

# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 25

2019

№ 7

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

САПР

КОМПЬЮТЕРНАЯ ГРАФИКА

МЕТОДЫ ПРОГРАММИРОВАНИЯ

ОПЕРАЦИОННЫЕ СИСТЕМЫ И СРЕДЫ

ТЕЛЕКОММУНИКАЦИИ  
И ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

НЕЙРОСЕТИ И  
НЕЙРОКОМПЬЮТЕРЫ

СТРУКТУРНЫЙ СИНТЕЗ

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ  
СИСТЕМЫ

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

ОПТИМИЗАЦИЯ И МОДЕЛИРОВАНИЕ

ИТ В ОБРАЗОВАНИИ

ГИС



Рисунки к статье А. В. Щёкина

## «КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКАЯ ПАРАМЕТРИЗАЦИЯ В СОСТАВЕ ИНТЕГРИРОВАННОЙ САМ-СИСТЕМЫ»

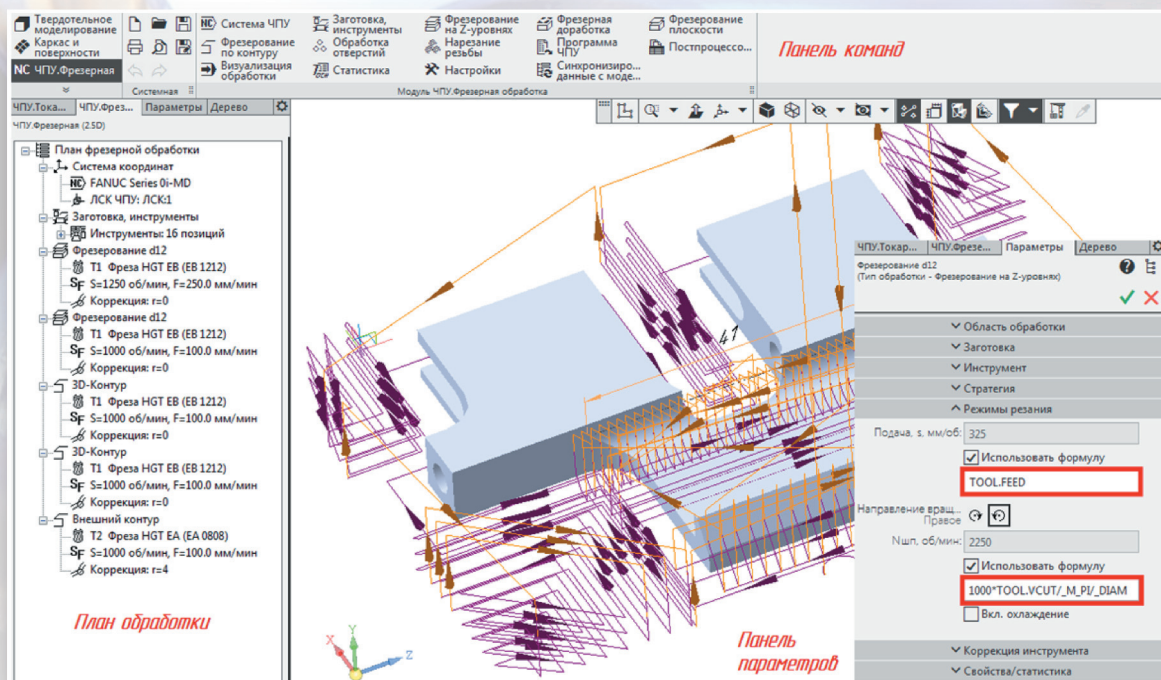


Рис. 2. Фрезерная обработка в окне КОМПАС-3D V18

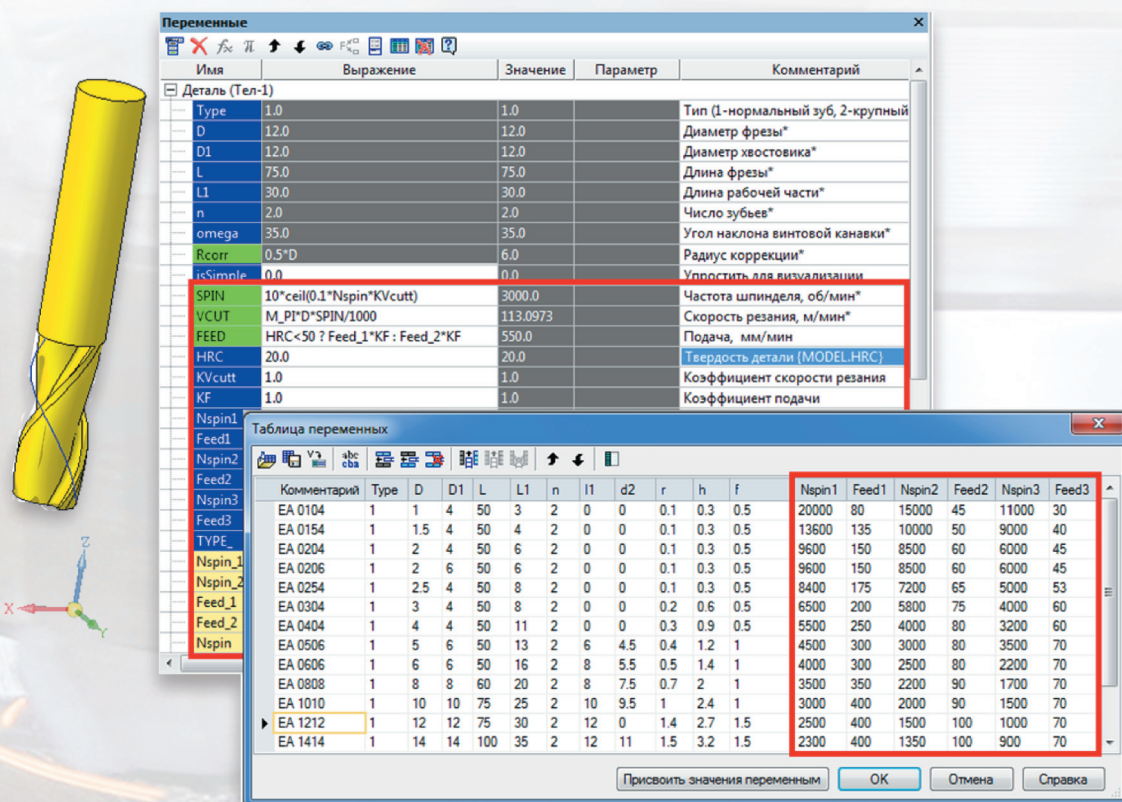


Рис. 5. Параметры 3D-модели инструмента

# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 25  
2019  
№ 7

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

DOI 10.17587/issn.1684-6400

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

## СОДЕРЖАНИЕ

### СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Щёкин А. В. Конструкторско-технологическая параметризация в составе интегрированной САМ-системы ..... 387

### WEB-ТЕХНОЛОГИИ

- Бова В. В., Кравченко Ю. А., Кулиев Э. В., Курейчик В. В. Моделирование поведения субъекта в Интернет-сервисах на основе модифицированного алгоритма бактериальной оптимизации ..... 397

### ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

- Шниперов А. Н., Сосновский М. С., Шипулин П. М. Робастный метод маркирования изображений цифровым водяным знаком, основанный на ортогональных моментах Цернике ..... 405

- Нагорнов Н. Н., Ляхов П. А., Червяков Н. И. Исследование шума квантования фильтров дискретного вейвлет-преобразования для обработки трехмерных изображений в медицине ..... 415

### ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

- Ефанов Д. В. Троичный код паритета в системах рабочего диагностирования устройств автоматики и вычислительной техники ..... 426

### БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Асратян Р. Э. Организация обработки защищенных сообщений в распределенных системах на основе Cryptographic Message Syntax ..... 435

### НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- Барский А. Б., Мельник Д. И. Нейросетевая модель целераспределения для вычислительной системы архитектуры *data flow* ..... 441

#### Главный редактор:

СТЕМПОВСКИЙ А. Л.,  
акад. РАН, д. т. н., проф.

#### Зам. главного редактора:

ИВАННИКОВ А. Д., д. т. н., проф.  
ФИЛИМОНОВ Н. Б., д. т. н., с.н.с.

#### Редакционный совет:

БЫЧКОВ И. В., акад. РАН, д. т. н.  
ЖУРАВЛЕВ Ю. И.,  
акад. РАН, д. ф.-м. н., проф.  
КУЛЕШОВ А. П.,  
акад. РАН, д. т. н., проф.  
ПОПОВ Ю. С.,  
акад. РАН, д. т. н., проф.  
РУСАКОВ С. Г.,  
чл.-корр. РАН, д. т. н., проф.  
РЯБОВ Г. Г.,  
чл.-корр. РАН, д. т. н., проф.  
СОЙФЕР В. А.,  
акад. РАН, д. т. н., проф.  
СОКОЛОВ И. А.,  
акад. РАН, д. т. н., проф.  
СУЕТИН Н. В., д. ф.-м. н., проф.  
ЧАПЛЫГИН Ю. А.,  
акад. РАН, д. т. н., проф.  
ШАХНОВ В. А.,  
чл.-корр. РАН, д. т. н., проф.  
ШОКИН Ю. И.,  
акад. РАН, д. т. н., проф.  
ЮСУПОВ Р. М.,  
чл.-корр. РАН, д. т. н., проф.

#### Редакционная коллегия:

АВДОШИН С. М., к. т. н., доц.  
АНТОНОВ Б. И.  
БАРСКИЙ А. Б., д. т. н., проф.  
ВАСЕНИН В. А., д. ф.-м. н., проф.  
ВАСИЛЬЕВ В. И., д. т. н., проф.  
ВИШНЕКОВ А. В., д. т. н., проф.  
ДИМИТРИЕНКО Ю. И., д. ф.-м. н., проф.  
ДОМРАЧЕВ В. Г., д. т. н., проф.  
ЗАБОРОВСКИЙ В. С., д. т. н., проф.  
ЗАРУБИН В. С., д. т. н., проф.  
КАРПЕНКО А. П., д. ф.-м. н., проф.  
КОЛИН К. К., д. т. н., проф.  
КУЛАГИН В. П., д. т. н., проф.  
КУРЕЙЧИК В. В., д. т. н., проф.  
ЛВОВИЧ Я. Е., д. т. н., проф.  
МАРТЫНОВ В. В., д. т. н., проф.  
МИХАЙЛОВ Б. М., д. т. н., проф.  
НЕЧАЕВ В. В., к. т. н., проф.  
ПОЛЕШУК О. М., д. т. н., проф.  
САКСОНОВ Е. А., д. т. н., проф.  
СОКОЛОВ Б. В., д. т. н., проф.  
ТИМОНИНА Е. Е., д. т. н., проф.  
УСКОВ В. Л., к. т. н. (США)  
ФОМИЧЕВ В. А., д. т. н., проф.  
ШИЛОВ В. В., к. т. н., доц.

#### Редакция:

БЕЗМЕНОВА М. Ю.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.  
Журнал включен в систему Российского индекса научного цитирования и базу данных RSCI на платформе Web of Science.  
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.

# INFORMATION TECHNOLOGIES

# INFORMACIONNYYE TEHNOLOGII

Vol. 25  
2019  
No. 7

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

ISSN 1684-6400

## CONTENTS

### CAD-SYSTEMS

**Shchekin A. V.** Design and Technological Parameterization in CAM ..... 387

### WEB-TECHNOLOGIES

**Bova V. V., Kravchenko Yu. A., Kuliev E. V., Kureichik V. V.** Behavior Modeling of the Subject in the Internet Services Based on the Bacterial Optimization Modified Algorithm ..... 397

### DIGITAL PROCESSING OF SIGNALS AND IMAGES

**Shniperov A. N., Sosnovskiy M. S., Shipulin P. M.** The Robust Image Digital Watermark Labeling Method Based on Orthogonal Zernike Moments ..... 405

**Nagornov N. N., Lyakhov P. A., Chervyakov N. I.** The Noise Research in Discrete Wavelet Transform Filters for 3D Images Processing in Medicine ..... 415

### COMPUTING SYSTEMS AND NETWORKS

**Efanov D. V.** The Ternary Parity Code in the On-Line Testing of Automation Devices ..... 426

### INFORMATION SECURITY

**Asratian R. E.** Protected Message Processing in Distributed Systems on the Basis of Cryptographic Message Syntax ..... 435

### NEUROTECHNOLOGIES

**Barsky A. B., Melnik D. I.** Neural Network Target Distribution Model for Computing System of Data Flow Architecture ..... 441

#### Editor-in-Chief:

Stempkovsky A. L., Member of RAS,  
Dr. Sci. (Tech.), Prof.

#### Deputy Editor-in-Chief:

Ivannikov A. D., Dr. Sci. (Tech.), Prof.  
Filimonov N. B., Dr. Sci. (Tech.), Prof.

#### Chairman:

Bychkov I. V., Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Zhuravljov Yu. I., Member of RAS,  
Dr. Sci. (Phys.-Math.), Prof.  
Kuleshov A. P., Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Popkov Yu. S., Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Rusakov S. G., Corresp. Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Ryabov G. G., Corresp. Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Soifer V. A., Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Sokolov I. A., Member of RAS,  
Dr. Sci. (Phys.-Math.), Prof.  
Suetin N. V.,  
Dr. Sci. (Phys.-Math.), Prof.  
Chaplygin Yu. A., Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Shakhnov V. A., Corresp. Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Shokin Yu. I., Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Yusupov R. M., Corresp. Member of RAS,  
Dr. Sci. (Tech.), Prof.

#### Editorial Board Members:

Avdoshin S. M., Cand. Sci. (Tech.), Ass. Prof.  
Antonov B. I.  
Barsky A. B., Dr. Sci. (Tech.), Prof.  
Vasenin V. A., Dr. Sci. (Phys.-Math.), Prof.  
Vasiliev V. I., Dr. Sci. (Tech.), Prof.  
Vishnekov A. V., Dr. Sci. (Tech.), Prof.  
Dimitrienko Yu. I., Dr. Sci. (Phys.-Math.), Prof.  
Domrachev V. G., Dr. Sci. (Tech.), Prof.  
Zaborovsky V. S., Dr. Sci. (Tech.), Prof.  
Zarubin V. S., Dr. Sci. (Tech.), Prof.  
Karpenko A. P., Dr. Sci. (Phys.-Math.), Prof.  
Kolin K. K., Dr. Sci. (Tech.)  
Kulagin V. P., Dr. Sci. (Tech.), Prof.  
Kureichik V. V., Dr. Sci. (Tech.), Prof.  
Ljvovich Ya. E., Dr. Sci. (Tech.), Prof.  
Martynov V. V., Dr. Sci. (Tech.), Prof.  
Mikhailov B. M., Dr. Sci. (Tech.), Prof.  
Nechaev V. V., Cand. Sci. (Tech.), Ass. Prof.  
Poleschuk O. M., Dr. Sci. (Tech.), Prof.  
Saksonov E. A., Dr. Sci. (Tech.), Prof.  
Sokolov B. V., Dr. Sci. (Tech.)  
Timonina E. E., Dr. Sci. (Tech.), Prof.  
Uskov V. L. (USA), Dr. Sci. (Tech.)  
Fomichev V. A., Dr. Sci. (Tech.), Prof.  
Shilov V. V., Cand. Sci. (Tech.), Ass. Prof.

#### Editors:

Bezmenova M. Yu.

Complete Internet version of the journal at site: <http://novtex.ru/IT>.

According to the decision of the Higher Certifying Commission of the Ministry of Education of Russian Federation, the journal is inscribed in "The List of the Leading Scientific Journals and Editions wherein Main Scientific Results of Theses for Doctor's or Candidate's Degrees Should Be Published"



# СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ CAD-SYSTEMS

УДК 621.9.06-52:004.4

DOI: 10.17587/it.25.387-396

**А. В. Щёкин**, зав. НИЛ, e-mail: schekin@inbox.ru,  
Национальный исследовательский Мордовский государственный университет  
имени Н. П. Огарева, г. Саранск

## Конструкторско-технологическая параметризация в составе интегрированной САМ-системы

*Приведен подход к программной организации конструкторско-технологической параметризации в области САМ (Computer-Aided Manufacturing). Подход основан на принципе многоуровневой параметризации и включает в себя наборы переменных, организованных на пяти уровнях иерархии: параметры конструкторской модели, параметры плана обработки, параметры инструментальных переходов, параметры конструкторско-технологических элементов (КТЭ), параметры инструкций управляющей программы. Иерархическая структура переменных позволяет обеспечить сквозную параметризацию и ассоциативность управляющей программы и предоставить единый набор параметров для задач, решаемых технологами при моделировании обработки в САМ-системе. Подсистема параметризации программно реализована в составе коммерческого САМ-приложения "Модуль ЧПУ. Фрезерная обработка", интегрированного в САД-платформу КОМПАС-3D. Приведены примеры использования параметризации для расчета режимов резания, создания пользовательской траектории трехкоординатного фрезерования и разработки постпроцессора.*

**Ключевые слова:** параметризация, САД/САМ-система, КОМПАС-3D, управляющая программа, 3D-модель, КТЭ, API (Application Program Interface)

### Введение

К разряду технологических САПР относятся системы технологической подготовки производства (САПР ТП) и автоматизированной генерации управляющих программ для станков с ЧПУ (САМ-системы). Использование в составе технологических САПР возможностей конструкторско-технологической параметризации дает пользователям этих систем дополнительные средства для автоматизации технологического проектирования. Технологическая параметризация похожа на геометрическую параметризацию, реализованную в большинстве систем конструкторского моделирования. Если посредством геометрической параметризации при изменении значений входных параметров перестраивается сама геометрическая модель, то технологическая параметризация приводит к автоматическому изменению числовых параметров технологического процесса.

Необходимость развития методов параметризации в технологических САПР связана тем, что существующие способы передачи инфор-

мации в САД/САМ-системах не обеспечивают достаточно глубокой интеграции данных, параметрические модели не позволяют исключить дублирования информации на разных этапах подготовки производства, параметры САПР используются хаотично, что создает несогласованность в использовании переменных и снижает надежность параметрических моделей.

Конструкторско-технологическая параметризация является числовой, так как она связывает числовые значения конструкторских и технологических параметров. Но ее использование невозможно без геометрической параметризации, которая при изменении значений управляемых параметров обеспечивает автоматическое перестроение геометрии модели и последующую актуализацию технологического процесса.

Разработке и исследованию методов геометрической параметризации посвящено много работ. Параметрический метод впервые встречается в трудах советского ученого Н. Ф. Четверухина [1, 2]. Его учеником В. С. Полозовым сформулированы базовые понятия теории па-

раметризации и разработаны теоретические основы моделирования в САПР с применением геометрических переменных [3]. Значительный вклад в развитие теории параметризации геометрических объектов внесли представители Нижегородской школы начертательной геометрии и компьютерной графики С. И. Ротков, И. И. Котов, Н. Н. Рыжов и их ученики [4–6]. Группой ученых из ЛЭТИ (Санкт-Петербург) под руководством Ю. Т. Лячека разработан аналитико-синтетический метод параметризации технических чертежей на основе адаптивных сеточных моделей [7–10]. Разработке и развитию вариационных методов параметризации посвящены работы Н. Н. Голованова, А. Г. Ершова, П. А. Копорушкина, В. В. Ермилова [11–15]; из западных исследователей наиболее известны две группы под руководством Джона Оуэна [16, 17] и Хоффманна [18, 19].

Одной геометрической параметризации недостаточно для технологической подготовки производства, поскольку она не учитывает технологические свойства изделия, необходимые для его изготовления. Логическим развитием геометрической параметризации является технологическая параметризация, реализованная в разной степени во многих САПР ТП. Вариант создания системы технологической параметризации на основе параметрических возможностей T-Flex CAD и САПР ТП "СИТЕП" [20] представлен в диссертации [21]. В этой работе запрограммированы расчетные модули для определения исходных параметров заготовки, режимов резания, межпереходных размеров и прочих параметров технологического процесса. Модули представляют собой наборы информационных связей в виде формул и иных зависимостей (в том числе SQL-запросов к базам данных) между параметрами конструкторского чертежа и параметрами технологической документации. Недостатком указанных модулей является игнорирование 3D-модели и ориентация на параметрический чертеж детали как источник конструкторской информации. В статье [22] приведен пример создания параметризованной модели комплексной детали для проектирования группового технологического процесса в САПР ТП "T-Flex. Технология", но при этом используется только геометрическая параметризация для получения размерных модификаций комплексной детали.

Таким образом, анализ литературы показывает, что технологическая параметризация традиционно рассматривается в контексте разработки технологических процессов в САПР

ТП, где информационные связи между переменными используются главным образом для заполнения шаблонов технологической документации. При моделировании обработки в САМ-системе перед технологами возникают аналогичные задачи (подбор заготовки, расчет режимов резания и т. д.), но главная цель САМ-системы — это генерация управляющих программ для станков с ЧПУ. При этом САМ-системы решают такие специфические задачи, как постпроцессирование и верификация траекторий. Создание системы сквозной параметризации для задач в области САМ дает дополнительные возможности для автоматизации подготовки управляющих программ.

### **1. Многоуровневая технологическая параметризация**

Термин "многоуровневая параметризация" впервые был использован в диссертации Е. С. Ермакова [6] в контексте конструирования изделий с иерархической структурой. Этому понятию Ермаков дал следующее определение: "под многоуровневой параметризацией будем понимать параметризацию объекта, имеющего иерархическую древовидную структуру, в рамках которой каждый элемент описывается определенным числом параметров, а геометрические и негеометрические условия, задаваемые при объединении элементов в узел или сборочную единицу, позволяют формировать набор параметров данного узла или сборочной единицы". Очевидным образом понятие "многоуровневая параметризация" применимо к любым объектам с иерархической структурой, в том числе и для технологических процессов механической обработки на станках с ЧПУ.

В контексте задач, решаемых технологами при моделировании обработки в САМ-системе, можно выделить пять уровней переменных (рис. 1):

- параметры обрабатываемой детали;
- параметры плана обработки;
- параметры инструментальных переходов (ИП);
- параметры конструкторско-технологических элементов (КТЭ);
- параметры инструкций управляющей программы (УП).

В состав параметров САМ-системы могут входить как внутренние переменные САМ-системы, так и внешние параметры обрабатываемой детали и режущих инструментов. Для реализации доступа к внешним перемен-



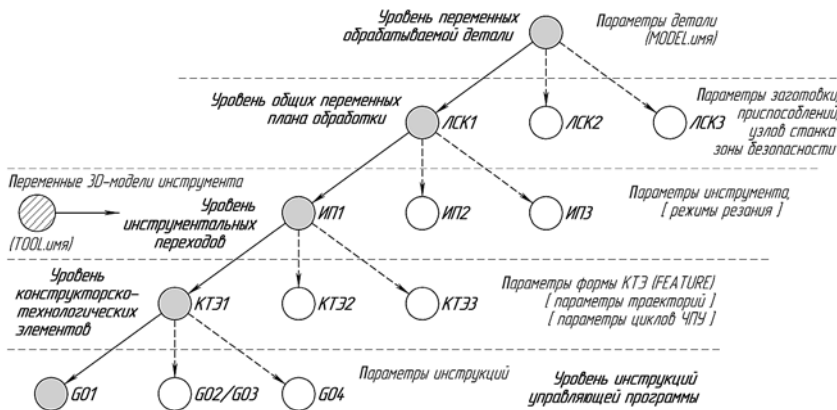


Рис. 1. Иерархия параметров САМ-системы

ным будем использовать следующие ссылки: *MODEL.имя переменной* — для доступа к параметрам детали; *TOOL.имя переменной* — для доступа к переменным 3D-модели режущего инструмента.

Переменные могут быть изменяемыми, которым присваивается математическое выражение, и неизменяемыми (доступными только для чтения). На рис. 1 изменяемые переменные приведены в квадратных скобках.

Такое разбиение переменных на уровни обусловлено не только иерархичностью технологического процесса, но и тем обстоятельством, что параметры имеют свои области применения: переменные нижних уровней могут зависеть от переменных вышестоящих уровней, но при этом они недоступны на верхних уровнях. Параметризация без ограничений доступа к переменным способна запутать пользователя и создать сложные и трудно выявляемые рекурсивные зависимости. Такие зависимости, в свою очередь, могут приводить к закликиванию при решении некоторых задач или к неправильному построению траекторий обработки. Иерархия параметров позволяет избежать хаотичного использования переменных и разграничить области их применения наиболее простым и понятным способом.

Параметры обрабатываемой детали  $P_i$  в общем случае включают в себя свойства материала  $p_m$ , параметрические переменные 3D-модели  $p_p$  и аннотации 3D-модели (размеры, шероховатости и прочие атрибуты)  $p_a$ :

$$P_i = \{p_m, p_p, p_a\}.$$

Каждый следующий уровень иерархии описывается кортежем

$$\langle P_i, M_{i-1}, R_i \rangle, M_i \in \{P_i, M_{i-1}\}, \\ i \in [2; 5], M_1 = P_1,$$

где  $P_i$  — множество параметров уровня  $i$ ;  $M_{i-1}$  — множество параметров предыдущих уровней (являются постоянными для решения задач  $i$ -го уровня);  $R_i$  — множество связей (отношений) между параметрами на  $i$ -м уровне.

Рассмотрим каждый уровень иерархии более подробно.

▲ **Уровень переменных конструкторской модели.** В общем случае под конструкторской моделью можно рассматривать как 3D-модель обрабатываемой детали, так и ее чертеж или сборку. Как правило, САМ-

системы чаще всего работают с 3D-моделями. Как в САПР ТП, так и в САМ-системах параметры конструкторской модели могут быть получены различными способами. Например, в работе [23] описаны пять групп методов преобразования конструкторской информации к виду, пригодному для использования в САПР ТП: "непосредственное кодирование таблиц с чертежа; ...на основе унификации деталей; идентификация поверхностей и КТЭ на основе использования OLE или API функций; установление свойств поверхностей на основе диалога; использование универсальных методов группирования чертежных элементов в классы при обработке геометрических форматов".

В работе [24] сравниваются три технологии интеграции прикладного программного обеспечения с системой AutoCAD: на основе обмена данными в формате DXF, COM-технология и программный интерфейс.NET API. Эти технологии также могут использоваться для получения исходных параметров обрабатываемой детали. API является наиболее функциональным и производительным способом анализа конструкторской модели для CAD/CAM-систем, построенных на базе единой CAD- платформы.

▲ **Уровень переменных плана обработки.** Процесс обработки детали в целом может включать в себя несколько планов обработки для различных операций (станков), установов и систем координат. К параметрическим переменным, общим для всей последовательности инструментальных переходов, будем относить: параметры заготовки, приспособлений, узлов станка, зоны безопасности. К ним можно отнести также общие параметры таблицы инструментов, например число позиций.

▲ **Уровень переменных инструментального перехода.** В план обработки входит последовательность инструментальных переходов. К пере-

менным инструментального перехода относятся параметры инструмента и режимы резания. В рассматриваемой САМ-системе параметры инструмента хранятся в его 3D-модели и извлекаются из модели через API-интерфейс. Переменные инструментального перехода являются закрытыми (недоступными) для других переходов. Они действуют только в пределах данного перехода, могут влиять на переменные нижних уровней и при этом зависеть от общих переменных плана обработки и внешних переменных конструкторской модели. Один и тот же инструмент может работать на нескольких переходах, но параметры инструмента по ссылке TOOL.имя переменной доступны только в пределах текущего перехода.

▲ **Уровень переменных КТЭ.** Конструкторско-технологический элемент (КТЭ) является одним из базовых понятий в современных САМ-системах. В иностранной литературе понятие КТЭ соответствует термин *manufacturing features* [25]. КТЭ является связующим звеном при передаче геометрической информации между CAD- и САМ-системами. В пределах одного инструментального перехода может быть обработано несколько КТЭ, и в общем случае параметры стратегии для их обработки могут различаться.

Параметры КТЭ можно разделить на две группы. Первая группа — это параметры формы и расположения КТЭ. Вторая группа — параметры траектории (способы удаления материала, глубины резания, припуски на чистовую обработку, режимы резания и т. д.). Иными словами, КТЭ включает в себя геометрию (*feature*) и траекторию обработки применительно к данной геометрии. Многие системы ЧПУ поддерживают постоянные циклы, которые упрощают программирование однотипных элементов детали. Анализ постоянных циклов различных систем ЧПУ показывает, что параметры этих циклов также можно разделить на геометрические и технологические параметры.

▲ **Уровень переменных управляющей программы.** Управляющая программа для станка с ЧПУ в САМ-системах генерируется, как правило, за два этапа: сначала создается последовательность кодов для описания траектории в промежуточном формате, затем промежуточный код транслируется в формат системы ЧПУ с помощью постпроцессора. Промежуточный формат содержит набор инструкций для описания траектории и команд управления станком. Эти инструкции в свою очередь могут содержать наборы параметров. Например, инструк-

ция для смены инструмента (функция T) может содержать номер инструментальной позиции и номер корректора, инструкция для линейной интерполяции (G01) — координаты начальной и конечной точек, инструкции для круговой интерполяции (G02/03) — дополнительно координаты центра дуги и направление обхода.

Разделение параметров на пять уровней является универсальным, поскольку отражает иерархическую структуру и модульность процесса обработки на станках с ЧПУ. При этом состав переменных, их обозначения, способы построения редакторов математических выражений могут различаться в зависимости от конкретной программной реализации.

## 2. Программная реализация

Описанный выше подход к организации подсистемы конструкторско-технологической параметризации программно реализован в составе коммерческого САМ-приложения "Модуль ЧПУ. Фрезерная обработка" [26] для платформы КОМПАС-3D. Приложение предназначено для автоматизации программирования 2,5- и 3-координатного фрезерования, поддерживает стратегии послойной обработки карманов на Z-уровнях, плоскостей, фрезерной доработки, фрезерования по контуру, 3-координатной обработки поверхностей сферическими фрезами и обработки отверстий. Более подробная информация о модулях САМ-системы представлена на сайте системы КОМПАС по ссылкам [26, 27] и в статьях [28, 29]. Средствами разработки САМ-системы являются Visual Studio C++, программный интерфейс API КОМПАС-3D и функции геометрического ядра C3D [30].

САМ-приложения полностью интегрированы в рабочее пространство КОМПАС-3D (рис. 2, см. вторую сторону обложки) и используют в качестве источника геометрической информации конструкторскую модель, созданную непосредственно в КОМПАС-3D или импортированную из другой CAD-системы. Автоматически созданные траектории обработки ассоциативны с конструктивными элементами модели. Ассоциативность означает связь траектории с опорными элементами 3D-модели и автоматическое перестроение траектории при изменении положений или размеров опорных объектов [31]. В качестве опорных объектов могут использоваться грани и ребра модели или дополнительные построения (кривые, эскизы).



Конструкторско-технологическая параметризация в составе САМ-системы представлена отдельной подсистемой. Архитектура подсистемы параметризации обусловлена требованиями повторного использования кода и возможностью наращивания ее функционала в будущем. САМ-система для платформы КОМПАС-3D в целом развивается по модульному принципу, поэтому подсистема параметризации должна уметь обслуживать любые САМ-модули в составе САМ-системы. Исходя из этих требований разработана иерархия классов (рис. 3), которая включает в себя абстрактный класс *Parameters* и пять производных классов, соответствующих пяти уровням иерархии параметрических переменных.

Входом в подсистему параметризации является метод *Calculate* (член класса *Parameters*), который принимает на вход строку с математическим выражением и вычисляет его значение. Переменные в формулах можно связывать алгебраическими и тригонометрическими выражениями, в состав формул могут входить скобки и операторы сравнения. Одновременно с синтаксическим разбором метод *Calculate* выполняет лексический анализ и проверку переменных на циклическую зависимость.

Значения переменных извлекаются с помощью метода *GetParameter*, который в базовом

классе представляет собой чистую виртуальную функцию и переопределяется в производных классах. Ниже показан листинг реализации *GetParameter* для класса параметров КТЭ (через указатель *owner* обеспечивается доступ к параметрам верхних уровней). Функция *GetParameter* принимает на вход имя параметра и возвращает указатель на переменную типа *double*. Переменная *pCycle* содержит указатель на объект КТЭ, являющийся внутренним объектом САМ-модуля. Работа через указатели исключает дублирование подсистемой параметризации внутренних параметров САМ-модуля и обеспечивает получение актуальных значений параметров в любой момент времени. Нулевое значение указателя означает, что параметр не обнаружен по входному имени.

```
double* FeatureParams::GetParameter(char* s /*имя параметра*/)
{
    double* p = GetCycleParameter(s, pCycle);
    if(p == NULL && pFeature) p = pFeature->GetParameter(s);
    if(p == NULL && pTrajectory) p = pTrajectory->GetParameter(s);
    if(p == NULL && owner) p = owner->GetParameter(s);
    return p;
}
```

Класс *ModelParams* содержит параметры обрабатываемой детали. Экземпляр этого класса создается один раз в момент подключения САМ-приложения к системе КОМПАС-3D.

В начальный момент синхронизации плана обработки с внутренними данными 3D-модели вызывается функция *GetModelParameters*, которая считывает параметры конструкторской модели через программный интерфейс API.

Классы *PlanParams* (параметры плана), *OperationParams* (параметры инструментального перехода), *FeatureParams* (параметры КТЭ) и *CodeParams* (параметры инструкции УП) содержат указатели на внутренние объекты САМ-модуля (соответственно, *pPlan\**, *pOperation\**, *pCycle\** и *pCadre\**). Используя эти указатели, САМ-приложение возвращает адреса своих внутренних переменных. Таким образом, каждый САМ-модуль может возвращать собственный набор параметров. Состав переменных САМ-приложения "Модуль ЧПУ. Фрезерная обработка" представлен в таблице.

Класс *FeatureParams* дополнительно содержит параметры формы

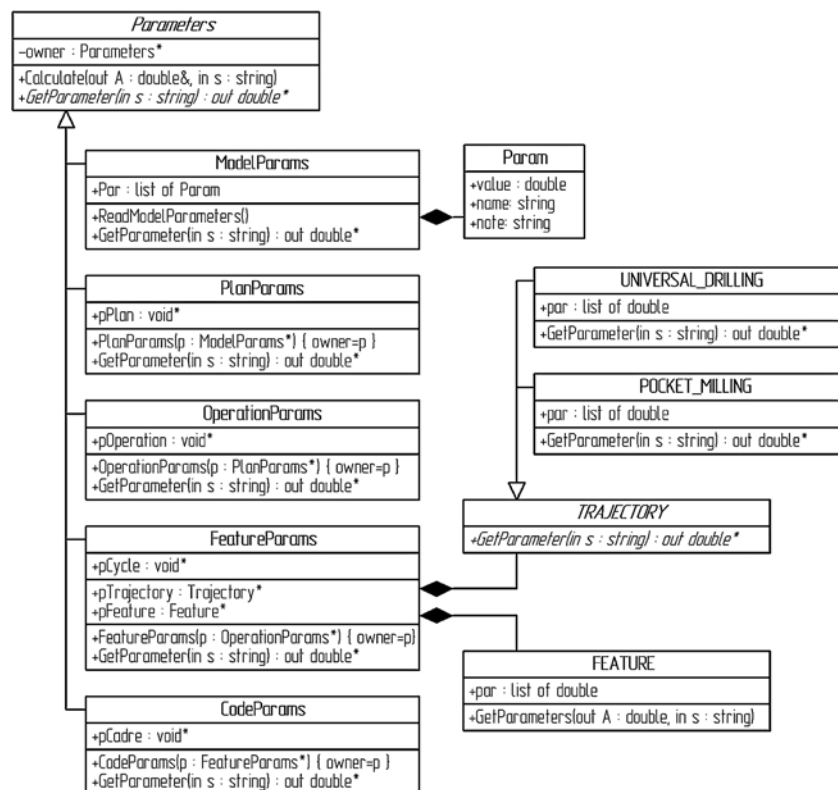


Рис. 3. Диаграмма классов подсистемы параметризации

**Параметры САМ-приложения  
"Модуль ЧПУ. Фрезерная обработка"**

|                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Переменные плана обработки (уровень 2)            | <i>_STX1, _STY1, _STZ1</i> — первая точка габарита заготовки,<br><i>_STX2, _STY2, _STZ2</i> — вторая точка габарита заготовки,<br><i>_ZTOP</i> — плоскость отвода для смены инструмента,<br><i>_ZSAF</i> — положение плоскости безопасности по оси Z.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Переменные инструментального перехода (уровень 3) | <i>_TPOS</i> — номер позиции инструмента,<br><i>_DIAM</i> — диаметр инструмента с учетом износа,<br><i>_RCOR</i> — радиус коррекции инструмента (0,5· <i>_DIAM</i> ),<br><i>_SPIN</i> — обороты при черновой обработке,<br><i>_FEED</i> — подача при черновой обработке,<br><i>_VREZ</i> — подвод по оси Z,<br><i>_SDIR</i> — направление вращения шпинделя.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Переменные КТЭ (уровень 4)                        | Параметры формы <i>FEATURE</i> фигурных циклов (рис. 4):<br><i>_FT, _FX, _FY, _FZ, _FD, _FL, _FW, _FH, _FA, _FB, _FR</i> .<br>Параметры траектории <i>UNIVERSAL_DRILLING</i><br><i>_DEPZ</i> — глубина сверления на первом проходе,<br><i>_CHIP</i> — способ удаления стружки,<br><i>_TPOT</i> — пауза после вывода сверла между проходами,<br><i>_PBOT</i> — пауза на дне отверстия,<br><i>_GAPZ</i> — длина отвода между проходами,<br><i>_KOFZ</i> — коэффициент уменьшения глубины прохода,<br><i>_MDEP</i> — минимальная глубина сверления,<br><i>_FWRK</i> — подача при черновой обработке.<br>Параметры фрезерной траектории <i>POCKET_MILLING</i> :<br><i>_VOLM</i> — объем обработки (черновая, чистовая),<br><i>_DEPZ</i> — максимальное расстояние между Z-уровнями,<br><i>_KOFR</i> — коэффициент перекрытия радиуса фрезы,<br><i>_CDIR</i> — направление обхода (попутное или встречное),<br><i>_FWRK</i> — подача при черновой обработке,<br><i>_BEGN</i> — начало траектории (от центра или от периферии),<br><i>_FINZ</i> — глубина чистовой обработки (глубина Z-уровня),<br><i>_USID</i> — припуск на чистовую обработку боковых сторон,<br><i>_UBOT</i> — припуск на чистовую обработку дна,<br><i>_FFIN</i> — подача при чистовой обработке,<br><i>_SFIN</i> — обороты при чистовой обработке,<br><i>_RADP</i> — радиус врезания в горизонтальной плоскости. |
| Переменные УП (уровень 5)                         | Общие: <i>X1, _Y1, _Z1, _X2, _Y2, _Z2, _T, _S, _F, _M345</i> .<br>Параметры G02/G03: <i>_X0, _Y0, _Z0, _R, _CW</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

и параметры траектории. Смысл параметров формы поясняет рис. 4. Определено девять форм: одна форма для отверстий (идентификатор *HOLE*); четыре формы для обработки карманов (*POCKET\_RECT*, *POCKET\_CIRC*, *POCKET\_SLOT* и *POCKET\_CIRCSLOT*); четыре формы для обработки цапф (*FACE\_RECT*, *FACE\_CIRC*, *FACE\_SLOT*, *FACE\_CIRCSLOT*). Параметр *\_FT* содержит тип формы (его значение совпадает с идентификатором формы), параметры *\_FX, \_FY, \_FZ* задают координаты начальной точки формы (на рис. 4 начальные точки обозначены нулями), параметр *\_FH* задает глубину кармана или высоту цапфы. Параметры формы автоматически распознаются по геометрии 3D-модели и доступны только для чтения.

В САМ-приложении реализованы две обобщенные траектории для постоянных циклов систем ЧПУ: универсальная траектория для сверления *UNIVERSAL\_DRILLING* и траектория *POCKET\_MILLING* для фрезерования карманов по эквидистантной схеме.

На самом нижнем уровне иерархии находятся параметры инструкций УП. Каждая инструкция содержит номер позиции инструмента *\_T*, текущие значения режимов резания *\_S* и *\_F*, координаты начальной и конечной точки. Инструкции для круговых интерполяций (G02, G03) дополнительно содержат центр дуги, радиус *\_R* и направление обхода *\_CW*.

### 3. Примеры использования параметризации в САМ

На основе трех примеров (расчет режимов резания, создание пользовательской траектории и разработка подсистемы конструкторско-технологической параметризации в составе САМ-модуля для платформы КОМПАС-3D.

**Расчет режимов резания.** Режущие инструменты и приспособления

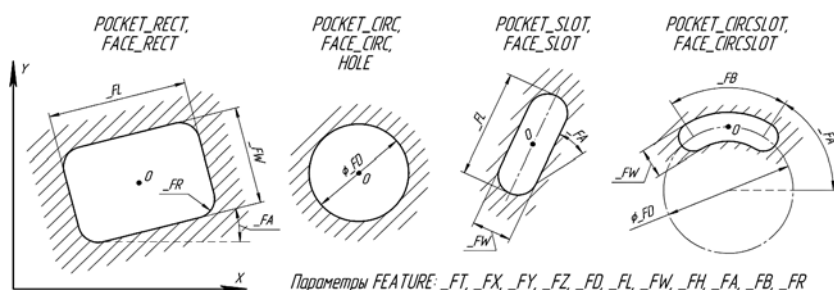


Рис. 4. Параметры формы фигурных циклов

в САМ-системе представляют собой параметризованные 3D-модели, которые управляют наборами входных параметров. При этом роль базы данных стандартных инструментов и приспособлений играет таблица переменных, встроенная непосредственно в 3D-модель. Управляемые параметры инструмента считываются из 3D-модели посредством API КОМПАС и доступны технологу для редактирования из САМ-приложения. Рассмотрим программирование расчета режимов резания на примере концевой фрезы тайваньской фирмы HGT серии EA.

Создадим параметризованную 3D-модель фрезы в системе КОМПАС (рис. 5. см. вторую сторону обложки). Стандартные размеры инструмента оформим в виде таблицы переменных в соответствии с данными из каталога HGT. Частота вращения шпинделя и скорость подачи по каталогу зависят от диаметра фрезы и твердости обрабатываемого материала *HRC*. Эти данные сгруппированы для трех значений твердости:  $\sim HRC30$ ,  $\sim HRC50$  и  $\sim HRC60$ .

В качестве исходных параметров для расчета режимов резания назначаем: твердость *HRC* и два дополнительных коэффициента  $KV_{cutt}$  и  $KF$ , чтобы пользователь мог самостоятельно влиять на частоту вращения шпинделя и подачу. В качестве выходных параметров объявляем переменные *SPIN* (частота,  $\text{мин}^{-1}$ ), *FEED* (подача,  $\text{мм/мин}$ ) и скорость резания *VCUT* ( $\text{м/мин}$ ). Входные и выходные параметры связываем между собой математическими формулами.

Входной параметр *HRC* определим через одноименную переменную *HRC* обрабатываемой детали. Но поскольку в редакторе формул системы КОМПАС-3D мы не можем явно использовать переменные САМ-системы, то в САМ-приложении "Модуль ЧПУ. Фрезерная обработка" реализована такая возможность: если в комментарии переменной встречается строка в фигурных скобках, то САМ-модуль распознает такую строку как математическое выражение и вычисляет его значение. Поэтому в комментарии к *HRC* пишем: "Твердость детали {MODEL.HRC}". Формулы в фигурных скобках вычисляются на уровне плана обработки до начала расчета параметров инструментальных переходов, поэтому в этих формулах можно использовать только переменные уровней плана обработки и обрабатываемой детали.

Чтобы применить алгоритм расчета, необходимо в САМ-модуле для инструментального перехода присвоить режимам резания выраже-

ния *TOOL.FEED* и *TOOL.SPIN* (на панели параметров, рис. 5). В формулах эти выражения присутствуют по умолчанию. Технологу остается только включить их использование. Для частоты вращения шпинделя рекомендуется использовать более универсальную формулу через скорость резания *TOOL.VCUT* как параметр, независимый от диаметра инструмента.

После выбора конструктором другого материала (с другим значением твердости *HRC*) режимы резания будут автоматически пересчитываться и попадать в управляющую программу. От технолога при этом требуется только заново сгенерировать управляющую программу и передать ее оператору станка с ЧПУ.

**Программирование траектории.** Штатных стратегий обработки, которые имеет в наличии САМ-система, может оказаться недостаточно для программирования траекторий. В таких случаях можно применить фрезерование по контуру и цикл копирования траектории. На рис. 6 (см. третью сторону обложки) показан пример чистовой обработки поверхности скругления концевой фрезой. Фреза должна многократно перемещаться вертикальными движениями вдоль поверхности с таким шагом, чтобы обеспечить заданный параметр чистоты обработки поверхности. Опорным объектом является эскиз в плоскости YZ, в котором построена проекция поверхности. При выборе эскиза строится однократная траектория, повторяющая кривую эскиза.

Чтобы создать многопроходную траекторию, вызываем цикл повтора контура. Диалог цикла выступает в качестве редактора переменных, в котором параметрам цикла можно присваивать математические выражения. Шаг по оси X определяем с учетом чистоты обработки (переменная *MODEL.C*). Для вычисления числа повторений по оси X длину обработки делим на шаг и округляем результат в большую сторону, при этом длину обработки выражаем через переменную детали *MODEL.A*, ассоциативно связанную с линейным размером на конструкторской модели. В результате получаем траекторию, которая автоматически перестраивается при изменении линейного размера и параметра чистоты обработки поверхности. Если технолог назначит другую фрезу, то шаг траектории автоматически изменится в соответствии с новым диаметром фрезы.

**Настройка постпроцессора.** Многообразие систем ЧПУ и их модификаций несколько велико, что требует открытого способа настройки постпроцессоров на стороне пользователей.



Применение специальных макросов и формул, связывающих параметрические переменные, позволяет сделать этот процесс простым и наглядным. Ниже показан фрагмент постпроцессора для системы ЧПУ FANUC.

```
ALPHABET(X, Y, Z, U, V, W, R, I, J, K, Q, T, H, S, F, M, G);
FORMAT(print = 1, number = 1, %m = %5.3m, %n = %5.3n, %f = %5.3f);
CODE('Head'); PRINT('%');
    PRINT('O00001');
CODE('Safe'); PRINT('G90 G40 G17');
CODE('T'); PRINT('T%02f M06', _T);
    PRINT('G54 G90 X0 Y0');
    PRINT('G43 H%2f Z%0n', _T, _ZTOP);
    PRINT('S%0m F%0m M%1f', _SPIN, _FEED, _SDIR);
CODE('S'); PRINT('S%0m', _S);
CODE('F'); PRINT('F%0m', _F);
CODE('G01'); PRINT('G%1m X%0m Y%0m Z%0m F%0m', 1, _X2, _Y2, _Z2, _F);
CODE('G02'); PRINT('G%1m X%0m Y%0m Z%0m R%0n F%0m', 2, _X2, _Y2, _Z2, _R, _F);
CODE('G03'); PRINT('G%1m X%0m Y%0m Z%0m R%0n F%0m', 3, _X2, _Y2, _Z2, _R, _F);
CODE('G04'); PRINT('G4 P%3.1n', _P);
CODE('Offset'); PRINT('G54 X%0m Y%0m Z%0m', _dX, _dY, _dZ);
CODE('Rotate'); PRINT('G73 H%0f', _A);
CODE('CallSubProg'); PRINT('M98 P%4f', 1000 + _P);
```

С помощью макроса PRINT и переменных настраивается вывод различных инструкций промежуточного формата в коды системы ЧПУ FANUC. Макрос CODE задает инструкцию УП, в отношении которой действуют последующие макросы PRINT. Название инструкции передается в макрос CODE как его аргумент. Например, строка 'Head' означает заголовок УП, 'Safe' — начальные коды безопасности, 'T' — смену инструмента и т. д. Макрос PRINT содержит переменное число аргументов. С помощью спецификаторов числовых форматов задается способ вывода десятичных чисел: %m — модальный вывод значения кадра, %n — немодальный вывод, %f — вывод числа безотносительно к модальности. Макрос FORMAT задает настройки вывода для всех последующих макросов PRINT. Для разработки постпроцессоров следует пользоваться таблицей, в которой перечислены переменные САМ-модуля фрезерной обработки и указана их принадлежность уровням иерархии.

### Заключение

В статье рассмотрены принципы организации подсистемы сквозной конструкторско-технологической параметризации для задач, решаемых инженерами-технологами при моделировании обработки в САМ-системе, и ее

программная реализация в составе интегрированного САМ-приложения для платформы КОМПАС-3D.

Стержнем подсистемы является иерархия параметрических переменных, сгруппированных на пяти уровнях подчиненности: параметры детали, параметры плана обработки, параметры инструментальных переходов, параметры КТЭ и параметры инструкций управляющей программы. Такая иерархия проста и понятна технологом, поскольку она отражает реальную структуру и модульность технологического процесса обработки на станках с ЧПУ. Интеграция конструкторских и технологических моделей на базе API обеспечивает сквозную ассоциативность по цепочке CAD — САМ — CNC, когда изменения, внесенные конструктором в деталь, автоматически отображаются в кодах управляющей программы.

Приведены примеры использования параметризации для решения трех задач: расчета режимов резания, создания пользовательской траектории и настройки постпроцессора. По мере своего развития подсистема может быть применена и для других задач в области САМ, например, для автоматизации подбора инструментов и приспособлений.

### Список литературы

1. Четверухин Н. Ф. Условные изображения и параметрический метод их построения // Вопросы современной начертательной геометрии: сб. статей под ред. Н. Ф. Четверухина. М.-Л.: ОГИЗ, 1947. С. 188—223.
2. Четверухин Н. Ф., Яцкевич Л. А. Параметризация и ее применение в геометрии // Математика в школе. 1963. № 5. С. 15—23.
3. Полозов В. С., Бudeков О. А., Ротков С. И. и др. Автоматизированное проектирование. Геометрические и графические задачи. М.: Машиностроение, 1983. 280 с.
4. Ротков С. И. Разработка средств геометрического моделирования и компьютерной графики пространственных объектов для CALS-технологий: дисс. докт. техн. наук, Н. Новгород, 05.01.01, ННГАСУ, 1999. 300 с.
5. Рыжов Н. Н. Параметрическая геометрия. М.: МАДИ, 1988. 56 с.
6. Ермаков Е. С. Принципы многоуровневой параметризации при формировании объектов: автореф. дис. канд. техн. наук: 05.13.12. Нижегород. гос. архитектур.-строит. ун-т Н. Новгород, 2007. 23 с.
7. Лячек Ю. Т. Геометрическое моделирование. Параметризация и модификация 3D-моделей и чертежей в САПР. СПб.: Изд-во СПбГЭТУ "ЛЭТИ", 2015. 160 с.
8. Бассам А. М. Исследование и разработка алгоритмического и программного обеспечения параметризации конструкторских чертежей на основе адаптивной сетевой модели: дис. канд. тех. наук: 05.13.12. Санкт-Петербург, 2013. 135 с.
9. Лячек Ю. Т., Алькади Л. Методы создания параметрических моделей геометрических объектов в современных САПР. Кибернетика и программирование. 2016. № 2. С. 42—51.

10. Алкади Л. Д. Г., Лячек Ю. Т., Мусайд А. М. О., Мустафа Ахмед Б. М. Параметрическая сеточная модель конструкторского чертежа: Международная конференция по мягким вычислениям и измерениям. 2016. Т. 2. Секции 4–7. С. 77–80.
11. Голованов Н. Н. Геометрическое моделирование: учебник для учреждений высш. проф. образования. М.: Издательский центр "Академия", 2011. 272 с.
12. Ершов А. Г. Алгоритмы и программные системы для геометрических задач параметрического проектирования: автореф. дис. канд. физ.-мат. наук: 05.13.11. Новосибирск, 2007. 19 с.
13. Копорущин П. А. Алгоритмы расчета параметрических моделей геометрических объектов // Автоматизированное проектирование в машиностроении. 2015. № 3. С. 95–97.
14. Копорущин П. А. Применение DM-декомпозиции при расчете параметрических моделей объектов // Современные проблемы теории машин. 2016. № 4–1. С. 24–26.
15. Ермилов В. В. Вариационное параметрическое геометрическое моделирование в САПР на основе онтологий: автореф. дис. канд. техн. наук: 05.13.12. Ижевск, 2008. 20 с.
16. Owen J. Algebraic solution for geometry from dimensional constraints // Proc 1st ACM Symposium on Solid Modeling and CAD/CAM Applications, ACM Press. 1991. P. 397–407.
17. Jackson B., Owen J. C. A characterisation of the generic rigidity of 2-dimensional point-line frameworks. Preprint, arXiv: 1407.4675, 2016.
18. Hoffmann C. M., Joan-Arinyo R. A brief on constraint solving. CAD&A, 2: 655–663, 2005.
19. Fudos I., Hoffmann C. M., Joan-Arinyo R. Tree-decomposable and underconstrained geometric constraint problems. Preprint, arXiv: 1608.05205, 2016.
20. Васильев С., Ушкевич В., Кузьмин В., Мазурин А. СИТЕП: инвариантная система технологического проектирования // САПР и графика. 2000. № 7. С. 43–47.
21. Калякулин С. Ю. Система автоматизированной конструкторско-технологической параметризации процессов изготовления деталей типа тел вращения: дис. ... канд. тех. наук: 05.13.06, 2016. 185 с.
22. Овчинников А. Ю., Князева Н. Ю. Разработка методики построения комплексной детали при применении группового технологического процесса с использованием САПР "Т-FLEX" // Вестник Мордовского университета. 2016. Т. 26. № 3. С. 312–324.
23. Кузьмин В. В., Максимовский Д. Е. Методы преобразования конструкторской информации машиностроительных деталей // Вестник МГТУ Станкин. 2012. № 2. С. 92–95.
24. Шишигин Д. С. К выбору технологии интеграции прикладного программного обеспечения с САПР // Труды СПИИРАН. 2016. № 4 (47). С. 211–224.
25. Shah J. J., Mäntylä M. Parametric and feature-based CAD/CAM: concepts, techniques, and applications. Wiley, New York, 1995. 619 p.
26. Официальный сайт АСКОН. САМ-приложение "Модуль ЧПУ. Фрезерная обработка". URL: <https://kompas.ru/kompas-3d/application/machinery/module-chpu-fo>
27. Официальный сайт АСКОН. САМ-приложение "Модуль ЧПУ. Токарная обработка". URL: <https://kompas.ru/kompas-3d/application/machinery/module-chpu>
28. Shchekin A. V., Sul'din S. P., Mitin E. V. Simulation of the Machining of a Bush in the KOMPAS-3D System // Russian Engineering Research. 2017. Vol. 37, N. 11. P. 987–990.
29. Щекин А. В., Сульдин С. П., Митин Э. В. Особенности САМ-приложения "Модуль ЧПУ. Токарная обработка" // ВСТИН. 2017. № 8. С. 16–18.
30. Камнев А. Интерфейс прикладного программирования геометрического ядра СЗД, его применение и главное отличие от API системы КОМПАС-3D // САПР и графика. 2016. № 5. С. 36–38.
31. Щекин А. В., Сульдин С. П. Ассоциативность траекторий в САМ-приложении "Модуль ЧПУ. Токарная обработка" // Мехатроника, автоматизация, управление. 2015. Т. 16. № 8. С. 570–575.

A. V. Shchekin, Head of Research Laboratory, e-mail: [schekin@inbox.ru](mailto:schekin@inbox.ru),  
N. P. Ogarev Mordovia State University (Saransk), Saransk, 430005, Russian Federation

## Design and Technological Parameterization in CAM

*The article describes the approach to the organization of a subsystem of design and technological parametrization applied to tasks in CAM (Computer-Aided Manufacturing). The approach is based on the principle of multilevel parametrization and includes sets of parametric variables organized at five levels of the hierarchy: external variables of the design model, general variables of the processing plan, processing variables, feature variables and control program variables. The hierarchical structure of parametric variables allows end-to-end parameterization and associativity of the control program, provides a uniform set of parameters for all tasks that users solve in a CAM system. A rigid hierarchy of variables allows you to divide the scope of use of parameters in a simple and reliable way. The subsystem of parametrization is implemented as a part of the commercial CAM-application for milling processing integrated into the KOMPAS-3D CAD-platform. Examples of calculating of cutting parameters, creating user trajectory for 3D-milling and programming postprocessor using parametrization are given.*

**Keywords:** parametrization, CAD/CAM, KOMPAS-3D, NC-program, 3D-model, manufacturing feature, API (Application Program Interface)

DOI: 10.17587/it.25.387-396

## References

1. Chetveruhin N. F. Conditional images and parametric method of its construction, *Questions of modern descriptive geometry. Collected papers edited by N. F. Chetveruhin*, Moscow-Leningrad, OGIZ, 1947, pp. 188–223 (in Russian).
2. Chetveruhin N. F., Jackevich L. A. *Mathematics in school*, 1963, no. 5, pp. 15–23 (in Russian).
3. Polozov V. S., Budekov O. A., Rotkov S. I. Automated design. Geometric and graphical tasks, Moscow, Mashinostroenie, 1983, 280 p. (in Russian).
4. Rotkov S. I. Development of geometric modeling tools and computer graphics of spatial objects for CALS-technologies. Doct. Diss., Nizhny Novgorod, 1999, 300 p. (in Russian).
5. Ryzhov N. N. Parametric Geometry, Moscow, MADI, 1988, 56 p. (in Russian).
6. Ermakov E. S. Principles of multilevel parametrization in the formation of objects. Abstract of thesis cand. of tech. sci., Nizhny Novgorod, 2007, 23 p. (in Russian).
7. Ljachev Ju. T. Geometric modeling. Parametrization and modification of 3D models and CAD drawings, SPb., Publishing house of SPbGJeTU "LJeTI", 2015, 160 p. (in Russian).
8. Bassam A. M. Research and development of algorithmic and software for parameterization of design drawings based on the adaptive network model. Thesis cand. of tech. sci., Sankt-Peterburg, 2013, 135 p. (in Russian).
9. Ljachev Ju. T., Al'kadi L. *Kibernetika i programirovanie*, 2016, no. 2, pp. 42–51 (in Russian).
10. Alkadi L. D. G., Ljachev Ju. T., Musaid A. M. O., Mustafa Ahmed B. M. *International Conference of Soft Computing and Measurement*, 2016, vol. 2, no. 4–7, pp. 77–80 (in Russian).
11. Golovanov N. N. Geometric modeling, Moscow, Izdatel'skij centr "Akademija", 2011, 272 p. (in Russian).
12. Ershov A. G. Algorithms and software systems for geometric problems of parametric design. Abstract of thesis cand. of fiz.-mat. sci., Novosibirsk, 2007, 19 p. (in Russian).
13. Koporushkin P. A. *Avtomatizirovannoe proektirovanie v mashinostroenii*, 2015, no. 3, pp. 95–97 (in Russian).
14. Koporushkin P. A. *Sovremennye problemy teorii mashin*, 2016, no. 4–1, pp. 24–26 (in Russian).
15. Ermilov V. V. Variational parametric geometric modeling in CAD on the basis of ontologies. Abstract of thesis cand. of tech. sci., Izhevsk, 2008, 20 p. (in Russian).
16. Owen J. Algebraic solution for geometry from dimensional constraints, *Proc 1st ACM Symposium on Solid Modeling and CAD/CAM Applications*, ACM Press (1991), pp. 397–407.
17. Jackson B., Owen J. C. A characterisation of the generic rigidity of 2-dimensional point-line frameworks, Preprint, arXiv: 1407.4675, 2016.
18. Hoffmann C. M., Joan-Arinyo R. A brief on constraint solving, *CAD&A*, 2: 655–663, 2005.
19. Fudos I., Hoffmann C. M., Joan-Arinyo R. Tree-decomposable and underconstrained geometric constraint problems, Preprint, arXiv: 1608.05205, 2016.
20. Vasil'ev S., Ushkevich V., Kuz'min V., Mazurin A. *SAPR i grafika*, 2000, no. 7, pp. 43–47 (in Russian).
21. Kaljakulin S. Ju. The system of automated design and technological parametrization of manufacturing processes for parts such as bodies of revolution. Thesis cand. of tech. sci., Moscow, 2016, 185 p. (in Russian).
22. Ovchinnikov A. Ju., Knyazeva N. Yu. *Vestn. Mordovsk. Gos. Univ.*, 2016, vol. 26, no. 3, pp. 312–324 (in Russian).
23. Kuz'min V. V., Maksimovskij D. E. *Vestnik MGTU Stankin*, 2012, no. 2, pp. 92–95 (in Russian).
24. Shishigin D. S. *Trudy SPIIRAN*, 2016, no. 4 (47), pp. 211–224 (in Russian).
25. Shah J. J., Mäntylä M. Parametric and feature-based CAD/CAM: concepts, techniques, and applications, Wiley, New York, 1995, 619 p.
26. Official web site of ASCON. CAM-application "CNC module. Milling", available at: <https://kompas.ru/kompas-3d/application/machinery/module-chpu-fo> (in Russian).
27. Official web site of ASCON. CAM-application "CNC module. Turning", available at: <https://kompas.ru/kompas-3d/application/machinery/module-chpu> (in Russian).
28. Shchekin A. V., Sul'din S. P., Mitin E. V. *Russian Engineering Research*, 2017, vol. 37, no. 11, pp. 987–990.
29. Shchekin A. V., Sul'din S. P., Mitin E. V. *Russian Engineering Research*, 2018, vol. 38, no. 2, pp. 123–125.
30. Kamnev A. *SAPR i grafika*, 2016, no. 5, pp. 36–38 (in Russian).
31. Shchekin A. V., Sul'din S. P. *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2015, vol. 16, no. 8, pp. 570–575 (in Russian).

**В. В. Бова**, доц., e-mail: vvbova@sfedu.ru,  
**Ю. А. Кравченко**, канд. техн. наук, доц., e-mail: yakravchenko@sfedu.ru,  
**Э. В. Кулиев**, канд. техн. наук, доц., e-mail: ekuliev@sfedu.ru,  
**В. В. Курейчик**, д-р техн. наук, зав. каф. САПР, e-mail: vkur@sfedu.ru,  
Южный федеральный университет, г. Таганрог

### Моделирование поведения субъекта в Интернет-сервисах на основе модифицированного алгоритма бактериальной оптимизации<sup>1</sup>

*Данная статья посвящена решению междисциплинарной научной проблемы идентификации несущих угрозу информационных событий в процессе взаимодействия субъекта с Интернет-сервисами. Предложены модели поведения интеллектуальных агентов в Интернет-пространстве и модифицированный алгоритм бактериальной оптимизации, позволяющий эффективно провести оценку семантической близости характеристик имеющихся прецедентов и текущего события.*

**Ключевые слова:** информационные угрозы, онтологические структуры, интеллектуальные агенты, роевые алгоритмы, модели поведения

#### Введение

Виктимное воздействие Интернет-сервисов на пользователя имеет различные типы и способы проявления. Часто информационные угрозы связаны с применением методов социальной инженерии, в основе которых находятся психологические приемы, позволяющие либо усыпить бдительность, либо ввести в заблуждение субъекта Интернет-общения [1–3] в целях проведения информационных воздействий (атак) различного характера, наносящих прямой или косвенный материальный и/или моральный ущерб субъекту. Для обеспечения информационной безопасности применяется комплекс мер различного характера, позволяющих предотвратить или уменьшить последствия подобных угроз [4]. Данная работа посвящена развитию методов и средств автоматизированной идентификации ключевой информации, указывающей на возможное виктимное воздействие с применением социальной инженерии, и повышению эффективности интеллектуальных систем-ассистентов, обеспечивающих безопасность и эффектив-

ность деятельности пользователя в Интернет-пространстве [1–4]. Это позволяет идентифицировать несущие угрозу информационные события в процессе взаимодействия и выработать (предложить) эффективную стратегию для предотвращения или уменьшения нежелательных последствий.

Сложность описанной проблемы двояка и противоречива, она заключается, с одной стороны, в необходимости обеспечения высокой скорости (в реальном масштабе времени) принятия решений создаваемыми интеллектуальными системами-ассистентами, а с другой — необходимостью постоянного анализа и обработки больших объемов информации в условиях неопределенности, так как реальная Интернет-среда представляет собой сложный для формализации комплекс взаимосвязанных предметных областей и отличается частичной наблюдаемостью, динамикой, непрерывностью, эпизодичностью, стохастичностью [5–8]. Для обработки подобной информационной среды авторы предлагают использовать концепцию агентности, предполагающую множественность, автономность одновременно со связанностью агентов, принимающих решения относительно процессов анализируемой среды и субъектов (пользователей) информационного взаимодействия.

<sup>1</sup> Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 18-29-22019.



## 1. Основные свойства интеллектуальных систем-ассистентов взаимодействия субъекта с Интернет-средой

Для эффективной идентификации информационных угроз разрабатываемые интеллектуальные системы-ассистенты должны обладать основными свойствами, такими как сопровождение взаимодействия пользователя с Интернет-средой, самообучения и самоорганизации. Система, обладающая перечисленными свойствами, позволяет извлекать и обрабатывать ценную инструктирующую информацию, указывающую на безопасность или, напротив, вредоносность получаемых пользователем данных. Построить такую систему-ассистента с нуля невозможно, ее можно только "вырастить", создав эффективный и быстрый инструмент, наделенный свойством самоорганизации. Подобная система самообучается, собирая по мере функционирования в заданной среде ценную информацию и создавая базу прецедентов для проведения процедур сравнения реальных ситуаций с имеющейся ретроспективой.

Для поддержки в системе процессов самоорганизации создаются необходимые внешние и внутренние условия на основе механизмов реализации поиска и отбора информации, имеющей определенную меру ценности. Управляемое создание условий для протекания такого процесса самоорганизации усложняется неструктурированностью решаемых задач и неопределенностью надлежащего момента времени, в котором интеграция разрозненных элементов ценной информации приведет к формированию знания нового качества. Дальнейшее протекание процесса накопления ценной информации может привести к рассеиванию преимуществ интеграции.

В настоящее время решение проблемы анализа неструктурированных задач накопления и обработки знаний требует формализации процессов структуризации знаний, которая обычно выполняется экспертами. В работе авторы предлагают решать данную задачу на основе модифицированного алгоритма бактериальной оптимизации. Эта задача является классической задачей искусственного интеллекта, суть которой состоит в анализе междисциплинарных связей различных предметных областей на основе исследования онтологической структуры, позволяющей реализовать механизм идентификации информационных угроз в режиме реального времени.

## 2. Описание моделей поведения интеллектуальных агентов в Интернет-пространстве

Для обеспечения эффективной защиты от нежелательных воздействий Интернет-пространства на пользователя используем событийный подход. Произошедшее в текущий момент времени событие не обязательно является непосредственной угрозой, но с течением времени может представлять опасность как одно из звеньев цепочки воздействий, суммарный вред которых превышает допустимый предел. Отслеживание таких событий и их последовательности авторы предлагают проводить с учетом исследований психологов [1–4] в прецедентах, сведенных в онтологическую структуру. Данную структуру представим в виде следующей модели:

$$O = \langle P, V, R, C \rangle,$$

где  $P$  — множество понятий;  $V$  — множество экземпляров понятий;  $R$  — множество предикатов (атрибутов) понятий;  $C$  — множество отношений.

Опишем полный цикл поведенческой модели субъекта, включающий следующие этапы:

- 1) идентификация текущего события как эквивалента одного из концептов онтологии прецедентов;
- 2) поиск траекторий возможного развития событий в целях обнаружения информационных угроз;
- 3) совершенствование онтологии прецедентов на основе анализа полученных траекторий;
- 4) учет и распределение изменений в модели поведения субъекта;
- 5) выработка уточненного плана действий;
- 6) сохранение результатов анализа текущей обстановки.

На основе описанных этапов построим интегрированную модель реакции субъекта на воздействие в контексте процессов взаимодействия с Интернет-пространством (рис. 1).

Большинство из описанных этапов влияют на изменения в самой модели поведения и инструментах ее настройки и доработки. Отметим, что при совпадении характеристик текущего события с прототипом из онтологии прецедентов представленная модель может быть упрощена.

При полной эквивалентности текущего события и однозначности выбранной траектории, не требующей изменения модели поведения, цикл реакции субъекта на воздействие примет вид, представленный на рис. 2.

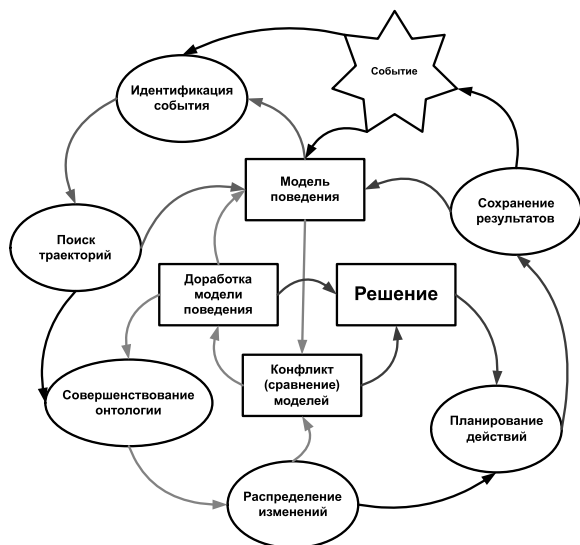


Рис. 1. Интегрированная модель реакции субъекта на воздействие

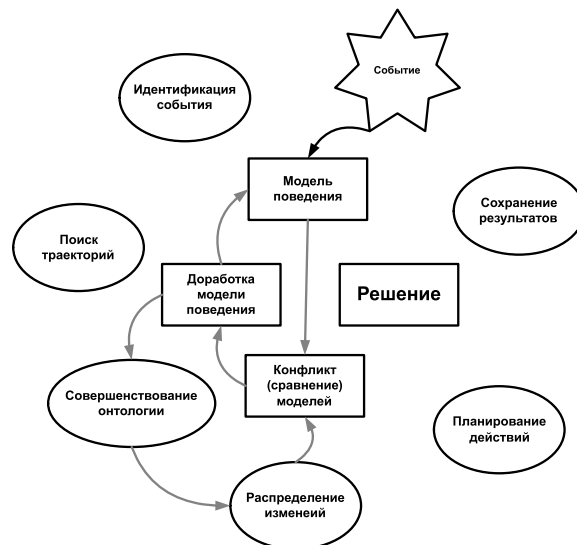


Рис. 3. Цикл доработки модели поведения

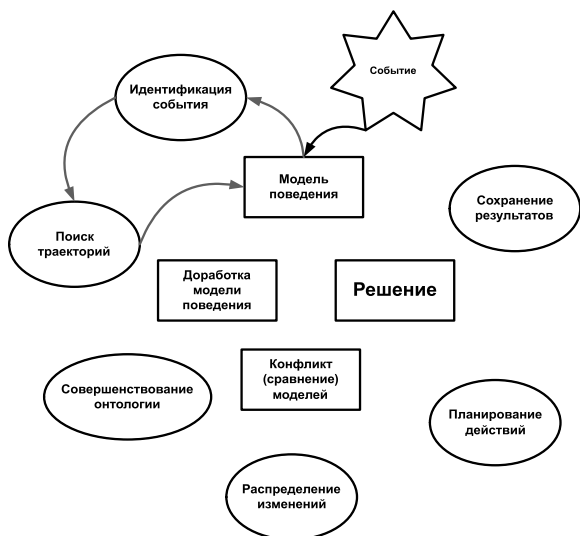


Рис. 2. Упрощенный цикл поведения субъекта

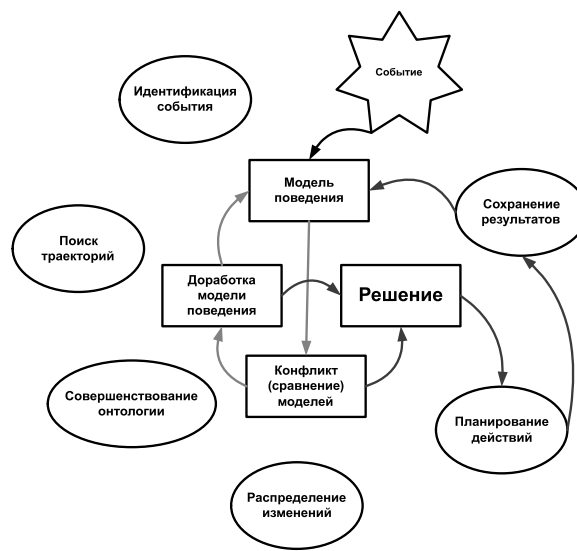


Рис. 4. Цикл принятия решений

Возможен вариант решения, предполагающий обработку новых событий только на основе применения имеющейся модели поведения субъекта. В этом случае пропускаются этапы идентификация текущего события как эквивалента одного из концептов онтологии прецедентов [9, 10] и поиска траекторий возможного развития событий в целях обнаружения информационных угроз. При этом обработка неизвестных событий будет происходить на основе сравнения и доработки модели поведения субъекта. Результат данной процедуры отразится в онтологии прецедентов (рис. 3).

Завершение процесса доработки модели поведения субъекта активирует этап принятия решений с последующим планированием мер по уменьшению последствий отрицатель-

ного воздействия и сохранением полученных результатов анализа текущих событий при взаимодействии с Интернет-пространством (рис. 4).

Данные модели поведения субъекта с учетом его индивидуальных особенностей используются при разработке алгоритмов функционирования интеллектуальных систем-ассистентов и автоматизированных систем-наставников. В связи с большой размерностью решаемых задач и высокой степенью информационной неопределенности авторы предлагают для получения набора квазиоптимальных решений применять методы биоинспирированного поиска [7, 11–14], в частности, один из методов роевого интеллекта — алгоритм бактериальной оптимизации [5, 6, 8, 9, 12, 15].

### 3. Постановка задачи

Каждое происходящее при использовании Интернет-сервисов событие имеет собственный набор атрибутов, характеризующих его с точки зрения особенностей текущего момента сетевого взаимодействия, а также со стороны специфики психофизиологического воздействия на личность и системы отношений с множеством наиболее вероятных предшествующих и последующих событий, называемых прецедентами.

Рассмотрим случайно выбранное текущее событие  $P_j^{real}$ ,  $j \in [1 : N]$ , где  $N$  — заданное экспертным путем максимальное число событий, рассматриваемых в одном цикле уточнения модели поведения субъекта в Интернет-пространстве. Для идентификации эквивалентности текущего события  $P_j^{real}$  с имеющимися прецедентами сравним его подмножество атрибутов  $R_j^{real} = \{R_{j1}^{real}; R_{j2}^{real}; \dots; R_{jk_j}^{real}\}$ , где  $j$  — порядковый номер текущего события,  $k_j$  — число атрибутов события  $P_j^{real}$ , с элементами множества предикатов  $R^{ont}$  концептов  $P_i^{ont}$ , где  $i \in [1 : M]$ ,  $M$  — число концептов онтологии  $O$ . Каждый элемент множества предикатов  $R^{ont} = \{R_1^{ont}; R_2^{ont}; \dots; R_M^{ont}\}$  является подмножеством конкретных атрибутов  $R_i^{ont} = \{R_{i1}^{ont}; R_{i2}^{ont}; \dots; R_{ik_i}^{ont}\}$ , принадлежащих каждому из понятий (концептов) онтологии, где  $i$  — порядковый номер концепта онтологии,  $k_i$  — число атрибутов обрабатываемого концепта  $P_i^{ont}$ .

Эквивалентность атрибутов текущего события и концептов онтологии определим на основе оценки семантической близости описанных подмножеств [10]  $R_j^{real}$  и части из подмножеств  $R_i^{ont}$ ,  $i \in [1 : M]$ . При обнаружении определенного сходства между атрибутами события и прецедента организуется поиск соответствия траекторий воздействия Интернет-сервиса на психофизиологическое состояние пользователя как в ретроспективе рассматриваемого события, что позволит повысить объективность прогноза, так и в перспективе. Поиск в перспективе является основной процедурой и осуществляется на основе оценки соответствия текущему событию понятий онтологии прецедентов. В свою очередь, для проверки корректности полученного прогноза в ретроспективе оценивается сходство события, произошедшего перед текущим, с понятиями онтологии прецедентов. Таким образом, множества атрибутов текущего и предшествующего событий являются статическими элементами,

а подмножества атрибутов концептов онтологии прецедентов — динамическим полем поиска. Множество атрибутов предшествующего события извлекается из памяти и определяется следующим образом:

$$R_{j-X}^{real} = \{R_{(j-X)1}^{real}; R_{(j-X)2}^{real}; \dots; R_{(j-X)k_{(j-X)}}^{real}\},$$

где  $X$  — случайное целое число, определяющее число шагов от текущего события к предшествующему, такое что  $\forall j \in [1; N]$ ,  $j - X > 0$ ,  $k_{(j-X)}$  — число атрибутов события  $P_{j-X}^{real}$ . Тогда целевая функция описанной задачи примет вид:

$$f(R_j^{real}, R_{j-X}^{real}, R^{ont}) \rightarrow \max,$$

а значение целевой функции будет вычисляться на основе следующего выражения:

$$f(R_j^{real}, R_{j-X}^{real}, R^{ont}) = \sum_{i=1}^m ((R_j^{real} \cap R_i^{ont}) + (R_{j-X}^{real} \cap R_i^{ont})) \geq d,$$

где  $d > 0$ ,  $d$  — пороговое значение, равенство или превышение которого означает присутствие значимой семантической близости между подмножествами атрибутов рассматриваемых событий и концептов онтологии прецедентов нежелательного воздействия Интернет-сервисов на психоэмоциональное и когнитивное состояние потенциального пользователя. Чем выше показатель  $d$ , тем более значимой является семантическая близость между подмножествами атрибутов.

### 4. Модифицированный алгоритм бактериальной оптимизации

Для проведения эффективных поисковых процедур в реальном масштабе времени соответствия атрибутов возникающих событий возможным отрицательным прецедентам, организованным в виде онтологических структур, авторами предложена модифицированная версия популяционного алгоритма бактериальной оптимизации на основе роевой модели поведения агентов-бактерий, обладающего преимуществами децентрализованного управления и распараллеливания вычислений [15]. Параллельность вычислений обеспечивается колонией бактерий, размерность которой имеет значение  $S$  и определяет число используемых в процедурах поиска агентов.

Основными этапами работы алгоритма бактериальной оптимизации являются хемотаксис,

репродукция и ликвидация (рассеивание) [8, 15]. Хемотаксис — движение бактерии в направлении высокой концентрации питательных веществ. При попадании в неблагоприятную для уровня здоровья бактерии среду она способна сделать "кувырок", изменяя направление движения от репеллента к аттрактанту. Репродукция — процесс увеличения числа бактерий на успешных направлениях поиска за счет исключения и замены бактерий-неудачников, что позволяет ускорить сходимость алгоритма. Процесс ликвидации и рассеивания позволяет уничтожить некоторое число бактерий, чтобы воспроизвести их с новыми координатами местоположения в пространстве поиска, что позволяет алгоритму выйти из локальных оптимумов и улучшить результаты поиска.

При кодировке решений для устранения возможного несоответствия размерностей подмножеств атрибутов различных понятий и событий установлена максимальная размерность *MAX* шаблона массива для заполнения усредненными значениями атрибутов. Если остаются свободные ячейки в конкретных вариантах наборов атрибутов, незанятые поля заполняются значением *NULL*.

Работа алгоритма начинается со случайного выбора для каждой бактерии набора атрибутов  $R_i^{ont}$  концепта онтологии прецедентов из множества  $R^{ont}$  в целях оценки его семантической близости с множеством атрибутов текущего события  $R_j^{real}$ . Тогда начальным условием для каждой бактерии заданной колонии станет пара множеств (рис. 5). Первый набор динамического элемента поиска является текущим положением  $X_{y, r, l} = X_{y, r, l}(t)$  бактерии  $s_y \in S$  на  $t$ -м шаге хемотаксиса,  $r$ -м шаге репродукции и  $l$ -м шаге ликвидации и рассеивания, здесь  $y \in [1:|S|]$ ,  $t \in [1:T]$ ,  $r \in [1:T']$ ,  $l \in [1:T'']$ , где  $T$ ,  $T'$ ,  $T''$  — общие числа шагов хемотаксиса, репродукции, ликвидации и рассеивания соответственно (свободные параметры алгоритма), а  $|S|$  — четное число бактерий в колонии. Число шагов репродукции предполагается вводить четным для дальнейшего определения в целочисленном значении середины протекания данной процедуры ( $r/2$ ) при реализации смены статуса элементов поиска. Число бактерий в колонии также задается четным для корректности проведения в дальнейшем процедуры репродукции, во

время которой исключается из рассмотрения половина ( $S/2$ ) наиболее слабых бактерий.

При отсутствии движения рассматриваемой бактерии, что выражается в локализации значений уровня текущего здоровья  $h_y$  бактерии  $s_y$  в средней части области допустимых значений, происходит изменение набора значений статичного элемента  $R_j^{real}$  на значения атрибутов  $R_{j-X}^{real}$  более раннего события. Для этого вводится переменная  $\alpha$  (рис. 5), которая изначально для атрибутов текущего события  $R_j^{real}$  имеет значение "1", а в данном случае для перехода к  $R_{j-X}^{real}$  меняет свое значение на "0". Смена значений данной переменной означает смену статичного элемента подмножеств атрибутов, а соответственно и текущего положения бактерии (рис. 6).

Далее алгоритм реализует случайный выбор набора атрибутов  $R_i^{ont}$  концепта онтологии прецедентов из множества  $R^{ont}$  для оценки его семантической близости с множеством атрибутов одного из предшествующих событий  $R_{j-X}^{real}$ .

Процедура смены статичных элементов происходит на шаге репродукции  $r/2$ . Таким образом, обнаружение максимального подобия (эквивалентности) между наборами подмножеств атрибутов указывает на высокую семантическую близость концептов и события, которым данные атрибуты принадлежат. Для определения текущего направления вектора хемотаксиса бактерии  $V_y(\epsilon)$  по вершинам графа онтологии авторы вводят переменную  $\epsilon_m$ , где  $m \in [1:M]$ . Значение данной переменной соответствует локальной степени (мощности окрестности) вершин графа онтологии, в которой осуществляется поиск (передвижение бактерий). Выбор направления движения бак-

|                                     |                 |                 |     |                   |                                                                     |
|-------------------------------------|-----------------|-----------------|-----|-------------------|---------------------------------------------------------------------|
| Статичный элемент<br>(событие)      | $R_{j1}^{real}$ | $R_{j2}^{real}$ | ... | $R_{jMAX}^{real}$ | $\alpha = 1$ (перспектива)                                          |
| Динамический элемент<br>(онтология) | $R_{i1}^{ont}$  | $R_{i2}^{ont}$  | ... | $R_{iMAX}^{ont}$  | Мощность окрестности ( $\epsilon$ )<br>Глубина поиска ( $\lambda$ ) |

Рис. 5. Элементы подмножеств атрибутов текущего события и атрибутов концепта онтологии и их статусы

|                                     |                     |                     |     |                       |                                                                     |
|-------------------------------------|---------------------|---------------------|-----|-----------------------|---------------------------------------------------------------------|
| Статичный элемент<br>(событие)      | $R_{(j-X)1}^{real}$ | $R_{(j-X)2}^{real}$ | ... | $R_{(j-X)MAX}^{real}$ | $\alpha = 0$ (ретроспектива)                                        |
| Динамический<br>элемент (онтология) | $R_{i1}^{ont}$      | $R_{i2}^{ont}$      | ... | $R_{iMAX}^{ont}$      | Мощность окрестности ( $\epsilon$ )<br>Глубина поиска ( $\lambda$ ) |

Рис. 6. Элементы подмножеств атрибутов более раннего события и атрибутов концепта онтологии и их статусы



терии соответствует траектории к вершине с максимальной мощностью окрестности.

Для уменьшения времени работы алгоритма авторы вводят переменную глубины поиска  $\lambda > 1$ , которая ограничивает передвижение бактерии по одной из выбранных траекторий. При обработке онтологий сверхбольшой размерности задается переменная  $\eta > 1$ , устанавливающая шаг передвижения бактерии [15], т. е. моделирующая скорость передвижения бактерии.

Локальную оптимизацию в алгоритме реализует процедура хемотаксиса. Новая дислокация  $X'_{y,r,l}$  бактерии  $s_y$  на  $(t + 1)$ -м шаге хемотаксиса определяется следующим выражением:

$$X'_{y,r,l} = X_{y,r,l} + \eta_y V_y(\varepsilon, \lambda).$$

Вектор  $V_y(\varepsilon, \lambda)$  остается неизменным, если происходит увеличение значения целевой функции. В этом случае  $V'_y(\varepsilon, \lambda) = V_y(\varepsilon, \lambda)$ . При уменьшении значений целевой функции, т. е. уменьшении числа атрибутов, указывающих на сходство исследуемых концептов и события, бактерия совершает "кувырок" [15], возвращаясь в вершину с максимальной  $\varepsilon$ , определенной на предыдущем шаге хемотаксиса, после чего движение возобновляется по новой траектории. Такая последовательность действий не позволяет бактерии покинуть найденный локальный оптимум, а хемотаксис продолжается до тех пор, пока значение целевой функции стабильно растет.

Для ускорения сходимости алгоритма используется механизм репродукции [8, 15], позволяющий интенсифицировать протекание процесса поиска локальных оптимумов. Зададим текущий уровень здоровья (healthstatus) бактерии как сумму значений целевой функции во всех точках пройденной траектории:

$$h_y = \sum_{\tau=1}^T f(R_j^{real}, R_{j-X}^{real}, R^{ont}), y \in [1 : |S|].$$

Для проведения процедуры репродукции после вычисления всех значений  $h_y$ , номера бактерий заносятся в список в порядке убывания значений их здоровья, чтобы на  $(r + 1)$ -м шаге исключить из рассмотрения половину наиболее слабых агентов, а каждого агента-бактерию из выживших дублировать копией с координатами, равными координатам дублируемого агента.

Например, если выжившая бактерия  $s_q$ ,  $q \in [1 : |S|]$  имеет положение  $X_{q,r,l}$ , тогда после репродукции появится бактерия  $s_w$ , причем

$$w = \frac{|S|}{2} + q, X_{q,r+1,l} = X_{q,r,l}, X_{w,r+1,l} = X_{q,r,l}.$$

Таким образом, после процедуры репродукции общее число бактерий в колонии остается неизменным.

При решении проблемы преждевременной сходимости, т. е. выхода бактерий из найденных локальных оптимумов, в предлагаемом алгоритме проводится процедура ликвидации и рассеивания [8, 15]. Для определения момента запуска процедуры ликвидации и рассеивания авторы вводят переменную  $\theta_z$ , где  $z \in [1 : Z]$ ,  $Z$  — число процедур репродукции до начала ликвидации и рассеивания. Достижение максимального значения  $Z$  указывает на момент начала процедуры ликвидации и рассеивания, которая начинается с уничтожения случайным образом выбранного числа  $g < |S|$  бактерий. Вместо уничтоженных создается аналогичное число бактерий с начальными координатами в новых случайно выбранных парах подмножеств атрибутов концептов и события. Далее процесс выполнения алгоритма происходит итерационно до получения набора квазиоптимальных решений.

#### 4. Экспериментальные исследования

Для подтверждения эффективности предложенного подхода разработан программный модуль, включающий в себя функции построения онтологии, генерации набора атрибутов исследуемых событий и проведения оценки семантической близости с расчетом траекторий потенциальных информационных угроз на основе модифицированного алгоритма бактериальной оптимизации. В качестве значений атрибутов применены усредненные абстрактные показатели, полученные случайным образом с заданной вероятностью.

В результате проведенных комплексных исследований получена временная сложность предложенного модифицированного алгоритма (рис. 7). Временная сложность алгоритма (ВСА) в представленном примере составила  $O(n^2)$ , где  $n$  — число анализируемых алгоритмом входных данных, которым в данном случае является число вершин графа онтологии прецедентов. В представленной модификации алгоритма бактериальной оптимизации предусмотрено два вложенных цикла прохождения по вершинам графа онтологии: один для осуществления поиска эквивалентности текущего события и

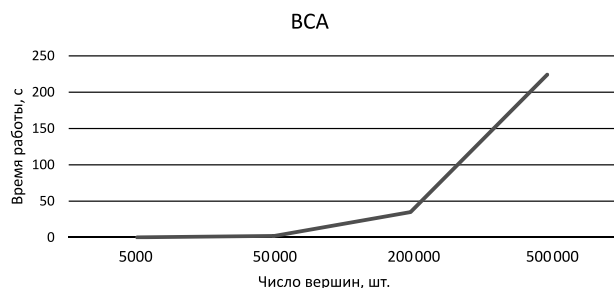


Рис. 7. ВСА в зависимости от числа вершин графа онтологии прецедентов

концепта онтологии прецедентов, а второй для обнаружения в ретроспективе рассматриваемого события возможной траектории нежелательного воздействия. Таким образом, максимально возможное число операций зависит от числа вершин графа онтологии как  $n * n$ .

Экспериментальные исследования предложенного алгоритма проводили на вычислительной системе с производительностью порядка 109 операций в секунду (Гфлопс). В этом случае, например, при  $n = 50\,000$  вершин время выполнения алгоритма не превысит 2,5 с, при  $n = 200\,000$  вершин — 40 с, при  $n = 500\,000$  вершин — 250 с, что наглядно иллюстрирует представленная на рис. 7 полученная зависимость времени выполнения предложенного алгоритма от числа вершин графа онтологии прецедентов.

Исследование скорости работы модифицированного алгоритма бактериальной оптимизации проводилось в сравнении с каноническим бактериальным алгоритмом и алгоритмом роя частиц (см. таблицу).

Из приведенных экспериментальных данных видно, что предложенный модифицированный алгоритм показал лучший результат по сравнению с каноническим алгоритмом и алгоритмом роя частиц. Эффективность предложенного алгоритма повышена за счет введения параметра мощности окрестности при выборе направления движения бактерии-агента и процедуры оценки семантической близости ретроспективы происходящих событий.

Сравнение времени работы алгоритмов

| Размерность графа | Модифицированный бактериальный алгоритм | Канонический бактериальный алгоритм | Алгоритм роя частиц |
|-------------------|-----------------------------------------|-------------------------------------|---------------------|
| 50 000            | 1,97                                    | 5,25                                | 2,77                |
| 200 000           | 34,85                                   | 65,81                               | 57,56               |
| 500 000           | 224,12                                  | 303,47                              | 279,44              |

## Заключение

В данной работе представлена разработка комплекса моделей поведения субъекта в интеллектуальных Интернет-сервисах, позволяющих идентифицировать несущие угрозу информационные события в процессе взаимодействия. Логика процессов оценки происходящих событий в данных моделях организована на основе применения биоинспирированного модифицированного алгоритма бактериальной оптимизации. Отметим, что колония бактерий значительной размерности формирует масштабные пространственно-временные структуры со сложной системой отношений, позволяющие повысить эффективность решения задачи оценки семантической близости при поиске информационных угроз в Интернет-пространстве.

При оценке качества работы предложенного модифицированного алгоритма бактериальной оптимизации были проведены экспериментальные исследования, подтверждающие эффективность разработки в сравнении с каноническим бактериальным алгоритмом и алгоритмом роя частиц. По проведенным оценкам временная сложность разработанного алгоритма не превышает полиномиальной.

## Список литературы

1. Lyz' N. A., Kupovich J. G., Prima A. K. Conceptual analysis and measurement of personal safety representations // SGEM International Multidisciplinary Scientific Conference on Social sciences and Arts. 2014. Vol. 1. P. 61—68.
2. Лызь Н. А., Куповых Ж. Г., Прима А. К. Опыт исследования представлений вузовской молодежи о безопасности // Вестник Пермского университета. Философия. Психология. Социология. 2016. № 3 (27). С. 86—95.
3. Лызь Н. А., Веселов Г. Е., Лызь А. Е. Информационно-психологическая безопасность в системах безопасности человека и информационной безопасности государства // Известия ЮФУ. Технические науки. 2014. Т. 157, № 8. С. 58—66.
4. Lyz' N., Opryshko A. Life-creating education: new educational meanings in modern world // On the Horizon. 2016. Vol. 24, Iss. 4. P. 377—386.
5. Abraham A., Grosan G., Ramos V. Swarm Intelligence in Data Mining // Berlin. Heidelberg: Springer Verlag, 2006. P. 267.
6. Курейчик В. В., Бова В. В., Лещанов Д. В. Модель семантического поиска в системах управления знаниями на основе генетических процедур // Информационные технологии. 2017. Т. 23, № 12. С. 876—883.
7. Kravchenko Y. A., Kursitys I. O., Bova V. V. Knowledge sifters in MDA technologies // International Conference Information Technologies in Business and Industry 2018 // IOP Conf. Series: Journal of Physics: Conf. Series 1015 (2018) 042024. doi: 10.1088/1742-6596/1015/4/042024.
8. Tripathy M., Mishra S. Bacteria Foraging-Based Solution to Optimize Both Real Poser Loss and Voltage Stability Limit // IEEE Transactions on Power Systems. 2007. N. 22 (1). P. 240—248.
9. Hu X., Shi Y., Eberhart R. C. (2004) Recent Advances in Particle Swarm // Proc. of Congress on evolutionary Computation (CEC). Portland, Oregon. P. 90—97.

10. Кравченко Ю. А. Модель фильтра знаний для задач семантической идентификации // Известия ЮФУ. Технические науки. 2018. № 4 (198). С. 175—185.

11. Гладков Л. А., Курейчик В. В., Курейчик В. М., Сороколетов П. В. Биоинспирированные методы в оптимизации. М.: Физматлит, 2009. 384 с.

12. Sousa T., Silva A., Neves A. (2004) Particle Swarm based Data Mining Algorithms for classification tasks // Parallel Computing. Vol. 30, Iss. 5—6. P. 767—783.

13. Родзин С. И., Курейчик В. В. Состояние, проблемы и перспективы развития биоэвристик // Программные системы и вычислительные методы. 2016. № 2. С. 158—172.

14. Родзин С. И., Курейчик В. В. Теоретические вопросы и современные проблемы развития когнитивных биоинспирированных алгоритмов оптимизации // Кибернетика и программирование. 2017. № 3. С. 51—79.

15. Карпенко А. П. Современные алгоритмы поисковой оптимизации: учеб. пособие. М.: Издательство МГТУ им. Н. Э. Баумана, 2014. 446 с.

V. V. Bova, Associate Professor, e-mail: vvbova@sfedu.ru,  
Yu. A. Kravchenko, Associate Professor, e-mail: yakravchenko@sfedu.ru,  
E. V. Kuliev, Associate Professor, e-mail: ekuliev@sfedu.ru,  
V. V. Kureichik, Head of CAD department, e-mail: vkur@sfedu.ru,  
Southern Federal University

## Behavior Modeling of the Subject in the Internet Services Based on the Bacterial Optimization Modified Algorithm

*This article is devoted to the solution of the fundamental and interdisciplinary scientific problem of identifying information events that pose a threat in the process of the subject's interaction with Internet services and develop effective strategies for preventing or reducing undesirable consequences. A special place in the solution of this problem takes the process of building models of intelligent agents' behavior. The large dimension of the problem being solved and the high degree of information uncertainty do not allow the use of mathematical analysis methods. In this paper, a modified bacterial algorithm is used to obtain quasioptimal solutions. In contrast to the canonical form, the proposed algorithm introduces new mechanisms to support search procedures. The specific scientific result was the swarm model of agents' behavior whose purpose of search procedures is to assess in real time the correspondence of the events' attributes to possible negative precedents organized in the form of ontological structures. The obtained results are relevant and in demand when creating methods and algorithms for using artificial intelligence to simulate, predict and support effective and psychologically safe informational and educational activities of a person in the Internet space. The work is directed to the study of this problem and the development of intelligent assistant systems that optimize educational activities in the Internet space, based on the methodology of biologically plausible machine learning methods. The developed models of information and educational activities and scenarios of the subject's behavior, taking into account its individual characteristics, will form the basis for the development of intelligent assistant systems and automated tutoring systems. Biologically plausible machine learning methods that provide the most effective solution to the problems of classification, clustering, recognition, optimization and coding of information will make it possible to find solutions that are close to optimal for an acceptable time in conditions of various kinds' Internet resources avalanche-like growth, on the one hand, it has a purposeful, self-organizing and self-developing character, on the other — signs of uncertainty and risk.*

**Keywords:** information threats, ontological structures, intelligent agents, swarm algorithms, behavior patterns

**Acknowledgements:** The study was carried out with the financial support of the Russian Foundation for Basic Research in the framework of the research project No.18-29-22019.

DOI: 10.17587/it.25.397-404

### References

1. Lyz' N. A., Kupovich J. G., Prima A. K. *SGEM International Multidisciplinary Scientific Conference on Social sciences and Arts*, 2014, vol. 1, pp. 61—68.

2. Lyz' N. A., Kupovich J. G., Prima A. K. *Vestnik Permskogo universiteta. Filosofija. Psihologija. Sociologija*, 2016, no. 3 (27), pp. 86—95 (in Russian).

3. Lyz' N. A., Veselov G. E., Lyz' A. E. *Izvestija JuFU. Tehnicheskie nauki*, 2014, vol. 157, no. 8, pp. 58—66 (in Russian).

4. Lyz' N., Opryshko A. *On the Horizon*, 2016, vol. 24, iss. 4, pp. 377—386.

5. Abraham A., Grosan G., Ramos V. *Swarm Intelligence in Data Mining*, Berlin, Heidelberg, Springer Verlag, 2006, 267 p.

6. Kureichik V. V., Bova V. V., Leshchanov D. V. *Informacionnye Tekhnologii*, 2017, vol. 23, no. 12, pp. 876—883 (in Russian).

7. Kravchenko Y. A., Kursitys I. O., Bova V. V. *International Conference Information Technologies in Business and Industry*

2018 — *IOP Conf. Series: Journal of Physics: Conf. Series*, 1015 (2018), 042024, doi: 10.1088/1742-6596/1015/4/042024.

8. Tripathy M., Mishra S. *IEEE Transactions on Power Systems*, 2007, no. 22 (1), pp. 240—248.

9. Hu X., Shi Y., Eberhart R. C. *Proceedings of Congress on evolutionary Computation (CEC)*, 2004, Portland, Oregon, pp. 90—97.

10. Kravchenko Yu. A. *Izvestija JuFU. Tehnicheskie nauki*, 2018, no. 4 (198), pp. 175—185 (in Russian).

11. Gladkov L. A., Kureichik V. V., Kureichik V. M., Sorokoleto P. V. *Bioinspired methods in optimization*, Moscow, Fizmatlit, 2009, 384 p. (in Russian).

12. Sousa T., Silva A., Neves A. *Parallel Computing*, 2004, vol. 30, iss. 5—6, pp. 767—783.

13. Rodzin S. I., Kureichik V. V. *Programmnye Sistemy i Vychislitel'nye Metody*, 2016, no. 2, pp. 158—172 (in Russian).

14. Rodzin S. I., Kureichik V. V. *Kibernetika i Programirovanie*, 2017, no. 3, pp. 51—79 (in Russian).

15. Karpenko A. P. *Sovremennye algoritmy poiskovoj optimizacii*, Moscow, Publishing house of MGTU im. N. Je. Bauman, 2014, 446 p. (in Russian).

**А. Н. Шниперов**, канд. техн. наук, доц., e-mail: ashniperv@sfu-kras.ru,  
Сибирский федеральный университет, г. Красноярск,  
**М. С. Сосновский**, инженер-программист, e-mail: maximsosnovskiy@gmail.com,  
**П. М. Шипулин**, инженер-программист, e-mail: pshipulin@gmail.com,  
Акционерное общество "Информационные спутниковые системы"  
имени академика М. Ф. Решетнева", г. Железногорск

### Робастный метод маркирования изображений цифровым водяным знаком, основанный на ортогональных моментах Цернике

*Рассмотрена проблема защиты авторских прав на цифровые изображения посредством их маркировки невидимым цифровым водяным знаком. Предложен новый способ маркирования полноцветных изображений, основанный на использовании моментов изображения Цернике в особых точках на маркируемом изображении, что позволило получить приемлемую скорость встраивания и извлечения цифрового водяного знака (ЦВЗ). Приведены базовые алгоритмы, а также полученные экспериментальные результаты оценки робастности к основным видам атак и временной сложности внедрения и извлечения ЦВЗ, подтверждающие эффективность предлагаемого метода.*

**Ключевые слова:** цифровые водяные знаки, стеганография, маркирование изображений, защита информации, защита авторских прав, моменты изображений, моменты Цернике

#### Введение

В последнее десятилетие наблюдается устойчивый темп роста производства и оборота цифрового контента, специфика которого накладывает значительные трудности в обеспечении защиты и охраны авторского права на этот контент. Изображения, включая фотографии, картографическую информацию, фонды электронных архивов и т. д., как разновидность цифрового контента занимают серьезную долю этого оборота. В силу простоты распространения такого контента у ее правообладателей очень часто возникает задача защиты своих авторских прав, однако на настоящий момент эта задача по-прежнему полностью не решена. Традиционно для защиты авторского права в таких изображениях применяются видимые водяные знаки, наносимые поверх всего изображения или его части. Однако такой подход во многих случаях не является приемлемым, а в некоторых и вовсе не решает задачу защиты авторского права. Это обстоятельство уже достаточно давно обозначило научно-техническую задачу по созданию различных подходов к быстрому маркированию изображений прозрачным (невидимым для че-

ловеческого глаза) цифровым водяным знаком (ЦВЗ), устойчивым к различным воздействиям на цифровое изображение в целях его деструкции. Концепция маркирования изображения ЦВЗ, как правило, лежит в области методов стеганографии. Однако требование к робастности для систем маркирования ЦВЗ существенно затрудняет использование многих известных методов стеганографии и ставит серьезную научно-техническую задачу перед исследователями. В данной статье для решения данной задачи предлагается метод маркирования ЦВЗ полноцветных изображений, который базируется на использовании моментов Цернике [1] в особых точках на маркируемом изображении [2], что позволяет работать с изображениями любых размеров, сохраняя робастные свойства метода и приемлемую скорость как внедрения ЦВЗ в изображение, так и его извлечения.

#### 1. Обзор подходов к маркированию изображений ЦВЗ и постановка задачи

Маркирование изображений посредством ЦВЗ в общем виде рассматривается как проблема передачи информационного сигнала

малой мощности в широкополосном сигнале. При этом выдвигаются несколько ключевых условий к системам маркирования изображений посредством ЦВЗ:

- вносимые ЦВЗ искажения в информационный широкополосный сигнал (изображение-контейнер) должны быть неразличимы для человеческого глаза;
- ЦВЗ, внедренный в изображение, должен обладать высокими показателями робастности, т. е. быть устойчивым к различным искажениям изображения-контейнера, как не преднамеренными, так и преднамеренными;
- приемлемая вычислительная сложность внедрения и извлечения ЦВЗ.

За последнее десятилетие различными исследователями было предложено множество различных методов маркирования статичных изображений, более того уже предложены подходы к их классификации [3, 4]. Так, по области встраивания методы маркирования обычно классифицируют на пространственные, частотные и моментные.

Пространственные методы встраивания ЦВЗ основаны на манипулировании цветовыми составляющими изображения или его яркостью в пространственной области. В этом случае изображение  $C_{M \times N}$  определяется как упорядоченное множество пикселей  $C_{M \times N} = \langle c(x, y) \rangle$ , для каждого из которого определен вектор цветовой составляющей каналов цветности модели *RGB* и яркости. Общий принцип всех пространственных методов заключается в замене избыточной информации об изображении, малозаметной человеческому глазу, битами ЦВЗ. Как правило, осуществляется манипуляция наименее значащими битами (*Least Significant Bit, LSB*) каналов цветности и/или яркости пикселей изображения по какому-то определенному алгоритму, линейному или нелинейному. Однако в силу того, что пространственные методы встраивания обладают низкой устойчивостью к воздействиям на изображения, а также ограниченностью применимости к графическим форматам, для маркирования ЦВЗ они практически не рассматриваются.

Частотные методы, являющиеся подклассом методов преобразования области изображения (*Transform domain methods*), основаны на "подмешивании" ЦВЗ в некоторые области исходного изображения. При этом процесс встраивания происходит в частотной (спектральной) области изображения посредством модификации конечного набора спектральных коэффициентов разложения изображения посредством

различных дискретных ортогональных преобразований, используемых в алгоритмах сжатия изображений с потерями. Этими преобразованиями могут выступать дискретное косинусное преобразование, дискретное вейвлет-преобразование, дискретное преобразование Фурье и другие. Данное направление развития систем маркирования изображений цифровым водяным знаком является весьма популярным, а также позволяет достигнуть высоких показателей робастности [5–9]. Однако частотные методы имеют довольно высокую чувствительность к исходному изображению, так как внедрение осуществляется, как правило, преимущественно в низкочастотную область в целях обеспечения робастности ЦВЗ. Неоптимальное внедрение ЦВЗ может привести к значительному искажению промаркированного изображения, при этом критерии оптимальности весьма зависимы от исходного изображения. Кроме того, частотные методы обладают низкими показателями устойчивости к атакам с использованием фильтрации, а также атакам на основе аффинных преобразований.

Моментные методы маркирования ЦВЗ являются сравнительно новым подходом к проблематике и основаны на встраивании ЦВЗ в моменты изображений, которые представляют собой дескрипторы особенностей изображений. При этом выделяют следующие виды моментов: геометрические и ортогональные [10]. В работе [11] впервые были представлены геометрические моменты, которые обладали свойствами устойчивости к вращению, масштабированию и смещению изображения. Данные свойства впоследствии были использованы в различных методах классификации и распознавания изображений, однако их неортогональность существенно усложняла задачу восстановления изображения и, как следствие, применимость геометрических моментов в задаче маркирования ЦВЗ. Основанные на теории ортогональных полиномов дальнейшие исследования показали, что изображения могут быть достаточно просто восстановлены рядом ортогональных моментов, таких как моменты Цернике, Чебышева, Лежандре и др. Это стало толчком для исследования применимости их в методах маркирования ЦВЗ, описанных в работах [12, 13, 14] и в ряде других работ. Данные моменты, по сравнению с геометрическими, также менее чувствительны к шуму и инвариантны к линейным преобразованиям (геометрическим атакам). Однако общим системным и весьма существенным недостатком



использования ортогональных моментов в методах маркирования изображений ЦВЗ являются большие вычислительные затраты на их вычисление, экспоненциально возрастающие по мере увеличения размера маркируемого изображения. Вследствие этого авторы данных работ в ходе проведения экспериментов и оценки эффективности в основном ограничиваются небольшими размерами маркируемых изображений, к тому же в градациях серого.

В данной работе предлагается новый **метод маркирования изображений любых размеров цифровым водяным знаком в особых точках на основе моментов изображений Цернике**, обладающий приемлемой вычислительной сложностью и устойчивый к различным атакам, в том числе к шумовым и геометрическим атакам на основе сжатия, фильтрации и аффинных преобразований.

## 2. Вычисление моментов Цернике и реконструкция изображения по ним

Вычисление двумерных моментов изображений Цернике может быть произведено с использованием полной системы ортогональных комплексных функций [12]:

$$v_{nm}(x, y) = v_{nm}(\rho, \theta) = R_{nm}(\rho)e^{jm\theta},$$

где  $x, y$  и  $\rho, \theta$  представлены в декартовой и полярной системах координат соответственно;  $R_{nm}(\rho)$  — радиальный полином Цернике;

$\rho = \sqrt{x^2 + y^2}$ ,  $j = \sqrt{-1}$ ,  $\theta = \arctg\left(\frac{y}{x}\right)$ ,  $|r| \leq 1$ ,  $n \geq 0$  — радиальный порядок момента,  $m$  — азимутальный порядок момента, причем  $|m| \leq n$ .

В работе [15] радиальный полином Цернике определяется как

$$R_{nm}(\rho) = \sum_{s=0}^{\frac{n+|m|}{2}} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} \rho^{n-2s}. \quad (1)$$

Значения комплексных ортогональных моментов Цернике для изображения размером  $N \times N$  определяются как

$$z_{nm} = \frac{n+1}{\pi} \iint_{x^2+y^2 \leq 1} f(x, y) v_{nm}^*(\rho, \theta) dx dy,$$

где функция  $v_{nm}^*(\rho, \theta)$  комплексно сопряжена с  $v_{nm}(\rho, \theta)$ . Функция  $f(x, y)$  представляет исходное изображение фиксированного размера.

Моменты Цернике являются комплексными функциями и поэтому могут быть представлены в виде амплитудно-частотного (АЧС) и фазочастотного спектров (ФЧС). Однако необходимо отметить, что вычисляемые таким образом моменты Цернике в строгом смысле слова не являются точными. В данном случае возможны два вида ошибок, получивших название геометрической ошибки аппроксимации и численной ошибки аппроксимации [12]. Эти ошибки ведут к некоторым искажениям при последующей реконструкции изображения в пространственной области (рис. 1).

Для вычисления моментов Цернике цифровое изображение должно быть отображено в пределах единичной окружности с началом в центре изображения. Например, если нам необходимо вычислить моменты для изображения с размером  $N \times N$  пикселя, то оно должно быть нормировано в единичном круге с фиксированным радиусом  $N/2$  пикселя. Пиксели вне единичного круга будут отбрасываться при вычислении, тем самым это создает геометрическую ошибку при вычислении моментов. Увеличение до  $N\sqrt{2}$  радиуса окружности, охватывающей все изображение, приводит к нивелированию геометрической ошибки, но увеличивает область вычислений.

По известным моментам Цернике  $z_{nm}$  можно реконструировать исходное изображение, используя функцию реконструкции, которая в общем виде имеет вид

$$\tilde{f}(x, y) = \sum_{n=0}^{n_{\max}} \sum_m z_{nm} v_{nm}(\rho, \theta). \quad (2)$$

Заметим, что чем больше порядок используемых моментов  $n$ , тем ближе функция реконструкции  $\tilde{f}(x, y)$  к функции изображения  $f(x, y)$ , однако тем больше объем требуемых вычислений. На рис. 1 показаны исходное изображение



Рис. 1. Реконструирование изображения по моментам Цернике: а — исходное изображение; б — реконструированное изображение

бражение и реконструированное изображение с использованием моментов изображения Цернике  $n_{\max} = 36$  порядка.

### Описание метода маркирования изображений ЦВЗ и алгоритмов

Как уже было отмечено, предлагаемый метод маркирования изображений ЦВЗ базируется на применении ортогональных моментов Цернике, которые обладают необходимыми свойствами в задаче обеспечения робастности ЦВЗ: помехоустойчивости, независимости от контекста исходного изображения и инвариантности ко многим преобразованиям. Однако, как уже было отмечено, вычисление моментов изображений, в том числе и моментов Цернике, является весьма сложной вычислительной задачей. Так, временная сложность вычисления моментов Цернике для изображения размером  $N \times N$  максимального порядка  $n_{\max}$  при использовании базового метода вычисления (1) составляет  $O(N^2 n_{\max}^3)$ . Применение более современных подходов к вычислению моментов Цернике, например, приводимых в работах [16, 17], дает весьма значительный прирост в скорости. Так, например, метод вычисления Кинтера [18] имеет временную сложность  $O(N^2 n_{\max}^2)$ , однако для изображений больших размеров она все еще очень высокая.

Предлагаемый в статье метод маркирования ЦВЗ базируется на подходе, в котором для вычисления моментов изображения Цернике используется не все изображение целиком, а его особые локальные фрагменты. Данный подход достаточно часто применяется исследователями, например в работе [19]. При этом для определения этих локальных областей используются окрестности особых точек, которые, в свою очередь, являются некоторыми локальными инвариантными особенностями изображения. Такой подход позволяет существенно уменьшить объем вычислений, сделав временную сложность маркирования изображения ЦВЗ приемлемой. При этом использование нескольких особых окрестностей особых точек изображения позволяет сохранять робастность метода маркирования на приемлемом уровне, а сам алгоритм внедрения вносит минимальные визуальные искажения в оригинальное изображение.

### ■ Алгоритм внедрения ЦВЗ в изображение

Предлагаемый метод маркирования изображения ЦВЗ предполагает последовательное внедрение копии ЦВЗ в окрестность каждой  $s$ -й выбранной особой точки на изображении. Таким образом, в целом процесс внедрения ЦВЗ происходит многократно для каждой выбранной особой точки и ее окрестности. Общая структурная схема процесса внедрения ЦВЗ в изображение представлена на рис. 2.

**Шаг 1.** Цифровой водяной знак представляется в виде бинарной последовательности  $B = \{b_i \in \{0, 1\}, i = 1, 2, \dots, L\}$ .

**Шаг 2 (прекодирование ЦВЗ).** Осуществляется кодирование  $B$  с использованием алгоритма корректирующего кодирования. Выбор того или иного алгоритма, в первую очередь, связан с объемом ЦВЗ, а также с числом используемых особых точек на изображении. В рамках данной работы нами использовался код Хэмминга.

**Шаг 3 (нормализация изображения).** Изображение разбивается на каналы цветовой модели RGB, и для дальнейшей работы алгоритма берется только синий ( $B$ -канал), поскольку искажения в синем канале наименее заметны человеческому глазу. Для выбранного канала выполняется нормализация. Нормализация изображения выполняется для достижения наилучших показателей внедрения и извлечения ЦВЗ. Канал изображения рассматривается как функция  $f(x, y)$ , для которой находят минимальные и максимальные значения  $f_{\min}$  и  $f_{\max}$  соответственно. Далее  $f(x, y)$  может быть нормализована трансформацией в функцию

$$g(x, y) = \frac{f(x, y) - f_{\min}}{f_{\max} - f_{\min}} 255.$$

**Шаг 4 (нахождение особых точек).** Для нахождения области встраивания ЦВЗ необходимо определение дескриптора особой точки посредством ее детектирования. Для этой

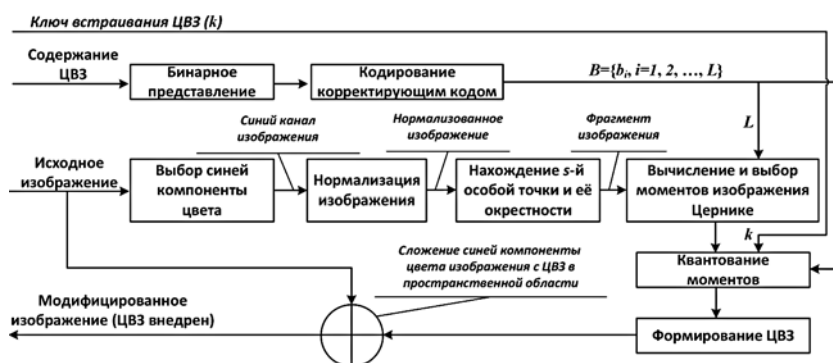


Рис. 2. Общая структурная схема внедрения копии ЦВЗ в изображение

цели в алгоритме используется детектор углов Ши-Томаси [20], который позволяет достаточно эффективно находить инвариантные к аффинным преобразованиям точки на изображении. Эти особые точки формируются из двух или более граней, которые, в свою очередь, обычно определяют границу между различными объектами и/или частями одного и того же объекта. Посредством детектора Ши-Томаси проводится поиск особых точек и выбирается самая стабильная из тех, которые еще не участвовали в процессе внедрения ЦВЗ. Вокруг выбранной особой точки вычисляются координаты окрестности области встраивания ЦВЗ размером  $256 \times 256$  пикселей.

**Шаг 5 (вычисление моментов и их выбор).** Вычисление и выбор моментов изображения Цернике. Для выбранной окрестности особой точки выполняется вычисление комплексных ортогональных моментов Цернике, а также значений радиального полинома, рассчитанного посредством метода Кинтера. Однако, как показывают исследования, в том числе в работах [12, 16, 17, 21], не все моменты могут быть задействованы для встраивания ЦВЗ. Так, моменты Цернике степени  $m$ -кратной четным, не могут быть вычислены точно, поэтому они не используются в дальнейших операциях и отбрасываются.

**Шаг 6 (квантование моментов).** Из отобранных моментов формируется вектор значений моментов Цернике  $Z_{nm} = \{z_{n_1m_1}, z_{n_2m_2}, \dots, z_{n_Lm_L}\}$  длиной  $L$  не менее объема внедряемого ЦВЗ. Каждый бит ЦВЗ  $b_i \in B$  встраивается в соответствующий элемент  $z_{n_im_i} \in Z_{nm}$  с помощью функции модуляции, которая осуществляет псевдослучайное изменение моментов Цернике на некоторую величину, которую будем называть "шумом". Для этого сначала вычисляется значение "шума", на которое необходимо изменить текущий момент изображения:

$$|\tilde{z}_{n_im_i}| = \left\lceil \frac{|z_{n_im_i}| - d(b_i)}{\Delta} \right\rceil \Delta + d(b_i), \quad (3)$$

где  $\Delta$  — шаг квантования, от которого зависит соотношение видимости и устойчивости ЦВЗ, а  $d(b_i)$  — функция дизеринга. Шаг квантования является настраиваемым параметром алгоритма. Значение функции дизеринга  $d(b)$  зависит от значения внедряемого бита. Для определения вектора всех ее значений сначала необходимо сформировать вектор значений

$$D_0 = \{d_1(0), d_2(0), \dots, d_L(0)\},$$

где  $d_i(0) \leftarrow \text{random}(k)$  и  $d_i(0) \in \left[0, \frac{\Delta}{2}\right]$ . Формирование значений  $d_i(0)$  осуществляется функцией  $\text{random}(k)$ , являющейся реализацией генератора псевдослучайных чисел (ГПСЧ), инициализированного ключом  $k$ . Затем формируется итоговый вектор значений  $D = \{d_i(b_i): i = 0, 1, \dots, L\}$ , в котором для формирования значений  $d_i(1)$  используется выражение  $d_i(1) = d_i(0) + \frac{\Delta}{2}$ .

Модификация исходных моментов Цернике осуществляется посредством внесения в них

"шума":  $\tilde{z}_{n_im_i} = \frac{|\tilde{z}_{n_im_i}|}{|z_{n_im_i}|} z_{n_im_i}$ ,  $i = 1, 2, \dots, L$ .

**Шаг 7 (формирование ЦВЗ).** После модификации выбранных моментов Цернике формируется сигнал водяного знака с использованием функции реконструкции изображения (2):

$$w(x, y) = \sum_{i=1}^L (\varepsilon_{n_im_i} V_{n_im_i} + \varepsilon_{n_i(-m_i)} V_{n_i(-m_i)}),$$

где  $\varepsilon_{n_im_i} = \tilde{z}_{n_im_i} - z_{n_im_i}$ ,  $\varepsilon_{n_i(-m_i)} = \tilde{z}_{n_i(-m_i)} - z_{n_i(-m_i)}$ ,  $i = 1, 2, \dots, L$ .

Итоговое изображение формируется путем сложения фрагмента исходного изображения в окрестности особой  $s$ -й точки и ЦВЗ в пространственной области:

$$\tilde{f}_s(x, y) = f_s(x, y) + w(x, y),$$

где  $f_s(x, y)$  — функция изображения в окрестности области встраивания ЦВЗ,  $w(x, y)$  — функция изображения ЦВЗ.

## ■ Алгоритм извлечения ЦВЗ из изображения

Алгоритм извлечения ЦВЗ из изображения практически аналогичен алгоритму встраивания и предполагает последовательное извлечение копии ЦВЗ из окрестности каждой  $s$ -й выбранной особой точки на изображении. Общая структурная схема процесса извлечения копии ЦВЗ из изображения представлена на рис. 3. После извлечения всех копий ЦВЗ осуществляется проверка их корректности (с заранее заданным порогом) и принимается решение об идентификации или неидентификации ЦВЗ в изображении.

**Шаг 1.** Осуществляется инициализация параметров извлечения (ключ извлечения —  $k$ , шаг квантования —  $\Delta$ , длина бинарной последовательности ЦВЗ —  $L$ ), предполагается, что цифровой водяной знак представлен в виде



Рис. 3. Общая структурная схема извлечения копии ЦВЗ из изображения

бинарной последовательности  $B = \{b_i \in \{0, 1\}, i = 1, 2, \dots, L\}$ , внедренной в окрестность  $s$ -й особой точки с использованием ключа  $k$ .

**Шаг 2 (нормализация изображения).** Осуществляется аналогично шагу 3 алгоритма встраивания.

**Шаг 3 (нахождение особых точек).** Осуществляется аналогично шагу 4 алгоритма встраивания.

**Шаг 4 (вычисление моментов и их выбор).** Осуществляется аналогично шагу 5 алгоритма встраивания.

**Шаг 5 (квантование моментов).** Из отобранных моментов формируется вектор значений моментов Цернике  $Z_{nm} = \{z_{n_1 m_1}, z_{n_2 m_2}, \dots, z_{n_L m_L}\}$  длиной  $L$ . Определяются векторы значений функции дизернга  $D_0 = \{d_1(0), d_2(0), \dots, d_L(0)\}$  и  $D_1 = \{d_1(1), d_2(1), \dots, d_L(1)\}$ , где  $d_i(0) \leftarrow \text{random}(k)$ ,  $d_i(0) \in \left[0, \frac{\Delta}{2}\right]$  и  $d_i(1) = d_i(0) + \frac{\Delta}{2}$ . Формирование значений  $d_i(0)$  осуществляется посредством ГПСЧ, инициализированного ключом извлечения  $k$ . Далее значение каждого момента изображения Цернике дважды квантуется с использованием выражения (3). При первом квантовании используются значения функции дизернга  $d_i(0) \in D_0$  с образованием модифицированных моментов  $|\tilde{z}_{n_i m_i}|_0$ , при втором квантовании используются значения функции дизернга  $d_i(1) \in D_1$  с образованием модифицированных моментов  $|\tilde{z}_{n_i m_i}|_1$ .

**Шаг 6 (восстановление ЦВЗ).** Минимальная разница между вычисленным моментом и квантованным (на предыдущем шаге) используется для нахождения бита, встроенного в текущий момент:

$$b'_i = \begin{cases} 1, & \text{если } (|\tilde{z}_{n_i m_i}|_1 - |z_{n_i m_i}|)^2 < (|\tilde{z}_{n_i m_i}|_0 - |z_{n_i m_i}|)^2; \\ 0 & \text{в противном случае,} \end{cases}$$

где  $i = 1, 2, \dots, L$ .

Другими словами, обнаружение битов ЦВЗ осуществляется по пикам в разнице моментов Цернике. После восстановления всех предполагаемых битов ЦВЗ  $B' = \{b'_i \in \{0, 1\}, i = 1, 2, \dots, L\}$  проводится их корректировка в  $B \leftarrow h(B')$  путем декодирования кода Хемминга.

## Эксперименты, оценка устойчивости и качества ЦВЗ

Предлагаемый в статье метод был реализован в виде законченного программного продукта [22], посредством которого проводилась оценка качества ЦВЗ и его устойчивости. Для оценки качества встроенного ЦВЗ и его устойчивости использовались различные распространенные метрики, среди которых:

- коэффициент ошибочных битов — *BER* (*Bit Error Rate*);
- пиковое соотношение сигнал/шум — *PSNR* (*Peak Signal-To-Noise Ratio*);
- среднеквадратическая ошибка между исходным изображением и его искаженной версией после внедрения в него ЦВЗ — *MSE* (*Mean Square Error*);
- индекс структурного сходства — *SSIM* (*Structural SIMilarity*).

Оценка данных метрик осуществлялась как на тестовых полутоновых изображениях размером  $256 \times 256$  пикселей с единичным внедрением ЦВЗ, так и на полноцветных изображениях различного содержания размерами до  $5184 \times 3456$  пикселей с внедрением копий ЦВЗ в окрестности особых точек. На рис. 4, б (см. третью сторону обложки) показан результат встраивания ЦВЗ объемом 256 бит во все тестовое полутоновое изображение "Lena" без обнаружения особых точек с шагом квантования  $\Delta = 2$ . В табл. 1 представлена зависимость метрик *PSNR*, *MSE*, *SSIM* от шага квантования, который отвечает за робастность ЦВЗ, а также за его незаметность.

Из табл. 1 видно, что показатели *PSNR* достаточно высоки, а показатели *SSIM* незначительно отличны от единицы, что говорит о вполне удовлетворительном качестве маркированного изображения в сравнении с исходным.

Как уже было отмечено ранее, работа с полноцветными изображениями осуществляется с использованием синей компоненты *RGB*-

модели. С математической точки зрения синяя компонента представляется матрицей, как и полутоновое изображение, где пиксели принимают значения в интервале [0, 255]. Поэтому ожидаемые характеристики метрик не должны принципиально изменяться при встраивании ЦВЗ в полноцветные изображения. Для проверки данной гипотезы нами использовались 30 широкоформатных полноцветных изображений формата *JPEG* различного содержания: пейзажи с разной погодой, пейзажи с водоемами, изображение мостов, макросъемка и портреты. *Встраивание копии ЦВЗ осуществлялось в окрестностях трех особых точек.* На рис. 5 (см. третью сторону обложки) представлены

Таблица 1  
Зависимости метрик от шага квантования  
для изображения "Lena"

| Шаг квантования, Δ | PSNR, дБ | MSE     | SSIM    |
|--------------------|----------|---------|---------|
| 0,8                | 46,96990 | 1,30644 | 0,99605 |
| 1                  | 45,44226 | 1,85717 | 0,99461 |
| 1,2                | 43,84681 | 2,62059 | 0,99381 |
| 1,4                | 42,52811 | 3,63302 | 0,99057 |
| 1,6                | 41,74942 | 4,34648 | 0,98908 |
| 1,8                | 40,69659 | 5,53884 | 0,98537 |
| 2                  | 39,36072 | 7,53369 | 0,98433 |

Таблица 2  
Зависимость метрик PSNR, MSE и SSIM от объема ЦВЗ

| Объем ЦВЗ, бит | PSNR, дБ | MSE     | SSIM    |
|----------------|----------|---------|---------|
| 64             | 54,48896 | 0,2313  | 0,99778 |
| 128            | 54,11923 | 0,25185 | 0,99764 |
| 256            | 53,51123 | 0,2897  | 0,99741 |

Таблица 3  
Время работы алгоритма по встраиванию и извлечению ЦВЗ

|                | Область встраивания<br>256×256 пикселей | Время работы, с |            |
|----------------|-----------------------------------------|-----------------|------------|
|                | Размер изображения,<br>пиксели          | Встраивание     | Извлечение |
|                | 2560 × 1680                             | 5               | 4,2        |
|                | 4000 × 2667                             | 6,2             | 4,9        |
| ЦВЗ<br>160 бит | 4000 × 2667                             | 6,4             | 5          |
|                | 5184 × 3456                             | 7,6             | 5,7        |
|                | 5184 × 3456                             | 7,6             | 5,7        |
|                | 4166 × 2343                             | 6,8             | 5,7        |
|                | 1024 × 1024                             | 4,7             | 3,7        |
|                | 4166 × 2343                             | 6               | 4,6        |

усредненные значения метрики *PSNR* в зависимости от шага квантования Δ для ЦВЗ различного объема. Как можно заметить, значения метрики вполне удовлетворительные, что также подтверждают сами изображения: промаркированные ЦВЗ визуально не отличимы от оригинальных.

В табл. 2 приведены усредненные значения основных метрик качества внедренного ЦВЗ.

На серии выбранных изображений были проведены тесты по определению зависимости коэффициента ошибок *BER* от размера ЦВЗ и области встраивания. По результатам тестов можно сделать вывод, что оптимальный размер встраиваемого ЦВЗ — 160 бит, а размер области для встраивания — 256×256 пикселей, поскольку при таком подборе параметров наблюдался минимум ошибок при извлечении ЦВЗ из маркированных изображений.

Скоростные характеристики работы алгоритмов проверялись с использованием персонального компьютера под управлением операционной системы ОС Linux Ubuntu 16.4, с процессором Intel Core i5-2410M и ОЗУ 4 ГБ. Среднее время работы алгоритмов встраивания и извлечения трех копий ЦВЗ в окрестностях особых точек для тестовых изображений отражено в табл. 3. Нормализация изображения в зависимости от его размера занимает от 0,2 до 1 с, остальное время занимает вычисление моментов Цернике и дальнейшее их внедрение или извлечение.

Для оценки робастности внедренного ЦВЗ проводили различные атаки на тестовые, маркированные полноцветные изображения, в которые записана одна копия ЦВЗ размером 160 бит с шагом квантования моментов Цернике Δ = 2, так как использование нескольких копий ЦВЗ увеличивает вероятность извлечения ЦВЗ. В силу ограниченного объема статьи далее рассмотрим лишь такие воздействия на изображения, которые являются наиболее распространенными: *JPEG*-сжатие, отражение, масштабирование, повороты, обрезка и зашумление. Данные виды воздействий приводят к необратимым изменениям маркированного изображения и, как следствие, представляют реальную угрозу для ЦВЗ. Основным показателем качества алгоритма встраивания ЦВЗ является устойчивость к выбранным типам атак. Показатель измеряется метрикой *BER* и является усредненным по значениям, полученным для каждого изображения в эксперименте.

**Устойчивость к сжатию.** Для проверки устойчивости к *JPEG*-сжатию маркированное

изображение подвергалось сжатию в диапазоне значений коэффициента качества 100...70. Результат представлен в табл. 4.

Таблица 4  
Зависимости *BER* от уровня сжатия для тестового изображения

| Качество изображения | <i>BER</i> | Качество извлеченного ЦВЗ (160 бит), % |
|----------------------|------------|----------------------------------------|
| 100                  | 0          | 100                                    |
| 95                   | 0          | 100                                    |
| 90                   | 0,023      | 93,75                                  |
| 85                   | 0,01567    | 93,75                                  |
| 80                   | 0,046      | 87,5                                   |
| 75                   | 0,015      | 93,75                                  |
| 70                   | 0,117      | 62,25                                  |

Таблица 5  
Влияние отражения изображения на ЦВЗ

| Наименование искажения   | <i>BER</i> | Качество извлеченного ЦВЗ (160 бит), % |
|--------------------------|------------|----------------------------------------|
| Вертикальное отражение   | 0          | 100                                    |
| Горизонтальное отражение | 0          | 100                                    |

Таблица 6  
Зависимости *BER* от коэффициента масштабирования

| Коэффициент масштабирования | <i>BER</i> | Качество извлеченного ЦВЗ, % |
|-----------------------------|------------|------------------------------|
| 1,5                         | 0,03125    | 96,875                       |
| 1,25                        | 0,0625     | 93,75                        |
| 1                           | 0          | 100                          |
| 0,87                        | 0,09375    | 90,625                       |
| 0,75                        | 0,10526    | 86,41                        |
| 0,5                         | 0,25       | 81,25                        |

Таблица 7  
Зависимости *BER* от угла поворота изображения

| Угол поворота, ° | <i>BER</i> | Качество извлеченного ЦВЗ (160 бит), % |
|------------------|------------|----------------------------------------|
| 1                | 0,0859     | 62,5                                   |
| 2                | 0          | 100                                    |
| 3                | 0          | 100                                    |
| 4                | 0,007      | 93,75                                  |
| 5                | 0,14       | 43,75                                  |
| 6                | 0,039      | 87,5                                   |
| 7                | 0,039      | 87,5                                   |
| 8                | 0,0625     | 68,75                                  |
| 9                | 0,039      | 87,5                                   |
| 10               | 0,039      | 87,5                                   |

**Устойчивость к отражению.** Маркированное изображение подвергалось атаке в виде отражений (табл. 5). Как показывают результаты, ЦВЗ абсолютно устойчив к данной атаке.

**Устойчивость к масштабированию.** Маркированное изображение подвергалось атаке в виде масштабирования (пропорциональное изменение размера), результат которой представлен в табл. 6, из которой видно, что ЦВЗ весьма устойчив к данной атаке.

**Устойчивость к поворотам.** Маркированное изображение подвергалось атаке в виде поворота на углы от 0 до 10°. Под поворотом здесь понимается вращение изображения относительно неподвижного холста (подложки). Результаты представлены в табл. 7 и показывают, что ЦВЗ устойчив к небольшим углам поворота.

**Устойчивость к обрезке.** Маркированные изображения подвергались атаке в виде обрезки. В случае если обрезка не затрагивает особую точку и ее окрестность, данная атака не нарушает ЦВЗ. Встраивание нескольких копий ЦВЗ в окрестности различных особых точек изображения повышает стойкость к данному виду атаки. Однако, как показали эксперименты, могут существовать такие изображения, для которых все множество особых точек может быть локализовано в одной части изображения (вне зависимости от используемого детектора), которая может быть обрезана, что, в свою очередь, будет означать удаление ЦВЗ. Справедливости ради стоит заметить, что число таких изображений существенно меньше тех, в которых данная особенность определения особых точек не наблюдается.

**Устойчивость к вносимому шуму.** Для проверки устойчивости к зашумлению в мар-

Таблица 8  
Зависимость *BER* от добавления шума в изображения

| Искажение                | <i>BER</i> | Качество извлеченного ЦВЗ (160 бит), % |
|--------------------------|------------|----------------------------------------|
| <i>Gaussian Blur</i> 1×1 | 0,01       | 90                                     |
| <i>Gaussian Blur</i> 3×3 | 0,03       | 85                                     |
| Шум 5 %                  | 0,0187     | 95                                     |
| Шум 10 %                 | 0          | 100                                    |
| Шум 15 %                 | 0,0125     | 95                                     |
| Шум 20 %                 | 0,0125     | 95                                     |
| Шум 25 %                 | 0,0125     | 95                                     |
| Шум 30 %                 | 0,0125     | 95                                     |
| Шум 35 %                 | 0,0375     | 85                                     |
| Шум 40 %                 | 0,0375     | 85                                     |



кированное изображение вносилось гауссово размытие и равномерный белый шум. Усредненные результаты теста для различных изображений представлены в табл. 8 и показывают удовлетворительные результаты.

### Заключение

В данной работе представлен метод маркирования изображений ЦВЗ, основанный на ортогональных моментах изображения Цернике, сопутствующие алгоритмы и некоторые значимые, оценочные экспериментальные результаты. В целом можно сделать вывод, что предложенный метод является применимым для маркирования ЦВЗ изображений различных форматов и характеристик, в том числе к наиболее популярному JPEG. Кроме того, метод показывает достаточно высокие показатели робастности и приемлемые вычислительные затраты на маркирование изображений ЦВЗ. Данные факты позволяют сделать выводы об эффективности предлагаемого решения для маркирования изображений невидимым ЦВЗ для решения задачи обеспечения защиты авторского права неподвижных изображений.

Однако предлагаемый в данной статье подход может быть не применим к изображениям, особые точки которого локализованы в пределах одной области на изображении таким образом, что их окрестности пересекаются. Также возможны ситуации, когда особые точки локализованы близко к краям и их окрестности выходят за пределы изображения. Данная проблема требует дальнейшего научно-технического изыскания, однако в целом не является критической в силу того, что такого рода изображений много меньше, чем тех, в которых отсутствуют такие особенности.

Предлагаемый метод маркирования изображений ЦВЗ в настоящий момент проходит опытно-промышленную эксплуатацию на специализированной цифровой платформе обмена знаниями и управления авторскими правами — IPUniversity [23], и работа над его усовершенствованием продолжается.

### Список литературы

1. Teh C. H., Chin R. T. On image analysis by the methods of moments // IEEE Trans Pattern Anal. Mach. Intell. 1988. N. 10. P. 496—513.
2. Harris C., Stephens M. A combined corner and edge detector // In Fourth Alvey Vision Conference, Manchester, UK. 1988. P. 147—151.

3. Грибунин В. Г., Оков И. Н., Туринцев И. В. Цифровая стеганография. М.: Солон-Пресс, 2002. 272 с.
4. Sharma C., Prashar D. Visible and invisible watermarking methods for quality loss of data // International Journal of Advanced Research in Computer Science and Electronics Engineering. 2012. Vol. 1, N. 3, P. 57—63.
5. Wu X., Sun W. Robust copyright protection scheme for digital images using overlapping DCT and SVD // Applied Soft Computing. 2013. Vol. 67. P. 652—664.
6. Huang Y., Lu W., Sun D. Long. Improved DCT-based detection of copy-move forgery in images // Forensic Science International. 2011. Vol. 206. P. 178—184.
7. Vaishnavi D. A., Subashini T. S. Secure and robust image watermarking system using normalization and arnold scrambling // International Journal of Network Security. Vol. 18, N. 5. P. 832—841.
8. Борисова С. Н. Использование алгоритма вейвлет-преобразования для встраивания цифровых водяных знаков в файлы изображений // XXI век: итоги прошлого и проблемы настоящего плюс. 2015. № 3 (25). С. 110—115.
9. Белобокова Ю. А., Булатников Е. В. Метод многократной маркировки цифровых фотографий для защиты от фальсификации // Известия высших учебных заведений. Проблемы полиграфии и издательского дела. 2014. № 2. С. 33—41.
10. Flusser J., Suk T., Zitova B. Moments and moment invariants in pattern recognition. John Wiley & Sons, 2009. 306 p.
11. Hu M. K. Visual pattern recognition by moment invariants // IRE Trans. Inf. Theory. 1962. Vol. 8, N. 2. P. 179—187.
12. Chen Q., Yang X., Zhao J. Robust image watermarking with Zernike moments // IEEE Canadian Conference on Electrical and Computer Engineering. 2005. P. 1340—1343.
13. Karthik P., Sathiya Priya E. Robust and High-Secured Watermarking System Using Zernike Moments // International Journal of Innovative Research in Computer and Communication Engineering. 2014. Vol. 2. P. 7074—7079.
14. Hui Zhang, Huazhong Shu, Gouenou Coatrieux, Jie Zhu, Jonathan Wu. Affine Legendre moment invariants for image watermarking robust to geometric distortions // IEEE Transactions on Image Processing, Institute of Electrical and Electronics Engineers. 2011. N. 20 (8). P. 2189—99.
15. Teague, M. R. Image analysis via the general theory of moments // Journal of the Optical Society of America. 1980. N. 70 (8). P. 920—930.
16. Ismail A. Ismail, Mohamed A. Shouman, Khalid M. Hosny, Hayam M. Abdel Salam. Invariant Image Watermarking Using Accurate Zernike Moments // Journal of Computer Science. 2010. N. 6 (1). P. 52—59.
17. Chandan Singha, Ekta Waliab. Fast and numerically stable methods for the computation of Zernike moments // Pattern Recognition. 2010. Vol. 43, P. 2497—2506.
18. Kintner E. C. A Recurrence Relation for Calculating the Zernike Polynomials // Journal of Modern Optics. 1976. N. 23 (6). P. 499—500.
19. Wang Xiang-yang, Liu Yu-nan, Li Shuo, Yang Hong-ying, Niu Pan-pan, Zhang Yan. A new robust digital watermarking using local polar harmonic Transform // Computers and Electrical Engineering. 2015. N. 46. P. 403—418.
20. Shi J., Tomasi C. Good Features to Track // IEEE Conference on Computer Vision and Pattern Recognition. 1994. P. 593—600.
21. Chandan Singh, Sukhjeet K. Ranade. An Effective Image Watermarking System for High Embedding Capacity // IJCA Proceedings on International Conference on Recent Advances and Future Trends in Information Technology. 2012. P. 22—28.
22. Сосновский М. С., Шниперов А. Н. Программный инструмент для маркирования изображений цифровым водяным знаком "ZWAT" // св-во о гос. регистрации программы для ЭВМ № 2017610758. Правообладатель ФГАОУ ВО "Сибирский федеральный университет (СФУ)", 2017.
23. Специализированная цифровая платформа обмена знаниями и управления авторскими правами IPUniversity. URL: <https://ipuniversity.ru> (дата обращения: 31.01.19).

A. N. Shniperov, Assistant Professor, e-mail: ashniperov@sfu-kras.ru,  
Siberian Federal University,

M. S. Sosnovskiy, Software Development Engineer, e-mail: maximsosnovskiy@gmail.com,

P. M. Shipulin, Software Development Engineer, e-mail: pshipulin@gmail.com,  
Academician M. F. Reshetnev Information Satellite Systems

## The Robust Image Digital Watermark Labeling Method Based on Orthogonal Zernike Moments

Nowadays visible watermarks are usually used for the copyright protection of images. But this method does not give the best result in many cases. The best practices lie in the digital steganography field. In this article authors suggest a digital watermark labeling method for full-color images based on orthogonal Zernike moments calculated for certain image points (and their neighborhoods). Using only certain points makes the method applicable for any image sizes with good robustness features and low computation time (orthogonal Zernike moments calculation for big image is quite a difficult computation task). To find certain points the Shi-Tomasi corner detector algorithm is used. These points are invariant under affine transformation. The suggested method uses only blue channel of the RGB-model, because distortions in this channel are less visible for the human eye. For performance evaluation of the method some popular metrics were used: BER (Bit Error Rate), PSNR (Peak Signal-To-Noise Ratio), MSE (Mean Square Error), SSIM (Structural SIMilarity). A series of experiments illustrates high values of these metrics not only in case of simple digital watermark embedding but in case of the most popular active attacks on the labeling method: JPEG compression, image reflexing, scaling, rotating, cropping, noise adding. The group of tests proves the labeling method immunity to the majority of popular active attacks. So the suggested method can be regarded as an effective method for labeling images by invisible watermarking for copyright protection purposes.

**Keywords:** digital watermarks, steganography, image labeling, information security, copyright protection, image moments, Zernike moments

DOI: 10.17587/it.25.405-413

### References

1. Teh C. H., Chin R. T. *IEEE Trans Pattern Anal. Mach. Intell.*, 1988, no. 10, pp. 496–513.
2. Harris C., Stephens M. *In Fourth Alvey Vision Conference*, Manchester, UKm 1988, pp. 147–151.
3. Gribunin V. G., Okov I. N., Turincev I. V. *Digital steganography*, Moscow, Solon-Press, 2002, 272 p. (in Russian).
4. Sharma C., Prashar D. *International Journal of Advanced Research in Computer Science and Electronics Engineering*, 2012, vol. 1, no. 3, pp. 57–63.
5. Wu X., Sun W. *Applied Soft Computing*, 2013, vol. 67, pp. 652–664.
6. Huang Y., Lu W., Sun D. Long. *Forensic Science International*, 2011, vol. 206, pp. 178–184.
7. Vaishnavi D. A., Subashini T. S. *International Journal of Network Security*, vol. 18, no. 5, pp. 832–841.
8. Borisova S. H. *XXI vek: itogi proshlogo i problemi nastoyashego plus*, 2015, no. 3 (25), pp. 110–115 (in Russian).
9. Belobokova Y. A., Bulatnikov E. V. *Izvestiya vishih uchebnykh zavedeniy. Problemi Poligrafii i Izdatelskogo Dela*, 2014, no. 2, pp. 33–41 (in Russian).
10. Flusser J., Suk T., Zitova B. *Moments and moment invariants in pattern recognition*, John Wiley & Sons, 2009, 306 p.
11. Hu M. K. *IRE Trans. Inf. Theory*, 1962, vol. 8, no. 2, pp. 179–187.
12. Chen Q., Yang X., Zhao J. *IEEE Canadian Conference on Electrical and Computer Engineering*, 2005, pp. 1340–1343.
13. Karthik P., Sathya Priya E. *International Journal of Innovative Research in Computer and Communication Engineering*, 2014, vol. 2, pp. 7074–7079.
14. Hui Zhang, Huazhong Shu, Gouenou Coatrieux, Jie Zhu, Jonathan Wu. *IEEE Transactions on Image Processing, Institute of Electrical and Electronics Engineers*, 2011, no. 20 (8), pp. 2189–99.
15. Teague M. R. *Journal of the Optical Society of America*, 1980, no. 70 (8), pp. 920–930.
16. Ismail A. Ismail, Mohamed A. Shouman, Khalid M. Hosny and Hayam M. Abdel Salam. *Journal of Computer Science*, 2010, no. 6 (1), pp. 52–59.
17. Chandan Singha, Ekta Waliab. *Pattern Recognition*, 2010, vol. 43, pp. 2497–2506.
18. Kintner E. C. *A Journal of Modern Optics*, 1976, no. 23 (6), pp. 499–500.
19. Wang Xiang-yang, Liu Yu-nan, Li Shuo, Yang Hong-ying, Niu Pan-pan, Zhang Yan. *Computers and Electrical Engineering*, 2015, no. 46, pp. 403–418.
20. Shi J., Tomasi C. *IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.
21. Chandan Singh, Sukhjeet K. Ranade. *IJCA Proceedings on International Conference on Recent Advances and Future Trends in Information Technology*, 2012, pp. 22–28.
22. Sosnovskiy M. S., Shniperov A. N. Software for Image labeling with digital watermark, svidetelstvo o gos. Registracii programmi dly EVM no 2017610758. Pravoobladatel FGAOU VO "Sibirskiy federalniy universitet (SFU)", 2017 (in Russian).
23. Special digital platform for knowledge exchanging and copyright management, available at: <https://ipuniversity.ru> (date of access: 16.01.19) (in Russian).

**Н. Н. Нагорнов**, аспирант, e-mail: sparta1392@mail.ru,  
**П. А. Ляхов**, канд. физ.-мат. наук, доц., e-mail: ljahov@mail.ru,  
**Н. И. Червяков**, д-р техн. наук, проф., e-mail: k-fmf-primath@stavsu.ru,  
Северо-Кавказский федеральный университет, г. Ставрополь

## **Исследование шума квантования фильтров дискретного вейвлет-преобразования для обработки трехмерных изображений в медицине**

*Решена задача анализа влияния шума квантования коэффициентов фильтров дискретного вейвлет-преобразования (ДВП) на результат обработки трехмерных медицинских изображений. Предложен метод квантования коэффициентов. Разработан метод расчета погрешности ДВП-изображений. ДВП выполняется с использованием чисел в формате с фиксированной точкой.*

**Ключевые слова:** шум квантования, вейвлет-преобразование, обработка трехмерных медицинских изображений

### **Введение**

В настоящее время в медицине активно используется визуализация двумерных и трехмерных данных, получаемых с помощью множества различных методов, в частности, магнитно-резонансных [1–3], рентгенологических [4, 5], радионуклидных [5], оптических [4, 6] и ультразвуковых [1, 7]. Необходимость очистки от шума двумерных и трехмерных медицинских изображений является постоянной проблемой в современных системах визуализации [1, 4, 5, 7]. С развитием цифровых устройств и технологий сканирования медицинские системы визуализации дают возможность получения все более точных изображений с улучшенным качеством, с использованием более высоких пространственных разрешений и битовой глубины. Такие улучшения увеличивают объем информации, которая должна обрабатываться, передаваться и храниться. Это особенно актуально при использовании технологий трехмерного сканирования [3]. Например, четыре набора медицинских изображений позитронно-эмиссионной томографии одного пациента могут потребовать более 4 ГБ дискового пространства для хранения [8]. Результаты оптической когерентной томографии могут занимать более 40 ГБ памяти [6]. С учетом нынешнего уровня развития технологий хранения данных, при котором емкость жестких дисков составляет в среднем 1...2 ТБ, сжатие трехмерных изображений также является актуальной проблемой в современных системах визуализации.

Для решения задач очистки от шума и сжатия двумерных и трехмерных медицинских

изображений на практике используются различные преобразования, наиболее распространенными из которых являются дискретное преобразование Фурье (ДПФ) [2, 3, 6, 8] и дискретное вейвлет-преобразование (ДВП) [1, 3, 4, 6–8]. В то время как ДПФ выделяет только частотную информацию, ДВП позволяет получить частотно-временную информацию о сигнале, которая содержит его локальные и частотные особенности. ДВП двумерных и трехмерных изображений осуществляется путем свертки с парой вейвлет-фильтров — низкочастотного и высокочастотного, выделяющих основную и детальную информацию соответственно. Путем манипуляций над детальной информацией осуществляются очистка от шума и сжатие в современных алгоритмах SPIHT [9], EZW [10] и др. Операция свертки имеет высокую вычислительную сложность. Одним из путей улучшения ее характеристик является аппаратная реализация на современных микроэлектронных устройствах (FPGA, ASIC и др.) [11–13], работающих с числами в формате с фиксированной точкой. При преобразовании коэффициентов вейвлет-фильтров в этот формат возникает шум квантования, из-за которого свертка выполняется с погрешностью. Возникает вопрос о точности представления коэффициентов вейвлет-фильтров в памяти устройства, эффективной с точки зрения затрачиваемых ресурсов и в то же время необходимой для достижения требуемого качества обработки изображений. В изученных материалах по аппаратной реализации ДВП медицинских изображений на FPGA и ASIC [13–22] отсутствуют упоминания о вы-

бранных разрядностях коэффициентов вейвлет-фильтров. В работе [23] авторы квантуют коэффициенты 16 битами, однако отсутствует какое-либо обоснование сделанного выбора. В работе [24] решена задача анализа шума квантования, возникающего в результате округления коэффициентов вейвлет-фильтров, при обработке обычных двумерных 8-битных полутоновых и 24-битных полноцветных изображений.

Целью данной работы является выявление зависимости качества ДВП трехмерных медицинских изображений различной глубины цвета от шума, возникающего при квантовании коэффициентов вейвлет-фильтров с компактным носителем. Особое внимание уделено определению минимальной разрядности коэффициентов, при которой этот шум не оказывает существенного влияния на результат обработки изображений либо не влияет на него вовсе, в терминах пикового отношения сигнала к шуму (*PSNR*) [25].

### ДВП трехмерных изображений

ДВП — преобразование сигнала, представляющее собой свертку входных данных с набором дискретных вейвлет-фильтров, переводящим их из временного представления в частотно-временное. Будем считать, что вейвлет-фильтры  $F$  состоят из коэффициентов  $f_{F,i}$ , где  $i = 0, \dots, k-1$ , — номер коэффициента, а  $k$  — число коэффициентов. Коэффициенты низкочастотных и высокочастотных вейвлет-фильтров анализа ( $LD$ ,  $HD$ ) и синтеза ( $LR$ ,  $HR$ ) связаны соотношениями [11]

$$f_{HD,i} = (-1)^{i+1} f_{LD,k-1-i},$$

$$f_{LR,i} = f_{LD,k-1-i}, \quad f_{HR,i} = (-1)^i f_{LD,i}.$$

Далее рассматриваются только вейвлеты с компактным носителем [26], наиболее распространенными из которых являются: вейвлеты Добеши  $dbk/2$ , где  $db1$  — вейвлет Хаара; симлеты  $symk/2$  и койфлеты  $coif k/6$ .

Представим трехмерное цифровое изображение  $I$  из  $X$  строк,  $Y$  столбцов и  $Z$  кадров как функцию  $I(x, y, z)$ , где  $0 \leq x \leq X-1$ ,  $0 \leq y \leq Y-1$  и  $0 \leq z \leq Z-1$  — пространственные координаты  $I$ . Значения вокселей (аналогов двумерных пикселей для трехмерного пространства) представляются в виде  $I(x, y, z)$  для полутоновых изображений и  $I(x, y, z, c)$  для полноцветных, где  $c$  — номер цвета (например, для изображений в формате RGB  $c = 1, 2, 3$  — красный, зеленый и синий соответственно).

Свертка вейвлет-фильтра  $F$  с трехмерным изображением выполняется по формулам

$$I'(x, y, z) = \sum_{i=0}^{k-1} I(x+i, y, z) f_{F,k-1-i};$$

$$I''(x, y, z) = \sum_{i=0}^{k-1} I(x, y+i, z) f_{F,k-1-i};$$

$$I'''(x, y, z) = \sum_{i=0}^{k-1} I(x, y, z+i) f_{F,k-1-i},$$

где  $I'$ ,  $I''$  и  $I'''$  — результаты свертки по строкам, столбцам и кадрам соответственно. ДВП трехмерного изображения осуществляется путем последовательной свертки с вейвлет-фильтрами согласно схеме, представленной на рис. 1, и включает перечисленные ниже этапы.

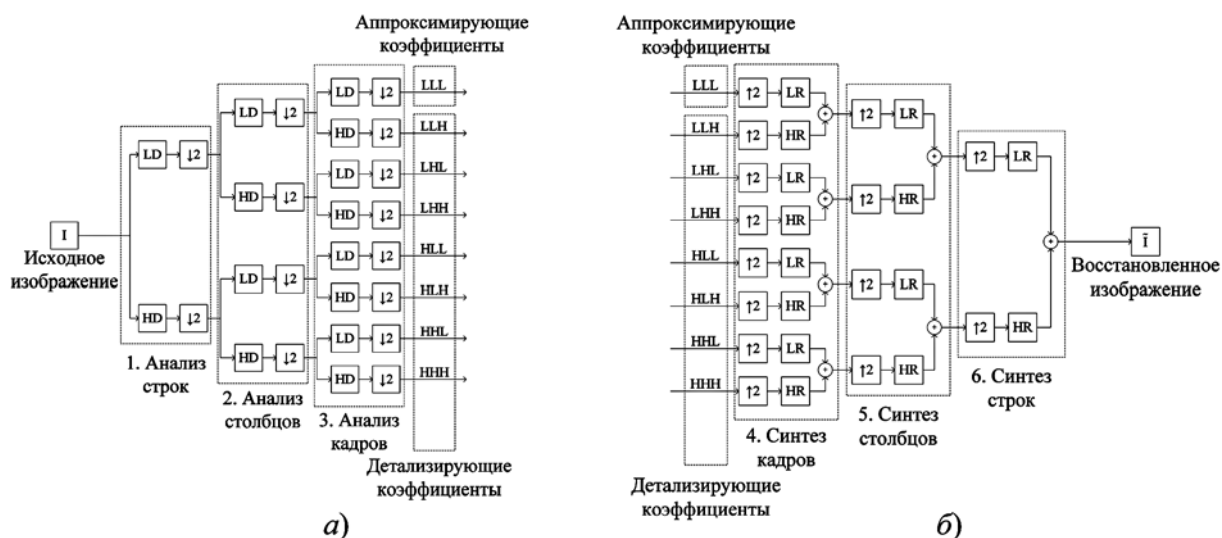


Рис. 1. Схема ДВП трехмерного изображения:

а — анализ; б — синтез

1. Анализ строк проводится путем разложения изображения  $I$  по строкам низкочастотным  $LD$  и высокочастотным  $HD$  вейвлет-фильтрами и понижающей в два раза дискретизации, обозначаемой символом  $\downarrow 2$ .

2. Анализ столбцов выполняется путем разложения коэффициентов, полученных на 1 этапе, по столбцам аналогично анализу строк.

3. Анализ кадров проводится путем разложения коэффициентов, полученных на втором этапе, по кадрам аналогично анализу строк.

В результате анализа происходит разложение исходного трехмерного изображения  $I$  на восемь наборов коэффициентов  $LLL$ ,  $LLH$ ,  $LHL$ ,  $LHH$ ,  $HLL$ ,  $HLH$ ,  $HHL$ ,  $HHH$ , которые можно разделить на две группы: аппроксимирующие ( $LLL$ ), которые соответствуют низкочастотной части сигнала и содержат основную информацию об изображении; детализирующие ( $LLH$ ,  $LHL$ ,  $LHH$ ,  $HLL$ ,  $HLH$ ,  $HHL$ ,  $HHH$ ), которые соответствуют высокочастотной части сигнала и содержат детальную информацию об изображении. Посредством манипуляций над коэффициентами разложения изображения осуществляются очистка от шума и сжатие.

4. Синтез кадров выполняется путем повышения в два раза дискретизации, обозначаемой символом  $\uparrow 2$ , коэффициентов разложения изображения по кадрам, восстановления низкочастотным  $LR$  и высокочастотным  $HR$  вейвлет-фильтрами и суммирования соответствующих результатов.

5. Синтез столбцов проводится путем восстановления коэффициентов, полученных на четвертом этапе, по столбцам аналогично синтезу кадров.

6. Синтез строк выполняется путем восстановления коэффициентов, полученных на пятом этапе, по строкам аналогично синтезу кадров.

В результате синтеза восьми наборов коэффициентов  $LLL$ ,  $LLH$ ,  $LHL$ ,  $LHH$ ,  $HLL$ ,  $HLH$ ,  $HHL$ ,  $HHH$  происходит восстановление изображения  $\tilde{I}$ . Теоретически исходное изображение должно быть полностью восстановлено, так как схема на рис. 1 обладает свойством точного восстановления сигнала [27]. Однако на практике из-за шума квантования, обусловленного цифровым форматом представления данных, в процессе ДВП появляется погрешность, искажающая как наборы коэффициентов  $LLL$ ,  $LLH$ ,  $LHL$ ,  $LHH$ ,  $HLL$ ,  $HLH$ ,  $HHL$ ,  $HHH$ , так и восстановленное изображение  $\tilde{I}$ . В зависимости от значения данной погрешности результат обработки изображений может иметь качество, неприемлемое для выполнения поставленной задачи.

Возникает вопрос о минимальной разрядности представления коэффициентов  $f_{F,i}$  вейвлет-фильтров  $F$ , эффективного с точки зрения аппаратной реализации на современных устройствах и необходимого для достижения обработанным изображением высокого качества. На современных устройствах скорость выполнения операций с числами в формате с фиксированной точкой существенно выше, чем в формате с плавающей точкой, что может быть использовано при разработке устройств обработки трехмерных изображений. Поэтому мы предлагаем квантовать коэффициенты вейвлет-фильтров следующим образом: умножать на  $2^n$  и округлять к большему:

$$f_{F,i}^* = \lceil 2^n f_{F,i} \rceil. \quad (2)$$

Разрядность  $r$  представления квантованных коэффициентов  $f_{F,i}^*$  в этом случае может быть вычислена по формуле  $r = n + 1$ . Значения вокселей обработанного трехмерного изображения  $I^*$ , полученного в результате ДВП, преобразуются делением на  $2^{6n}$  (по  $2^n$  за каждую свертку с вейвлет-фильтрами) и округлением к меньшему:

$$\tilde{I} = \lfloor 2^{-6n} I^* \rfloor. \quad (3)$$

Операции округления к большему и к меньшему выполняются путем отбрасывания дробной части числа с добавлением единицы в случае округления к большему. При округлении в разные стороны погрешности этих округлений будут иметь разные знаки и частично компенсировать друг друга. Применение операций округления именно в таком порядке требует меньших ресурсов при аппаратной реализации, чем операции округления к ближайшему целому. Это объясняется тем, что коэффициенты вейвлет-фильтров известны априори и их квантование с округлением вверх может быть выполнено предварительно. Таким образом, в аппаратной части коэффициенты  $f_{F,i}^*$  будут использоваться в виде констант. Выполнение свертки осуществляется с использованием арифметико-логических устройств, и округление ее результатов вниз осуществляется простым отбрасыванием дробной части и не требует дополнительных аппаратных и временных затрат.

В качестве критерия оценки качества обработки изображений используется числовая характеристика  $PSNR$ , или пиковое отноше-

ние сигнал-шум между двумя изображениями (оригиналом  $I$  и полученным изображением  $\tilde{I}$ ), вычисляемая следующим образом [28]:

$$PSNR = 10\lg(M^2/MSE),$$

где  $M$  — максимальное значение яркости вокселей изображений;  $MSE$  — среднеквадратическое отклонение значений яркости, определяемое для полутоновых ( $MSE_{grayscale}$ ) [29] и полноцветных ( $MSE_{color}$ ) [30] трехмерных изображений по формулам

$$\begin{aligned} MSE_{grayscale} &= \\ &= \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} \sum_{z=0}^{Z-1} (I(x, y, z) - \tilde{I}(x, y, z))^2 / (XYZ), \\ MSE_{color} &= \\ &= \frac{1}{C} \sum_{c=1}^C \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} \sum_{z=0}^{Z-1} (I(x, y, z, c) - \tilde{I}(x, y, z, c))^2 / (XYZ). \end{aligned}$$

Характеристика  $PSNR$  имеет логарифмическую природу и измеряется в децибелах (дБ). Чем лучше качество обработки изображения, тем больше значение  $PSNR$ . Для одинаковых (тождественно равных) изображений  $PSNR = \infty$ . Качество обработки изображения считается высоким, если  $PSNR \geq 40$  дБ, так как значение 40 дБ описывает разницу между двумя изображениями, практически незаметную для человека [25]. В следующем разделе показано, как разрядность коэффициентов вейвлет-фильтров влияет на качество обработки трехмерных медицинских изображений.

### Теоретический анализ максимальной погрешности ДВП трехмерных медицинских изображений

Погрешность при ДВП изображений возникает в результате преобразования коэффициентов фильтров ДВП (шум квантования) по формуле (2). Затем она возрастает при свертках, повышающих дискретизациях и сложениях результатов свертки. Округление после масштабирования значений вокселей восстановленного изображения также оказывает влияние. Отметим следующие важные факты:

1. Согласно формуле (1) рассматриваемые анализирующие и синтезирующие вейвлет-фильтры состоят из одинаковых коэффициентов, следовательно, и максимальные абсолютные погрешности вычислений также будут равны. По этой причине в рамках проводимых

расчетов вейвлет-фильтры подразделяются только на низкочастотные  $L$  и высокочастотные  $H$ .

2. Суммы коэффициентов низкочастотных  $L$  и высокочастотных  $H$  вейвлет-фильтров равны

$$\sum_{i=0}^{k-1} f_{L,i} = \sqrt{2} \text{ и } \sum_{i=0}^{k-1} f_{H,i} = 0 \text{ соответственно [11].}$$

3. Абсолютная погрешность ДВП изображения максимальна, когда максимальны все значения яркости вокселей изображения, так как в этом случае максимальна погрешность свертки изображения с низкочастотным фильтром, многократно превосходящая погрешность свертки изображения с высокочастотным фильтром в силу предыдущего факта.

Введем обозначения.

$E_{j,F}$  — предельная абсолютная погрешность (ПАП) вычисления значения коэффициента изображения на  $j$ -м этапе, возникшая в результате свертки с последовательностью вейвлет-фильтров  $F$ ;

$S_F$  — точное значение суммы коэффициентов вейвлет-фильтра  $F$ ;

$T_{j,F}$  — точное значение вычислений на  $j$ -м этапе, после свертки с последовательностью вейвлет-фильтров  $F$ .

В результате каждой повышающей дискретизации  $\uparrow 2$  погрешности  $a$  для всех наборов коэффициентов  $LLL, LLH, LHL, LHH, HLL, HLH, HHL, HHH$  разделяются на две группы. В ходе восстановления изображения повышающая дискретизация применяется трижды. В результате мы получаем восемь групп погрешностей  $a_\varepsilon$ , где  $\varepsilon = 1, 2, 3, 4, 5, 6, 7, 8$ . Таким образом, к введенным обозначениям необходимо добавить дополнительный индекс  $\varepsilon$ , обозначающий вычисления по пространственным характеристикам коэффициентов. Далее представлен метод проведения расчетов для оценки максимальной погрешности вычислений ДВП трехмерных изображений:

**Этап 1. Квантование коэффициентов вейвлет-фильтров:**

$$S_L = \sum_{i=1}^k 2^n f_{L,i} = 2^n \sum_{i=1}^k f_{L,i} = 2^n \sqrt{2} = 2^{n+\frac{1}{2}};$$

$$S_H = \sum_{i=1}^k 2^n f_{H,i} = 2^n \sum_{i=1}^k f_{H,i} = 2^n \cdot 0 = 0;$$

$$S_{L,1} = \sum_{i=1}^{\frac{k}{2}} 2^n f_{L,2(i-1)}; \quad S_{L,2} = \sum_{i=1}^{\frac{k}{2}} 2^n f_{L,2i-1};$$

$$S_{H,1} = \sum_{i=1}^{\frac{k}{2}} 2^n f_{H,2(i-1)}; \quad S_{H,2} = \sum_{i=1}^{\frac{k}{2}} 2^n f_{H,2i-1};$$



$$\begin{aligned}
E_{1,L} &= \sum_{i=1}^k \left( \left\lceil 2^n f_{L,i} \right\rceil - 2^n f_{L,i} \right); \\
E_{1,H} &= \sum_{i=1}^k \left( \left\lceil 2^n f_{H,i} \right\rceil - 2^n f_{H,i} \right); \\
E_{1,L,1} &= \sum_{i=1}^{\frac{k}{2}} \left( \left\lceil 2^n f_{L,2(i-1)} \right\rceil - 2^n f_{L,2(i-1)} \right); \\
E_{1,L,2} &= \sum_{i=1}^{\frac{k}{2}} \left( \left\lceil 2^n f_{L,2i-1} \right\rceil - 2^n f_{L,2i-1} \right); \\
E_{1,H,1} &= \sum_{i=1}^{\frac{k}{2}} \left( \left\lceil 2^n f_{H,2(i-1)} \right\rceil - 2^n f_{H,2(i-1)} \right); \\
E_{1,H,2} &= \sum_{i=1}^{\frac{k}{2}} \left( \left\lceil 2^n f_{H,2i-1} \right\rceil - 2^n f_{H,2i-1} \right).
\end{aligned}$$

**Этап 2. Разложение по строкам:**

$$T_{2,L} = S_L M; \quad E_{2,L} = E_{1,L} M; \quad E_{2,H} = E_{1,H} M.$$

Все  $T_{j, F}$  представляющие собой результат свертки с фильтром  $H$ , равны нулю, так как  $T_{j, F}$  одинаковы для всех вокселей и  $\sum_{i=0}^{k-1} f_{H,i} = 0$  [11].

**Этап 3. Разложение по столбцам:**

$$\begin{aligned}
T_{3,LL} &= T_{2,L} S_L; \\
E_{3,LL} &= (T_{2,L} + E_{2,L})(S_L + E_{1,L}) - T_{3,LL}; \\
E_{3,LH} &= (T_{2,L} + E_{2,L}) E_{1,H}; \\
E_{3,HL} &= E_{2,H} (S_L + E_{1,L}); \quad E_{3,HH} = E_{2,H} E_{1,H}.
\end{aligned}$$

**Этап 4. Разложение по кадрам:**

$$\begin{aligned}
T_{4,LLL} &= T_{3,LL} S_L; \\
E_{4,LLL} &= (T_{3,LL} + E_{3,LL})(S_L + E_{1,L}) - T_{4,LLL}; \\
E_{4,LLH} &= (T_{3,LL} + E_{3,LL}) E_{1,H}; \\
E_{4,LHL} &= E_{3,LH} (S_L + E_{1,L}); \\
E_{4,LHH} &= E_{3,LH} E_{1,H}; \\
E_{4,HLL} &= E_{3,HL} (S_L + E_{1,L}); \quad E_{4,HLLH} = E_{3,HL} E_{1,H}; \\
E_{4,HHL} &= E_{3,HH} (S_L + E_{1,L}); \quad E_{4,HHH} = E_{3,HH} E_{1,H}.
\end{aligned}$$

**Этап 5. Восстановление по кадрам ( $\epsilon = 1, 2$ ):**

$$\begin{aligned}
T_{5,LLLL,\epsilon} &= T_{4,LLL} S_{L,\epsilon}; \\
E_{5,LLLL,\epsilon} &= (T_{4,LLL} + E_{4,LLL})(S_{L,\epsilon} + E_{1,L,\epsilon}) - T_{5,LLLL,\epsilon}; \\
E_{5,LLHH,\epsilon} &= E_{4,LLH} (S_{H,\epsilon} + E_{1,H,\epsilon}); \\
E_{5,LHLL,\epsilon} &= E_{4,LHL} (S_{L,\epsilon} + E_{1,L,\epsilon});
\end{aligned}$$

$$\begin{aligned}
E_{5,LHHH,\epsilon} &= E_{4,LHH} (S_{H,\epsilon} + E_{1,H,\epsilon}); \\
E_{5,HLLL,\epsilon} &= E_{4,HLL} (S_{L,\epsilon} + E_{1,L,\epsilon}); \\
E_{5,HLHH,\epsilon} &= E_{4,HLH} (S_{H,\epsilon} + E_{1,H,\epsilon}); \\
E_{5,HHLL,\epsilon} &= E_{4,HHL} (S_{L,\epsilon} + E_{1,L,\epsilon}); \\
E_{5,HHHH,\epsilon} &= E_{4,HHH} (S_{H,\epsilon} + E_{1,H,\epsilon}).
\end{aligned}$$

**Этап 6. Суммирование по кадрам ( $\epsilon = 1, 2$ ):**

$$\begin{aligned}
E_{6,LL,\epsilon} &= E_{5,LLLL,\epsilon} + E_{5,LLHH,\epsilon}; \\
E_{6,LH,\epsilon} &= E_{5,LHLL,\epsilon} + E_{5,LHHH,\epsilon}; \\
E_{6,HL,\epsilon} &= E_{5,HLLL,\epsilon} + E_{5,HLHH,\epsilon}; \\
E_{6,HH,\epsilon} &= E_{5,HHLL,\epsilon} + E_{5,HHHH,\epsilon}.
\end{aligned}$$

**Этап 7. Восстановление по столбцам:**

$$\begin{aligned}
T_{7,LL,1} &= T_{5,LLLL,1} S_{L,1}, \quad T_{7,LL,2} = T_{5,LLLL,2} S_{L,1}; \\
T_{7,LL,3} &= T_{5,LLLL,1} S_{L,2}, \quad T_{7,LL,4} = T_{5,LLLL,2} S_{L,2}; \\
E_{7,LL,1} &= (T_{5,LLLL,1} + E_{6,LL,1})(S_{L,1} + E_{1,L,1}) - T_{7,LL,1}; \\
E_{7,LL,2} &= (T_{5,LLLL,2} + E_{6,LL,2})(S_{L,1} + E_{1,L,1}) - T_{7,LL,2}; \\
E_{7,LL,3} &= (T_{5,LLLL,1} + E_{6,LL,1})(S_{L,2} + E_{1,L,2}) - T_{7,LL,3}; \\
E_{7,LL,4} &= (T_{5,LLLL,2} + E_{6,LL,2})(S_{L,2} + E_{1,L,2}) - T_{7,LL,4}; \\
E_{7,LH,1} &= E_{6,LH,1} (S_{H,1} + E_{1,H,1}); \\
E_{7,LH,2} &= E_{6,LH,2} (S_{H,1} + E_{1,H,1}); \\
E_{7,LH,3} &= E_{6,LH,1} (S_{H,2} + E_{1,H,2}); \\
E_{7,LH,4} &= E_{6,LH,2} (S_{H,2} + E_{1,H,2}); \\
E_{7,HL,1} &= E_{6,HL,1} (S_{L,1} + E_{1,L,1}); \\
E_{7,HL,2} &= E_{6,HL,2} (S_{L,1} + E_{1,L,1}); \\
E_{7,HL,3} &= E_{6,HL,1} (S_{L,2} + E_{1,L,2}); \\
E_{7,HL,4} &= E_{6,HL,2} (S_{L,2} + E_{1,L,2}); \\
E_{7,HH,1} &= E_{6,HH,1} (S_{H,1} + E_{1,H,1}); \\
E_{7,HH,2} &= E_{6,HH,2} (S_{H,1} + E_{1,H,1}); \\
E_{7,HH,3} &= E_{6,HH,1} (S_{H,2} + E_{1,H,2}); \\
E_{7,HH,4} &= E_{6,HH,2} (S_{H,2} + E_{1,H,2}).
\end{aligned}$$

**Этап 8. Суммирование по столбцам ( $\epsilon = 1, 2, 3, 4$ ):**

$$\begin{aligned}
E_{8,L,\epsilon} &= E_{7,LL,\epsilon} + E_{7,LH,\epsilon}; \\
E_{8,H,\epsilon} &= E_{7,HL,\epsilon} + E_{7,HH,\epsilon}.
\end{aligned}$$

### Этап 9. Восстановление по строкам:

$$\begin{aligned}
 T_{9,1} &= T_{7,LL,1}S_{L,1}; \quad T_{9,2} = T_{7,LL,2}S_{L,1}; \\
 T_{9,3} &= T_{7,LL,3}S_{L,1}; \quad T_{9,4} = T_{7,LL,4}S_{L,1}; \\
 T_{9,5} &= T_{7,LL,1}S_{L,2}; \quad T_{9,6} = T_{7,LL,2}S_{L,2}; \\
 T_{9,7} &= T_{7,LL,3}S_{L,2}; \quad T_{9,8} = T_{7,LL,4}S_{L,2}; \\
 E_{9,L,1} &= (T_{7,LL,1} + E_{8,L,1})(S_{L,1} + E_{1,L,1}) - T_{9,1}; \\
 E_{9,L,2} &= (T_{7,LL,2} + E_{8,L,2})(S_{L,1} + E_{1,L,1}) - T_{9,2}; \\
 E_{9,L,3} &= (T_{7,LL,3} + E_{8,L,3})(S_{L,1} + E_{1,L,1}) - T_{9,3}; \\
 E_{9,L,4} &= (T_{7,LL,4} + E_{8,L,4})(S_{L,1} + E_{1,L,1}) - T_{9,4}; \\
 E_{9,L,5} &= (T_{7,LL,1} + E_{8,L,1})(S_{L,2} + E_{1,L,2}) - T_{9,5}; \\
 E_{9,L,6} &= (T_{7,LL,2} + E_{8,L,2})(S_{L,2} + E_{1,L,2}) - T_{9,6}; \\
 E_{9,L,7} &= (T_{7,LL,3} + E_{8,L,3})(S_{L,2} + E_{1,L,2}) - T_{9,7}; \\
 E_{9,L,8} &= (T_{7,LL,4} + E_{8,L,4})(S_{L,2} + E_{1,L,2}) - T_{9,8}; \\
 E_{9,H,1} &= E_{8,H,1}(S_{H,1} + E_{1,H,1}); \\
 E_{9,H,2} &= E_{8,H,2}(S_{H,1} + E_{1,H,1}); \\
 E_{9,H,3} &= E_{8,H,3}(S_{H,1} + E_{1,H,1}); \\
 E_{9,H,4} &= E_{8,H,4}(S_{H,1} + E_{1,H,1}); \\
 E_{9,H,5} &= E_{8,H,1}(S_{H,2} + E_{1,H,2}); \\
 E_{9,H,6} &= E_{8,H,2}(S_{H,2} + E_{1,H,2}); \\
 E_{9,H,7} &= E_{8,H,3}(S_{H,2} + E_{1,H,2}); \\
 E_{9,H,8} &= E_{8,H,4}(S_{H,2} + E_{1,H,2}).
 \end{aligned}$$

### Этап 10. Суммирование по строкам

( $\epsilon = 1, 2, 3, 4, 5, 6, 7, 8$ ):

$$E_{10,l} = E_{9,L,l} + E_{9,H,l}$$

### Этап 11. Преобразование значений вокселей восстановленного изображения

( $\epsilon = 1, 2, 3, 4, 5, 6, 7, 8$ ):

$$E_{11,\epsilon} = \lfloor 2^{-6n} E_{10,\epsilon} \rfloor.$$

Полученные значения  $E_{11,\epsilon}$  ( $\epsilon = 1, 2, 3, 4, 5, 6, 7, 8$ ) представляют собой результирующую погрешность метода и позволяют вычислить характеристику оценки качества обработки изображения

$$PSNR = 10 \lg \left( 8M^2 / \sum_{\epsilon=1}^8 E_{11,\epsilon}^2 \right), \quad (4)$$

где  $MSE_{grayscale} = MSE_{color} = 1/8 \sum_{\epsilon=1}^8 E_{11,\epsilon}^2$ .

Формула (4) позволяет определить минимальное качество трехмерного изображения, полученного в результате ДВП, в зависимости от максимального значения яркости  $M$  и выбранной разрядности  $r = n + 1$  коэффициентов  $f_{F,i}$  вейвлет-фильтров  $F$ .

В табл. 1—3 представлены минимальные значения разрядностей  $r$ , при которых, согласно формуле (4) результат ДВП изображений с 8-, 12- и 16-битными цветовыми каналами достигает высокого ( $PSNR \geq 40$  дБ) и максимального качества ( $PSNR = \infty$ ) при обработке вейвлетами Добеши  $db\ k/2$ , симлетами  $sym\ k/2$  и койфлетами  $coif\ k/6$  с числом коэффициентов  $k$ .

На основе данных из табл. 1—3 мы можем сделать следующие выводы.

1. Минимальную разрядность  $r$  коэффициентов вейвлет-фильтров с компактным носителем, при которой результат ДВП трехмерного изображения гарантированно не содержит видимых искажений ( $PSNR \geq 40$ ), можно определить по формуле

$$r = 11 + \lfloor \sqrt{k/2 - 1} \rfloor, \quad (5)$$

где  $k$  — число коэффициентов вейвлет-фильтров.

2. Минимальную разрядность  $r$  коэффициентов вейвлет-фильтров с компактным носителем, при которой результат ДВП трехмерного изображения гарантированно не содержит

Таблица 1

Минимальные значения  $r$ , при которых результат ДВП изображения вейвлетами Добеши достигнет определенного качества

| Канал, бит | $PSNR$ , дБ | $db1$ | $db2$ | $db3$ | $db4$ | $db5$ | $db6$ | $db7$ | $db8$ | $db9$ | $db10$ |
|------------|-------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|
| 8          | 40          | 11    | 11    | 12    | 12    | 12    | 13    | 13    | 13    | 13    | 14     |
|            | $\infty$    | 13    | 13    | 14    | 14    | 14    | 15    | 15    | 15    | 15    | 15     |
| 12         | 40          | 11    | 11    | 12    | 12    | 12    | 13    | 13    | 13    | 13    | 14     |
|            | $\infty$    | 16    | 17    | 17    | 18    | 19    | 19    | 19    | 19    | 19    | 20     |
| 16         | 40          | 11    | 11    | 12    | 12    | 13    | 13    | 13    | 13    | 13    | 14     |
|            | $\infty$    | 20    | 21    | 22    | 22    | 23    | 22    | 23    | 23    | 23    | 24     |

Таблица 2

Минимальные значения  $r$ , при которых результат ДВП изображения симлетами достигнет определенного качества

| Канал, бит | PSNR, дБ | <i>sym1</i> | <i>sym2</i> | <i>sym3</i> | <i>sym4</i> | <i>sym5</i> | <i>sym6</i> | <i>sym8</i> | <i>sym10</i> |
|------------|----------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|--------------|
| 8          | 40       | 11          | 11          | 12          | 12          | 12          | 13          | 13          | 13           |
|            | $\infty$ | 13          | 13          | 14          | 14          | 15          | 15          | 15          | 15           |
| 12         | 40       | 11          | 11          | 12          | 12          | 13          | 13          | 13          | 14           |
|            | $\infty$ | 16          | 17          | 17          | 18          | 19          | 19          | 19          | 19           |
| 16         | 40       | 11          | 11          | 12          | 12          | 13          | 13          | 13          | 14           |
|            | $\infty$ | 20          | 21          | 22          | 22          | 23          | 23          | 23          | 24           |

Таблица 3

Минимальные значения  $r$ , при которых результат ДВП изображения койфлетами достигнет определенного качества

| Канал, бит | PSNR, дБ | <i>coif1</i> | <i>coif2</i> | <i>coif3</i> | <i>coif4</i> | <i>coif5</i> |
|------------|----------|--------------|--------------|--------------|--------------|--------------|
| 8          | 40       | 12           | 13           | 13           | 14           | 14           |
|            | $\infty$ | 14           | 15           | 15           | 16           | 16           |
| 12         | 40       | 12           | 13           | 13           | 14           | 14           |
|            | $\infty$ | 18           | 19           | 19           | 20           | 20           |
| 16         | 40       | 12           | 13           | 13           | 14           | 14           |
|            | $\infty$ | 22           | 22           | 23           | 24           | 24           |

искажений ( $PSNR = \infty$ ), можно определить по формуле

$$r = 5 + \log_2(M + 1) + \left\lfloor \sqrt{k/2} - 1 \right\rfloor, \quad (6)$$

где  $M$  — максимальное значение яркости вокселей изображения.

Формулы (5) и (6) применимы как для полутоновых, так и для полноцветных изображений. В следующем разделе продемонстрировано их использование на примере ДВП трехмерных медицинских изображений.

### Моделирование ДВП трехмерных медицинских изображений

Моделирование проведено в программной среде MATLAB версии R2018a для двух трехмерных томографических полутоновых изображений: "wmri" (рис. 2, а, см. четвертую сторону обложки) — 8-битное изображение размера  $128 \times 128 \times 27$ ; "Trufi\_COR" (рис. 3, а) — 12-битное изображение размера  $320 \times 320 \times 30$ . ДВП изображений осуществлено следующим образом: с помощью команды "wfilters" получены коэффициенты  $f_{F,i}$  фильтров  $db\ k/2$ ,  $sym\ k/2$  и  $coif\ k/6$ ; далее они квантованы и преобразованы в формат с фиксированной точкой умножением на  $2^n$  ( $n = 1, 2, 3, \dots, 20$ ) и округлением к большему по формуле (2); с помощью команды "dwt3" осуществлено ДВП изображений; результаты об-

работки  $I^*$  преобразованы делением на  $2^{6n}$  и округлением к меньшему по формуле (3).

Примеры обработки изображений "wmri" и "Trufi\_COR" вейвлетом *db8* представлены на рис. 2 (см. четвертую сторону обложки) и рис. 3 соответственно. Из рисунков видно, что при увеличении разрядности  $r$  качество обработ-

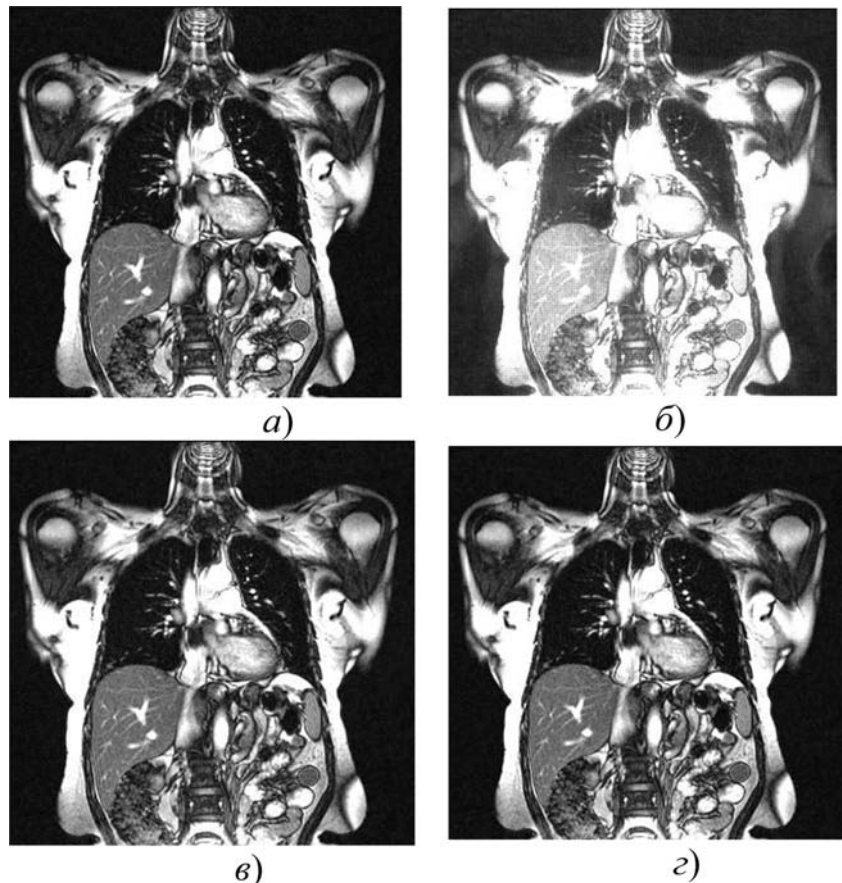


Рис. 3. Пример ДВП 12-битного изображения "Trufi\_COR" (15-й кадр) вейвлетом *db8*:

а) исходное изображение; обработанное изображение: б)  $r = 7$ ;  $PSNR = 30,27$  дБ; в)  $r = 9$ ;  $PSNR = 46,08$  дБ; г)  $r = 17$ ;  $PSNR = \infty$

ки постепенно улучшается: на рис. 2, б и 3, б видны искажения; на рис. 2, в и 3, в восстановленные изображения неотличимы на глаз от исходных изображений; на рис. 3, г и 4, г восстановленные изображения тождественно равны соответствующим исходным. Результаты моделирования имеют более высокое качество по сравнению с результатами теоретического анализа.

В табл. 4—6 представлены выборочные теоретические данные из табл. 1—3, дополненные

Таблица 6

**Минимальные значения  $r$ , при которых результат ДВП изображения койфлетами достигает определенного качества**

| Канал, бит | $PSNR$ , дБ | Результаты    | $coif1$ | $coif2$ | $coif3$ | $coif4$ | $coif5$ |
|------------|-------------|---------------|---------|---------|---------|---------|---------|
| 8          | 40          | Теоретические | 12      | 13      | 13      | 14      | 14      |
|            |             | Практические  | 10      | 12      | 12      | 12      | 12      |
|            |             | Разница       | 2       | 1       | 1       | 2       | 2       |
|            | $\infty$    | Теоретические | 14      | 15      | 15      | 16      | 16      |
|            |             | Практические  | 14      | 15      | 15      | 16      | 16      |
|            |             | Разница       | 0       | 0       | 0       | 0       | 0       |
| 12         | 40          | Теоретические | 12      | 13      | 13      | 14      | 14      |
|            |             | Практические  | 7       | 9       | 9       | 10      | 10      |
|            |             | Разница       | 5       | 4       | 4       | 4       | 4       |
|            | $\infty$    | Теоретические | 18      | 19      | 19      | 20      | 20      |
|            |             | Практические  | 17      | 17      | 18      | 18      | 18      |
|            |             | Разница       | 1       | 2       | 1       | 2       | 2       |

Таблицы 4

**Минимальные значения  $r$ , при которых результат ДВП изображения вейвлетами Добеши достигает определенного качества**

| Канал, бит | $PSNR$ , дБ | Результаты    | $db2$ | $db4$ | $db6$ | $db8$ | $db10$ |
|------------|-------------|---------------|-------|-------|-------|-------|--------|
| 8          | 40          | Теоретические | 11    | 12    | 13    | 13    | 14     |
|            |             | Практические  | 10    | 11    | 11    | 12    | 12     |
|            |             | Разница       | 1     | 1     | 2     | 1     | 2      |
|            | $\infty$    | Теоретические | 13    | 14    | 15    | 15    | 15     |
|            |             | Практические  | 13    | 14    | 15    | 15    | 15     |
|            |             | Разница       | 0     | 0     | 0     | 0     | 0      |
| 12         | 40          | Теоретические | 11    | 12    | 13    | 13    | 14     |
|            |             | Практические  | 7     | 8     | 9     | 9     | 9      |
|            |             | Разница       | 4     | 4     | 4     | 4     | 5      |
|            | $\infty$    | Теоретические | 17    | 18    | 19    | 19    | 20     |
|            |             | Практические  | 16    | 17    | 17    | 17    | 18     |
|            |             | Разница       | 1     | 1     | 2     | 2     | 2      |

Таблица 5

**Минимальные значения  $r$ , при которых результат ДВП изображения симлетами достигает определенного качества**

| Канал, бит | $PSNR$ , дБ | Результаты    | $sym2$ | $sym4$ | $sym6$ | $sym8$ | $sym10$ |
|------------|-------------|---------------|--------|--------|--------|--------|---------|
| 8          | 40          | Теоретические | 11     | 12     | 13     | 13     | 13      |
|            |             | Практические  | 10     | 11     | 11     | 12     | 12      |
|            |             | Разница       | 1      | 1      | 2      | 1      | 1       |
|            | $\infty$    | Теоретические | 13     | 14     | 15     | 15     | 15      |
|            |             | Практические  | 13     | 14     | 15     | 15     | 15      |
|            |             | Разница       | 0      | 0      | 0      | 0      | 0       |
| 12         | 40          | Теоретические | 11     | 12     | 13     | 13     | 14      |
|            |             | Практические  | 7      | 8      | 8      | 9      | 9       |
|            |             | Разница       | 4      | 4      | 5      | 4      | 5       |
|            | $\infty$    | Теоретические | 17     | 18     | 19     | 19     | 19      |
|            |             | Практические  | 16     | 16     | 17     | 18     | 18      |
|            |             | Разница       | 1      | 2      | 2      | 1      | 1       |

практическими результатами моделирования ДВП и разницей между ними.

На основе сделанных расчетов и моделирования, результаты которых представлены в табл. 4—6, мы можем сделать следующие выводы.

1. Все теоретические значения  $PSNR$ , полученные в предыдущем разделе, не превосходят соответствующие значения, полученные в результате моделирования, что подтверждает верность теоретического анализа. Таким образом, формулы (5) и (6) могут быть использованы для определения минимальной разрядности  $r$  коэффициентов вейвлет-фильтров с компактным носителем, при которой результат ДВП трехмерного медицинского изображения имеет высокое ( $PSNR \geq 40$  дБ) или максимальное ( $PSNR = \infty$ ) качество соответственно.

2. Для достижения высокого качества ДВП трехмерного медицинского 8-битного изображения на практике потребовалось на 1...2 бита меньше, чем того требуют результаты, полученные теоретически, так как в теоретическом анализе предсказан худший случай.

3. Для достижения высокого и максимального качества ДВП трехмерного медицинского 12-битного изображения на практике потребовалось на 4...5 битов меньше и на 1...2 бита меньше соответственно, чем того требуют результаты теоретического анализа. Наличие данной разницы объясняется тем, что в трехмерных медицинских 12-битных изображениях отношение среднего значения яркости вокселей к максимальному  $M$  в среднем непре-

небрежимо меньше, чем у 8-битных изображений, в результате чего мы получаем более высокие значения *PSNR*.

4. Качество ДВП трехмерных медицинских изображений в большей степени зависит от их цветовой глубины, от требуемого качества обработки, от числа коэффициентов фильтров выбираемого вейвлета и в меньшей степени от вида вейвлета.

### Заключение

В статье решена задача анализа влияния шума квантования коэффициентов фильтров дискретного вейвлет-преобразования (ДВП) на результат обработки трехмерных медицинских изображений. Предложен метод квантования коэффициентов вейвлет-фильтров, позволяющий минимизировать ресурсы при аппаратной реализации путем упрощения выполнения операций округления. Разработан метод, позволяющий рассчитать максимальную погрешность вычислений, которая может возникнуть в результате ДВП трехмерного полутонного или полноцветного изображения с различной глубиной цвета. Выведена формула (4), позволяющая определить минимальное качество ДВП трехмерных изображений в зависимости от используемого вейвлета, разрядности коэффициентов и цветовой глубины. Доказано, что формулы (5) и (6) позволяют определить минимальную разрядность коэффициентов вейвлет-фильтров с компактным носителем, при которой результат ДВП трехмерных изображений достигает высокого ( $PSNR \geq 40$  дБ) и максимального ( $PSNR = \infty$ ) качества в зависимости от используемого вейвлета соответственно. Моделирование ДВП трехмерных томографических изображений показало, что по сравнению с теоретическими результатами можно достичь сокращения разрядности коэффициентов, сохраняя при этом требуемое качество обработки. В предложенном методе реализации ДВП все данные представлены в формате с фиксированной точкой и упрощено выполнение операций округления.

Полученные результаты открывают возможность для эффективной аппаратной реализации ДВП на современных устройствах (FPGA, ASIC и др.) для очистки от шума и сжатия трехмерных медицинских изображений.

Авторы выражают свою признательность Ставропольскому краевому клиническому консультативно-диагностическому центру за пре-

доставление томографических изображений. Работа выполнена при финансовой поддержке: базовой части государственного задания № 2.6035.2017/БЧ; РФФИ, проекты № 18-07-00109 А, № 19-07-00130 А и № 18-37-20059 мол-а-вед; совета по грантам Президента Российской Федерации, проект СП-2245.2018.5.

### Список литературы

1. Barnhill E., Hollis L., Sack I., Braun J., Hoskins P. R., Pankaj P., Brown C., Van der Beek E. J. R., Roberts N. Nonlinear multiscale regularisation in MR elastography: Towards fine feature mapping // *Medical Image Analysis*. 2017. Vol. 35. P. 133–145.
2. Manjón J. V., Coupé P., Buades A. MRI noise estimation and denoising using non-local PCA // *Medical Image Analysis*. 2017. Vol. 22, N. 1. P. 35–47.
3. Lucas L. F. R., Rodrigues N. M. M., Cruz L. A. S., Van der Faria S. M. M. Lossless Compression of Medical Images Using 3D Predictors // *IEEE Transactions on Medical Imaging*. 2017. Vol. 36, N. 11. P. 2250–2260.
4. Schirmacher F., Köhler T., Endres J., Lindnerberger T., Husvogt L., Fujimoto J. G., Hornegger J., Dörfler A., Hoelter P., Maier A. K. Temporal and volumetric denoising via quantile sparse image prior // *Medical Image Analysis*. 2018. Vol. 48. P. 131–146.
5. Xu Z., Gao M., Papadakis G. Z., Luna B., Jain S., Mollura D. J., Bagci U. Joint solution for PET image segmentation, denoising, and partial volume correction // *Medical Image Analysis*. 2018. Vol. 46. P. 229–243.
6. Fang L., Li S., Kang X., Izatt J. A., Farsiu S. 3-D Adaptive Sparsity Based Image Compression With Applications to Optical Coherence Tomography // *IEEE Transactions on Medical Imaging*. 2015. Vol. 34, N. 6. P. 1306–1320.
7. Wu, H., Huynh T. T., Souvenir R. Echocardiogram enhancement using supervised manifold denoising // *Medical Image Analysis*. 2015. Vol. 35. P. 172–180.
8. Parikh S. S., Ruiz D., Kalva H., Fernández-Escribano G., Adzic V. High Bit-Depth Medical Image Compression with HEVC // *IEEE Journal of Biomedical and Health Informatics*. 2017. Vol. 22, N. 2. P. 552–560.
9. Song X., Huang O., Chang S., He J., Wang H. Three-dimensional separate descendant-based SPIHT algorithm for fast compression of high-resolution medical image sequences // *IET Image Processing*. 2017. Vol. 11, N. 1. P. 80–87.
10. Naveen C., Gupta T. V. S., Satpute V. R., Gandhi A. S. A simple and efficient approach for medical image security using chaos on EZW // *Eighth International Conference on Advances in Pattern Recognition (ICAPR)*. 2015. P. 1–6.
11. Bailey D. G. Design for Embedded Image Processing on FPGAs. Singapore: John Wiley & Sons, IEEE Press, 2011. 496 p.
12. Meyer-Baese U. Digital Signal Processing with Field Programmable Gate Arrays, fourth ed. Tallahassee: Springer-Verlag Berlin Heidelberg, 2014. 949 p.
13. Madanayake A., Cintra R. J., Dimitrov V., Bayer F. M., Wahid K. A., Kulasekera S., Edirisuriya A., Potluri U. S., Madi-shetty S. K., Rajapaksha N. Low-Power VLSI Architectures for DCT/DWT: Precision vs Approximation for HD Video, Biomedical, and Smart Antenna Applications // *IEEE Circuits and Systems Magazine*. 2015. Vol. 15, N. 1. P. 1–6.
14. Jiang R. M., Crookes D. FPGA Implementation of 3D Discrete Wavelet Transform for Real-Time Medical Imaging //

18th European Conference on Circuit Theory and Design. 2007. P. 519–522.

15. **Arafa A. A., Saleh H. I., Ashour M., Salem A.** FFT- and DWT-Based FPGA Realization of Pulse Shape Discrimination in PET system // 4th International Conference on Design & Technology of Integrated Systems in Nanoscale Era. 2009. P. 299–302.

16. **Li N.-q., Nie Y.-j., Zhu W.** The Application of FPGA-based Discrete Wavelet Transform System in EEG Analysis // Second International Conference on Intelligent System Design and Engineering Application. 2012. P. 1306–1309.

17. **Ahmad A., Ja'afar N. H., Amira A.** FPGA-based Implementation of 3-D Daubechies for Medical Image Compression // IEEE-EMBS Conference on Biomedical Engineering and Sciences. 2012. P. 683–688.

18. **Ja'afar N. H., Ahmad A., Amira A.** Distributed Arithmetic Architecture of Discrete Wavelet Transform (DWT) with Hybrid Method // IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS). 2013. P. 501–507.

19. **Mimouni E. H. E., Karim M.** An FPGA-Based Implementation of a Pre-Processing Stage for ECG Signal Analysis Using DWT // Second World Conference on Complex Systems (WCCS). 2014. P. 649–654.

20. **Vijendra V., Kulkarni M.** ECG signal Filtering using DWT haar wavelets Coefficient techniques // International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS). 2016. P. 1–6.

21. **Elsayed M., Badawy A., Mahmuiddin M., Elfouly T., Mohamed A., Abualsaud K.** FPGA Implementation of DWT EEG

Data Compression for Wireless Body Sensor Networks // IEEE Conference on Wireless Sensors (ICWiSE). 2016. P. 21–25.

22. **Sharmila A., Geethanjali P.** DWT Based Detection of Epileptic Seizure from EEG Signals Using Naive Bayes and k-NN Classifiers // IEEE Access. 2016. Vol. 4. P. 7716–7727.

23. **Larrotta D. M. B., Enciso D. M. M., Barrera A. E. G.** Compression of biomedical signals on FPGA by DWT and run-length // IEEE ANDESCON. 2010. P. 1–5.

24. **Chervyakov N. N., Lyakhov P. A., Kaplun D. I., Butusov D. N., Nagornov N. N.** Analysis of the Quantization Noise in Discrete Wavelet Transform Filters for Image Processing // Electronics. Vol. 7, N. 8. P. 135.

25. **Lalithakumari S., Pandian R., Rani J., Vinothkumar D., Sneha A., Amalarani V., Bestley J.** Selection of optimum compression algorithms based on the characterization on feasibility for medical image // Biomedical Research. 2017. Vol. 28, N. 13. P. 5633–5637.

26. **Daubechies, I.** Ten Lectures on Wavelets. Philadelphia: Society for Industrial and Applied Mathematics, 1992. 380 p.

27. **Vaidyanathan P. P.** Multirate Systems and Filter Banks. Englewood Cliffs: Prentice-Hall, 1993. 925 p.

28. **Rao K. R., Yip P. C.** The Transform and Data Compression Handbook // Boca Raton: CRC Press, 2001. 399 p.

29. **Ravichandran D., Ahamad M. G., Dhivakar A.** Performance Analysis of Three-Dimensional Medical Image Compression Based on Discrete Wavelet Transform // 22nd International Conference on Virtual System & Multimedia (VSMM). 2016. P. 1–8.

30. **Basso A., Cavagnino D., Pomponui V., Vernone A.** Blind watermarking of color images using Karhunen–Love transform keying // The Computer Journal. 2011. Vol. 54, N. 7. P. 1076–1090.

**N. N. Nagornov**, PhD student, North-Caucasus Federal University, e-mail: spartal392@mail.ru,  
**P. A. Lyakhov**, Candidate of Physics and Mathematics Sciences, Assistant Professor, e-mail: ljahov@mail.ru,  
**N. I. Chervyakov**, Doctor of Technical Sciences, Professor, Head of Department of Applied Mathematics and Mathematical Modeling, e-mail: k-fmf-primath@stavs.ru

## The Noise Research in Discrete Wavelet Transform Filters for 3D Images Processing in Medicine

*Many different methods are used to denoise and compress various medical images in practice. Discrete wavelet transform (DWT) underlies many of them. The quantization noise of DWT filters coefficients for 3D medical image processing analyzed in this paper. We proposed a new method for coefficients quantizing using rounding up and rounding down operations which reduce the error and require less resources for software and hardware implementation. The method for estimating maximum error of 3D images DWT with various bits per color (BPC) was developed. We defined the dependence of the minimum peak signal-to-noise ratio (PSNR) of grayscale and color images transform result on wavelet used, the effective bit-width of filters coefficients and BPC is. Approximate formulas for determining the minimum bit-width that provide a high ( $PSNR \geq 40$  dB) and maximum ( $PSNR = \infty$  dB) quality of 3D medical image processing using DWT filter bank was derived. Software simulation of 3D tomographic 8-bit and 12-bit grayscale images transform confirmed obtained theoretical results. Concluded that the quality of images DWT primarily depends on their BPC, on the required processing quality, on the number of wavelet filters coefficients and to a lesser extent on the type of wavelet. We only used fixed-point numbers in the proposed method for 3D medical imaging making possible its efficient software and hardware implementation on modern devices such as field-programmable gate arrays and application-specific integrated circuits.*

**Keywords:** quantization noise, wavelet transform, medical imaging

DOI: 10.17587/it.25.415-425



## References

1. Barnhill E., Hollis L., Sack I., Braun J., Hoskins P. R., Pankaj P., Brown C., Van der Beek E.J.R., Roberts N. Nonlinear multiscale regularisation in MR elastography: Towards fine feature mapping, *Medical Image Analysis*, 2017, vol. 35, pp. 133–145.
2. Manjon J. V., Coupe P., Buades A. MRI noise estimation and denoising using non-local PCA, *Medical Image Analysis*, 2017, vol. 22, no. 1, pp. 35–47.
3. Lucas L. F. R., Rodrigues N. M. M., Cruz L. A. S., Van der Faria S. M. M. Lossless Compression of Medical Images Using 3D Predictors, *IEEE Transactions on Medical Imaging*, 2017, vol. 36, no. 11, pp. 2250–2260.
4. Schirmacher F., Kohler T., Endres J., Lindnerberger T., Husvogt L., Fujimoto J.G., Hornegger J., Dorfner A., Hoelter P., Maier A.K. Temporal and volumetric denoising via quantile sparse image prior, *Medical Image Analysis*, vol. 48, pp. 131–146.
5. Xu Z., Gao M., Papadakis G. Z., Luna B., Jain S., Mol-lura D. J., Bagci U. Joint solution for PET image segmentation, denoising, and partial volume correction, *Medical Image Analysis*, 2018, vol. 46, pp. 229–243.
6. Fang L., Li S., Kang X., Izatt J.A., Farsiu S. 3-D Adaptive Sparsity Based Image Compression With Applications to Optical Coherence Tomography, *IEEE Transactions on Medical Imaging*, 2015, vol. 34, no. 6, pp. 1306–1320.
7. Wu H., Huynh T. T., Souvenir R. Echocardiogram enhancement using supervised manifold denoising, *Medical Image Analysis*, 2015, vol. 35, pp. 172–180.
8. Parikh S. S., Ruiz D., Kalva H., Fernandez-Escribano G., Adzic V. High Bit-Depth Medical Image Compression with HEVC, *IEEE Journal of Biomedical and Health Informatics*, 2017, vol. 22, no. 2, pp. 552–560.
9. Song X., Huang O., Chang S., He J., Wang H. Three-dimensional separate descendant-based SPIHT algorithm for fast compression of high-resolution medical image sequences, *IEE Image Processing*, 2017, vol. 11, no. 1, pp. 80–87.
10. Naveen C., Gupta T. V. S., Satpute V. R., Gandhi A. S. A simple and efficient approach for medical image security using chaos on EZW, *Eighth International Conference on Advances in Pattern Recognition (ICAPR)*, 2015, pp. 1–6.
11. Bailey D. G. *Design for Embedded Image Processing on FPGAs*, Singapore, John Wiley & Sons, IEEE Press, 2011, 496 p.
12. Meyer-Baese U. *Digital Signal Processing with Field Programmable Gate Arrays*, fourth ed., Tallahassee, Springer-Verlag Berlin Heidelberg, 2014, 949 p.
13. Madanayake A., Cintra R. J., Dimitrov V., Bayer F. M., Wahid K. A., Kulasekera S., Edirisuriya A., Potluri U. S., Madishetty S. K., Rajapaksha N. Low-Power VLSI Architectures for DCT/DWT: Precision vs Approximation for HD Video, *Biomedical, and Smart Antenna Applications, IEEE Circuits and Systems Magazine*, 2015, vol. 15, no. 1, pp. 1–6.
14. Jiang R. M., Crookes D. FPGA Implementation of 3D Discrete Wavelet Transform for Real-Time Medical Imaging, *18th European Conference on Circuit Theory and Design*, 2007, pp. 519–522.
15. Arafa A. A., Saleh H.I., Ashour M., Salem A. FFT- and DWT-Based FPGA Realization of Pulse Shape Discrimination in PET system, *4th International Conference on Design & Technology of Integrated Systems in Nanoscale Era*, 2009, pp. 299–302.
16. Li N.-q., Nie Y.-j., Zhu W. The Application of FPGA-based Discrete Wavelet Transform System in EEG Analysis, *Second International Conference on Intelligent System Design and Engineering Application*, 2012, pp. 1306–1309.
17. Ahmad A., Ja'afar N.H., Amira A. FPGA-based Implementation of 3-D Daubechies for Medical Image Compression, *IEEE-EMBS Conference on Biomedical Engineering and Sciences*, 2012, pp. 683–688.
18. Ja'afar N. H., Ahmad A., Amira A. Distributed Arithmetic Architecture of Discrete Wavelet Transform (DWT) with Hybrid Method, *IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS)*, 2013, pp. 501–507.
19. Mimouni E. H. E., Karim M. An FPGA-Based Implementation of a Pre-Processing Stage for ECG Signal Analysis Using DWT, *Second World Conference on Complex Systems (WCCS)*, 2014, pp. 649–654.
20. Vijendra V., Kulkarni M. ECG signal Filtering using DWT haar wavelets Coefficient techniques, *International Conference on Emerging Trends in Engineering, Technology and Science (ICE-TETS)*, 2016, pp. 1–6.
21. Elsayed M., Badawy A., Mahmuddin M., Elfouly T., Mohamed A., Abualsaud K. FPGA Implementation of DWT EEG Data Compression for Wireless Body Sensor Networks, *IEEE Conference on Wireless Sensors (ICWiSE)*, 2016, pp. 21–25.
22. Sharmila A., Geethanjali P. DWT Based Detection of Epileptic Seizure from EEG Signals Using Naive Bayes and k-NN Classifiers, *IEEE Access*, 2016, vol. 4, pp. 7716–7727.
23. Larrotta D. M. B., Enciso D. M. M., Barrera A. E. G. Compression of biomedical signals on FPGA by DWT and run-length, *IEEE ANDESCON*, 2010, pp. 1–5.
24. Chervyakov N. N., Lyakhov P. A., Kaplun D. I., Butusov D. N., Nagornov N. N. Analysis of the Quantization Noise in Discrete Wavelet Transform Filters for Image Processing, *Electronics*, vol. 7 (8), 135 p.
25. Lalithakumari S., Pandian R., Rani J., Vinothkumar D., Sneha A., Amalarani V., Bestley J. Selection of optimum compression algorithms based on the characterization on feasibility for medical image, *Biomedical Research*, 2017, vol. 28 (13), pp. 5633–5637.
26. Daubechies I. *Ten Lectures on Wavelets*, Philadelphia, Society for Industrial and Applied Mathematics, 1992, 380 p.
27. Vaidyanathan P. P. *Multirate Systems and Filter Banks*, Englewood Cliffs: Prentice-Hall, 1993, 925 p.
28. Rao K. R., Yip P. C. *The Transform and Data Compression Handbook*, Boca Raton, CRC Press, 2001, 399 p.
29. Ravichandran D., Ahamad M.G., Dhivakar A. Performance Analysis of Three-Dimensional Medical Image Compression Based on Discrete Wavelet Transform, *22nd International Conference on Virtual System & Multimedia (VSMM)*, 2016, pp. 1–8.
30. Basso A., Cavagnino D., Pomponui V., Vernone A. Blind watermarking of color images using Karhunen-Loeve transform keying, *The Computer Journal*, 2011, vol. 54, no. 7, pp. 1076–1090.

Д. В. Ефанов, д-р техн. наук, доц., e-mail: TrES-4b@yandex.ru,  
ООО "ЛокоТех-Сигнал",  
Российский университет транспорта (РУТ (МИИТ)), Москва

## Троичный код паритета в системах рабочего диагностирования устройств автоматики и вычислительной техники

*В статье обозначен интерес к развитию методов рабочего диагностирования устройств автоматики, функционирующих в троичной логике. Представлены аналогии между подходами к организации систем рабочего диагностирования устройств двоичной и троичной логики. Описан троичный код паритета и некоторые его свойства, учет которых целесообразен при организации систем диагностирования. Доказано, что все информационные векторы кода паритета распределены равномерно между всеми его контрольными векторами. Приведена формула подсчета числа необнаруживаемых ошибок в информационных векторах троичных кодов с равномерным распределением всех информационных векторов между всеми контрольными векторами. Доказана теорема о коде с наименьшим общим числом необнаруживаемых ошибок в информационных векторах троичных кодов для фиксированных значений длин информационных и контрольных векторов. Приведены правила построения модифицированных троичных кодов паритета.*

**Ключевые слова:** автоматика и вычислительная техника, троичная логика, рабочее диагностирование, результаты вычисления, косвенный контроль, ошибки на выходах, троичный код паритета

### Введение

Устройства автоматики и вычислительной техники в современном мире реализуются на основе компонентов, использующих принципы двоичного представления сигналов. Высокий уровень потенциала соответствует сигналу "логическая 1", а низкий — сигналу "логический 0" (или наоборот). Широкое распространение элементы, реализующие принципы двоичной логики, получили благодаря простоте реализации и вполне понятным принципам обеспечения надежности функционирования. Более того, признание основной именно двоичной логики для реализации вычислительных систем было зафиксировано в известной архитектуре фон Неймана [1].

Тем не менее, за годы развития компьютерной техники во второй половине XX — первой четверти XXI века были удачные попытки реализации вычислительных машин, функционирующих на принципах троичной логики [2]. Примерами тому являются троичные вычислительные системы "Сетунь", реализованная в МГУ в 1958 г. (под руководством Н. П. Бру-

сенева) [3, 4], и ТСА2 (версия v2.0), созданная в Калифорнийском политехническом университете в 2008 г. (Дж. Коннели, К. Пател, А. Чавез) [5]. В ряде публикаций [6—8] используется идея реализации устройств троичной логики, "собранных" на традиционных устройствах двоичной логики. О преимуществе троичной логики перед двоичной говорят некоторые разработчики квантовых компьютеров, предлагая использовать вместо кубитов информации кутриты, объясняя это серьезным уменьшением числа квантовых вентилей [9]. Кроме того, троичная логика может оказаться эффективной для информационной защиты современных устройств, использующих IoT [10].

Преимущества троичной логики перед двоичной (и перед любой другой) объясняются более плотной записью чисел. Кроме того, троичная логика "покрывает" двоичную и может использовать все ее преимущества, а вычислительные устройства, реализованные на троичной логике, должны оказаться более быстросрабатывающими [11].

Развитие компьютерных технологий и разработка киберфизических систем управле-

ния на фоне колоссального прогресса в области информатизации [12] подогревают интерес большого числа исследователей цифровых устройств, реализуемых в троичной логике [13–19]. Параллельно развитию самой троичной техники и методов ее описания должны развиваться методы технической диагностики систем, реализованных в троичной логике [20, 21].

В данной работе проводится параллель между элементарными подходами в технической диагностике (с упором на методы рабочего диагностирования [22]) традиционных устройств, функционирующих в двоичной (бинарной) логике, и устройств, реализованных в троичной (тернарной) логике (*ternary logic* или *three-valued logic*).

## 1. Ошибки на выходах логических устройств

Будем рассматривать цифровые устройства, реализованные в троичной логике. При этом следует пояснить тот факт, что в троичной логике существуют несколько вариантов систем обозначения и трактовки логических сигналов:  $(0, 1, 2)$ ,  $(-1, 0, 1)$ ,  $(0, 1/2, 1)$  и пр. В общем смысле троичная система счисления реализуется в двух вариантах, образуя несимметричную и симметричную системы счисления. В несимметричной системе счисления используются обозначения логических сигналов  $(0, 1, 2)$ . В симметричной системе счисления знаки обозначения логических сигналов следующие:  $(-1, 0, 1)$ . Вообще, символы, обозначающие логические уровни для устройств автоматики троичной логики, могут быть использованы любые (с оговоркой старшинства). Далее в рассуждениях будем использовать следующий набор символов:  $x, f \in \{i; 0; 1\}$ , ориентируясь на вариант симметричной системы счисления. Следует отметить, что все описанные далее результаты универсальны и не привязаны к конкретному виду системы обозначения, а сам символ третьего сигнала в виде "i" использован для простоты записи троичных векторов.

Один трит информации может принимать значения из множества  $f \in \{i; 0; 1\}$ , другими словами, иметь три варианта значений. На рис. 1 изображено некоторое устройство, реализованное на элементах троичной логики (они обозначены на рисунке как *TG* — *ternary gate*), на выходах которого реализуется троичный вектор  $\langle f_1 f_2 \dots f_{m-1} f_m \rangle$  (общее число таких троичных векторов определяется величиной  $3^m$ ). В процессе функционирования

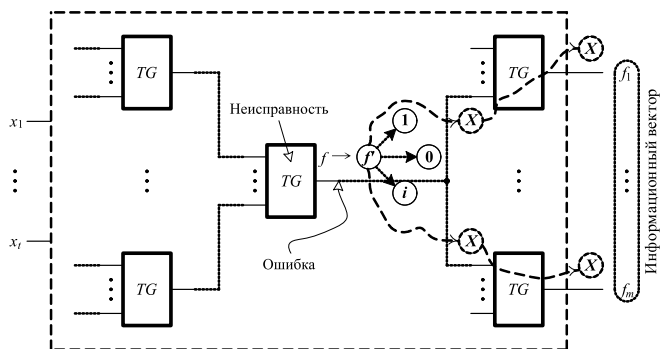
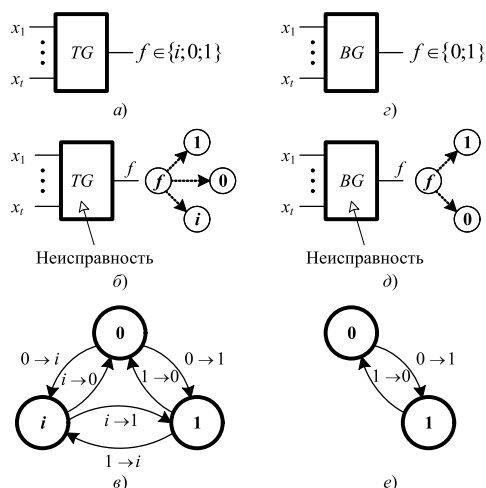


Рис. 1. Распространение ошибок на выходы устройства

устройства не исключены физические дефекты (неисправности), природа которых может быть самой разнообразной (например, в виду электромагнитных помех). Неисправности приводят к установлению на линиях схемы устройства ложных сигналов, а это, в свою очередь, является причиной трансляции неверных результатов вычислений на рабочие выходы устройства (на рис. 1 неизвестный сигнал показан символом "X" — это может быть как верный сигнал при компенсации ошибки, так и неверный сигнал). Ошибка в вычислениях на какой-либо линии схемы устройства приводит к формированию на его выходах ошибочного информационного вектора.

Элементарной моделью неисправностей для устройств, реализованных в троичной логике, по аналогии с двоичной логикой может служить модель одиночной константной неисправности (*stuck-at fault*). Суть ее сводится к тому, что в любой момент времени в устройстве возможна только одиночная неисправность, приводящая к установлению на выходах какого-либо элемента ложного сигнала  $i, 0$  или  $1$ . На рис. 2 в качестве примера представлены возможные варианты искажений значений на выходах логических элементов троичной и двоичной логики — соответственно трита и бита информации.

Таким образом, внутренние ошибки устройства приводят к искажениям значений выходного информационного вектора. Общее число таких ошибок определяется величиной  $N_m^3 = 3^m(3^m - 1)$ , где  $m$  — длина информационного вектора. Для сравнения число вариантов искажений двоичного информационного вектора длиной  $m$  определяется по формуле  $N_m^2 = 2^m(2^m - 1)$ . К примеру, при  $m = 3$  для двоичного вектора имеем  $N_3^2 = 56$  вариантов искажений, а для троичного —  $N_3^3 = 702$  (более чем в 12 раз!). С увеличением значения  $m$  число вариантов искажений троичных информационных векторов существенно растет (не-



**Рис. 2. Ошибки на выходах логических элементов троичной и двоичной логики (ошибки в тритах и битах):**

*a* — троичный логический элемент (*TG* — *ternary gate*); *б* — искажения функций троичного логического элемента; *в* — граф искажений значений трита; *г* — двоичный логический элемент (*BG* — *binary gate*); *д* — искажения функций двоичного логического элемента; *е* — граф искажений значений бита

сравнимо с числом вариантов искажений двоичных векторов):

$$\lim_{m \rightarrow \infty} \chi_m = \lim_{m \rightarrow \infty} \frac{N_m^2}{N_m^3} = \lim_{m \rightarrow \infty} \frac{2^m(2^m - 1)}{3^m(3^m - 1)} =$$

$$= \lim_{m \rightarrow \infty} \left( \frac{2}{3} \right)^m \lim_{m \rightarrow \infty} \frac{2^m - 1}{3^m - 1} = 0. \quad (1)$$

Уже при  $m = 5$  значение  $\chi_5 = 0,01687$ , а при  $m = 10$   $\chi_{10} = 0,0003$ .

Более того, ошибки в троичных информационных векторах гораздо "богаче" ошибок в двоичных информационных векторах (это как раз следует из многообразия вариантов искажений одного трита).

Возникает следующая задача: определить наличие неисправности в устройстве, рассматривая его как "черный ящик" и анализируя только выходные его значения. Эта задача решается методами рабочего диагностирования [23, 24]. В двоичной логике широко распространено применение избыточного кодирования для решения задачи рабочего диагностирования [25–28]. Избыточное кодирование может быть применено и для решения аналогичной задачи в троичной логике.

## 2. Троичный код паритета

Избыточный код, ориентированный на обнаружение ошибок в троичных информационных векторах, может быть построен путем использования элементарных троичных функций. В табл. 1 представлены описания основных используемых функций троичной логики (всего в троичной логике функций от двух переменных может быть построено  $3^{3^2} = 3^9 = 19\,683$ ). К слову, теория построения троичных кодов

Таблица 1

Описание троичных функций от двух переменных

| Название функций                   | Обозначение             | $x_1 x_2$ |      |      |      |      |      |      |      |      |
|------------------------------------|-------------------------|-----------|------|------|------|------|------|------|------|------|
|                                    |                         | $ii$      | $i0$ | $i1$ | $0i$ | $00$ | $01$ | $1i$ | $10$ | $11$ |
| Конъюнкция                         | $x_1 \cdot x_2$         | $i$       | $i$  | $i$  | $i$  | $0$  | $0$  | $i$  | $0$  | $1$  |
| Дизъюнкция                         | $x_1 \vee x_2$          | $i$       | $0$  | $1$  | $0$  | $0$  | $1$  | $1$  | $1$  | $1$  |
| Логическое умножение по модулю три | $x_1 \otimes x_2$       | $1$       | $0$  | $i$  | $0$  | $0$  | $0$  | $i$  | $0$  | $1$  |
| Логическое сложение по модулю три  | $x_1 \oplus x_2$        | $1$       | $i$  | $0$  | $i$  | $0$  | $1$  | $0$  | $1$  | $i$  |
| Функция Вебба                      | $x_1   x_2$             | $0$       | $1$  | $i$  | $1$  | $1$  | $i$  | $i$  | $i$  | $i$  |
| Пороговое сложение                 | $x_1 + x_2$             | $i$       | $i$  | $0$  | $i$  | $0$  | $1$  | $0$  | $1$  | $1$  |
| Исключающий максимум               | $x_1 \uparrow x_2$      | $i$       | $0$  | $1$  | $0$  | $i$  | $1$  | $1$  | $1$  | $i$  |
| Среднее                            | $x_1 \rightarrow x_2$   | $i$       | $0$  | $i$  | $0$  | $1$  | $0$  | $i$  | $0$  | $i$  |
| Сравнение                          | $x_1 \text{ cmp } x_2$  | $0$       | $i$  | $i$  | $1$  | $0$  | $i$  | $1$  | $1$  | $0$  |
| Сильная конъюнкция                 | $x_1 \&_L x_2$          | $i$       | $i$  | $i$  | $i$  | $i$  | $0$  | $i$  | $0$  | $1$  |
| Импликация Лукасевича              | $x_1 \rightarrow_L x_2$ | $1$       | $1$  | $1$  | $0$  | $1$  | $1$  | $i$  | $0$  | $1$  |
| Конъюнкция Клини                   | $x_1 \wedge_+ x_2$      | $i$       | $0$  | $i$  | $0$  | $0$  | $0$  | $i$  | $0$  | $1$  |
| Импликация Клини                   | $x_1 \rightarrow_+ x_2$ | $1$       | $0$  | $i$  | $1$  | $0$  | $0$  | $1$  | $0$  | $1$  |
| Импликация Гейтинга (Геделя)       | $x_1 \rightarrow_G x_2$ | $1$       | $1$  | $1$  | $i$  | $1$  | $1$  | $i$  | $0$  | $1$  |
| Материальная импликация            | $x_1 \rightarrow_M x_2$ | $1$       | $1$  | $1$  | $0$  | $0$  | $1$  | $i$  | $0$  | $1$  |
| Функция следования Брусенцова      | $x_1 \rightarrow_B x_2$ | $1$       | $0$  | $0$  | $0$  | $0$  | $0$  | $i$  | $0$  | $1$  |
| Тождество                          | $x_1 \equiv x_2$        | $1$       | $i$  | $i$  | $i$  | $1$  | $i$  | $i$  | $i$  | $1$  |

активно развивается [29—32], известны и методы построения блочных равномерных троичных кодов, например, аналога равновесного кода из бинарной логики, так называемого композиционного троичного кода [33, 34].

Для решения задач технической диагностики, в том числе рабочего диагностирования в условиях необходимости снижения избыточности устройства и упрощения технических средств диагностирования, полезными могут оказаться простейшие избыточные коды. Наипростейшим является *троичный код паритета (ternary parity code)*. Он строится по аналогии с двоичным кодом паритета путем использования функции *свертки по модулю три*, или *логического сложения по модулю три*:

$$g = f_1 \oplus f_2 \oplus \dots \oplus f_{m-1} \oplus f_m. \tag{2}$$

Таким образом, троичный код паритета имеет всего один контрольный бит, что позволяет обнаруживать любые одиночные ошибки в информационных векторах.

Известны практические реализации функции сложения по модулю три на традиционных микроэлектронных компонентах. Например, на рис. 3 приведена принципиальная схема реализации функции сложения по модулю три из работы [35], реализованная на элементах двоичной логики. В схеме присутствуют транзисторы  $VT_1$  и  $VT_2$ , представляющие собой пару взаимно дополнительных усилителей. Сигнал на выходе этой пары совпадает по фазе с входным сигналом и вдвое больше его по амплитуде. Напряжение смещения транзистора  $VT_3$  выбирается таким образом, чтобы сигнал через сопротивления  $R_1$  и  $R_2$  переключал ток, протекающий через элементы  $VD_1$  и  $R_3$ , что приводит к увеличению выходного тока транзистора  $VT_2$ .

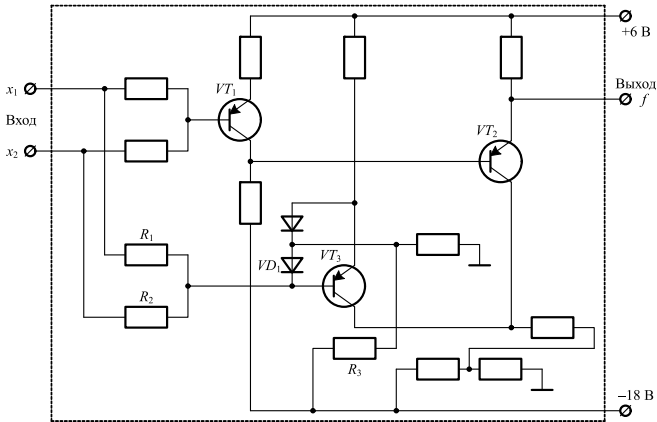


Рис. 3. Принципиальная схема устройства сложения по модулю три

Таблица 2  
Описание функции сложения по модулю три

| $x_1$ | $x_2$ |     |     |
|-------|-------|-----|-----|
|       | $i$   | 0   | 1   |
| $i$   | 1     | $i$ | 0   |
| 0     | $i$   | 0   | 1   |
| 1     | 0     | 1   | $i$ |

Ключевым отличием функции сложения по модулю три от остальных часто используемых функций (табл. 1) является тот факт, что ровно на трети входных наборов она принимает значения  $i$ , на трети входных наборов — значения 0 и на трети входных наборов — значения 1 (другими словами, распределение значений равномерно относительно всех входных наборов). Компактное описание функции сложения по модулю три приведено в табл. 2.

Обозначим троичный код паритета как  $P^T(m, k)$ -код, где  $k$  — число контрольных разрядов. Пример такого кода ( $P^T(3, 1)$ -кода) приведен в табл. 3.

Таблица 3  
Кодовые векторы  $P^T(3,1)$ -кода

| №  | $f_1$ | $f_2$ | $f_3$ | $g$ |
|----|-------|-------|-------|-----|
| 1  | $i$   | $i$   | $i$   | 0   |
| 2  | $i$   | $i$   | 0     | 1   |
| 3  | $i$   | $i$   | 1     | $i$ |
| 4  | $i$   | 0     | $i$   | 1   |
| 5  | $i$   | 0     | 0     | $i$ |
| 6  | $i$   | 0     | 1     | 0   |
| 7  | $i$   | 1     | $i$   | $i$ |
| 8  | $i$   | 1     | 0     | 0   |
| 9  | $i$   | 1     | 1     | 1   |
| 10 | 0     | $i$   | $i$   | 1   |
| 11 | 0     | $i$   | 0     | $i$ |
| 12 | 0     | $i$   | 1     | 0   |
| 13 | 0     | 0     | $i$   | $i$ |
| 14 | 0     | 0     | 0     | 0   |
| 15 | 0     | 0     | 1     | 1   |
| 16 | 0     | 1     | $i$   | 0   |
| 17 | 0     | 1     | 0     | 1   |
| 18 | 0     | 1     | 1     | $i$ |
| 19 | 1     | $i$   | $i$   | $i$ |
| 20 | 1     | $i$   | 0     | 0   |
| 21 | 1     | $i$   | 1     | 1   |
| 22 | 1     | 0     | $i$   | 0   |
| 23 | 1     | 0     | 0     | 1   |
| 24 | 1     | 0     | 1     | $i$ |
| 25 | 1     | 1     | $i$   | 1   |
| 26 | 1     | 1     | 0     | $i$ |
| 27 | 1     | 1     | 1     | 0   |

Следует отметить важную закономерность, присущую  $P^T(m, k)$ -кодам.

**Теорема 1.**  $P^T(m, k)$ -коды имеют равномерное распределение информационных векторов между всеми контрольными векторами.

**Доказательство** справедливости положения теоремы 1 тривиально и вытекает из того факта, что функция сложения по модулю три, используемая при вычислении контрольного разряда, имеет равномерное распределение относительно всех входных наборов. При построении кода как раз и рассматриваются все входные векторы. *Теорема доказана.*

Положение теоремы, несмотря на его простоту, в теории рабочего диагностирования логических устройств играет важную роль. Код, обладающий свойством равномерности распределения информационных векторов между всеми возможными контрольными векторами, будет иметь наименьшее число необнаруживаемых в информационных векторах ошибок среди всех возможных кодов для данных значений  $m$  и  $k$ .

**Теорема 2.** Троичный код с параметрами  $m$  и  $k$  будет обладать минимальным общим числом необнаруживаемых ошибок при условии, что все  $3^m$  информационных вектора будут равномерно распределены между всеми  $3^k$  контрольными векторами, а общее число необнаруживаемых ошибок в таком коде будет определяться по формуле

$$N_{m,k}^{\min} = 3^m (3^{m-k} - 1). \quad (3)$$

**Доказательство.** Число различных контрольных векторов, или *контрольных групп*, определяется числом разрядов в контрольных векторах и равно  $3^k$ . Так как всего существуют  $3^m$  информационных векторов, в каждой контрольной группе будет размещено ровно

по  $q = \frac{3^m}{3^k} = 3^{m-k}$  информационных векторов.

Число необнаруживаемых ошибок в одной такой контрольной группе будет равно  $q(q-1) = 3^{m-k}(3^{m-k}-1)$ . Так как число контрольных групп равно  $3^k$ , то умножая на данную величину полученное ранее выражение, получаем выражение (3).

Докажем, что именно формула (3) определяет минимальное общее число необнаруживаемых в информационных векторах ошибок. Для этого покажем, что минимальное нарушение равномерности распределения информационных векторов между контрольными векторами приведет к увеличению числа необнаруживаемых

ошибок по сравнению с рассчитанным по формуле (3).

Пусть в  $3^k - 2$  контрольных группах размещено ровно по  $q = 3^{m-k}$  информационных векторов, а в двух оставшихся размещены  $q-1 = 3^{m-k}-1$  и  $q+1 = 3^{m-k}+1$  векторов соответственно. Не будем рассматривать  $3^k - 2$  контрольные группы с  $q = 3^{m-k}$  информационными векторами, а остановимся на подсчете общего числа необнаруживаемых ошибок в двух оставшихся группах. Суммарно это следующее число необнаруживаемых ошибок:

$$\begin{aligned} Q_1 &= (3^{m-k}-1)(3^{m-k}-1-1) + \\ &+ (3^{m-k}+1)(3^{m-k}+1-1) = \\ &= (3^{m-k}-1)(3^{m-k}-2) + (3^{m-k}+1)3^{m-k} = \quad (4) \\ &= 3^{2(m-k)} - 3 \cdot 3^{(m-k)} + 2 + 3^{2(m-k)} + 3^{(m-k)} = \\ &= 2 \cdot 3^{2(m-k)} - 2 \cdot 3^{(m-k)} + 2. \end{aligned}$$

Для двух групп с  $q = 3^{m-k}$  информационными векторами число необнаруживаемых ошибок равно

$$Q_2 = 2 \cdot 3^{m-k} (3^{m-k} - 1) = 2 \cdot 3^{2(m-k)} - 2 \cdot 3^{m-k}. \quad (5)$$

Найдем значение величины  $\Delta = Q_1 - Q_2$ :

$$\begin{aligned} \Delta &= (2 \cdot 3^{2(m-k)} - 2 \cdot 3^{(m-k)} + 2) - \\ &- (2 \cdot 3^{2(m-k)} - 2 \cdot 3^{m-k}) = 2. \end{aligned} \quad (6)$$

Значение величины  $\Delta = 2 > 0$  при любых значениях  $m$  и  $k$ . Другими словами, число необнаруживаемых ошибок в результате нарушения равномерности увеличилось на 2. Еще большее нарушение равномерности распределения всех  $3^m$  информационных векторов между всеми  $3^k$  контрольными векторами приводит только к увеличению числа необнаруживаемых ошибок.

Таким образом, минимальное общее число необнаруживаемых ошибок в троичном коде с параметрами  $m$  и  $k$  определяется формулой (3). Это, в свою очередь, означает, что код с минимальным общим числом ошибок в информационных векторах будет иметь равномерное распределение информационных векторов между всеми контрольными векторами, что и требовалось доказать.

В табл. 4 все информационные векторы  $\langle f_1 f_2 f_3 \rangle P^T(3,1)$ -кода распределены между всеми однотритными контрольными векторами. Распределение равномерно.

Таблица 4

Распределение информационных векторов на контрольные группы для  $P^T(3,1)$ -кода

| $g$                           |       |       |
|-------------------------------|-------|-------|
| $i$                           | 0     | 1     |
| $\langle f_1 f_2 f_3 \rangle$ |       |       |
| $ii1z$                        | $iii$ | $ii0$ |
| $i00$                         | $i01$ | $i0i$ |
| $i1i$                         | $i10$ | $i11$ |
| $0i0$                         | $0i1$ | $0ii$ |
| $00i$                         | $000$ | $001$ |
| $011$                         | $01i$ | $010$ |
| $1ii$                         | $1i0$ | $1i1$ |
| $101$                         | $10i$ | $100$ |
| $110$                         | $111$ | $11i$ |

Анализ контрольных групп, соответствующих контрольным векторам кода, позволяет подсчитать общее число необнаруживаемых кодом ошибок. В каждой контрольной группе такого кода находится по  $\frac{3^m}{3} = 3^{m-1}$  информационных вектора. Искажение не будет обнаружено кодом в том случае, если информационный вектор одной группы перейдет в результате искажения в информационный вектор другой группы. Число таких переходов внутри группы равно  $3^{m-1}(3^{m-1} - 1)$ . Так как группы три, то общее число необнаруживаемых рассматриваемым кодом ошибок равно  $N_{m,1}^3 = 3 \cdot 3^{m-1}(3^{m-1} - 1) = 3^m(3^{m-1} - 1)$ . Троичным кодом паритета не обнаруживаются  $N_{3,1}^3 = 3^3(3^{3-1} - 1) = 27 \cdot 8 = 216$  ошибок в информационных векторах (среди них 162 двукратных и 54 трехкратных ошибок). Для сравнения, бинарный аналог данного кода — код паритета — не обнаруживает  $N_{m,1}^2 = 2^m(2^{m-1} - 1)$  ошибок в информационных векторах. Для трехбитного вектора это число равно  $N_{3,1}^2 = 24$  ошибки, что в девять раз меньше, чем в троичном коде паритета для той же длины информационного вектора.

Определим, какую долю ошибок в информационных векторах можно обнаружить с помощью троичного кода паритета:

$$\begin{aligned} \lim_{m \rightarrow \infty} \gamma_{m,k} &= \lim_{m \rightarrow \infty} \frac{N_{m,1}^3}{N_m^3} = \\ &= \lim_{m \rightarrow \infty} \frac{3^m(3^{m-1} - 1)}{3^m(3^m - 1)} = \lim_{m \rightarrow \infty} \frac{3^{m-1} - 1}{3^m - 1} = \frac{1}{3}. \end{aligned} \quad (7)$$

Бинарный код паритета не обнаруживает в пределе  $m \rightarrow \infty$  50 % ошибок, тогда как тернарный код паритета — 33,3 %.

Полученный результат показывает аналогию между кодами паритета в двоичной и троичной логике, а также говорит о том, что доля необнаруживаемых  $P^T(m, k)$ -кодами ошибок от общего их числа в информационных векторах меньше, чем аналогичный показатель для двоичного кода паритета. Тем не менее, троичные коды паритета могут быть использованы при разработке технических средств диагностирования логических устройств с учетом основной своей особенности — обнаружения однократных искажений. Это позволяет транслировать известные подходы по применению двоичных кодов паритета [36, 37] при организации систем рабочего диагностирования. При этом следует установить особенности возникающих в троичных информационных векторах (и на рабочих выходах объектов диагностирования) ошибок, подобно тому, как это сделано в работе [38] для бинарных разделимых кодов.

Для построения  $P^T(3,1)$ -кода используется функция  $g = f_1 \oplus f_2 \oplus f_3$ . При синтезе кодера такого кода потребуется каскадное подключение двух сумматоров по модулю три (рис. 3). Для улучшения характеристик обнаружения ошибок в информационных векторах без внесения дополнительной сложности в кодер может быть использовано выделение двух контрольных тритов:  $g_1 = f_1 \oplus f_2$  и  $g_2 = f_2 \oplus f_3$ . Для построенного таким образом  $P^T(3,2)$ -кода распределение информационных векторов между всеми контрольными векторами представлено в табл. 5. Согласно формуле (3)  $P^T(3,2)$ -кодом не будет обнаруживаться 54 ошибки, что в четыре раза меньше, чем число ошибок, обнаруживаемых  $P^T(3,1)$ -кодом. Кроме того, в отличие от  $P^T(3,1)$ -кода у  $P^T(3,2)$ -кода все векторы внутри контрольных групп имеют кодовое расстояние Хэмминга  $d = 3$ , а значит, ими

Таблица 5

Распределение информационных векторов на контрольные группы для  $P^T(3,2)$ -кода

| $\langle g_1 g_2 \rangle$     |       |       |       |       |       |       |       |       |
|-------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| $ii$                          | $i0$  | $i1$  | $0i$  | $00$  | $01$  | $1i$  | $10$  | $11$  |
| $\langle f_1 f_2 f_3 \rangle$ |       |       |       |       |       |       |       |       |
| $i0i$                         | $i00$ | $101$ | $i11$ | $i1i$ | $i10$ | $ii0$ | $ii1$ | $iii$ |
| $0i0$                         | $0i1$ | $0ii$ | $00i$ | $000$ | $001$ | $011$ | $01i$ | $010$ |
| $111$                         | $11i$ | $110$ | $1i0$ | $1i1$ | $1ii$ | $10i$ | $100$ | $101$ |



не обнаруживаются только трехкратные ошибки в информационных векторах (всего 54 трехкратных ошибки). Таким образом, распределение необнаруживаемых ошибок по кратностям, что немаловажно, смещено в сторону ошибок большей кратности.

Отметим, что способ выделения двух контрольных трит можно использовать аналогично контролю двух групп выходов устройства. Для кодов же с большим числом информационных разрядов можно строить модифицированные коды паритета с тремя и более контрольными разрядами (в том числе аналогично двоичных полиномиальных кодов и кодов Рида—Маллера [39]).

В заключение следует обратить внимание и на то, что представленные правила не являются единственными и возможны другие альтернативные способы построения кодов паритета. Например, в работе [40] отмечается, что известен способ построения кода с выделением двух контрольных битов, каждый из которых предназначен для отдельной проверки четности суммы чисел "i" и "1".

### Заключение

При синтезе технических средств диагностирования для устройств автоматики, функционирующих в троичной логике, могут эффективно применяться троичные коды, ориентированные на обнаружение ошибок в информационных векторах. Одним из эффективных кодов для этих целей является троичный код паритета. Его можно применять при реализации технических средств рабочего диагностирования следующими способами. Первый состоит в том, что осуществляется поиск групп независимых выходов устройств и строятся отдельные схемы контроля результатов вычислений, а контрольные выходы получаемых схем контроля объединяются на входах самопроверяемых компараторов троичных сигналов. Второй способ связан с преобразованиями структур объектов диагностирования в структуры с независимыми выходами и контролем общей группы на основе кода паритета.

Троичный код паритета относится к троичным кодам с наименьшим общим числом необнаруживаемых ошибок в информационных векторах для своих параметров  $m$  и  $k$ . Могут быть также построены такие коды с улучшенными характеристиками обнаружения ошибок в информационных векторах на основе выбора

нескольких функций свертки по модулю три и образования из них контрольных векторов.

Использование модифицированных троичных кодов паритета может оказаться эффективным с позиции снижения затрат на реализацию технических средств диагностирования по сравнению с дублированием.

### Список литературы

1. **Aspray W.** John von Neumann and the Origins of Modern Computing (History of Computing). Boston, Cambridge: MIT, 1990, 396 p.
2. **Карпенко А. С.** Развитие многозначной логики. М.: Издательство ЛКИ, 2010, 448 с.
3. **Брусенцов Н. П.** Отчего математическая логика не содержательна // Историко-математические исследования / Под редакцией С. С. Демидова. Вторая серия. Вып. 11 (46), М.: Янус-К, 2006. С. 228—234.
4. **Брусенцов Н. П., Маслов С. П., Розин В. П., Тишулина А. М.** Малая цифровая вычислительная машина "Сетунь". М.: Издательство МГУ, 1962. 140 с.
5. **Connely J.** Ternary Computing Testbed 3-Trit Computer Architecture. California Polytechnic State University of San Luis Obispo, August 29th, 2008. 184 p.
6. **Roy D., Merril Jr.** Ternary Logic in Digital Computers // Proceedings of the SHARE design automation project (DAC '65), ACM New York, NY, USA. P. 6.1—6.17. DOI: 10.1145/800266.810759.
7. **Hu M., Smith K. C.** Self-Checking Binary Logic Systems Using Ternary Logic Circuits // Canadian Electrical Engineering Journal. 1984. Vol. 9, Iss. 3. P. 100—104. DOI: 10.1109/CEEJ.1984.6593793.
8. **Wu J.** Ternary Logic Circuit for Error Detection and Error Correction // Proceedings of 19th International Symposium on Multiple-Valued Logic. 29—31 May 1989. Guangzhou, China. P. 94—99. DOI: 10.1109/ISMVL.1989.37766.
9. **Lanyon B. P., Barbieri M., Almeida M. P., Jennewein T., Ralph T. C., Resch K. J., Pryde G. J., O'Brien J. L., Gilchrist A., White A. G.** Simplifying Quantum Logic Using Higher-Dimensional Hilbert Spaces // Nature Physics. 2009. Vol. 5, Iss. 2. P. 134—140. DOI: 10.1038/nphys1150.
10. **Cambou B., Flikkema P. G., Palmer J., Telesca D., Philabaum C.** Can Ternary Computing Improve Information Assurance? // Cryptography. 2018. Vol. 2, Iss. 1 (March 2018). P. 1—16. DOI: 10.3390/cryptography2010006.
11. **Петров А.** Троичный компьютер: да, нет, может быть // Популярная механика. 2011. № 9. С. 72—76.
12. **Hahanov V.** Cyber Physical Computing for IoT-driven Services. New York, Springer International Publishing AG, 2018. 279 p.
13. **Vudadha C., Katragadda S., Phaneendra P. S.** 2:1 Multiplexer Based Design for Ternary Logic Circuits // IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia). 19—21 December 2013. Visakhapatnam, India. P. 46—51. DOI: 10.1109/PrimeAsia.2013.6731176.
14. **Ahmad S., Alam M.** Balanced-Ternary Logic for Improved and Advanced Computing // International Journal of Computer Science and Information Technologies (IJCSIT). 2014. Vol. 5, Iss. 4. P. 5157—5160.
15. **Гиниятуллин В. М., Арсланов И. Г., Богданова П. Д., Габитов Р. Н., Салихова М. А.** Способы реализации функций троичной логики // Программные системы и вычислительные методы. 2014. № 2. С. 239—254. DOI: 10.7256/2305-6061.2014.2.11820.
16. **Nair R. S. P., Smith S. C., Di J.** Delay Insensitive Ternary CMOS Logic for Secure Hardware // Journal of Low Power Electronics and Applications. 2015. Iss. 5. P. 183—215. DOI: 10.3390/jlpea5030183.

17. Uma Mahesh R. N., Sudeep J. Design and Novel Approach for Ternary and Quaternary Logic Circuits // 2nd International Conference for Convergence in Technology (I2CT). 7–9 April 2017. Mumbai, India. P. 1224–1227. DOI: 10.1109/I2CT.2017.8226322.
18. Kim S., Lim T., Kang S. An Optimal Gate Design for the Synthesis of Ternary Logic Circuits // 23rd Asia and South Pacific Design Automation Conference (ASP-DAC). 22–25 January 2018. Jeju, South Korea. P. 476–481. DOI: 10.1109/ASP-DAC.2018.8297369.
19. Vudadha C., Rajagopalan S., Dusi A., Phaneendra P. S., Srinivas M. B. Encoder-Based Optimization of CNFET-Based Ternary Logic Circuits // IEEE Transactions on Nanotechnology. 2018. Vol. 17. Iss. 2. P. 299–310. DOI: 10.1109/TNANO.2018.2800015.
20. Rahman Md. R., Rise J. E. On Designing a Ternary Reversible Circuit for Online Testability // IEEE Pacific Rim Conference on Communications, Computers and Signal Processing. 2011. P. 1–7. DOI: 10.1109/PACRIM.2011.6032878.
21. Nayeem N. M., Rice J. E. Design of an Online Testable Ternary Circuit from the Truth Table // Lecture Notes in Computer Science book series (LNCS, volume 7581): Reversible Computation, 4th International Workshop on Reversible Computation (RC 2012). Copenhagen, Denmark, July 2–3. 2012. P. 152–159.
22. Пархоменко П. П., Согомонян Е. С. Основы технической диагностики (оптимизация алгоритмов диагностирования, аппаратные средства). М.: Энергоатомиздат, 1981. 320 с.
23. Gössel M., Ocheretny V., Sogomonyan E., Marienfeld D. New Methods of Concurrent Checking: Edition 1. Dordrecht: Springer Science + Business Media B. V., 2008. 184 p.
24. Kharchenko V., Kondratenko Yu., Kacprzyk J. Green IT Engineering: Concepts, Models, Complex Systems Architectures // Springer Book series "Studies in Systems, Decision and Control". 2017. Vol. 74. 305 p.
25. Согомонян Е. С., Слабаков Е. В. Самопроверяемые устройства и отказоустойчивые системы. М.: Радио и связь, 1989. 208 с.
26. Siewiorek D., Swarz R. Reliable Computer Systems: Design and Evaluation. USA, Bedford: Digital Press, 1992. 908 p.
27. Goessel M., Graf S. Error Detection Circuits. London: McGraw-Hill, 1994. 261 p.
28. Mitra S., McCluskey E. J. Which Concurrent Error Detection Scheme to Choose? // Proceedings of International Test Conference, 2000, USA, Atlantic City, NJ, 03–05 October 2000. P. 985–994. DOI: 10.1109/TEST.2000.894311.
29. Brouwer A. E., Hamalainen H. O., Ostergard P. R. J., Sloane N. J. A. Bounds on Mixed Binary/Ternary Codes // IEEE Transactions on Information Theory. 1988. Vol. 44, Iss. 1. P. 140–161. DOI: 10.1109/18.651001.
30. Gulliver T. A., Ostergard P. R. J. Improved Bounds for Ternary Linear Codes of Dimension 7 // IEEE Transactions on Information Theory. 1997. Vol. 43, Iss. 4. P. 1377–1381. DOI: 10.1109/18.605613.
31. Bitouze N., Graell i Amat A., Rosnes E. Error Correcting Coding for a Nonsymmetric Ternary Channel // IEEE Transactions on Information Theory. 2010. Vol. 56, Iss. 11. P. 5715–5729. DOI: 10.1109/TIT.2010.2069211.
32. Laaksonen A., Östergård P. R. J. New Lower Bounds on Error-Correcting Ternary, Quaternary and Quinary Codes // Lecture Notes in Computer Science 10495, Springer: Coding Theory and Applications. 5th International Castle Meeting, ICMCTA 2017, Vihula, Estonia, August 28–31, 2017. P. 228–237.
33. Svanström M. A Lower Bound for Ternary Constant Weight Codes // IEEE Transactions on Information Theory. 1997. Vol. 43. P. 1630–1632.
34. Svanström M., Östergård P. R. J., Bogdanova G. T. Bounds and Constructions for Ternary Constant-Composition Codes // IEEE Transactions on Information Theory. 2002. Vol. 48. P. 101–111.
35. Пospelov Д. А. Логические методы анализа и синтеза схем. М.: Энергия, 1974. 368 с.
36. Abramovici M., Breuer M. A., Friedman A. D. Digital System Testing and Testable Design. Computer Science Press, 1998. 652 p.
37. Ефанов Д. В., Сапожников В. В., Сапожников Вл. В. Синтез самопроверяемых комбинационных устройств на основе выделения специальных групп выходов // Автоматика и телемеханика. 2018. № 9. С. 79–94.
38. Сапожников В. В., Сапожников Вл. В., Ефанов Д. В. Классификация ошибок в информационных векторах систематических кодов // Известия вузов. Приборостроение. 2015. Т. 58, № 5. С. 333–343. DOI: 10.17586/0021-3454-2015-58-5-333-343.
39. Сагалович Ю. Л. Введение в алгебраические коды. М.: Ин-т проблем передачи информации им. А. А. Харкевича РАН, 2010. 302 с.
40. Mirzaee R. F., Daliri M. S., Navi K., Bagherzadeh N. A Single Parity-Check Digit for One Trit Error Detection in Ternary Communication Systems: Gate-Level and Transistor-Level Designs // Journal of multiple-valued logic and soft computing. 2017. Vol. 29, N. 3–4. P. 303–326.

**D. V. Efanov**, D. Sc., Associate Professor,

Head of the Direction of monitoring and diagnosis systems department of "LocoTech-Signal" LLC,  
Professor of Russian university of transport, Moscow, Russian Federation, e-mail: TrES-4b@yandex.ru

## The Ternary Parity Code in the On-Line Testing of Automation Devices

*The author of the article raises the question about the development of methods for the on-line testing of ternary automation devices. The article presents an analogy between the approaches to the organization of systems for devices on-line testing using binary and ternary logic. The ternary parity code and some of its properties are described, the accounting of which is expedient when organizing diagnostic systems. It is proved that all parity code data vectors are distributed evenly among all its check vectors. A formula for counting the number of undetectable errors in the ternary code data vectors with a uniform distribution of all data vectors between all check vectors is given. A coding theorem is proved with the least total number of undetectable errors in the ternary code data vectors for fixed values of the lengths of data and check vectors. The rules for constructing modified ternary parity codes are given.*

**Keyword:** automation and computer engineering; ternary logic; on-line testing; the calculated result; indirectly controlled; errors at the outputs; ternary parity code

DOI: 10.17587/it.25.426-434

## References

1. **Aspray W.** John von Neumann and the Origins of Modern Computing (History of Computing), Boston, Cambridge, MIT, 1990, 396 p.
2. **Karpenko A. S.** Razvitie mnogoznachnoj logiki (The development of multi-valued logic), Moscow, Pub. House LKI, 2010, 448 p. (in Russian).
3. **Brusencov N. P.** *Istoriko-matematicheskie issledovaniya; Pod redakciej S. S. Demidova. Vtoraya seriya*, iss. 11 (46), Moscow, Yanus-K, 2006, pp. 228–234 (in Russian).
4. **Brusencov N. P., Maslov S. P., Rozin V. P., Tishulina A. M.** Small digital computing machine Setun, Moscow, Pub. House of MGU, 1962, 140 p. (in Russian).
5. **Connely J.** Ternary Computing Testbed 3-Trit Computer Architecture, California Polytechnic State University of San Luis Obispo, August 29th, 2008, 184 p.
6. **Roy D., Merril Jr.** *Proceedings of the SHARE design automation project (DAC '65)*, ACM New York, NY, USA, pp. 6.1–6.17, DOI: 10.1145/800266.810759.
7. **Hu M., Smith K. C.** *Canadian Electrical Engineering Journal*, 1984, vol. 9, iss. 3, pp. 100–104, DOI: 10.1109/CEEJ.1984.6593793.
8. **Wu J.** *Proceedings of 19th International Symposium on Multiple-Valued Logic*, 29–31 May 1989, Guangzhou, China, pp. 94–99, DOI: 10.1109/ISMVL.1989.37766.
9. **Lanyon B. P., Barbieri M., Almeida M. P., Jennewein T., Ralph T. C., Resch K. J., Pryde G. J., O'Brien J. L., Gilchrist A., White A. G.** *Nature Physics*, 2009, vol. 5, iss. 2, pp. 134–140, DOI: 10.1038/nphys1150.
10. **Cambou B., Flikkema P. G., Palmer J., Telesca D., Philabaum C.** *Cryptography*, 2018, vol. 2, iss. 1 (March 2018), pp. 1–16, DOI: 10.3390/cryptography2010006.
11. **Petrov A.** *Populyarnaya mekhanika*, 2011, iss. 9, pp. 72–76 (in Russian).
12. **Hahanov V.** *Cyber Physical Computing for IoT-driven Services*, New York, Springer International Publishing AG, 2018, 279 p.
13. **Vudadha C., Katragadda S., Phaneendra P. S.** *IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia)*, 19–21 December 2013, Visakhapatnam, India, pp. 46–51, DOI: 10.1109/PrimeAsia.2013.6731176.
14. **Ahmad S., Alam M.** *International Journal of Computer Science and Information Technologies (IJCSIT)*, 2014, vol. 5, iss. 4, pp. 5157–5160.
15. **Giniyatullin V. M., Arslanov I. G., Bogdanova P. D., Gabitov R. N., Salihova M. A.** *Programmnye Sistemy i Vychislitel'nye Metody*, 2014, iss. 2, pp. 239–254, DOI: 10.7256/2305-6061.2014.2.11820 (in Russian).
16. **Nair R. S.P., Smith S. C., Di J.** *Journal of Low Power Electronics and Applications*, 2015, iss. 5, pp. 183–215, DOI: 10.3390/jlpea5030183.
17. **Uma Mahesh R. N., Sudeep J.** *2nd International Conference for Convergence in Technology (I2CT)*, 7–9 April 2017, Mumbai, India, pp. 1224–1227, DOI: 10.1109/I2CT.2017.8226322.
18. **Kim S., Lim T., Kang S.** *23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, 22–25 January 2018, Jeju, South Korea, pp. 476–481, DOI: 10.1109/ASPDAC.2018.8297369.
19. **Vudadha C., Rajagopalan S., Dusi A., Phaneendra P. S., Srinivas M. B.** *IEEE Transactions on Nanotechnology*, 2018, vol. 17, iss. 2, pp. 299–310, DOI: 10.1109/TNANO.2018.2800015.
20. **Rahman Md.R., Rise J. E.** *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 2011, pp. 1–7, DOI: 10.1109/PACRIM.2011.6032878.
21. **Nayeem N. M., Rice J. E.** *Lecture Notes in Computer Science book series (LNCS, volume 7581): Reversible Computation, 4th International Workshop on Reversible Computation (RC 2012)*, Copenhagen, Denmark, July 2–3, 2012, pp. 152–159.
22. **Parkhomenko P. P., Sogomonyan E. S.** *Technical Diagnosis Fundamentals (Diagnostic Algorithm Optimization, Apparatus Means)*, Moscow, Energoatomizdat, 1981, 320 p. (in Russian).
23. **Göessel M., Ocheretny V., Sogomonyan E., Marienfeld D.** *New Methods of Concurrent Checking: Edition 1*, Dordrecht: Springer Science + Business Media B. V., 2008, 184 p.
24. **Kharchenko V., Kondratenko Yu., Kacprzyk J.** *Green IT Engineering: Concepts, Models, Complex Systems Architectures, Springer Book series "Studies in Systems, Decision and Control"*, vol. 74, 2017, 305 p.
25. **Sogomonyan E. S., Slabakov E. V.** *Self-checking devices and fault-tolerant systems*, Radio & Svyaz', Moscow, 208 p. (in Russian).
26. **Siewiorek D., Swarz R.** *Reliable Computer Systems: Design and Evaluation*, USA, Bedford, Digital Press, 1992, 908 p.
27. **Goessel M., Graf S.** *Error Detection Circuits*, London, McGraw-Hill, 1994, 261 p.
28. **Mitra S., McCluskey E. J.** *Proceedings of International Test Conference*, 2000, USA, Atlantic City, NJ, 03–05 October 2000, pp. 985–994, DOI: 10.1109/TEST.2000.894311.
29. **Brouwer A. E., Hamalainen H. O., Ostergard P. R. J., Sloane N. J. A.** *Bounds on Mixed Binary/Ternary Codes* // *IEEE Transactions on Information Theory*, 1988, vol. 44, iss. 1, pp. 140–161, DOI: 10.1109/18.651001.
30. **Gulliver T. A., Ostergard P. R.J.** *IEEE Transactions on Information Theory*, 1997, vol. 43, iss. 4, pp. 1377–1381, DOI: 10.1109/18.605613.
31. **Bitouze N., Graell i Amat A., Rosnes E.** *IEEE Transactions on Information Theory*, 2010, vol. 56, iss. 11, pp. 5715–5729, DOI: 10.1109/TIT.2010.2069211.
32. **Laaksonen A., Östergård P. R.J.** *Lecture Notes in Computer Science 10495, Springer: Coding Theory and Applications. 5th International Castle Meeting, ICMCTA 2017*, Vihula, Estonia, August 28–31, 2017, pp. 228–237.
33. **Svanström M.** *IEEE Transactions on Information Theory*, 1997, vol. 43, pp. 1630–1632.
34. **Svanström M., Östergård P. R. J., Bogdanova G. T.** *IEEE Transactions on Information Theory*, 2002, vol. 48, pp. 101–111.
35. **Pospelov D. A.** *Logical methods of analysis and synthesis of circuits*, Moscow, Energija, 1974, 368 p. (in Russian).
36. **Abramovici M., Breuer M. A., Friedman A. D.** *Digital System Testing and Testable Design*, Computer Science Press, 1998, 652 p.
37. **Efanov D. V., Sapozhnikov V. V., Sapozhnikov V. V.** *Avtomatika i telemekhanika*, 2018, iss. 9, pp. 79–94. (in Russian).
38. **Sapozhnikov V. V., Sapozhnikov V. V., Efanov D. V.** *Izvestiya Vysshikh Uchebnykh Zavedeniy. Priborostroenie*, 2015, vol. 58, iss. 5, pp. 333–343 (in Russian).
39. **Sagalovich Yu. L.** *Introduction to Algebraic Codes*, Moscow, Institut problem peredachi informacii im. A. A. Harkevicha RAN, 2010, 302 p. (in Russian).
40. **Mirzaee R. F., Daliri M. S., Navi K., Bagherzadeh N.** *Journal of multiple-valued logic and soft computing*, 2017, 29 (3–4), pp. 303–326.

**Р. Э. Асратян**, канд. техн. наук, вед. науч. сотр., e-mail: rea@ipu.ru,  
Институт проблем управления им. В. А. Трапезникова РАН

### Организация обработки защищенных сообщений в распределенных системах на основе Cryptographic Message Syntax

*Рассмотрены методы построения сетевой службы защищенных сообщений, предназначенной для реализации безопасной обработки информационных запросов в распределенных информационных системах. Отличительными особенностями службы являются тесная интеграция функций информационной защиты данных с функциями информационного взаимодействия в сети. Описаны особенности реализации службы на основе средств поддержки стандарта Cryptographic Message Syntax (CMS).*

**Ключевые слова:** распределенные системы, web-технологии, интернет-технологии, информационное взаимодействие, информационная безопасность

#### Введение

С момента своего появления сетевая архитектура .Net и технология web-сервисов заняли ведущее положение в теории и практике создания распределенных информационных систем. Решающую роль здесь сыграл целый ряд признанных достоинств этой технологии: высокая гибкость в определении сервисных функций, высокое быстродействие, эффективная поддержка в целом ряде современных систем программирования, удачные сетевые стандарты WSDL, SOAP и т. п. [1, 2]. Тем не менее, разработчики распределенных систем нередко испытывают трудности с обеспечением защиты данных в сети. Эти трудности чаще всего бывают связаны с отсутствием в архитектуре .Net встроенных средств защиты и аутентификации сетевых сообщений и более всего проявляются в разработках систем, предназначенных для работы в сложных, мультисерверных и мультисетевых средах в условиях высоких требований к информационной безопасности [3–5].

В работе [6] описана новая сетевая служба PMS (Protected Message Service), разработанная в целях преодоления вышеуказанного недостатка. Суть подхода заключается в тесной интеграции функций сетевого информационного обмена с функциями защиты и аутентификации данных. Внешне эта интеграция проявляется в том, что отмеченные функции входят в набор методов главного программного клас-

са службы — класса "Защищенное сообщение", отображающего электронный документ (информационный запрос или ответ), снабженный одной или несколькими удостоверяющими электронными цифровыми подписями (ЭЦП). В отличие, например, от технологии web-сервисов описываемая служба опирается не на модель вызова функций объектов на удаленных серверах, а на модель обмена сообщениями. В данном случае это означает, что все сервисные обрабатывающие функции (методы) имеют одинаковую жесткую спецификацию: они получают объект класса "Защищенное сообщение" в качестве параметра и возвращают объект того же класса. Эти обрабатывающие функции группируются в одну или несколько динамических библиотек, которые подключаются к серверу PMS в момент его запуска (каждая библиотека может рассматриваться как отдаленный аналог web-сервиса в .Net), и становятся доступными для клиентских компонентов.

Реализация PMS на основе криптосистемы "КриптоПро" версии 3.6 и проведенные лабораторные эксперименты показали достаточно высокое быстродействие новой службы, не уступающее, а в отдельных случаях превосходящее быстродействие web-сервисов в одинаковых условиях. Однако при данном подходе возникает жесткая "привязанность" PMS к определенной криптосистеме, что может создать неудобства для разработчиков распределенных систем.

В данной работе рассматривается новый подход к архитектурному построению PMS, основанный на применении стандарта Cryptographic Message Syntax (CMS) и его программной поддержки в среде Windows в качестве базисного средства реализации. Главное преимущество этого подхода заключается в том, что он позволяет PMS "унаследовать" способность гибкой настройки на использование любой криптосистемы, поддерживающей стандарт CMS, и тем самым устранить ту жесткую привязку к определенной криптосистеме, о которой говорилось выше.

### Краткие сведения о PMS

Как и всякая сетевая служба, основанная на базовом сетевом протоколе TCP/IP [7], PMS поддерживается клиентским и серверным программным обеспечением. Сервер PMS представляет собой постоянно активную программу, обслуживающую запросы на обработку от клиентов (по умолчанию используется порт 8132). Клиентское программное обеспечение представляет собой библиотеку функций PmsBase.dll, реализующих прикладной программный интерфейс (API) к PMS. Этот интерфейс является "лицом" PMS с точки зрения пользователя.

На рис. 1 представлен фрагмент кода в нотации C#, иллюстрирующий простое обращение к обрабатывающей функции с применением средств защиты данных. Две первые строки кода определяют две переменные типа PmsMessage, представляющего собой главный класс PMS ("Защищенное сообщение"). Первой переменной (Request) присваивается значение: объект класса PmsMessage, инициализированный символьной строкой (например, содержащей XML-документ информационного запроса). Вторая переменная

(Reply) предназначена для хранения результата обработки. В третьей строке определяется и инициализируется переменная класса PmsConnection, предназначенного для создания и прекращения сетевого соединения с сервером.

В четвертой и пятой строках кода выполняется формирование подписей в запросе. Сначала определяется переменная SenderCerts класса PmsCertList. Этот класс предназначен для хранения в памяти списков сертификатов с открытыми ключами в стандарте X509 и содержит несколько конструкторов для загрузки сертификатов из файлов или из системных хранилищ с поиском по имени владельца или по серийному номеру. В переменную SenderCerts загружаются два сертификата, соответствующих именам владельцев "Иванов" и "Петров", для последующего формирования двух ЭЦП в запросе. (Такой способ использования означает, что с этими сертификатами обязательно должны быть связаны парные им закрытые ключи, иначе формирование ЭЦП закончится неудачно.) Вызов метода AddSignatures позволяет сформировать две ЭЦП в сообщении Request.

Собственно вызов обрабатывающей функции начинается с вызова метода Connect, устанавливающего сетевое соединение с сервером с указанным сетевым именем или адресом. Далее выполняется первая сетевая операция — запрос сертификата сервера (метод GetServerCertificate) с занесением результата в переменную ReceiverCert уже знакомого нам класса PmsCertList для последующего шифрования информационного запроса. Сразу же после успешного получения сертификата сервера выполняется вызов удаленной функции с помощью применения метода Process к переменной Request с использованием все того же соединения и полученного сертификата. Предполагается, что к серверу PMS подключена библиотека MyLib.dll, содержащая код функции MyFunc. При запуске функции на сервере ей передаются значение переменной Request и опциональный строковый параметр param в качестве фактических параметров. Результат обработки заносится в переменную Reply. Подчеркнем, что шифрование запроса и дешифрование ответа выполняются автоматически в методе Process.

Последующие строки обеспечивают последовательную проверку всех ЭЦП в полученном ответе сервера (вызов метода VerifySignature) и запись в стандартный вывод сведений о подписантах.

Пример заканчивается записью результата обращения в стандартный вывод (предполага-

```
PmsMessage Request= new PmsMessage("<request> ... </request>");
PmsMessage Reply;
PmsConnection MyConn = new PmsConnection();
PmsCertList SenderCerts = PmsCertList( new string [] { "Иванов", "Петров" });
Request.AddSignatures(SenderCerts);

MyConn.Connect("MyServer");
PmsCertList ReceiverCert = MyConn.GetServerCertificate();

Reply=Request.Process (MyConn, "MyLib.MyFunc param", ReceiverCert);
if(Reply != null)
{
    string Signer;
    for(int i=0; Reply.Signatures.Length; i++)
        if(Reply.VerifySignature(i, out Signer) >= 0)
            Console.WriteLine ("Ответ подписал: " + Signer);
    Console.WriteLine (Reply.GetString());
}
else
    Console.WriteLine ("Ошибка: " + MyConn.ErrMsg);
MyConn.Disconnect();
```

Рис. 1. Пример использования PMS

ется, что результат, как и запрос, имеет форму символьной строки) и закрытием соединения с сервером с помощью метода Disconnect, так как в данном примере оно больше не нужно.

Необходимо отметить, что из фрагмента кода намеренно удалены операторы обработки исключений.

### Методы реализации PMS

На рис. 2 проиллюстрированы два архитектурных подхода к реализации PMS:

- на основе прямого подключения определенной криптосистемы к программным модулям PMS использования CMS (архитектура "PMS-Криптосистема"),
- на основе CMS (архитектура "PMS-CMS-Криптосистема") с возможностью использования различных криптосистем.

Первый подход основан на прямом вызове функций определенной криптосистемы из кода программных модулей PMS для выполнения операций формирования ЭЦП, шифрования и т. п. Очевидно, что этот подход обеспечивает наименьше "накладные расходы", но реализация PMS оказывается жестко привязана к выбранной криптосистеме.

Второй подход основан на использовании средств поддержки CMS в качестве промежуточной прослойки между программными модулями PMS и используемой криптосистемой. Главное преимущество — возможность гибкой настройки на применение любой криптосистемы, поддерживающей стандарт CMS без изменения кода программных модулей службы, которые используют универсальный программный интерфейс CMS для выполнения криптофункций.

Как видно из рис. 2, важной составной частью каждого из двух архитектурных решений является используемый сетевой протокол. Независимо от выбранного решения этот протокол организуется по следующим общим принципам:

- PMS в полной мере использует двоичную природу TCP/IP [7,8]. Взаимодействие

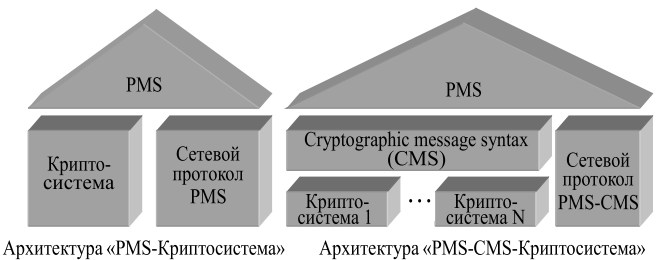


Рис. 2. Архитектурные решения для реализации PMS

- между клиентом и сервером PMS осуществляется по специальному, достаточно простому протоколу, ориентированному на передачу двоичных сетевых сообщений (PMS-сообщений) в обоих направлениях (никакие преобразования двоичных данных в текстовую форму типа base64 не применяются). Каждое такое сообщение в общем случае содержит два массива байтов: заголовок сообщения и тело сообщения (рис. 3). Первые 4 байта заголовка или тела сообщения содержат целое число — его длину;
- при передаче запроса от клиента к серверу в заголовок сетевого сообщения помещается строка, содержащая полное имя вызываемой функции, а в тело сообщения упаковывается структура PmsMessage в открытой или зашифрованной форме, содержащая информационный запрос. Строка заголовка используется сервером для организации вызова соответствующей обрабатывающей функции;
  - при передаче результата обработки от сервера к клиенту в заголовок сетевого сообщения помещается строка диагностического сообщения (значение параметра Msg, сформированное обрабатывающей функцией), а в тело сообщения упаковывается структура PmsMessage, содержащая ответ сервера в открытой или зашифрованной форме, предварительно подписанный собственным закрытым ключом сервера. Никакие двоично-текстовые преобразования (типа base64) не применяются. Полученное от сервера диагностическое сообщение автоматически присваивается члену ErrMsg объекта класса PmsSrvLibraries на стороне клиента (см. рис. 1).

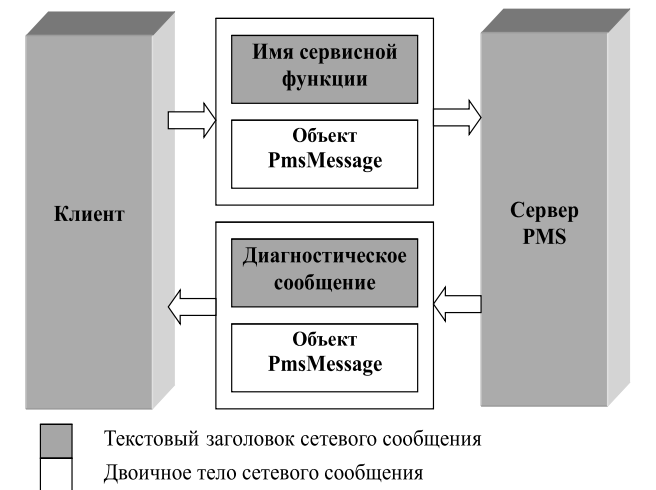


Рис. 3. Сетевой протокол Службы защищенных сообщений

Тем не менее, в архитектуре "PMS-Крипто-система" детали реализации сетевого протокола PMS могут различаться в зависимости от выбранной криптосистемы, так как от последней зависит состав и структура криптоданных, включенных в подписанный и/или зашифрованный объект PmsMessage в теле сетевого сообщения. Важнейшее преимущество архитектуры "PMS-CMS-Криптосистема" заключается в том, что в этом случае реализация сетевого протокола (протокол "PMS-CMS") не зависит от применяемых криптосистем и целиком основывается на универсальном стандарте представления защищенных данных в CMS (см. RFC 5652 в <https://tools.ietf.org/html/rfc5652>).

### Реализация PMS на базе CMS

Данный подход к реализации PMS основан на функциональном сходстве ее главного класса (PmsMessage) с главным классом CMS (SignedCms): оба класса представляют контейнер для хранения произвольных данных, оснащенный необходимыми методами для формирования и проверки электронных подписей. Вместе с тем CMS не содержит классов и методов для удаленной обработки данных в сети (аналогов класса PmsConnection или метода PmsMessage.Process). Фактически описываемый подход можно рассматривать как создание своего рода "надстройки" над CMS, направленной на сетевую обработку данных.

На рис. 4 проиллюстрировано соотношение основных классов и методов PMS и CMS, представляющее собой основу описываемого подхода. Стрелки обозначают прямой вы-

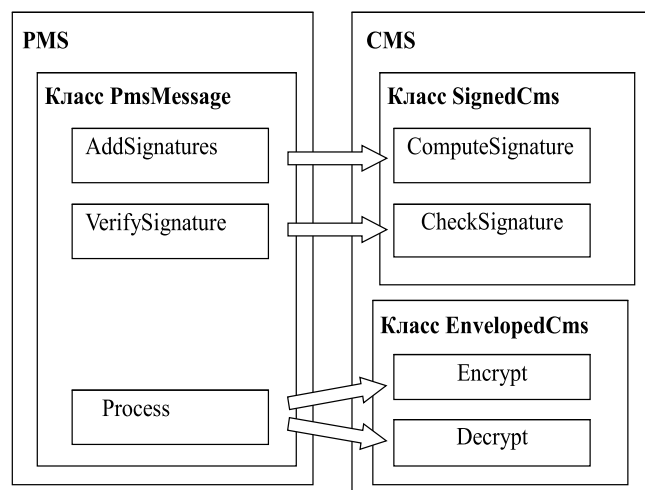


Рис. 4. Основные классы и методы PMS и CMS

зов одного метода другим. В частности, метод AddSignatures класса PmsMessage (см. рис. 1) выполняет вызов метода ComputeSignature класса SignedCms для формирования каждой ЭЦП в защищенном сообщении, а метод Process опирается в своей работе на методы класса EnvelopedCms для выполнения шифрования отправляемых в сеть данных и дешифрования данных, принятых из сети (методы Encrypt и Decrypt соответственно).

### Временные оценки

Основная цель экспериментов с PMS заключалась в сравнении быстродействия двух методов ее реализации (на основе архитектур "PMS-Криптосистема" и "PMS-CMS-Криптосистема") между собой, а также с быстродействием web-сервисов в одинаковых условиях. Главное внимание уделялось вызовам сервисных функций с относительно малым (от нескольких миллисекунд до нескольких сотен миллисекунд) временем выполнения (при более длительной обработке разница между двумя технологиями практически нивелируется) с применением средств ЭЦП и шифрования сообщений на основе криптосистемы "КриптоПро" версии 3.6, соответствующей требованиям действующих в России ГОСТ в области криптографической защиты информации. В экспериментах с PMS использовались средства криптозащиты "КриптоПро", интегрированные в клиентскую библиотеку PmsBase.dll и сервер PMS или напрямую или посредством средств CMS. В экспериментах с web-сервисами средства криптосистемы "КриптоПро" подключались непосредственно к программе клиента и программе web-сервиса. И серверы PMS с модельными библиотечными функциями, и Internet Information Server с модельными Web-сервисами были установлены на одном и том же четырехъядерном сервере приложений с тактовой частотой 2,4 ГГц в операционной среде Window 2003 Server, а в качестве клиентской рабочей станции использовался одноядерный компьютер с тактовой частотой 2,8 ГГц.

На рис. 5 показаны характерные результаты экспериментов с очень быстрой сервисной функцией, выполняющей простое перекодирование полученного строчного сообщения в верхний регистр и возврат результата клиенту, при длине сообщения в 2 Кбайт, 50 Кбайт и 100 Кбайт соответственно. На рис. 5 приведена диаграммы времен выполнения операции



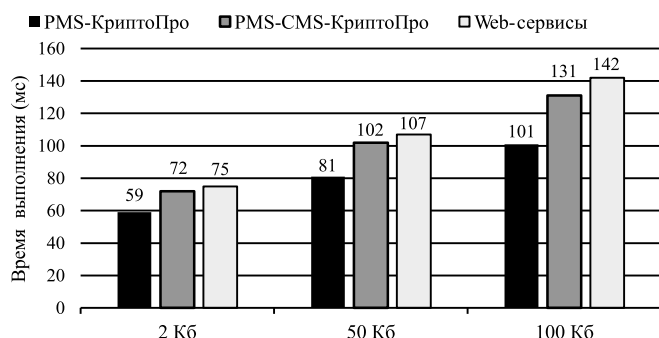


Рис. 5. Оценки быстродействия двух реализаций PMS и Web-сервиса

на сервере в реализации "PMS-КриптоПро" (черный столбик), в реализации "PMS-CMS-КриптоПро" (серый столбик) и с помощью web-сервиса (белый столбик). В каждом режиме время выполнения вычислялось как среднее значение для ста последовательных вызовов сервисной функции.

На рис. 6 и 7 (см. четвертую сторону обложки) приведены результаты экспериментов с относительно медленными сервисными функциями со временем выполнения в несколько сотен миллисекунд. Главное внимание в этой серии экспериментов уделялось сравнению быстродействия двух реализаций PMS и web-сервисов в условиях высокой нагрузки: при одновременной обработке пакетов информационных запросов и при применении криптозащиты. Скорость обработки вычислялась как частное от деления числа запросов в пакете на полное время его выполнения. Характерные кривые, приведенные на рис. 6 и 7, отражают зависимость скорости обработки от числа запросов в пакете для двух реализаций PMS и web-сервисов при обращении к модельным сервисным функциям со временами выполнения 0,5 и 1 с соответственно с применением средств криптозащиты и длине входного и выходного сообщений 50 Кбайт.

Как видно из рис. 6, 7 (см. четвертую сторону обложки), все кривые ведут себя в целом одинаково. При увеличении числа запросов в пакете растет и скорость обработки, что объясняется положительным эффектом от многопоточной обработки запросов и в ПИС и в сервере PMS. Например, при двадцати запросах в пакете и времени выполнения сервисной функции 500 мс скорость обработки достигает 15 запросов в секунду у PMS и 14 — у web-сервиса (при последовательной обработке скорость не могла бы превысить значения 2). Однако при дальнейшем увеличении размеров пакета рост скорости обработки замедляется,

а потом и вовсе останавливается вследствие достижения предельной производительности. Как видно из графиков, в этой серии экспериментов обе реализации PMS несколько превышают web-сервисы по скорости обработки, но это превышение не является значительным.

В целом результаты экспериментов позволяют сформулировать следующие выводы:

- реализация PMS на основе архитектуры "PMS-CMS-КриптоПро" в целом несколько уступает в быстродействии "прямой" реализации на основе архитектуры "PMS-КриптоПро", но в случае использования относительно медленных сервисных функций (с временем выполнения более 0,5 с) разница становится пренебрежимо малой (рис. 7, см. четвертую сторону обложки);
- обе реализации PMS не уступают в быстродействии web-сервисам;
- применение средств криптозащиты в обеих реализациях PMS не разрушает положительного эффекта от многопоточной обработки запросов;
- как и web-сервисы, обе реализации PMS вполне позволяют поддерживать скорость обработки до нескольких и даже нескольких десятков запросов в секунду даже при использовании средств криптозащиты, что обычно бывает достаточным для большинства информационных систем.

## Заключение

Главное преимущество реализации PMS на основе CMS заключается в возможности гибко настраиваться на работу с различными криптосистемами, поддерживающими этот стандарт. Отметим, что настройка осуществляется непосредственно во время выполнения в зависимости от используемых сертификатов. Например, в одном из экспериментов автор использовал на рабочих станциях самоподписанные сертификаты с криптоалгоритмами RSA, а на сервере — сертификат, сформированный удостоверяющим центром "КриптоПро" с криптоалгоритмами, соответствующими российским ГОСТ, и все взаимодействующие стороны прекрасно "понимали друг друга" (разумеется, "КриптоПро" была предустановлена на рабочих станциях и на сервере, а сертификаты обоих типов были снабжены закрытыми ключами для формирования ЭЦП). При этом смена типа используемого сертификата на сетевом узле в процессе эксперимента не создавала никаких проблем.

Вместе с тем следует отметить, что если правовые или корпоративные нормы жестко ориентируют разработчиков на использование определенной криптосистемы, то упомянутая "гибкость" оказывается невостребованной и даже нежелательной, в особенности если речь идет о проекте с высокими требованиями к информационной безопасности. Более того, даже если с технической точки зрения криптосистема удовлетворяет требования проекта (и это подтверждено надлежащими сертификатами соответствия), правомочность ее использования через посредничество средств поддержки CMS остается не вполне ясной. Поэтому в этих условиях простая архитектура "PMS-Криптосистема", основанная на прямых вызовах криптофункций, может все-таки оказаться более предпочтительной.

1. Шапошников И. В. Web-сервисы Microsoft.NET. СПб.: БХВ-Петербург, 2002. 336 с.
2. Мак-Дональд М., Шпушта М. Microsoft ASP.NET 3.5 с примерами на C# 2008 и Silverlight 2 для профессионалов. М.: Вильямс, 2009. 1408 с.
3. Згоба А. И., Маркелов Д. В., Смирнов П. И. Кибербезопасность: угрозы, вызовы, решения // Вопросы кибербезопасности. 2014. № 5. С. 30–38.
4. Щеглов А. В. Защита компьютерной информации от несанкционированного доступа. СПб.: Наука и техника, 2004. 384 с.
5. Козлов А. Д., Орлов В. Л. Методы и средства обеспечения информационной безопасности распределенных корпоративных систем. М.: ИПУ РАН, 2017. 156 с.
6. Асратян Р. Э. Интернет-служба защищенной обработки информационных запросов в распределенных системах // Программная инженерия. 2016. № 11. С. 490–497.
7. Снейдер Й. Эффективное программирование TCP/IP. Библиотека программиста. СПб.: Символ-Плюс, 2002. 320 с.
8. Хант К. TCP/IP. Сетевое администрирование. СПб.: Питер, 2007. 816 с.

R. E. Asratian, Leading Researcher, rea@ipu.ru,  
V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences,  
Moscow, 117997, Russian Federation

## Protected Message Processing in Distributed Systems on the Basis of Cryptographic Message Syntax

*The principles of the implementation of the new network service — Protected Message Service (PMS) — intended for protected queries processing in the distributed information systems are considered. Distinctive feature of PMS is the close integration of authentication and data protection functions with functions of network information exchange and data processing. From the client point of view, the service architecture is based on two main program classes: "Protected message" (PmsMessage) and "Network Connection" (PmsConnection). This classes offer necessary functionality not only for creating and protecting messages, but also for transferring them to remote server via established network connections for processing. The essence of the approach consists in using Cryptographic Message Syntax (CMS) standard as a basis of protected data representation in the network. This approach to the implementation of PMS is based on the functional similarity of its main class (PmsMessage) with the main class CMS (SignedCms): both classes represent a container for storing arbitrary data, equipped with the necessary methods for the formation and verification of electronic signatures. However, CMS does not contain classes and methods for remote data processing in the network (analogs of PmsConnection class or PmsMessage.Process method). Actually, the described approach can be considered as creation of some kind of "superstructure" over CMS directed to network data processing. The experimental implementation of PMS over CMS in C# for Microsoft Framework 4.0 and study of performance of new service were carried out. The results of this study (in comparison with web services in .NET architecture and with "direct" PMS implementation without CMS) are presented.*

**Keywords:** distributed systems, Web-technologies, Internet-technologies, network interactions, data security

DOI: 10.17587/it.25.435-440

### References

1. Shaposhnikov I. V. Web-servisy Microsoft.NET (Web-services of Microsoft.NET), SPb, BHV-Peterburg, 2002, 336 p. (in Russian).
2. MacDonald M., Szpuszta M. Microsoft ASP.NET 3.5 s primerami na C# 2008 i Silverlight 2 dlja professionalov (Pro Microsoft ASP.NET 3.5 in C# 2008 includes Silverlight 2), Moscow, Viljams, 2009, 1408 p. (in Russian).
3. Zgoba A. I., Markelov D. V., Smirnov P. I. Kiberbezopasnost: ugrozy, vyzovy, reshenija (Cybersafety: threats challenges, decisions), Voprosy Kiberbezopasnosti, 2014, no. 5, pp. 30–38 (in Russian).
4. Shheglov A. V. Zashhita komp'yuternoj informacii ot nesankcionirovannogo dostupa (Protection of computer information against illegal access), SPb.: Nauka i Tehnika, 2004, 384 p. (in Russian).
5. Kozlov A. D., Orlov V. L. Metody i sredstva obespechenija informacionnoj bezopasnosti raspredelennyh korporativnyh sistem (Methods and means of ensuring of information security in distributed enterprise systems), Moscow, IPU RAN, 2017, 156 p. (in Russian).
6. Asratian R. E., Internet-sluzhba zashhishhennoj obrabotki informacionnyh zaprosov v raspredelennyh sistemah (Internet service for protected information queries processing in distributed systems), Programmnaja Inzhenerija, 2016, no. 11, pp. 490–497 (in Russian).
7. Snader J. Effektivnoe programmirovaniye TCP/IP (Effective TCP/IP programming), SPb, Simvol-Pljus, 2002, 320 p. (in Russian).
8. Hunt C. TCP/IP. Setevoe administrirovaniye (TCP/IP Network administration), SPb, Piter, 2007, 816 p. (in Russian).

**А. Б. Барский**, д-р техн. наук, проф., e-mail: arkbarsk@mail.ru,  
**Д. И. Мельник**, канд. техн. наук, ст. науч. сотр., e-mail: mdi\_dim@mail.ru,  
НИИЦ (г. Москва) ЦНИИ ВВКО Минобороны России

### Нейросетевая модель целераспределения для вычислительной системы архитектуры *data flow*

*Предлагается метод применения логической нейронной сети для решения задачи целераспределения. Предполагается, что с помощью модели обороняемого объекта или района проведена пристрелка стрельбовых комплексов по опорным траекториям и построена база знаний (БЗ), представленная логической нейронной сетью. Значения коэффициентов опорных траекторий, а также значения готовности стрельбовых комплексов связаны в БЗ с решениями о назначении этих комплексов для поражения цели. Приводится программа выполнения основных действий нейрокомпьютера на вычислительной системе архитектуры data flow. Предложен способ снижения трудоемкости вычисления скалярного произведения сильно разреженных (нулями) векторов, являющегося базовой операцией при вычислении значения функции активации нейронов.*

**Ключевые слова:** целераспределение, опорная траектория, стрельбовый комплекс, логическая нейронная сеть, скалярное произведение векторов, архитектура data flow

#### Введение

С начала 80-х годов прошлого века перспективные исследования в области вычислительной техники во многом определялись концепцией "японского вызова". В рамках этой концепции были обоснованы два положения:

- должны быть созданы нейросетевые методы вычислений часто решаемых оптимизационных задач высокой сложности;
- вычислительная система (ВС) должна объединять десятки тысяч микропроцессоров, взаимодействующих по технологии *data flow*.

Первое положение должно было обеспечить рост эквивалентной производительности ВС за счет реализации более простых алгоритмов решения нейросетевых и нейроподобных задач.

Второе положение должно было обеспечить параллельное решение этих задач с минимальными накладными расходами в высокопараллельной ВС, способной в том числе воспроизводить структуру нейронной сети. В этом случае реальная производительность нейрокомпьютера эквивалентна той производительности, что требуется для решения конкретной задачи численными методами.

Несмотря на невозможность полного воплощения фантастических идей, они до сих пор оказывают серьезное влияние на попытки достижения сверхвысокой производительности вычислительных средств, использующихся в сложных системах обороны [1, 2].

Одной из центральных задач, решаемых в начале боевого цикла системы обороны, является задача целераспределения, к оперативному решению которой предъявляются наиболее высокие требования [3, 4]. В классической постановке это транспортная задача линейного программирования с рядом дополнительных условий и ограничений. Такая задача высокой сложности требует применения методов искусственного интеллекта. Однако попытки использования "классических" нейронных сетей так и не привели к успеху. Разработка логических нейронных сетей [5], в основе которых лежит математическая логика событий, а также принцип ассоциативного мышления человека, послужила основой оперативного решения многих задач распознавания, управления и принятия решений.

Возможность применения логических нейронных сетей для оперативного решения за-

дачи целераспределения на принципиальном уровне показана в настоящей статье. Приводится программа центрального модуля этой задачи на ВС архитектуры *data flow* [6].

## 1. Задача целераспределения

Пусть территориально распределенная система обороны включает станции обнаружения и несколько стрельбовых комплексов, обладающих зонами поражения. Станции обнаружения с помощью своих вычислительных средств, используя, например, метод наименьших квадратов при обработке группы обнаруженных точек, формируют полиномы целей как функции времени  $t$  для расчета их текущих координат и параметров движения. Обычно это полиномы четвертой степени вида

$$\begin{aligned}x &= a_{01} + a_{11}t + a_{21}t^2 + a_{31}t^3 + a_{41}t^4; \\y &= a_{02} + a_{12}t + a_{22}t^2 + a_{32}t^3 + a_{42}t^4; \\z &= a_{03} + a_{13}t + a_{23}t^2 + a_{33}t^3 + a_{43}t^4.\end{aligned}$$

Представленные полиномы могут описывать центр групповой цели. По результатам дальнейших наблюдений эти полиномы уточняются.

Таким образом, данные полиномы описывают модель движущейся цели, пересекающей зоны поражения и в разной степени достижимой средствами перехвата стрельбовых комплексов с учетом их готовности.

Эффективность системы целераспределения характеризуется:

- минимальным предотвращенным ущербом;
- временем готовности стрельбового комплекса к выполнению новой задачи;
- временем подготовки пуска перехватчика с момента выдачи целеуказания при предпочтительности дальнего перехвата;
- коэффициентом загрузки стрельбового комплекса по групповой цели;
- параметрами взаимного пространственного расположения цели и перехватчика в момент выдачи главной команды и пр.

Известны попытки решения задачи целераспределения в статическом режиме для отражения единичной или групповой цели как транспортной задачи линейного программирования высокой сложности. Однако следует ориентироваться на динамический характер отражения многих целей массированного налета в течение относительно длительных боевых действий,

при которых необходимо учитывать ущерб, нанесенный прорвавшимися целями.

Тогда задача целераспределения превращается в трудно формализуемую логико-комбинаторную оптимизационную задачу, оперативное решение которой основано на переборе и анализе большого числа связей вида "если — то". Для решения подобных задач используется аппарат логических нейронных сетей [5], имитирующий ассоциативное мышление человека на основе накопленного опыта и обучения. Решение задачи, во многом воспроизводящее артиллерийские приемы накопления опыта пристрелки и переноса огня, строится на основе экспертных оценок, на базе натурных экспериментов и результатов моделирования. По результатам таких исследований создается база знаний (БЗ), отображающая связи между возможными ситуациями и принимаемыми по ним решениями. При формировании этих связей (при обучении нейронной сети<sup>1</sup>) учитываются требования эффективности решений. Ситуации задаются возбуждением рецепторов. Возбуждение нейронов, реализующих пороговую функцию активации, указывает на рекомендуемое назначение стрельбового комплекса. При нечисловом характере вывода решение принимается по наиболее "возбуждавшемуся" нейрону, что и свойственно задаче целераспределения.

Наполнение базы знаний возможно по принципу очевидной целесообразности при оценке взаимного расположения объектов и возможных характеристик налета. Сложные случаи требуют создания *комплексной имитационной модели* с применением средств анимации и компьютерной графики. Модель должна обеспечить возможность имитации различных сценариев действий противника для формирования различных вариантов налета. Она должна отображать различные способы противодействия не только в поиске успешных, но и оптимальных в смысле предъявляемых требований, описанных выше. При этом модель должна адекватно воспроизводить физические законы движения управляемых объектов и целей. Задачей моделирования является покрытие факторного пространства отношениями вида "если — то" ("параметры удара — выбор средства отражения") для решения задачи целераспределения в рамках более общей задачи управления боевыми действиями.

<sup>1</sup>Логическая нейронная сеть создается обученной. В этом ее преимущество перед "классическими" нейронными сетями.

## 2. Логическая нейронная сеть для решения задачи целераспределения

Пусть на основе моделирования построены  $M$  опорных траекторий, равномерно или с учетом опасных направлений покрывающих обороняемую территорию. Выбраны оптимальные варианты их "обслуживания" основными стрельбовыми комплексами (СК) и их дублерами (в случае низкой готовности или выхода из строя основных средств).

Составим примерную, легко развиваемую логическую нейронную сеть (рис. 1), позволяющую принять решение о назначении главного СК и его дублера по значениям коэффициентов полиномов цели. Рассматривая возможности целераспределения на принципиальном уровне, допустим, что двух коэффициентов в уравнениях каждой координаты движения достаточно для представления о характере цели. Иными словами, по этим коэффициентам можно с достаточной точностью определить опорную траекторию, на которую в большой степени "похожа" анализируемая траектория, подлежащая "обслуживанию".

Кроме того, для лучшего понимания принципа назначения СК для поражения цели на рис. 1 отражен ряд допущений.

1. Предполагается, что значения коэффициентов  $a_{01}^1, a_{01}^2, \dots, a_{01}^M$  по координате  $x$  для  $M$  опорных траекторий образуют возрастающую последовательность, и за каждым из них закрепляется рецептор. В действительности этот порядок следования может быть нарушен. Если же для двух или более из  $M$  траекторий эти коэффициенты равны, то их значение представляет лишь один рецептор [1].

2. Значения коэффициентов  $a_{11}^1, a_{11}^2, \dots, a_{11}^M$  по координате  $x$  для всех опорных траекторий тоже образуют возрастающую последовательность отличных друг от друга значений. В ней повторяется тот же порядок представления  $M$  траекторий, что и в последовательности  $a_{01}^1, \dots, a_{01}^M$ . В действительности возрастающая последовательность этих коэффициентов в общем слу-

чае отображает другой порядок представления траекторий, а одинаковые значения коэффициентов определяют другое число рецепторов, закрепленных за вторым коэффициентом в уравнении по координате  $x$ .

3. То же касается последовательности коэффициентов  $a_{02}^1, a_{02}^2, \dots, a_{02}^M$  и последовательности  $a_{12}^1, a_{12}^2, \dots, a_{12}^M$  по координате  $y$ , а также последовательности  $a_{03}^1, a_{03}^2, \dots, a_{03}^M$  и последовательности  $a_{13}^1, a_{13}^2, \dots, a_{13}^M$  по координате  $z$ .

Таким образом, в общем случае коэффициенты одной опорной траектории оказываются независимо рассредоточены по возрастающим последовательностям, а общее число рецепторов, закрепленных за значениями коэффициентов, лишь не превышает значение  $6M$ . Однако связи с нейронами должны строго соот-

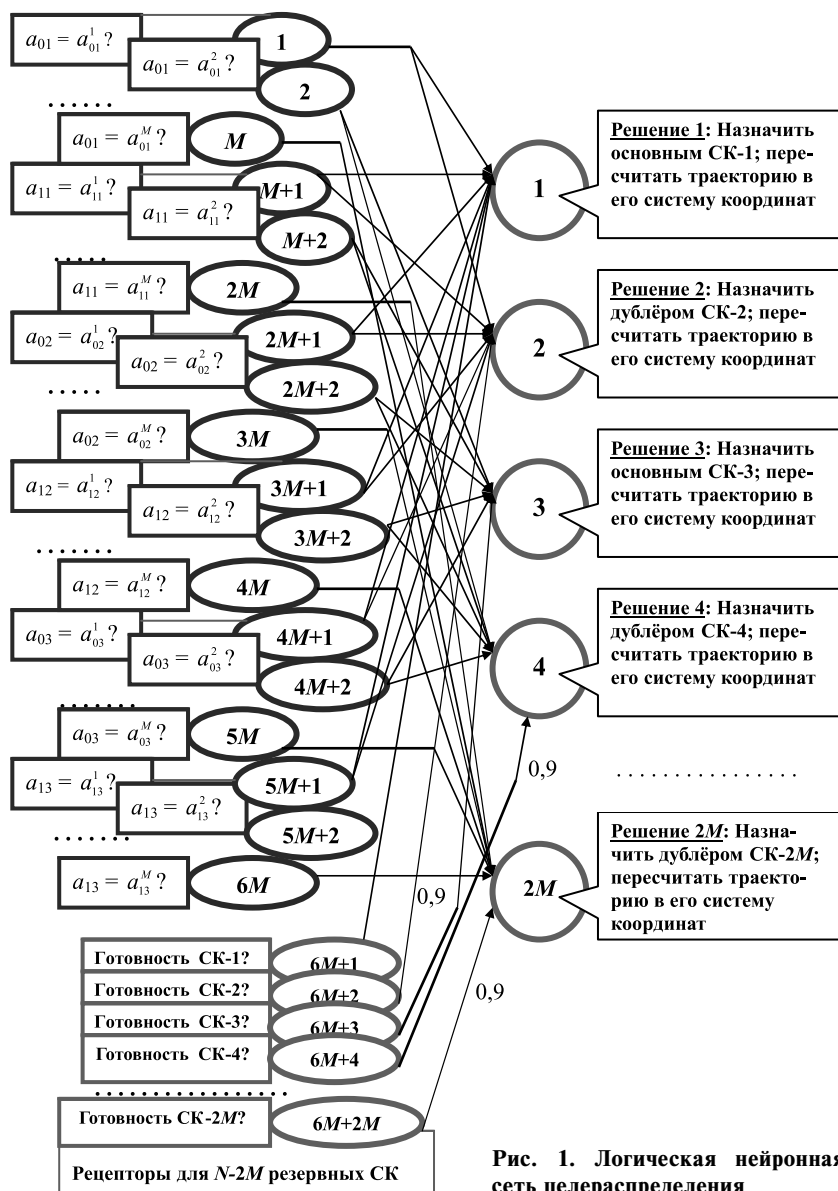


Рис. 1. Логическая нейронная сеть целераспределения

ветствовать принадлежности коэффициентов опорным траекториям, и из одного рецептора может исходить более одной связи.

4. Порядок нумерации стрельбовых комплексов и их оптимальное закрепление за опорными траекториями выбран так, чтобы, как и в пунктах 1—3, не в ущерб общности примера упростить индексацию, обеспечить наглядность и не запутать читателя нагромождением "правильных" абстрактных выкладок.

5. Стрельбовых комплексов, с учетом назначения основных и дублеров, хватает для "обслуживания"  $M$  опорных траекторий и близких к ним реальных траекторий, т. е.  $N \geq 2M$ .

6. В данном примере предполагается назначение основного и дублирующего СК для поражения одной цели. Однако в действительности на особенно опасном направлении может быть предусмотрено закрепление за опорными траекториями более одного СК-дублера. Тогда по числу назначаемых средств поражения формируется и число нейронов, закрепляемых за соответствующими опорными траекториями, а число нейронов в логической нейронной сети может превышать значение  $2M$ . Веса связей от дублирующих СК образуют убывающую последовательность, указывающую на приоритет дублеров. Так повышается вероятность назначения СК-дублера только в том случае, когда основной СК и СК-дублеры более высокого приоритета обладают более низкой готовностью: временно заняты, исчерпали боекомплект или поражены.

Функция активации нейрона имеет вид

$$V_i = \begin{cases} \left( \left( \sum_{\{j\}} \omega_j v_j \right) / \left( \sum_{\{j\}} \omega_j \right) \right), & \text{если это значение} \\ \text{не менее порога } h; \\ 0 & \text{в противном случае.} \end{cases}$$

Здесь  $V_i$  — возбуждение  $i$ -го нейрона;  $\{j\}$  — множество связей  $i$ -го нейрона с рецепторами;  $\omega_j$  — вес  $j$ -й связи нейрона;  $v_j$  — возбуждение  $j$ -го рецептора. Порог  $h$  выбирается экспериментально так, чтобы не "погасить" возбуждение нейронов, обусловленное СК-дублерами при допустимом значении их готовности.

Напомним [5], как данная логическая сеть работает в составе системы целераспределения.

Обращение к логической нейронной сети для обработки каждой новой траектории проводится отдельно, т. е. множество новых траекторий обрабатывается последовательно в цикле.

Если значение коэффициента  $a_{01}$  совпадает с одним из значений этого коэффициента, за

которым закреплен рецептор, значение возбуждения этого рецептора полагается равным единице. Если значение  $a_{01}$  не совпадает ни с одним значением, за которым закреплен рецептор, находятся значения  $a_{01}^i$  и  $a_{01}^{i+1}$ , за которыми существуют закрепленные рецепторы<sup>2</sup> и для которых выполняется условие:  $a_{01}^i < a_{01} < a_{01}^{i+1}$ . Значение возбуждения рецептора  $i$  полагается равным  $(a_{01}^{i+1} - a_{01}) / (a_{01}^{i+1} - a_{01}^i)$ . Значение возбуждения рецептора  $i + 1$  дополняет найденное значение до единицы.

Аналогично "возбуждаются" один или два рецептора по значению коэффициента  $a_{11}$ . Таким же образом "возбуждаются" рецепторы по уравнениям координат  $y$  и  $z$ .

Оценки готовности стрельбовых комплексов формируются на основе функционального контроля и системы управления боевыми действиями.

По сформированным значениям возбуждения рецепторов с помощью функции активации рассчитываются значения возбуждения всех нейронов. Максимально "возбуждившийся" нейрон указывает на принимаемое решение. Оценка готовности используемого СК корректируется.

Если возбуждение ни одного нейрона не превысило порог, следует сделать вывод об ожидании освободившегося СК с высокой готовностью или об оперативном подключении других средств противодействия. Снижение порога не приведет к какому-либо пересмотру действий по "обработке" данной цели.

Обратим внимание на важный элемент самообучения, реализуемый в ходе боевых действий. Траектории некоторых пораженных целей можно включать в опорные, тем самым развивая БЗ и ее логическую нейронную сеть. (Это действие подобно тому, как в артиллерии команда "Стой! Записать..." в дальнейшем служит эффективному применению тактики переноса огня вместо трудоемкой подготовки данных по новым целям.)

Наглядное изображение логической нейронной сети служит отражению смысла задачи, но не предоставляет формальных механизмов для компьютерной реализации. Метод вычислений основан на применении *матрицы связей* [5], соответствующей этой сети. Матрица связей и некоторая сопутствующая информация, ускоряющая расчет, представлена в таблице.

<sup>2</sup> Если одного из таких значений нет, принимается простое соглашение, например, об использовании "среднего расстояния между рецепторами" для возбуждения рецептора, оказавшегося "крайним".

### Матрица связей

| Нейроны,<br>указыва-<br>ющие на<br>решения | Рецепторы и места для значения их возбуждения |   |     |     |         |         |     |      |          |     |          |          |          |          |     | Списки индексов<br>связей с ненулевыми<br>весами |                                                     |
|--------------------------------------------|-----------------------------------------------|---|-----|-----|---------|---------|-----|------|----------|-----|----------|----------|----------|----------|-----|--------------------------------------------------|-----------------------------------------------------|
|                                            | 1                                             | 2 | ... | $M$ | $M + 1$ | $M + 2$ | ... | $2M$ | $2M + 1$ | ... | $6M + 1$ | $6M + 2$ | $6M + 3$ | $6M + 4$ | ... |                                                  | $1/(\Sigma \text{ весов})$                          |
| Нейрон 1,<br>Решение 1                     | 1                                             |   |     |     | 1       |         |     |      | 1        |     | 1        |          |          |          |     | $7^{-1}$                                         | $S_1 = \{1, M + 1, 2M + 1, \dots, 5M + 1, 6M + 1\}$ |
| Нейрон 2,<br>Решение 2                     | 1                                             |   |     |     | 1       |         |     |      | 1        |     |          | 0,9      |          |          |     | $6,9^{-1}$                                       | $S_2 = \{1, M + 1, 2M + 1, \dots, 5M + 1, 6M + 2\}$ |
| Нейрон 3,<br>Решение 3                     |                                               | 1 |     |     |         | 1       |     |      |          |     |          |          | 1        |          |     | $7^{-1}$                                         | $S_3 = \{2, M + 2, 2M + 2, \dots, 5M + 2, 6M + 3\}$ |
| Нейрон 4,<br>Решение 4                     |                                               | 1 |     |     |         | 1       |     |      |          |     |          |          |          | 0,9      |     | $6,9^{-1}$                                       | $S_4 = \{2, M + 2, 2M + 2, \dots, 5M + 2, 6M + 4\}$ |
| .....                                      |                                               |   |     |     |         |         |     |      |          |     |          |          |          |          |     |                                                  | .....                                               |
| Нейрон $2M$ ,<br>Решение $2M$              |                                               |   |     | 1   |         |         |     | 1    |          |     |          |          |          |          |     | $6,9^{-1}$                                       | $S_{2M} = \{M, 2M, \dots, 6M, 6M + 2M\}$            |

При построении матрицы связей для каждого нейрона целесообразно формировать список индексов ненулевых весов связей с рецепторами. Это необходимо для ускоренного выполнения операции скалярного произведения разреженных (нулями) векторов как основной трудоемкой операции счета значения функции активации. Как видно из рис. 1 и таблицы, в каждом скалярном произведении участвуют всего лишь семь элементов вектора весов каждого нейрона, какими бы большими ни были значения  $M$  и  $N$ .

### 3. Решение задачи целераспределения на проектируемой ВС архитектуры *data flow*

В работе [6] содержится достаточно примеров программирования на ВС, управляемой потоком данных по технологии *data flow*. Однако каждая новая область применения порождает и новые рекомендации по системе команд.

Отметим, что каждый разработчик параллельных вычислительных систем утверждает, что он реализовал архитектуру *data flow*. Дело в том, что какими бы приемами не осуществлялось распараллеливание вычислений, оно всегда основано на оценке готовности данных для программных модулей, процедур, инструкций и пр., распределяемых между исполнительными устройствами. Распределение, как правило, выполняется принудительно с помощью диспетчеров ОС или аппаратными средствами, приводящими к серьезным накладным расходам производительности. Рассматриваемый здесь "чистый" принцип *data flow* отличается тем, что данные, являющиеся операндами в некоторой операции, по мере готовности поступают непосредственно в *текст*

инструкции для выполнения этой операции. Операции, для которых в текст соответствующей инструкции поступили все необходимые данные, готовы к выполнению и могут быть "подхвачены" свободными ресурсами.

Составим *план загрузки* множества  $R = \{r_1, \dots, r_{2m-2}\}$  виртуальных вычислителей решающего поля ВС для счета значения скалярного произведения двух векторов  $A = \{a_1, a_2, \dots, a_m\}$  и  $B = \{b_1, b_2, \dots, b_m\}$  способом "пирамиды". Этот способ обеспечивает максимальные возможности распараллеливания таких связанных процессов, как "свертка массивов" или преобразование вида "вектор—скаляр". Для определенности положим  $m = 7$  (рис. 2). В овалах

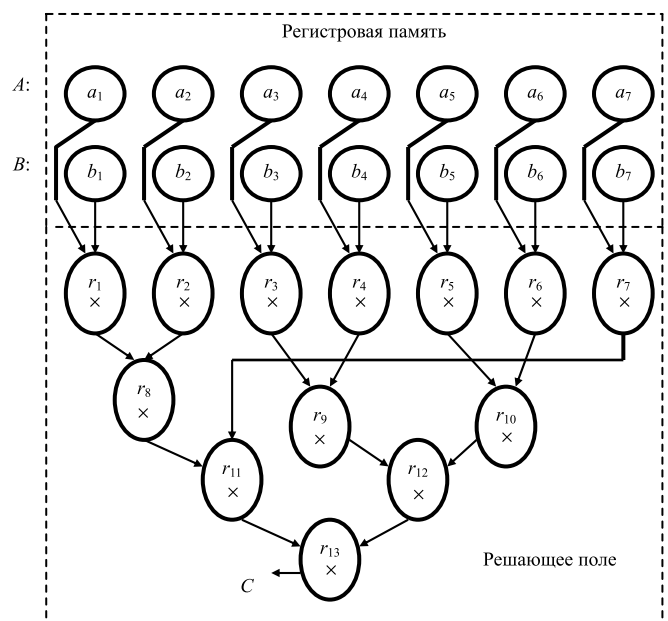


Рис. 2. План загрузки вычислителей ВС *data flow* для счета скалярного произведения векторов



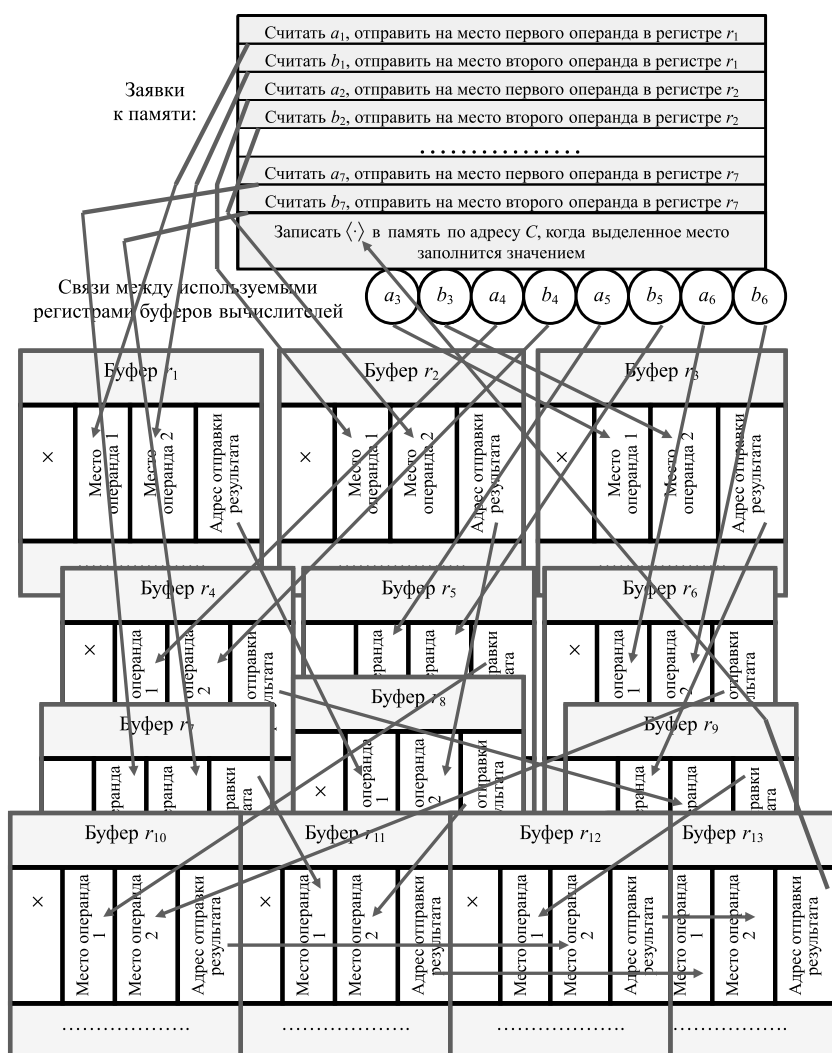


Рис. 3. Коммутация вычислителей решающего поля

указаны виртуальные адреса вычислителей и выполняемые ими операции.

Для счета по данной схеме динамически формируются заявки к памяти, а также занимаются и частично заполняются инструкциями очередные свободные регистры в буферах физических вычислителей. Так создается *схема коммутации* вычислителей для выполнения данного алгоритма (рис. 3). Для краткости изложения предположим, что виртуальные адреса вычислителей на рис. 2 автоматически переведены в физические с сохранением их значения. Место временного хранения результата в регистре не показано.

Представим программу нахождения решения о назначении СК для поражения цели с помощью логической нейронной сети. Фактически это программа стандартной процедуры НЕЙРОКОМПЬЮТЕР для ВС архитектуры *data flow* (рис. 4).

Поясним применение некоторых команд.

Команда ЦИКЛ1 выполняет простейший цикл типа арифметической прогрессии.

Команда ЦИКЛ2 соответствует конструкции, применяющей список индексов и даже выражений для их расчета в полном описании ALGOL-60 Бэкуса—Бауэра [7]. В соответствии с этим обобщением предполагается непосредственно использовать список индексов элементов одного из двух векторов, отличных от нуля, для сокращения времени счета их скалярного произведения. Первый индекс из списка выбирается при выполнении команды ЦИКЛ2. Остальные индексы формируются по команде КЦ2 (конец цикла 2). Предполагается, что список оканчивается либо пустым словом, либо специальным символом, выбор которого служит сигналом выхода из цикла. Это следует учитывать при формировании таблицы, где размеры всех ее позиций должны быть фиксированными в соответствии с целью применения. Это упрощает переадресацию. Более того, условия оперативного применения процедуры НЕЙРОКОМПЬЮТЕР требуют использования сверхоперативной, т. е. регистровой памяти.

Дважды применяемая команда УСЛ реализует конструкцию "if — then — else" [6] в двух модификациях. По команде 16 сравнение с порогом  $h$  позволяет окончательно сформировать значение пороговой функции активации, т. е. получить значение возбуждения нейрона: если  $h \leq ((M3) + r_1)$ , с помощью пятого адреса подтверждается рассчитанное значение, в противном случае значение этой функции равно нулю.

В команде 17 используется модификация, при которой также по третьему адресу направляются данные пятого адреса, если значение по первому адресу меньше значения по второму адресу. В противном случае по третьему адресу отправляются данные четвертого адреса. Кроме того, если пересылка была проведена с пятого адреса, выполняется следующая команда 18. Такое выполнение команд 17 и 18 позволило зафиксировать новое значение максимально

| №к | КОП                 | $I_1$ | $A_1$                      | $I_2$ | $A_2$                 | $I_3$ | $A_3$      | Пояснения                                                                                            |
|----|---------------------|-------|----------------------------|-------|-----------------------|-------|------------|------------------------------------------------------------------------------------------------------|
| 0  | ЗАГ                 | $M1$  |                            | $M2$  |                       | $M3$  |            | Очистка модификаторов                                                                                |
| 1  | ЗАГ                 | $M4$  | $\langle P \rangle$        |       |                       |       |            | $P$ – число нейронов, $P \geq 2M$                                                                    |
| 2  | ЗАП                 |       |                            |       |                       |       | $V_{\max}$ | Очистка ячейки для накопления $V_{\max}$                                                             |
| 3  | ЦИКЛ1               | $M4$  |                            |       |                       |       |            | Цикл по нейронам                                                                                     |
| 4  | ЦИКЛ2               | $M1$  |                            | $M2$  | $S_1$                 |       |            | Выборка первого индекса строки                                                                       |
| 5  | $\times$            | $M1$  | $\langle \omega_1 \rangle$ | $M1$  | $\langle v_1 \rangle$ | $M3$  | $r_1$      | Очередное умножение по индексу                                                                       |
| 6  | $I+$                | $M3$  | 0..01                      |       |                       |       |            | Переадресация вычислителя                                                                            |
| 7  | КЦ2                 | $M1$  |                            | $M2$  | $S_1$                 |       |            | Выборка следующего индекса, выход                                                                    |
| 8  | $I+$                | $M2$  | $\langle D \rangle$        |       |                       |       |            | Переадресация списка                                                                                 |
| 9  | ЗАГ                 | $M5$  | $M3$                       | $M1$  |                       |       |            | Число вычислителей и очистка                                                                         |
| 10 | $\times_{\text{ц}}$ |       | $M5$                       |       | 2                     |       | $M5$       | Уточнение числа вычислителей                                                                         |
| 11 | ЦИКЛ1               | $M5$  | 7...7                      |       |                       |       |            | Цикл на $(M5)-1$ повторений                                                                          |
| 12 | $+$                 | $M1$  | $r_1$                      | $M1$  | $r_2$                 | $M3$  | $r_1$      | Сложение по схеме «пирамида»                                                                         |
| 13 | $I+$                | $M1$  | 0..02                      | $M3$  | 0..01                 |       |            | Переадресация по схеме «пирамида»                                                                    |
| 14 | КЦ1                 |       |                            |       |                       |       |            | Конец цикла                                                                                          |
| 15 | $\times$            | $M3$  | $r_1$                      | $M2$  | $S_2$                 | $M3$  | $r_1$      | Окончание счёта функции активации                                                                    |
| 16 | УСЛ                 |       | $\langle h \rangle$        | $M3$  | $r_1$                 | $M3$  | $r_1$      | Сравнение с $h$ в соответствии со счётом значения пороговой функции                                  |
| 17 | УСЛ                 |       | $V_{\max}$                 | $M3$  | $r_1$                 |       | $V_{\max}$ | Модификация команды УСЛ с выполнением следующей команды, если запись была произведена из 5-го адреса |
| 18 | ЗАП                 |       | $M4$                       |       |                       |       | $N_{\max}$ | Засылка номера максимально «возбуждённого» нейрона                                                   |
| 19 | $I+$                | $M2$  | $l$                        |       |                       |       |            | Переадресация на строку следующего нейрона                                                           |
| 20 | КЦ1                 |       |                            |       |                       |       |            |                                                                                                      |

Рис. 4. Программа НЕЙРОКОМПЬЮТЕР для назначения стрельбового комплекса

"возбуждённого" нейрона, а при выполнении следующей команды — получить значение параметра цикла, т. е. номер этого нейрона. Если пересылка была проведена с четвертого адреса, следующая команда пропускается.

### Заключение

1. На основе аппарата логических нейронных сетей построен "быстрый" алгоритм целераспределения. Логическая нейронная сеть интерпретирует базу знаний, в которой отражено соответствие между наиболее опасными опорными траекториями движения целей, "покрывающими" обороняемый район, и средствами их эффективного поражения. Назначение средств поражения для вновь обнаруженной траектории цели проводится по принципу ассоциативных вычислений, по наибольшей схожести данной траектории на одну из опорных. Множество опорных траекторий и принимаемые по ним решения выбираются на основе очевидных соображений по размещению объектов и средств, а также по результатам моделирования боевых действий. При закреплении стрельбовых комплексов за опорными траекториями (при обучении логической нейронной сети) учитываются

требования эффективности системы целераспределения. Избранные траектории пораженных целей могут включаться в состав опорных, развивающих самообучающуюся, таким образом, базу знаний. В возможности самообучения заключается преимущество такого метода искусственного интеллекта, как аппарат логических нейронных сетей.

2. Вычислительные системы, управляемые потоком данных (*data flow*), определяют высокий коэффициент полезной загрузки оборудования при решении задачи целераспределения нейросетевым методом. Пример решения данной задачи демонстрирует возможность необходимого наращивания мощности таких систем структурными методами при соблюдении требований стандартизации, унификации и высокой надежности. Накладные расходы на управление вычислительной системой при рассмотренном способе распараллеливания практически отсутствуют.

3. Представленная программа нейрокомпьютера на базе логической нейронной сети является универсальной процедурой, которая может быть включена в состав общего программного обеспечения вычислительной системы архитектуры *data flow*.

4. Для повышения эффективности процедуры НЕЙРОКОМПЬЮТЕР многопроцессорная система архитектуры *data flow* должна воплощать принцип *SPMD*-технологии, где на всех процессорах выполняются копии одной программы по распределенным данным при простейшей синхронизации по общим данным. Применительно к рассмотренной задаче реализуется план вычислений, при котором строки нейронов в матрице связей динамически закрепляются за процессорами в соответствии с номерами тех и других. Каждый процессор находит среди "своих" нейронов максимально "возбуждённый". Абсолютный максимум находится на процессоре, помимо того выполняющем функции головного. Он и назначает стрельбовый комплекс для перехвата цели.

### Список литературы

1. Бурцев В. С. Система массового параллелизма с автоматическим распределением аппаратных средств супер-

ЭВМ в процессе решения задачи // Юбилейный сборник трудов институтов ОИВТА РАН. Т. II. М.: ОИВТА РАН, 1993. С. 5–27.

2. Змеев Д. Н., Климов А. В., Левченко Н. Н., Окунев А. С., Стемповский А. Л. Потокная модель вычислений как парадигма программирования будущего // Информатика и ее применение. 2015. Т. 9, Вып. 4. С. 29–36.

3. Трошин Г. И. Григорий Васильевич Кисунко — основоположник противоракетной обороны СССР, выдающийся радиофизик XX века, писатель, поэт (научная биография). М.: "Новые технологии". 2015. 220 с.

4. Третьяков Ю. Н. 50 лет в ракетно-космической обороне / Под ред. О. Ю. Аксенова. М.: ООО "ТИПОГРАФИЯ КЕМ", 2010. 800 с.

5. Барский А. Б. Нейросетевые методы оптимизации решений. СПб.: ИЦ "Интермедия", 2016. 312 с.

6. Барский А. Б., Шилов В. В. Потокная вычислительная система: программирование и оценка эффективности // Информационные технологии. № 7. Приложение. 2003. 24 с.

7. Бэкус Д. Б., Бауэр Ф. А. и др. Сообщение об алгоритмическом языке АЛГОЛ-60 // Журнал вычислительной математики и математической физики. 1961. Т. 1, № 2. С. 308–342.

A. B. Barsky, D. Tech. Sc., Professor, e-mail: [arkbarsk@mail.ru](mailto:arkbarsk@mail.ru),

D. I. Melnik, Ph. D., Senior Researcher, e-mail: [mdi\\_dim@mail.ru](mailto:mdi_dim@mail.ru),

Scientific Research Institute (Moscow) Central Research Institute VVKO, Ministry of Defense of Russia

## Neural Network Target Distribution Model for Computing System of Data Flow Architecture

*A method of applying a logical neural network for solving the task of target distribution is proposed. The trajectory of the target for each coordinate is described by polynomials. It is assumed that, using a model of the object to be defended, the shooting complexes were fired at reference trajectories, and a knowledge base representing the logical neural network was built. It connects the values of the coefficients of the reference trajectories and the readiness values of the firing complexes with the decisions on their appointment to hit the target. To perform the associative sampling procedure, the receptor excitation values are set. The most "energized" neuron as a result of the activation function count indicates the selected shooting complex. The knowledge base is developed in the process of successful operation of the system. In the commands of the computing system of the data flow architecture, a program of the traditional type is given, according to which switching of actuators is carried out to perform the NEUROCOMPUTER procedure. A method is proposed for reducing the complexity of the scalar product of highly rarefied (by zeros) vectors, which is the basic operation for counting the values of the neuron activation function. To do this, use the CYCLE command with the listed parameter values, as was indicated in the original description of the ALGOL language.*

**Keywords:** target distribution, reference trajectory, shooting complex, logical neural network, scalar product of vectors, data flow architecture

DOI: 10.17587/it.25.441-448

### References

1. Burtsev V. S. Jubilee collection of works of institutes of OIVTA RAS. T. P., Moscow, OIVTA RAS, 1993, pp. 5–27 (in Russian).

2. Zmeev D. N., Klimov A. V., Levchenko N. N., Okunev A. S., Stempkovsky A. L. *Informatika i ee primeneniye*, 2015, vol. 9, iss. 4, pp. 29–36 (in Russian).

3. Troshin G. I. Grigory Vasilyevich Kisunko - the founder of the antimissile defense of the USSR, an outstanding radio physi-

cist of the twentieth century, a writer, a poet (scientific biography), Moscow, *Novye tekhnologii*, 2015, 220 p. (in Russian).

4. Tretyakov Yu. N. 50 years in the rocket-space defense, Moscow, ООО "ТИПОГРАФИЯ КЕМ", 2010, 800 p. (in Russian).

5. Barsky A. B. Neural network methods for optimizing solutions, SPb, ITs "Intermediya", 2016, 312 p. (in Russian).

6. Barskii A. B., Shilov V. V. *Informatsionnye tekhnologii*, Prilozheniye, 2003, no. 7, 24 p. (in Russian).

7. Bekus D. B., Bauer F. A. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*. 1961, vol. 1, no. 2, pp. 308–342 (in Russian).

### Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: [it@novtex.ru](mailto:it@novtex.ru)

Технический редактор Е. В. Конова.

Корректор Н. В. Яшина.

Сдано в набор 29.04.2019. Подписано в печать 25.06.2019. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ ИТ719. 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 "Авансд солишнз". Отпечатано в ООО "Авансд солишнз". 119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: [www.aov.ru](http://www.aov.ru)

Рисунок к статье А. В. Щёкина

# «КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКАЯ ПАРАМЕТРИЗАЦИЯ В СОСТАВЕ ИНТЕГРИРОВАННОЙ САМ-СИСТЕМЫ»

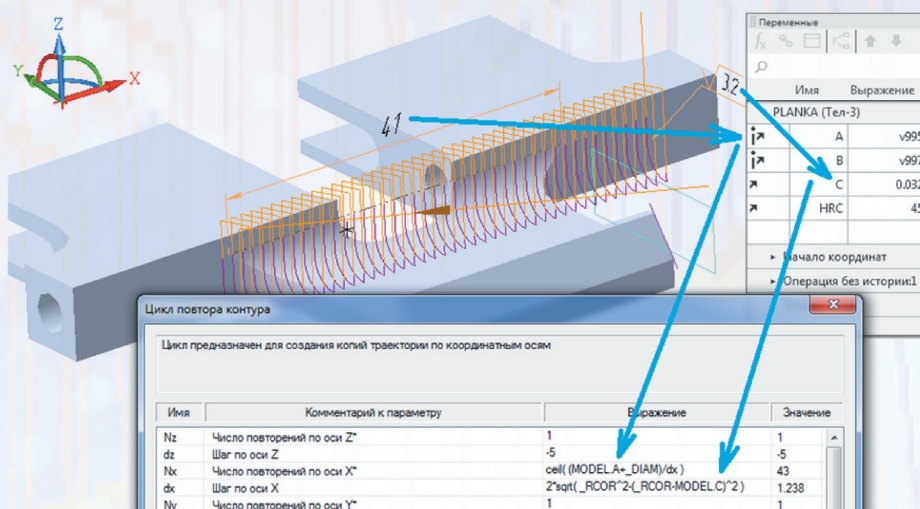


Рис. 6. Программирование пользовательской траектории

Рисунок к статье А. Н. Шниперова, М. С. Сосновского, П. М. Шипулина

# «РОБАСТНЫЙ МЕТОД МАРКИРОВАНИЯ ИЗОБРАЖЕНИЙ ЦИФРОВЫМ ВОДЯНЫМ ЗНАКОМ, ОСНОВАННЫЙ НА ОРТОГОНАЛЬНЫХ МОМЕНТАХ ЦЕРНИКЕ»



Рис. 4. Встраивание ЦВЗ в полутоновое изображение:

а – исходное изображение; б – изображение, маркированное ЦВЗ;

в – ЦВЗ объемом 256 бит, представленный в пространственной области

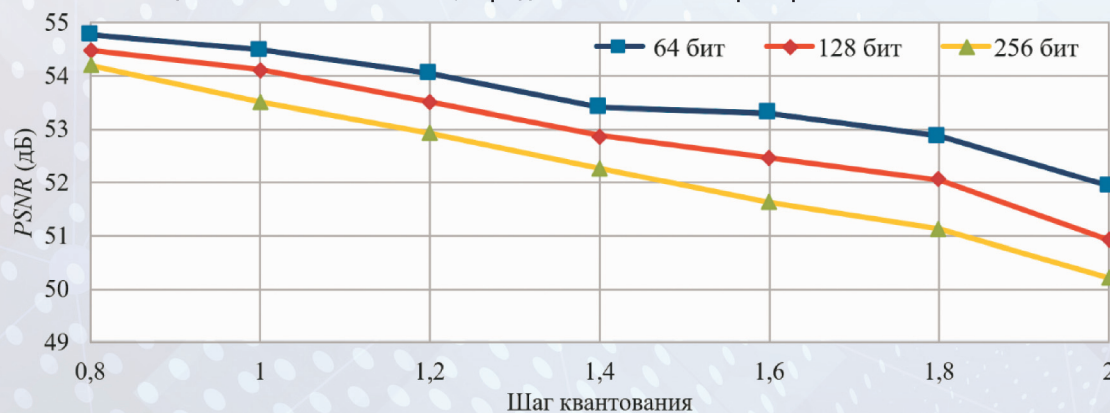
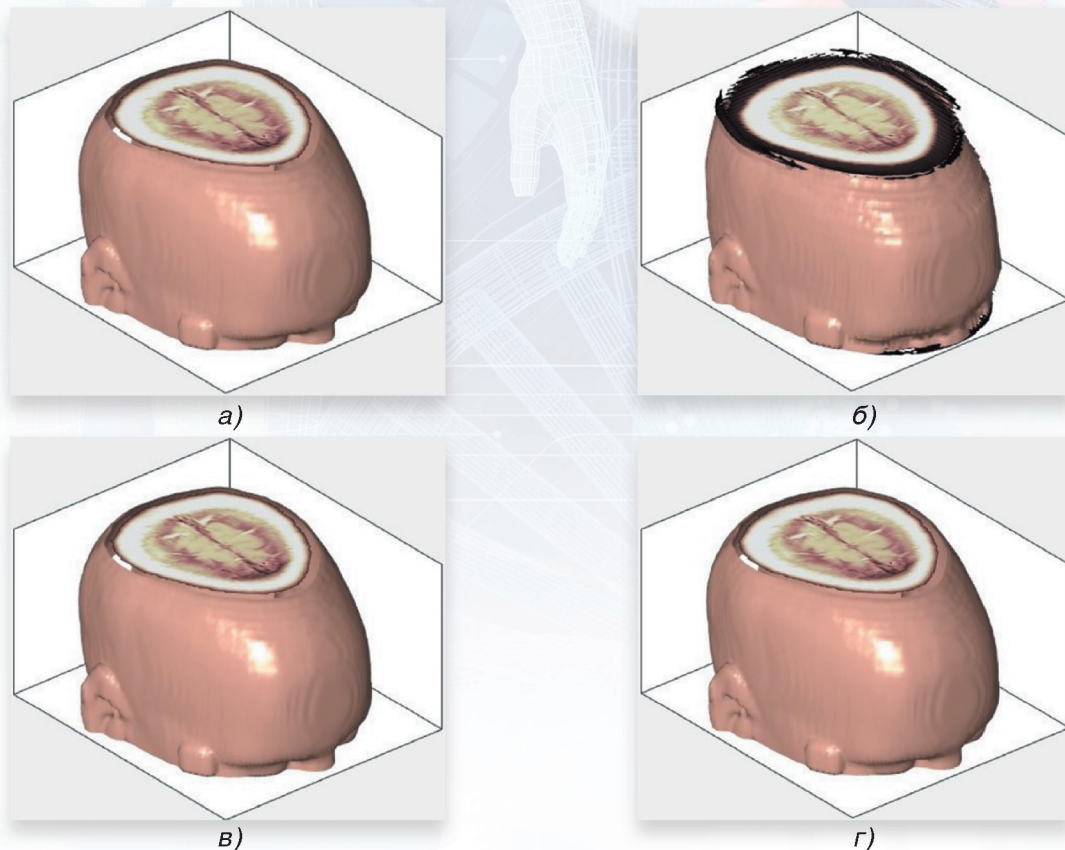


Рис. 5. Зависимость метрики PSNR от шага квантования и объема ЦВЗ

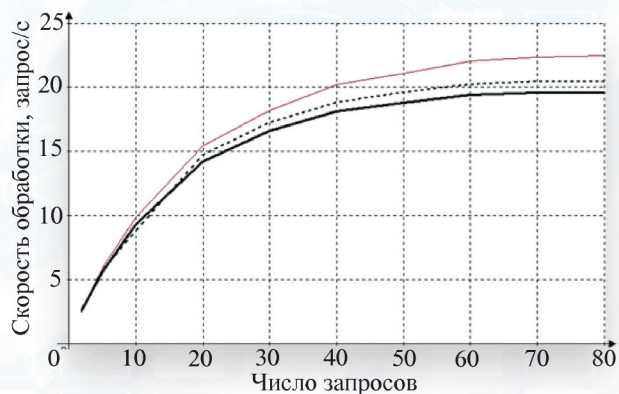


Рисунок к статье Н. Н. Нагорнова, П. А. Ляхова, Н. И. Червякова  
**«ИССЛЕДОВАНИЕ ШУМА КВАНТОВАНИЯ ФИЛЬТРОВ ДИСКРЕТНОГО  
 ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ ДЛЯ ОБРАБОТКИ  
 ТРЕХМЕРНЫХ ИЗОБРАЖЕНИЙ В МЕДИЦИНЕ»**

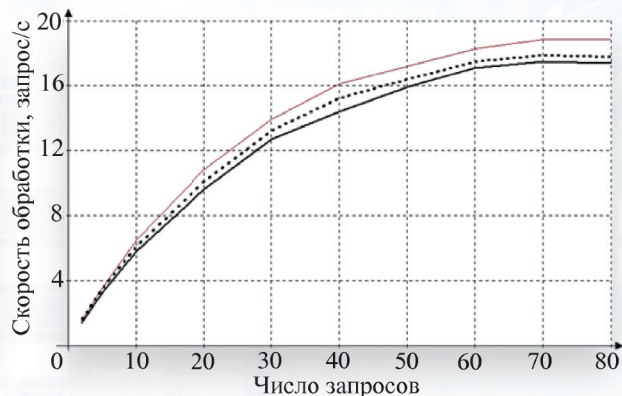


**Рис. 2. Пример ДВП 8-битного изображения «wmri» вейвлетом:**  
 а – исходное изображение; обработанное изображение: б –  $r = 9$ ,  $PSNR = 27,62$  дБ;  
 в –  $r = 12$ ,  $PSNR = 47,11$  дБ; г –  $r = 15$ ,  $PSNR = \infty$

Рисунки к статье Р. Э. Асратяна  
**«ОРГАНИЗАЦИЯ ОБРАБОТКИ ЗАЩИЩЕННЫХ СООБЩЕНИЙ В РАСПРЕДЕЛЕННЫХ  
 СИСТЕМАХ НА ОСНОВЕ CRYPTOGRAPHIC MESSAGE SYNTAX»**



**Рис. 6. Скорость обработки пакетов запросов (время обработки запроса 0,5 с)**



**Рис. 7. Скорость обработки пакетов запросов (время обработки запроса 1 с)**