценки эффективности человеко-машинного интерфейса тренажеров управления опасным технологическим процессом // Научные труды XII международной научно-практической конференции "Фундаментальные и прикладные проблемы приборостроения, информатики и экономики. Информатика". — М.: МГУПИ, 2009. — С. 152—156.
8. **Борисенко Е. В., Мартынов П. Н.** Применение генетического алгоритма при анализе модели взаимодействия пользователя с интерактивным программным обеспечением // Проблемы полиграфии и издательского дела, 2010. № 5. С. 91—97.

---

### Новая книга

**И. И. Абрамов.**

Лекции по моделированию элементов интегральных схем микроэлектроники.

LAP LAMBERT Academic Publishing, Saarbrücken, Germany. 2012. 116 с.

В учебном пособии дан единый взгляд на построение моделей элементов кремниевых интегральных схем (ИС) микроэлектроники на основе фундаментальной системы уравнений физики полупроводниковых приборов. Приведены классификация моделей и подходы к их синтезу. Описаны основные используемые на практике модели и рассмотрены методы идентификации их параметров. Приведены сведения о программном обеспечении моделирования элементов и фрагментов ИС, а также дана его классификация. Курс лекций по данному предмету читается автором в Белорусском государственном университете информатики и радиоэлектроники, начиная с 1994 года, и отражает его опыт работы в этой области с 1975 года. При этом в лекции был включен лишь "золотой фонд" дисциплины. Пособие предназначается для студентов старших курсов соответствующих специальностей и может быть полезно аспирантам, магистрантам и инженерам, использующим модели ИС в своих исследованиях.

С книгой можно ознакомиться и заказать, обратившись на сайт <https://www.ljubluknigi.ru/>

**С. А. Бражник**, инженер-программист,  
 ЗАО ИТФ "Системы и технологии", аспирант,  
 Владимирский государственный университет,

**С. И. Малафеев**,

д-р техн. наук, проф., гл. науч. сотрудник,  
 ООО "Компания "Объединенная Энергия",  
 г. Москва,  
 e-mail: sim\_vl@newmail.ru

## Синхронизация в промышленных контроллерах с операционной системой LINUX

*Рассматривается использование программной системы синхронизации в промышленных контроллерах с операционной системой Linux в составе АСКУЭ. Источником точного времени служат приемники GPS. Сигналы синхронизации используются для коррекции времени и подстройки хода системных часов контроллеров с погрешностью, не превышающей 1,3 мкс.*

**Ключевые слова:** АСУ ТП, АСКУЭ, Linux, время, контроллер, синхронизация

### Введение

В современных Автоматизированных системах контроля и учета энергоресурсов (АСКУЭ) и АСУ ТП с большим числом компонентов для устранения неопределенности измерений по времени используются специальные системы обеспечения единого времени. В соответствии с Федеральным законом "О связи" № 126 от 7 июля 2003 года, Статья 49 — "Учетно-отчетное время в области связи", в технологических процессах передачи и приема сообщений электросвязи, их обработки в пределах территории Российской Федерации операторами электросвязи и операторами почтовой связи должно применяться единое учетно-отчетное время — московское. Для этого все операторы должны организовывать системы точного времени — комплекс технических средств, обеспечивающих периодическую передачу цифровой информации о значении текущего времени от эталонного источника ко всем сетевым элементам в целях синхронизации их внутренних часов. Применительно к цифровому оборудованию сетей электросвязи, в котором происходит обработка различных данных в режиме реального времени, должно обеспечиваться одновременное выполнение определенных внутренних технологических процессов [1].

Серверы единого времени обычно оснащаются приемниками сигналов GPS (или ГЛОНАСС) для синхронизации собственных часов по глобальному времени UTC (Universal Time Coordinated). Для обеспечения повышенной надежности серверы единого времени обычно имеют высокоточные встроенные часы, которые также могут использоваться для формирования сигналов синхронизации. Эти сигналы передаются в различных форматах по проводным линиям, оптоволоконным линиям связи или с использованием локальной сети Ethernet (NTP-сервер) [2].

В работе рассматривается программная реализация коррекции внутренних часов (сервера синхронизации времени TimeSynchro) для АСКУЭ, использующих цифровые электроизмерительные приборы [3].

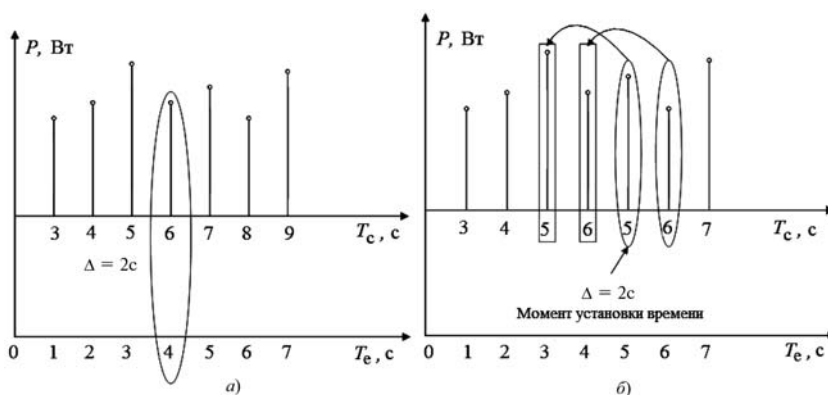
### Алгоритмы синхронизации внутренних часов

При нарушении синхронизации в АСКУЭ возможно использование двух основных способов коррекции внутренних часов:

- жесткий перевод времени;
- плавная коррекция.

Реализация первого способа осуществляется с помощью импульсной системы фазовой автоподстройки частоты [4]. Рассмотрим процесс измерения расхода электрической энергии с использованием цифрового счетчика с шагом дискретизации 1 с результатов измерения активной электрической мощности  $P$  (рис. 1). Допустим, что в момент времени  $t = 4$  с обнаружено опережение внутренних часов ( $T_c$ ) по отношению к точному времени ( $T_e$ ) на 2 с. При жестком переводе внутренних часов (рис. 2) происходит потеря данных одного отсчета ( $t = 5$  с).

В случае отставания внутренних часов жесткий перевод времени не приведет к потере данных. Из приведенного примера следует, что жесткий перевод внутренних часов приводит к искажению данных в системе. Это особенно важно при высокой частоте передачи данных.



**Рис. 1. Жесткий перевод времени: системные часы ( $T_c$ ) опережают мировое время ( $T_e$ ):**

*a* — до коррекции; *б* — после коррекции

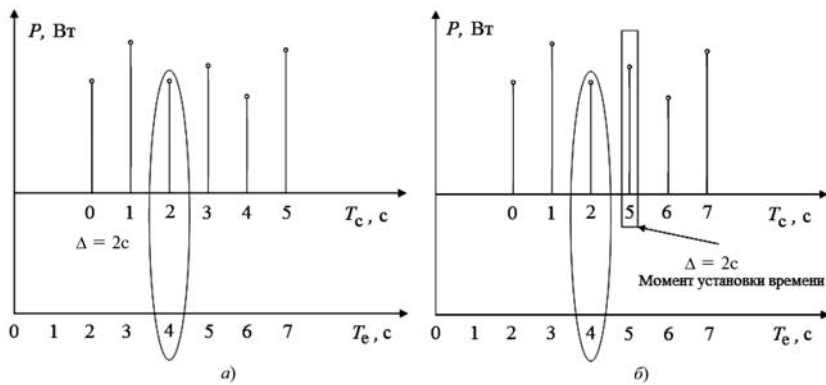


Рис. 2. Жесткий перевод времени: системные часы ( $T_c$ ) отстают от мирового времени ( $T_e$ ):

*a* — до коррекции; *б* — после коррекции

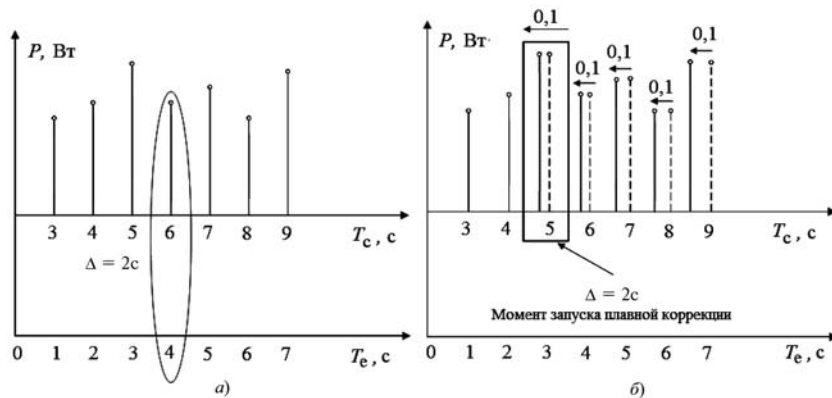


Рис. 3. Плавная коррекция: системные часы опережают мировое время:

*a* — до коррекции; *б* — после коррекции

Плавная коррекция внутренних часов основана на постепенном уменьшении расхождения времени за определенный период (рис. 3). В результате во всех компонентах системы поддерживается единое время. Процедура коррекции регистрируется в журнале и при необходимости формируется сообщение о нарушении хода внутренних часов в конкретном устройстве.

### Коррекция внутренних часов при использовании системы Linux

Коррекция внутренних аппаратных часов LinPAC выполняется через системные вызовы ОС Linux. Для плавной подстройки времени в Linux используются две основные системные функции [5]:

- `adjtime(const struct timeval *delta, struct timeval *olddelta);`
- `adjtimex(struct timex *buf).`

#### 1. Функции `adjtime()`:

```
#define_BSD_SOURCE
#include <sys/time.h>
int adjtime(const struct timeval * delta,
struct timeval * olddelta);
```

Задача этой функции — медленная коррекция времени путем добавления или вычитания (в зависимости от того, спешат или отстают часы) некоторого значения от системного времени за каждый определенный промежуток времени.

Назначение положительного или отрицательного значения аргументу `delta` обеспечивает для ядра ОС прибавление или вычитание этого значения от текущего системного времени за определенный интервал времени.

Если задать значения аргумента `delta` равным `NULL`, а в `olddelta` передать структуру `timeval`, то в `olddelta` вернется остаток текущей процедуры коррекции, если таковая была запущена.

#### 2. Функции `adjtimex()`

```
#include <sys/time.h>
int adjtimex(struct timex * adj);
```

Эта функция предоставляет совершенно новый интерфейс для программного управления системными часами контроллера.

Задача этой функции — замедлить или увеличить скорость хода системных часов в зависимости от переданных ей параметров. Если изначально в ядре Linux существовала только функция `adjtime`, которая постепенно добавляла малое значение рассогласования времени и за определенный период удаляла расхождение полностью, то теперь у разработчиков систем синхронизации появился новый, очень действенный инструмент для коррекции времени в АСУ ТП и АСКУЭ.

Скорость замедления или увеличения зависит от параметров структуры `timex`, которую мы передаем в функцию.

Рассмотрим эту структуру подробнее.

```
struct timex {
    int modes; // режим работы
    long offset; // сдвиг в микросекундах
    long freq; // сдвиг частоты
    long maxerror; // максимальная ошибка
    long esterror; // расчетная ошибка
    int status; // статус часов
    long constant; // константа времени
    long precision; // точность часов
    long tolerance; // доступное отклонение частоты
    struct timeval time; // текущее системное время
    long tick; // тики часов
};
```

Чтобы провести коррекцию системного времени с помощью этой функции, нужно правильно рассчитать коэффициенты `tick` и `freq` и полю `modes` назначить константные значения `ADJ_TICK` и `ADJ_FREQUENCY`.

#### Алгоритм расчета коэффициентов `freq` и `tick`

##### Исходные данные:

1. `Ticks_per_day = 10000` [тик/сутки];  
Часы осуществляют 10000 тиков в сутки. Следовательно, 1 тик будет равен 8,64 с.



1. [Tick] = 8,64 [с];
2. 1 [f] = 0.000001318359375 [с];  
65536 [f] = 0.0864 [с];
3. USER\_HZ = 100 прерываний в секунду  
(900000/USER\_HZ) < Tick\_range <  
< (1100000/USER\_HZ)  
(9000) < Tick\_range < (11000)

При таком диапазоне тиков максимальное значение дельта может быть DELTA\_MAX = (+\ -)\*8600 с.

Выполним оценку точности алгоритма.

Поле tick позволяет регулировать расхождения системного и мирового времени более чем 8,64 с, что является достаточно грубой коррекцией.

Поле freq предназначено для того, чтобы обеспечивать значения коррекции, меньшие 8,64 с. Поле freq не может принимать вещественное значение, поэтому можно оценить точность алгоритма, предоставляемого функцией **adjtimex**.

Если 1 [f] = 0,000001318359375 [с], то

$$1 [f] = 1,31 [\text{мкс}].$$

Следовательно, алгоритм обладает точностью (с погрешностью не более 1,31 мкс), что является очень хорошей оценкой его работы.

#### Порядок расчета коэффициентов.

1. Определим, не превышает ли дельта константу DELTA\_MAX. Должно выполняться условие

$$[DELTA\_MAX > \text{abs}(\text{delta})].$$

2. Найдем число тиков, необходимых для коррекции:

$$T = \text{mod}(\text{delta}/\text{tick}) = \text{mod}(\text{delta}/8,64).$$

3. Вычислим итоговое число тиков, которое будет подставлено в структуру timex:

$$\text{Ticks} = (\text{Ticks\_per\_day} - T) = (10000 - T).$$

4. Проверим, попадает ли число тиков в диапазон допустимых значений:

$$(9000) < \text{Tick\_range} < (11000).$$

5. Вычислим частоту для более точной подстройки:

$$F = [\text{delta} - (T * \text{Tick}) * 65536] / 0,0864.$$

6. Если delta — отрицательное число (часы опаздывают), то частота будет со знаком "+", если delta — положительное число (часы спешат), то частота будет со знаком "-".

$$F = (+/-) * F.$$

Рассмотрим алгоритм на **примере**. Будем считать, что с AgentITV пришло рассогласование системного времени с точным delta, равное 20 с. В этом случае расчеты будут следующими.

1. Определим, не превышает ли DELTA константу DELTA\_MAX:

$$[DELTA\_MAX > \text{abs}(\text{delta})];$$

$$[8600 > \text{abs}(20)] = \text{true}.$$

2. Найдем число тиков, необходимых для коррекции:

$$T = \text{mod}(\text{delta}/\text{tick}) = \text{mod}(20/8,64) = 2.$$

3. Вычислим итоговое число тиков, которое будет подставлено в структуру timex:

$$\text{Ticks} = (\text{Ticks\_per\_day} - T) = (10000 - 2) = 9998.$$

4. Проверим, попадает ли число тиков в диапазон допустимых значений:

$$(9000) < 9998 < (11000).$$

5. Вычислим частоту для более точной подстройки:

$$F = [\text{delta} - (T * \text{Tick}) * 65536] / 0,0864 =$$

$$= [20 - (2 * 8,64) * 65536] / 0,0864 = 2063170,37.$$

6. Вычислим знак, с которым следует записать частоту:

$$\text{If}(\text{delta} > 0) \{ F = (-) * F; \}$$

$$\text{else If}(\text{delta} < 0) \{ F = F; \} \text{delta} = -2063170,37.$$

### Заключение

Программная плавная коррекция системных часов контроллеров обеспечивает простое и эффективное решение задачи синхронизации процессов обработки информации в распределенной системе управления.

В контроллерах с операционной системой LINUX плавная коррекция времени обеспечивается системными вызовами ОС adjtime() и adjtimex(). Алгоритм функции adjtimex() может замедлить или увеличить скорость хода системных часов контроллера с погрешностью, не превышающей 1,3 мкс.

Рассмотренная программная реализация коррекции системных часов использована и успешно апробирована в АСКУЭ, разрабатываемой ЗАО НТФ "Системы и технологии" для промышленных предприятий.

*Статья подготовлена с использованием результатов, полученных при проведении поисковой научно-исследовательской работы в рамках реализации ФЦП "Научные и научно-педагогические кадры инновационной России" на 2009—2013 годы, государственный контракт № 16.740.11.0397 от 01.12.2010.*

### Список литературы

1. Сулейманова А. М. Системы реального времени: учеб. пособие. Уфа: Уфимск. гос. авиац. техн. ун-т, 2004. 292 с.
2. Парк Дж., Маккей С., Райт Э. Передача данных в системах контроля и управления: практическое руководство / Пер. с англ. М.: Группа ИДТ, 2007. 482 с.
3. Малафеев С. И., Мамай В. С., Орлов В. А. Электронный счетчик активной энергии однофазного тока // Проектирование и технология электронных средств. 2002. № 4. С. 50—51.
4. Кудрявцев А. А., Поскребышев И. А., Промыслов В. Г. Выбор оптимальных параметров системы единого времени в промышленной компьютерной сети на основе анализа математической модели / Труды конференции "Технические и программные средства систем управления, контроля и измерения", Москва, октябрь, 2010. — М.: Изд. ИПУ РАН, 2010. С. 747—750.
5. Бовет Д., Чезати М. Ядро Linux. — СПб: БХВ-Петербург, 2007. 1104 с.

УДК 519.72

**В. Б. Вяткин**, канд. техн. наук,  
г. Екатеринбург,  
e-mail: vbvzbv@yandex.ru

## Информационно-квантовые характеристики и отраженные образы конечных множеств

*С позиций синергетической теории информации рассматриваются квантовые аспекты информации, отражаемой конечными множествами элементов. При этом вводятся в рассмотрение такие понятия, как кванты информации и кванты отражения, биты отражения и их квантовая емкость, отраженные образы множеств и информационные границы их существования.*

**Ключевые слова:** количество информации, синтропия, бит, квант, конечное множество, отраженный образ

### Введение

В работах [1, 2] представлена синергетическая теория информации, в которой за информацию принимаются сведения о конечном множестве как едином целом. При этом независимо от традиционных подходов к определению количества информации [3–5] получены формула информации  $I_A$ , самоотражаемой произвольным конечным множеством  $A$ , и формула количества информации  $I_{AB}$ , которую отражают (воспроизводят) друг о друге как о целостном образовании два пересекающихся множества  $A$  и  $B$ :

$$I_A = \log_2 M_A; \quad (1)$$

$$I_{AB} = \frac{M_K^2}{M_A M_B} \log_2 M_K, \quad (2)$$

где  $K = A \cap B$ ;  $M_A$ ,  $M_B$ ,  $M_K$  — число элементов в составе множеств  $A$ ,  $B$ ,  $K$ .

В настоящей статье на основе формул (1) и (2) рассматриваются квантовые аспекты информации и проводится соответствующий анализ особенностей отражения друг через друга двух пересекающихся конечных множеств. При этом для повышения адекватности понимания излагаемого материала и исключения терминологической путаницы предварительно даются разъяснения как относительно употребления некоторых терминов, так и относительно формального подобия различных информационных мер.

### Терминологические замечания

◆ Прежде всего, нужно оговориться, что синергетическая теория информации обязана своим названием не научной дисциплине *синергетика*, основанной Г. Хакеном и изучающей самоорганизацию открытых динамических систем [6], а исходному значению слова синергетика, которое в переводе с греческого языка означает *совместный, согласованно действующий*. Дело в том, что в данной теории рассматриваются информационные аспекты отражения конечных множеств как целостных образований. Элементы множеств при этом принимают участие в информационных процессах отражения одновременно всей своей совокупностью без какого-либо выделения любого из них в качестве самостоятельного события, результата испытания и т. п., как это принято делать в традиционной теории информации. Поэтому включение в название указанной теории слова *синергетическая* представляется вполне оправданным. Также можно заметить, что аналогичным образом обосновывал использование слова *синергетика* и Г. Хакен, говоря: "Я назвал новую дисциплину "синергетикой". В ней исследуется совместное действие многих подсистем, в результате которого на макроскопическом уровне возникает структура и соответствующее функционирование." [6, с. 15]. Примечательно, что при этом в арсенал познавательных средств синергетики была включена существующая теория информации.

◆ На первый взгляд может показаться, что формула самоотражаемой множеством информации (1) представляет собой информационную меру Р. Хартли [4], взятую при единичном выборе и двоичном основании логарифма, к которой при одинаковой вероятности событий сводится также информационно-энтропийная мера К. Шеннона [5]:

$$H = \log_2 N, \quad (3)$$

где  $N$  — разнообразие элементов множества по какому-либо признаку  $P = P_1, P_2, \dots, P_N$ .

Между тем формулы (1) и (3) получены различными путями и имеют только внешнее формально-математическое сходство, что видно хотя бы из того, что их аргументы  $M_A$  и  $N$  характеризуют множество с разных сторон. Соответственно, в одной и той же ситуации  $I_A$  и  $H$  в общем случае имеют различные значения, причем  $H \leq I_A$ . В этом можно убедиться на примере буквенных последовательностей конечной длины, информационные оценки которых по формулам (1) и (3) приведены в табл. 1 и 2.

Таблица 1

№	Буквенная последовательность	$M_A$	$N$	$I_A$	$H$
1	$a, b, c, d, e, f, g, h$	8	8	3	3
2	$a, a, b, b, c, c, d, d$	8	4	3	2
3	$a, a, a, a, b, b, b, b$	8	2	3	1
4	$a, a, a, a, a, a, a, a$	8	1	3	0

Таблица 2

№	Буквенная последовательность	$M_A$	$N$	$I_A$	$H$
1	$a, b$	2	2	1	1
2	$a, a, b, b$	4	2	2	1
3	$a, a, a, a, b, b, b, b$	8	2	3	1

Из табл. 1 видно, что  $I_A$  не зависит от разнообразия букв, образующих последовательность, а табл. 2, в свою очередь, показывает, что  $H$  не зависит от общей длины буквенной последовательности. Иначе говоря, самоотражаемая множеством информация (1) и информационная мера Р. Хартли (3) инвариантны относительно друг друга, а их значения равны между собой только в частном случае, когда  $M_A = N$ .

♦ Количественные аспекты информации в проводимых исследованиях рассматриваются с позиций воспроизведения конечных множеств как друг друга ( $I_{AB}$ ), так и через самих себя ( $I_A$ ), т. е. с позиций отражения. Тем самым, в определенной мере формализуются философские воззрения о неразрывной взаимосвязи понятия информации с категорией отражения. Впервые на эту взаимосвязь указал в 1959 г. И. Б. Новик [7], а наиболее обстоятельное ее рассмотрение провел в ряде своих работ А. Д. Урсул [8–10], который пришел к заключению, что информацию "можно определить в самом общем случае как отраженное разнообразие" [8, с. 284]. Но при этом было оговорено, что "при определении тех или иных особенных форм понятия информации более важным может оказаться не признак разнообразия, а какой-либо иной" [10, с. 29]. К таким "особенным формам" относятся информации  $I_A$  и  $I_{AB}$ , которые, как это следует из предыдущего пункта настоящего раздела, не могут интерпретироваться как *отраженное разнообразие*. Эти разновидности информации связаны с объединением всех элементов множества в единое неделимое целое, и, соответственно, в философском отношении понятие информации в данных разновидностях выступает как *отраженное целое*.

♦ В предыдущих работах по анализу информационных аспектов отражения конечных множеств [1, 2, 11–14] информация  $I_{AB}$ , отражаемая друг о друге двумя пересекающимися множествами, именовалась как *негэнтропия отражения*. Это вносило определенную путаницу в общее понимание

термина *негэнтропия*, введенного в научный обиход Л. Бриллюэном [15] в качестве замены словосочетания отрицательная энтропия, которым пользовался Э. Шредингер [16]. Чтобы исключить такую путаницу в дальнейшем и тем самым дистанцироваться от работ Л. Бриллюэна, необходимо отказаться от использования в синергетической теории информации термина негэнтропия, заменив его иным, более адекватно соответствующим содержательной сущности формулы (2). В качестве такой замены в настоящей статье используется греческое слово *синтропия* (*syntropy*), которое может быть переведено на русский язык как *взаимная связь образов, совместный путь, сродство* (приставка *syn-* соответствует приставке *со-*, а корень *trop* может иметь ряд значений, среди которых *образ, путь, манера*). То есть в нижеследующем изложении информация  $I_{AB}$  именуется как *синтропия отражения*, или просто *информационная синтропия*.

Ранее термин *синтропия* в теории информации не применялся, но имеются прецеденты его использования в других областях. Впервые данный термин стали использовать в 1921 г. немецкие педиатры М. Пфаундлер и В. фон Зехт [17], которые назвали синтропией "взаимную склонность, притяжение" двух болезней. Термин был воспринят медицинским сообществом и сейчас довольно широко используется в медицинской практике, в том числе при проведении генетических исследований [18]. Позднее, в 1942 г. итальянский математик Л. Фантаппи при попытке создать объединенную теорию физического и биологического мира [19] назвал синтропией то, что противоположно по смыслу энтропии. Однако работы Л. Фантаппи не увенчались большим успехом и его синтропия не нашла широкого распространения, "уступив дорогу" негэнтропии Л. Бриллюэна. В отличие от указанных синтропий информационная синтропия (2) выражает взаимосвязь абстрактных множеств, в силу чего она имеет более универсальный характер и может использоваться в различных предметных областях. (Например, формула (2) может служить количественной характеристикой синтропии Пфаундлера—Зехта, если за  $M_A$ ,  $M_B$ ,  $M_K$  принять соответствующее число людей с заболеваниями  $A$  и/или  $B$ .)

### Кванты информации

При последовательном увеличении числа элементов конечного множества на единицу множество пробегает соответствующий ряд состояний, в каждом из которых количество самоотражаемой им информации по отношению к предыдущему состоянию, в соответствии с выражением (1), увеличивается на некоторую величину. Так как число элементов конечного множества может принимать только целочисленные значения, то это увеличение происходит отдельными порциями  $\Delta I$ , или

квантами. То есть информация  $I_A$ , которую множество  $A$  отражает о самом себе как едином целом, при увеличении числа его элементов от  $M_A = k$  до  $M_A = k + 1$ , где  $k$  — любое целое число, возрастает на квант информации  $\Delta I_k$ , равный

$$\begin{aligned} \Delta I_k &= I_A|_{M_A = k+1} - I_A|_{M_A = k} = \\ &= \log_2(k+1) - \log_2 k = \log_2\left(1 + \frac{1}{k}\right). \end{aligned} \quad (4)$$

На основе выражения (4) кванту информации можно дать следующее определение: *квант информации — это порция информации, которую приобретает конечное множество при увеличении числа своих элементов на единицу.*

Из формулы (4) видно, что квант информации является переменной величиной, которая монотонно убывает с ростом числа элементов множества. При этом в силу того, что  $M_A = 1 \Rightarrow I_A = 0$ , первый квант информации  $\Delta I_1$  появляется при увеличении числа элементов множества от одного до двух. Отсюда следует, что информация  $I_A$ , самоотражаемая конечным множеством  $A$  с числом элементов  $M_A$ , может быть представлена в виде суммы из  $(M_A - 1)$  квантов информации различной величины:

$$I_A = \sum_{k=1}^{M_A-1} \log_2\left(1 + \frac{1}{k}\right) = \log_2 \prod_{k=1}^{M_A-1} \left(1 + \frac{1}{k}\right). \quad (5)$$

При этом отметим, что выражение (5) равносильно формуле самоотражаемой информации (1), в чем легко убедиться, представив произведение под знаком логарифма в данном выражении в развернутом виде:

$$\begin{aligned} &\prod_{k=1}^{M_A-1} \left(1 + \frac{1}{k}\right) = \\ &= \left(\frac{1+1}{1}\right) \left(\frac{2+1}{2}\right) \dots \left(\frac{M_A-2+1}{M_A-2}\right) \left(\frac{M_A-1+1}{M_A-1}\right) = M_A. \end{aligned}$$

Средняя величина квантов самоотражаемой информации соответственно равна

$$\overline{\Delta I} = \frac{I_A}{M_A-1} = \frac{\log_2 M_A}{M_A-1}. \quad (6)$$

Элементы множества принимают участие в его отражении всей своей совокупностью без какого-либо индивидуального выделения и, соответственно, вклад каждого элемента в самоотражаемую множеством информацию, в общем случае, не может быть оценен с помощью того или иного кванта информации. Строго говоря, для такой оценки не может использоваться и средняя величина информационных квантов (6), так как число последних на единицу меньше, чем общее число элементов множества и, соответственно, относя эту среднюю

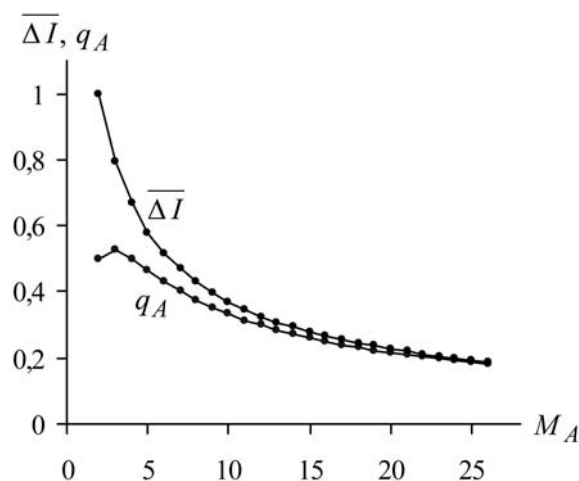


Рис. 1. Зависимость  $\overline{\Delta I}$  и  $q_A$  от  $M_A$

величину к каждому элементу множества, при обратном суммировании результатов мы всегда будем получать завышенное значение самоотражаемой информации. Поэтому вклад каждого элемента в самоотражаемую множеством информацию будем оценивать с помощью удельной информации, которую назовем квантом отражения ( $q$ ). То есть квант отражения  $q_A$  множества  $A$ , состоящего из  $M_A$  элементов, равен

$$q_A = \frac{I_A}{M_A} = \frac{\log_2 M_A}{M_A}. \quad (7)$$

В соответствии с формулой (7) дадим кванту отражения следующее определение: *квант отражения — это часть информации, самоотражаемой конечным множеством, которая приходится на каждый его элемент.*

Из сравнения выражений (6) и (7) следует, что при  $M_A \gg 1$  квант отражения по своей величине становится практически неотличимым от среднего значения квантов информации и при увеличении числа элементов асимптотически приближается к нему снизу, что иллюстрирует рис. 1.

Из приведенного рисунка видно, что кванты отражения при  $M_A = 3$  имеют максимальное значение ( $q_A^{\max} = \log_2 3/3 = 0,528$ ), после которого с ростом числа элементов множества монотонно убывают. Это объясняется тем, что выражение (7) представляет собой сужение функции  $f(x) = \log_a x/x$ ,  $x > 0$ ,  $a > 0$ ,  $a \neq 1$  на множество натуральных чисел при  $a = 2$ , а число 3 является наиболее близким натуральным числом к точке экстремума данной функции  $x_0 = e = 2,718...$ <sup>1</sup>

<sup>1</sup> В соответствии с необходимым условием экстремума функции

$$\frac{d}{dx} f(x) = \frac{1 - \ln a \log_2 x}{x^2 \ln a} = 0, \text{ откуда } x_0 = a^{\frac{1}{\ln a}} = e.$$

Квантовые функции (4)—(7) совместно с самоотражаемой информацией (1) дают вполне исчерпывающее информационно-количественное описание отдельно взятого конечного множества одинаковых по некоторому признаку элементов, для стандартизации которого необходимо определиться с тем, что следует считать единицей измерения информации в проводимых информационно-теоретических исследованиях.

### Биты отражения

В традиционных логарифмических мерах информации (комбинаторной и вероятностной) основание логарифмов является произвольным, что допускает использование различных единиц измерения информации (двоичных, десятичных, натуральных и т. п.). Наибольшей популярностью, в силу технического удобства, используются логарифмы с основанием два, а получаемые при этом двоичные единицы традиционно именуются битами. То есть, 1 бит =  $\log_2 2$ . Так как в комбинаторном и вероятностном подходах информация атрибутивно связана с процедурами выбора [3—5], то содержательно бит обычно интерпретируется как количество информации, получаемой при выборе одной из двух равновероятных возможностей.

В синергетических мерах информации (1) и (2) основание логарифмов может быть равно только двум [1] и, соответственно, единицей измерения может также служить бит. При этом одному биту соответствует равенство  $I_A = 1$ , которое выполняется при  $M_A = 2$ . Вместе с тем, поскольку в синергетическом подходе к количественному определению информации [1], в отличие от вероятностного и комбинаторного подходов, рассматривается иной (не связанный с выбором) вид информации, то бит здесь должен иметь и иную содержательную интерпретацию. Поэтому, чтобы отличать друг от друга численно равные количества различных видов информации, будем называть единицу измерения в синергетической теории информации битом отражения, который определим следующим образом: *бит отражения — это количество информации, которую отражает о самом себе как едином целом множество из двух элементов*. Рассматривая эту единицу измерения информации с квантовых позиций (4), можно также сказать, что бит отражения равен максимальному кванту информации, который образуется при увеличении числа элементов множества от одного до двух.

На основе сказанного бит отражения может быть представлен в следующем виде:

$$1 \text{ бит отражения} = I_A|_{M_A=2} = \Delta I^{\max} = \log_2 2. \quad (8)$$

Из сравнения выражений (7) и (8) следует, что бит отражения является более крупной информационной величиной, чем квант отражения. При

этом, если бит отражения всегда равен единице, то величина кванта отражения изменяется в зависимости от числа элементов отражаемого множества. Это позволяет при рассмотрении отражения конкретных множеств характеризовать биты отражения со стороны их квантовой емкости  $C$ , под которой будем понимать число квантов отражения, соответствующих одному биту. То есть квантовая емкость  $C_A$  одного бита отражения множества  $A$  с числом элементов  $M_A$  равна

$$C_A = \frac{1 \text{ бит отражения}}{q_A} = \frac{M_A}{\log_2 M_A}. \quad (9)$$

Из выражения (9) также следует, что квантовая емкость  $C_A$  увеличивается с ростом числа элементов  $M_A$  и показывает, сколько элементов множества  $A$  приходится на один бит его отражения. Это открывает путь для непосредственной оценки числа элементов одного множества, отраженного через пересекающееся с ним другое множество.

### Отраженные образы множеств

При рассмотрении отражения друг через друга двух пересекающихся конечных множеств  $A$  и  $B$  (рис. 2) перед нами неизбежно встает вопрос о том, в каком виде воспроизводится при этом каждое из множеств. Иначе говоря, нас интересуют отраженные образы множества  $A$  через множество  $B$  ( $A^A \rightarrow B$ ) и множества  $B$  через множество  $A$  ( $B^B \rightarrow A$ ).

Собственно о множествах  $A$  и  $B$  помимо того, что они выделены по некоторым отличительным признакам  $P_A$  и  $P_B$ , нам известно только то, что число элементов в их составе равно  $M_A$  и  $M_B$ . Поэтому поставленный вопрос сводится к задаче определения числа элементов  $M_{A^A \rightarrow B}$  и  $M_{B^B \rightarrow A}$ , в составе образов  $A^A \rightarrow B$  и  $B^B \rightarrow A$  соответственно.

Предварительно отметим, что число элементов в составе каждого из множеств  $A$  и  $B$  может быть оценено уже с помощью синтропии отражения  $I_{AB}$ :

$$I_{AB} \geq 1 \Rightarrow M_A \geq 2^{I_{AB}}, M_B \geq 2^{I_{AB}}. \quad (10)$$

Условие  $I_{AB} \geq 1$  в выражении (10) обусловлено тем, что поскольку  $K \subset A$ ,  $K \subset B$ , то из формулы (2) следует, что при любом  $I_{AB} > 0$  число элементов в составе множеств  $A$  и  $B$  не может быть меньше двух. Но независимо от данного условия величина  $2^{I_{AB}}$  не может быть принята за число элементов

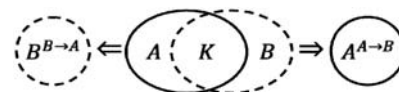


Рис. 2. Модель отражения конечных множеств  $A$  и  $B$  друг через друга

в составе отраженных образов  $A^A \rightarrow B$  и  $B^B \rightarrow A$ . В качестве причины этого укажем следующее. Во-первых, одному и тому же значению синтропии  $I_{AB}$  могут соответствовать различные совокупности значений  $M_A$  и  $M_B$  (даже при постоянстве  $M_K$ ), и логично потребовать, чтобы каждая такая совокупность формировала свои отраженные образы  $A^A \rightarrow B$  и  $B^B \rightarrow A$ . Во-вторых, в квантовом отношении значения синтропии  $I_{AB}$  в общем случае неравноценны по отношению к каждому из множеств  $A$  и  $B$ , что должно также учитываться при определении  $M_{A^A \rightarrow B}$  и  $M_{B^B \rightarrow A}$ .

В соответствии со сказанным естественным решением задачи определения  $M_{A^A \rightarrow B}$  и  $M_{B^B \rightarrow A}$  выглядит произведение квантовой емкости битов отражения на величину синтропии отражения. Действительно, квантовая емкость (9) показывает, сколько элементов отражаемого множества приходится на один бит его отражения, а синтропия (2) выражает число этих битов, воспроизведенное через отражающее множество. В соответствии с этим число элементов в отраженных образах  $A^A \rightarrow B$  и  $B^B \rightarrow A$  равно (с округлением до ближнего меньшего целого):

$$M_{A^A \rightarrow B} = C_A I_{AB} = \frac{M_K^2 \log_2 M_K}{M_B \log_2 M_A}; \quad (11)$$

$$M_{B^B \rightarrow A} = C_B I_{AB} = \frac{M_K^2 \log_2 M_K}{M_A \log_2 M_B}. \quad (12)$$

Синтропийная оценка (10) показывает нижний порог возможных значений числа элементов в составе множеств  $A$  и  $B$  и, соответственно, число элементов (11) и (12) в отраженных образах данных множеств должно быть не ниже этого порога. Иначе говоря, должны выполняться неравенства

$$I_{AB} \geq 1 \Rightarrow M_{A^A \rightarrow B} \geq 2^{I_{AB}}, \quad M_{B^B \rightarrow A} \geq 2^{I_{AB}}. \quad (13)$$

Докажем, что эти неравенства действительно выполняются, для чего на примере  $M_{A^A \rightarrow B}$  подставим его значение из (11) в (13) и прологарифмируем полученное выражение. В результате этой операции получаем:

$$\log_2 M_A - \log_2(\log_2 M_A) + \log_2 I_{AB} \geq I_{AB}.$$

Делая необходимые перестановки, приходим к неравенству

$$\log_2 M_A - I_{AB} \geq \log_2(\log_2 M_A) - \log_2 I_{AB}.$$

Так как  $\log_2 M_A = I_A \geq I_{AB}$ , то последнее неравенство при  $I_{AB} \geq 1$  является очевидным, что доказывает справедливость неравенств (13).

Проведем теперь сравнение отраженных образов  $A^A \rightarrow B$  и  $B^B \rightarrow A$ , которое может быть осуществлено с двух сторон: со стороны соотношения образов по числу элементов и со стороны их соответствия исходным множествам  $A$  и  $B$ . В первом случае вследствие того, что квантовые емкости  $C_A$  и  $C_B$  увеличиваются по мере роста числа элементов  $M_A$  и  $M_B$ , из выражений (11) и (12) следует, что из *отраженных образов двух пересекающихся конечных множеств превосходство по числу элементов имеет образ того множества, у которого большее число элементов*. Во втором случае, принимая за меру соответствия образа оригиналу отношение числа его элементов к числу элементов исходного множества, мы приходим к тому, что эта мера равна относительной синтропии отражения  $J$ , характеризующей полноту отражения одного множества через пересекающееся с ним другое множество:

$$\frac{M_{A^A \rightarrow B}}{M_A} = \frac{M_K^2 \log_2 M_K}{M_A M_B \log_2 M_A} = \frac{I_{AB}}{I_A} = J_{A \rightarrow B}; \quad (14)$$

$$\frac{M_{B^B \rightarrow A}}{M_B} = \frac{M_K^2 \log_2 M_K}{M_A M_B \log_2 M_B} = \frac{I_{AB}}{I_B} = J_{B \rightarrow A}. \quad (15)$$

Так как  $I_A$  и  $I_B$  тем больше, чем больше соответственно  $M_A$  и  $M_B$ , то из выражений (14) и (15) следует вывод о том, что *при отражении друг через друга двух пересекающихся конечных множеств более адекватный отраженный образ имеет то множество, которое состоит из меньшего числа элементов*.

Естественно считать, что отражение пересекающихся множеств  $A$  и  $B$ , как их воспроизведение друг через друга, существует только тогда, когда их отраженные образы  $A^A \rightarrow B$  и  $B^B \rightarrow A$  не являются пустыми множествами, т. е., когда  $M_{A^A \rightarrow B} \geq 1$ ,

$M_{B^B \rightarrow A} \geq 1$ . В том случае, когда отраженный образ

того или иного множества включает в себя только один элемент, синтропия  $I_{AB}$  как это следует из (11) и (12), равна соответствующему кванту отражения исходного множества:  $M_{A^A \rightarrow B} = 1 \Rightarrow I_{AB} = q_A$

и  $M_{B^B \rightarrow A} = 1 \Rightarrow I_{AB} = q_B$ .

Сказанное означает, что условием наличия отражения множеств  $A$  и  $B$  друг через друга являются неравенства  $I_{AB} \geq q_A$  и  $I_{AB} \geq q_B$ , откуда следует, что отражение каждого из пересекающихся множеств имеет свою информационную (синтропийную) границу в виде равенства синтропии отражения соответствующему кванту отражения. Это приводит нас к выводу о том, что *если величина синтропии отражения  $I_{AB}$  меньше кванта отражения какого-либо из множеств  $A$  и  $B$ , то по отношению к этому множеству ее значение представляет собой информационный шум*. Образно выражаясь, при выполне-

нии неравенств  $0 < I_{AB} < q_A$  и  $0 < I_{AB} < q_B$  множества  $A$  и  $B$  как бы "ощущают" друг друга, но силы (информации) этого "ощущения" недостаточно для того, чтобы воспроизвести их непустые образы.

### Особенности взаимного отражения системы и части

Материал предыдущего раздела относится к наиболее общему случаю пересечения множеств  $A$  и  $B$ , когда  $A \neq B$ ,  $K \neq A$ ,  $K \neq B$ . Вместе с тем, в практической деятельности мы весьма часто имеем дело с ситуацией, когда, например, множество  $A$  рассматривается как автономная дискретная система, а множество  $B$  выступает в качестве ее части или подсистемы, т. е. когда  $K = B \subset A$ . Поэтому остановимся на соответствующих моментах.

Прежде всего отметим, что в данном случае  $M_K = M_B$ , в силу чего синтропия отражения (2) приобретает вид

$$B \subset A \Rightarrow I_{AB} = \frac{M_B}{M_A} \log_2 M_B. \quad (16)$$

Соответственно, число элементов в отраженных образах системы  $A$  и ее части  $B$  согласно (11) и (12) равно

$$B \subset A \Rightarrow \begin{cases} M_{A \rightarrow B} = M_B \frac{\log_2 M_B}{\log_2 M_A}; \\ M_{B \rightarrow A} = \frac{M_B^2}{M_A}. \end{cases} \quad (17)$$

Относительная синтропия отражения системы и части друг через друга, в свою очередь, равна

$$B \subset A \Rightarrow \begin{cases} J_{A \rightarrow B} = \frac{M_B \log_2 M_B}{M_A \log_2 M_A}; \\ J_{B \rightarrow A} = \frac{M_B}{M_A}. \end{cases} \quad (18)$$

На основании (17) и (18) ранее сделанные заключения о соотношении отраженных образов множеств по числу элементов и соответствию их оригиналам по отношению к системе и части предстают перед нами в следующем виде: *образ системы, отраженный через ее часть, содержит большее число элементов, чем образ части, отраженный через систему, и полнота отражения части системой больше, чем полнота отражения системы частью.*

Величина синтропии взаимного отражения системы и части (16), в отличие от общего случая пересечения множеств (2), зависит только от значений  $M_A$  и  $M_B$ . Вследствие этого информационным границам отражения системы и части соответству-

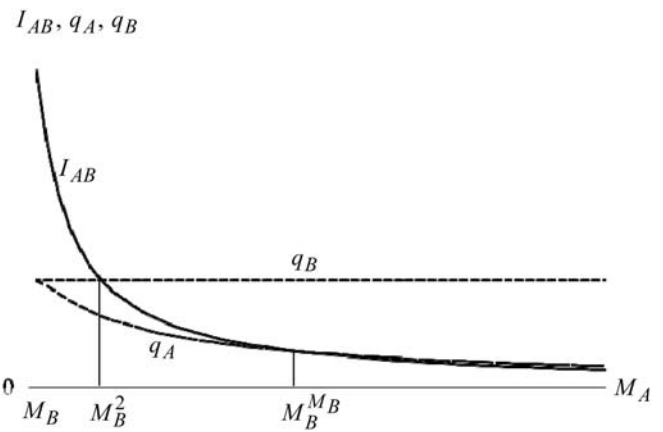


Рис. 3. Зависимость  $I_{AB}$ ,  $q_A$ ,  $q_B$  от  $M_A$  при  $M_B = \text{const}$

ют строго определенные соотношения  $M_A$  и  $M_B$ , которые, согласно (7) и (16), имеют вид

$$B \subset A \Rightarrow \begin{cases} I_{AB} = q_A \Rightarrow M_A = M_B^{M_B}; \\ I_{AB} = q_B \Rightarrow M_A = M_B^2. \end{cases} \quad (19)$$

Наглядное представление о соотношениях (19) дает рис. 3, где приведены графики зависимости  $I_{AB}$ ,  $q_A$ ,  $q_B$  от  $M_A$  при фиксированном  $M_B$ .

Из рисунка видно, что проекции информационных границ отражения  $I_{AB} = q_A$  и  $I_{AB} = q_B$  на горизонтальную ось делят область возможных значений  $M_A$  на три интервала, каждый из которых по отношению к отраженным образам  $A^{A \rightarrow B}$  и  $B^{B \rightarrow A}$  обладает качественным своеобразием:

$$B \subset A \Rightarrow \begin{cases} M_B \leq M_A \leq M_B^2 \Rightarrow A^{A \rightarrow B}, B^{B \rightarrow A} \neq \emptyset; \\ M_B^2 < M_A \leq M_B^{M_B} \Rightarrow A^{A \rightarrow B} \neq \emptyset, B^{B \rightarrow A} = \emptyset; \\ M_A > M_B^{M_B} \Rightarrow A^{A \rightarrow B}, B^{B \rightarrow A} = \emptyset. \end{cases} \quad (20)$$

В содержательном плане из выражения (20) следует, что увеличение системы  $A$  за счет элементов, обладающих только признаком  $P_A$ , сопровождается тем, что часть  $B$ , выделенная по признаку  $P_B$ , воспроизводит непустой образ системы несоизмеримо дольше, чем система образ части. При этом уже при переходе числа элементов системы через границу  $M_A = M_B^2$  признак  $P_B$  становится для системы в информационном отношении несущественным явлением. Вместе с тем, является очевидным, что эта несущественность признака не имеет отношения к системам с малым числом элементов (подобно тому, как не имеет смысла говорить о выполнении второго начала термодинамики в случае небольших совокупностей молекул идеального газа или, например, судить о законе распределения случайной величины по небольшому числу ее наблюдаемых значений).

Помимо изложенного в информационных взаимоотношениях системы и части (множества и подмножества) обращает на себя внимание тот факт, что отношение  $M_B/M_A$ , фигурирующее в выражениях (16)—(18), при достаточно большом  $M_A$  может рассматриваться как вероятность  $P(B)$  встречи признака  $P_B$  среди элементов множества  $A$ . В то же время это отношение, с одной стороны, представляет собой относительную синтропию отражения части через систему, а с другой стороны, равно отношению числа элементов отраженного образа части к числу элементов самой части (см. формулы (18) и (17)). То есть

$$B \subset A \Rightarrow J_{B \rightarrow A} = P(B) = \frac{M_B}{M_A} = \frac{M_{B \rightarrow A}}{M_B}. \quad (21)$$

Выражение (21) свидетельствует о глубокой взаимосвязи синтропии отражения и отраженных образов конечных множеств с вероятностью случайных событий. Вместе с тем, детальное рассмотрение этой взаимосвязи выходит за рамки настоящей статьи и является предметом отдельных исследований.

### Заключение

С позиций синергетической теории информации в статье рассмотрены квантовые аспекты информации, отражаемой конечными множествами элементов. При этом за информацию приняты сведения о конечном множестве как едином целом, а информации, которую отражают друг о друге два пересекающихся множества, дано название *синтропия отражения*. В результате проведенных исследований в оборот такие понятия, как квант информации и квант отражения, бит отражения и его информационная емкость, которым даны формализованные определения. Также решена задача оценки числа элементов в отраженных образах пересекающихся конечных множеств и установлены информационные (синтропийные) границы существования этих образов. Последнее может использоваться на практике, например, при решении задач распознавания образов (оценка информативности признаков по адекватности воспроизведения ими эталонных множеств объектов распознавания и минимизация признакового пространства с помощью информационных границ отражения).

В целом изложенный материал является развитием синергетической теории информации и расширяет представления о количественной стороне того вида информации, который существует независимо от управления.

### Список литературы

1. **Вяткин В. Б.** Синергетический подход к определению количества информации // Информационные технологии. 2009. № 12. С. 68—73.
2. **Вяткин В. Б.** Введение в синергетическую теорию информации // Информационные технологии. 2010. № 12. С. 67—73.
3. **Колмогоров А. Н.** Три подхода к определению понятия "количество информации" // Проблемы передачи информации. 1965. Т. 1, № 1. С. 3—11.
4. **Хартли Р. В. Л.** Передача информации // Теория информации и ее приложения. М.: Физматгиз, 1959. С. 5—35.
5. **Шеннон К.** Работы по теории информации и кибернетике. М.: Изд. иностр. лит., 1963. 830 с.
6. **Хакен Г.** Синергетика. М.: Мир, 1980. 404 с.
7. **Новик И. Б. В. И.** Ленин о единстве мира // Великое провозвешение воинствующего материализма. М.: Соцэкономиздат, 1959. С. 165—181.
8. **Урсул А. Д.** Природа информации. М.: Политиздат, 1968. 288 с.
9. **Урсул А. Д.** Отражение и информация. М.: Мысль, 1973. 231 с.
10. **Урсул А. Д.** Проблема информации в современной науке. М.: Наука, 1975. 288 с.
11. **Вяткин В. Б.** К вопросу информационной оценки признаков при прогнозно-геологических исследованиях // Известия Уральского горного института. Сер.: Геология и геофизика. 1993. Вып. 2. С. 21—28.
12. **Вяткин В. Б.** Информационно-энтропийный анализ отражения системных объектов // Техногенез и экология: Информационно-тематический сборник. Екатеринбург: УГГА, 1999. С. 50—68.
13. **Вяткин В. Б.** Информационный закон отражения системных объектов // Проблемы методологии междисциплинарных исследований и комплексного обеспечения научно-исследовательской деятельности. Екатеринбург: УрО РАН. 2001. Вып. 3. С. 24—42.
14. **Вяткин В. Б.** Синергетическая теория информации: общая характеристика и примеры использования // Наука и оборонный комплекс — основные ресурсы российской модернизации. Материалы межрегиональной научно-практической конференции. Екатеринбург: УрО РАН, 2002. С. 361—390.
15. **Бриллюэн Л.** Наука и теория информации. М.: Физматгиз, 1960. 392 с.
16. **Шредингер Э.** Что такое жизнь? Точка зрения физика. М.: Атомиздат, 1972. 88 с.
17. **Pfaundler M., von Seht L.** Weiteres uber Syntropie kindlicher Krankheitszustande // Zeitschr. f. Kinderheilk. 1921, bd. 30. S. 298—313.
18. **Пузырев В. П.** Генетический взгляд на феномен сочетанной патологии у человека // Медицинская генетика. 2008. № 9. С. 3—9.
19. **Fantappiè L.** Principi di una teoria unitaria del mondo fisico e biologico. Rome: Accademia d'Italia. 1942.



УДК 004.75:378.1

**Е. В. Федорова**, канд. хим. наук, зав. каф.,  
e-mail: elena.fedorova@pharminnotech.com,

**М. А. Гетьман**, канд. фарм. наук,  
директор научно-образовательного центра,

**Е. В. Савельева**, зам. директора  
ГБОУ ВПО СПХФА Минздравсоцразвития России

## Использование облачных технологий Google Apps Education Edition для создания саморазвивающейся информационной платформы вуза

*Описан опыт создания и развития единой информационной платформы в ГБОУ ВПО Санкт-Петербургская химико-фармацевтическая академия на основе использования общедоступных облачных информационных технологий Google Apps Education Edition. Представлены основные этапы разработки, внедрения, структура информационной платформы, преимущества использования Web-приложений на основе облачных вычислений для образовательного процесса.*

**Ключевые слова:** Google Apps Education Edition, единое информационное пространство, корпоративная информационная система (КИС), образовательные информационные технологии

С 2006 по 2010 г. в ГБОУ ВПО Санкт-Петербургская химико-фармацевтическая академия Минздравсоцразвития России (далее — СПХФА) функционировал Интернет-сайт на основе HTML, который выполнял в основном справочную и представительскую функцию, а наполнение сайта информацией обеспечивал ограниченный круг администраторов. Основные недостатки концепции заключались в дефектах дизайна, несовершенстве принципов управления информацией, отсутствии возможности формирования контента профессорско-преподавательским составом и ведения совместной работы студентов, преподавателей и администрации с документами. Хостинг сайта осуществлялся на собственной серверной группе СПХФА, что приводило к частым сбоям в его работе.

Сложившаяся ситуация не соответствовала требованиям времени, а также рекомендациям Феде-

рального агентства по образованию, касающимся разработки систем управления образовательными учреждениями, к важнейшим характеристикам которых относятся интегрированность, адаптируемость, распределенность, масштабируемость и качество [1—3].

В целях совершенствования используемых информационных технологий в ноябре 2010 г. СПХФА приобрела право безвозмездного использования информационного продукта с открытым кодом Google Apps Education Edition (далее — GAEE) для создания на ее основе единой информационной системы вуза.

GAEE представляет собой специальное интегрированное Web-решение, основанное на использовании облачных технологий и разработанное компанией Google Inc. (США) для использования в учебных заведениях среднего и высшего образования в любой стране мира [4].

Основными достоинствами GAEE являются:

- интегрированность во все Web-продукты Google Inc., а также совместимость со многими приложениями и функциональными модулями, разрабатываемыми компаниями-партнерами Google Inc.;
- бесплатное размещение всего сервиса на серверах компании Google Inc., которая обеспечивает высочайшую надежность хранения данных (99,97 %) и техническое обслуживание. Благодаря этому учебному заведению — пользователю GAEE — не требуется создание собственной серверной базы и формирование группы по техническому обслуживанию;
- совместимость с программным обеспечением мобильных устройств (Android, Windows 7);
- мультилингвальность, позволяющая создавать интерфейсы на любом из 53 мировых языков, на которых работает продукция компании Google Inc., а также дающая возможность автоматического перевода содержания;
- широкие возможности и простота разработки Web-страниц (в рамках одной лицензии компания Google Inc. позволяет создавать до 100 тыс. доменов третьего уровня);
- предоставление на безвозмездной основе многофункционального почтового сервиса, размещаемого на основном домене учебного заведения (до 3000—5000 аккаунтов);
- эффективная защита от спама и вирусов;

- доступность разнообразных утилит и приложений (бесплатных и платных) на Google Apps Marketplace<sup>1</sup>.

Дополнительно компания Google Inc. предоставляет дисконтированные возможности по регистрации новых доменных имен и по хостингу. В то же время не существует никаких ограничений на использование собственных доменных имен и национальных доменных зон (например, RU).

При авторизации на Web-портале компании Google Inc. разработчиками СПХФА было приобретено доменное имя [pharminnotech.com](http://pharminnotech.com) (впоследствии также [pharminnotech.ru](http://pharminnotech.ru) и [pharminnotech.org](http://pharminnotech.org)), а также зарегистрирован аккаунт владельца — главного администратора системы, который согласно протоколу обладает максимальными полномочиями, в том числе по коммуникациям с Google Inc. в вопросах сервисов, технической поддержки, расширения мандата и т. д. Доступ к сервисам GAEE предоставляется в течение нескольких дней после регистрации и только после предоставления доказательств того, что заявитель является образовательной организацией. В случае СПХФА в Google Inc. были представлены сканированные копии российской образовательной лицензии и свидетельства об аккредитации. Уведомление об открытии доступа поступает в электронной форме на аккаунт главного администратора системы вуза.

*Первым шагом* по реализации проекта создания единой информационной платформы СПХФА стала разработка на домене второго уровня Web-сайта "Аптекарский остров"<sup>2</sup> — информационной площадки сообщества студентов, преподавателей, выпускников и ветеранов СПХФА.

Сайт сообщества представляет собой набор Web-страниц, содержащих информацию, представленную в текстовой, изобразительной, аудио- и видеоформах, а также модули регистрации пользователей, доступа к аккаунтам, справочной информации по использованию платформы GAEE. Несмотря на то, что у разработчиков, являющихся сотрудниками СПХФА, полностью отсутствовал опыт Web-дизайна, первый сайт был разработан в течение четырех недель.

Простота Web-конструктора GAEE является одним из его важнейших преимуществ и позволяет создавать информационную систему вуза как саморазвивающуюся модель. Фактически любой учащийся или преподаватель способен самостоятельно и быстро (при использовании шаблона всего за несколько часов) разработать индивидуальную систе-

<sup>1</sup> Подробно возможности системы описаны на сайте разработчика <http://www.google.com/a/help/intl/ru/edu/university.html>. Там же приведена информация об образовательных учреждениях, которые используют Google Apps.

<sup>2</sup> Первоначально сайт был разработан на домене второго уровня [www.pharminnotech.com](http://www.pharminnotech.com) и впоследствии переведен на домен третьего уровня: <http://alumni.pharminnotech.com/>, на котором располагается в настоящее время.

му функционально оснащенных Web-страниц. Отсутствие технологических барьеров, легкая познаваемость и интуитивность интерфейса делает GAEE особенно удобным для учебных заведений, чьи информационные потребности многообразны и требуют вовлечения в информационный обмен максимально широкого круга заинтересованных лиц, не опосредованного узкой группой администраторов системы.

Web-сайты, создаваемые в рамках платформы, могут публиковаться как без ограничений (вся сеть Интернет), так и с ограничениями доступа (только для аккаунтов системы либо только для некоторых аккаунтов/групп аккаунтов системы).

*Вторым шагом* стала регистрация индивидуальных аккаунтов для каждого учащегося (включая заочное отделение, интернатуру и аспирантуру) и сотрудника СПХФА. Формат адресной строки аккаунта может выбираться разработчиком по своему усмотрению. В СПХФА был выбран стиль **имя.фамилия@pharminnotech.com**. Аккаунты были собраны в группы: факультеты — учебные группы — курсы для учащихся и кафедры — подразделения сотрудников. Каждый аккаунт может участвовать в неограниченном числе групп, что позволяет вузу создавать удобную систему электронных оповещений, а также селективного доступа к информационным ресурсам. Регистрация аккаунтов, их группирование и предоставление полномочий может осуществляться только аккаунтами, обладающими полномочиями администраторов. В свою очередь, полномочия администраторам предоставляет только аккаунт владельца — главного администратора системы.

Окно аккаунта GAEE представляет собой рабочий стол, оснащенный инструментами электронной почты (аналогичный Gmail), разработки сайтов, создания и хранения документов, системой планирования времени, управления контактами, формирования групп общения, текстовым, голосовым и видеочатом, в том числе с возможностями проведения конференций, панелью управления и настройки аккаунта, а также его инструментов.

*Третьим шагом* стало проведение силами двух человек нескольких лекций и семинаров в студенческих аудиториях и компьютерных классах. Общее число часов обучения составило 20. В ходе этих мероприятий с возможностями информационной системы познакомилось около 200 человек.

В мае 2011 г. была сформирована рабочая группа из восьми человек (преподавателей и студентов СПХФА), координирующая развитие и поддержку платформы согласно совместно разработанному плану.

В течение полугода на базе платформы GAEE СПХФА были созданы сайты практически всех учебных, научных и административных отделов, лабораторий и центров академии, сайты подразде-

лений (кафедр), сайты сообщества стали пополняться персональными страницами участников, страницами студенческих учебных групп. В настоящее время в системе функционирует более 600 Web-страниц. В связи с этим необходимо отметить, что некоторые учащиеся, преподаватели, научные работники СПХФА по собственной инициативе начали создавать Web-сайты и поддерживать их работоспособность.

Архитектура информационной платформы (корпоративной информационной среды — КИС) СПХФА представлена на рисунке. Два основных домена — главный сайт академии на домене *srcpa.ru* (I) и интранет на домене *pharminnotech.com* (II) — функционально связаны между собой. Представительский сайт, расположенный на домене *srcpa.ru* (I), был создан с помощью общедоступного конструктора Joomla, предоставляющего разработчикам исключительные возможности по оформлению, цветовой гамме и другим элементам дизайна, кроме того, совместимый с GAEE. Сайт содержит общую информацию об академии, правовую и справочную информацию, документацию по системе менеджмента качества и является порталом в интранет. Обеспечена и обратная связь: все новости, публикуемые на сайтах информационной платформы, автоматически появляются в

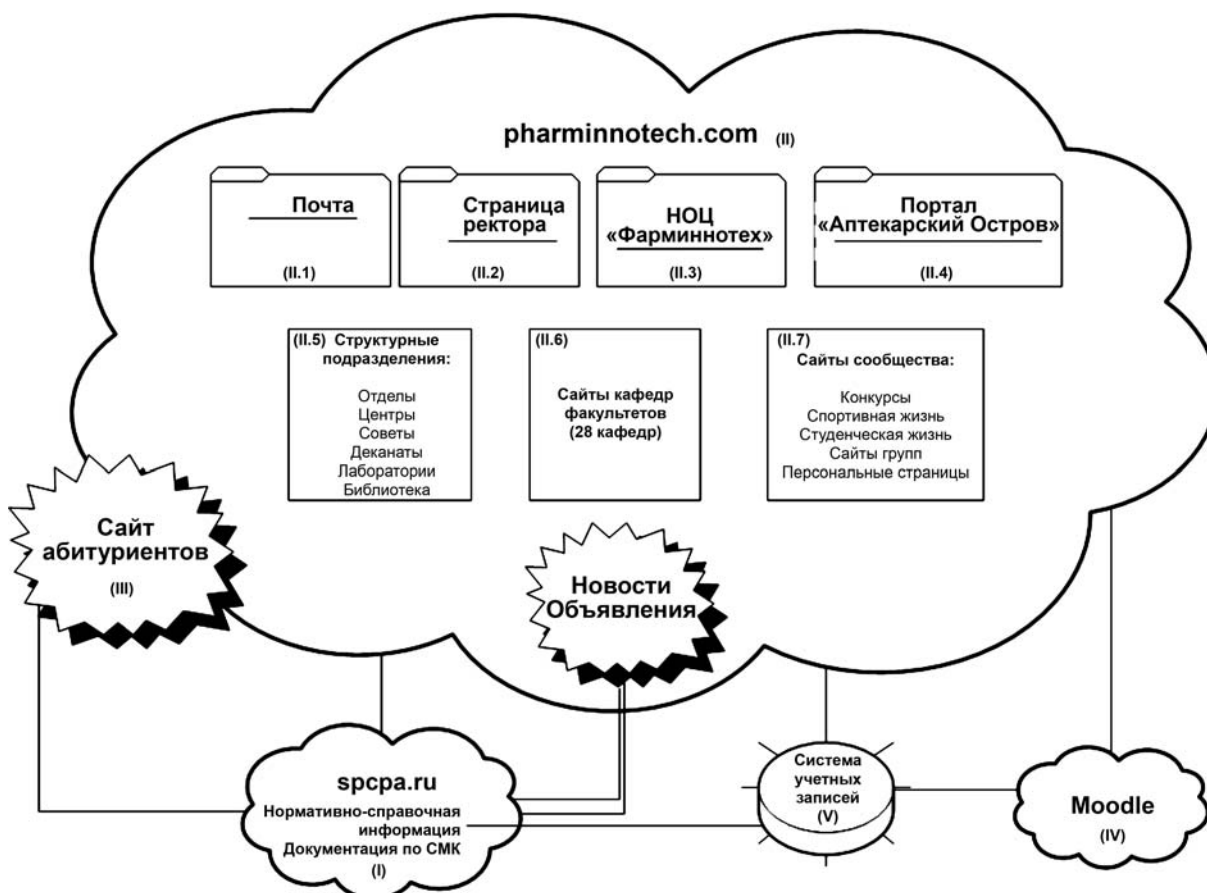
соответствующих рубриках на главном сайте академии.

Для информационной поддержки проведения приемной кампании и профориентационной работы на базе GAEE был разработан сайт для абитуриентов на домене третьего уровня *abiturient.pharminnotech.com* (III).

С информационной платформой СПХФА связан комплекс дистанционного обучения Moodle (IV), доступ к которому предоставляется через аккаунты системы (V) по протоколу SSO.

Интранет включает в себя электронную почту (II.1), обратную связь с администрацией (II.2), научно-образовательный портал научно-образовательного центра (НОЦ) (II.3), социальный портал (II.4), сайты структурных подразделений (II.5), сайты кафедр (II.6) и сайты сообщества (II.7). Сайты сообщества представлены персональными Web-страницами сотрудников, студентов, студенческих групп и организаций. Единая система аккаунтов позволяет пользователям самостоятельно создавать сайты и устанавливать права доступа к ним.

Благодаря широким возможностям сервисов GAEE в СПХФА учащимся обеспечивается дистанционный доступ к кафедральным учебно-методическим комплексам, что особенно важно для студентов заочного отделения. Осуществляется



Архитектура информационной платформы (корпоративной информационной среды — КИС) СПХФА

совместная работа с документами, тестирование и анкетирование, реализуются программы дистанционного обучения на базе Moodle. Активно проводится воспитательная работа и работа по социальному ориентированию (сайты, посвященные студенческой жизни, спорту в СПХФА, деятельности профкома студентов и др.). Организуются студенческие онлайн-конференции, работает централизованная система информирования участников сообщества (новости, события, гранты, конференции), развернуты дискуссионные площадки, форумы, интернет-радио. Учащиеся имеют возможность прямого общения с преподавателями, администрацией благодаря единому встроенному справочнику аккаунтов в форме переписки, текстового, голосового, видеочата или на странице ректора.

Все учащиеся и сотрудники СПХФА имеют неограниченный доступ к коммуникационным, информационным, научным и учебным ресурсам информационной платформы СПХФА с любого устройства, способного подключаться к сети Интернет.

Наиболее активными пользователями облачных сервисов GAEE являются студенты начальных курсов дневной формы обучения, администрация и преподаватели. Несмотря на разнообразие инструментария не все приложения используются одинаково активно. Наиболее популярными приложениями среди пользователей СПХФА являются электронная почта, календарь, документы, сайты, чат и группы. Менее активно используются Google Wave, Picasa, Google Maps, iGoogle, Google Reader. Остальные сервисы используются редко.

Необходимо признать, что основной проблемой, с которой разработчики столкнулись в начале проекта, явилась неготовность учащихся и сотрудников к работе в универсальной информационной среде и, как следствие, низкая пользовательская активность. Так, по данным Google Analytics (еще один мощный SEO-инструмент, средство Web-аналитики корпоративного уровня) из 3000 зарегистрированных пользовательских аккаунтов через три месяца после запуска проекта активными были лишь 180 (6 %). Среди основных причин — низкий уровень пользовательской грамотности; недостаточная развитость внутривузовской инфраструктуры — компьютеров и сетевого доступа (не все учебные корпуса и общежития были обеспечены доступом к сети Интернет).

С учетом этих обстоятельств рабочей группой был предпринят ряд мер:

- были дополнительно организованы курсы обучения по работе с информационной платформой среди учащихся и профессорско-преподавательского состава;
- тематические разделы по созданию Web-сайтов, документов, тестов и т. п. на базе инструментов информационной платформы были включены в квалификационный цикл "Преподаватель высшей школы".

В целях привлечения студенческой аудитории начали проводиться конкурсы (лучший спортсмен, лучшая авторская колонка, лучший преподаватель глазами студента, лучшее видео), созданы сайты, посвященные спорту в СПХФА и студенческой жизни. Это позволило за три—четыре месяца увеличить число активных аккаунтов примерно в три раза (16 %). Дополнительно администрация СПХФА активно совершенствует инфраструктуру доступа к информационной платформе: обновляется парк компьютерной техники, осуществляется замена кабелей связи и точек подключения, открываются зоны Wi-Fi, бесплатные для аккаунтов информационной платформы.

Создание информационной платформы СПХФА на базе технологий GAEE внесло многочисленные усовершенствования в учебную, научную и общественную жизнь вуза и дало сильный толчок к саморазвитию учащихся и сотрудников академии в вопросах использования современных информационных технологий. Это также повлияло, согласно опросу абитуриентов 2011 г., на принятие ими решения о выборе вуза.

#### Список литературы

1. Некрасова Е. Информационное пространство вуза // СЮ. 2006. № 12.
2. Иванников В. П. Облачные вычисления в образовании, науке и госсекторе // Материалы семинара "Облачные вычисления для органов государственной власти и местного самоуправления", Москва, 2010. URL: <http://www.iis.ru/docs/ivannikov.pdf>
3. Аманжолов Р. М., Ботыгин И. А. CLOUD COMPUTING: в науке и образовании // Технологии Microsoft в теории и практике программирования: сб. докл. VIII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых (Томск, 23—24 марта 2011 г.). 2011. URL: <http://www.lib.tpu.ru/fulltext/v/Conferences/2011/K01/055.pdf>
4. Сейдаметова З. С., Сейтвелиева С. Н. Облачные сервисы в образовании // Информационные технологии в образовании. 2011. № 9. С. 105—111.

---

*Редакция приносит свои извинения авторам и читателям за допущенную орфографическую ошибку при печати англоязычного названия метода анализа конкретных ситуаций в статье С. М. Авдошина, А. А. Савельевой "Междисциплинарный подход к изучению информационной безопасности на основе анализа конкретных ситуаций в журнале "Информационные Технологии" № 5, 2012, с. 58. Правильный вариант: "case study".*

---

**ЖУРНАЛ В ЖУРНАЛЕ**

**НЕЙРОСЕТЕВЫЕ  
ТЕХНОЛОГИИ**

**№ 7**

**ИЮЛЬ**

**2012**

**Главный редактор:**

ГАЛУШКИН А. И.

**Редакционная коллегия:**

АВЕДЬЯН Э. Д.  
БАЗИАН Б. Х.  
БЕНЕВОЛЕНСКИЙ С. Б.  
БОРИСОВ В. В.  
ГОРБАЧЕНКО В. И.  
ЖДАНОВ А. А.  
ЗЕФИРОВ Н. С.  
ЗОЗУЛЯ Ю. И.  
КРИЖИЖАНОВСКИЙ Б. В.  
КУДРЯВЦЕВ В. Б.  
КУЛИК С. Д.  
КУРАВСКИЙ Л. С.  
РЕДЬКО В. Г.  
РУДИНСКИЙ А. В.  
СИМОРОВ С. Н.  
ФЕДУЛОВ А. С.  
ЧЕРВЯКОВ Н. И.

**Иностранные  
члены редколлегии:**

БОЯНОВ К.  
ВЕЛИЧКОВСКИЙ Б. М.  
ГРАБАРЧУК В.  
РУТКОВСКИЙ Л.

**Редакция:**

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

**Скрибцов П. В., Казанцев П. А., Червоненкис М. А.**

Применение искусственных нейронных сетей для решения обратных задач в гидрологии . . . . . 62

**Борисов В. В., Мисник А. Е.**

Комбинированный нейросетевой способ моделирования для оперативного управления сложными системами . . . . . 69

**Степанов С. Ю., Кабак И. С.**

Алгоритм фрагментации больших нейронных сетей и исследование его сходимости . . . . . 73

**П. В. Скрибцов**, канд. техн. наук, ген. директор,  
**П. А. Казанцев**, канд. техн. наук,  
 математик-программист,  
**М. А. Червоненкис**, вед. разработчик,  
 ООО "Павлин Технологии", www.pawlin.ru  
 e-mail: pantera2008@gmail.com

## Применение искусственных нейронных сетей для решения обратных задач в гидрологии

*Рассматриваются преимущества и недостатки нейросетевых подходов к решению обратных задач гидрологии, которые являются ярким примером задач математического моделирования сложных нелинейных систем. Приводятся общий метод решения подобных задач с помощью нейронных сетей и примеры решения обратных задач гидрологии и других областей физики.*

**Ключевые слова:** гидрология, нейронные сети, обратные задачи

### Введение

Обратные задачи гидрологии, в том числе и задачи краткосрочного прогнозирования паводков, по степени своей некорректности имеют много общего с прикладными задачами из других областей, таких как гидроакустика, электродинамика, прикладная спектроскопия, оптика, астрофизика и др. Все эти задачи объединяет важное свойство — крайне сложно формализуемая математическая модель того или иного объекта исследования. Для их решения пользуются значительными упрощениями модели объекта, при этом на решение накладываются стандартные регуляризирующие процедуры, адекватность которых в той или иной задаче редко аргументируется. Нейросетевой подход полагается, в первую очередь, на реальные экспериментальные данные, строя по ним аппроксимацию модели объекта. Ничто, однако, не препятствует тому, чтобы обучить нейронную сеть на данных строгой математической модели, дополненной реальными экспериментальными данными. Регуляризация же осуществляется путем, например, фиксации слабо варьируемых параметров объекта.

В настоящем исследовании проводится анализ нейросетевого подхода не только к решению обратных задач гидрологии, но и других сложных нелинейных динамических задач, готовые решения которых послужат заделом для разработки нейросетевого алгоритма решения задачи оперативного краткосрочного прогнозирования паводков.

### 1. Обратные задачи

Обратная задача — тип задач, в которых значения параметров модели должны быть получены из наблюдаемых данных.

Решение прямой задачи связано с отысканием следствия некоторого процесса по его известной причине. В отличие от прямых задач трудности решения обратных задач связаны с тем, что один и тот же эффект может быть порожден разными причинами. Если для решения прямой задачи достаточно знать одну причинно-следственную связь, то для успешного решения обратной задачи этого явно недостаточно и необходимо учитывать многие причинно-следственные связи, сопровождающие изучаемый процесс, и, следовательно, иметь гораздо более богатый опыт и экспериментальный материал, чем для решения прямой задачи.

Обратные задачи являются некорректно поставленными задачами. Из трех условий корректно поставленной задачи (существование решения, единственность решения и его устойчивость) в обратных задачах наиболее часто нарушается последнее. В этом случае возникает парадоксальная ситуация: несмотря на то, что задача математически сформирована, ее решение невозможно получить обычными методами. Действительно, какой смысл имеет решение, которое испытывает формально бесконечно большие возмущения при малых возмущениях результатов наблюдений, всегда получающихся с некоторой неопределенностью (ошибкой). Многие задачи обработки и интерпретации данных и результатов физических экспериментов являются обратными и, следовательно, некорректно поставленными. До появления современных научно обоснованных методов исследователь либо, используя детальную физическую модель изучаемого явления, сводил обратную задачу к нахождению небольшого числа параметров, либо, основываясь на физической интуиции, отбирал из множества допустимых решений то, которое лучше всего соответствует здравому смыслу. Однако такие результаты решения обратной задачи зачастую оставляют желать лучшего в плане точности: в первом случае часто бывает, что детальная физическая модель, допускающая жесткую параметризацию решения, неадекватна используемым наблюдательным данным (что приводит к большим остаточным отклонениям наблюдательных данных от теоретических предсказаний); во втором случае выбор решения субъективен, что нехарактерно для научного метода исследований.

Математически под обратной задачей понимается задача отыскания функции  $z(s)$  по функции

$u(x)$ , получаемой в результате эксперимента или наблюдений, из уравнения вида

$$u(x) = A[x, z(s)], \quad (1)$$

где  $A$  — некоторый оператор, устанавливающий причинно-следственную связь между  $z(s)$  и  $u(x)$ . В уравнении (1) по наблюдаемым следствиям  $u(x)$  процесса исследователь должен судить о причинах  $z(s)$ , породивших его.

Во многих случаях обратная задача (1) может быть представлена интегральным уравнением Фредгольма 1-го рода:

$$u(x) = \int_a^b K(x, s) \cdot z(s) ds, \quad (2)$$

где  $K(x, s)$  — ядро (непрерывное или квадратично суммируемое по переменным  $(x, s)$ ), которое описывает конкретную модель исследуемого процесса.

Математические трудности решения обратных задач связаны с тем, что обратный оператор  $A^{-1}$  (см. уравнение (1)) не является непрерывным. Поэтому если данные наблюдений  $u(x)$  получены с некоторой ошибкой (обозначим приближенные данные символом), то соответствующее приближенное решение, полученное стандартным методом,

$$z_{\delta}(s) = A^{-1}[u_{\delta}(x)], \quad (3)$$

будет сколь угодно сильно отклоняться от решения, соответствующего идеально точным входным данным  $u(x)$ .

Предлагаемые ранее методы решения обратных некорректных задач основывались, прежде всего, на интуиции авторов, однако позднее была создана математическая теория некорректно поставленных задач и разработаны эффективные методы их решения (регуляризирующие алгоритмы).

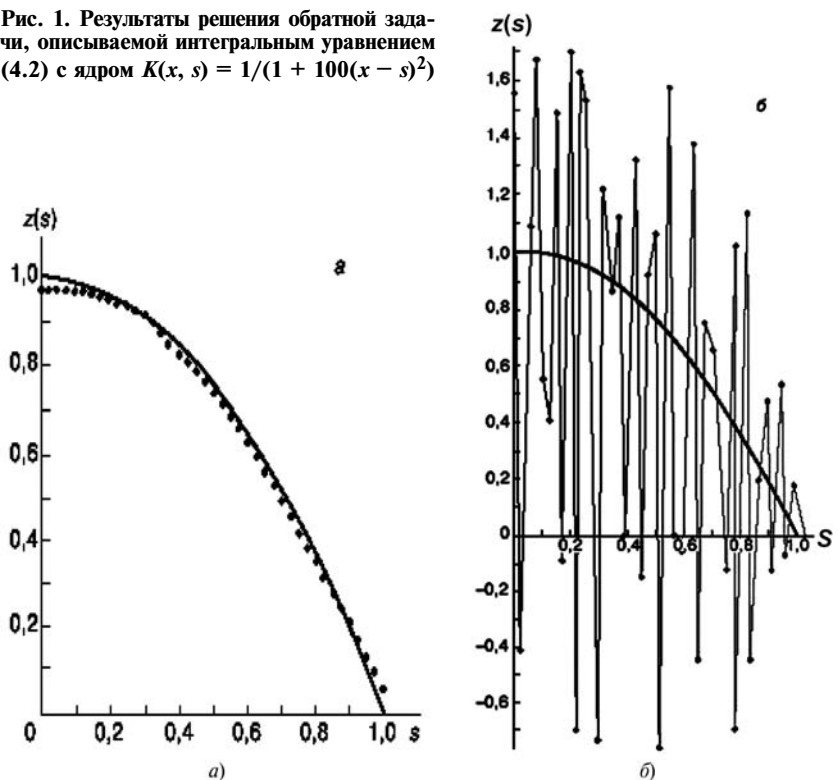
Некорректно поставленные задачи рассматриваются как физически недоопределенные. Они плохо поставлены, множества их приближенных решений очень широки, даже неограниченны. Поэтому некорректные задачи нужно доопределить. Для этого необходима дополнительная информация об искомом решении  $z(s)$ , вытекающая из обширного опыта всесторонних исследований данного процесса. Эта дополнительная информация об искомом решении должна быть известна *a priori*, до решения соответствующей некорректной задачи. Априорная информация позволяет сформулировать критерий отбора приближенного решения из множества приближенных решений уравнения (1) и построить регуляризирующий алгоритм.

Такой информацией могут служить априорные сведения о гладкости искомого решения  $z(s)$ , его монотонности, выпуклости, неотрицательности, принадлежности к конечно-параметрическому семейству и т. п.

Рис. 1, *a*, содержащий иллюстрационный пример из работы [1], отображает точное и приближенное решения некорректной задачи — интегрального уравнения Фредгольма 1-го рода (2), причем приближенное решение получено с помощью регуляризирующего алгоритма — сужения множества допустимых решений до множества выпуклых функций. Попытка решения этой же задачи без регуляризации является безуспешной (рис. 1, *б*).

На рис. 1 сплошной линией представлено точное решение  $z(s)$ , которое было задано заранее. Это решение подставлялось под знак интеграла в уравнение (2) и вычислялась соответствующая ему функция  $u(x)$  — идеально точные "входные данные" обратной задачи (2). Затем в полученную функцию  $u(x)$  вносилась погрешность, равная 3% от максимального значения, и решалась обратная задача: по возмущенной функции находилось приближенное решение. Приближенное решение (точки), представленное на рис. 1, *a*, получено с помощью регуляризирующего алгоритма, использующего априорную информацию о выпуклости искомого решения. При попытке решить эту же задачу без регуляризации (рис. 1, *б*) получаются сколь угодно большие отклонения "приближенного решения"

Рис. 1. Результаты решения обратной задачи, описываемой интегральным уравнением (4.2) с ядром  $K(x, s) = 1/(1 + 100(x - s)^2)$



(точки) от истинного. Так проявляется некорректность обратной задачи (2).

В настоящее время развитая теория решения некорректно поставленных задач успешно применяется для решения многих обратных задач электродинамики, гидрологии, гидроакустики, астрофизики, оптики и спектроскопии, оптимального планирования и т. п.

Важно отметить, что регуляризирующие алгоритмы гарантируют сходимость последовательности приближенных решений к точному решению обратной задачи, т. е. при стремлении ошибки наблюдений к нулю приближенное решение стремится к точному. Это дает веские основания считать, что приближенное решение некорректной задачи, полученное с помощью регуляризирующего алгоритма, близко к решению, соответствующему идеально точным данным наблюдений  $u(x)$ . Стихийные интуитивные методы решения обратных некорректных задач, отмеченные выше, не обладают этим достоинством.

## 2. Основы нейросетевого метода решения задач

В данном разделе приводится краткий обзор нейросетевого метода решения, в принципе, любых задач, в том числе и обратных.

Внутренние регуляризирующие особенности нейронных сетей позволяют решать также обратные и комбинированные задачи с локальной оценкой точности. Например, в работе [3] для некорректно поставленных задач моделирования предложена нейросетевая информационная технология построения гибридной нейроархитектуры, содержащей кластеризующую карту Кохонена и семейство сетей с обратным распространением, обучаемых на данных индивидуальных кластеров. В этой технологии выявляются области частичной корректности задачи, в которых дается решение с высокой локальной точностью. Для остальных областей признакового пространства нейросеть автоматически корректно отвергает пользовательские запросы.

Искусственные нейронные сети как инструмент для обработки и распознавания образов и изображений используются в задачах, которые плохо поддаются алгоритмизации, а также, когда входные данные не полны, зашумлены или противоречивы. Основными преимуществами нейронных сетей являются высокая помехоустойчивость, независимость от характера шума, гибкость и возможность аппроксимации любой многомерной функции.

Последнее свойство является следствием теоремы Колмогорова (1957 г.) [7], которая является математической основой нейронных сетей. Теорема Колмогорова утверждает, что любая непрерывная функция  $F$ , определенная на  $n$ -мерном единичном кубе, может быть представлена в виде суммы  $2n + 1$

суперпозиций непрерывных и монотонных отображений единичных отрезков:

$$F(x_1, \dots, x_n) = \sum_{q=1}^{2n+1} g_q \left( \sum_p \phi_{pq}(x_p) \right), \quad (4)$$

где  $x = (x_1, \dots, x_n)$ ,  $0 \leq x_i \leq 1$ . Слева в этой формуле стоит произвольная непрерывная функция, определенная на многомерном кубе, справа — функции, определенные на отрезках  $[0, 1]$ . Конечно, нужно иметь в виду, что отображения  $\phi_{pq}$  имеют достаточно сложную структуру. Заметим, что монотонные непрерывные отображения  $\phi_{pq}$  не зависят от конкретной функции  $F$ . В свете этой теоремы задача построения и обучения сети ставится на строго математическую основу.

Эта теорема означает, что для реализации функций многих переменных достаточно операций суммирования и композиции функций одной переменной. Конечно, применить эту теорему на практике достаточно сложно, поскольку ничего неизвестно о виде функций, входящих в выражение для  $F$ . Однако эта теорема показала принципиальную возможность реализации сколь угодно сложных зависимостей с помощью относительно простой нейронной сети, называемой многослойным персептроном, состоящим из многих нейронов, схема нейрона приведена на рис. 2. Математическая интерпретация нейрона выглядит следующим образом:

$$v_k = \sum_{j=0}^m w_{kj} x_j + b_k; \\ y_k = f(v_k), \quad (5)$$

где  $x_1, x_2, \dots, x_m$  — входные сигналы;  $w_{k1}, w_{k2}, \dots, w_{km}$  — синаптические веса  $k$ -го нейрона;  $v_k$  — линейная комбинация входных воздействий и порогового элемента  $b_k$ ;  $f$  — функция активации;  $y_k$  — выходной сигнал нейрона.

Схема нейронной сети показана ниже (рис. 3). На данной схеме  $\mathbf{W}$  — матрица весов связей между входными нейронами и нейронами скрытого слоя, которые собственно и реализуют функцию актива-

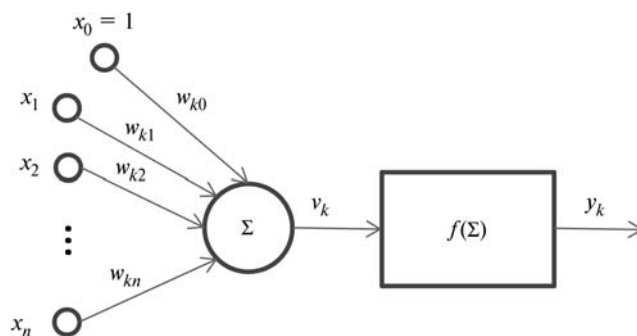


Рис. 2. Схема нейрона



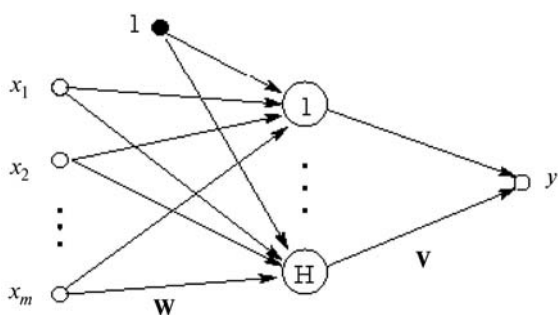


Рис. 3. Схема многослойного персептрона с одним выходом (входной слой, скрытый слой, выходной слой — один нейрон)

ции;  $V$  — матрица весов связей между выходами нейронов скрытого слоя и выходным нейроном сети. Собственно такую сеть называют трехслойным персептроном, имея в виду входной слой, выходной слой и скрытый слой нейронов, реализующих функцию активации. Такая сеть реализует следующее отображение:

$$y = \sum_{i=1}^H v_i \cdot f(w_{i,0} + w_{i,1}x_1 + \dots + w_{i,m}x_m), \quad (6)$$

где  $f$  — функция активации нейрона скрытого слоя.

Новый вариант теоремы Колмогорова был опубликован в 1989 г. одновременно несколькими авторами [8–10]. Теорема формулируется теперь в таком виде.

**Теорема.** Пусть  $F(x_1, x_2, \dots, x_m)$  — любая непрерывная функция, определенная на ограниченном множестве, и  $\varepsilon > 0$  — любое сколь угодно малое число, означающее точность аппроксимации. Обозначим еще через  $\sigma(x)$  сигмоидную функцию активации. Тогда существует такое число  $H$ , числа  $w_{ij}$ ,  $u_i$  и числа  $v_i$  такие, что функция

$$\begin{aligned} f(x_1, x_2, \dots, x_m) = \\ = \sum_{i=1}^H v_i \sigma(w_{i,1}x_1 + \dots + w_{i,m}x_m + u_i) \end{aligned} \quad (7)$$

приближает данную функцию  $F(x_1, x_2, \dots, x_m)$  с погрешностью не более  $\varepsilon$  на всей области определения.

Эта формула совпадает с функцией, реализуемой персептроном. В терминах теории нейросетей эта теорема формулируется так: любую непрерывную функцию нескольких переменных можно с любой точностью реализовать с помощью обычного трехслойного персептрона с достаточным числом нейронов в скрытом слое.

В общем случае многослойный персептрон (или просто нейронная сеть прямого распространения) может содержать несколько слоев скрытых нейронов и осуществлять отображение из одного векторного пространства признаков произвольной размерности в другое векторное пространство другой произвольной размерности.

С помощью нейронных сетей можно получить устойчивое решение обратных некорректных задач вследствие обобщающих свойств, в результате чего из сигнала выделяются наиболее характерные детали. Это позволяет эффективно отделять полезный сигнал от шума. К тому же устойчивость решения можно повысить путем включения методов регуляризации в процесс обучения сети.

Работа нейронной сети сводится к формированию некоторого полезного выходного набора данных в ответ на совокупность входных данных, т. е. реализации некоторого функционального преобразования данных.

Процесс настройки связей в целях придания сети необходимых свойств называется обучением нейронной сети. Наиболее распространенный метод обучения — обучение с учителем — предполагает наличие набора пар входных и соответствующих им желаемых выходных данных, называемого обучающей выборкой. Используя невязку реального и желаемого ответа сети для всех примеров обучающей выборки, оптимизируют связи сети, минимизируя эту невязку. Основополагающими работами по алгоритмам обучения нейронных сетей являются работы [11, 12], позже развитые в [13–15]. Для того чтобы сделать решение устойчивым к возможным шумам и ошибкам в измеренных данных, к входным данным обучающей выборки в процессе обучения может быть добавлена шумовая компонента.

### 3. Примеры постановок и решения обратных практических задач с помощью нейронных сетей

Как было отмечено выше, движение воды в речном русле представляет собой сложную динамическую систему, математическое описание которой затрудняется ввиду отсутствия однозначного понимания того, как те или иные факторы влияют на систему (осадки, рельеф дна, плотность грунта, и т. п.). Данный факт ставит обратные задачи гидрологии в один ряд с другими практическими задачами, в которых требуется описать сложный нелинейный объект, и также решается обратная задача.

Ранее авторами данного исследования был решен ряд таких задач и получен положительный результат, свидетельствующий об адекватном выборе пути решения. Опыт, полученный в ходе решения этих задач в части составления плана экспериментов, обучающей последовательности, выбора архитектуры сети, регуляризирующих ограничений, целесообразно применить при решении обратных задач гидрологии, адекватно проведя аналогии между основными этапами решения.

В начале описания решения данных задач приведен краткий анализ того, почему эти задачи являются сложно формализуемыми и схожи по сложности с задачами гидрологии, чем обосновывается уместность рассмотрения их решения в данной работе.

Как отмечалось выше, нейросетевые методы решения обладают определенной степенью универсальности. Меняя параметры нейросетевого алгоритма (число входов и выходов нейронной сети, число слоев и нейронов в них, функционал оптимизации и т. п.) и не меняя общей структуры алгоритма, можно строить нейросетевые решения задач согласно стандартному маршруту разработки [29], что значительно повышает ее эффективность и позволяет сравнивать нейросетевые алгоритмы по фиксированному числу параметров.

В работе [18] рассматривается нейросетевой метод решения задачи гидроакустики [16, 17], а в работах [19–21] — задачи синтеза микрополосковых антенн (электродинамика).

#### **4. Нейросетевые решения обратных задач для оперативных краткосрочных прогнозов паводков**

Выше были рассмотрены примеры решения практических обратных задач с помощью нейронных сетей. Аналогично, для решения обратных задач гидрологии существуют все предпосылки использования нейросетевого аппарата, главными из которых являются сложности в формализации самого физического явления, а также трудности с поиском регулирующих процедур. Это вызвано сложностью строения русел, которые имеют обширную пойму, большое число рукавов и протоков.

В зарубежных публикациях [26, 27], отмечаются следующие преимущества нейронных сетей (НС):

- НС способны распознавать зависимость между входными и выходными данными без рассмотрения явной физической зависимости;
- НС дают приемлемый результат, даже если обучающие выборки содержат шумы и ошибки измерения;
- НС обладают способностью адаптироваться к изменяемым условиям;
- НС, будучи обученными, просты в использовании.

В дополнении к указанному списку преимуществ добавим, что НС, как было показано в разд. 2, имеют существенно параллельную структуру (мелкозернистый параллелизм), что делает их распараллеливание на вычислительные средства гораздо более эффективным, чем классические алгоритмы решения.

Зачастую при решении задач гидрологии либо отсутствует полное понимание физического процесса, либо задача поставлена настолько некорректно, что использование физического описания этих задач не дает значимых результатов. Даже если и существуют некие физические модели, то они базируются на неких предположениях, что говорит в пользу нейросетевых методов. Более того, НС моделируют нелинейность процесса, лежащего в основе явления, избавляя разработчиков алгоритма от составления сложных систем дифференциальных уравнений. В отличие от регрессионных

методов при использовании НС не требуется формулировать предположения о математической форме зависимости между входами и выходами. Зашумленные данные обрабатываются НС без существенной потери точности за счет их распределенной архитектуры. Все это, а также нелинейность функций активации, обуславливает сильную обобщающую способность нейронных сетей, что делает их эффективным средством при решении широкого класса задач гидрологии.

Хотя исследования [26, 27] подтверждают эффективность нейронных сетей при решении задач гидрологии, не следует забывать и об их недостатках. Качество нейросетевого приложения зависит как от качества, так и от количества имеющихся данных. Это условие не всегда выполнимо, так как гидрологические архивы зачастую содержат недостаточно полную историю записей. Довольно часто требуемых данных нет в наличии, что обуславливает необходимость их генерации другими средствами, такими как, например, проверенная модель. Даже если в наличии имеется обширная история гидрологических записей, нельзя быть до конца уверенным, что прочие условия оставались неизменными за исследуемый промежуток времени. Поэтому желательна наличие данных наблюдения, измеренных на относительно стабильном объекте, который не подвергался влиянию человека. Представление вариаций во времени зачастую реализуется путем подачи предыдущих выходов и выходов на текущие (нейронная сеть с задержкой или сеть с обратной связью). Однако может быть совершенно не очевидным то, какое значение задержки следует избрать, чтобы смоделировать динамический эффект. Это приводит к усложнению архитектуры сети. Еще одним важным недостатком нейронных сетей является отсутствие в них физически понятных зависимостей. Это, впрочем, и было одной из главных причин скептического отношения к данной методологии. Тот факт, что отсутствует некий стандартизированный подход к выбору структуры нейронной сети, также является объектом критики. Выбор архитектуры, алгоритма обучения, функции ошибки зачастую определяется прежним опытом разработчика и его предпочтениями, а не физическими особенностями задачи.

Последний недостаток является следствием того, что разработка алгоритма настройки нейронной сети при нестандартном функционале ошибки (функционал вторичной оптимизации) представляет собой трудоемкий процесс даже для простой архитектуры сети (без обратных или перекрестных связей между слоями, и т. п.). В основном разработчики пользуются средней квадратичной ошибкой выходов, так как она достаточно просто реализуется программно, а также присутствует во многих математических пакетах и библиотеках. Эта проблема наиболее полно описана в монографии [28], в ко-

торой подробно разбираются принципы синтеза оптимальных моделей нейронных сетей, включая алгоритм настройки и архитектуру.

Так, обучаемая на реальных данных нейронная сеть, в результате будет представлять собой аппроксимацию математической зависимости физических величин, а регуляризирующие процедуры могут быть заданы не математическими функционалами, адекватность которых может быть сомнительна в применении к данной конкретной задаче, а физическими ограничениями на диапазоны варьирования тех или иных параметров (в предельном случае — их фиксация).

Для повышения точности прогноза уровня воды требуется максимальное уточнение геометрии русла, что на практике вызывает безусловные трудности ввиду его постоянной изменчивости. В нейросетевой постановке задачи зависимость от геометрии русла, подобно зависимости от рельефа дна в случае моря в задаче, рассмотренной в работе [18], может быть задана неявно в генерируемых примерах обучающей выборки. То же самое может быть осуществлено и с другими параметрами системы, влияющими на уровень воды.

Аналогично задачам, рассмотренным в работах [18—21], для генерации обучающей последовательности потребуется некая математическая модель, учитывающая все эти параметры и реализующая решение прямой задачи. Программное же обеспечение, реализующее эту модель, должно обладать возможностью варьировать эти параметры в выбранных диапазонах. Так, например, в программе целесообразно ввести возможность генерации русел рек, задавая параметры рукавов и стоков, а также их местоположение, аналогично и с погодными условиями, осадками. Для задач прогноза необходимо иметь возможность варьирования интервала сбора архивных данных и интервала прогнозирования. На выходе же программы будет получен набор примеров — соответствий значений  $Q$  (вход нейронной сети) и значений  $H$  (вход нейронной сети), который целесообразно дополнить реальными экспериментальными данными, полученными с реальных объектов. Обучаясь на такую последовательность, нейронная сеть построит функцию

$$H = F(Q). \quad (8)$$

Входы нейронной сети  $H$  и  $Q$  могут иметь различный состав, от которого будет зависеть конкретная постановка задачи. Так, например, можно представить их наборами значений соответствующих величин, увязанных в один вектор как  $H = (h_1, \dots, h_K)$  и  $Q = (q_1, \dots, q_M)$  при этом дискретности  $h$  и  $q$ , могут быть разделены как пространственно, так и по времени, а также по обоим измерениям. В общем случае:  $h_k = p(x_{hk}, t_{hk})$  и  $q_m = g(x_{qm}, t_{qm})$ .

С учетом (5), (6) для многослойной нейронной сети с  $K$  выходами

$$h_k = f \left( \sum_{n_L=1}^{N_L} w_{n_L k} \cdot \dots f \left( \sum_{n_2=1}^{N_2} \left[ w_{n_2 n_1} f \left( \sum_{n_1=1}^{N_1} [w_{n_1, m} q_m] + w_{n_1, 0} \right) + w_{n_2, 0} \right] + w_{n_L 0} \right), \quad (9)$$

где  $L$  — число слоев нейронной сети;  $N_l$  — число нейронов в слое  $l$ ;  $f$  — нелинейная непрерывно дифференцируемая функция активации нейрона, например сигмоида:

$$f = \frac{1}{1 + e^{-Ax}}. \quad (10)$$

Согласно теореме Колмогорова (см. разд. 2) и (7), варьируя  $L$ ,  $N_l$  и, возможно  $f$ , используя данные, сгенерированные с помощью математической модели, решающей прямую задачу, и дополненные реальными данными с объектов, несущих неявным образом информацию о шумах измерений, можем получить динамическую модель для предсказания уровня воды в зависимости от измеряемых параметрах стока. Зависимости  $h_k$  и  $q_m$  подлежат экспериментальному исследованию для различных объектов.

Данный подход — это полное нейросетевое представление системы. Возможен, однако, и другой способ, в котором с помощью нейронной сети аппроксимируются только части зависимости (или нескольких частей). Можно показать, что в уравнении Сен-Венана можно выделить несколько функциональных зависимостей, таких как, например, зависимость  $F(x, H)$  — площадь поперечного сечения. В различных работах были сделаны попытки аппроксимации этой зависимости полиномами различных степеней. Полиномиальные зависимости в этих формулах могут быть заменены нейросетевыми аппроксимациями, аналогичными (9), построенными по реальным данным, только вместо  $q_m$  в формуле будут фигурировать входы  $x$  и  $H$ .

Таким образом, уравнение Сен-Венана будет содержать члены, вычисленные аналитически и аппроксимированные нейронными сетями. Такой подход возможно даст большую точность, поскольку, варьируя число нейросетевых аппроксимаций в формуле, можно пошагово отбрасывать неудачные варианты, которые возникнут вследствие некачественного обучения нейронной сети, вызванного, например, недостаточностью данных для обучения.

В работе [27] рассматривается применение нейросетевого метода с вейвлетной предобработкой для предсказания паводков на один день вперед, с использованием экспериментальных данных, полученных на реках Лиф-ривер (приток Миссисипи, США) и Серен (приток реки Йонны, впа-

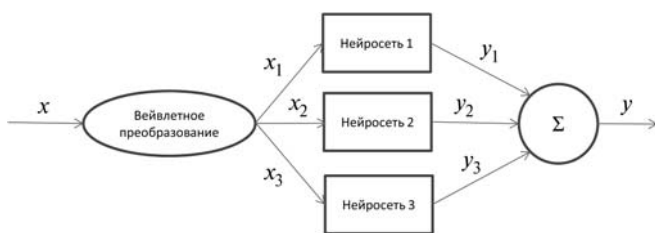


Рис. 4. Схема алгоритма с применением вейвлетной преобразования для ансамбля сетей

дающей в Сену). В данной работе с помощью вейвлета Морле временной ряд экспериментальных данных разлагается в частотный спектр, после чего спектр разбивается на три интервала, из которых составляются обучающие выборки для трех различных нейронных сетей (рис. 4). Нейронные сети обучаются независимо, после чего объединяющим элементом выносятся решение.

В результате экспериментов сделан вывод о небольшом преимуществе нейросетевого подхода перед классическим методом прогнозирования, однако авторы работы упоминают о возможном несовершенстве разработанного метода и перспективах продолжения разработки.

Использование предварительной фильтрации данных перед их подачей на нейронную сеть подлежит отдельному исследованию.

### Заключение

Преимуществом нейросетевого подхода является способность нейронных сетей аппроксимировать сложные нелинейные динамические зависимости без знания структурных данных об объекте моделирования. Однако важным шагом при построении нейросетевой модели является создание репрезентативной обучающей выборки. Отсутствие согласованных экспериментальных данных негативно влияет на способность нейронной сети к отображению входного вектора в выходной, так как нейронные сети не используют какой-либо математической трактовки физического процесса, с помощью которой можно было бы скомпенсировать недостаток информации. Для устранения этого недостатка целесообразно провести эксперименты с комбинированной обучающей выборкой, часть которой состоит из данных, построенной на основе адекватной математической модели, а другая часть — из экспериментальных данных, несущих информацию о шумах измерений и реальном объекте моделирования. Как показывает практика решения обратных задач с помощью нейронных сетей из других областей, такой подход значительно повышает точность решения, но требует дополнительных усилий по разработке математической модели объекта (возможно с определенными упрощениями) и генерации объемной обучающей выборки на основе

данной модели. Важным моментом при разработке нейросетевой модели системы является выбор регуляризирующих факторов, как, например, фиксация определенных параметров объекта, которые имеют гораздо больший период изменения, нежели физические величины, подаваемые на вход сети. По данной проблеме целесообразно провести исследование глубины исторического ряда реальных данных, необходимой для качественного обучения нейронной сети, а также зависимость от нее периода прогнозирования.

### Список литературы

1. Черепашук А. М. Обратные задачи в астрофизике. Мир науки и культуры, 2009.
2. Горбань А. Н., Дунин-Барковский В. Л., Кирдин А. Н. и др. Нейроинформатика. Новосибирск: Наука, 1998. 296 с.
3. Севастьянов А. А. Решение обратных некорректных задач в прикладной спектроскопии с помощью вейвлет-анализа и нейронных сетей: дис. ... канд. физ.-мат. наук, Казань, 2004.
4. Корженевский А. В. Нейросетевые алгоритмы решения обратных задач радиочастотной томографии // Нейрокомпьютеры: разработка и применение. 2002. № 9—10. С. 26—33.
5. Adler A. and Guardo R. A neural network image reconstruction technique for electrical impedance tomography // IEEE Trans. Med. Imag. 1994. V. 13. P. 594—600.
6. Севастьянов А. А., Харинцев С. С., Салахов М. Х. Нейросетевая регуляризация решения обратных некорректных задач прикладной спектроскопии // Исследовано в России. Электронный журнал. 2003. № 189. С. 2254—2266. URL: <http://zhurnal.apel.rn.ru/articles/2003/1/89.pdf>
7. Колмогоров А. Н. О представлении функций многих переменных в виде суперпозиции непрерывных функций одного переменного и операции сложения // ДАН СССР. 1957. Т. 114. Вып. 57.
8. Hornik K., Stinchcombe M., White H. Multilayer Feedforward Networks are Universal Approximators // Neural Networks. 1989. V. 2. № 5.
9. Cybenko G. V. Approximation by Superpositions of a Sigmoidal Function // Mathematical Control Signals Systems. 1989. N 2.
10. Funahashi K. On the Approximate Realization of Continuous Mappings by Neural Networks // Neural Networks. 1989. V. 2. № 3.
11. Галушкин А. И. Синтез многослойных систем распознавания образов. М.: Энергия, 1974.
12. Werbos P. J. Beyond regression: New tools for prediction and analysis in the behavioral sciences: Ph. D. thesis. Harvard University, Cambridge, MA, 1974.
13. Rumelhart D. E., Hinton G. E., Williams R. J. Learning Internal Representations by Error Propagation // Parallel Distributed Processing. Vol. 1. Cambridge, MA, MIT Press, 1986. P. 318—362.
14. Барцев С. И., Охонин В. А. Адаптивные сети обработки информации. Красноярск: Ин-т физики СО АН СССР, 1986. Препринт N 59Б. 20 с.
15. Барцев С. И., Гилев С. Е., Охонин В. А. Принцип двойственности в организации адаптивных сетей обработки информации // Динамика химических и биологических систем. Новосибирск: Наука, 1989. С. 6—55.
16. Акустика океана / Под ред. Л. М. Бреховских. М.: Наука, 1974.
17. Бреховских Л. М., Лысанов Ю. П. Теоретические основы акустики океана. Ленинград: Гидрометеиздат, 1982.
18. Казанцев П. А. Определение глубины источника в морском пространстве по гидроакустическим параметрам среды с помощью нейронных сетей // Труды 9-й Международной конференции DSPA'07. М.: ИПУ РАН, 2007. С. 543—546.
19. Казанцев П. А. Нейросетевой синтез микрополосковой антенны с возбуждением коаксиальным зондом // Труды 9-й Международной конференции DSPA'07. М.: ИПУ РАН, 2007. С. 537—539.
20. Казанцев П. А. Построение моделей микрополосковых антенн с помощью нейронных сетей прямого распространения //

Нейрокомпьютеры: разработка и применение. 2007. № 9, приложение (CD).

21. Галушкин А. И., Казанцев П. А., Козлов К. В., Лодягин А. М., Лось В. Ф. Нейросетевой синтез микрополосковой антенны, возбуждаемой коаксиальным зондом // Антенны. 2007. № 9.

22. Barber D. C. and Brown V. H. Applied potential tomography // J. Phys. E: Sci. Instrum. 1984. V. 17. P. 723—733.

23. Корженевский А. В., Черепенин В. А. Магнитоиндукционная томография // РЭ. 1997. Т. 42. N 4. С. 506—512.

24. Корженевский А. В., Черепенин В. А. Магнитоиндукционная томография // Журнал радиоэлектроники. 1998. № 1. URL: <http://jre.cplire.ru/win/dec98/1/text.html>

25. Anctil F., Tape D. G. An exploration of artificial neural network rainfall—runoff forecasting combined with wavelet decomposition // J. Environ. Eng. Sci. 2004. Vol. 3 (Suppl. 1).

26. Artificial neural networks in Hydrology: By the ASCE Task Committee on Application of Artificial Neural Networks in Hydrology // Journal of Hydrologic Engineering. April 2000. P. 115—123.

27. Галушкин А. И. Теория нейронных сетей. М.: Издательское предприятие редакции журнала Радиотехника, 2000.

28. Казанцев П. А. Разработка и исследование маршрута проектирования нейросетевого приложения с аппаратной поддержкой: дис. ... канд. техн. наук по специальности 05.13.01. Москва, 2008.

УДК 004.942; 004.896

**В. В. Борисов**, д-р техн. наук, проф.,

**А. Е. Мисник**, аспирант,

Филиал МЭИ в г. Смоленске,

e-mail: [vborisov@etna-it.ru](mailto:vborisov@etna-it.ru), [anton@misnik.by](mailto:anton@misnik.by)

## Комбинированный нейросетевой способ моделирования для оперативного управления сложными системами

*Предложен комбинированный нейросетевой способ моделирования для оперативного управления сложными системами. Способ основан, во-первых, на предварительной декомпозиции и построении аналитической модели функционирования системы, во-вторых, на построении и предварительной настройке нейросетевых моделей отдельных элементов системы, в-третьих, на создании комбинированной нейросетевой модели системы, в-четвертых, на дообучении нейросетевых моделей отдельных элементов системы и модели системы в целом в процессе ее функционирования (управления функционированием) нейронной сетью-супервизором. Данный способ реализован при моделировании и управлении функционированием участков системы городского теплоснабжения.*

**Ключевые слова:** нейросетевая модель, нейросетевой способ моделирования

Сложные организационно-технические системы имеют следующие особенности: сложность структуры, многокомпонентность, большое число параметров и переменных; постоянное изменение структуры и параметров; неполнота исходной информации о системе; разнообразие и вероятностный характер воздействий; наличие сложных нелинейных зависимостей; необходимость принятия решений в реальном времени; ограниченные возможности экспериментального исследования.

Ограниченность возможностей управления такими системами обуславливает необходимость исполь-

зования моделирования в "контуре" оперативного управления. Однако невозможность охвата различных событий и явлений в процессе функционирования таких систем приводит к нецелесообразности создания и использования традиционных моделей этих процессов и модели системы в целом [1]. Конструктивным подходом к преодолению этой проблемы является декомпозиция системы на элементы, создание моделей элементов, построение комбинированной нейросетевой модели системы и адаптация модели к изменяемым условиям в процессе управления функционированием системы. Для декомпозиции системы, идентификации ее элементов и инвариантного отображения "топологии" системы предлагается использовать структурные модели в виде ориентированных графов, а для описания функциональной и динамической модели — эквивалентные схемы. Модели элементов предлагается формировать узловым методом, используя данные из матрицы инцидентности. На основе полученной матрицы строится система уравнений [2].

Как правило, такая система уравнений является довольно громоздкой и результаты ее решения не могут быть использованы для управления функционированием системы в режиме реального времени в силу того, что ее решение занимает весьма существенное время.

Однако результаты решения системы уравнений могут быть использованы для первичного обучения нейросетевых моделей, которыми можно описывать функционирование системы. Такие нейросетевые модели определяются набором входных и выходных параметров, характеризующих систему, а внутренняя их топология определяется эвристически на основе соотношения входных и выходных параметров. Входные параметры должны представлять собой как статические, так и динамические характеристики системы, зависящие от режимов ее функционирования.

Результаты решения системы уравнений в разных начальных условиях не позволяют максимально

точно обучить нейросетевые модели. Поэтому дообучение нейросетевых моделей необходимо проводить с использованием реальных данных о функционировании системы.

Комбинируя и объединяя нейросетевые модели элементов сложной системы, можно осуществить композицию модели всей системы.

Для проверки корректности композиции нейросетевых моделей предлагается использовать нейронную сеть-супервизор, входами которой являются входные параметры всех моделей элементов системы, а выходами — выходные модели, характеризующие ее целевые параметры.

Сформированная библиотека моделей элементов системы представляет собой совокупность типизированных нейросетевых моделей, преобученных до состояния, в котором из них можно выполнить композицию модели системы в целом. В дальнейшем, с учетом реальных данных о функционировании системы, нейросетевые модели элементов должны быть дообучены для того, чтобы адекватно отображать не только структуру, но и функционирование системы.

Нейронная сеть-супервизор через заданные промежутки времени должна переобучаться для идентификации возможных структурных и параметрических изменений в системе.

Таким образом, предлагаемый комбинированный способ учитывает особенности традиционного и нейросетевого моделирования сложных систем и включает в себя следующие основные этапы.

Этап 1. Выделение элементов системы, определение их параметров и взаимосвязей.

Этап 2. Построение структурной модели системы и матрицы инцидентности.

Этап 3. Построение системы уравнений на основе матрицы инцидентности.

Этап 4. Получение решений системы уравнений при различных начальных условиях.

Этап 5. Выбор типизированных нейросетевых моделей для элементов системы.

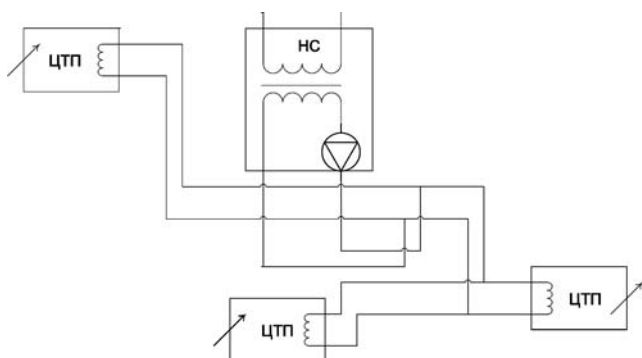


Рис. 1. Упрощенная модель тепловой сети

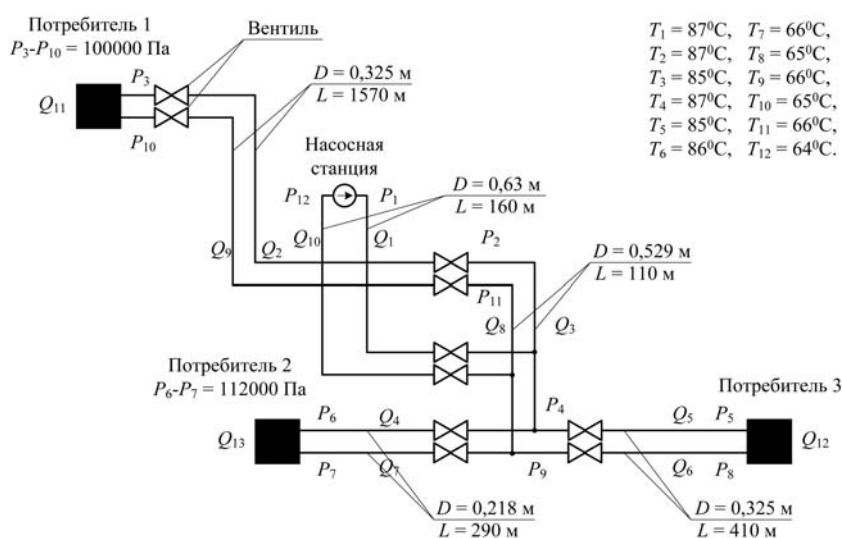


Рис. 2. Схема теплосети

Этап 6. Обучение нейросетевых моделей на основе решений системы уравнений и начальных условий.

Этап 7. Композиция нейросетевых моделей элементов в единую модель системы.

Этап 8. Дообучение нейросетевых моделей на основе данных о функционирующей системе.

Этап 9. Формирование нейросетевого супервизора и его обучение.

По результатам использования данного способа осуществляется управление функционированием системы.

Данный способ реализован при моделировании и управлении функционированием участками теплосети г. Могилева. Рассмотрим упрощенную модель теплосети, включающую в себя насосную станцию (НС) и трех потребителей — центральные тепловые пункты (ЦТП) (рис. 1).

После того как построена матрица инцидентности, на ее основе строится система уравнений. Для лучшего представления об элементах системы покажем дополнительную информацию (рис. 2).

После построения матрицы, на ее основе строится система уравнений Бернулли, описывающая состояние системы:

$$z_1 - z_2 + \frac{P_1 - P_2}{\rho g} = \left( 0,11 \left( 68 \frac{\eta_1 \pi d_1^2}{4 Q_1 d_{r1} \rho} + \frac{\Delta_1}{d_1} \right)^{0,25} \cdot \frac{l_1}{d_1} + \xi_1 \right) \frac{8 Q_1^2}{g \pi^2 d_1^4},$$

$$z_2 - z_3 + \frac{P_2 - P_3}{\rho g} = \left( 0,11 \left( 68 \frac{\eta_2 \pi d_2^2}{4 Q_2 d_{r2} \rho} + \frac{\Delta_2}{d_2} \right)^{0,25} \cdot \frac{l_2}{d_2} + \xi_2 \right) \frac{8 Q_2^2}{g \pi^2 d_2^4},$$

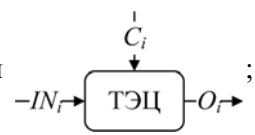
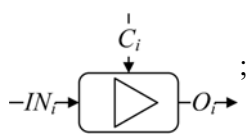
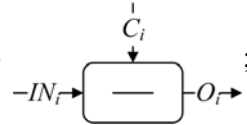
$$Q_{13} - Q_7 = 0, Q_{12} - Q_6 = 0, Q_6 + Q_7 - Q_8 = 0, \\ Q_{11} - Q_9 = 0, Q_8 + Q_9 - Q_{10} = 0,$$

$$P_{12} = 250000, Q_1 = 1000, Q_{10} = 1000,$$

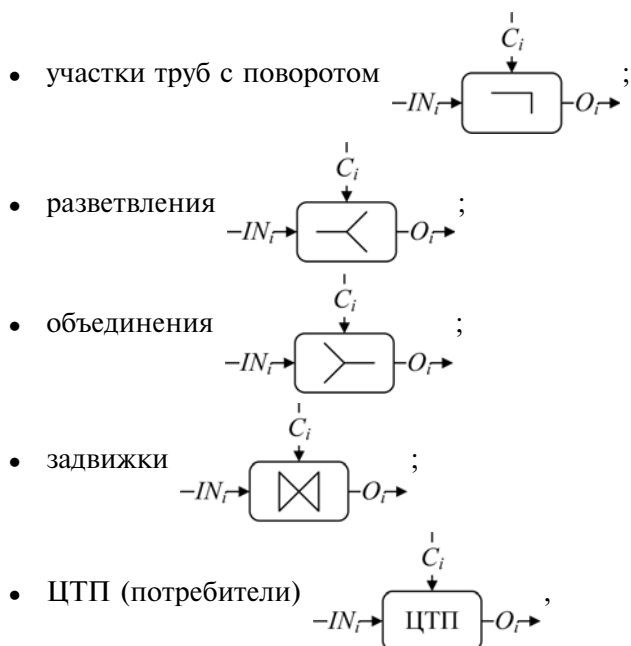
где  $z_i$  — высота над условной поверхностью (например над уровнем моря) точки теплосети, м;  $P_i$  — давление внутри теплосети в заданной ее точке, Па;  $Q_i$  — расход теплоносителя в контуре, участке теплосети между двумя заданными точками, м<sup>3</sup>/ч;  $d_i$  — диаметр поперечного сечения контура (по умолчанию, трубы), м;  $d_{r_i}$  — гидравлический диаметр поперечного сечения контура (по умолчанию равен диаметру), м;  $l_i$  — длина контура, м;  $\eta_i$  — динамический коэффициент вязкости теплоносителя, Па·с;  $\Delta_i$  — значение шероховатостей внутри контура, м;  $\xi_i$  — местные потери давления в контуре, вызванные изменением геометрической формы контура, Па;  $\rho$  — плотность теплоносителя, кг/м<sup>3</sup>;  $g$  — ускорение свободного падения 9,81 м/с<sup>2</sup>.

Индекс  $i$ , стоящий при переменных  $P$  и  $z$ , обозначает номер теплоузла: точка разветвления контуров, точка соединения контура и потребителя, контура и насосной станции. Индекс  $i$ , стоящий при переменных  $Q$ ,  $d$ ,  $l$ ,  $d_r$ ,  $\eta$ ,  $\Delta$ ,  $\xi$ , обозначает номер участка теплосети, соединяющего две точки.

Для решения системы уравнений используется метод градиентного спуска. Таким образом, варьируя исходные данные, можно получить расчетные значения состояний системы. Для реализации нейросетевых моделей используются динамические библиотеки, экспортированные из системы MATLAB. Для работы модели системы каждый ее элемент описывается отдельной нейронной сетью, входы и выходы которых соединены исходя из матрицы инцидентий, полученной ранее. Для моделирования структурных элементов систем теплоснабжения разработана библиотека следующих типизированных нейросетевых моделей:

- источники теплоты  ;
- повысительная насосная станция (ПНС)  ;
- прямые участки труб  ;

$$z_2 - z_4 + \frac{P_2 - P_4}{\rho g} = \\ = \left( 0,11 \left( 68 \frac{\eta_3 \pi d_3^2}{4 Q_3 d_{r3} \rho} + \frac{\Delta_3}{d_3} \right)^{0,25} \cdot \frac{l_3}{d_3} + \xi_3 \right) \frac{8 Q_3^2}{g \pi^2 d_3^4}, \\ z_4 - z_6 + \frac{P_4 - P_6}{\rho g} = \\ = \left( 0,11 \left( 68 \frac{\eta_4 \pi d_4^2}{4 Q_4 d_{r4} \rho} + \frac{\Delta_4}{d_4} \right)^{0,25} \cdot \frac{l_4}{d_4} + \xi_4 \right) \frac{8 Q_4^2}{g \pi^2 d_4^4}, \\ z_4 - z_5 + \frac{P_4 - P_5}{\rho g} = \\ = \left( 0,11 \left( 68 \frac{\eta_5 \pi d_5^2}{4 Q_5 d_{r5} \rho} + \frac{\Delta_5}{d_5} \right)^{0,25} \cdot \frac{l_5}{d_5} + \xi_5 \right) \frac{8 Q_5^2}{g \pi^2 d_5^4}, \\ z_8 - z_9 + \frac{P_8 - P_9}{\rho g} = \\ = \left( 0,11 \left( 68 \frac{\eta_6 \pi d_6^2}{4 Q_6 d_{r6} \rho} + \frac{\Delta_6}{d_6} \right)^{0,25} \cdot \frac{l_6}{d_6} + \xi_6 \right) \frac{8 Q_6^2}{g \pi^2 d_6^4}, \\ z_7 - z_9 + \frac{P_7 - P_9}{\rho g} = \\ = \left( 0,11 \left( 68 \frac{\eta_7 \pi d_7^2}{4 Q_7 d_{r7} \rho} + \frac{\Delta_7}{d_7} \right)^{0,25} \cdot \frac{l_7}{d_7} + \xi_7 \right) \frac{8 Q_7^2}{g \pi^2 d_7^4}, \\ z_9 - z_{11} + \frac{P_9 - P_{11}}{\rho g} = \\ = \left( 0,11 \left( 68 \frac{\eta_8 \pi d_8^2}{4 Q_8 d_{r8} \rho} + \frac{\Delta_8}{d_8} \right)^{0,25} \cdot \frac{l_8}{d_8} + \xi_8 \right) \frac{8 Q_8^2}{g \pi^2 d_8^4}, \\ z_{10} - z_{11} + \frac{P_{10} - P_{11}}{\rho g} = \\ = \left( 0,11 \left( 68 \frac{\eta_9 \pi d_9^2}{4 Q_9 d_{r9} \rho} + \frac{\Delta_9}{d_9} \right)^{0,25} \cdot \frac{l_9}{d_9} + \xi_9 \right) \frac{8 Q_9^2}{g \pi^2 d_9^4}, \\ z_{11} - z_{12} + \frac{P_{11} - P_{12}}{\rho g} = \\ = \left( 0,11 \left( 68 \frac{\eta_{10} \pi d_{10}^2}{4 Q_{10} d_{r10} \rho} + \frac{\Delta_{10}}{d_{10}} \right)^{0,25} \cdot \frac{l_{10}}{d_{10}} + \xi_{10} \right) \frac{8 Q_{10}^2}{g \pi^2 d_{10}^4}, \\ P_3 - P_{10} = 100000, P_5 - P_8 = 120000, \\ P_6 - P_7 = 112000, \\ Q_1 - Q_2 - Q_3 = 0, Q_2 - Q_{11} = 0, \\ Q_3 - Q_4 - Q_5 = 0, Q_5 - Q_{12} = 0, Q_4 - Q_{13} = 0,$$



где  $C_i$  — число характеристик элемента (доступная информация об объекте: материал труб, степень износа, глубина, высота местности и т. п.; информация о состоянии окружающей среды), подающиеся на входы нейронной сети элемента;  $IN_i$  — число входов нейронной сети элемента;  $O_i$  — число выходов нейронной сети элемента. Входами  $IN_i$  и выходами  $O_i$  являются следующие данные: давление; расход теплоносителя; температура [3].

Топология нейронной сети определяется следующим образом [4]. Число нейронов во входном слое задается числом входов:

$$NC_1 = C_i + IN_i.$$

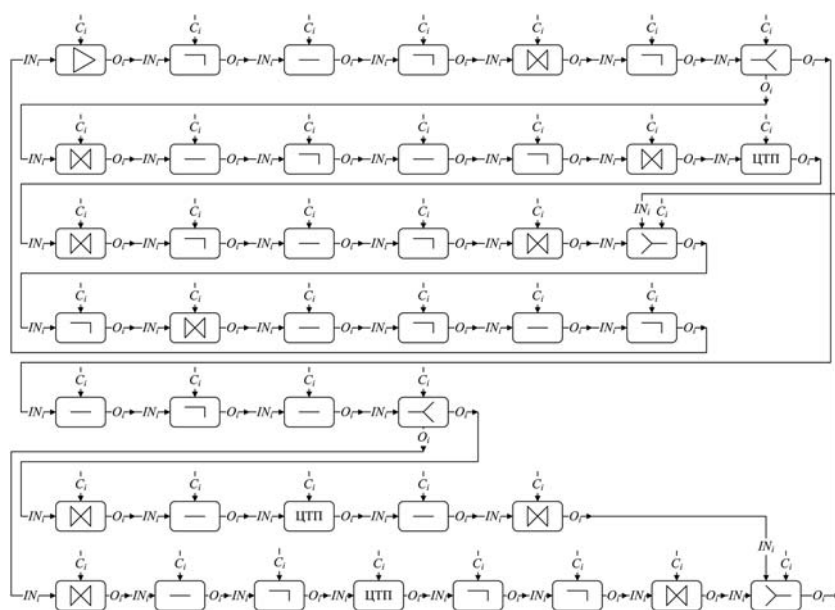


Рис. 3. Схема соединения нейросетевых моделей

### Перечень входов $C_i$ типизированных нейросетевых моделей

Название модели	Набор входов $C_i$
Источник теплоты	Исходная температура; давление, создаваемое насосами
ПНС	Давление, создаваемое насосами
Прямой участок трубы	Материал, изоляция, год прокладки, длина
Участок с поворотом	Материал, изоляция, год прокладки, длина, угол поворота
Разветвление	Материал, изоляция, год прокладки, длина, входной диаметр, выходной диаметр первой трубы, выходной диаметр второй трубы
Объединение	Материал, изоляция, год прокладки, длина, выходной диаметр, входной диаметр первой трубы, входной диаметр второй трубы
Задвижка	Материал, тип, изоляция, год установки
ЦТП	Диаметр шайбы

Число нейронов в выходном слое задается числом выходов  $O_i$ . Первоначальное число промежуточных слоев нейронной сети

$$L = C_i + IN_i - O_i - 1.$$

Число нейронов в промежуточном слое

$$NC_j = C_i + IN_i - j, j \in [1, L].$$

Перечень входов  $C_i$  типизированных нейросетевых моделей представлен в таблице.

На рис. 3 показана композиция отдельных нейросетевых моделей в модель системы.

При структурных изменениях в системе осуществляется добавление новых элементов в существующую модель, после чего она сразу готова к использованию. Дообучение нейросетевых моделей выполняется уже в работающей модели системы на основе информации о ее текущем состоянии. Нейронная сеть-супервизор и предельно допустимые данные, полученные по структурно-матричному методу, используются для идентификации возможных ошибок обучения либо несоответствия структуры системы.

### Список литературы

1. Марка Д., МакГоуэн К. Методология структурного анализа и проектирования. М.: МетаТехнология, 1993. 240 с.
2. Тарасик В. П. Математическое моделирование технических систем: учебник для вузов. Минск: ДизайнПРО, 2004. 640 с.
3. Борисов В. В., Мисник А. Е. Иерархическая нейросетевая модель систем теплоснабжения // Сб. трудов Междунар. науч.-техн. конф. "Энергетика, информатика, инновации — 2011 — ЭИИ-2011". В 2 т. Т. 2. Смоленск: РИО филиала ГОУВПО МЭИ (ТУ) в г. Смоленске, 2011. С. 205—207.
4. Хайкин С. Нейронные сети: полный курс, 2-е изд., испр. М.: Вильямс, 2006. 1104 с.



**С. Ю. Степанов,**

канд. техн. наук, зам. нач. отдела,  
Администрация городского округа Климовск,

**И. С. Кабак,**

канд. техн. наук, доц., проф.,  
Московский государственный  
университет "СТАНКИН",  
Московский авиационный институт,  
e-mail: ikabak@mail.ru.

## Алгоритм фрагментации больших нейронных сетей и исследование его сходимости

*При разработке больших адаптивных систем управления автономными объектами, такими как беспилотные космические станции, все чаще применяют системы с искусственным интеллектом, в частности, с использованием технологии искусственных нейронных сетей. Искусственная нейронная сеть для сложной задачи будет состоять из большого числа нейронов, и современные технологии микроэлектроники не позволяют выполнить ее как один элемент. Для создания аппаратного комплекса, реализующего нейронную сеть, необходима фрагментация сети на отдельные подсети ограниченного размера. От качества разбиения сети в существенной степени зависит быстрдействие, стоимость и другие параметры. Рассмотрен алгоритм такой фрагментации и исследуется его сходимость.*

**Ключевые слова:** объемно-модульные системы, коммутаторная нейронная сеть, доменная сеть, фрагментация, сходимость

### Введение

В военной промышленности, авиации и космонавтике необходимы:

- обработка звуковых сигналов с их разделением и идентификацией;
- распознавание целей и их идентификация по сигналам радаров;
- обработка инфракрасных сигналов и сигналов видимой части спектра и ультрафиолетовых сигналов;
- обобщение полученной информации;
- распознавание образов;
- классификация;
- адаптивное управление;
- прогнозирование поведения объекта;
- диагностика;
- техническое зрение;
- автоматическое пилотирование.

Все вышеперечисленные задачи могут быть решены на базе интеллектуальных технологий, осно-

ванных на применении искусственных нейронных сетей.

В настоящее время в большинстве ведущих индустриально развитых стран ведется разработка систем управления сложными техническими объектами на базе нейронных сетей. Нейронные сети состоят из большого числа параллельно работающих крайне простых вычислительных элементов (нейронов). Нейроны взаимодействуют между собой и образуют нейронную сеть — совокупность информационно связанных между собой нейронов. Нейроны выполняют достаточно простые одинаковые действия, вычисляя функцию активации по входной информации и передавая ее значение связанным с ними нейронам. Достаточно большое число и параллельность работы нейронных структур обеспечивают высокое быстрдействие.

Искусственные нейронные сети являются перспективным научным направлением, позволяя решать ряд задач, которые традиционные компьютеры либо вообще не могут решить, либо решают неэффективно. Существенной особенностью нейронных структур является иной способ использования компьютера. Традиционно компьютер выполняет программу, определяющую процесс решения задачи. Для нейронных сетей используется обучение. Обучение — определение весов, или масштабных коэффициентов, с которыми выходы одного нейрона передаются другим. Обученная нейронная сеть применяет полученные знания к новым входным значениям, при этом и создаются возможности решения интеллектуальных задач.

Разработки в области нейрокомпьютеров поддерживаются целым рядом международных и национальных программ. В настоящее время эксплуатируются не менее 50 нейросистем в самых различных областях — от финансовых прогнозов до экспертизы [1, 2].

Исследования в области современных нейронных сетей проводят многие коммерческие компании, научно-исследовательские лаборатории и университеты. К коммерческим компаниям относятся Micro Devices (США), AT&T (США), Hitachi (Япония), Adaptive Solutions (США), NeuroLogix Bellcore (США), Echelon (США), INTEL (США), Texas Instruments (США), IBM (США), Computer Recognition Systems (США), причем модель CRS 1000 (Computer Recognition Systems) уже нашла применение в промышленных системах автоматического контроля, Siemens и др.

К научно-исследовательским лабораториям и университетам в первую очередь можно отнести Naval Lab (США), MIT Lab (США), а также университеты: Пенсильванский, Колумбийский, Иллинойский, Аризонский (США), Берлинский тех-

нический университет, Технический университет в Карлсруе (ФРГ), Российские университеты и др.

Для обработки нейронной сети обычно используют специальные компьютеры и достаточно редко универсальные суперкомпьютеры, например, "Кремниевый мозг" для обработки аэрокосмических изображений (производительность 80 Пфлопс, объем оперативной памяти равен объему мозга человека [9]). Для задач управления автономными аппаратами использование универсальных суперкомпьютеров достаточно проблематично по экономическим и техническим (габаритным, весовым, энергетическим) причинам. В данной статье в качестве аппаратной базы для реализации искусственной нейронной сети предложена специальная многопроцессорная архитектура, построенная исходя из необходимого числа модулей трех типов, имеющих унифицированный интерфейс и одинаковые размеры.

Высокое быстродействие модульной системы достигается за счет большого числа параллельно работающих аппаратных модулей. Нейронная сеть разбивается на ряд подсетей, каждая из которых имеет ограниченное число нейронов. Ограничение числа нейронов позволяет разделить нейронную сеть на несколько (достаточно большое число) отдельных модулей, при этом связность нейронной сети и ее обучение сохраняются. Сама процедура создания объемно-модульной системы, ее состав и описание приведены в работах [5—7].

Модули объемной системы связаны между собой специальной системой транспорта информации (сетью). Для успешного функционирования такого транспорта желательна особая структура нейронной сети, когда отдельные нейронные подсети подобны как отдельному нейрону, так и всей сети в целом. Таким образом, сеть и ее элементы являются самоподобными объектами, или фракталами.

Каждый базовый модуль системы хранит отдельную подсеть и выполняет ее функции. При этом кроме собственно нейронов подсеть должна быть дополнена устройством для связи нейронов данной подсети с нейронами других подсетей. Такое устройство было названо шлюзом домена и кроме вышеупомянутой связи позволяло иметь различные адресные пространства нейронов внутри и вне подсети. Мы разбиваем нейронную сеть на ряд автономных фрагментов — доменов. Структура доменов одинакова, каждый домен имеет свою внутреннюю адресацию. Важно, что домен может находиться в аппаратном модуле полностью, ограничения на его размер выбираются исходя из возможностей модуля. Технология доменных сетей приведена в работе [6]. Нейронная сеть в целом представлена как совокупность доменов-модулей. В свою очередь домен также является совокупностью нейронов.

Существенным для обучения, модернизации и реализации такой системы является другой подход

к видению нейронной сети. В традиционных нейронных сетях нейрон выполняет две функции:

- функцию активации — расчет выходного значения по сумме входных значений;
- транспорт информации к связанным с ним нейронам.

Введем в нейронную сеть дополнительный элемент — коммутатор, обеспечивающий транспорт информации между нейронами, освободив нейрон от транспортной задачи. Тогда можно показать, что нейрон такой сети может иметь только один вход и один выход, полностью структурно совпадая с доменом. Накопление информации для входа нейрона осуществляет коммутатор, у которого для каждого нейрона имеется аддитивный буфер, так что мультиплицирование входов и выходов обеспечивает коммутатор. Структура нейронной сети и масштабные коэффициенты передачи информации между нейронами хранятся в служебных таблицах коммутаторов.

Базовая нейронная сеть — нейронная сеть, которую обслуживает один коммутатор. Масштабирование нейронной сети осуществляется увеличением числа базовых нейронных сетей. Для интеграции отдельных базовых сетей в единую нейронную сеть используются коммутаторы верхнего уровня. Такая нейронная сеть была названа коммутаторной [6]. В коммутаторной технологии структура связей в нейронной сети представляет собой древовидную структуру.

При связи нейронов между собой путь передаваемой информации (трафик) существенно зависит от взаимного расположения нейронов в древовидной структуре. Чем дальше нейроны расположены друг от друга, тем больше трафик передаваемой информации. Правомерна задача оптимизации трафика информации в нейронной сети за счет рационального расположения нейронов на коммутаторах. Внутри одной базовой нейронной сети сократить трафик невозможно. Переставляя нейроны между различными базовыми нейронными сетями, можно существенно снизить трафик в нейронной сети. Важным аспектом этой задачи является логическая группировка нейронов, по результатам которой возможно создание отдельных групп, отвечающих за отдельные подзадачи общей задачи нейронной сети.

Рациональная группировка нейронов дает возможность по обученной нейронной сети определить, какие подмножества нейронов отвечают за отдельные подзадачи, поскольку существование более или менее изолированной группы нейронов позволяет говорить, что данная группа выполняет более или менее изолированную подзадачу общей задачи. При этом чем меньше связей эта группа имеет с другими группами нейронов, тем более изолированную подзадачу она выполняет.

Для нейронных сетей небольшого размера данную задачу группирования можно решить методом перебора. Для нейронных сетей, размер которых превышает несколько тысяч нейронов, решение этой задачи будет затруднительно. Возникает необходимость в разработке алгоритма группирования нейронов для сокращения трафика в коммутаторной нейронной сети.

### Алгоритм фрагментации нейронной сети

Представим нейронную сеть в виде двух пересекающихся множеств: множество объектов, передающих информацию (имеющих выходы), и множество объектов, являющихся получателями информации. Эти множества пересекаются (заштрихованная область на рис. 1), но в общем случае не совпадают, поскольку в задаче сокращения трафика информации внешние связи нейронной сети не учитываются.

Для каждого элемента из множества  $A$  существует правило, по которому он связан с одним и более элементами из множества  $B$ . Это правило задается структурой нейронной сети и зависит от наличия и направления связи между нейронами. Для описания архитектуры коммутаторной нейронной сети необходимо использовать множество  $C$  (множество коммутаторов) — элементов, передающих информацию. Таким образом для каждого правила связи элементов множеств  $A$  и  $B$  в нейронной сети будет существовать правило связи элемента множества  $A$  с последовательным рядом элементов из множества  $C$ , последний из которых связан с одним из элементов множества  $B$ . Последовательности связей элементов множества  $C$  между собой определяются древовидной структурой коммутаторной нейронной сети.

Требуется сократить общее число элементов из множества  $C$ , необходимых для осуществления передачи информации от элементов множества  $A$  к элементам множества  $B$ .

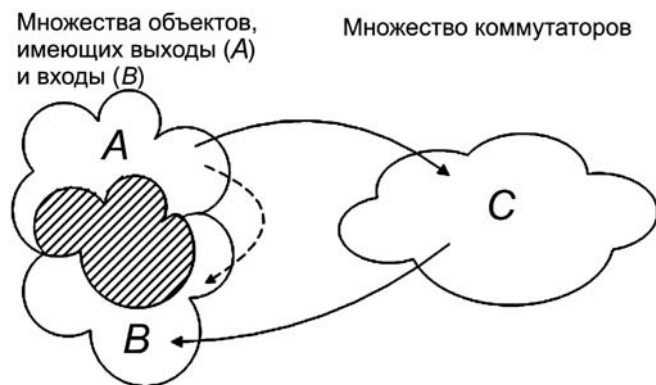


Рис. 1. Представление коммутаторной нейронной сети в виде множеств объектов и коммутаторов

Был разработан алгоритм для решения задачи группировки нейронов на группы заданного размера [3].

Рассмотрим плоскость, где расположена координатная сетка, в узлах которой в произвольном порядке размещена группа объектов, соответствующих нейронам сети. Введем импульс притяжения между объектами таким образом, что если между двумя нейронами в сети существует связь, то эти две вершины притягиваются друг к другу. Импульс притяжения должен быть постоянным для каждой пары объектов.

Для предотвращения слияния объектов в общее ядро введем импульс отталкивания между всеми объектами. Этот импульс обратно пропорционален расстоянию между объектами. Для того чтобы объекты могли объединяться в группы, следует уменьшить импульс отталкивания, если объекты сблизятся на определенное расстояние.

Определив результирующий импульс, получим направление движения для каждого объекта и значение перемещения. Через некоторое время объекты, имеющие между собой импульсы притяжения, должны оказаться рядом на плоскости. Изменяя параметры импульсов притяжения и отталкивания, можно варьировать число и размеры получаемых элементарных сегментов.

### Теоретическая сходимость алгоритма

Введем обозначения:

$A$  — коэффициент притяжения, постоянный положительный;

$W(i, k)$  — вес связи между объектами  $i$  и  $k$  в рассматриваемой нейронной сети;

$B$  — коэффициент отталкивания, постоянный положительный;

$S(i, k)$  — расстояние между объектами  $i$  и  $k$  на плоскости;

$P_{\text{прит}}$  — импульс притяжения  $[\text{кг} \cdot \text{м}/\text{с}] = A|W(i, k)|$ ;

$P_{\text{отт}}$  — импульс отталкивания  $[\text{кг} \cdot \text{м}/\text{с}] = B/S(i, k)$ ;

$t$  — время, такт исполнения алгоритма, принято  $t = 1$ ;

$P = mV$ ;

$m$  — масса одного объекта;  $m$  равно числу нейронов сети, входящих в объект;

$V$  — скорость движения объекта при работе алгоритма;

$(P_{\text{прит}}/m)t$  — расстояние, на которое приблизятся объекты;

$(P_{\text{отт}}/m)t$  — расстояние, на которое отдалятся объекты;

$S_0$  — текущее расстояние;

$S_1$  — итоговое расстояние;

$\Delta S$  — суммарное перемещение всех объектов на плоскости;

$\Delta S(i)$  — вектор перемещения объекта  $i$  на плоскости;

$S_0(i, k)$  — начальное расстояние между объектами  $i$  и  $k$  на плоскости;

$-1(i, k)$  — единичный вектор, определяющий положение объектов  $i$  и  $k$  относительно осей координат при отталкивании;

$\Delta x$  — суммарное перемещение всех объектов по оси  $OX$ ;

$\Delta y$  — суммарное перемещение всех объектов по оси  $OY$ ;

$x(i)$  — начальная координата объекта  $i$  на оси  $OX$ ;

$x(k)$  — начальная координата объекта  $k$  на оси  $OX$ ;

$y(i)$  — начальная координата объекта  $i$  на оси  $OY$ ;

$y(k)$  — начальная координата объекта  $k$  на оси  $OY$ ;

$\Delta x(i)$  — перемещение объекта  $i$  по оси  $OX$ ;

$\Delta y(i)$  — перемещение объекта  $i$  по оси  $OY$ ;

$\Delta x(i, k)$  — перемещение объекта  $i$  относительно объекта  $k$  по оси  $OX$ ;

$\Delta y(i, k)$  — перемещение объекта  $i$  относительно объекта  $k$  по оси  $OY$ ;

$S(i, k)_1$  — расстояние между объектами  $i$  и  $k$  на первом такте;

$S(i, k)_2$  — расстояние между объектами  $i$  и  $k$  на втором такте;

$\Delta S(i, k)$  — изменение расстояния между объектами  $i$  и  $k$ ;

$\Delta S(i, k)_1$  — изменение расстояния между объектами  $i$  и  $k$  на первом такте;

$\Delta S(i, k)_2$  — изменение расстояния между объектами  $i$  и  $k$  на втором такте;

$$S_1 = S_0 - (P_{\text{прит}}/m)t + (P_{\text{отт}}/m)t = S_0 + (t/m)(B/S_0 - A).$$

Покажем, что через определенное число тактов алгоритм в соответствии с начальными условиями и заданными ограничениями позволит получить результирующее положение объектов на плоскости, определяющее принадлежность объектов к тем или иным группам.

Каждый объект имеет координаты  $(x, y)$  на плоскости. Расстояние  $S$  между любыми двумя объектами  $i$  и  $k$  определяется формулой

$$S(i, k) = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}.$$

Для того чтобы система находилась в состоянии покоя, необходимо чтобы импульсы притяжения и отталкивания были уравновешены, в таком случае суммарное перемещение всех объектов на плоскости будет равняться нулю:

$$\Delta S = \sum_{i=1}^N \Delta S_i = \sum_{i=1}^N \sum_{k=1}^N \left[ S_0(i, k) + \left( \frac{B}{S_0(i, k)} - A|W(i, k)| \cdot (-1(i, k)) \right) \right] = 0.$$

Суммарные перемещения всех объектов по осям координат будут определяться формулами

$$\Delta x = \sum_{i=1}^N \sum_{k=1}^N \left[ \left( \frac{B}{\sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}} - A|W(i, k)| \right) \frac{x_i - x_k}{\sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}} \right];$$

$$\Delta y = \sum_{i=1}^N \sum_{k=1}^N \left[ \left( \frac{B}{\sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}} - A|W(i, k)| \right) \frac{y_i - y_k}{\sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}} \right].$$

Коэффициенты притяжения, отталкивания и группирования в общем случае принимаются константами, вес связей является постоянным и не изменяется. Для пар объектов, не имеющих связей, вес связи между ними принят равным нулю. Для того чтобы суммарное перемещение объектов со временем стало нулевым, необходимо чтобы на каждом шаге  $\Delta S$  уменьшалось.

Таким образом, перемещение по каждой из осей координат должно уменьшаться с каждым тактом расчета действующих импульсов и перемещения объектов по плоскости. Объекты на плоскости могут располагаться произвольным образом с соблюдением единственного необходимого условия — изолированности, непринадлежности к группам.

В общем случае при соблюдении начальных условий задачи получим следующее:

$$\frac{B}{\sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}} - A|W(i, k)| < 0 \text{ при } W(i, k) > 0;$$

при  $x(i) > x(k); y(i) > y(k)$  получим  $\Delta x(i) < 0; \Delta y(i) < 0$ ;

при  $x(i) > x(k); y(i) < y(k)$  получим  $\Delta x(i) < 0; \Delta y(i) > 0$ ;

при  $x(i) < x(k); y(i) > y(k)$  получим  $\Delta x(i) > 0; \Delta y(i) < 0$ ;

при  $x(i) < x(k); y(i) < y(k)$  получим  $\Delta x(i) > 0; \Delta y(i) > 0$ .

Следовательно, для любого соотношения координат  $(x(i), y(i))$  и  $(x(k), y(k))$  справедлива формула

$$S(i, k)_2 = S(i, k)_1 + \Delta S(i, k) = \sqrt{(x_i - x_k + \Delta x_{ik})^2 + (y_i - y_k + \Delta y_{ik})^2} < S(i, k)_1.$$

При выполнении условия  $B/S > A$ :  $x(i) \rightarrow x(k)$ ;  $y(i) \rightarrow y(k)$  и  $\Delta S(i, k)_2 < \Delta S(i, k)_1$ .

При нарушении этого условия получаем следующее:

при  $x(i) > x(k); y(i) > y(k)$  получим  $\Delta x(i) > 0; \Delta y(i) > 0$ ;

при  $x(i) > x(k); y(i) < y(k)$  получим  $\Delta x(i) > 0; \Delta y(i) < 0$ ;

при  $x(i) < x(k); y(i) > y(k)$  получим  $\Delta x(i) < 0; \Delta y(i) > 0$ ;

при  $x(i) < x(k); y(i) < y(k)$  получим  $\Delta x(i) < 0; \Delta y(i) < 0$ .

Для выражения

$$\Delta S(i, k) = \left[ \frac{B}{S_0(i, k)} - A|W(i, k)| \cdot (-1(i, k)) \right]$$

справедливо неравенство

$$\Delta S(i, k)_2 < \Delta S(i, k)_1.$$

Вне зависимости от расстояния при применении законов движения  $x = f(x, y)$ ;  $y = f(x, y)$  на каждом такте получим, что:

$$\lim_{t \rightarrow \infty} \Delta S(i) = \frac{BN}{\sum_{k=1}^N S(k)} - A \sum_{k=1}^N |W(i, k)| \cdot (1(i, k)) = 0;$$

$$\lim_{t \rightarrow \infty} \Delta x =$$

$$= \sum_{i=1}^N \sum_{k=1}^N \left[ \left( \frac{B}{S(i, k)} - A|W(i, k)| \right) \left( \frac{x_i - x_k}{S(i, k)} \right) \right] = 0;$$

$$\lim_{t \rightarrow \infty} \Delta y =$$

$$= \sum_{i=1}^N \sum_{k=1}^N \left[ \left( \frac{B}{S(i, k)} - A|W(i, k)| \right) \left( \frac{y_i - y_k}{S(i, k)} \right) \right] = 0.$$

Рассматриваем случай, когда все объекты являются нейронами нейронной сети, соответственно у каждого объекта имеется хотя бы одна связь с другим объектом, все объекты образуют цепь таким образом, что от каждого объекта можно найти такую последовательность связанных с ним объектов, что можно построить соединение каждый — каждый. В противном случае будет рассматриваться задача группирования двух и более независимых нейронных сетей. При этом  $\Delta S(i, k)_2 < \Delta S(i, k)_1$ , но итоговое  $\Delta S(i, k)$  никогда не примет значения, равного нулю. Результирующее расстояние с остальными объектами для каждого элемента определяется формулой

$$\lim_{t \rightarrow \infty} \sum_{k=1}^N S(i, k) = \frac{BN}{\sum_{k=1}^N (A|W(i, k)| \cdot |-1(i, k)|)}.$$

Таким образом, при соблюдении начальных условий (при группировании нейронной сети) алгоритм является детерминированным, понятным, завершаемым, массовым и в общем смысле результативным. Результативность алгоритма будет определяться соотношением коэффициентов  $A$ ,  $B$  и  $C$ . Регулируя значения коэффициентов, можно разбивать нейронную сеть на группы разного размера.

### Экспериментальная сходимость алгоритма

Проверим экспериментально полученные выводы. При моделировании алгоритма на персональном компьютере вычисления являются диск-

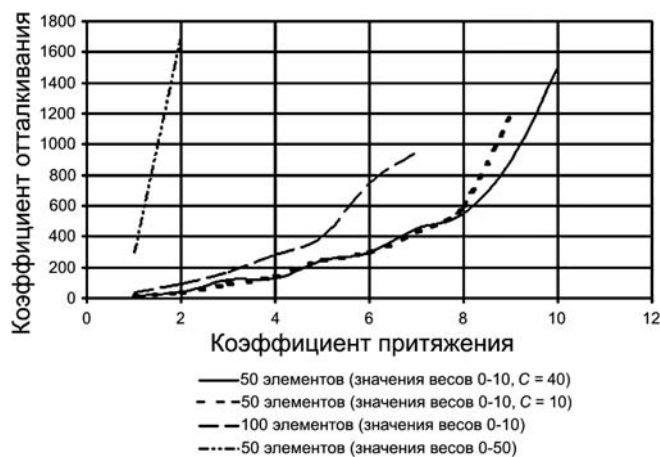


Рис. 2. Границы экспериментальных сходимостей алгоритма в зависимости от коэффициентов отталкивания и притяжения

ретными. Были проведены эксперименты по определению экспериментальной сходимости алгоритма. Полученные результаты иллюстрирует рис. 2. Кривая отмечает экспериментальные границы сходимостей, ниже которых алгоритм сходится.

Уменьшив такт, компенсируем дискретность вычислений. Это аналогично кратному сокращению коэффициентов  $A$  и  $B$ . В таком случае пропорционально уменьшаются величины  $\Delta x$  и  $\Delta y$ , и для любой пары коэффициентов алгоритм сходится.

### Заключение

В работе [4] показана эффективность работы алгоритма при группировании нейронной сети с числом связей в пределах одного порядка от числа объектов. Эффективность алгоритма существенно зависит от структуры и размеров нейронной сети, числа и величин связей. В ходе проведения экспериментальных исследований полносвязных нейронных сетей, включающих число нейронов примерно  $10^2$ , результирующее число линий передачи данных составляло от 0,97 до 0,8 от первоначального значения. Для больших нейронных сетей, состоящих из сотен и тысяч нейронов, сокращение числа связей в коммутаторной нейронной сети зависит от соотношения числа связей и числа элементов. И при равном соотношении при увеличении размеров сети эффективность работы алгоритма увеличивается.

Используя разработанный алгоритм фрагментации нейронной сети, можно решить задачу декомпозиции нейронной сети на автономные группы, необходимой для перехода от структуры коммутаторной нейронной сети к доменной нейронной сети [5, 6]. С учетом рассмотренного можно сделать вывод о том, что разработанный алгоритм фрагментации нейронной сети может применяться для задачи сокращения трафика в коммутаторной архитектуре

нейронных сетей. Использование данного алгоритма позволяет при построении искусственных нейронных сетей большого размера на основе коммутаторной архитектуры сократить затраты на передачу информации между нейронами и число информационных связей, используя переход у доменной архитектуре нейронной сети.

#### Список литературы

1. **Кабак И. С., Суханова Н. В.** Нейронная сеть. Патент на ПМ № 75247.
2. **Теория** нейронных сетей: учеб. пособие для вузов. Кн. 1 / Под общей ред. А. И. Галушкина. М.: ИПРЖР, 2000. 416 с.

3. **Степанов С. Ю.** Снижение трафика информации в коммутаторной нейронной сети на основе ее фрагментации // Тр. XVI Междунар. научно-техн. конф. "Информационные средства и технологии." В 3 т., Т. 3. М.: Изд. МЭИ. 2008. С. 236—241.

4. **Степанов С. Ю.** Группирование нейронов в двухуровневой коммутаторной нейронной сети // Тр. Междунар. научно-техн. конф. "Информационные средства и технологии." 16—18 октября 2007. В 3-х т. Т. 3. М.: МЭИ. 2007. С. 178—181.

5. **Кабак И. С.** Доменная организация коммутаторных нейронных сетей // Тр. Междунар. научно-техн. конф. "Информационные средства и технологии." 16—18 октября 2007. В 3-х т. Т. 3. М.: МЭИ. 2007. С. 128—131.

6. **Кабак И. С., Суханова Н. В.** Доменная нейронная сеть. Патент на ПМ № 72084 РФ.

7. **Кабак И. С., Суханова Н. В.** Модульная вычислительная система. Патент на ПМ № 75247 РФ.

## CONTENTS

### **Kukhareno V. G., Ponomarev D. I.** *Gaussian Mixture Approximation in Switching Kalman Filter Model for Identification of Time Series Oscillation Regimes.* . . . . . 2

Time series of oscillation regime changes are under study. Auto regression models of the Prony method in state space are in use in switching Kalman filter as dynamical models of time series oscillation regimes. Hidden state probabilities are approximated by Gaussian mixtures. The detection of repeated pattern (operator gesture) successions is demonstrated in the control signal of remote manipulator.

**Keywords:** time-series, oscillation regime, the Prony method, autoregression model, switching Kalman filter, Gaussian mixture model (GMM), pattern, telerobotic manipulator

### **Andreev D. V.** *Universal Logical Module for Conversion of Multiple-Valued Data* . . . . . 7

The article considers mathematical model, element basis and circuit decision of the new analog module, universal in a class of all functions from  $n$  arguments of  $k$ -valued logic. The way of minimization of the specified model at corresponding restriction of a class of reproduced functions is presented.

**Keywords:** the multiple-valued logic, multilevel signals, universal modules

### **Sanduleanu L. N., Strijov V. V.** *Feature Selection in Autoregression Forecasting.* . . . . . 11

The authors investigate the optimal model selection problem with application to the auto-regression forecasting. To solve the problem one has to select a maximum well-defined feature subset, subject to some given value of the error function. To select the feature set the modified add-del feature selection algorithm is used. This paper suggests a method of time series forecasting model selection. The computational experiment compares the electricity hourly prices forecasts.

**Keywords:** feature selection, multicollinearity, stepwise regression, Belsley method, time series forecasting

### **Chepurko V. A., Chepurko S. V.** *Geometrical Processes Degradation Coefficient Non-Parametric Estimation* . . . . . 16

Article deals with the construction of the point and interval non-parametric estimates of the denominator of the geometric process. Using the interval estimation it is possible to test the hypothesis that the observed recovery process is geometric. The investigated model is linearized by taking the logarithm of the geometric process. The proposed method is based on the slope nonparametric Theil estimation of simple linear regression.

**Keywords:** geometric process, renewal process, denominator of the process, slope, nonparametric Theil estimation, Kendall statistics

<b>Otsokov Sh. A.</b> <i>Method of Checking Necessity of the Rounding Off During the High-Precision Calculations in Modular Arithmetic</i> . . . . .	21
--	----

In article the method of check of necessity of a rounding off during high-precision calculations with binary fractions in modular number system is introduced. Experimental estimations of efficiency of high-precision calculations in modular arithmetic on an example of a finding of scalar product of vectors are offered.

**Keywords:** modular arithmetic, binary fractions, high-precision calculations

<b>Kosiakov S. V., Gadalov A. B., Sadykov A. M.</b> <i>Simulation of Spatial Data Solving Discrete Optimization Problems in a GIS</i> . . . . .	27
---	----

The problems and methods of calculating and maintaining in actual state matrix of transportation costs using GIS are described. Matrix is used as input data in vehicle routing problems and optimal facility location problems. Describes the method of reducing the time a full matrix routing transportation task and the database structure that allows to automate the process of updating this matrix in the GIS.

**Keywords:** spatial data, matrix routes, GIS, vehicle routing problem

<b>Astafurov V. G., Skorokhodov A. V.</b> <i>Neural Network Cloudiness Classifier from Satellite Data</i> . . . . .	32
---	----

An algorithm for cloudiness type classification based on three-layer perceptron with the use of information on the satellite image texture is suggested. Based on the expert estimate, 14 unique textures of different cloudiness types have been identified that can be identified using the suggested classifier. The Gray-Level Co-occurrences Matrix Approach based on calculation of adjacency matrices of brightness gradation is used to describe the cloud image texture. The network architecture and the algorithm of its learning are described. Results of cloudiness classification are discussed.

**Keywords:** classification, clouds, neural networks, textural parameters

<b>Krivoshein D. Yu., Marchenko A. M.</b> <i>Incremental All-Pairs Shortest Path Algorithm</i> . . . . .	38
--	----

In this paper the dynamic shortest path problem on graphs is considered. For static graphs various classic algorithms are known, such as Floyd-Warshall all-pairs shortest path algorithm or Dijkstra's single-source shortest path algorithm. In the paper a new incremental algorithm to recalculate shortest paths between all pairs of vertices in graph is presented for the case when the weight of one edge is increasing. This algorithm can be used when a large number of shortest path recalculation is performed, e.g. in routing phase of VLSI design, where graphs may have a large amount of vertices and classic algorithms are ineffective. This algorithm was also applied on sparse graphs, resulting in  $O(n^{4/3})$  time complexity.

**Keywords:** dynamically changing graph, all-pairs shortest path algorithm, incremental algorithm, Floyd-Warshall algorithm, VLSI routing

<b>Martynov P. N., Kovshov E. E.</b> <i>Automation User Interface Research Development in Man-Machine Control Systems</i> . . . . .	42
---	----

Discussing research methods of the software interface quality and their applicability in the field of human-machine control systems. The author's technique and developed software package are invited to use as an alternative to existing interface testing methods.

**Keywords:** man-machine interaction, graphical interface, usability testing

<b>Brazhnik S. A., Malafeev S. I.</b> <i>Synchronisation in Industrial Controllers with Operating System Linux</i> . . . . .	47
--	----

Using of program system of synchronisation of time in industrial controllers with operating system Linux in structure of Automatic system for commercial accounting of power consumption is considered. As an exact time source receivers GPS serve. Synchronisation signals are used for correction of time and fine tuning of industrial controller system clocks with a margin error, not exceeding 1,3 msec.

**Keywords:** automatic control system, automatic system for commercial accounting of power consumption, Linux, time, synchronisation

<b>Vyatkin V. B.</b> <i>Information-Quantum Characteristics and the Reflected Images of Final Sets</i> . . . . .	50
--	----

Quantum aspects of the information reflected by final sets of elements from positions synergistic information theory are considered. Such concepts, as quanta of the information and quanta of reflection, a bits of reflection and their quantum capacity, the reflected images of sets and information borders of their existence are thus entered into consideration.

**Keywords:** quantity of the information, syntropy, bit, quantum, final set, reflected image

**Fedorova E. V., Getman M. A., Savelieva E. V.** *Use of Google Apps Education Edition Cloud Technologies to Set a Self-Developing Informational Platform in a Higher Education Institution* ..... 57

This article surveys the experience of setting-up and use of the general information platform in Saint-Petersburg Chemical and Pharmaceutical Academy based upon Google Apps Education Edition, the open-coded cloud web-information technologies. Main stages of development and introduction of the informational platform, its structure and use benefits of web-based cloud computing applications in educational process are elucidated.

**Key words:** Google Apps Education Edition, common information space, corporate information systems (CIS), information technologies for education

**Skribtsov P. V., Kazantsev P. A., Chervonenkis M. A.** *Artificial Neural Networks Approaches to Inverse Hydrology Problems* ..... 62

This article examines advantages and drawbacks of neural network approaches to inverse hydrology problems which represent a good example of mathematical modeling of complex nonlinear systems. A generalized neural network approach to this problem to such a problems is presented. Solutions to other physical problems are referred as suggestive examples of successful approaches to ill-posed problems.

**Keywords:** hydrology, neural networks, inverse problems

**Borisov V. V., Misnik A. E.** *Combined Neural Network Method of Modelling for the Operative Control of Complex Systems* ..... 69

The combined neural network method of modelling for the operative control of complex systems is offered. This method is based: first, on preliminary decomposition and design of analytical model of functioning of system; secondly, on design and initial tune-up neural network models of separate elements of system, thirdly, on design of combined neural network model of system; fourth, on post-training of neural network models of separate elements and of neural network model of system model during its functioning (management of functioning) by means of a neural network-supervisor. This method is realised at modelling and management of functioning of system of a city heat supply.

**Keywords:** neural network, neural network method of modelling

**Stepanov S. Yu., Kabak I. S.** *Convergence Analysis of Commutated Neural Networks Traffic Reduction Algorithm* ..... 73

This paper gives a brief description of the goal of reducing traffic information in the commutated neural network, neural network fragmentation algorithm and the proof of convergence of the algorithm.

**Keywords:** commutated neural network, fragmentation, convergence

---

---

**Адрес редакции:**

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *Е. В. Конова*.

Корректор *М.Г. Джавадян*.

Сдано в набор 10.05.2012. Подписано в печать 21.06.2012. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 9,8. Заказ ИТ712. 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.

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".

105120, г. Москва, ул. Нижняя Сыромятническая, д. 5/7, стр. 2, офис 2.