

# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

3(187)  
2012

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ  
Издательство "Новые технологии"

## СОДЕРЖАНИЕ

### ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

- Афонин С. А., Бахтин А. В. Построение иерархии понятий на основе лексических шаблонов . . . . . 2  
Костенко К. И. Операции унифицированной технологии построения цифровых пространств знаний . . . . . 8  
Ходашинский И. А. Идентификация нечетких систем на базе алгоритма имитации отжига и методов, основанных на производных . . . . . 14  
Паламарь И. Н., Юлин С. С. Анализ применимости современных методов статистического моделирования для распознавания и понимания речи . . . . . 20

### СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Андрейчиков А. В., Хорычев А. А., Андрейчикова О. Н. Экспертная система для начальных стадий проектирования инноваций . . . . . 26  
Бобков С. Г., Задябин С. О., Крыницкий А. В., Новожилов Е. Е. Верификация модели контроллера ведущего интерфейса USB 2.0 . . . . . 33  
Кузнецов Е. С., Талицкий Е. Н., Шумарин С. В. Автоматизация проектирования виброзащиты ячеек электронной аппаратуры демпфирующими вставками . . . 37

### БАЗЫ ДАННЫХ

- Керимов С. Г., Керимова Е. С. Репозитории в информационных системах и программном обеспечении . . . . . 43

### КОДИРОВАНИЕ И ОБРАБОТКА СИГНАЛОВ

- Елесина С. И., Никифоров М. Б. Повышение эффективности генетического алгоритма . . . . . 49  
Васильев В. В., Симак Л. А. Полиномиальная аппроксимация сигналов как операционное исчисление: применение к моделированию динамических систем дробного порядка . . . . . 55

### Журнал в журнале

### НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- Добровотровский А. С. Оценка перспектив использования ассоциативных инструментальных платформ квантовых компьютеров, работающих в многозначном алфавите . . . . . 62  
Соломаха А. А., Милова К. А., Горбаченко В. И. Нейросетевая экспертная медико-технологическая система прогнозирования риска развития осложнений у больных с гнойно-деструктивными заболеваниями легких . . . . . 66  
Девятисильный А. С., Гриняк В. М. Нейросетевой алгоритм сопровождения судов в системах управления движением . . . . . 70  
Петрушан М. В., Самарин А. И. Метод контрастирования описаний изображений лиц для системы санкционированного доступа . . . . . 74  
Contents . . . . . 78  
Приложение. Меркушева А. В. Элементы прикладной теории времячастотных преобразований и обработки нестационарных сигналов в информационно-измерительных системах

Главный редактор  
НОРЕНКОВ И. П.

Зам. гл. редактора  
ФИЛИМОНОВ Н. Б.

Редакционная  
коллегия:

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

Редакция:

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

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

УДК 004.822

С. А. Афонин, канд. физ.-мат. наук, вед. науч. сотр.,  
e-mail: serg@msu.ru,

А. В. Бахтин, вед. программист,  
e-mail: anton@bakhtin.ru,

НИИ механики МГУ им. М. В. Ломоносова

## Построение иерархии понятий на основе лексических шаблонов

*Поисковые системы Интернет позволяют получить множество коротких фрагментов документов, содержащих заданные ключевые слова. В данной статье предложен метод автоматического построения гипонимических пар (общее—частное) на основе анализа таких фрагментов. Этот метод совмещает несколько известных приемов. Используются лексические шаблоны, чтобы построить гиперонимы-кандидаты для заданного термина и фильтрация, основанная как на лексических шаблонах, так и на частотном анализе. Приведены экспериментальные данные, свидетельствующие о применимости предлагаемого метода для решения практических задач.*

**Ключевые слова:** лексические шаблоны, онтология, гипонимия

### Введение

Отношение гипонимии (или общее—частное) является основой любой онтологии (модели предметной области). Наличие иерархических отношений (например, *шенилл — обивочная ткань*) позволяет человеку определить характеристики новых объектов по аналогии с известными ему однотипными объектами. Такие отношения могут быть использованы и во многих областях, связанных с автоматической обработкой текстов на естественном языке, таких как уточнение поискового запроса [1], аннотирование текста [2], категоризация [7], ответы на вопросы [8] и многих других. К сожалению, в большинстве случаев онтологии созданы вручную и охватывают ограниченный набор тематических областей. Более того, сложно поддерживать онтологию актуальной в быстро развивающихся областях, когда часто появляются новые объекты и связи. Таким образом, построение онтологий становится "узким местом" для практической реализации многих проектов на этом направлении. Один из подходов к его устранению — автоматическое или полуполупри автоматическое построение онтологий.

Большинство работ, направленных на построение иерархии терминов, устанавливают отношение гипонимии на основании автоматического анализа некоторого корпуса — совокупности текстов на естественном языке. В качестве такого корпуса обычно используются статьи энциклопедических словарей, газетные публикации или техническая документация. Выделяются два основных подхода к задаче построения иерархии понятий по текстам на естественном языке. Первый основан на *распределительной гипотезе* [3], которая утверждает, что семантически близкие термины встречаются в похожих контекстах. Примером статистического подхода к построению онтологий может служить работа [10], использующая следующую гипотезу: множество документов, где два термина встречаются одновременно, должно быть подмножеством множества документов, в котором встречается их гипероним.

Второй класс работ, к которому относится и данная, использует *лексические шаблоны* [4], которые сигнализируют о возможных семантических отношениях между словами. Например, в случае английского языка для поиска кандидатов на гипонимы-гиперонимы можно найти в корпусе выражения вида "*B is a A*", "*B is a kind of A*", или "*B, C and other As*". Данный подход основан на предположении, что в достаточно большом корпусе текстов содержится "определение" слова *B*, написанное по одному из таких шаблонов. Набор лексических шаблонов может быть построен как вручную, так и автоматически, с помощью методов машинного обучения [11]. В отличие от статистического подхода, лексические шаблоны позволяют выделять гиперонимы даже при единичном вхождении соответствующей фразы в корпус текстов.

Применение лексических шаблонов сталкивается с двумя затруднениями: небольшим набором шаблонов, которые можно использовать для обычных текстов (что, впрочем, одновременно является и плюсом), и шумом на выходе. Можно выделить две основные причины ошибки выделения гипонимов. Первый тип ошибок связан с *неправильной* структурой предложения. Этот факт означает, что синтаксическое значение шаблона не совпадает с предполагаемым. Так, предложение "... *urban birds in cities such as pigeons ...*" подходит под шаблон "*Hs such as T*", где *T* — это "*city*", а *Y* — "*pigeon*". Получаем, что *pigeon* (голубь) — гипоним для *city*. Второй тип ошибок связан с определением формы того, что мы хотим найти. Рассмотрим пример: *Typhoon is a twin engine canard delta wing multirole aircraft*. Из этого предло-

жения можно выделить целый ряд корректных гиперонимов, а именно — *aircraft*, *multirole aircraft*, *twin engine canard delta aircraft*, *twin engine canard delta wing multirole aircraft* и другие, а также некоторые некорректные — *twin canard aircraft*, *wing aircraft*. В рамках данной работы будем рассматривать в качестве гиперонимов как слова, так и словосочетания, используя понятие "фраза" для обоих вариантов. Задача в этом случае состоит в определении того, какая фраза является гиперонимом.

В последнее время в качестве корпуса используются документы сети Интернет, поскольку это наиболее распространенный и постоянно пополняемый источник данных. Можно ожидать, что для любого тематического направления в сети найдутся соответствующие документы, а скорость обновления документов в Интернет гораздо выше, чем в словарных статьях. Однако в этом случае устранение шума становится еще более серьезной задачей, поскольку при работе с такими данными нельзя рассчитывать на корректную синтаксическую структуру предложений. Эксперименты показывают, что методы, разработанные для работы со словарями и другими "качественными" источниками, оказываются неприменимыми для работы с Интернет. Следует также заметить, что проверка предложений на синтаксическую корректность при работе с поисковой системой не представляется возможной в связи с тем обстоятельством, что возвращаемые поисковой системой фрагменты текстов редко содержат предложения целиком.

Цель исследования, результаты которого представлены в настоящей работе, — создание метода автоматического построения гиперонимов заданного слова по текстам из Интернет без явного использования специальных словарей или другой информации, изменение которой в дальнейшем потребует работы экспертов. В качестве единственного источника текстовых данных выступает система Интернет-поиска. Поскольку описываемая в работе техника лексических шаблонов ранее применялась в основном для англоязычных текстов и ее адаптация к русскому языку представляет самостоятельную задачу, в данной работе используются тексты на английском языке.

### 1. Наивный алгоритм выделения гиперонимов

В статье [4] был предложен метод автоматического построения гипонимических пар, основанный на лексических шаблонах. С помощью шаблона "*Hs such as T*" по известному гиперониму *H* находили его гипонимы. Данный метод позволял добиться приемлемых результатов, однако имел два существенных ограничения. Во-первых, поиск вниз (от гиперонима к гипонимам) представляет меньший интерес, чем поиск вверх (от гипонима к гиперонимам), так как поиск вверх позволяет дополнительно ответить на вопрос "что это?". Во-вторых,

поиск проводился по энциклопедии (Academic American Encyclopedia, 1980). Как уже отмечалось, использование специально подготовленных текстов позволяет уменьшить число ошибок (в связи с более стандартизированной лексикой и отсутствием фактологических ошибок), но существенно уменьшает терминологический охват.

В данной работе в качестве корпуса выступают все тексты из Интернет и для поиска по ним используется система YahooBOSS API, позволяющая выполнять запросы к поисковой системе Yahoo и получать результаты (фрагменты текста, *сниппеты*) в удобном для машинной обработки виде (XML или JSON). Кроме стандартного поиска, при котором искомые слова находятся в произвольных местах страницы, синтаксис языковых запросов позволяет искать как точное вхождение фразы, используя кавычки, так и "почти" точное (когда искомые слова находятся "близко" друг от друга), используя ключевое слово "NEAR" (например, "*chimpanzee is a NEAR monkey*"). Кроме того, при поиске точного вхождения внутри кавычек можно использовать символ звездочки "\*", соответствующий одному произвольному слову. Синтаксис запросов был расширен символом переменной звездочки "*\*[n:m]*". Такой запрос распадается на *m - n* запросов, каждый из которых содержит от *n* до *m - 1* звездочек включительно.

Наивный алгоритм поиска гиперонимов заданного термина *T* состоит в выполнении запроса "*T is a \**" (например, "*chimpanzee is a \**") и выборе слов, которые стоят на месте звездочки в результатах выполнения запроса. Заметим, что описанный способ выделения гиперонимов совпадает с предложенным в работе [4] за исключением того, что мы используем "*is a*" вместо "*such as*". Для ранжирования полученных результатов можно предложить два варианта. Во-первых, каждому слову можно поставить в соответствие число найденных сниппетов, в которые оно входило. Во-вторых, для каждого слова *H* можно вычислить число хитов для фразы "*chimpanzee is a H*" (здесь и далее под числом хитов понимается число найденных страниц в поисковой системе для данного запроса, а все запросы выполняются "с кавычками"). Однако для слова *chimpanzee* наиболее частотными оказываются слова *great*, *very*, *huge* и *tour*, ни одно из которых не является корректным гиперонимом. Предполагалось, что вместо *H* будут стоять понятия, однако в действительности получаем произвольные слова.

Анализ показывает, что самое частотное вхождение (*great*) почти всегда порождается фразой вида: "*...Dwarf or Gracile Chimpanzee is a great ape ...*". Естественной эвристикой в данном случае представляется выбор последнего существительного в первой именной группе, что может быть сделано с использованием статистических методов определения частей речи (так называемые POS-тэггеры). В ре-

зультате ее применения наиболее частотными кандидатами для слова *chimpanzee* оказываются "ape", "monkey" и "hug" (крепкое объятие, захват). Однако частоты этих слов настолько близки, что никакого естественного решающего правила, которое бы приняло первые два слова, но отвергло бы "hug", построить нельзя. Более того, самое частотное слово не всегда является корректным гиперонимом. Так, для слова *orthodoxy* (православие) первый правильный гипероним *religion* занимает 7-ю строку. Таким образом, алгоритм из работы [4] может использоваться для построения множества кандидатов, однако для выделения гиперонимов необходимо проведение дополнительной фильтрации.

## 2. Фильтрация и уточнение результатов

Повышение эффективности наивного алгоритма связано с разработкой метода фильтрации кандидатов, который позволит исключить или уменьшить количество "мусора", а также с более точным определением фразы, являющейся в рамках заданного шаблона гипонимом или гиперонимом. Для создания фильтрующего критерия предлагается ввести набор числовых характеристик, признаков, позволяющих оценить гипонимичность заданной пары слов. Уточнение гипонимов и гиперонимов осуществляется на основании частотного анализа.

### 2.1. Правые и левые шаблоны

Первый критерий — это число вхождений пары ( $T$ ,  $H$ ) в шаблоны специального вида. В работе [9] было предложено поделить лексические шаблоны на *правые* и *левые* (те, в которых гипероним стоит справа и слева, соответственно). Пример правого шаблона — стандартный "*T is a (kind of) H*", примеры левого — "*Hs such as T*" или "*Hs including T*". В работе было отмечено, что для каждой гипонимической пары ( $T$ ,  $H$ ) требование вхождения хотя бы в один правый шаблон и хотя бы в один левый шаблон позволяет существенно увеличить точность при сохранении полноты. Из метода построения кандидатов не следует, что каждый из них вместе с исходным словом входит в какой-либо правый шаблон. Это обстоятельство связано со спецификой выделения и определения термов. Например, в случае, если в качестве термов рассматриваются именные группы, могут быть выделены некорректные кандидаты, для которых запрос "*T is a H*" может вернуть пустой результат (например для запроса "*typhoon is a twin canard aircraft*"). По этой причине на данном шаге нельзя ограничиться только левыми шаблонами.

Кроме правых и левых шаблонов предлагается использовать несколько шаблонов, позволяющих оценить распространенность гипонима и гиперонима. Они призваны помочь ответить на следующие вопросы: Десять хитов для шаблона "*T is a H*" — это

много или мало? Является ли  $T$  элементарным понятием? Является ли  $H$  высшим, наиболее общим понятием? Такими шаблонами являются " $T$ ", " $T is a *$ ", " $H is a *$ " и " $* is a T$ ".

### 2.2. Построение братьев для данного гипонима

В работе [6] было введено понятие *семантических классов* — множества слов, имеющих общий гипероним. Примерами семантических классов служат названия стран или игр с мячом. В терминах иерархии понятий элементы семантических понятий — братья (т. е. элементы поддеревя с корнем в гиперониме). Для поиска братьев  $T$  относительно  $H$  в работе [6] предлагается использовать шаблон "*Hs such as T and \**". Слова, стоящие на месте звездочки, и будут искомыми братьями. Преимуществом данного алгоритма является малое количество мусора за счет использования двух слов —  $H$  и  $T$ . Однако данный алгоритм имеет и недостатки. Во-первых, такой шаблон накладывает слишком жесткие требования на расположение слов. Вместо "*and*" могут использоваться запятые, а исходный элемент семантического класса  $T$  необязательно встречается сразу после "*such as*". Во-вторых, использование одного шаблона ограничивает область поиска.

Предлагаемая далее модификация решает эти проблемы. Определим *уровень братьев* следующим образом. Братями  $T$  относительно  $H$  уровня 0 называется множество, содержащее единственный элемент —  $T$ . Братями  $T$  относительно  $H$  уровня  $n + 1$  называется множество, для каждого терма в котором существует брат  $n$ -го уровня, встречающийся с ним в одном шаблоне, и не существует братьев меньшего уровня, удовлетворяющих этому условию. Для поиска братьев использовались запросы вида " $T*[0:3]$  and other  $Hs$ " и " $Hs$  such as  $*[0:3]$   $T$ ", которые возвращают связанные фрагменты текстов, содержащие  $H$  и  $T$ .

Задача извлечения братьев представляет отдельный интерес (например, в статье [6] поиск семантических классов рассматривается в качестве основной задачи). В рамках данной работы в большей степени важна чувствительность данного метода к ошибкам определения гиперонимов. Рассмотрим, например, самый частотный гипероним-кандидат для "*orthodoxy*" — "*series*". Для этой пары алгоритм не находит ни одного брата. Таким образом, число братьев на каждом уровне можно считать содержательным признаком пары.

### 2.3. Расширение гиперонимов и гипонимов

В предыдущем разделе предложены признаки, которые призваны помочь в построении критерия "хорошести" пары. Вместе с тем, не менее интересен и другой вопрос: откуда берутся неправильные варианты? Если не рассматривать ошибки POS-тэггера и фактологические ошибки, то можно выделить две причины: либо в исходном фрагменте

определялось что-то не то, что мы искали, либо не так, как мы нашли. В первом случае слева от "is a" стояло более специализированное (и, наверное, более длинное) понятие. Во втором случае мы выделили понятие справа слишком грубо. Именно методы расширения, уточнения фраз, разработанные для идентификации и исправления таких ошибок, описаны далее в настоящем разделе.

В качестве примера расширения гиперонимов рассмотрим пару (*ubuntu, distribution*), полученную при выделении кандидатов, например из фрагмента "I have to admit folks, Ubuntu is the best distribution in all these years". Является ли она гипонимической? С одной стороны, представляется очевидным, что пара (*ubuntu, linux distribution*) является гипонимической, и, как следствие, исходную пару стоит считать такой. С другой стороны, основное значение слова "distribution" — распространение, распределение. Среди нескольких первых страниц по запросу "distribution" нет ни одного результата, связанного с дистрибутивами операционных систем.

Попробуем удлинить гипероним влево на одно слово. Выполним запрос "*ubuntu is a \* distribution*" и выберем 500 результатов. В найденных фрагментах выделим множество слов, стоящих на месте звездочки. Каждому слову можно поставить в соответствие пару — число найденных сниппетов, в которые это слово входило, и отношение этого числа к числу всех сниппетов. Очевидно, что если какое-то расширение содержится в большинстве результатов, то его следует принимать в качестве "лучшего" гиперонима. Наиболее популярными расширениями оказываются **linux** (381), **great** (33) и **popular** (16). Согласно предыдущему рассуждению вместо "distribution" следует принять "linux distribution". Дальнейшее расширение влево не дает явного лидера. Аналогично можно расширять фразы вправо, а также комбинировать направления.

Однако расширения не являются панацеей от ошибок выделения кандидатов. Более того, они привносят новые задачи. Во-первых, поиск функции штрафа, функции остановки является нетривиальной задачей. Более того, выбор единственного кандидата не всегда является наилучшим решением. Например, расширение "type of" вправо в контексте "*chimpanzee is a type of*" приводит к двум равноправным вариантам: "*chimpanzee is a type of ape*" и "*chimpanzee is a type of monkey*". Процент вхождения каждого в выборку — 40 %. Таким образом, использование глобальной "планки" неэффективно. В реализованном алгоритме используется комбинированный подход. С помощью "низкой планки" отсекаются результаты, которые не могут рассматриваться как расширения. Далее, среди оставшихся с помощью кластеризации выбираются "головные" распределения.

Во-вторых, так как расширения работают со словами как со знаками, нет никаких гарантий

синтаксической корректности результатов. Например, после расширения "creature" в контексте "*elephant is a creature*" имеем два варианта: "*nobble creature of special*" и "*creature of majesty and great power and may*". Оба этих расширения не являются синтаксически целостными. Для решения этой задачи можно применить таггер и отбросить все слова после последнего существительного. Получим: "*nobble creature*" и "*creature of majesty and great power*".

Как отмечено ранее, результат расширения, даже в одну сторону, неоднозначен. Должно ли расширение заменять исходное слово, а именно — является ли наличие расширения (множества расширений) признаком "качества" исходной пары? В текущей реализации все эти вопросы переложены на механизмы фильтрации в пространстве признаков: все найденные расширения рассматриваются как полноценные кандидаты в гиперонимы.

Аналогичным образом можно проводить и расширение гипонимов. В некоторых случаях найденный гипероним соответствует некоторому побочному значению гипонима. Например, рассмотрим пару (*scanner, program*). Очевидно, что она является ошибочной. Если посмотреть на фрагменты, в которых встречается данная пара, то можно встретить выражения вида "*scanner is a program designed for searching for httpsocks*". Для выделения таких случаев предлагается использовать расширение гипонимов. Выполним запрос "*\* scanner is a program*" и обработаем полученные фрагменты так же, как и ранее. Получим *proxu scanner* (147), *a scanner* (56), *ftp scanner* (51), *port scanner* (29). Таким образом, наличие нетривиального расширения можно рассматривать как признак того, что гипероним относится к побочному значению гипонима, а наличие тривиального расширения — дополнительным признаком "качества" пары. Под тривиальными расширениями понимается исходное слово, а также артикль в сочетании с исходным словом.

Очевидно, что для более точного, длинного гиперонима существование расширенного гипонима более вероятно, чем гиперонима, представленного одним словом. Вместе с тем, если гипероним расширяется "мягко" (то есть с низким порогом, при котором расширение принимается), то и наличие расширения гипонима для какого-либо другого варианта расширения гиперонима более вероятно. В предельном случае, если принимаются все найденные фрагменты (все, что стоит после "is a") в качестве расширений, то для каждого из них будет существовать расширение гипонима — все, что стоит до "is a". По этой причине используются только "жесткие" расширения. Жесткое левое расширение получается в результате рекурсивного расширения гиперонима влево, до тех пор, пока есть вариант, входящий в более чем половину сниппетов в относительном исчислении и более чем в 30 сниппетов в абсолютном. Аналогичным образом получаем

правое жесткое расширение. Например, для пары (*scanner*, *tool*) такими расширениями будут "*network scan tool*" и "*tool*", соответственно.

### 3. Результаты тестирования

Введем обозначения для числа хитов, найденных для шаблонов, описанных в разд. 2:

- *isa\_strict* — "Т is a Н"
- *isan\_strict* — "Т is an Н"
- *isthe\_strict* — "Т is the Н"
- *kindof\_strict* — "Т is a kind of Н"
- *suchas\_strict* — "Нs such as Т"
- *including\_strict* — "Нs including Т"
- *especially\_strict* — "Нs especially Т"
- *word\_hits* — "Т"
- *isa* — "Т is a \*"

Пусть *brothers1* — число братьев первого уровня пары; *hyponym*, *hypernym* — гипоним и гипероним соответственно; *hyponym\_extension\_all*, *hyponym\_extension\_all\_from\_left*, *hyponym\_extension\_all\_from\_right* — расширения гипонима в общем случае, при жестком левом расширении гиперонима и при жестком правом расширении гиперонима соответственно; *extension\_depth* — 1, если гипероним является расширением, и 0 — иначе. Для краткости обозначим  $\ln(\text{word\_hits})$  через *ln\_w\_hits*.

Тогда критерий для гиперонимов, которые не являются расширениями, формулируется следую-

щим образом. Гипероним является корректным, если одновременно выполнены следующие условия:

- число братьев первого уровня больше 10;
- $\text{isa\_strict} > \ln\_w\_hits * 1.5$  или  $\text{isan\_strict} > \ln\_w\_hits * 1.5$  или  $\text{isthe\_strict} > \ln\_w\_hits * 1.5$  или  $\text{kindof\_strict} > \ln\_w\_hits$ ;
- $\text{suchas\_strict} > \ln\_w\_hits$  или  $\text{especially\_strict} > \ln\_w\_hits$  или  $\text{including\_strict} > \ln\_w\_hits$ ;
- расширения гипонима в общем случае, при жестком левом и жестком правом расширениях гиперонима либо не существуют, либо содержат тривиальное расширение.

Критерий для гиперонимов, которые являются расширениями, формулируется следующим образом. Гипероним является корректным, если одновременно выполнены следующие условия:

- существуют братья первого уровня;
- $\text{isa\_strict} > 0$  или  $\text{isan\_strict} > 0$  или  $\text{isthe\_strict} > 0$  или  $\text{kindof\_strict} > 0$ ;
- $\text{suchas\_strict} > 0$  или  $\text{especially\_strict} > 0$  или  $\text{including\_strict} > 0$ ;
- расширения гипонима в общем случае, при жестком левом и жестком правом расширениях гиперонима, либо не существуют, либо содержат тривиальное расширение.

Структура критерия продиктована идеями, обсуждаемыми в разд. 2. Численные значения были подобраны с помощью обучающей выборки.

Пары слов, отобранные на основе критерия, и их свойства

Гипоним	Гипероним	brothers1	isa_strict	isan_strict	isthe_strict	kindof_strict	suchas_strict	including_strict	especially_strict
ak47	<b>gun</b>	27	30	0	15	0	35	46	2
ak47	<b>machine gun</b>	1	12	0	1	0	1	13	0
ak47	<b>weapon</b>	40	28	0	48	0	113	100	4
begonia	<b>flower</b>	36	33	0	11	0	25	22	1
begonia	<b>plant</b>	115	29	0	1	0	182	32	3
chimpanzee	<b>mammal</b>	20	11	0	41	0	6	21	0
chimpanzee	<b>primate</b>	28	21	0	9	1	84	111	10
daiquiri	<b>cocktail</b>	16	60	0	3	1	17	5	0
daiquiri	<b>drink</b>	12	23	0	9	0	16	0	3
elephant	<b>animal</b>	110	26	281	103	5	1140	626	63
elephant	<b>creature</b>	31	44	0	0	0	29	19	0
elephant	<b>film</b>	14	114	0	12	0	76	50	7
elephant	<b>herbivore</b>	37	67	47	0	0	74	46	6
elephant	<b>large mammal</b>	42	32	0	0	0	112	105	5
elephant	<b>mammal</b>	74	121	0	9	5	332	256	10
guitar	<b>c instrument</b>	8	26	2	0	0	5	45	0
guitar	<b>combination</b>	21	243	0	23	0	21	14	0
guitar	<b>course</b>	55	118	2	19	0	52	51	2
guitar	<b>fretted string instrument</b>	3	26	0	0	0	17	39	0
guitar	<b>hobby</b>	37	135	0	2	1	56	60	0
guitar	<b>part</b>	31	192	0	92	0	87	98	6
guitar	<b>skill</b>	43	190	1	9	0	62	29	2
guitar	<b>string</b>	25	175	0	52	4	46	185	11
guitar	<b>style</b>	13	110	0	62	0	14	98	6
guitar	<b>talent</b>	14	79	0	4	0	50	35	4
guitar	<b>thing</b>	56	155	0	145	0	116	63	21

Для тестирования был выбран набор слов, для каждого из которых построены кандидаты в гиперонимы, что дало 5311 пар фраз. К ним был применен предложенный в данном разделе критерий гипонимичности. Некоторые примеры представлены в таблице. Корректные (на основе экспертной оценки) гиперонимы выделены жирным. Их количество в процентном отношении ко всем гиперонимам составляет 62 %. Следует отметить, что отношение выбранных критерием пар ко всем парам — 2 %.

### Заключение

Иерархию понятий можно применять не только как основу для построения онтологий, но и для расширения поисковых запросов, классификации текстов и решения других задач, связанных с обработкой естественных языков. В данной работе предлагается метод автоматического построения множества гиперонимов заданного термина. В основе предлагаемого метода лежат лексические шаблоны, предложенные в работе [4]. Ключевое отличие данной работы от работы [4] заключается в том, что используются неподготовленные тексты, полученные с помощью интернет-поисковика, и построение ведется снизу вверх (от гипонима к гиперониму). В работе исследована проблема выделения и уточнения кандидатов-гиперонимов; оптимизированы методы фильтрации, основанные на анализе количества хитов и количества братьев; предложен метод отделения побочных значений исходного термина; представлены результаты тестовых испытаний программных средств, реализующих предложенные алгоритмы.

В качестве возможного продолжения работы, кроме реализации данных методов для текстов русского языка, можно рассматривать следующие задачи. Результаты тестирования показывают, если слово имеет небольшое число предполагаемых гиперонимов (например, "ak47", "begonia", "elephant", и другие), то ошибочных среди них мало. Вместе с тем, поскольку для каждого слова был найден как минимум один гипероним, то возникает идея необходимости ранжирующего, а не бинарного критерия. Возможно, отсортировав кандидатов по значимости, можно будет для каждого слова выделять корректный гипероним с большой точностью.

Также представляет интерес более детальный анализ братьев. Разделение братьев по уровням,

предложенное в данной работе, в содержательном плане является попыткой выделить братьев, близких исходному слову. Однако можно использовать более точный метод: провести кластеризацию всех братьев (используя некоторую меру семантической близости, например SOCPMI [5]), и выбрать в качестве братьев первого уровня слова, входящие в кластер, содержащий исходное слово.

Представленный в работе метод выделения побочных значений гипонима в рамках данной работы используется исключительно как способ дополнительной фильтрации гиперонимов по той причине, что авторов интересовало только основное значение гипонима. Однако получаемые расширения (например, "proxy scanner", "virus scanner") могут быть использованы в рамках задачи поиска редких значений слова.

*Работа выполнена при частичной поддержке грантом РФФИ № 09-07-00366-а и госконтракта 07.514.11.4146 Министерства образования и науки РФ.*

### Список литературы

1. **Chirita P. A., Firan C. S., Nejd W.** Personalized query expansion for the web // Proc. of the SIGIR '07. New York: ACM, 2007. P. 7—14.
2. **Dang C., Luo X., Zhang H.** Wordnet-based summarization of unstructured document // W. Trans. on Comp. 2008. Vol. 7, № 9. P. 1467—1472.
3. **Harris Z. S.** Mathematical Structures of Language. New York: Krieger Pub Co, 1968. 230 p.
4. **Hearst M. A.** Automatic acquisition of hyponyms from large text corpora // Proc. of the 14th International Conference on Computational Linguistics. 1992. P. 539—545.
5. **Islam A., Inkpen D.** Semantic text similarity using corpus-based word similarity and string similarity // ACM Transactions on Knowledge Discovery from Data. July 2008. № 2. P. 10:1—10:25.
6. **Kozareva Z., Riloff E., Hovy E.** Semantic class learning from the web with hyponym pattern linkage graphs // Proc. of the Association for Computational Linguistics. 2008. P. 1048—1056.
7. **Li J., Zhao Y., Liu B.** Fully automatic text categorization by exploiting wordnet // Proc. of the AIRS'09. Berlin: Springer-Verlag, 2009. P. 1—12.
8. **Lopez V., Uren V., Motta E., Pasin M.** Aqualog: An ontology-driven question answering system for organizational semantic intranets // Web Semant. 2007. Vol. 5. № 2. P. 72—105.
9. **Ritter A., Soderland S., Etzioni O.** What is this, anyway: Automatic hypernym discovery // Proc. of AAAI-09 Spring Symposium on Learning, 2009. P. 88—93.
10. **Sanderson M., Croft B.** Deriving concept hierarchies from text // Proc. of the SIGIR'99. New York: ACM. 1999. P. 206—213.
11. **Snow R., Jurafsky D., Ng A.** Learning syntactic patterns for automatic hypernym discovery // Advances in neural information processing systems. 2005. N 17. P. 1297—1304.

**К. И. Костенко,**  
канд. физ.-мат. наук, доц., зав. каф.,  
Кубанский государственный университет  
e-mail: kostenko@kubsu.ru

## **Операции унифицированной технологии построения цифровых пространств знаний**

*Определена система операций для построения компонентов пространств знаний. С помощью предложенных операций реализуются процессы трансформации и развития системы формальных моделей для компонентов пространств знаний в практически применимые модели, допускающие эффективные программные реализации. Схемы применения операций составляют основу технологии построения цифровых пространств знаний.*

**Ключевые слова:** пространство знаний, алгебраическая система, гомоморфизм моделей, интеграция моделей

### **Введение**

Исследование теоретических моделей пространств знаний позволяет развивать фундаментальные и системные представления о свойствах целостных многообразий знаний в различных предметных областях. Такие представления обеспечивают лучшее понимание возможных схем формализации таких пространств. Составляющие формализм абстрактного пространства знаний математические системы согласованы с эмпирическими представлениями и, прежде всего, позволяют получать решение задач, связанных со свойствами представлений многообразий знаний [1, 2]. В то же время остается актуальной задача практического применения разработанных математических систем. Успехи в ее решении во многом зависят от того, насколько удастся преодолеть различие между уровнями абстрактно-формальных и прикладных моделей, используемых для представления пространств знаний. Создание эффективных механизмов трансформации и адаптации таких моделей к потребностям практики способствует развитию обоснованных прикладных технологий цифровых пространств знаний. Математическая состоятельность моделей абстрактных пространств знаний существенно зависит от компактности, однородности и общности используемых в них информационных и функциональных структур. Перечисленные свойства существенны для получения общих фундаментальных результатов.

Применение теоретических моделей на практике обычно сталкивается с целями, требующими значительного расширения и детализации моделей [3].

Это связано с потребностью одновременного учета многих атрибутов пространств знаний как сложных систем. В таком случае стремление к построению теоретических моделей, в полной мере учитывающих особенности прикладных пространств знаний, связано с опасностью потери компактности, однородности и общности таких моделей, что, как правило, приводит к уменьшению их фундаментальности. Тем не менее, сами создаваемые модели оказываются лучше приспособленными для реализации на их основе эффективных прикладных пространств знаний, обеспечивающих достижение целей соответствующих областей деятельности.

Последовательное преобразование фундаментальных моделей в программно реализуемые прототипы прикладных систем можно реализовать с использованием специальных операций, обеспечивающих возможность управления процессами разработки пространств знаний. Общие операции, составляющие указанные процессы, должны поддерживать создание, комбинирование и развитие моделей. Эти операции позволяют конструировать развиваемые семейства формальных моделей, ориентированных на достижение разных целей и имеющих разные уровни детальности. Указанные операции востребованы в качестве инструментов технологии пространств знаний в форме унифицированной системы этапов построения и реализации моделей для таких пространств.

В качестве основы создаваемой технологии целесообразно сформировать обоснованное семейство специальных классов операций, обеспечивающих общее управление процессами построения пространств знаний. Для этого в настоящей работе определяются классы операций интеграции, уточняющие представления об объединении отдельных составляющих такое пространство систем. Вводятся операции гомоморфного расширения, соответствующие преобразованиям общих моделей в более детальные и точные системы с наследованием основных структурных и функциональных свойств. Рассматриваются также операции, обратные к интеграции и гомоморфному расширению — расщепления и гомоморфные вложения. Стандартизация процессов построения систем моделей для пространств знаний и их программных реализаций обеспечивается с использованием специальной двухуровневой диаграммы, которая представлена в ключевой части настоящей работы. Ее верхний (абстрактный) уровень составляют теоретические системы, которые конструируются с помощью выбранной системы операций. Нижний (прикладной) уровень диаграммы составляют системы, которые являются программными реализациями моделей.



## 1. Структурная организация моделей компонентов абстрактного пространства знаний

В качестве основы для формализации пространств знаний будем использовать алгебраические системы, адаптированные к особенностям таких пространств. В данном случае каждая отдельная модель может быть представлена в виде тройки  $\Sigma = (T, F, P)$ .

Здесь  $T = \{t_1, \dots, t_m, \dots\}$ ,  $F = \{f_i^j(x_1, \dots, x_j) | i, j \in N\}$

и  $P = \{p_i^j(x_1, \dots, x_j) | i, j \in N\}$  — соответственно носитель (множество данных); множество морфизмов; множество предикатов на  $T$ . Верхний индекс в записях отображений (предикатов) обозначает число переменных, а нижний — порядковый номер морфизма (предиката) с заданным значением верхнего индекса.

На множествах  $T$ ,  $F$  и  $P$  определены вычислимые семейства классов  $CT$ ,  $CF$  и  $CP$ . Такие семейства содержат все элементы данных множеств. Такие семейства структурированы разрешимыми отношениями вложения и агрегирования классов, которые обозначаются в виде  $\subseteq$  и  $\triangleleft$ .

**Агрегирование классов данных.** Семейство  $CT$  составляют разрешимые подмножества множества  $T$ , которые будем называть классами данных. Класс  $T_i$  получается агрегированием классов  $T_{j_1}, \dots, T_{j_k}$ , если существует эффективная процедура, перечисляющая элементы  $T_i$ , использующая для этого  $T_{j_1}, \dots, T_{j_k}$ . При этом существуют эффективные обратные процедуры, перечисляющие  $T_{j_1}, \dots, T_{j_k}$  по  $T_i$ .

Пусть  $CT^1 = \{T_1^1, \dots, T_m^1\}$  и  $CT^2 = \{T_1^2, \dots, T_n^2\}$  — семейства классов данных в составе формальных моделей  $\Sigma_1$  и  $\Sigma_2$ . Для этих семейств имеет место агрегирование классов из  $CT^1$  в классы из  $CT^2$ , если  $T_1^2, \dots, T_n^2$  могут быть эффективно построены с помощью  $T_1^1, \dots, T_m^1$ . Этот факт означает, что существуют такие вычислимые преобразования  $\tau_1, \dots, \tau_m$ , для которых  $T_i^2 = \tau_i(T_1^1, \dots, T_m^1)$ , где  $i \in \{1, \dots, n\}$ .

**Агрегирование классов морфизмов.** Пусть  $f \in C$  и  $\forall C_i (C_i \subseteq C \rightarrow f \notin C_i)$ , где  $C \in CF$  связан отношением  $\triangleleft$  с классами  $C_1, \dots, C_k$ . Тогда морфизм  $f$  может быть представлен некоторой композицией элементов, содержащихся в классах  $C_1, \dots, C_k$ , называемых подморфизмами морфизма  $f$ . В общем случае всякая композиция интерпретируется с помощью алгоритмической процедуры, моделирующей процесс вычисления  $f$  по процедурам вычисления подморфизмов. Соответствующая этому морфизму процедура извлекает из  $C_1, \dots, C_k$  подходящие элементы и определяет порядок их выполнения при нахождении значений  $f$ . Если такое преобразование

оказывается вычислимым для всех подходящих морфизмов  $f$ , то появляется возможность алгоритмизации процессов вычисления любых элементов из  $F$  на основе их структурных представлений.

**Агрегирование классов предикатов.** Пусть класс  $P \in CP$  связан отношением  $\triangleleft$  с классами  $P_1, \dots, P_k$ . Это означает, что если  $\pi \in P$  и  $\forall P' (P' \subseteq P \rightarrow \pi \notin P')$ , то  $\pi$  представляется логической формулой, составленной из предикатов, принадлежащих классам  $P_1, \dots, P_k$ . Будем рассматривать случай, когда с формальной моделью  $\Sigma$  связана эффективная процедура, которая для произвольного предиката  $\pi \in P$  извлекает предикаты  $\pi_i \in P_i$  и порождает формулу, выражающую  $\pi$  с помощью предикатов  $\pi_i \in P_i$ ,  $i \in P\{1, \dots, k\}$ .

Требование вычислимости формальных моделей, представляющих компоненты пространств знаний, обеспечивается конструктивностью множеств (данных, морфизмов и предикатов) и разрешимостью их основных свойств. Использование структуры классов данных позволяет реализовать возможность ограничения множеств значений компонентов для наборов областей определения морфизмов и предикатов отдельными эффективно определяемыми классами данных.

## 2. Операции гомоморфного расширения и интеграции формальных моделей

Формализм абстрактного пространства знаний составляют модели, допускающие представление в виде алгебраических систем [1, 2]. Они соответствуют содержательно полному многообразию компонентов универсальной унифицированной онтологической структуры области метазнаний для цифрового пространства знаний, представленной в работе [1]. Система компонентов абстрактных и цифровых пространств знаний включает:

- семантические пространства;
- пространства элементарных и сложных знаний;
- пространства целей и задач, операций и процессов;
- системы типовых структур знаний и процессов, эволюции знаний;
- классификаторы и свойства знаний;
- многообразия операций над знаниями.

Все такие системы, рассматриваемые совместно, составляют категорию пространств знаний. Ее объектами являются формальные модели разных типов и их программные реализации. Технология проектирования формальных моделей для компонентов пространств знаний связана с последовательной разработкой семейства развивающихся и взаимно дополняющих одна другую теоретических и прикладных моделей. Процесс построения объектов в категории пространств знаний регулируется с помощью условий, выполнимость которых обеспечивает соответствие формальных моделей эмпирическим представлениям о составе и структурно-функциональных свойствах компонентов пространств знаний.

Свойства однородности и универсальности для элементов технологии пространств знаний могут быть достигнуты, если фиксировано многообразие применяемых в ней схем этапов и операций. В качестве базы таких операций будем использовать вычислимые преобразования, моделирующие общие вычислимые теоретико-множественные и алгебраические операции. Многообразие таких операций составляют аналоги операций объединения и гомоморфизма, а также обратных к ним операций. Операциями первого вида представляются схемы соединения систем без изменения их элементов. Гомоморфизмами реализуются схемы детализации и качественного усложнения моделей и их реализаций.

Интеграции и гомоморфизмы формальных моделей, реализующие системы  $\Sigma_1 = (T^1, F^1, P^1)$  и  $\Sigma_2 = (T^2, F^2, P^2)$ , в общем случае предполагают применение вычислимых функциональных механизмов согласования структурных и функциональных свойств данных систем. Для этого необходимо определить соответствия между классами и элементами множеств  $T^1$  и  $T^2$ ,  $F^1$  и  $F^2$ ,  $P^1$  и  $P^2$ . В частности, уточняются эффективные схемы формирования новых структур систем классов данных, морфизмов и предикатов, а также схем агрегирования для конкретных отображений и предикатов. Для гомоморфных расширений также требуется задать соотношения областей определения сопоставляемых морфизмов и предикатов, которые могут иметь разные размерности.

Ниже приводится система условий, уточняющих понятия интеграции и гомоморфного вложения моделей в составе категории пространства знаний. Эти условия образуют фрагмент онтологии цифровых пространств знаний, используемый механизмами управления процессами построения и применения таких пространств. В общем виде они составляют систему унифицированных требований к структуре и характеристикам процессов трансформации формальных моделей, близких по структурным и функциональным свойствам. Автоматизация таких процессов является одной из целей технологии цифровых пространств знаний.

### 2.1. Интеграция формальных моделей в пространствах знаний

Операция интеграции преобразует любые две формальные модели в новую модель, множества морфизмов и предикатов которой составляют объединения аналогичных множеств исходных моделей. Дополнительно формируются структуры классов данных, морфизмов и предикатов новой модели в отношениях вложения и агрегирования классов.

Пусть  $\Sigma_1 = (T^1, F^1, P^1)$  и  $\Sigma_2 = (T^2, F^2, P^2)$  — произвольные формальные модели категории пространств знаний. Множества отображений (предикатов) данных этих моделей структурированы в системы классов  $CF^1 = \{C_i^1 | i \in N\}$  и  $CF^2 = \{C_i^2 | i \in N\}$ ,

$(CP^1 = \{P_i^1 | i \in N\}$  и  $CP^2 = \{P_i^2 | i \in N\})$  с помощью отношений вложения и агрегирования. Формирование аналогичных структур для модели  $\Sigma_3$ , являющейся результатом интеграции  $\Sigma_1$  и  $\Sigma_2$ , связано с эффективным преобразованием семейств  $F^1$  и  $F^2$  ( $P^1$  и  $P^2$ ) в семейства морфизмов  $F^3$  (предикатов  $P^3$ ) и классов морфизмов  $CF^3 = \{C_i^3 | i \in N\}$  (классов предикатов  $CP^3 = \{P_i^3 | i \in N\}$ ) модели  $\Sigma_3$ .

Рассмотрим систему условий, определяющих структуры классов морфизмов в отношениях агрегирования и вложения классов.

1. Система классов  $CF^3$  определяется с помощью такого вычислимого сюръективного отображения  $Id : CF^1 \cup CF^2 \rightarrow CF^3$ , что всякий класс  $C_k^3 \in CF^3$  является объединением всех таких классов  $C \in CF^1 \cup CF^2$ , для которых  $Id(C) = C_k^3$ . Для определенных с помощью  $Id$  системы классов  $CF^3$  должно быть разрешимо свойство вложения.

2. Структура агрегирований для классов из  $CF^3$  задается с помощью вычислимого отображения  $\Upsilon : CF^3 \rightarrow (CF^3)^*$ , где отображение  $\Upsilon$  сопоставляет отдельным классам  $CF^3$  конечные подмножества  $CF^3$ . С помощью  $\Upsilon$  задается фрагмент отношения агрегирования классов из  $CF^3$ , определяемый соотношением  $\forall C_1, C_2 \in CF^3 (C_2 \in \Upsilon(C_1) \rightarrow C_1 \triangleleft C_2)$ . Полное определение отношения  $\triangleleft$  получается с помощью транзитивного и рефлексивного замыкания фрагмента данного отношения, порождаемого отображением  $\Upsilon$ .

Дополнительно для  $\Upsilon$  требуется, чтобы выполнялись условия:

$$a) \forall C_1, C_2, \dots \in CF^3 (\forall i \in N (C_{i+1} \in \Upsilon(C_i)) \rightarrow \forall i, j (i \neq j \rightarrow C_i \neq C_j));$$

$$b) \forall d \in \{1, 2\} \forall C_i^d, C_j^d \in CF^d (C_i^d \triangleleft C_j^d \rightarrow Id(C_j^d) \triangleleft Id(C_i^d)).$$

Здесь первое условие обеспечивает выполнимость свойства антисимметричности для  $\triangleleft$ . Вторым условием гарантируется сохранение отношений агрегирования, существовавших между классами из  $CF^d$ , где  $d \in \{1, 2\}$ , в классах из  $CF^3$ .

3. отождествление морфизмов из  $CF^1$  и  $CF^2$  в  $CF^3$  осуществляется с помощью вычислимого отображения  $\vartheta : F^1 \cup F^2 \rightarrow F^3$ , сопоставляющего морфизмам из  $F^1 \cup F^2$  элементы  $F^3$ , для которого:

$$a) f_j^2 = f_i^1 \leftrightarrow \vartheta(f_j^2) = \vartheta(f_i^1);$$

$$b) f_j^d, f_i^d \rightarrow C_v^d \rightarrow \vartheta(f_j^d), \vartheta(f_i^d) \in Id(C_v^d), \text{ где } d \in \{1, 2\}.$$

4. Схемы агрегирования морфизмов определяются с помощью вычислимого отображения  $\Psi: F^3 \rightarrow (F^3)^*$ , сопоставляющего отдельным морфизмам из множества  $F^3$  конечные семейства морфизмов (называемых подморфизмами). Для  $\Psi$  и  $\Upsilon$  должны выполняться условия:

а) если  $f_j^3 \in C_i^3$ , а также  $\forall C(C \subseteq C_i^3 \rightarrow f_j^3 \notin C)$

и  $\Upsilon(C_i^3) = \{C_1, \dots, C_r\}$ , то  $\Psi(f_j^3) = \{f_1, \dots, f_r\}$ , где  $f_i \in C_i$  для  $i \in \{1, \dots, r\}$ ;

б) если  $f_j^3 = \vartheta(f_j^d)$ ,  $d \in \{1, 2\}$ , и в  $\Sigma_d$  морфизм  $f_k^d$  является подморфизмом для  $f_i^d$ , то  $\vartheta(f_k^d)$  также является подморфизмом для  $f_j^3$ .

Схемы агрегирования и вложения классов предикатов при интеграции формальных моделей представляются системой требований, аналогичных рассмотренным выше условиям для классов морфизмов моделей.

## 2.2. Специальные классы операций интеграции

Операция интеграции формальных моделей определяет общее семейство вычисляемых преобразований, согласованных с унифицированной структурой моделей, составленной системами классов данных, морфизмов и предикатов. Практическое применение операции, полностью использующее ее возможности, связано с одновременным определением нескольких разных отображений. Следствием этого может стать необходимость одновременного формирования громоздкой конструкции, усложняющей процессы разработки и реализации моделей. По этой причине представляют интерес декомпозиции операций интеграции на этапы или более простые частные случаи данной операции. Рассмотрим естественные примеры таких случаев.

а. *Прямая сумма формальных моделей, реализующая преобразование, близкое к их объединению.* Для этого множества морфизмов и предикатов интегрируемых моделей должны быть дизъюнктивными. Операция интеграции использует тождественное отображение  $Id$ , преобразующее классы морфизмов и предикатов моделей  $\Sigma_1$  и  $\Sigma_2$  в равные им классы в  $\Sigma_3 = \Sigma_1 \oplus \Sigma_2$ . Остальные отображения из определения операции интеграции сохраняют структуры вложений и агрегирования классов морфизмов и предикатов. Имена классов, морфизмов и предикатов из  $\Sigma_1$  и  $\Sigma_2$  в модели  $\Sigma_3$  сохраняются.

б. *Агрегирование формальных моделей, расширяющее операцию прямой суммы моделей  $\Sigma_1$  и  $\Sigma_2$  с помощью дополнительных отношений агрегирования классов одной модели классами другой модели.* В данном случае агрегирование модели  $\Sigma_1 = (T^1, F^1, P^1)$  в модель  $\Sigma_2 = (T^2, F^2, P^2)$  реализуется в два этапа. Сначала конструируется прямая сумма  $\Sigma_3 = \Sigma_1 \oplus \Sigma_2$ . На сле-

дующем этапе классам из  $CF^3$  ( $CP^3$ ), которые представляют классы из  $CF^2$  ( $CP^2$ ), сопоставляются семейства классов, представляющих классы из  $CF^1$  ( $CP^1$ ). Они расширяют отношения агрегирования классов морфизмов и предикатов в  $CF^3$  ( $CP^3$ ).

## 2.3. Гомоморфизм формальных систем абстрактного пространства знаний

Пусть  $\Sigma_1 = (T^1, F^1, P^1)$  и  $\Sigma_2 = (T^2, F^2, P^2)$  — формальные модели для некоторой компоненты пространства знаний, где  $T^1 = \{T_1^1, \dots, T_m^1\}$  и  $T^2 = \{T_1^2, \dots, T_n^2\}$  структурированы отношениями вложения и агрегирования для классов данных в  $\Sigma_1$  и  $\Sigma_2$ . Эти же отношения структурируют множества классов морфизмов (предикатов)  $F^1 = \{f_i^j(x_1, \dots, x_j) | i, j \in N\}$  и  $F^2 = \{\varphi_i^j(y_1, \dots, y_j) | i, j \in N\}$  ( $P^1 = \{p_i^j(x_1, \dots, x_j) | i, j \in N\}$  и  $P^2 = \{\pi_i^j(y_1, \dots, y_j) | i, j \in N\}$ ) в  $\Sigma_1$  и  $\Sigma_2$ .

Уточним понятие вычислимого гомоморфизма для формальных моделей, представляющих отдельные компоненты пространства знаний. Как и в случае операции интеграции, определяется семейство специальных вычисляемых отображений.

1. Соответствие иерархий классов морфизмов и предикатов определяют вычисляемые отображения  $c_F: CF^1 \rightarrow CF^2$  и  $c_P: CP^1 \rightarrow CP^2$ , которые монотонны относительно отношений вложения и агрегирования классов.

2. Соответствия морфизмов и предикатов определяют вычисляемые отображения  $h_F^1: N \times N \rightarrow N$ ,  $h_F^2: N \times N \rightarrow N$  и  $h_P^1: N \times N \rightarrow N$ ,  $h_P^2: N \times N \rightarrow N$ , называемые индексированиями морфизмов и предикатов, задаваемые с помощью правил:

морфизму  $f_i^j(x_1, \dots, x_j)$  сопоставляется

$$\varphi_{h_F^1(i,j)}^{h_F^2(i,j)}(y_1, \dots, y_{h_F^2(i,j)});$$

предикату  $p_i^j(x_1, \dots, x_j)$  сопоставляется

$$\pi_{h_P^1(i,j)}^{h_P^2(i,j)}(y_1, \dots, y_{h_P^2(i,j)}).$$

3. Уточнение классов данных для областей значений переменных морфизмов и предикатов осуществляется для элементов множеств морфизмов  $F^1$  и  $F^2$ , а также предикатов  $P^1$  и  $P^2$  с помощью вычисляемых отображений  $t_{F^1}: N \times N \rightarrow N$ ,  $t_{F^2}: N \times N \rightarrow N$ ,  $t_{P^1}: N \times N \rightarrow N$ ,  $t_{P^2}: N \times N \rightarrow N$ . Значения классов данных, соответствующих переменным морфизмов и предикатов, определяются по правилам:

а) переменная  $x_k$  ( $k \leq j$ ) морфизма  $f_i^j(x_1, \dots, x_j)$  принимает значения из множества  $T_{t_{F^1(i,j,k)}}^1$ ; переменная  $x_k$  ( $k \leq j$ ) морфизма  $\phi_i^j(x_1, \dots, x_j)$  принимает значения из множества  $T_{t_{F^2(i,j,k)}}^1$ ;

б) переменная  $x_k$  ( $k \leq j$ ) предиката  $P_i^j(x_1, \dots, x_j)$  принимает значения из множества  $T_{t_{P^1(i,j,k)}}^1$ ; переменная  $x_k$  ( $k \leq j$ ) предиката  $\pi_i^j(x_1, \dots, x_j)$  принимает значения из множества  $T_{t_{P^2(i,j,k)}}^1$ .

4. Соответствие областей определения морфизмов (предикатов) задаются с помощью вычислимого семейства отображений, которые для каждой пары морфизмов (предикатов)  $f_i^j(x_1, \dots, x_j)$  и  $\phi_{h_F^1(i,j)}^2(y_1, \dots,$

$y_{h_F^2(i,j)})$  ( $P_i^j(x_1, \dots, x_j)$  и  $\pi_{h_P^1(i,j)}^2(y_1, \dots, y_{h_P^2(i,j)})$ ) определяют множество отображений  $\zeta_k(x_1, \dots, x_j)$ , где  $k = 1, \dots, h_F^2(i, j)$ , ( $\zeta_k(x_1, \dots, x_j)$ ,  $k = 1, \dots, h_P^2(i, j)$ ).

Такие отображения преобразуют элементы области определения морфизма  $f_i^j$  в значения отдельных переменных морфизма  $\phi_{h_F^1(i,j)}^2(y_1, \dots, y_{h_F^2(i,j)})$  (элементы области определения  $P_i^j$  в значения отдельных переменных предиката  $\pi_{h_P^1(i,j)}^2(y_1, \dots, y_{h_P^2(i,j)})$ ).

Вычислимым гомоморфизмом формальных моделей  $\Sigma_1$  в  $\Sigma_2$  называется вычислимое отображение  $h_T: \bigcup_{i=1, \dots, m} T_i^1 \rightarrow \bigcup_{i=1, \dots, n} T_j^2$ , для которого существуют такие вычисляемые соответствия классов отображений и предикатов, морфизмов и предикатов, областей значений переменных морфизмов и предикатов, областей определения отображений и предикатов, что

$$h_T(f_i^j(x_1, \dots, x_j)) = \phi_{h_F^1(i,j)}^2(\zeta_1(x_1, \dots, x_j), \dots, \zeta_{h_F^2(i,j)}(x_1, \dots, x_j));$$

$$h_T(P_i^j(x_1, \dots, x_j)) = \pi_{h_P^1(i,j)}^2(\zeta_1(x_1, \dots, x_j), \dots, \zeta_{h_P^2(i,j)}(x_1, \dots, x_j)).$$

Если существует вычислимый гомоморфизм  $\Sigma_1$  в  $\Sigma_2$ , то  $\Sigma_1$  гомоморфно вкладывается в  $\Sigma_2$ . Назовем  $\Sigma_1$  гомоморфным расширением  $\Sigma_2$ , если существует вычислимый гомоморфизм  $\Sigma_1$  в  $\Sigma_2$ . Если при этом вычисляемые соответствия множеств морфизмов и предикатов моделей сюръективны, то  $\Sigma_1$  называется полным гомоморфным расширением  $\Sigma_2$ .

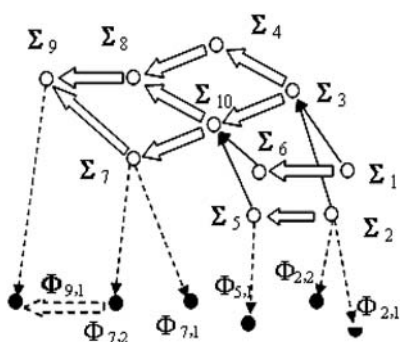
### 3. Обращение операций интеграции и гомоморфного расширения

Операции интеграции и гомоморфного расширения реализуют основные схемы построения формальных моделей, связанные с развитием моделей. Указанные операции не являются обратимыми, поскольку определяемые ими соответствия моделей не инъективны и многозначны. Уточнения операций преобразования, обратных к преобразованиям интеграции и гомоморфных расширений, будем называть расщеплениями и гомоморфными вложениями. Расщеплением модели  $\Sigma$  называется всякая такая пара моделей  $\Sigma_1$  и  $\Sigma_2$ , что  $\Sigma$  может быть получена из  $\Sigma_1$  и  $\Sigma_2$  с помощью операции интеграции. Последняя операция полезна для формирования моделей, являющихся фрагментами других моделей. Операция гомоморфного вложения преобразует модель  $\Sigma_1$  в модель  $\Sigma_2$ , если существует вычислимый гомоморфизм  $\Sigma_1$  в  $\Sigma_2$ .

### 4. Диаграммы для систем разрабатываемых формальных моделей

Технология построения пространств знаний связана с решением задачи управления процессами развития системы формальных моделей компонентов пространств и процессами создания программных реализаций моделей. Указанными процессами реализуются этапы абстрактного и прикладного моделирования пространств знаний. Структуру процесса построения моделей и их программных реализаций будем представлять в виде двухуровневой диаграммы, вершинами которой являются формальные модели и их программные реализации. Вершины диаграммы связывают ориентированные ребра четырех основных типов, а именно — интеграции, гомоморфного расширения, реализации и трансформации реализаций. Модели и их реализации будем обозначать в виде  $\Sigma_\alpha$  и  $\Phi_{\alpha, \beta}$ ,  $\alpha \in I$ , где  $I$  — множество индексов (имен) формальных моделей, а  $\beta \in J_\alpha$ , где  $J_\alpha$  — множество индексов реализаций для  $\Sigma_\alpha$ .

Вершинами первого (абстрактного) уровня диаграммы являются модели. Ребра, соединяющие такие вершины, представляют связи интеграции и гомоморфного расширения, которые изображаются на соответствующей диаграмме сплошными двойными и одинарными стрелками. Реализации формальных моделей представляют собой конкретные информационные системы, составляющие нижний



Фрагмент диаграммы моделей и их реализаций

(прикладной) уровень диаграммы. В них уточнены классы данных (морфизмов и предикатов) и алгоритмы их вычисления. Пример фрагмента такой диаграммы приведен на рисунке.

На данном рисунке изображены формальные модели ( $\Sigma_1 - \Sigma_{10}$ ) и их программные реализации ( $\Phi_{2,1}, \Phi_{2,2}, \Phi_{5,1}, \Phi_{7,1}, \Phi_{7,2}, \Phi_{9,1}$ ), обозначаемые на диаграмме с помощью черных и белых вершин. Для представления ребер между моделями на диаграмме использованы разные виды стрелок. Связи гомоморфного расширения формальных моделей обозначаются двойными стрелками. Если такая стрелка ведет из модели  $\Sigma_\alpha$  в модель  $\Sigma_\beta$ , то это означает, что  $\Sigma_\beta$  является расширением  $\Sigma_\alpha$ . Так, например,  $\Sigma_8$  является гомоморфным расширением  $\Sigma_4$  и  $\Sigma_{10}$ . Сплошные одинарные стрелки на диаграмме представляют связи интеграции моделей. Семейство таких стрелок, ведущих в вершину  $\Sigma_\alpha$ , распадается на пары. Если некоторую такую пару составляют стрелки с началами  $\Sigma_\beta$  и  $\Sigma_\gamma$ , то это означает, что  $\Sigma_\alpha$  получается из  $\Sigma_\beta$  и  $\Sigma_\gamma$  с помощью интеграции. Одинарные штриховые стрелки на диаграмме соединяют формальные модели с их реализациями. Наконец, двойные штриховые стрелки соединяют реализации моделей, представляя механизмы трансформации структур данных, функциональных и логических инструментов, а также массивов данных в них. Примером рассматриваемой ситуации является ребро, соединяющее вершины  $\Phi_{9,1}$  и  $\Phi_{7,2}$ .

## 5. Базовые модели компонентов пространств знаний

Базовые модели относятся к абстрактному уровню для диаграммы процессов построения про-

странств знаний. Остальные модели получаются из базовых с помощью операций интеграции и гомоморфного расширения. Выразительные возможности перечисленных и обратных к ним операций позволяют использовать в качестве базовой специальную модель  $\Sigma^0 = (\{\perp, T\}, \{f_1^1\}, \{p_1^1\})$ . Носитель  $\Sigma^0$  составляет класс данных  $\{\perp, T\}$ . Морфизмом  $f_1^1$  представляется тождественное отображение на  $T = \{\perp, T\}$ . Предикат  $p_1^1$  является ложным для  $\perp$  и истинным для  $T$ . Из  $\Sigma^0$  с помощью определенных выше операций можно формировать модели компонентов пространств знаний. Семейства морфизмов в составе моделей уточняются с помощью обязательного указания областей определения и значений, а также специальных условий (ограничений), представляемых с использованием элементов множеств морфизмов и предикатов формальных моделей.

## Заключение

Предложенная в работе двухуровневая технология разработки пространств знаний основана на математически понятном и регулярном семействе схем создания и преобразования формальных моделей. Она удобна для организации процесса конструирования развиваемого семейства моделей и их программных реализаций. Данная технология поддерживает возможность использования аналогов фундаментальных алгебраических инвариантов для решения практических задач. Последнее означает появление качественно новых схем процессов создания и применения интеллектуальных информационных систем.

## Список литературы

1. Костенко К. И. Классификация операций в пространствах знаний // XII национальная конференция по искусственному интеллекту с международным участием (труды конференции). Тверь 2010, 20–24 сентября. Т. 2. С. 155–163.
2. Костенко К. И., Кузьменко И. П., Левицкий Б. Е. Классы операций цифровых пространств знаний // Информатизация образования и науки. 2010. № 2 (6). С. 137–152.
3. Бениаминов Е. М. Алгебраические методы в теории баз данных и представлений знаний. М.: Научный мир. 2003. 184 с.

**И. А. Ходашинский**, д-р техн. наук, проф.,  
e-mail: hodashn@rambler.ru  
Томский государственный университет  
систем управления и радиоэлектроники

## Идентификация нечетких систем на базе алгоритма имитации отжига и методов, основанных на производных<sup>1</sup>

*Работа направлена на развитие методологии идентификации нечетких систем. Для этой цели используются алгоритмы имитации отжига, фильтр Калмана, метод Левенберга—Марквардта, градиентный метод и их совместное применение в виде гибридного алгоритма. Работоспособность алгоритмов успешно подтверждена серией имитационных экспериментов.*

**Ключевые слова:** идентификация нечетких систем, алгоритм имитации отжига, фильтр Калмана, метод Левенберга—Марквардта, градиентный метод

### Введение

Моделирование стало основной методологией познания действительности, научно обоснованным способом "получения оценок параметров и изучения свойств сложных систем" [1]. Традиционные подходы к математическому моделированию требуют от разработчиков достаточно глубоких математических знаний и навыков. Поэтому большинство математических моделей — это результат работы математика, но не исследователя конкретной проблемы или предметной области. Однако, даже если этот исследователь не в состоянии разработать математическую модель, он может описать изучаемую систему и ее поведение лингвистически. Остается решить вопрос, как преобразовать лингвистическое описание в математическое представление.

Нечеткое моделирование — эффективный подход к преобразованию лингвистических данных в математические формулы и алгоритмы. Нечеткая модель представляет реальную систему в форме, легко понимаемой и интерпретируемой непрофессиональным математиком, такая модель может быть легко модифицирована. В системе нечеткого моделирования собственно знания или модель изучаемой системы отделены от средств манипулирования этими знаниями, что позволяет исследователям ускорить процесс разработки модели, сосредоточившись на вопросе "что делать" вместо вопроса "как делать" [2].

<sup>1</sup> Работа выполнена при финансовой поддержке РФФИ (проект № 09-07-99008).

Известно, что моделирование систем включает два основных этапа: идентификацию структуры и идентификацию параметров. Идентификация структуры — определение таких характеристик нечеткой системы, как число нечетких правил, число лингвистических термов, на которое разбиты входные и выходные переменные. Идентификация параметров — это определение неизвестных параметров antecedентов и консеквентов нечетких правил путем оптимизации работы нечеткой системы по заданному критерию [3, 4]. Для оптимизации параметров нечетких систем используют две группы методов. Первая группа — классические методы оптимизации, основанные на производных: метод наименьших квадратов, градиентный метод, фильтр Калмана, метод Левенберга—Марквардта. Эти методы дают точные результаты, но имеют тенденцию сходиться к локальным оптимумам. Трудности применения классических методов оптимизации, в частности проблема локального экстремума и "проклятие размерности", заставляют обратиться ко второй группе методов — метаэвристических, таких как алгоритмы муравьиной колонии, роящихся частиц, имитации отжига, генетические алгоритмы [4]. Метаэвристика — это метод оптимизации, многократно использующий простые правила или эвристики для достижения оптимального или субоптимального решения. Достоинство метаэвристических методов заключается в большей устойчивости. Но это методы грубой настройки, требующие больших временных ресурсов. Кроме того, применение метаэвристик не гарантирует нахождения оптимального решения и, как правило, связано с эмпирической настройкой параметров используемых алгоритмов. Использование гибридных алгоритмов позволит объединить преимущества метаэвристических методов с преимуществами классических методов, основанных на производных. Такое объединение повысит качество решений при умеренном количестве ресурсов и за приемлемое время.

Цель работы — описание и анализ идентификации параметров нечетких систем на основе метода имитации отжига, метода наименьших квадратов, градиентного метода, фильтра Калмана, метода Левенберга—Марквардта и на их совместном применении.

### Постановка задачи

В нечеткой системе типа синглтон  $i$ -е правило имеет следующий вид:

$$\text{IF } x_1 = A_{1i} \text{ AND } x_2 = A_{2i} \text{ AND } \dots \text{ AND } x_m = A_{mi} \text{ THEN } y = r_i$$

где  $A_{ij}$  — лингвистический терм, которым оценивается переменная  $x_j$ ;  $r_i$  — действительное число,

которым оценивается выход  $y$ . Нечеткая система осуществляет отображение  $f: \mathfrak{R}^m \rightarrow \mathfrak{R}$ :

$$f(\mathbf{x}) = \sum_{i=1}^n \mu_{A_{1i}}(x_1) \cdot \mu_{A_{2i}}(x_2) \cdot \dots \cdot \mu_{A_{mi}}(x_m) r_i / \sum_{i=1}^n \mu_{A_{1i}}(x_1) \cdot \mu_{A_{2i}}(x_2) \cdot \dots \cdot \mu_{A_{mi}}(x_m),$$

где  $\mathbf{x}$  — входной вектор;  $n$  — число правил;  $m$  — число входных переменных;  $\mu_{A_{ij}}$  — функция принадлежности, определяемая набором своих параметров, например, треугольная — тремя параметрами, трапецевидная — четырьмя, гауссова и параболическая — двумя.

Нечеткая система типа Такаги—Сугено имеет правила следующего вида:

$$\text{IF } x_1 = A_{1i} \text{ AND } x_2 = A_{2i} \text{ AND } \dots \text{ AND } x_m = A_{mi} \text{ THEN } y = a_{0i} + a_{1i}x_1 + \dots + a_{mi}x_m.$$

Отображение  $f$  для системы типа Такаги—Сугено определяется формулой:

$$f(\mathbf{x}) = \frac{\sum_{i=1}^n \mu_{A_{1i}}(x_1) \mu_{A_{2i}}(x_2) \cdot \dots \cdot \mu_{A_{mi}}(x_m) (a_{0i} + a_{1i}x_1 + a_{2i}x_2 + \dots + a_{mi}x_m)}{\sum_{i=1}^n \mu_{A_{1i}}(x_1) \mu_{A_{2i}}(x_2) \cdot \dots \cdot \mu_{A_{mi}}(x_m)}.$$

Нечеткая система может быть представлена как

$$y = f(\mathbf{x}, \boldsymbol{\theta}),$$

где  $\boldsymbol{\theta} = \|\theta_1, \dots, \theta_N\|$  — вектор параметров;  $N = mab$ ,  $a$  — число параметров, описывающих одну функцию принадлежности;  $b$  — число термов, описывающих одну входную переменную;  $y$  — скалярный выход системы.

Пусть дано множество обучающих данных (таблица наблюдений)  $\{(\mathbf{x}_p; t_p), p = 1, \dots, K\}$ , тогда средняя квадратическая функция ошибки (СКО), являющаяся численным критерием адекватности модели, вычисляется по следующей формуле:

$$E(\boldsymbol{\theta}) = \frac{1}{K} \sqrt{\sum_{p=1}^K (t_p - f(\mathbf{x}_p, \boldsymbol{\theta}))^2}. \quad (1)$$

Проблема идентификации сводится к проблеме поиска минимума заданной функции в многомерном пространстве, координаты которого соответствуют параметрам  $\boldsymbol{\theta}$  нечеткой системы.

### Алгоритм имитации отжига

Алгоритм имитации отжига использует упорядоченный случайный поиск, основываясь на аналогии с процессом кристаллизации с минимальной энергией при охлаждении [6]. Особенностью алгоритма является допустимость принятия решений, приводящих к увеличению ошибки. На начальном этапе работы алгоритма создается текущее решение, которое при его изменении случайным образом

переходит в рабочее. Рабочее решение может опять вернуться в текущее либо перейти в лучшее при условии уменьшения энергии. Но рабочее решение может быть принято в качестве текущего, даже если его энергия превышает энергию текущего, в том случае, когда выполняется критерий допуска, который основывается на следующем уравнении:

$$P(\delta E) = \exp(-\delta E/T),$$

где  $\delta E$  — разность ошибок рабочего и текущего решений, вычисленных по формуле (1). Значение коэффициента  $T$  — "температуры отжига" изменяется согласно принятому закону. Таким образом, чем ниже температура, тем меньше вероятность принятия решения, которое не улучшит текущую ситуацию. Алгоритм завершает работу, если температура достигла конечного значения  $t_0$ . На протяжении всего процесса оптимизации в отдельной переменной сохраняются координаты точки, которой соответствует лучшее из найденных решений.

При высокой температуре плохие решения принимаются чаще, чем отбрасываются. При снижении температуры вероятность принятия худшего решения уменьшается. Таким образом, при высоких температурах алгоритм отжига выполняет поиск оптимального решения во всем диапазоне решений. При понижении температуры диапазон поиска уменьшается.

Существуют различные варианты снижения температуры и генерации текущего решения. В нашей работе используются два. Первый вариант — схема тушения, когда изменение параметров происходит по равномерному закону распределения в допустимых границах, а температура снижается линейно:

$$T_{i+1} = \alpha T_i,$$

где  $\alpha$  — коэффициент снижения температуры,  $0 < \alpha < 1$ .

Второй вариант — схема Больцмана, здесь изменение параметров происходит по нормальному закону распределения со значением самого параметра в качестве математического ожидания и со значением температуры в качестве среднего квадратического отклонения. Температура снижается по закону

$$T(k) = \frac{T}{\ln(1+k)}.$$

Собственно пошаговый алгоритм имитации отжига для параметрической идентификации нечетких систем приведен в работах [3, 5].

### Настройка консеквентов правил

При настройке консеквентов правил применяется метод наименьших квадратов (МНК). В системе типа синглтон необходимо найти значение выходной переменной  $r_i$  для каждого  $i$ -го правила. Для этого необходимо минимизировать целевую функцию

$$\sum_{k=1}^K \left( t_k \prod_{j=1}^m \mu_{A_{ji}}(x_{jk}) - r_i \prod_{j=1}^m \mu_{A_{ji}}(x_{jk}) \right)^2 \rightarrow \min.$$

После нахождения производной от целевой функции, приравнивания ее к нулю и решения получившегося уравнения получим формулу для нахождения консеквента каждого правила базы правил:

$$r_i = \frac{\sum_{k=1}^K t_k \left( \prod_{j=1}^m \mu_{A_{ji}}(x_{jk}) \right)^2}{\sum_{k=1}^K \left( \prod_{j=1}^m \mu_{A_{ji}}(x_{jk}) \right)^2},$$

где  $x_{jk}$ ,  $t_k$  — взятые из таблицы наблюдений значения входных и выходных переменных.

Для нечеткой системы типа Такаги—Сугено необходимо найти значение вектора коэффициентов  $\mathbf{a}_i^T = [a_{0i}, a_{1i}, \dots, a_{mi}]$  для консеквента каждого  $i$ -го правила. В матричном виде уравнение для нахождения вектора коэффициентов консеквента каждого правила имеет следующий вид:

$$\mathbf{a}_i = [\mathbf{X}^T \mathbf{W}_i \mathbf{X}]^{-1} \mathbf{X}^T \mathbf{W}_i \mathbf{Y},$$

где

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & \dots & x_{m1} \\ 1 & x_{12} & \dots & x_{m2} \\ \dots & \dots & \dots & \dots \\ 1 & x_{1K} & \dots & x_{mK} \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} t_1 \\ t_2 \\ \dots \\ t_K \end{bmatrix},$$

$$\mathbf{W}_i = \begin{bmatrix} \left( \prod_{j=1}^m \mu_{A_{ji}}(x_{j1}) \right)^2 & 0 & \dots & 0 \\ 0 & \left( \prod_{j=1}^m \mu_{A_{ji}}(x_{j2}) \right)^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \left( \prod_{j=1}^m \mu_{A_{ji}}(x_{jK}) \right)^2 \end{bmatrix}.$$

### Методы, основанные на производных

*Фильтр Калмана* (ФК) в идентификации систем используется достаточно давно и успешно. Однако в идентификации нечетких систем этот аппарат применяется крайне редко [7].

Нечеткая система однозначно характеризуется состоянием, заданным значениями параметров функций принадлежности и значениями консеквентов правил. Вычисляемой (измеряемой, наблюдаемой) переменной, определенным образом связанной с состоянием системы, является выход нечеткой системы. Два последовательно определенных вектора состояния связаны рекуррентным уравнением (модель процесса). Определенным уравнением связаны вектор измерений и вектор состояний (модель измерения). Общая схема алгоритма такова: сначала задаются значения вектора состояния и значения его матрицы ковариаций. Далее в заданные дискретные времена выполняются две операции — прогноз и корректировка. Прогноз связан с опре-

делением ожидаемого значения вектора состояния и предварительной оценкой корреляционной матрицы. На этапе корректировки вычисляется матричный "коэффициент Калмана", обновляется оценка вектора состояния исходя из новых результатов измерения, и обновляется оценка корреляционной матрицы. Сходимость алгоритма и точность определения параметров нечеткой системы зависят от выбора значений матриц ковариаций шума процесса и шума измерения.

Фильтр Калмана представляется в виде линейной системы, дискретной во времени. Для задачи параметрической идентификации система может быть описана следующим образом:

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + \omega_n, \quad \mathbf{d}_n = h_n(\boldsymbol{\theta}_n) + v_n,$$

где  $\boldsymbol{\theta}_n$  — вектор состояния системы в момент времени  $n$ , например, при двух входных переменных, определенных на  $l$  термах с треугольными функциями принадлежности, описываемыми тройкой параметров  $(a, b, c)$ , в системе типа синглтон вектор параметров будет выглядеть следующим образом:

$$\boldsymbol{\theta}_n = [a_{11} b_{11} c_{11} \dots a_{1l} b_{1l} c_{1l} a_{21} b_{21} c_{21} \dots a_{2l} b_{2l} c_{2l}];$$

$\mathbf{d}$  — целевой вектор выхода нечеткой системы;  $h(\cdot)$  — нелинейная функция состояния системы, описывающая выход нечеткой системы с текущими параметрами;  $\omega_n$ ,  $v_n$  — шумы процесса и измерения, соответственно, которые вводятся искусственно для устойчивости работы алгоритма.

Процесс фильтрации сводится к следующей рекурсии:

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \mathbf{K}_n [t - y(\boldsymbol{\theta}_{n-1})];$$

$$\mathbf{K}_n = \mathbf{P}_n \mathbf{H}_n (\mathbf{R}_n + \mathbf{H}_n^T \mathbf{P}_n \mathbf{H}_n)^{-1};$$

$$\mathbf{P}_{n+1} = (\mathbf{P}_n - \mathbf{K}_n \mathbf{H}_n^T \mathbf{P}_n) + \mathbf{Q}_n,$$

где  $t$  — желательное выходное значение, заданное таблицей наблюдений;  $y(\boldsymbol{\theta}_{n-1})$  — нелинейная функция состояния системы, описывающая выход нечеткой системы с текущими параметрами;  $\mathbf{K}_n$  — матрица усиления фильтра Калмана;  $\mathbf{P}_n$  — ковариационная матрица погрешности оценивания состояния системы;  $\mathbf{R}_n$  — ковариационная матрица шума измерения;  $\mathbf{Q}_n$  — ковариационная матрица шума процесса;  $\mathbf{H}_n^T = \left. \frac{\partial y(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_n}$  — матрица частных про-

изводных нечеткого выхода относительно параметров функций принадлежности.

Алгоритм параметрической идентификации на основе фильтра Калмана приведен ниже.

*Вход:* начальное значение вектора параметров  $\boldsymbol{\theta}_0$ , вектор выхода таблицы наблюдений  $\mathbf{t}$ , ковариационная матрица погрешности оценивания состоя-



ния  $\mathbf{P}_0$ , значения матриц шумов  $\mathbf{R}$  и  $\mathbf{Q}$ , требуемое число итераций.

*Выход:* оптимизированный вектор параметров функций принадлежности, значение ошибки нечеткого вывода.

*Шаг 1.* Вычислить матрицу частных производных выхода от параметров нечеткой системы по формуле.

*Шаг 2.* Вычислить матрицу усиления  $\mathbf{K}$ .

*Шаг 3.* Вычислить вектор параметров нечеткой системы по формуле  $\theta$ .

*Шаг 4.* Вычислить значение ковариационной матрицы ошибок параметров  $\mathbf{P}$ .

*Шаг 5.* Применить МНК для настройки консеквентов.

*Шаг 6.* Если проведено заданное число итераций, то выход, иначе — перейти на шаг 1.

*Метод Левенберга—Марквардта* (ЛМ) является комбинацией простейшего градиентного метода и метода Гаусса—Ньютона и имеет преимущества того и другого [8]. Итерации в методе Левенберга—Марквардта проводятся по следующей формуле:

$$\mathbf{u} = -(\mathbf{J}^T\mathbf{J} + \alpha\text{diag}[\mathbf{J}^T\mathbf{J}])^{-1}\mathbf{J}^T\mathbf{E},$$

где  $\mathbf{E}$  — вектор ошибок на всех наблюдениях;  $\mathbf{J}$  — Якобиан, матрица частных производных от этих ошибок по параметрам;  $\mathbf{J}^T\mathbf{J}$  — Гессеиан;  $\text{diag}[\mathbf{J}^T\mathbf{J}]$  — диагональная матрица, полученная из матрицы Гессе;  $\alpha$  — управляющий параметр.

Первый член в формуле ЛМ соответствует линейной модели, а второй член — градиентному спуску. Управляющий параметр  $\alpha$  задает относительную значимость этих двух подходов. Всякий раз, когда удастся снизить ошибку, управляющий параметр уменьшается, усиливая тем самым роль линейных предположений и стремления прыгнуть сразу в точку минимума. Каждый раз, когда ошибку уменьшить не удастся, управляющий параметр увеличивается, придавая тем самым большее значение градиентному спуску и уменьшая значение шага. Это делается для того, чтобы обеспечить уменьшение ошибки в данной точке.

Алгоритм ЛМ для настройки нечетких систем приведен ниже.

*Вход:* начальное значение вектора параметров  $\theta_0$ , регулирующий коэффициент  $\alpha > 0$ , вектор выхода таблицы наблюдений  $\mathbf{t}$ , требуемое число итераций.

*Выход:* оптимизированный вектор параметров функций принадлежности, значение ошибки нечеткого вывода.

*Шаг 1.* При заданных  $\theta_k$  и  $\alpha$  найти  $\mathbf{u}_k$ .

*Шаг 2.* Вычислить  $\theta_{k+1} = \theta_k + \mathbf{u}_k$ .

*Шаг 3.* Применить МНК для настройки консеквентов правил.

*Шаг 4.* Вычислить значение ошибки нечеткого вывода  $\mathbf{E}_{k+1}$ .

*Шаг 5.* Если  $\mathbf{E}_{k+1} > \mathbf{E}_k$ , то увеличить  $\alpha = 10 \cdot \alpha$ , иначе уменьшить  $\alpha = \alpha/10$ . Присвоить  $k = k + 1$ .

*Шаг 6.* Если выполнено заданное число итераций, то выход, иначе перейти на шаг 1 с текущими параметрами  $\alpha$  и  $k$ .

*Градиентный метод* (ГМ). В практике обучения нечетких правил этот метод используется давно. Как известно, суть метода заключается в том, что последующее приближение функции получается из предыдущего движением в направлении, противоположном направлению градиента целевой функции. При оценивании параметров нечетких моделей целевой функцией является средняя квадратичная ошибка, а вектор параметров определен на множестве параметров функций принадлежности и параметров консеквентов правил. Основное достоинство метода — его быстрая сходимость, недостаток — способность застревать в локальных оптимумах. Применение ГМ для идентификации параметров нечеткой системы подробно описано в работе [9].

*Гибридный алгоритм.* Методы, основанные на производных, обеспечивают поиск оптимального решения по-разному. Градиентный спуск "двигает" каждый параметр последовательно для каждого элемента обучающей выборки. Фильтр Калмана относительно долго вычисляет шаг с помощью матричных преобразований, кардинально меняя положение точки в пространстве и значительно улучшая решение. Метод Левенберга—Марквардта находит компромисс между линейной моделью и градиентным спуском, варьируя значение управляющего коэффициента. Каждый метод, основанный на производных, рано или поздно застревает в локальном минимуме, т. е. при текущей итерации не может улучшить решение, полученное ранее. Здесь можно передать управление сразу следующему методу, также использующему производные, который, возможно, найдет способ продолжать поиски глобального минимума. Однако не исключено, что и следующий алгоритм не сможет улучшить решение из полученной точки пространства. Одним из решений этой проблемы является применение алгоритма имитации отжига, который в силу своего стохастического характера практически неуязвим для локальных экстремумов. Целесообразно начинать работу гибридного алгоритма с метода имитации отжига, поскольку заранее неизвестно, насколько удачны начальные параметры нечеткой системы.

### Исследование алгоритмов

Исследование алгоритмов идентификации нечетких систем проводилось при решении задач аппроксимации. В качестве тестовых были выбраны следующие функции:

$$A) F(x) = [1 + 10\exp(-100(x - 0,7)^2)] \cdot \left( \frac{\sin\left(\frac{125}{x + 1,5}\right)}{x + 0,1} \right),$$

где  $x \in [0; 1]$ ;

Таблица 1

Параметры нечетких систем

Функция	Число входных переменных	Число правил	Число точек в таблице наблюдений	Число термов, представляющих входную переменную
A	1	12	100	12
B	2	25	121	5
C	3	27	125	3
D	2	25	121	5

B)  $F(x, y) = \sin\left(x\frac{2}{\pi}\right) \sin\left(y\frac{2}{\pi}\right)$ , где  $x, y \in [-5; 5]$ ;

C)  $F(x, y, z) = 1 + x^{1/2} + y^{-1} + z^{-3/2}$ , где  $x, y, z \in [1; 5]$ ;

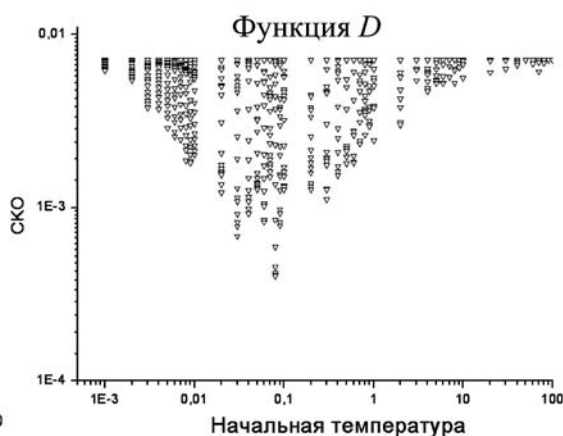
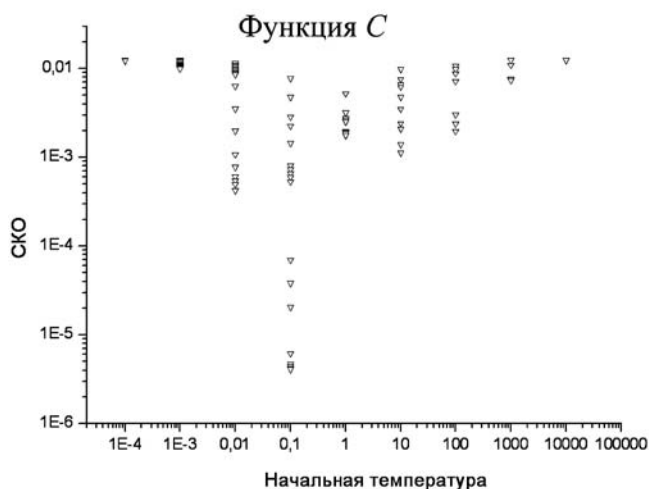
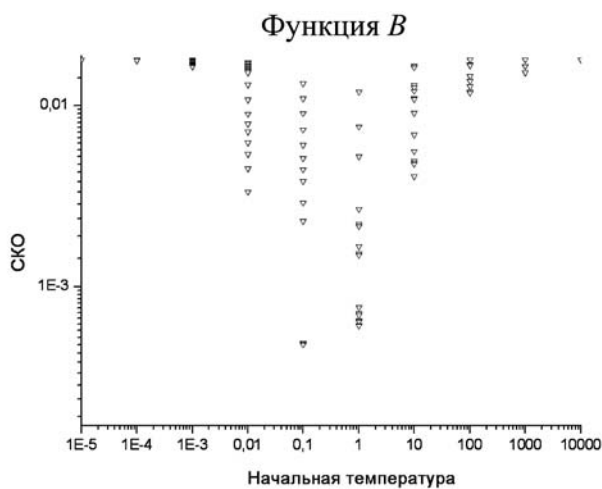
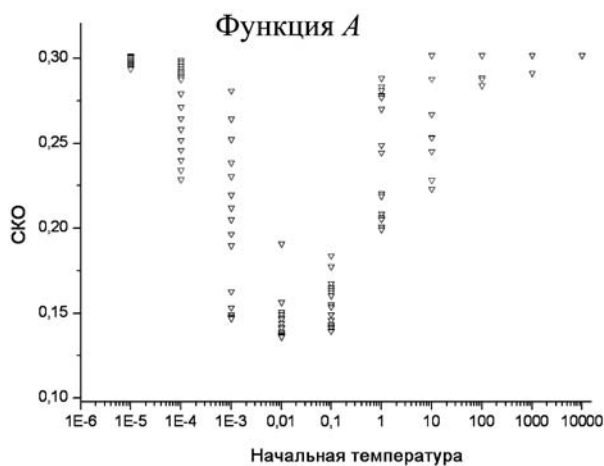
D)  $F(x, y) = y \sin x$ , где  $x, y \in [-\pi/2; \pi/2]$ .

Параметры нечетких систем, аппроксимирующих заданные функции, представлены в табл. 1. Симметричные треугольные функции принадлежности изначально равномерно распределены по области определения входных переменных.

В качестве изменяемых параметров для алгоритма ФК использованы значения диагональных элементов

ковариационных матриц шума процесса **Q** и шума измерения **R**. В результате проведенных экспериментов было установлено, что на процесс параметрической идентификации влияют не сами значения, а их соотношение. Оптимальный (исходя из соотношения точность—время идентификации) результат  $E = 0,1512926$  для функции *A* достигается при соотношении  $\text{diag}[\mathbf{R}]:\text{diag}[\mathbf{Q}] = 1:1E-5$ ; результат  $E = 0,0003586752$  для функции *B* — при соотношении 1:10; для функции *C* ошибка  $E = 2,1289620E-16$  будет при соотношении 1:10; для функции *D* ошибка  $E = 7,0341640E-13$  — при соотношении 1:1.

Изменяемыми параметрами алгоритма имитации отжига (схема тушения) являются значение начальной температуры *T* и коэффициента понижения  $\alpha$ . Эксперименты показали, что малое значение  $\alpha$  в меньшей степени "разрешает" алгоритму отдаляться от принятого на первых итерациях решения. В то время как значение  $\alpha$ , близкое к 1, позволяет продолжать стохастический поиск, сохраняя при этом лучшее решение. Чем меньше *T* и  $\alpha$ , тем реже алгоритм будет принимать решения, не улучшающие текущую итерацию, однако именно таким образом алгоритм сможет найти глобально лучшее решение.



Исследование алгоритма имитации отжига, схема Больцмана

Сравнение алгоритмов

Алгоритм	СКО для тестовых функций			
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
Фильтр Калмана	0,1519	$3,586 \cdot 10^{-4}$	$2,128 \cdot 10^{-16}$	$7,034 \cdot 10^{-13}$
Метод Левенберга—Марквардта	0,1423	$4,744 \cdot 10^{-4}$	$3,81 \cdot 10^{-14}$	$1,151 \cdot 10^{-16}$
Градиентный спуск с изменяющимся шагом	0,1623	$2,28 \cdot 10^{-12}$	$1,448 \cdot 10^{-12}$	$1,992 \cdot 10^{-12}$
Метод имитации отжига	0,1165	$5,167 \cdot 10^{-4}$	$8,178 \cdot 10^{-6}$	$4,032 \cdot 10^{-4}$
Гибридный алгоритм	0,113	$2,04 \cdot 10^{-12}$	$2,021 \cdot 10^{-16}$	$1,13 \cdot 10^{-16}$
Алгоритм муравьиной колонии [10]	0,0206	0,00235	$4,93 \cdot 10^{-6}$	$2,361 \cdot 10^{-8}$
Алгоритм Mitaim и Kosko [11]	1,426	—	—	—
Алгоритм Lisin и Gennert [11]	0,247	—	—	—
Алгоритм Teng, Wang, Chiu [12]	—	<0,001	—	—
Алгоритм Aliyari, Teshnehlab, Sedigh [13]	—	—	0,00243	—

Вместе с тем, когда  $T$  и  $\alpha$  слишком велики, работа алгоритма напоминает случайный поиск. Большие значения коэффициентов удобны для поиска глобального оптимума на начальных стадиях, а малые, когда нужно сделать несколько итераций, затратив немного времени, например в составе гибридного алгоритма.

В схеме Больцмана входным параметром является значение начальной температуры  $T$ . На рисунке представлены результаты исследования влияния значения начальной температуры на сходимость алгоритма для схемы Больцмана. Для тестовой функции  $D$  приведен график динамического изменения ошибки по итерациям для одного прогона алгоритма на каждом значении температуры  $T$ . Как видно из представленных графиков, несмотря на стохастичность алгоритма, определенные значения начальной температуры  $T$  выгодно отличаются от остальных. Это объясняется расстоянием между вершинами соседних термов входных лингвистических переменных нечетких систем: предпочтительно выбирать  $T$ , равное или на порядок меньшее этого расстояния. В таком случае генерация новых параметров будет соответствовать размерности конкретной нечеткой системы.

Результаты сравнения алгоритмов между собой, а также гибрида и других методов, использованных для задачи параметрической идентификации, приведены в табл. 2. Как видно из этой таблицы, ни один из "чистых" алгоритмов не является предпочтительным для аппроксимации всех тестовых функций. Специфика данных каждой нечеткой системы и начальные значения ее параметров являются определяющими факторами в вопросе выбора метода идентификации. Однако заранее предсказать это невозможно, поэтому можно предположить, что сочетание различных алгоритмов (гибридный метод) является наилучшим способом решения проблемы параметрической идентификации.

## Заключение

Исследования алгоритма фильтра Калмана для задачи параметрической идентификации нечетких систем позволили сделать следующие выводы: время идентификации при увеличении числа итераций растет линейно; значения диагональных элементов матриц  $\mathbf{R}$  и  $\mathbf{Q}$  не важны, важно их соотношение; соотношение диагональных элементов ковариационных матриц шумов необходимо выбирать, исходя из значений переменных в таблице наблюдений.

Метод Левенберга—Марквардта требует меньше вычислительных ресурсов, однако и движение в сторону экстремума у него не столь уверенное. Самонастройка коэффициента позволяет алгоритму самостоятельно "перезапускаться", если шаг алгоритма становится слишком мал или на протяжении нескольких итераций не было проведено улучшений. Именно такое самообучение не позволяет алгоритму ни при каких условиях ухудшать текущее положение: это одновременно и плюс (будет принято только более удовлетворительное решение) и минус (алгоритм часто застревает в локальных минимумах на время поиска подходящего значения шага).

Алгоритм имитации отжига отличается большими затратами времени. Однако он совершенно не ограничен спецификой данных конкретной нечеткой системы. Дополнительно нужно отметить, что для схемы Больцмана генерация новых параметров должна учитывать не только значение температуры, как указано в классическом описании алгоритма, но и область определения входной переменной. Простота реализации имитации отжига позволяет ему справляться с задачей настройки как антецедентов, так и консеквентов нечетких правил. Кроме того, алгоритм удобно применять в составе гибридного, когда один из методов, основанный на производных, застревает в локальном оптимуме.

Применение гибридного алгоритма позволяет избежать основных недостатков, присущих отдельным алгоритмам, однако выявляет другой недоста-

ток — большое потребление ресурсов. Поэтому для решения реальных задач лучше всего применять не все исследованные методы, а только в паре — метаэвристика (имитация отжига) и один из методов, базирующихся на производных.

#### Список литературы

1. Пашенко Ф. Ф. Введение в состоятельные методы моделирования систем. Часть 1. Математические основы моделирования систем. М.: Финансы и статистика, 2006.
2. Tron E., Margalot M. Mathematical modeling of observed natural behavior: a fuzzy logic approach // Fuzzy Sets and Systems. 2004. Vol. 146. P. 437—450.
3. Ходашинский И. А., Гнездилова В. Ю., Дудин П. А., Лавыгина А. В. Основанные на производных и метаэвристические методы идентификации параметров нечетких моделей // Труды VIII международной конф. "Идентификация систем и задачи управления" (SICPRO '08). М.: Изд. Института проблем управления им. В. А. Трапезникова РАН, 2009. С. 501—529.
4. Dreo J., Petrowski A., Siarry P., Taillard E. Metaheuristics for hard optimization. Methods and case studies. Berlin: Springer, 2006.
5. Ходашинский И. А. Идентификация нечетких систем: методы и алгоритмы // Проблемы управления. 2009. № 4. С. 15—23.
6. Kirkpatrick S., Gelatt C. D., Vecchi M. P. Optimization by Simulated Annealing // Science. 1983. Vol. 220. P. 671—680.
7. Simon D. Training fuzzy systems with the extended Kalman filter // Fuzzy Sets and Systems. 2002. V. 132. P. 189—199.
8. Hongxia Y., Huang Z. H., Qi L. The Convergence of a Levenberg—Marquardt Method for Nonlinear Inequalities // Numerical functional analysis and optimization. 2008. Vol. 29. P. 687—716.
9. Ходашинский И. А. Технология идентификации нечетких моделей типа синглтон и Мамдани // Труды VII международной конф. "Идентификация систем и задачи управления" (SICPRO'08). М.: Изд. Института проблем управления им. В. А. Трапезникова, 2008. С. 137—163.
10. Lisin D., Gennert M. A. Optimal Function Approximation Using Fuzzy Rules // Proc. 18<sup>th</sup> Int. Conf. North American Fuzzy Information Processing Society. 1999. P. 184—188.
11. Ходашинский И. А., Дудин П. А. Идентификация нечетких систем на основе прямого алгоритма муравьиной колонии // Труды Двенадцатой национальной конференции по искусственному интеллекту с международным участием КИИ-2010. Т. 2. М.: Физматлит, 2010. С. 302—310.
12. Teng Y.-W., Wang W.-J., Chiu C.-H. Function approximation via particular input space partition and region-based exponential membership functions // Fuzzy Sets and Systems. 2004. Vol. 142. P. 267—291.
13. Aliyari M., Teshnehlab M. Sh., Sedigh A. K. Novel Hybrid Learning Algorithms for Tuning ANFIS Parameters Using Adaptive Weighted PSO // IEEE International Conference on Fuzzy Systems. 2007. P. 111—116.

УДК 004.934

**И. Н. Паламарь**, канд. техн. наук, проф.,  
**С. С. Юлин**, аспирант,  
ГОУ ВПО "Рыбинская государственная  
авиационная технологическая академия  
имени П. А. Соловьева",  
Ярославская область, г. Рыбинск  
e-mail: irina.palamar@mail.ru

## **Анализ применимости современных методов статистического моделирования для распознавания и понимания речи**

*Дан обзор современных методов распознавания и понимания речи на основе статистического моделирования. Предложена классификационная схема методов. Рассмотрены представители генеративных и дискриминативных моделей и проведен их сравнительный анализ. Обосновано, что методы на основе условных случайных полей являются адекватной базой для разработки системы понимания речи.*

**Ключевые слова:** распознавание речи, дискриминативные и генеративные статистические модели, байесовский и максимальный энтропийный классификатор, скрытые марковские модели, условные случайные поля

### **Введение и постановка задачи**

Речевой сигнал, воспринимаемый человеком, несет в себе неопределенность как на акустическом, так и на лингвистическом уровне. Такой вид неопределенности может быть устранен путем накопления статистической информации. Опыты языковедческой и литературоведческой наук, накопленные к настоящему времени знания о языке и его стилях позволяют утверждать, что одним из реальных оснований применения статистики в изучении языка и речи надо признать объективную присущность языку количественных признаков, количественных характеристик [1].

Применение методов статистической теории распознавания образов стало важным этапом в развитии автоматического распознавания речи (АРР). Использование аппарата математической статистики и теории вероятностей привело к существенному повышению качества распознавания. На сегодняшний день практически все известные системы распознавания речи основаны на статистических методах. Статистические модели в большинстве случаев подходят для моделирования как акустического, так и лингвистического уровней языка. Это означает, что можно составлять иерархические конструкции из таких моделей, в которых модель самого нижнего уровня будет представлять модель фона, модель следующего уровня будет состоять из комбинации моделей предыдущего и представлять

собой модель слов, и так далее вплоть до модели высказываний.

Альтернативой статистическим методам являются методы, основанные на использовании искусственных нейронных сетей (ИНС). Несмотря на потенциальные возможности по классификации кратковременных акустико-фонетических единиц ИНС не стали основной моделью для создания систем АРР. Причиной тому послужил недостаток ИНС, связанный со сложностью моделирования длительных последовательностей наблюдений, таких как, например, слова и целые высказывания, так как эти последовательности обычно обладают сильной временной изменчивостью. Эту проблему не решило даже использование рекуррентных архитектур сети, поскольку эффективность ИНС сильно снижается, когда на входе появляется некоторая динамика, т. е. образы подвержены, например, нелинейным изменениям во времени.

Использование статистических моделей может решить не только проблему дикторонезависимого распознавания речи, но и проблему ее понимания. Для решения этих задач необходим метод, способный учитывать динамику изменения исследуемых объектов (фонов, слогов, слов и т. д.), а также зависимость одних объектов от других. Проведем анализ методов статистического моделирования и определим наиболее адекватные модели для решения задачи распознавания и понимания речи.

### Классификация статистических моделей распознавания речи

Статистические модели адекватны процессу распознавания в некоторой внешней среде, вероятностные характеристики которой могут быть оценены по обучающей выборке. Статистическая неопределенность может проявиться и на этапе выбора информативных признаков, и на этапе описания — определения границ классов, а следовательно, и на этапе классификации — отнесения наблюдения одному из классов. Предполагается, что система распознавания работает в некоторой внешней среде, вероятностные характеристики которой могут быть оценены по обучающей выборке.

В рамках классификации речевых образов на акустическом уровне возможные слова, которые надо распознать, представляются как случайная переменная  $y$ , а каждое конкретное слово является значением этой переменной. Задача классификации состоит в определении того, насколько наблюдаемые данные соответствуют значению переменной  $y$  (классу). В случае речевых образов наблюдаемыми данными является набор спектральных коэффициентов  $x$ , характеризующих то или иное слово. Вектор  $x$  представ-

ляет собой вектор случайных переменных, каждая случайная переменная принимает определенные значения, равные значениям спектральных коэффициентов. Проблема состоит в том, что заранее невозможно определить, какой набор спектральных коэффициентов соответствует тому или иному слову (значению переменной  $y$ ). Задача определения соответствия обучающей последовательности  $x_0$  одному из значений переменной  $y$  является задачей обучения. Один из вариантов обучения может проходить по принципу выбора того значения  $y$ , которое максимизирует значение вероятности  $p(y|x_0)$ . После того как процедура обучения выполнена, задача классификации решается как оценка наблюдаемой последовательности  $x_p$ , т. е. вычисление вероятности  $p(y|x_p)$  для каждого значения  $y$ . Выражение с наибольшим значением вероятности и определяет то слово (значение  $y$ ), которое соответствует наблюдаемым спектральным коэффициентам.

Как видно, для распознавания речевого сигнала необходимо решить два типа задач:

- задачу обучения (настройки вероятностной модели);
- задачу классификации (соотнесения наблюдения с моделью).

Обучение и классификация связаны с выбором некоторой статистической модели. Моделью называется условное или совместное распределение вероятности переменной класса  $y$  и наблюдаемых данных  $x$ . Выбираемая модель оказывает влияние на такие параметры, как объем данных для обучения и требования к вычислительным ресурсам, точность распознавания, способность к обобщению данных из обучающей выборки.

Выделяют два принципиально различающихся подхода к созданию статистических моделей решения задачи распознавания речи, определяющих следующие классы моделей [2, 3]:

- генеративные (*generative*) модели;
- дискриминативные (*discriminative*) модели.

Классификация существующих статистических моделей приведена на рис. 1. Генеративный подход

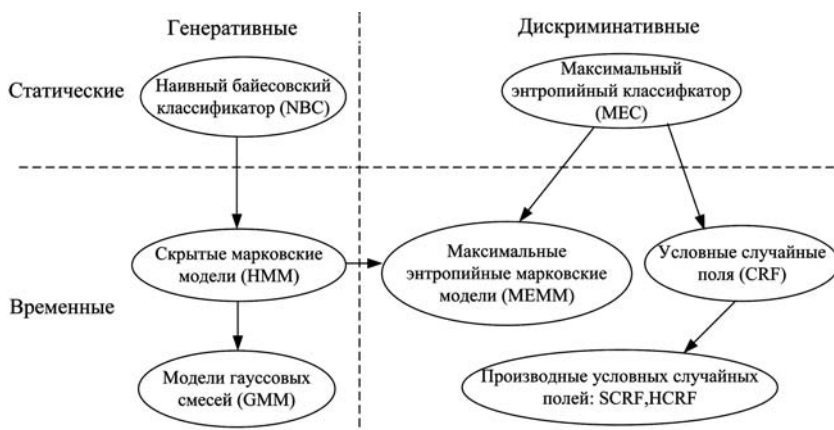


Рис. 1. Классификация статистических моделей

основывается на модели совместного распределения вероятности:  $p(y, \mathbf{x})$  — для статических моделей и  $p(y, \mathbf{x})$  — для временных, а дискриминативный — на модели условного распределения:  $p(y|\mathbf{x})$  — для статических и  $p(y|\mathbf{x})$  — для временных.

Статические модели, в отличие от временных, не позволяют учитывать изменение переменной  $y$  во времени. При этом во временных моделях переменная представляется вектором  $\mathbf{y}$ , каждая компонента которого  $y_i$  — переменная  $y$  в момент времени  $i$ .

### Анализ подходов к распознаванию на примере статических моделей

В статических моделях слова представляются как образы, не изменяющиеся во времени. Обозначим через  $y$  — класс (слово), через  $\mathbf{x} = x_1, \dots, x_m$  — набор наблюдаемых признаков.

В случае **генеративного подхода** классификация выполняется как вычисление совместного распределения вероятности  $p(y, \mathbf{x})$ . Модель, описывающая такое распределение вероятности, называется наивный байесовский классификатор (*naive Bayes classifier* (NBC))

$$p(y|\mathbf{x}) = \frac{p(y)p(\mathbf{x}|y)}{p(\mathbf{x})} \sim p(y, \mathbf{x}) = p(y) \prod_{k=1}^m p(x_k|y). \quad (1)$$

Как видно, в формуле имеет место произведение вероятностей, которое говорит о том, что в модели используется предположение о независимости друг от друга элементов последовательности наблюдения  $\mathbf{x}$ . Это предположение и обеспечивает эквивалентность левой и правой частей данного выражения.

Графически такую модель можно представить в виде направленного графа, изображенного на рис. 2. Направленность указывает на совместное распределение вероятностей, что характерно для генеративных моделей.

В случае **дискриминативного подхода** классификация выполняется как вычисление условного распределения вероятности  $p(y|\mathbf{x})$ . Модель, описывающая такое распределение вероятности, называется **максимальный энтропийный классификатор** (*maximum entropy classifier* (MEC)). Она основывается на прин-

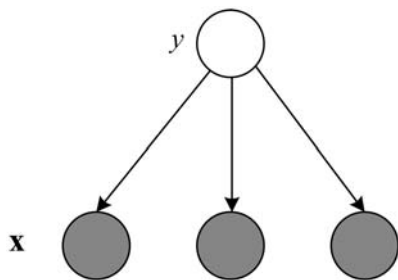


Рис. 2. Наивный байесовский классификатор в виде направленного графа

ципе максимальной энтропии, который утверждает, что если имеется неполная информация о распределении вероятности, то наиболее подходящим распределением вероятности будет то, которое максимизирует энтропию, проявляющуюся в результате ограниченности обучающей последовательности. Модель максимального энтропийного классификатора описывается следующей зависимостью

$$p(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \left\{ \sum_{k=1}^m \lambda_k f_k(y, \mathbf{x}) \right\}, \quad (2)$$

где  $Z(\mathbf{x}) = \sum_y \exp \left\{ \sum_{k=1}^m \lambda_k f_k(\mathbf{x}, y) \right\}$  — нормализующая

константа;  $\lambda_k$  — весовые коэффициенты, изменяющиеся в процессе обучения (формирования класса  $y$ );  $f_k(\mathbf{x}, y) \in \{0, 1\}$  ( $1 \leq k \leq m$ ) — признак-функция, функция, не равная нулю только для одного класса  $y$ , которому соответствует наблюдаемая последовательность.

Графически такую модель можно представить в виде фактор-графа, изображенного на рис. 3. Фактор-граф — это двудольный граф, состоящий из вершин, соответствующих случайным величинам, и вершин, соответствующих факторам [2]. Фактором называется функция  $\Psi_k(y, \mathbf{x}) = p(x_k|y)$ .

Сравнительный анализ представителей двух типов моделей (генеративной — NBC и дискриминативной — MEC) позволил выделить их основные характеристики, которые представлены в таблице.

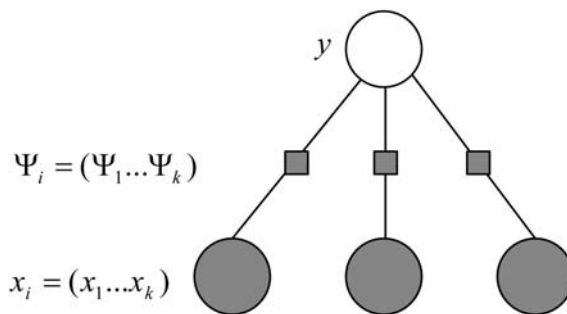


Рис. 3. Максимальный энтропийный классификатор в виде фактор-графа

#### Характеристики генеративной и дискриминативной моделей

Генеративные модели	Дискриминативные модели
Основаны на совместном распределении вероятности $p(y, \mathbf{x})$	Основаны на условном распределении вероятности $p(y \mathbf{x})$
Включают моделирование $p(\mathbf{x})$	Не используют моделирование $p(\mathbf{x})$
Используют независимость наблюдений $\mathbf{x}$	Используют независимость классов $y$ и зависимость наблюдений $\mathbf{x}$

### Анализ временных статистических моделей

Рассмотрим тенденции развития временных статистических моделей генеративного и дискриминативного подходов.

Основным недостатком генеративных моделей, который влияет как на процесс обучения, так и на процесс классификации, является то, что вводится предположение о независимости значений наблюдаемой последовательности, так как часто в прикладных задачах, а тем более в задаче распознавания речи это предположение неверно. На этот недостаток генеративных моделей еще в 1989 г. указывал Л. Р. Рабинер в работе [4]. **Скрытые марковские модели** (*Hidden Markov Model* (НММ)) являются развитием идеи байесовского классификатора. НММ относятся к классу последовательных временных моделей, использование которых наиболее применимо к моделированию динамических случайных сигналов, меняющихся во времени, поэтому обеспечивают высокое качество распознавания речи. Аналитически НММ представляется в виде формулы

$$p(\mathbf{y}, \mathbf{x}) = \prod_{i=1}^m p(y_i|y_{i-1})p(x_i|y_i). \quad (3)$$

Графическое представление изображено на рис. 4.

В НММ значение случайной переменной класса  $y$  представляется как динамический процесс последовательной смены состояний этого значения. Фактически  $y_i$  — это временной срез переменной класса  $y$ . В случае распознавания речи слово (класс) представляется как последовательность смены фонетических конструкций (фонем). Но таким же образом можно представить и сам фон как последовательную смену составляющих его более мелких конструкций. Каждое наблюдение  $x_i$  зависит только от состояния класса  $y_i$  в соответствующий момент времени и не зависит от предыдущих или будущих наблюдений, что значительно ограничивает возможности модели. Из формулы (3) видно, что модель описывает дважды стохастический процесс, моделируя две функции распределения вероятности: вероятность переходов от текущего состояния к следующему и вероятность появления наблюдаемой последовательности в новом состоянии.

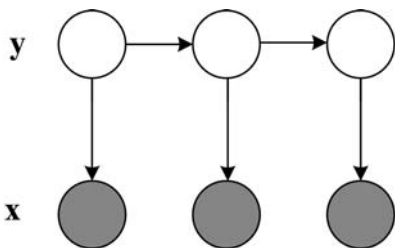


Рис. 4. Скрытая марковская модель в виде направленного графа

**Модель гауссовых смесей** (*Gaussian Mixture Model* (GMM)) представляет собой усовершенствованную НММ, в которой вероятность появления наблюдения и вероятность перехода в следующее состояние аппроксимируются смесью гауссовых функций плотности распределения вероятности. Сама же модель аналогична представленной в формуле (3), за тем лишь исключением, что плотности распределения вероятностей заданы гауссовыми функциями. Такая модель за счет использования непрерывных распределений вероятности хорошо подходит для систем распознавания с большим словарем [5].

В 2000 г. в работе Andrew McCallum была предложена дискриминативная **максимальная энтропийная марковская модель** (*Maximum Entropy Markov Model* (МЕММ)), использующая условное распределение вероятности [6]. Она основывается на использовании максимального энтропийного классификатора в качестве модели, описывающей появление наблюдаемой последовательности в новом состоянии, тогда как НММ использует модель наивного байесовского классификатора. В отличие от НММ, в которой текущее наблюдение зависит только от текущего состояния, в МЕММ текущее наблюдение может также зависеть от предыдущего состояния. Аналитически МЕММ представляется в виде формулы

$$p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^m p(y_i|y_{i-1}) \frac{1}{Z(\mathbf{x})} \exp \left\{ \sum_{k=1}^m \lambda_k f_k(y_i, x_i) \right\}, \quad (4)$$

графическое описание изображено на рис. 5.

МЕММ присущи все особенности описанных выше дискриминативных моделей. Она обладает способностью моделировать динамически изменяемый во времени процесс, в отличие от простого максимального энтропийного классификатора. Недостатком является то, что при использовании дискриминативных моделей невозможно предсказать вероятность появления произвольного наблюдения, так как не моделируется распределение вероятности  $p(\mathbf{x})$ , но в практически решаемой задаче классификации не требуется выполнять такое предсказание, поэтому данный недостаток является несущественным.

Классические вероятностные автоматы, дискриминативные марковские модели, максимальная энтропийная модель и МЕММ обладают существ-

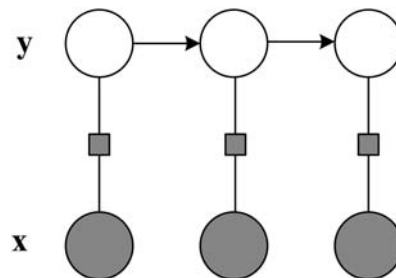


Рис. 5. Максимальная энтропийная марковская модель в виде графа

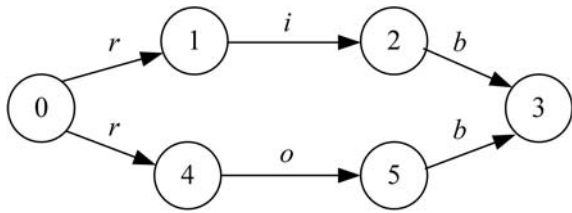


Рис. 6. Иллюстрация проблемы "смещения состояния"

венным недостатком — проблемой "смещения состояния" [7]. Ситуация, при которой возникает эта проблема, наглядно показана на рис. 6. Впервые она была описана Bottou в 1991 г. [8].

На рисунке изображены две модели (последовательности смены состояний) слова, совмещенные вместе для наглядности. Модели соответствуют двум словам "rib" и "rob" [7]. Переходы 1—2 и 4—5 будут равновероятны независимо от того, какому слову будет соответствовать наблюдаемая последовательность. Переход в новое состояние будет осуществлен абсолютно независимо от текущего наблюдения в этом состоянии. Следовательно, модели слов "rib" и "rob" с точки зрения вероятностей переходов, идентичны.

Решение данной проблемы требует моделей, которые формируют сразу целые последовательности состояний, позволяя выбирать из переходов более предпочтительные в зависимости от соответствующих наблюдений. В вышеупомянутом примере переходы из состояния начала имели бы очень слабое влияние на оценку пути, в то время как переходы от состояний 1 и 4 будут иметь более сильное влияние, следовательно, эти два пути уже не будут равновероятными.

Решение проблемы "смещения состояния" носит не только теоретический характер, но также непосредственно относится к задаче распознавания речи. Экспериментальные исследования доказали, что человеческое восприятие речи базируется главным образом на определенных ориентирах в речевом сигнале [9]. Большинство этих ориентиров находится на границах фонетических конструкций. Это доказывает, что наличие хорошего представления вероятности перехода, по крайней мере, столь же важно, как наличие хорошего представления этих фонетических конструкций речи. Исследования акустических ориентиров привели к поискам нового способа моделирования перехода между состояниями, при котором вероятности перехода в моделях должны быть зависимыми от наблюдения [5].

Проблема "смещения состояния" была успешно решена чисто вероятностным способом с появлением в 2001 г. новой дискриминативной **математической модели условных случайных полей** (Conditional Random Fields (CRF)) [7]. Она позволяет учитывать зависимости между предыдущими и текущими наблюдениями без моделирования распределения этих зависимостей, а также зависимость перехода в

следующее состояние от текущего наблюдения. CRF является развитием идеи максимального энтропийного классификатора. Аналитическое описание представляется в виде выражения

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left\{ \sum_{k=1}^m \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\}, \quad (5)$$

где  $Z(\mathbf{x}) = \sum_y \exp\left\{ \sum_{k=1}^m \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\}$  — нормализующая константа, графическое — изображено на рис. 7.

Существенным отличием CRF от MEMM является то, что вероятность перехода в следующее состояние и вероятность появления наблюдаемой последовательности задаются не двумя разными распределениями, а сосредоточены в одной экспоненциальной модели. Математическая зависимость (5) позволяет оценивать вероятность перехода в следующее состояние не только от текущего и предыдущих состояний, но и от соответствующих наблюдений.

Главное ограничение CRF на сегодняшний день — медленная сходимость обучающего алгоритма по сравнению с MEMM и тем более с алгоритмом НММ, для которого существует эффективная процедура обучения.

Существуют разновидности CRF, используемые в распознавании речи: SCRf [9] и HCRF [10].

**SCRf (Segmental CRF)** использует применение дискриминативной модели к последовательностям наблюдений на уровне сегментов. Результат — сегментация данных на фреймы и присвоение обозначения каждому фрейму. SCRf расширяет модель CRF, работая на уровне сегмента, в котором множественные смежные наблюдения могут быть смешаны в сегмент с единственным обозначением, особенности сегмента могут быть извлечены для дальнейшего использования. Модели SCRf при использовании в распознавании речи обеспечивают следующие преимущества:

- наравне с максимальными энтропийными моделями они предлагают удобный способ объединить многочисленные, возможно избыточные признаки наблюдений;

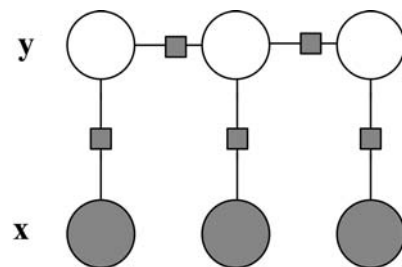


Рис. 7. Модель линейных условных случайных полей в виде фактор-графа



- в отличие от моделей НММ, в которых методы обучения, такие как MMI (максимальной взаимной информации), MPE (максимальной фонемной ошибки) и ML (максимального правдоподобия), применены как отдельный процесс, в SCRF обучение встроено в саму модель.

**HCRF (Hidden CRF)** — условные случайные поля со скрытыми состояниями, которые могут смоделировать подфонемы и спектральные компоненты вместо целых фонем. HCRF выигрывают у НММ в задаче фонемной классификации как с предварительной разметкой границ фонем [11], так и без предварительной разметки [12].

### Заключение

Большинство коммерческих продуктов распознавания речи на сегодняшний день используют в своей основе аппарат НММ. Это продукты фирм "Центр речевых технологий" [13], "Loquendo" [14], "Nokia" [15]. Такая популярность НММ обусловлена их хорошо изученностью и большим опытом практического применения. Однако в настоящее время, как показывают академические тесты, достигнут предел точности распознавания речи в системах, основанных только на статистических моделях.

Модели распознавания речи, основанные на CRF и их производных, с математической точки зрения имеют наилучшую структуру для учета семантики с использованием контекста и решают множество проблем, возникающих при построении подобных систем. Основными недостатками моделей CRF являются:

- плохая сходимость методов обучения модели;
- не разработаны способы реализации моделей для конкретных задач распознавания и понимания речи.

Сравнительный анализ системы фонемного распознавания на основе моделей НММ и HCRF с использованием обучающей выборки из фонемной базы TIMIT показал, что процент ошибок при использовании НММ составляет 30,5 %, тогда как процент ошибок при использовании HCRF — 28,3 % [11]. В работах по лингвистическому моделирова-

нию CRF показывают также очень хороший результат благодаря использованию условных распределений плотности вероятностей, которые в полной мере позволяют учитывать контекст произносимых слов. Наиболее часто на данный момент CRF используются в задаче определения частей речи (*part of speech tagging task*). Эксперименты показывают [7], что процент ошибок при использовании CRF для решения этой задачи составляет 4,27 %, MEMM — 4,81 %, НММ — 5,69 %. Следовательно, именно модель CRF, использующая статистические свойства языка, является адекватной основой для разработки системы понимания речи.

### Список литературы

1. Головин Б. Н. Язык и статистика. М.: Просвещение, 1971. 190 с.
2. Tomanek K. Classical Probabilistic Models and Conditional Random Fields. Dortmund: Faculty of Computer Science Algorithm Engineering, 2007. 32 с.
3. Sutton C., McCallum A. An Introduction to Conditional Random Fields for Relational Learning. USA: MIT Press, 2006. 35 с.
4. Рабинер Л. П. Скрытые Марковские модели и их применение в избранных приложениях при распознавании речи: Обзор // ТИИЭР. 1989. Т. 77. № 2. С. 86—120.
5. Kamal O., Hasegawa M., Levinson S. Gaussian mixture models of phonetic boundaries for speech recognition // IEEE Workshop on Automatic Speech Recognition and Understanding. 2001. С. 33—36.
6. McCallum A., Freitag D., Pereira F. Maximum Entropy Markov Models for Information Extraction and Segmentation // ICML-2000. 2000. С. 591—598.
7. Lafferty J., McCallum A., Pereira F. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data // 18th International Conf. on Machine Learning. 2001. С. 282—289.
8. Bottou L. Une approche théorique de l'apprentissage connexionniste: Applications à la reconnaissance de la parole // Докт. дисс. Université de Paris XI, 1991.
9. Liu S. A. Landmark detection for distinctive featurebased speech recognition // Acoust. Soc. Am., 1996. С. 3417—3430.
10. Zweig G., Nguyen P. SCARF: A Segmental CRF Speech Recognition System. USA: Microsoft Technical Report, 2009. 14 с.
11. Gunawardana A., Mahajan M., Acero A. Hidden Conditional Random Fields for Phone Classification // Interspeech. 2005. 4 с.
12. Sung Y., Jurafsky D. Hidden Conditional Random Fields for Phone Recognition // IEEE Workshop on Automatic Speech Recognition and Understanding. 2009. С. 107—112.
13. Технологии распознавания речи "Центра Речевых Технологий". URL: <http://www.speechpro.ru/techno/recognition> (дата обращения 5.03.2011).
14. Loquendo: ASR. URL: [http://www.loquendo.com/site\\_media/brochures/en/asr.pdf](http://www.loquendo.com/site_media/brochures/en/asr.pdf)
15. Ольсен Е. Способ и устройство распознавания речи // Патент Российской Федерации № 2393549. 2010.

УДК 658.512.2

**А. В. Андрейчиков**, д-р техн. наук, проф.,  
Московский государственный университет  
путей сообщений,

e-mail: alexandrol@mail.ru,

**А. А. Хорычев**, аспирант,

Нижегородский государственный университет,

**О. Н. Андрейчикова**, д-р техн. наук, проф.,  
Московский физико-технический институт

## Экспертная система для начальных стадий проектирования инноваций

*Предложен подход и интеллектуальное программное обеспечение для решения творческих задач прогнозирования и проектирования стратегических инноваций с использованием баз данных по изобретениям и баз знаний создателей новой техники.*

**Ключевые слова:** экспертные системы, эвристическая классификация, прогнозирование новых рынков, изобретательство по аналогии

### Введение

На ранних стадиях проектирования технических систем, целью которых является формирование облика и принципа действия будущего изделия, конструкторы и проектировщики традиционно используют в качестве аналогов и прототипов известные изобретения [1—4]. Удачно выбранные из патентного фонда аналоги и прототипы во многом определяют качество и конкурентоспособность разрабатываемой продукции, так как в них отражены новые оригинальные идеи и их технические реализации, ряд из которых доведен до уровня опытных и серийных образцов. Однако использование в проектировании наукоемких информационных ресурсов по изобретениям встречает серьезные трудности, обусловленные объективными причинами следующего характера. Описания абсолютного большинства новых изобретений таковы, что не несут полной исчерпывающей информации, необходимой конструкторам, для принятия оптимальных обоснованных решений по выбору того или иного прототипа. Основные характеристики технических систем в описаниях к авторским свидетельствам и патентам, как правило, завуалированы, поэтому данный вид информации с позиции многих конструкторов, не имеющих достаточно высокий уровень квалифи-

кации в конкретной предметной области, обладает свойствами "неполноты информации".

Мировой фонд изобретений состоит из классов инновационных решений, объединенных одной главной функцией назначения, например, класс виброизоляторов включает несколько тысяч устройств, предназначенных для защиты объектов от вибрации; класс насосов объединяет тысячи различных типов насосов для перемещения жидкости и т. д. Работа конструкторов даже с отдельно взятыми классами требует от них больших интеллектуальных и временных затрат. В связи с этим качественный многоаспектный анализ всего накопленного мировой практикой опыта по инновационным конструкторским и технологическим решениям становится невыполнимым в условиях отсутствия у создателей новой техники эффективных информационных технологий, что снижает качество проектирования и конкурентоспособность изделий. Использование же для решения задачи проектирования и экспертизы новых технических систем известных автоматизированных систем поиска патентной информации не обеспечивает решения указанной проблемы, поскольку поисковые образы описаний технических решений, хранящиеся в указанных базах данных, составлены из самых общих классификационных признаков без предварительной многоаспектной смысловой обработки патентных документов высококвалифицированными специалистами предметных областей техники.

В этом видится одна сторона проблемы, связанная с качеством и сроками проектирования новой техники. Другая сторона проблемы состоит в том, что разработкой технических систем, относящихся к одному классу, занимаются, как правило, многочисленные отраслевые организации в течение длительного времени, за которое происходит смена нескольких поколений проектировщиков. Каждому такому поколению приходится в течение своей профессиональной деятельности многократно обращаться к одним и тем же информационным массивам изобретений. В связи с этим проектировщикам уже сейчас требуется такая информационная технология, которая помогла бы им осваивать опыт по определенному классу технических систем с минимальными временными и интеллектуальными затратами.

Для решения данной проблемы предлагается на начальных стадиях проектирования использовать экспертную систему, основанную на фонде изобретений и выполняющую функции информационно-поисковой системы; системы поддержки при-

нятия решений по многим критериям и средства для систематизации знаний и данных в целях прогнозирования новых перспективных направлений поиска инноваций [5—12].

### База знаний экспертной системы

База знаний экспертной системы по изобретениям представляет собой хранилище структурированной информации, различной по природе и форме представления. База знаний включает как процедурные, так и декларативные знания. Эти знания можно представить в виде рис. 1, где  $Z_0$  — множество знаний, общепринятых в области анализа и проектирования технических инноваций, описанных в научно-технической литературе (так называемые "общие знания"), оно характеризуется наличием теории по методам синтеза и принятия решения;  $Z_l$  — множество знаний, приобретенных в процессе профессиональной деятельности разработчиками новой техники и создателями автоматизированных экспертных систем анализа и синтеза новых объектов техники ("личные знания").

Пересечение множеств  $Z_0$  и  $Z_l$  представляет собой канонизированную часть  $Z_k$  личных знаний, в которых нет расхождений между отдельными специалистами и экспертами.

В канонизированную часть знаний входят знания о закономерностях строения и развития технических систем, иерархическом способе описания функций и структур систем, эвристических методах синтеза и эвристических правилах поиска новых принципов действия.

Множество  $Z_l$  содержит подмножество  $Z_p$ , которое не имеет пересечения с  $Z_0$ . Это подмножество

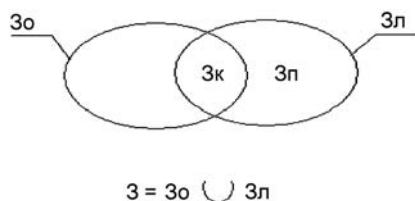


Рис. 1. Знания, используемые в экспертной системе

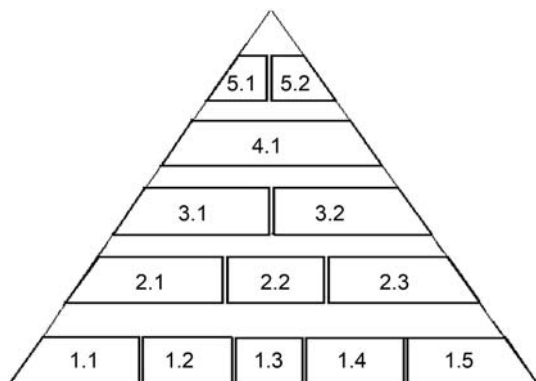


Рис. 2. Пирамидальная стратифицированная структура базы знаний экспертной системы систематизации и прогнозирования инноваций

представляет ту часть личных знаний, которая обусловлена профессиональным опытом, мастерством и интуицией. Подмножество  $Z_p$  в рамках рассматриваемой экспертной системы содержит:

- модернизированные и адаптированные к исследуемому классу технических объектов эвристические методы поиска новых принципов действия и структур технических решений;
- выявленные новые закономерности строения и развития технических систем и специализированные эвристические правила;
- оригинальные формы формализованного описания логических процедур синтеза от поиска функциональных требований к принципам действия и от них к структурам технических решений;
- систематизированные для конкретных классов технических систем и представленные в виде словарей, списков, каталогов функциональные структуры, известные и прогнозируемые принципы действия и структуры конструкции.

Степень эффективности экспертной системы тем выше, чем больше удельный вес знаний  $Z_0$  по отношению к знаниям  $Z_k$  и  $Z_p$ .

Используемые в экспертной системе по инновационным техническим решениям данные и знания упорядочены пирамидальной стратифицированной структурой, представленной на рис. 2.

Разработка базы знаний экспертной системы осуществляется от основания к вершине пирамиды.

Рассмотрим состав пирамиды. Самый нижний — первый уровень пирамиды — образуют общеизвестные знания, которые извлекаются из специальной научно-технической литературы по проектированию новой техники, физическим эффектам и явлениям, а также патентных источников:

- о принципах многоуровневого иерархического описания технических систем (1.1);
- физических эффектах и явлениях (1.2) [13];
- структуре и принципах функционирования технических систем (1.3);
- общих закономерностях строения и развития техники (закон соответствия функций, закон эволюции технических систем и др.) (1.4);
- об информационной модели внешней среды функционирования технических систем, включающей факторы взаимодействия последних с технической, физической, биологической средой, человеком и обществом (1.5).

На втором уровне пирамиды структурируются знания, описывающие единичные технические системы, каждая из которых отображается функциональной структурой (2.1), техническим принципом действия (2.2) и физическим принципом действия (2.3).

В основе функциональных структур лежит понятие функции технической системы, получившее две наиболее распространенные формы формального описания:

- 1)  $F = (D, G, H)$ ;
- 2)  $F = (Am, E, Cm)$ ,

где  $F$  — функция;  $D$  — действие, производимое технической системой;  $G$  — объект, на который направлено действие;  $H$  — особые условия и ограничения;  $Am$ ,  $Cm$  — соответственно входной и выходной потоки энергии, вещества и сигналов;  $E$  — наименование элементарной физической операции по превращению  $Am$  в  $Cm$  (излучение/поглощение, проводимость/изоляция, сбор/рассеивание, проведение/не проведение, преобразование/обратное преобразование, увеличение/уменьшение, изменение направления, выравнивание/колебание, связь/прерывание, соединение/разъединение, накопление/выдача).

Технический принцип действия представляется конструктивными и функциональными признаками в виде иерархического графа, вершины которого описывают иерархическую соподчиненность элементов (отношение "целое—часть") и направление действия функций.

Физический принцип действия технической системы представляется иерархической или сетевой моделью. При иерархическом представлении вершинами графа являются физические эффекты, на основе которых функционируют конструктивные элементы или техническая система в целом. Дуги графа в данном случае указывают иерархическую соподчиненность между функциями. При сетевом представлении вершинами являются физические эффекты, а дуги указывают входные и выходные потоки энергии, вещества или сигналов. Каждый физический эффект формально представляется трехкомпонентной структурой:  $\Phi Э = (A, B, C)$ , где  $A$ ,  $C$  — соответственно входной и выходной потоки энергии, вещества или сигналов;  $B$  — физический объект, обеспечивающий преобразование  $A$  в  $C$ .

Третий уровень пирамиды образован на основе обобщения знаний второго уровня и представлен обобщенными функциональной (3.1) и конструктивной (3.2) моделями конкретного класса технических объектов, для которого разрабатывается экспертная система. Здесь обобщенные модели представлены И-ИЛИ-графами. Вершины графа типа "И" объединяют различные основания деления понятий (признаков), характеризующих один объект в различных аспектах. Вершины типа "ИЛИ" объединяют значения признаков, т. е. альтернативы одного деления понятия.

Для описания множества технических решений, входящих в один класс, используется несколько сотен или тысяч функциональных или конструктивных признаков и их значений. В частности, класс виброзащитных устройств, включающий более  $10^4$  описаний изобретения, потребовал построения функциональной модели, включающей более 500 признаков и их значений, и структурной модели, включающей 2500 признаков и их значений. Функциональная и структурная модели основаны на наиболее важных и устойчивых атрибутах — функциях и

структурах технических объектов и их элементов. Поэтому эти две модели определяют функционально-структурную систематику конкретного класса технических объектов. Под систематикой понимается область знаний, в рамках которой решаются задачи упорядоченного определенным образом обозначения и описания совокупности объектов, образующих некоторую сферу реальности. Необходимость систематики возникает во всех науках, которые имеют дело со сложными, внутренне разветвленными и дифференцированными системами объектов. Самая совершенная систематика представляет собой естественную систему объектов, основанную на объективных законах. В технической сфере создаются исключительно искусственные систематики и классификации из соображения полезности и удобства в решении определенных практических задач — исследовательского и поискового характера.

На четвертом уровне на основе функционально-структурной систематики разрабатывается семантическая сеть (4.1), которая позволяет реализовать удобный диалоговый интерфейс между экспертной системой и ее потенциальными пользователями. Семантическая сеть позволяет использовать множество причинно-следственных отношений, указывающих последовательность шагов исследователю, формирующему в диалоговом режиме с экспертной системой различные поисковые предписания (технические задания) на поиск необходимых описаний изобретений в базах данных. При этом каждое сформированное поисковое предписание с использованием семантической сети не содержит взаимоисключающих признаков и их значений. Причинно-следственные отношения представлены простейшими продукционными правилами типа: "если выбрано значение признака  $Z$ , то следовать на признак  $\Pi$ ". Установление всех причинно-следственных отношений между признаками и значениями признаков, входящими в функционально-структурную систематику класса изобретений, осуществляется высококвалифицированным специалистом или группой специалистов в этой области техники или технологии.

На пятом уровне формируются знания, полученные в результате исследований, выполненных проектировщиком, проводившим изучение с различных точек зрения баз данных изобретений. Например, основными результатами работы экспертной системы здесь могут быть: множество прогностических систематик (5.1) исследуемого класса, сформированных с различной степенью конкретизации функциональных или конструктивных признаков, и/или морфологические таблицы (5.2), сформированные в результате анализа нескольких классов технических систем, различающихся функциональным назначением [14, 15].

Пополнение экспертной системы новыми знаниями осуществляется по двум направлениям. Знания, расположенные в стратифицированной

пирамиде от первого до четвертого уровня, пополняются в основном инженерами по знаниям и специалистами конкретных предметных областей.

Знания, отнесенные к пятому уровню пирамиды, пополняются в результате решения в автоматизированном режиме конкретных задач, связанных с систематизацией, прогнозированием и синтезом новых рациональных технических систем для различных научно-исследовательских и проектных ситуаций.

### Исследовательские сценарии

Работа исследователя с экспертной системой по изобретениям осуществляется по принципиально различающимся сценариям.

**Первый сценарий** используется при решении задач поиска в базе данных описаний аналогов и прототипов технических систем по функциональным и структурным (конструктивным) признакам. В этих задачах функционально-структурный образ искомой технической системы строго определен в поисковом предписании (техническом задании). Данный образ описан множеством функционально-структурных признаков. Здесь мы имеем такую ситуацию, когда ответ на задачу содержится в значительной степени в условиях задачи, т. е. исследователь знает, что хочет найти в базе данных. Например, если в поисковом предписании указан признак "пневматический виброизолятор", то отсюда следует, что в базе данных исследователь хочет найти именно этот тип виброизолятора, а не торсионный, динамический или ударный.

Автоматизированный поиск в общем виде представляет процедуру пошагового усечения множества классификационных признаков и выделение соответствующего ему в базе данных множества описаний изобретений (рис. 3, а). Чем больше шагов делает исследователь по отбору классификационных признаков в поисковое предписание, тем больше он сужает число релевантных ему описаний технических объектов, хранящихся в базе данных (площадь эллипсов, изображенных в правой части рис. 3, а, пропорциональна числу технических систем). Отбор признаков и значений признаков при работе по этому сценарию осуществляется исследователем эвристическим интуитивным методом (на основе интуиции и опыта).

Поиск в базе данных описаний технических систем, релевантных поисковому предписанию, может осуществляться по усмотрению исследователя по различным критериям. Критерии релевантности основаны на частичном и полном (1) совпадении значений признаков поискового

предписания и поискового образа или на основе какой-либо меры сходства (2) [16, 17]:

$$m(S_1 \cap S_2) \geq \alpha(S_2). \quad (1)$$

Мера сходства Чекановского—Серенсена

$$C(S_1, S_2) = 2m(S_1 \cap S_2)/m(S_1) + m(S_2), \quad (2)$$

обладает следующими свойствами:

$$0 \leq C(S_i, S_j) \leq 1 \text{ для } i \neq j;$$

$$C(S_i, S_j) = 1 \text{ для } i = j;$$

$$C(S_i, S_j) = C(S_j, S_i).$$

В приведенных выше формулах (1) и (2) приняты следующие обозначения:  $S_1, S_2$  — множество значений признаков текущего сравниваемого поискового образа описания технического решения, хранящегося в базе данных, и поискового предписания соответственно;  $\alpha$  — пороговый коэффициент, задаваемый исследователем и устанавливающий значение необходимой доли совпадения значений признаков ( $0 < \alpha \leq 1$ );  $m()$  — мощность множества.

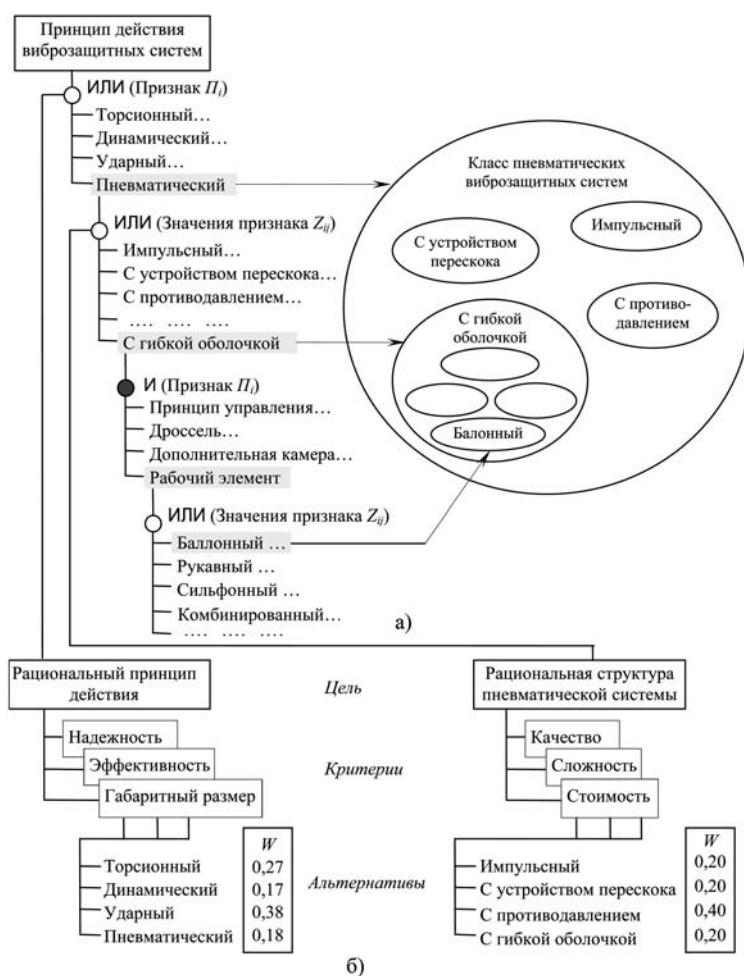


Рис. 3. Фрагмент структурной иерархической модели, являющейся частью систематики класса виброзащитных систем;  $W$  — интегральный вектор приоритетов альтернатив

Мера сходства рассчитывается на основании бинарной матрицы

$$[B] = \|x_{ij}\|,$$

где  $i = \overline{1, p}$ ;  $j = 1, 2$ ;  $p$  — число строк матрицы;  $j = 2$  — число столбцов матрицы;  $x_{ij} = \{0, 1\}$ . Столбцы матрицы представляют объединение значений признаков, принадлежащих текущему рассматриваемому поисковому образу технического решения и поисковому предписанию.

Для практических вычислений критериев релевантности  $K_p$  на основе матрицы  $[B]$  необходимо пользоваться следующими выражениями:

$$K_{p1} = \sum_{i=1}^n (x_{i1} x_{i2}) / \sum_{i=1}^n x_{i2} \geq \alpha, (0 < \alpha \leq 1);$$

$$K_{p2} = 2 \sum_{i=1}^n (x_{i1} x_{i2}) / \sum_{i=1}^n x_{i1} + \sum_{i=1}^n x_{i2} \geq \alpha, (0 < \alpha \leq 1),$$

где  $n$  — число значений признаков поискового предписания.

Пример вычисления значений коэффициента релевантности  $K_{p2}$ , основанного на мере сходства Чекановского—Серенсена, приведен в таблице.

$$K_{p2} = 2(1 + 0 + 0 + 1 + 1)/5 + 3 = 6/8 = 0,75.$$

Значение признака	Бинарная матрица	
	Поисковое предписание $x_{i1}$	Поисковый образ $x_{i2}$
$Z_1$ — пневматический	1	1
$Z_2$ — резинокордный	1	0
$Z_3$ — двухкамерный	1	0
$Z_4$ — управляемый	1	1
$Z_5$ — частотно-зависимый	1	1

**Второй сценарий** направлен на поиск в базе данных рациональных технических решений по предварительно оптимизированному признаковому образу (поисковому предписанию). Поисковое предписание, состоящее из множества функциональных и конструктивных признаков, оптимизируется последовательно по нисходящему принципу от более общих признаков к частным. Отбор признаков в каждой вершине типа "ИЛИ" (рис. 3, б) осуществляется с использованием системы принятия решений, основанной на методе анализа иерархий. В поисковое предписание включается значение признака, имеющее максимальное значение в глобальном (интегральном) векторе приоритетов. Формирование требуемого набора критериев, построение иерархии критериев и экспертное оценивание проводится исследователем в каждой вершине "ИЛИ" с учетом конкретной решаемой задачи. Таким образом, здесь, в отличие от первого сценария, поисковый образ

(поисковое предписание) заранее не определен, а оптимизируется (формируется) в процессе решения задачи [17—19].

**Третий сценарий** предназначен для поиска рациональных технических решений, хранящихся в базе данных, по множеству критериев качества методами принятия решений, ориентированными на работу в условиях неопределенности. В этом случае поисковый функционально-конструктивный образ не формируется, а экспертная система работает в режиме системы поддержки принятия решений с большим числом альтернатив. Оценка альтернатив в данном случае проводится методом стандартов, если используется метод анализа иерархий, или одним из методов, основанным на теории нечетких множеств.

**Четвертый сценарий** обеспечивает построение в автоматизированном режиме прогнозирующих информационных моделей. Информационные модели строятся двух видов и имеют табличную структуру. Первый вид моделей основан на использовании функциональных и конструктивных (структурных) признаков, во втором виде моделей используются функционально-конструктивные признаки и физические эффекты. В первом случае с помощью информационных моделей прогнозируются перспективные направления развития и поиска новых структур технических решений, а во втором случае — выявляются новые физические принципы действия технических систем.

**Пятый сценарий** дает возможность упорядочивать технические системы, извлекаемые из базы данных, в виде эволюционных цепочек, позволяющих выявлять закономерности развития классов, родов и видов технических систем.

**Шестой сценарий** предназначен для построения прогностических морфологических таблиц (матриц) и структурных матриц.

### Построение систематики технических решений для выявления инновационных ниш

Систематизация структур технических инноваций основана на функционально-структурной табличной модели представления данных и системе предпочтений, формируемой методами принятия решений, ориентированными на решение задач в условиях неопределенности.

В функционально-структурных таблицах, которые строятся специалистом (экспертом) в автоматизированном режиме, систематизируются различные функции, выполняемые техническими системами в виде множеств функциональных признаков и их значений (наименование строк), и структурные свойства систем в виде множеств конструктивных признаков и их значений (наименование столбцов). На пересечении строк и столбцов в ячейках таблицы фиксируются известные описания инновационных решений (изобретения), извлекаемые из базы дан-

ных, и их поисковые образы, состоящие из множества функциональных и конструктивных признаков.

Функционально-структурные таблицы строятся с различной степенью детализации функциональных и конструктивных признаков. Чем более детально описаны функция и структура, тем меньшее число инновационных решений попадает из базы данных в соответствующую ячейку таблицы.

Функционально-структурные прогнозирующие таблицы отдельных классов технических систем являются открытыми, т. е. могут быть дополнены вновь выявленными функциональными и конструктивными признаками, а также техническими объектами.

Качество функционально-структурных таблиц зависит от уровня профессиональной подготовки специалистов, которые занимаются их построением. В свою очередь, специалисты в процессе формирования и исследования прогнозирующих таблиц обогащаются новыми знаниями, позволяющими им существенно превзойти профессиональный уровень других специалистов, работающих в аналогичных областях техники.

Систематизация существующей информации по соответствующей тематике в виде функционально-структурной таблицы позволяет исследователю использовать всю массу знаний, накопленную предшествующими поколениями ее создателей. Эта масса знаний по классу технических решений сконцентрирована и классифицирована:

- ✓ по функционально-структурным признакам, являющимся наиболее важным источником информации для создателей новой техники;
- ✓ по числу патентов и авторских свидетельств, указанных в определенной клетке таблицы, что позволяет оценивать степень разработанности проблемы, долю национального вклада и различных стран в разработку задачи создания данной разновидности технического решения.

Обобщение этих данных дает возможность сделать вывод о целесообразности учета особенностей конструктивного и технологического подхода к разработке технических решений данной разновидности различными научно-проектными организациями как внутри страны, так и за рубежом.

Особый интерес при прогнозировании представляют "пустые" клетки таблицы, в которых не указано ни одного источника информации (изобретения или патента).

Во-первых, из анализа специальной научно-технической литературы можно установить, что технические решения, относящиеся к некоторым "пустым" клеткам таблицы, неработоспособны или неэффективны. Эти клетки таблицы отмечаются с указанием в них библиографических данных публикаций, на основании которых сделан этот вывод.

Во-вторых, оставшиеся пустые клетки таблицы можно подвергнуть более детальному анализу в целях

прогнозирования новых технических решений, относящихся к этим клеткам. В этом случае использование функционально-структурных таблиц позволяет преодолевать психологическую инерцию, в силу которой при бессистемном переборе вариантов разработчик не видит того, как были решены аналогичные проблемы в других подклассах технических решений, расположенных в соседних ячейках таблицы.

### Выявление инновационных ниш

Фрагмент функционально-структурной таблицы, систематизирующий технические системы, принадлежащие классу виброзащитных технологий, приведен на рис. 4. В ячейках таблицы изображены наиболее типовые конструкции, характеризующие подмножество технических решений, попавшее в них из базы данных. В функционально-структурной таблице обнаруживаются незаполненные ячейки ("белые пятна"), сигнализирующие о не открытых инновационных решениях.

В этом плане функционально-структурные таблицы напоминают систематику химических элементов в виде периодической таблицы. Поэтому систематизация структур технических инноваций в предложенном виде позволяет, с одной стороны, выявить новые направления поиска технических решений с уникальными свойствами, а с другой — является мощным подсказчиком идей для их синтеза по аналогии с уже созданными решениями, стоящими в соседних ячейках таблицы.

В верхнем левом углу каждой заполняемой ячейки таблицы указывается экспертная оценка степени перспективности рассматриваемого подмножества решений для применения в различных отраслях техники (для определенности, например, верхний левый квадрант характеризует амортизирующие подвески автомобилей, нижний левый квадрант — виброзащитные системы радиоэлектронной аппаратуры летательных аппаратов, верхний правый квадрант — рессорное подвешивание железнодорожных экипажей, нижний правый квадрант — виброзащиту человека — оператора транспортных машин). Оценка проведена по следующей шкале: 0 — невозможность применения технических решений в соответствующей отрасли; 3, 5, 7 и 9 — соответственно слабая, средняя, сильная и очень сильная перспективность применения. Здесь возможно применение многокритериальной оценки перспективности или эффективности технических решений. В верхнем правом углу ячеек таблицы указывается число охранных документов, патентов, которое пропорционально площади приведенных кругов.

Анализ функционально-структурной таблицы, как было указано выше, позволяет в первую очередь выявить "белые пятна" в изучаемой области технических систем, устранение которых дает воз-

возможность получить принципиально новые решения на уровне изобретений, близких к пионерным.

Приведенный пример функционально-структурной таблицы для класса виброзащитных систем (рис. 4) показывает, что ни один представленный в ней род (наименование столбцов таблицы) не удовлетворяет всем функциональным признакам (наименование строк таблицы). В большей степени этому требованию отвечает род пневматических

виброизоляторов. Это говорит о том, что последние обладают более широкими функциональными возможностями по сравнению с остальными родами виброизоляторов.

### Выводы

- Предложен подход для решения задач систематизации и прогнозирования направлений развития инноваций на основе автоматизированных эвристических классификаций.
- Разработаны эвристические классификации для класса виброзащитных технологий.
- Приведены примеры практического использования эвристических классификаций для прогнозирования инновационных решений.

### Список литературы

1. **Артемьев Е. И., Кравец Л. Г.** Изобретения, уровень техники, управление. М.: Энергия, 1977. 239 с.
2. **Кадровский О. Н.** Новые задачи автоматизации процессов обработки патентной информации // Вопросы изобретательства. 1982. № 7. С. 35–40.
3. **Основы инновационного менеджмента: учеб. пособие /** Под ред. проф. В. В. Коссова. М.: Магистр, 2009. 429 с.
4. **Конов Ю. П., Мазнев С. Ф.** Ускорение использования изобретений (Прогнозирование, эффективность). М.: Машиностроение, 1989. 152 с.
5. **Андрейчиков А. В., Дворянкин А. М., Половинкин А. И.** Об использовании экспертных систем в автоматизированном банке инженерных знаний для поискового проектирования и конструирования // Известия АН СССР. Техническая кибернетика. 1989. № 1. С. 183–189.
6. **Глушков В. М.** Кибернетика, изобретательство и ЭВМ // Изобретатель и рационализатор. 1973. № 2. С. 4–7.
7. **Диксон Д.** Проектирование систем: изобретательство, анализ, принятие решений: Пер. с англ. М.: Мир, 1969. 440 с.
8. **Искусственный интеллект: В 3-х кн. Кн. 1. Модели и методы: Справочник /** Под ред. Э. В. Попова. М.: Радио и связь, 1990. 464 с.
9. **Искусственный интеллект: В 3-х кн. Кн. 2. Модели и методы: Справочник /** Под ред. Д. А. Поспелова. М.: Радио и связь, 1990. 304 с.
10. **Представление и использование знаний: Пер. с япон. /** Под ред. Х. Уэно, М. Исидзука. М.: Мир, 1989. 220 с.
11. **Андрейчиков А. В., Андрейчикова О. Н.** Интеллектуальные информационные системы. М.: Финансы и статистика, 2006. 424 с.
12. **Dorokhov I. N., Andreichikov A. V.** Invention Expert System for Development of new technologies // 3-th International Conference on Information Technologies & Telecommunication, IT & TC-2007, Ganja, Azerbaijan, October 4–6. 2007. P. 17–22.
13. **Камаев В. А., Фоменков С. А., Сипливая М. Б., Колесников С. Г.** Физические явления из материалов заявок на открытия по физике. Волгоград: Волг ГТУ, 1995. 224 с.
14. **Одрин В. М.** Метод морфологического анализа технических систем. М.: ВНИИПИ, 1989. 312 с.
15. **Тринг М., Лейтуэйт Э.** Как изобретать?: Пер. с англ. М.: Мир, 1980. 272 с.
16. **Андреев В. Л.** Классификационные построения в экологии и систематике. М.: Наука, 1980. 143 с.
17. **Андрейчиков А. В., Андрейчикова О. Н.** Анализ, синтез, планирование решений в экономике. М.: Финансы и статистика, 2004. 464 с.
18. **Макеев С. П., Шахнов И. Ф.** Упорядочение объектов в иерархических системах // Известия АН СССР. Техническая кибернетика. 1991. № 3. С. 29–46.
19. **Саати Томас Л.** Принятие решений при зависимостях и обратных связях: Аналитические сети: Пер. с англ. / Науч. ред. А. В. Андрейчиков, О. Н. Андрейчикова. М.: Изд-во ЛКИ, 2008. 360 с.

Функция F(i)	Конструкторские реализации виброизоляторов			
	Торсионный	Резиновый	Комбинированный	Пневматический
F <sub>1</sub> 		$\frac{013}{010}$ 	$\frac{010}{010}$ 	$\frac{517}{715}$ 
F <sub>2</sub> 	$\frac{313}{515}$ 	$\frac{013}{310}$ 	$\frac{313}{313}$ 	$\frac{517}{715}$ 
F <sub>3</sub> 				$\frac{517}{715}$ 
F <sub>4</sub> 		$\frac{013}{310}$ 	$\frac{315}{315}$ 	
F <sub>5</sub> 	$\frac{513}{517}$ 	$\frac{313}{310}$ 	$\frac{713}{313}$ 	$\frac{517}{713}$ 
F <sub>6</sub> 	$\frac{313}{515}$ 	$\frac{315}{010}$ 	$\frac{315}{515}$ 	$\frac{517}{717}$ 
F <sub>7</sub> 				$\frac{517}{717}$ 
F <sub>8</sub> 		$\frac{015}{310}$ 	$\frac{013}{710}$ 	$\frac{017}{710}$ 
F <sub>9</sub> 		$\frac{015}{010}$ 		

Рис. 4. Фрагмент функционально-структурной таблицы для прогнозирования новых направлений поиска инноваций: F<sub>1</sub> — несимметричная характеристика; F<sub>2</sub> — регулируемая квазилинейная; F<sub>3</sub> — многоступенчатая; F<sub>4</sub> — скачкообразная; F<sub>5</sub> — кусочно-линейная; F<sub>6</sub> — отрицательная жесткость; F<sub>7</sub> — частотно-зависимая; F<sub>8</sub> — амплитудный "провал"; F<sub>9</sub> — "проскок"



**С. Г. Бобков,**

д-р техн. наук, ст. науч. сотр., зав. отделением,  
e-mail: bobkov@cs.niisi.ras.ru,

**С. О. Задябин,** зав. отд.,

e-mail: zserg@cs.niisi.ras.ru,

**А. В. Крыницкий,** зав. сектором,

e-mail: alexis@cs.niisi.ras.ru,

**Е. Е. Новожилов,** мл. науч. сотр.,

e-mail: eugene@cs.niisi.ras.ru,

НИИ системных исследований РАН, г. Москва

## Верификация модели контроллера ведущего интерфейса USB 2.0

Обобщается опыт применения методологии VMM в проекте верификации блока СНК 1890ВМ6Я — контроллера ведущего универсальной последовательной шины USB 2.0 с регистровой моделью стандарта ЕHCI.

**Ключевые слова:** цифровая схемотехника, верификация СБИС, СНК, тестирование методом "черного ящика"

### Введение

Возрастающая функциональная сложность моделей передач на уровне регистров (RTL-моделей) в функциональных блоках СБИС не позволяет в настоящий момент при их верификации предусмотреть все возможные краевые ситуации, в них возникающие. Поэтому традиционный подход с разработкой отдельных тестов для каждой функции проверяемого блока уже не отвечает современным требованиям, так как отдельные сценарии, приводящие к ошибке, могут просто не быть предусмотрены разработчиками тестов.

Для решения данной проблемы были разработаны методы формирования случайных тестовых воздействий на RTL-модель СБИС, которые с большей вероятностью при достаточных временных и вычислительных ресурсах позволяют охватить ошибочные краевые ситуации, возникающие в модели. Для внедрения указанных методов научным сообществом был разработан стандарт *System Verilog* (IEEE Std. 1800—2005) — стандарт на единый язык для описания и верификации аппаратуры. Необходимо выделить следующие особенности языка:

1. Реализована поддержка методов ООП и формальной верификации.

2. Добавлены операторы для формирования псевдослучайных последовательностей на основе внешних ограничений.

Это позволило производителям САПР, предлагающим продукты для верификации RTL-моделей, разработать методологии построения тестового окружения на основе набора базовых классов, реализо-

ванных на языке *System Verilog* для верификации произвольного блока СБИС с применением методики формирования случайных тестовых воздействий.

Также данный подход позволяет стандартизировать типовой состав тестового окружения, что, в свою очередь, снижает трудозатраты по верификации СБИС за счет заимствования. Часть необходимых функций для каждого компонента тестовой системы уже реализуется в базовых классах, и универсальный состав тестового окружения позволяет создавать переносимые тестовые компоненты VIP (*Verification IP*). Таким образом, отдельные тестовые компоненты стало возможным переносить из проекта в проект.

### Описание методологии VMM

Для верификации блока контроллера USB 2.0 была выбрана открытая методология верификации VMM (*Verification Methodology Manual*), разработанная компанией *Synopsys Inc*.

Методология VMM предполагает разделение тестового окружения на уровни, на каждом из уровней присутствует фиксированный набор компонентов (рис. 1).

Рассмотрим эти уровни:

1. *Уровень тестов* — здесь задаются параметры работы генератора сценариев, проводится выбор конкретного сценария, а также конфигурации тестируемого устройства для выбранного сценария.

2. *Уровень сценариев* — здесь располагаются генераторы последовательностей транзакций, которые формируют конкретные последовательности псевдослучайных транзакций на основе уточняющих ограничений, заданных в программе пользователем.

3. *Уровень транзакций* — здесь располагаются компоненты драйвер, монитор и координатор. Информация между ними передается в виде высокоуровневых транзакций (объектов, наследованных от общего класса `vmm_data`).

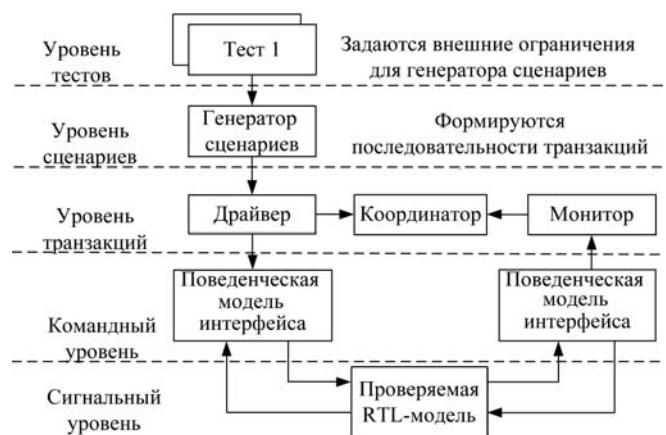


Рис. 1. Модель тестового окружения, предлагаемая методологией VMM

3.1. Компонент драйвер, получив транзакции от генератора сценариев, вызывает соответствующие процедуры поведенческой модели интерфейса, а затем передает полученную транзакцию в координатор.

3.2. Монитор, получив информацию об отклике устройства от поведенческой модели интерфейса, передает эту информацию в виде транзакции координатору.

3.3. Координатор на основе данных об исходной транзакции и полученном отклике проверяет правильность реакции тестируемой RTL-модели. Кроме того, в его функции входит задача сбора информации о тех тестовых воздействиях, которые проводил драйвер, для последующего анализа пользователем при необходимости.

4. *Командный уровень* — здесь располагаются поведенческие модели интерфейсов, содержащие процедуры для формирования полученных от драйвера воздействий и для передачи монитору информации об отклике проверяемой модели.

5. *Сигнальный уровень* — на данном уровне выполняется подключение сигналов поведенческой модели интерфейса к внешним портам RTL-модели.

### Описание структуры тестовой системы

Структура тестовой системы контроллера USB 2.0 стандарта EHCI выстроена согласно методологии VMM, что делает ее модульной и переносимой.

Применяемая методология была расширена путем применения принципов ООП к классам сценариев, описывающих последовательности формируемых дескрипторов (путем формирования иерархии сценариев, когда сценарий более высокого уровня вызывал сценарии более низкого уровня). Это на 40 % снизило трудозатраты по разработке сложных сценариев, описывающих крайние ситуации работы контроллера (по метрике объем кода) и увеличило ясность их описания.

Основная задача проверяемого контроллера USB 2.0 заключается в преобразовании данных, описываемых дескрипторами, расположенными в памяти, доступной по магистрали AMBA 3 AXI, в циклы параллельной шины ULPI (USB Low Pin Interface) с соблюдением предусмотренных стандартом EHCI правил обработки дескрипторов, а также временных интервалов, установленных стандартом магистрали USB 2.0. Поэтому основным случайным компонентом транзакций стал EHCI-дескриптор, формируемый случайным образом на основе ограничений конкретного теста и сценария. Это привело к необходимости разработки дополнительного компонента, преобразующего одни случайные транзакции (дескрипторы) в другие (описывающие пакеты шины AMBA 3 AXI).

Стандарт EHCI содержит пять видов дескрипторов, поэтому стандартный подход VMM был несколько изменен — в базовом классе ehci\_descriptor,

являющемся элементарной транзакцией в терминах VMM, было сделано несколько подклассов, каждый из которых описывает конкретные поля одного из видов дескриптора и передаваемые с ним данные.

Для краткости изложения некоторые внесенные новшества в отклики моделей USB-устройств, такие как механизмы инъекции ошибок, подробно рассматриваться не будут. Общий вид тестовой системы приведен на рис. 2.

Описание основных компонентов тестовой системы.

1. Генераторы сценариев для периодического и асинхронного расписаний формируют на выходе массивы транзакций — объектов класса ehci\_descriptor, с подклассами для указанного вида передач.

2. Блок абстрактного описания регистровой модели контроллера формирует наборы транзакций, описывающих операции записи и чтения регистров проверяемой модели.

3. Блок axi\_traffic\_translator — компонент тестовой системы, задача которого заключается в получении набора транзакций от генератора сценариев (массива объектов класса ehci\_descriptor) и преобразования их в массивы транзакций интерфейса AMBA 3 AXI, содержащих информацию о значениях полей дескриптора, адресах их размещения в памяти и сопутствующих дескриптору массивов данных. Также этот компонент отвечает за преобразование транзакций, полученных от блока абстрактного описания регистровой модели в AXI-транзакции, адресованные регистрам тестируемого блока. Все сформированные AXI-транзакции передаются драйверу интерфейса AMBA 3 AXI.

4. Драйвер и монитор интерфейса AMBA 3 AXI (интерфейс ведущего). Вызывая процедуры поведенческой модели, драйвер осуществляет преобразование полученных транзакций в циклограммы интерфейса AMBA 3 AXI. Встроенный монитор контролирует корректность циклограмм.

5. Драйвер и монитор интерфейса AMBA 3 AXI (интерфейс ведомого) представляют собой поведенческую модель памяти с интерфейсом ведомого AMBA 3 AXI. Встроенный монитор контролирует корректность циклограмм. Этот компонент имеет также набор процедур, вызываемых в случае выполнения определенных условий AXI-передачи. Например, он содержит процедуру, вызываемую в момент записи контроллером в область памяти дескрипторов. Эта процедура на основании записанных данных формирует транзакцию, описывающую EHCI-дескриптор, которая передается в компонент координатор.

6. Драйвер и монитор интерфейса ULPI содержат поведенческую модель интерфейса ULPI. После получения новой законченной передачи по шине ULPI с тремя фазами (маркер, данные, подтверждение) — монитор интерфейса формирует объект, описывающий совершенную USB-транзакцию и передает его

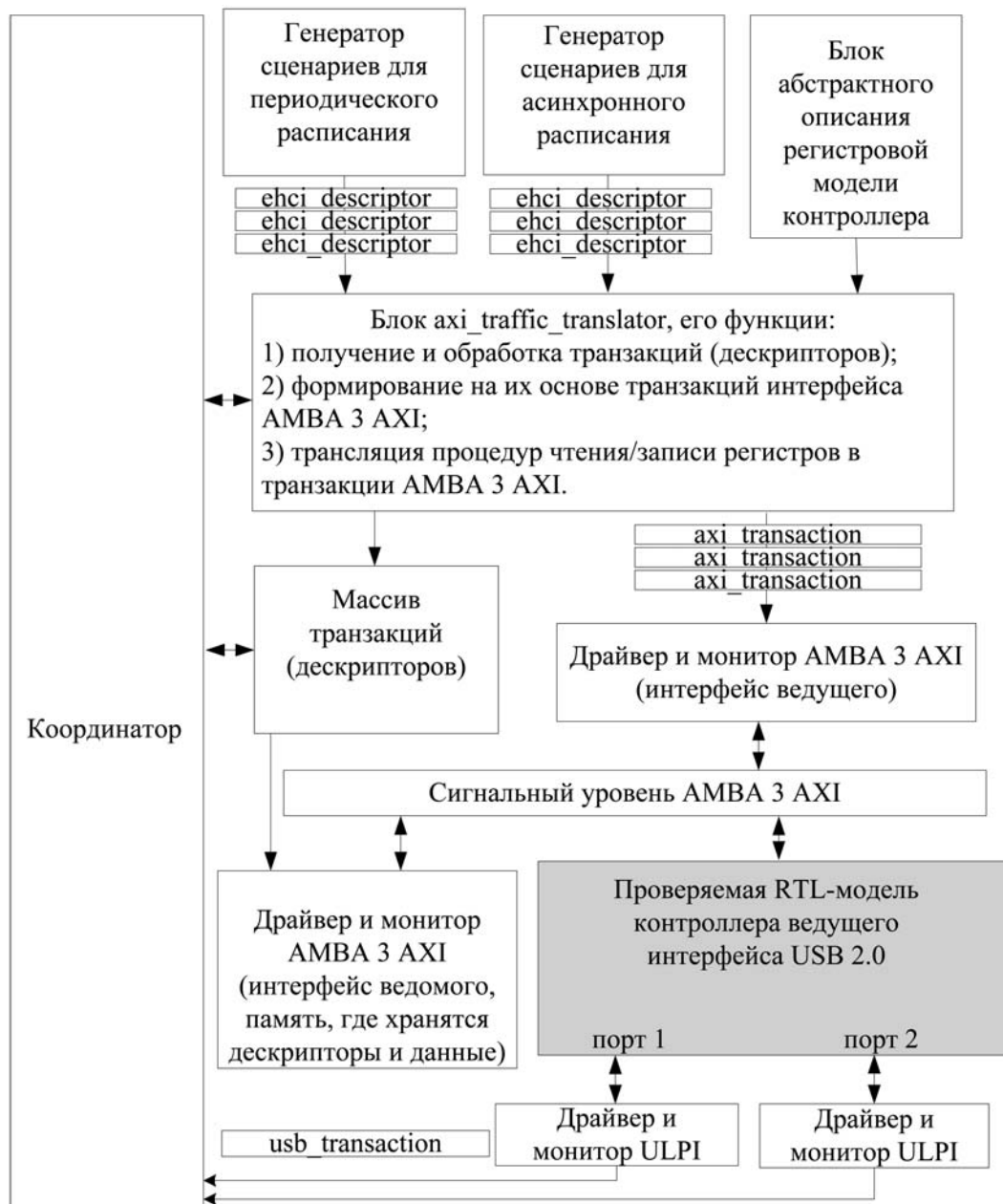


Рис. 2. Тестовая система контроллера USB 2.0 стандарта EHCI

координатору. Драйвер интерфейса ULPI при необходимости формирования отклика вызывает соответствующие процедуры поведенческой модели.

7. Координатор содержит в себе процедуры преобразования объектов, описывающих EHCI-дескрипторы, и объектов, описывающих USB-транзакции, в наборы элементарных объектов `sb_item`. Эти объекты содержат общий для исходных структур набор полей: передаваемые данные, номер USB-устройства, номер конечной точки, тип передачи и др. Эти элементарные объекты позволяют координатору сравнивать EHCI-дескриптор и набор сформированных на его основе USB-транзакций, тем самым контролируя корректность поведения проверяемого EHCI-контроллера.

### Порядок работы тестовой системы

#### Этап 1. Формирование исходных данных

За формирование исходных данных (а именно сценариев — массивов связанных по цепочке дескрипторов с данными) отвечает генератор сценариев. Большинство полей формируемых дескрипторов задается случайным образом в соответствии с ограничениями, наложенными пользователем в конкретном сценарии или тесте.

Формирование сценария происходит следующим образом.

Псевдослучайно формируются поля объекта общей конфигурации (с учетом ограничений, заданных в конкретном тесте). Этот объект описывает об-

шую конфигурацию тестовой системы (разрешенные типы передач, соответствующие им сценарии, допустимое число экземпляров каждого сценария, параметры конфигурации поведенческих моделей USB-устройств и т. д.). Поля данного объекта остаются постоянными на протяжении всего теста.

Работа генератора сценариев сводится к циклическому выполнению следующих шагов:

1. Если число сформированных экземпляров сценариев не достигло соответствующего максимального количества, установленного в объекте общей конфигурации, то генератор сценариев начинает формирование нового периодического или асинхронного сценария.

2. Формирование сценария выполняется в два этапа — сначала происходит выбор типа реализуемого сценария из множества доступных для выбранного типа передач. Сценарии сделаны иерархическими, т. е. один тип сценария более высокого уровня может вызывать один или несколько сценариев более низкого уровня. Тип сценария накладывает определенные специфические ограничения на поля дескрипторов или последовательность типов дескрипторов, а также на их максимальное количество. После выбора типа сценария проводится рандомизация всех дескрипторов и данных к ним с учетом требований выбранного типа сценария.

### **Этап 2. Расположение сформированных данных в памяти**

Сформированные цепочки дескрипторов от генераторов сценариев поступают в компонент `axi_traffic_translator`, который формирует последовательности AXI-транзакций, передаваемые драйверу ведущего интерфейса AXI, а тот в свою очередь формирует циклограммы на шине AMBA 3 AXI. Драйвер ведомого интерфейса AMBA 3 AXI реагирует на указанные циклограммы и размещает полученные дескрипторы и данные в памяти. Таким образом, имитируется реальная активность процессора по размещению дескрипторов и данных в памяти в конкурентном режиме с проверяемым контроллером.

### **Этап 3. Осуществление проверки целостности данных**

*Для исходящих USB-транзакций из контроллера EHCI (OUT-транзакций).*

1. Сформированные дескрипторы с данными передаются в компонент координатор, где разбиваются на единичные структурные элементы `sb_item`, соответствующие USB-транзакциям, которые должны быть сформированы проверяемым контроллером на основе указанных дескрипторов. Эти отдельные структурные элементы помещаются в соответствующие исходящие очереди координатора (очереди HOST->DEVICE). Для каждого типа дескрипторов существует своя очередь.

2. Монитор USB-устройства с интерфейсом ULPI (монитор ULPI), получив данные, преобра-

зует полученный USB-пакет к формату, принятому в структурном элементе `sb_item` координатора, и передает этот элемент координатору.

3. Координатор проводит поиск аналогичного элемента в своей исходящей очереди (очереди HOST->DEVICE). Если совпадение найдено, оба элемента уничтожаются. Если ни одного такого элемента в исходящей очереди не было найдено, фиксируется ошибка типа "потеря данных контроллером".

4. При записи проверяемым контроллером статуса дескриптора в память драйвера ведомого интерфейса AMBA 3 AXI в последнем автоматически вызывается процедура, которая анализирует правильность закрытия статуса дескриптора. В противном случае фиксируется ошибка "неверный статус закрытия дескриптора".

*Для входящих USB-транзакций в контроллер EHCI (IN-транзакций).*

1. В момент преобразования EHCI-дескриптора, описывающего транзакцию для направления IN, компонент `axi_traffic_translator` помещает его также в специальную исходящую очередь (для того, чтобы знать исходные значения полей дескриптора).

2. Получив от EHCI контроллера маркер USB пакета для направления IN, драйвер ULPI формирует случайную USB-транзакцию в рамках ограничений, заданных для конкретного устройства и конечной точки в объекте общей конфигурации, и вызывает процедуры поведенческой модели ULPI интерфейса для формирования необходимых циклограмм на интерфейсе ULPI. Сформированная USB-транзакция также преобразуется к виду элементарной структуры данных координатора — `sb_item` и помещается в соответствующую исходящую очередь DEVICE->HOST координатора.

3. При закрытии контроллером EHCI очередного дескриптора в памяти драйвера ведомого интерфейса AMBA 3 AXI из данного дескриптора формируется массив элементов данных для координатора (`sb_item`), каждый элемент данного массива сравнивается координатором с соответствующим ему элементом исходящей очереди DEVICE->HOST. При совпадении элементы уничтожаются. Если элемент не найден, формируется ошибка типа "запись ошибочных данных контроллером по завершении приема USB-пакета".

Таким образом, при успешном выполнении теста к его концу очереди HOST->DEVICE и DEVICE->HOST координатора должны оказаться пустыми. Наличие элементов `sb_item` в данных очередях к концу теста свидетельствует о том, что указанные данные были потеряны проверяемым контроллером.

### **Заключение**

Таким образом, спроектирована тестовая система в соответствии с методологией VMM, позволяющая проводить верификацию модели EHCI контроллера ведущего интерфейса USB 2.0. Целостность

передаваемых контроллером данных в обоих направлениях контролируется независимыми от него компонентами тестовой системы.

Изложенные принципы построения и алгоритмы функционирования тестовой системы могут быть применены для любых блоков СНК, осуществляющих преобразование потоков данных на основе информации в дескрипторах.

Примененный механизм формирования иерархических сценариев показал свою эффективность, снизив трудозатраты по формированию случайных сценариев на 40 %.

Модульность тестовой системы позволяет переносить отдельные ее компоненты из проекта в проект и осуществлять над ней работу параллельно нескольким разработчикам, что сокращает общее время верификации RTL-модели. Для дальнейшего сокращения трудозатрат и стандартизации интерфейсов между компонентами все компоненты тестовой системы основаны на базовых классах из методологии VMM.

Большую роль в повышении качества проверки модели также сыграло применение встроенных формальных утверждений (ФУ) в коде RTL-модели. Данные ФУ позволили быстро локализовать места возникновения ошибок при случайном тестировании, а также благодаря использованию библиотеки ФУ для стандартных интерфейсов, выявили крас-

ную ситуацию по запрещенному переходу 4-килобайтной границы в блоке ПДП передач, подключенного к магистрали AMBA 3 AXI. Ошибка была выявлена с помощью статического формального анализа выполнимости указанных ФУ из библиотеки утверждений для магистрали AMBA 3 AXI.

Несмотря на необходимость изучения языка *System Verilog*, новых методик верификации общее время верификации блока СНК осталось на прежнем уровне по сравнению с традиционным подходом с применением языка Verilog — верификация RTL-модели блока у коллектива из трех инженеров заняла 8 месяцев (от момента принятия решения о разработке контроллера до получения полностью работоспособного макетного образца). Таким образом, при дальнейшем использовании данной методики ожидается снижение времени верификации RTL-моделей аналогичных блоков СБИС до 30 %.

#### Список литературы

1. Bergeron J., Cerny E., Hunter A. Verification Methodology Manual for SystemVerilog. New York : Springer, 2005.
2. Бобков С. Г. Методика тестирования микросхем для компьютеров серии "Багет" // Программные продукты и системы. 2007. № 3. С. 2—5.
3. Mintz M., Ekendahl R. Hardware verification with SystemVerilog — an object-oriented framework. New York: Springer Science + Business Media, LLC, 2007.
4. IEEE 1800—2005. IEEE Standard for SystemVerilog — unified hardware design, specification and verification language. 2005.

УДК 621.396.2

Е. С. Кузнецов, аспирант,

Е. Н. Талицкий, д-р техн. наук, проф.,

С. В. Шумарин, канд. техн. наук, доц.,

Владимирский государственный университет

им. А. Г. и Н. Г. Столетовых,

e-mail: ental@vpti.vladimir.ru

## Автоматизация проектирования виброзащиты ячеек электронной аппаратуры демпфирующими вставками

*Предлагается алгоритм проектирования и оптимизации виброзащиты ячеек электронной аппаратуры демпфирующими вставками. Рассматриваются системы конечно-элементного анализа для реализации этого алгоритма. Приводятся результаты экспериментальных исследований, подтверждающие возможность применения разработанной САПР для расчета и оптимизации виброзащиты ячеек электронной аппаратуры демпфирующими вставками.*

**Ключевые слова:** электронная аппаратура, виброзащита, демпфирующие вставки, САПР ANSYS

## Введение

Электронная аппаратура (ЭА) авиационной и ракетно-космической техники при эксплуатации может подвергаться воздействию вибраций в широком диапазоне частот, что приводит к резонансным колебаниям печатных плат (ПП) и других плоских элементов конструкций и, как следствие, к снижению надежности аппаратуры.

Возможным решением данной проблемы является уменьшение амплитуд резонансных колебаний ЭА путем увеличения демпфирующих свойств конструкции [1, 2]. Это достигается применением полимерных демпферов в виде демпфирующих слоев, демпфирующих ребер и др.

Одним из эффективных видов полимерных демпферов являются вставки, выполненные из вибропоглощающих материалов. Вставки могут размещаться на свободных местах ячеек ЭА, например между рядами электрорадиоэлементов. Таким образом, их можно применять и как "лечебное" средство, когда конструкция уже разработана, но не выдержала виброиспытания. Кроме того, они занимают минимальную площадь на плате (рис. 1, 2). Условием эффективного использования демпфирующих вставок (ДВ), наряду с требованием высо-

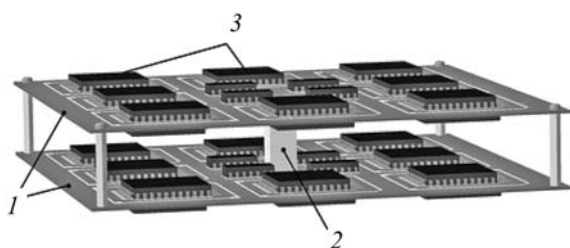


Рис. 1. Ячейки с демпфирующей вставкой:  
1 — ячейки; 2 — ДВ; 3 — электрорадиоэлементы

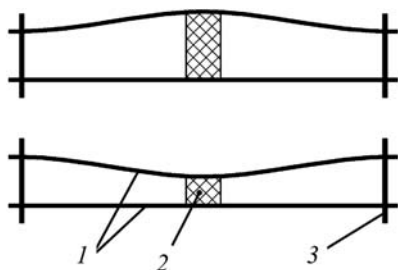


Рис. 2. Вид деформации вставки при резонансе верхней ячейки:  
1 — ячейки; 2 — ДВ; 3 — стойка

ких демпфирующих свойств вибропоглощающего материала, является требование обеспечения различия собственных частот колебаний (СЧК) ячеек не менее чем в 2 раза [3]. Для этого необходимо определить такое соотношение между жесткостями ячеек и ДВ, которое обеспечивает максимальное ее деформирование, а следовательно, и максимальную эффективность вставки.

Сложность также заключается в том, что это соотношение зависит от температуры и частоты вибрации, вследствие резкой зависимости упругодемпфирующих характеристик вибропоглощающих полимеров от этих факторов. Так как эти характеристики полимеров удается экспериментально определить в частотном и температурном диапазонах с определенной степенью дискретизация, то расчет параметров ДВ носит рекуррентный характер и обычно осуществляется методом последовательного приближения на основе аналитических выражений [4] или численного моделирования [5].

### Методы расчета

Аналитические методы имеют высокое быстродействие, не требуют специальных программных средств, но пригодны лишь для расчета простейших конструкций с типовыми моделями крепления, т. е. для которых известны граничные условия (жесткое защемление, свободное опирание или свободный край) [4]. Численные методы, среди которых важное место занимают методы конечно-элементного анализа (КЭА), не имеют подобных ограничений, однако реализуются в сравнительно сложных программных продуктах, требуют высокой вычислительной мощности и длительной подготовки данных [6].

Среди программных систем, реализующих КЭА, можно выделить MSC/NASTRAN, COSMOS, ANSYS, LS-DYNA, ABAQUS и другие. Они, в том числе, позволяют проводить динамический анализ механических конструкций и обеспечивают импорт-экспорт проектов с большинством известных систем инженерного конструирования, что позволяет реализовать технологию "сквозного проектирования" ЭА [5].

Однако указанные системы КЭА в явном виде не предназначены для расчета конструкций с полимерными демпферами. В них возможен учет зависимости динамического модуля упругости (ДМУ) и коэффициента механических потерь (КМП) только от температуры, а частотная зависимость ДМУ и КМП не реализована. Это вызывает необходимость использования при расчете метода последовательного приближения, что значительно увеличивает время расчета. Частично сократить это время можно за счет использования пакетных файлов, реализующих данный (рекуррентный) метод расчета.

Другое затруднение связано с учетом демпфирующих свойств конструкции. В работе [7] показано, что при инженерных расчетах ячеек с полимерными демпферами можно использовать только демпфирование в материалах конструкционных слоев, демпфирующей вставки и за счет трения в местах крепления.

Предлагаемый алгоритм разработан с учетом этих факторов. Он сочетает высокое быстродействие аналитических выражений и универсальность систем КЭА и основан на использовании метода последовательного приближения (рис. 3). В этом алгоритме система КЭА ANSYS используется в качестве вычислителя, что существенно упрощает работу проектировщика, так как не требует специальных знаний по использованию данной системы.

Так как вибрации электронной аппаратуры часто носят случайный характер, алгоритмом предусматривается возможность анализа конструкции также и при широкополосной случайной вибрации (ШСВ).

### Описание алгоритма

Основной задачей при применении ДВ является снижение амплитуд резонансных колебаний (АРК) ячеек до допустимого уровня. Добиться этого, не изменяя массогабаритные характеристики ячеек, можно только варьируя жесткость демпфирующей вставки

$$k_d = \frac{ES}{H},$$

где  $E$  — модуль упругости вибропоглощающего материала;  $S$ ,  $H$  — площадь и высота ДВ.

Часто необходимо добиться того, чтобы максимальный коэффициент передачи (КП) обеих ячеек снизился до одного уровня, т. е. необходимо найти

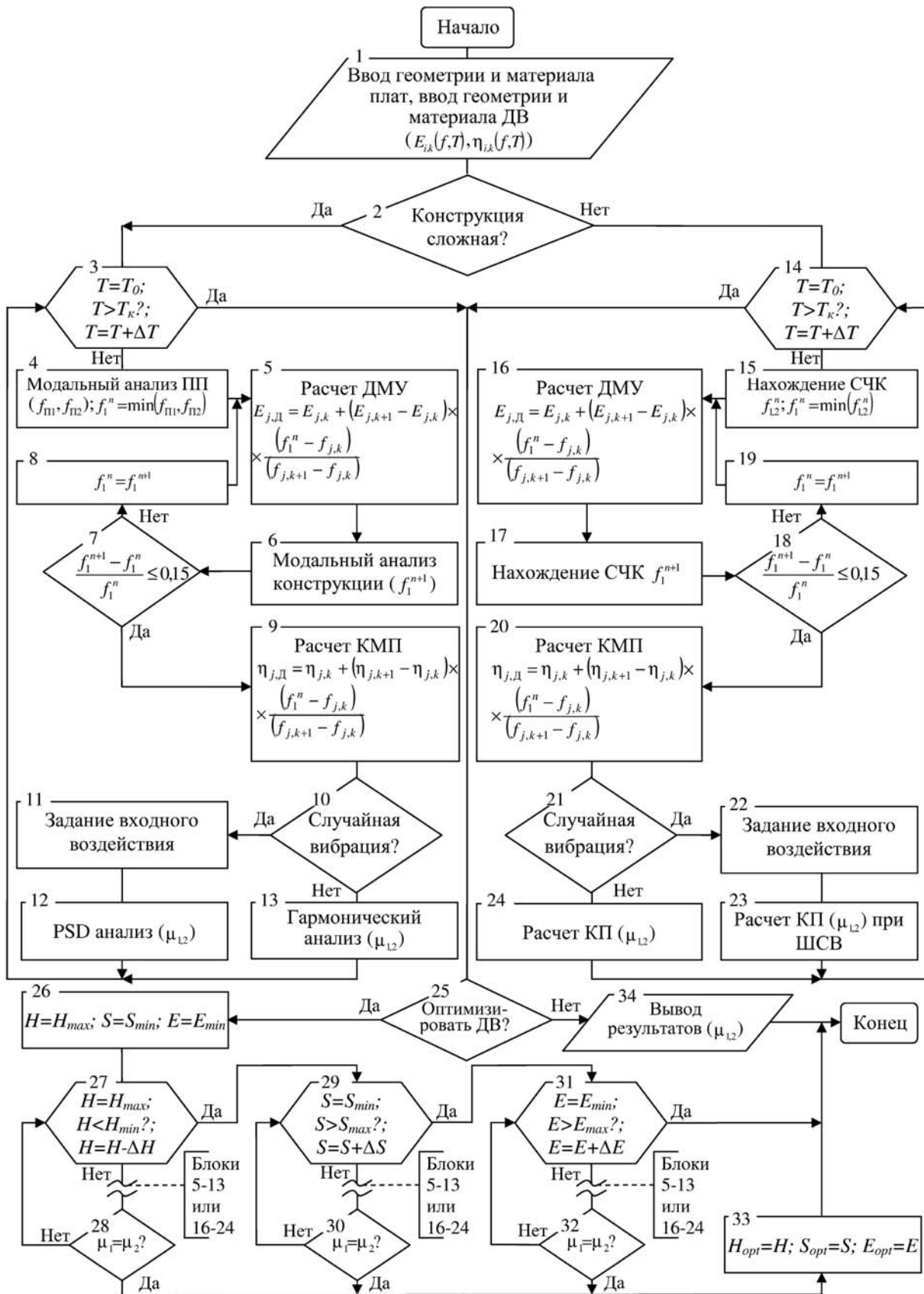


Рис. 3. Блок-схема алгоритма проектирования и оптимизации виброзащиты конструкций электронной аппаратуры демпфирующими вставками

такую жесткость вставки, при которой коэффициенты передачи  $\mu_1 = \mu_2$ .

Тогда в общем виде целевая функция примет вид:

$$\begin{cases} \mu_1 = f(E, S, H) \rightarrow \min \\ \mu_2 = f(E, S, H) \rightarrow \min. \end{cases}$$

При этом должны выполняться ограничения:

$$\begin{cases} \mu_1 = \mu_2; \\ E_{\min} \leq E \leq E_{\max}; \\ S_{\min} \leq S \leq S_{\max}; \\ H_{\min} \leq H \leq H_{\max}. \end{cases}$$

Поиск оптимальных значений параметров ДВ может быть осуществлен любым методом одномерной оптимизации: дихотомического деления, золотого сечения, чисел Фибоначчи, полиномиальной аппроксимации, Гаусса—Зейделя и т. д.

Процесс проектирования начинается с задания геометрии и материалов всех элементов конструкции (блок 1 здесь и далее на рис. 3), частотно- и температурно-зависимых ДМУ  $E_{j,k}(f, T)$  и КМП  $\eta_{j,k}(f, T)$  материала ДВ №  $j$  ( $k$  — индекс ближайшего нижнего значения частоты относительно  $f_1^n$ ). Пример таких характеристик для пенополиуретана ППУ202-2 представлен в табл. 1.

Далее, исходя из сложности конструкции, выбирается способ ее расчета: аналитическими или численными методами. Прямоугольные ПП с типовыми способами крепления рассчитываются с помощью аналитических выражений, более сложные конструкции — в системе КЭА (блок 2).

Расчет численным методом начинается с модального анализа (определения СЧК) всех ПП, входящих в конструкцию (блок 4), с последующим нахождением минимального значения из них — СЧК конструкции в первом приближении  $f_1^n$ , которое необходимо для учета частотной зависимости ДМУ [5].

Таблица 1

Пример характеристик вибропоглощающего материала ППУ202-2

Материал ППУ202-2						
Температура, °С	200 Гц		600 Гц		1200 Гц	
	$E \cdot 10^5$ , Па	$\eta$	$E \cdot 10^5$ , Па	$\eta$	$E \cdot 10^5$ , Па	$\eta$
-60	950	0,10	1200	0,10	2200	0,10
-40	350	0,17	500	0,10	1000	0,25
-20	200	0,12	250	0,13	450	0,17
0	120	0,11	150	0,11	180	0,14
+20	50	0,12	70	0,14	80	0,13
+40	40	0,11	50	0,12	60	0,11
+60	35	0,18	40	0,18	45	0,20
+70	30	0,23	35	0,22	40	0,25

При вычислении по аналитическим формулам возможно произвольное задание СЧК первого приближения, так как время расчета практически не изменится, а конечная СЧК при любой базовой будет той же. Блоки 5—8 и 16—19 (см. рис. 3) реализуют рекуррентное вычисление СЧК конструкции с последующей корректировкой ДМУ до тех пор, пока расхождение между двумя соседними значениями частот не станет меньше требуемой точности. Найденные значения СЧК также необходимы для учета частотно-зависимых демпфирующих свойств конструкции (блоки 9 и 20).

В блоках 10 и 21 происходит выбор внешнего вибрационного воздействия: гармоническое или случайное. При ШСВ необходимо задать форму вибрационного воздействия (блоки 11 и 22).

Гармонический анализ и анализ на случайную вибрацию конструкции ЭА предполагает нахождение коэффициентов передачи виброускорения всех ПП ( $\mu_{1,2}$  в блоках 12 и 23) [5]. Окончательно ЭА считается виброзащищенной, когда действующее виброускорение всех точек и элементов конструкции не превышает допустимого значения, определяемого ТУ электрорадиоэлементов и используемых материалов.

Данный алгоритм реализует анализ конструкций с ДВ во всем заданном температурном диапазоне с шагом  $\Delta T$  (блоки 3 и 14). После превышения конечной температуры  $T_k$ , если нет необходимости в оптимизации параметров ДВ (блок 25), происходит вывод амплитудно-частотной характеристики каждой ячейки (блок 34).

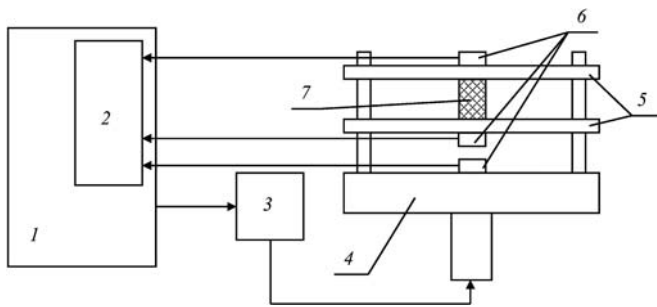
В случае недопустимо высокого отклика конструкции на внешнее вибрационное воздействие алгоритмом предусматривается возможность оптимизации параметров ДВ (блоки 26—33), которая сводится к нахождению оптимальных значений  $E_{opt}$ ,  $S_{opt}$  и  $H_{opt}$  (блоки 27—33). После каждого шага итерации (блоки 27, 29, 31) выполняются блоки 5—13 или 16—24 в зависимости от сложности конструкции. В случае нахождения оптимальных параметров ДВ осуществляется их вывод (блок 33), в противном случае необходимо изменить функциональные и/или прямые ограничения оптимизации или параметры конструкции.

Представленный алгоритм позволяет быстро и эффективно применять ДВ для повышения виброзащищенности ячеек ЭА, что значительно сокращает время их разработки. На основе представленного алгоритма была разработана программа [8], интегрированная в Комплекс программ анализа механических воздействий на радиоэлектронную аппаратуру [9].

### Экспериментальные исследования

Для оценки точности расчетов на основе разработанного алгоритма были проведены экспериментальные исследования на установке, структурная схема





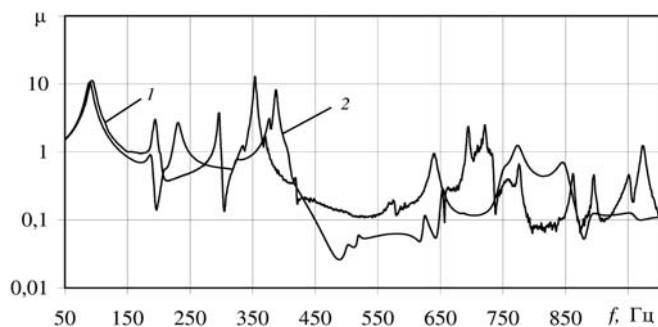
**Рис. 4. Структурная схема установки для испытаний:**  
 1 — ЭВМ; 2 — устройство сбора данных; 3 — усилитель; 4 — вибростенд IMV VS-600/SA1; 5 — ячейки; 6 — вибродатчики типа IMV VP-32; 7 — ДВ

которой представлена на рис. 4. Исследовались ячейки прямоугольной формы из стеклотекстолита с размерами  $145 \times 125 \times 0,8$  мм и  $145 \times 125 \times 1,5$  мм.

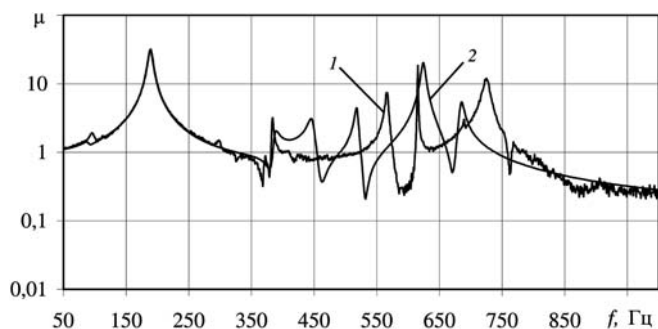
Эксперимент проводился при температуре окружающей среды  $22^\circ\text{C}$  при внешнем случайном воздействии со спектральной плотностью ускорения  $0,01 \text{ g}^2/\text{Гц}$ . Ячейки испытывались в диапазоне частот от 50 до 1000 Гц.

Коэффициент передачи виброускорения  $\mu$  при случайной вибрации определяется по формуле

$$\mu = \sqrt{\frac{S_{\text{ВЫХ}}}{S_{\text{ВХ}}}}$$



**Рис. 5. Спектрограмма конструкции с ДВ в центре ячейки толщиной 0,8 мм:**  
 1 — результаты эксперимента; 2 — результаты моделирования



**Рис. 6. Спектрограмма конструкции с ДВ в центре ячейки толщиной 1,5 мм:**  
 1 — результаты эксперимента; 2 — результаты моделирования

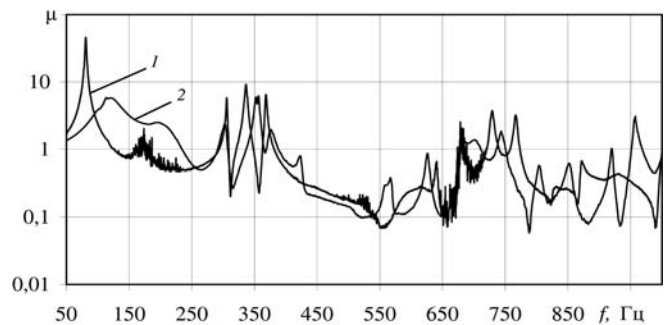
где  $S_{\text{ВЫХ}}$  — спектральная плотность ускорения в центре ячейки;  $S_{\text{ВХ}}$  — спектральная плотность ускорения на платформе вибростола.

Коэффициенты передачи ячеек определялись до установки ДВ, после установки ДВ (рис. 5, 6) и после установки оптимальной ДВ (рис. 7, 8).

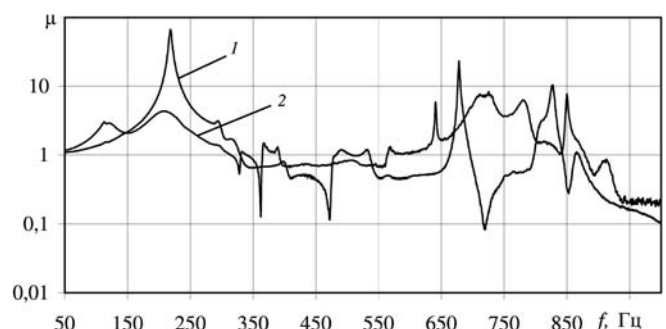
До установки ДВ коэффициенты передачи ячеек при случайной вибрации составляли 45,6 и 66,5 для первой (0,8 мм) и второй (1,5 мм) ячейки соответственно (рис. 7, 8). Так как СЧК ячеек отличаются более чем в 2 раза (81 и 218 Гц), то между ними в центре была установлена ДВ (длина — 10 мм, ширина — 10 мм, высота — 20 мм) из опытного образца пенополиуретана с характеристиками:  $E = 572 \text{ кПа}$ ,  $\rho = 100 \text{ кг/м}^3$ ,  $\eta = 0,855$ . Параметры ДМУ и КМП были определены методом резонансных колебаний при температуре окружающей среды  $22^\circ\text{C}$ .

Значения КП при случайной вибрации, полученные в ходе эксперимента, сравнивались с результатами численного моделирования в системе КЭА ANSYS. Полученные графики-зависимости КП обеих ячеек, входящих в конструкцию, после установки ДВ приведены на рис. 5, 6.

Анализ графиков на рис. 5 и 6 показывает, что коэффициенты передачи ячеек отличаются более чем в 2 раза (12,9 и 31,9), т. е. параметры ДВ для данной конструкции не оптимальны. Используя разработанный алгоритм проектирования и оптимизации (см. рис. 3), а также программу на его ос-



**Рис. 7. Спектрограмма конструкции в центре ячейки толщиной 0,8 мм:**  
 1 — без ДВ; 2 — с оптимальной ДВ



**Рис. 8. Спектрограмма конструкции в центре ячейки толщиной 1,5 мм:**  
 1 — без ДВ; 2 — с оптимальной ДВ

Сравнение результатов эксперимента и моделирования в системе КЭА ANSYS ячеек макета без ДВ

Параметр	Эксперимент		Система КЭА ANSYS		Ошибка, %	
	Плата толщиной 0,8 мм	Плата толщиной 1,5 мм	Плата толщиной 0,8 мм	Плата толщиной 1,5 мм	Плата толщиной 0,8 мм	Плата толщиной 1,5 мм
СЧК (1-я), Гц	81	218	87	214	7	2
КП	45,6	66,5	39,2	68,4	14	3

Таблица 3

Сравнение результатов эксперимента и моделирования в системе КЭА ANSYS ячеек макета с оптимальной ДВ

Параметр	Эксперимент		Система КЭА ANSYS		Ошибка, %	
	Плата толщиной 0,8 мм	Плата толщиной 1,5 мм	Плата толщиной 0,8 мм	Плата толщиной 1,5 мм	Плата толщиной 0,8 мм	Плата толщиной 1,5 мм
СЧК (с КП <sub>max</sub> ), Гц	356	725	371	709	4	2
КП <sub>max</sub>	6,2	8,2	7,0	7,0	13	15

нове, при неизменном ДМУ материала вставки была найдена ее оптимальная площадь ( $S = 625 \text{ мм}^2$ ) и высота (15 мм), при которых КП обеих ячеек должен был снизиться до 7,0 (рис. 7, 8).

Анализ графиков (рис. 5, 6) показывает, что основной вклад в амплитуду виброускорений начинают вносить более высокие резонансные частоты, что затрудняет нахождение оптимальных параметров ДВ. Сравнительные характеристики точности моделирования в системе КЭА ANSYS с результатами эксперимента приведены в табл. 2 и табл. 3.

Представленный алгоритм и разработанная на его основе программа [8] позволяют учитывать не только сложный характер зависимости динамических характеристик демпфирующих вставок от частоты и температуры, но также сложность конструкции и вибрационного воздействия благодаря применению как аналитических математических моделей, так и систем конечно-элементного анализа.

#### Список литературы

1. Зеленев Ю. В., Кирилин А. А., Слободник Э. Б., Талицкий Е. Н. Виброзащита электронной аппаратуры полимерными

компаундами / Под. ред. Ю. В. Зеленева. М.: Радио и связь, 1984. 120 с.

2. Нашиф А., Джоунс Д., Хендерсон Дж. Демпфирование колебаний: пер с англ. М.: Мир, 1988. 488 с.

3. Талицкий Е. Н. Моделирование виброустойчивых конструкций РЭА с полимерным демпфером // Вопросы радиоэлектроники. Сер. ТПО, вып. 2. 1988.

4. Варакин А. А., Талицкий Е. Н. Исследование конструкций ячеек радиотехнических устройств демпфирующими вставками // Проектирование и технология электронных средств. 2002. № 4. С. 9–11.

5. Кузнецов Е. С., Талицкий Е. Н., Шумарин С. В. Инструментальная среда ANSYS для проектирования ячеек электронной аппаратуры с демпфирующими вставками // Проектирование и технология электронных средств. 2010. № 1. С. 53–56.

6. Норенков И. П. Основы автоматизированного проектирования. М.: Изд-во МГТУ им. Н. Э. Баумана, 2002. 336 с.

7. Цедерштрём А. А., Талицкий Е. Н. Расчет виброустойчивых ячеек радиотехнических устройств с использованием системы ANSYS // Перспективные технологии в средствах передачи информации: Труды VIII Международной научно-технической конференции. — Владимир, 2009.

8. Кузнецов Е. С., Талицкий Е. Н., Шумарин С. В. Программа проектирования виброзащиты ячеек радиотехнических устройств демпфирующими вставками / Свидетельство об официальной регистрации программы для ЭВМ № 2010615876, 18.11.2010.

9. Шумарин С. В., Талицкий Е. Н. Программный комплекс проектирования виброзащиты ячеек электронной аппаратуры // Программные продукты и системы. 2008. № 3. С. 103–104.

УДК 004

**С. Г. Керимов**, д-р техн. наук, проф.,  
чл.-корр. НАН Азербайджана, зав. каф.,  
e-mail: kerimov.sabit@gmail.ru,

**Е. С. Керимова**, канд. техн. наук, ст. науч. сотр.,  
Азербайджанская Государственная  
нефтяная академия (г. Баку)

## Репозитории в информационных системах и программном обеспечении

*Рассматривается роль репозитория в информационных системах и программном обеспечении. Излагаются особенности разработки репозитория для систем OLTP, OLAP и пакетов программ, размещенных в Интернете. Приводится пример построения репозитория базы данных. Дается обзор известных репозитория программно-обеспечения, опубликованного в Интернете.*

**Ключевые слова:** репозиторий, информационная система, база данных, хранилище данных, программное обеспечение, Интернет

### Введение

Метаданные, определяемые как "данные о данных", служат для идентификации информационных ресурсов и программных средств. В фактографических информационных системах (ИС), которые ориентированы на хранение и поиск структурированных данных, метаданные используются для описания данных, определения структур, управления доступом, проверки правильности имен атрибутов, таблиц, баз данных. В таких системах метаданные представляются в виде словарей-справочников. В информационно-поисковых системах, являющихся классическими документальными системами, так называемые поисковые образы (индексы) документов, состоящие из дескрипторов, фактически представляют собой набор метадаанных.

В настоящее время область использования метадаанных расширилась. Они используются: в хранилищах информационных ресурсов по различным отраслям, всемирной информационной сети Интернет, электронных библиотеках, в системах дистанционного обучения и т. д.

Для описания метадаанных существуют много способов и языков, наиболее эффективными из

которых считаются языки разметки. Известный язык разметки HTML, используемый для описания Web-документов, имеет тег <META>, предназначенный для описания основного содержания документа. Описание метадаанных в HTML встраивается в сам документ, что позволяет легко его идентифицировать.

Однако включение метадаанных в документ не позволяет самостоятельно их использовать другими системами, т. е. при этом не обеспечивается интероперабельность. Поэтому известные модели и технологии, претендующие на стандарт для работы с метадаанными, учитывают требования интероперабельности и предусматривают отделение метадаанных от самих данных (документов). Среди них наибольшую перспективу имеют: стандарт DC (Dublin Core), язык XML, модель RDF, технологии CORBA и OMA. Сведения об этих и других моделях и технологиях даны в работе [1].

Для хранения метадаанных используется база метадаанных, называемая **репозиторием**. Но термин "репозиторий" определяют не только базу метадаанных, а также хранилище, где хранятся и поддерживаются версии программ различного назначения. Тут речь идет в основном о хранилище программ в Интернете.

В статье рассматриваются функции, структуры и организация репозитория в информационных системах и в программном обеспечении.

### Репозитории в системах оперативной обработки транзакций

Системы оперативной обработки транзакций (On-Line Transaction Processing, OLTP) основываются на использовании баз данных. Поэтому под репозиторием системы OLTP понимается репозиторий базы данных.

Репозиторий базы данных (БД) служит для хранения и поддержки данных об объектах БД, о пользователях, об экземплярах (в распределенных БД). В некоторых случаях там хранятся также регламентные запросы, называемые "храняемыми" запросами. Репозиторий БД часто называют *словарем данных*, или *базой метадаанных* [2, 3].

В корпоративных информационных системах (КИС), использующих несколько БД, репозитории организуются как минимум на двух уровнях (рис. 1). На первом уровне организуется центральный репозиторий, включающий в себя данные о базах

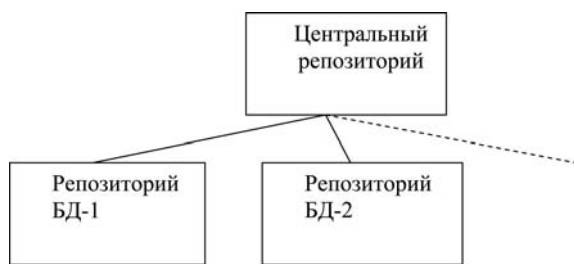


Рис. 1. Структура репозиторий КИС

Таблица 1

Метатаблица TABLES

T-NAME	R-NAME
ZONES	Зоны
OBJECTS	Объекты
SAMPLES	Пробы
HYDROMETEROLOGY	Гидрометеорология
HYDROCHEMICAL	Гидрохимия
TAKING	Измерение

Таблица 2

Метатаблица ATTRIBUTES

T-NAME	A-NAME	NAME	DATATYPE	KA
ZONES	IDZ	Идентификатор зоны	INTEGER	✓
ZONES	SPACE	Площадь	INTEGER	
ZONES	POPUL	Население	INTEGER	
ZONES	DEGofPOL	Степень загрязнения	STRING	
ZONES	LENofCOL	Длина береговой линии	FLOAT	
OBJECTS	IDO	Идентификатор объекта	INTEGER	✓
OBJECTS	TYPofOBJ	Тип объекта	STRING	
OBJECTS	SPAofRES	Площадь водосбора	INTEGER	
OBJECTS	DEPofCUR	Глубина течения	FLOAT	
OBJECTS	SPEofCUR	Скорость течения	FLOAT	
OBJECTS	DEGofPOL	Степень загрязнения	STRING	
SAMPLES	X	Координата X	FLOAT	
SAMPLES	Y	Координата Y	FLOAT	
SAMPLES	DIStoMOU	Расстояние до устья	FLOAT	
SAMPLES	DATE-S	Дата измерения	DATETIME	✓

Таблица 3

Метатаблица DOMAINS

D-NAME	NAME	DATATYPE
ID	Идентификатор зоны/объекта	INTEGER
SPAS	Площадь	INTEGER
POP	Население	INTEGER
DOP	Степень загрязнения	STRING
DIS	Расстояние	FLOAT
TYP	Тип объекта	STRING
DEP	Глубина	FLOAT
SPE	Скорость	FLOAT
XY	Координаты	FLOAT
DATE	Дата	DATETIME

КИС: название, объем, дата создания, указатель БД и указатель репозитория БД. Указатель БД указывает местонахождение БД, а указатель репозитория — местонахождение репозитория БД.

В репозитории БД хранятся данные о следующих объектах БД: таблицы (tables), атрибуты (attributes), домены (domains), индексы (indexes), разрезы данных (views), сохраненные процедуры (stored procedures), триггеры (triggers), генераторы (generators) [4].

Как известно, реляционные БД хранят все данные в таблицах. Репозиторий также организуется в виде реляционной базы данных, состоящей из таблиц. Каждая таблица репозитория БД описывает соответствующий объект основной БД. Эти таблицы называют системными таблицами или метатаблицами.

В качестве примера рассмотрим БД экологического мониторинга Прикаспийских зон. Список таблиц репозитория (метатаблица TABLES) данной БД приведен в табл. 1, где T-NAME означает наименование таблиц на английском языке, а R-NAME — наименование таблиц на русском языке. Сведения об атрибутах перечисленных таблиц приведены в метатаблице ATTRIBUTES (табл. 2). Для краткости показаны атрибуты только первых трех таблиц. Поле KA содержит признак, указывающий на то, что данный атрибут используется в качестве ключа.

В принципе метаинформация о *доменах* вложена в метатаблицу ATTRIBUTES. Однако во избежание избыточности информации целесообразно группировать общие свойства атрибутов в отдельной метатаблице DOMAINS. Фрагмент метатаблицы DOMAINS, где приведены домены, соответствующие атрибутам метатаблицы TABLES, показан в табл. 3.

*Индекс* служит для определения адреса записи (строки таблицы) по ключу. Однако имеются и другие методы определения адреса записи по ключу (например, вычисление адреса по значению ключа, метод хеширования и т. д.) [5]. Поэтому в таких случаях в метабаза не хранится информация об индексах.

*Разрез данных* — это виртуальная таблица, которая не хранится физически в БД и содержит данные из одной или более таблиц или других разрезов.

Разрезы используются для хранения результатов часто используемых запросов. Поэтому для ускорения ответов на типовые и часто используемые запросы создание и хранение в метабаза разрезов считается целесообразным (они, как правило, хранятся вместе с идентификаторами запросов).

*Сохраненные процедуры* — это готовые программы, написанные на SQL или расширенном SQL. Они являются обязательной частью базы метаданных. Сохраненные процедуры могут получать входные параметры, возвращать значения приложению и могут быть вызваны из приложения подстановкой вместо имени таблицы в инструкции SELECT. Сохраненные процедуры реализуются в сетевых системах управления базами данных (СУБД) и выполняются сервером.

**Триггеры** представляют собой готовые программы, ассоциированные с таблицами или разрезами, которые автоматически запускаются при добавлении, изменении или удалении строки в таблице или в разрезе. Триггеры предназначены для:

- автоматического контроля значений вводимых данных;
- прослеживания и выполнения операций по обновлению и корректировке данных;
- автоматического документирования изменений таблиц. Триггеры также являются обязательной частью базы метаданных.

**Генератор** — это программа, которая автоматически создает уникальные номера (обычно порядковые номера строк) и вставляет их в столбец таблицы БД, когда выполняются операции INSERT или UPDATE. Далее эти номера используются как первичный ключ. Генератор работает в случае, когда пользователь не определил ключ для таблицы. Для БД может быть определено любое число генераторов, и каждый из них должен иметь уникальное имя. В метабазу включаются все генераторы БД.

### Репозитории в системах аналитической обработки данных

Системы аналитической обработки данных (On-Line Analytical Processing, OLAP) основываются на применении хранилища данных (Data Warehouse). Поэтому, когда говорят о репозитории системы OLAP, подразумевают репозиторий хранилища данных.

Роль метаданных для системы OLAP важнее, чем в системах OLTP. Одним из главных аспектов использования метаданных в хранилище данных (ХД) является их предметная ориентация. Метаданные должны ответить на главные вопросы: какие данные представлены в системе и как их получать в нужном для анализа виде.

Состав метаданных ХД мало отличается от состава метаданных БД. Базовые компоненты метаданных ХД включают в себя **описание таблиц, их атрибутов, ключей, разрезов данных** и т. д.

Существенным отличием репозитории ХД является **поддержка версионности** метаданных. Дело в том, что время жизни данных в ХД гораздо больше (5—10 лет, а то и более), чем в БД. Большой временной горизонт данных в ХД влияет на некоторые принципы управления данными. Хронология изменений структур данных в БД в данном временном горизонте должна быть зафиксирована в ХД. Метаданные должны отражать изменения внутренней структуры данных источников. Таким образом, поддержка версионности метаданных позволяет в каждый момент времени в прошлом обеспечить правильное описание структур данных и дает возможность узнать, какие данные, когда и как попали в ХД.

Репозиторий ХД содержит также **описание преобразований данных**. Описание преобразований данных для ХД включает в себя:

- идентификацию атрибутов источников данных;
- соответствие между атрибутами объектов источников данных и объектов ХД;
- преобразование атрибутов, таблиц кодировки и ссылочных таблиц;
- изменение наименований;
- изменение ключевых атрибутов;
- значения полей по умолчанию;
- приоритетность источников данных;
- алгоритмы преобразования данных и т. д.

Компоненты преобразования дают информацию о том, как данные были получены и введены в ХД.

Еще одним компонентом репозитория является **история поступления** в него данных. Этот компонент несет информацию о том, когда данные поступили в ХД, а также позволяет судить о полноте представления данных в ХД. Такая информация является очень важной для проведения анализа данных, поскольку на ее основе дается оценка корректности анализа данных и надежности его результатов.

В репозитории ХД хранится также **информация о терминологическом соответствии понятий**. Метаданные должны поддерживать в запросах понятную для пользователя терминологию независимо от того, какие наименования таблиц и атрибутов были использованы проектировщиком ХД.

Одним из важных компонентов репозитория ХД является **информация о состояниях и статистике использования данных** ХД. Эта информация используется для оптимизации производительности ХД. К ней относятся данные о числе строк в таблицах, темпе роста таблиц, среднее и максимальное число запросов на день, статистика архивирования и удаления данных, индексирование таблиц, частота использования индексов в запросах и т. д.

Следующим компонентом репозитория ХД являются **алгоритмы агрегации и обобщения данных, критерии выборки данных** из источников, **правила преобразования данных** источников перед загрузкой в ХД, **описание взаимосвязей** между объектами ХД, их **кардинальность** и т. д. Такая информация играет важную роль при проведении анализа данных и часто требуется аналитикам для достижения надежности результатов анализа.

Еще одним компонентом репозитория ХД является информация о том, **кто отвечает за содержание и актуальность различных источников данных**. Эта информация нужна для группы сопровождения ХД и позволяет организационно решать вопросы качества, точности и надежности данных в ХД.

Для описания метаданных ХД и обмена ими используется общепризнанная стандартная модель CWMI (Common Warehouse Metadata Interchange — обмен общими метаданными хранилища данных) [1].

Она была разработана совместными усилиями коалиции MDC (Meta Data Coalition) и консорциума OMG (Object Management Group) и была принята в 2001 г. CWMI определяет интерфейсы, используемые для обмена метаданными между ХД и аналитическими приложениями с помощью инструментальных средств ХД, аппаратно-программных платформ и репозитория ХД в распределенных системах.

Метамодель CWMI основывается на трех стандартах:

- UML — Unified Modeling Language (стандарт OMG для моделирования): определяет язык объектно-ориентированного моделирования;
- MOF — Meta Object Facility (стандарт OMG для метамоделирования): определяет средства для определения модели метаданных и обеспечивает программные средства для хранения и доступа к метаданным в репозитории;
- XMI — XML Metadata Interchange (стандарт OMG для обмена метаданными): определяет спецификации для обмена метаданными в формате XML.

Перечисленные стандарты формируют ядро архитектуры репозитория метаданных OMG (рис. 2).

Репозитории баз данных и хранилищ данных реализуются и управляются средствами реляционной СУБД. Большинство современных СУБД поддерживает репозитории. Для интерпретации метаданных структура репозитория (схема метаданных) и их семантика хранятся отдельно.



Рис. 2. Ядро архитектуры репозитория метаданных OMG

### Репозитории программного обеспечения

Репозитории программного обеспечения (ПО) непосредственно связаны с Интернетом. Они организуются и хранятся на соответствующих серверах и распространяются через Интернет. В Интернете репозиторий используется как хранилище программ и как каталог программ.

В репозитории ПО включаются так называемые "свободные" программные пакеты. "Свободный" означает, что автор данной программы поместил ее в Интернет как открытую для использования про-

грамму. В репозитории хранятся все версии программ с историей их изменения и другой служебной информацией, что позволяет использовать их в системах управления версиями.

Большая часть ПО в Интернете хранится в специальных ".deb"-файлах, в которых содержатся программы и необходимые библиотеки. Репозитории — это специальные серверы-хранилища таких файлов.

Для подключения к репозиториям используются специальные утилиты (например, в репозитории "Ubuntu" — утилита "Synaptic"). Большинство утилит поддерживают простой поиск по ключевым словам и способны разбивать группы пакетов по категориям. Связка "репозиторий — утилита" позволяет использовать простой, централизованный метод установки/удаления программ, а также представляет удобный способ выкладывания обновлений.

В качестве примера рассмотрим несколько репозиториях, известных из Интернета.

**Репозиторий "CentOS"** (англ. Community enterprise Operating System) является дистрибутивом Linux и основан на коммерческом программном обеспечении "Red Hat Enterprise Linux" компании "Red Hat", представляющей все исходные программные коды. Разработчики "CentOS" используют исходный код для создания окончательного продукта, доступного через сеть Интернет [6].

Имеются несколько репозиториях, работающих с "CentOS", которые предлагают пакеты программ, не включенные по умолчанию в основные базовые репозитории. Например, репозиторий "CentOS Plus" содержит программные пакеты, которые обновляют те, что входят в состав базового репозитория. Он вносит в "CentOS" изменения, которые отсутствуют в исходном дистрибутиве. Другой репозиторий "CentOS Extras" содержит компоненты, которые обеспечивают дополнительную функциональность для "CentOS" без нарушения совместности с исходным дистрибутивом. Этот репозиторий включен в "CentOS" по умолчанию.

**Репозиторий "Debian"** [7] — это набор пакетов Debian, размещенных в специальном дереве каталогов, в которых содержатся также несколько дополнительных файлов с индексами и контрольными суммами пакетов. Стандартный репозиторий состоит как минимум из одного каталога, в котором хранятся ".deb"-файлы (программы) и два специальных файла: индексный файл для рабочих программ и индексный файл для исходных программ. Индексный файл содержит название, версию, размер, краткое или полное описание программы и программы, которые используются данной программой. Индексный файл для исходных программ содержит название, версию и программы, которые нужны для сборки данной программы. Кроме этого, может существовать файл, содержащий информацию о данном репозитории.

В "Ubuntu" (для операционной системы "Linux") имеются четыре основных репозитория [8].

- репозиторий, содержащий программные пакеты на момент выхода репозитория;
- репозиторий, содержащий пакеты критических обновлений безопасности;
- репозиторий, содержащий пакеты обновления системы;
- репозиторий, содержащий более новые версии ПО, которое доступно только в нестабильных версиях "Ubuntu".

Все программное обеспечение в "Ubuntu" делится на четыре секции:

- свободное ПО, которое официально поддерживается;
- официально поддерживаемые драйверы устройств;
- свободное ПО, которое официально не поддерживается;
- официально не поддерживаемые драйверы устройств.

Репозитории "Ubuntu" содержат большое число программ. Однако существуют программы, отсутствующие в репозиториях "Ubuntu". Их можно получить от сторонних репозиториях.

В "Ubuntu" предусмотрена защита репозитория от подмены. Она реализована путем сверки цифровых подписей репозитория и клиента. В случае, когда репозиторий имеет цифровую подпись, а пользовательский компьютер содержит открытый ключ для этого репозитория, такой репозиторий считается доверенным.

**Репозиторий "Sesypus"** [8] является еще одним репозиторием для "Linux". Это ежедневно обновляемый репозиторий пакетов свободных программ. В основе "Sesypus" лежит технология сборки программ и учета зависимостей между ними, а также отработанные процессы по взаимодействию между разработчиками. "Sesypus" — это набор постоянно развивающихся решений, на основе которых возможно создание универсальных или специализированных дистрибутивов Linux и одиночных внедрений.

**Репозиторий CVS** (Concurrent Version System — Система контроля версий) является хранилищем версий исходных пакетов программ, который получил широкое распространение в UNIX-подобных операционных системах. В репозиторий CVS хранится каждое изменение в тексте программы с сопутствующей записью в журнале, описывающей смысл данного изменения. Для того чтобы просмотреть эти изменения, можно пользоваться Web-интерфейсом. С помощью соответствующих утилит можно получить полную копию репозитория CVS.

**Репозиторий "Microsoft Repository" (MSR)** [9] представляет собой универсальное хранилище метаинформации о приложениях, компонентах, ХД, БД и т. д. В разработке репозитория MSR кроме корпорации Microsoft принимали участие также несколько

ведущих производителей ПО, таких как Platinum, IBM, Oracle и т. д. Репозиторий MSR доступен широкому спектру компаний, разрабатывающих корпоративные ИС и программное обеспечение.

Так как "Microsoft Repository" основан на технологии COM (Component Object Model), его функциональные возможности могут быть легко расширены. Пользователи (фирмы) могут добавлять требуемый функционал в репозиторий.

Основу MSR составляет методология объектного моделирования OIM (Open Information Model — Открытая информационная модель), предложенная корпорацией Microsoft и переданная в 1999 г. коалиции MDC [10]. В качестве языка описания моделей методология OIM предусматривает использование языка UML, а в качестве формата обмена данными — язык XML. Модель OIM структурирована по предметным областям и учитывает все возможные типы (более 300) объектов. Она открыта для всех пользователей и является функционально расширяемой и настраиваемой под конкретные нужды конкретного производителя. Для работы с данными MSR использует открытый COM-интерфейс, а для хранения данных — язык SQL. Впервые MSR был распространен в составе MS Visual Basic 5.0.

Репозиторий MSR используется:

- в качестве активного хранилища для обмена данными между средствами визуального моделирования или средствами визуальной разработки приложений. MSR является составной частью системы построения моделей. Например, система "Visual Modeler" хранит модели в MSR, что дает возможность обмениваться моделями между различными моделирующими системами. Другие моделирующие системы также могут использовать MSR для хранения своих моделей. В настоящий момент среди партнеров Microsoft, использующих репозиторий MSR в качестве хранилища моделей, свыше 50 крупных компаний. Для описания моделей, хранимых в MSR, используется стандартный язык UML;
- в качестве хранилища для разделяемых и повторно используемых компонентов при разработке и эксплуатации систем. Эту функцию выполняет программный компонент "Visual Component Manager" (VCM), являющийся составной частью системы "Visual Studio 6.0". VCM позволяет разделять между инструментами, разработчиками и приложениями такие компоненты разрабатываемой системы, как модели, файлы, файлы-заголовки, библиотеки типов, элементы управления, структуры данных и т. д. Причем список объектов, с которыми умеет работать VCM, можно расширять;
- в качестве хранилища метаданных для системы OLAP. В этом случае в репозитории MSR хранятся следующие разновидности метаинформации: схемы баз данных, откуда берутся данные

## Заключение

для ХД, операционные схемы, информация о преобразовании данных, информация о планировании событий, о моделях многомерных данных и т. п. Репозиторий в данном случае использует в качестве хранилища моделей MS SQL Server;

- в качестве репозитория приложений для бизнес-объектов и процессов. Здесь кроме основных метаданных БД, используемых в приложениях бизнес-объектов, хранятся также расширенные метаданные транзакции: структура транзакции, связь полей со справочниками, структура этих справочников и их связь друг с другом, обязательные для заполнения поля, связанные с транзакцией отчеты с параметрами их запуска, словесные описания параметров и т. д.

Репозиторий бизнес-объектов генерируется с помощью "Мастера" репозитория на сервере приложений. К репозиторию бизнес-объектов можно обратиться из любого Windows-приложения.

Таким образом, репозиторий MSR обеспечивает разделение метаданных БД, ХД и приложений, хранит модели объектов и данных, описание компонент, OLTP- и OLAP-схемы, модели протекания бизнес-процессов и т. д. MSR упрощает создание и развитие моделей, обеспечивает расширение спектра хранящихся в нем моделей, структура данных полностью открыта и хранится в обычной базе данных. Так как имеется возможность хранить различные версии и конфигурации объектов и данных, MSR представляет собой не просто хранилище типов объектов, а инструмент для управления версиями объектов, метаданных и программ.

Область использования метаданных расширяется. Поэтому репозиторий, предназначенный для хранения и управления метаданными становится важным компонентом информационных систем. Репозиторий используется также в качестве хранилища программных пакетов в Интернете. По мнению авторов, изложенные особенности репозитория для БД и ХД, рассмотренный пример построения репозитория БД и приведенный обзор репозитория программных пакетов, размещенных в Интернете, будут полезными для читателей и разработчиков информационных систем и программных средств.

## Список литературы

1. Керимов С. Г. Метаданные в информационных системах // Информационные технологии. 2003. № 5. С. 37–42.
2. Деон-Хонг Б., Плагман Б. Системы словарей-справочников данных. М.: Финансы и статистика, 1986. 311 с.
3. Словарь данных. URL: <http://ru.wikipedia.org/wiki>
4. Объекты и концепции базы данных. URL: <http://www.codenet.ru/db/interbase/ibsql/objs.php>
5. Глушаков С. В., Ломатко Д. В. Базы данных: Учебный курс. Харьков: Фолио, 2000. 504 с.
6. CentOS. URL: <http://www.centos.org/modules/tinvcntent/index.php?id=2>
7. HOWTO: репозиторий Debian. URL: <http://\ION-russian.alioth.debian.org/repository-howto.ru.html>
8. Репозиторий Sisyphus. URL: <http://sisyphus.ru>
9. Microsoft Repository: обзор свойств и стратегия развития. URL: [http://citforum.ru/seminars/cis99/visual\\_2.shtml](http://citforum.ru/seminars/cis99/visual_2.shtml)
10. Открытая информационная модель репозитория Microsoft. URL: [http://mf.grsu.by/other/lib/olap/com\\_ross/doc08.htm](http://mf.grsu.by/other/lib/olap/com_ross/doc08.htm)

## Новая книга

**Липаев В. В.**

**Проектирование и производство сложных заказных программных продуктов.** — М.: СИНТЕГ, 2011. — 400 с.

Монография состоит из двух частей, в которых изложены методы и процессы проектирования и производства сложных заказных программных продуктов для технических систем реального времени. Все компоненты и комплексы программ должны соответствовать требованиям заказчика, высокому качеству и минимальным рискам посредством верификации, тестирования, испытаний и сертификации, обеспечиваемыми коллективами квалифицированных специалистов. При изложении активно используются современные международные и отечественные стандарты, планирование производственных процессов с учетом ограниченных экономических ресурсов крупных проектов.

Монография предназначена для руководителей предприятий и проектов технических систем, для специалистов, ответственных за проектирование и производство сложных заказных программных продуктов реального времени высокого качества, также может использоваться в качестве учебного пособия по программной инженерии.



УДК 621.396.933:527.8

**С. И. Елесина**, ст. преподаватель,  
e-mail: elesinas@mail.ru,

**М. Б. Никифоров**, канд. техн. наук, доц.,  
Рязанский государственный  
радиотехнический университет

## Повышение эффективности генетического алгоритма

*Проведены исследования генетического алгоритма в целях получения оптимальных настроек при использовании его в корреляционно-экстремальных навигационных системах. Для уменьшения трудоемкости поиска глобального экстремума предлагаются различные стратегии кластеризации области поиска. Исследуется их эффективность. Показано, что для корреляционно-экстремальных навигационных систем, работающих на базе радиолокационных изображений, производительность возрастает на порядок.*

**Ключевые слова:** глобальный экстремум, целевая функция, область поиска, кластеризация, генетический алгоритм, корреляционно-экстремальные навигационные системы

Корреляционно-экстремальные навигационные системы (КЭНС) являются широко используемыми системами автоматического управления движущимися объектами. Одной из главных задач КЭНС является определение местоположения летательного аппарата в реальном времени. Принцип работы КЭНС основан на сравнении текущего изображения (ТИ), получаемого от РЛС, и эталонного изображения (ЭИ), извлекаемого из бортовой базы данных. Совмещение достигается путем определения глобального экстремума (ГЭ) функции их корреляционного совмещения. В этом случае решается задача глобальной оптимизации целевой функции (ЦФ), которую можно определить следующим образом: на двумерном пространстве  $P \subset R^2$  алгоритмически задана многоэкстремальная функция  $f(x, y)$ , при этом требуется найти

$$f_* = \max_{(x, y) \in P} f(x, y), \quad (1)$$

где  $P = \{(x, y) \in R^2: x_{\min} \leq x \leq x_{\max}, y_{\min} \leq y \leq y_{\max}\}$ .

Для радиолокационного изображения характерный вид ЦФ корреляционного совмещения изображений показан на рис. 1. В качестве ЦФ выбрана нормированная взаимная корреляционная функция

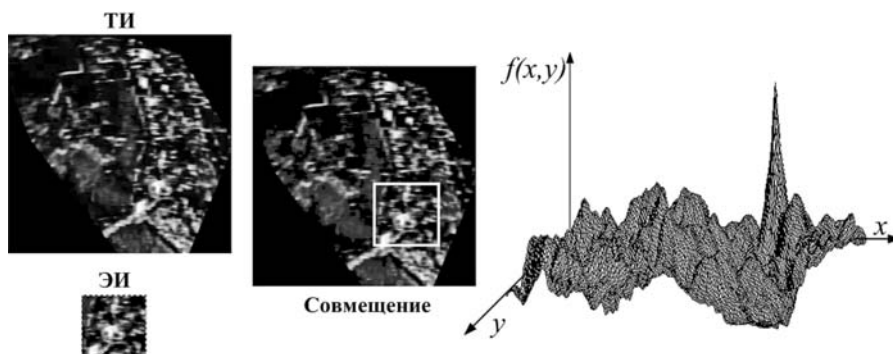
$$f(x, y) = \frac{\sum_i \sum_j (S_0(i, j) S_T(i - x, j - y))}{\sqrt{\sum_i \sum_j S_0^2(i, j) \sum_i \sum_j S_T^2(i - x, j - y)}}, \quad (2)$$

где  $S_0(i, j)$ ,  $S_T(i, j)$  — яркость эталонного и текущего изображений в точке  $(i, j)$ ;  $M, N$  — размеры ЭИ. Мульти-modalность ЦФ затрудняет поиск ГЭ в режиме реального времени, причем число локальных экстремумов может достигать нескольких сотен. В данных условиях для поиска экстремума ЦФ необходимо использовать методы глобальной оптимизации.

Наиболее надежный метод поиска ГЭ основан на полном переборе (сканировании) точек возможного взаимного расположения ТИ и ЭИ. В связи с большой трудоемкостью вычисления корреляционной функции по сравнению с другими операциями КЭНС данный метод приводит к большим вычислительным затратам, что не позволяет реализовывать его в реальном времени в бортовых системах навигации. Поэтому следует выбрать более эффективный метод поиска ГЭ, которому необходимо меньшее число вычислений ЦФ при тех же требованиях к точности решения.

В настоящее время среди методов глобальной оптимизации значительное распространение получили эволюционные алгоритмы (ЭА). Одним из самых распространенных видов ЭА являются генетические алгоритмы (ГА).

Целью работы является исследование возможности применения в КЭНС ГА в качестве метода



**Рис. 1.** Характерный вид целевой функции при совмещении текущего и эталонного изображений

глобальной оптимизации и повышение его эффективности за счет кластеризации области поиска.

Идеи ГА [1] позаимствованы у природы. Они основаны на генетических процессах живых организмов: биологические популяции развиваются в течение нескольких поколений, подчиняясь законам естественного отбора по принципу "выживает наиболее приспособленный". Подражая этому процессу, ГА способны "развивать" решения реальных задач, закодированных соответствующим образом. ГА используют эволюционные принципы наследственности, изменчивости и естественного отбора. Общая схема ГА в теории и в приложении к данной предметной области представлена на рис. 2.

ГА работает с некоторым случайным набором исходных решений, называемых популяцией особей, в хромосоме (генотип) каждой из которых закодировано возможное решение задачи (фенотип). Изначально популяция формируется случайным образом (блок "Инициализация популяций решений" на рис. 2). Оценивание качества закодированных решений проводится с помощью функции приспособленности (блок "Оценивание особей"). По результатам оценивания наиболее приспособленные особи выбираются (блок "Селекция") для скрещивания. Посредством применения генетического оператора кроссовера из выбранных особей создается потомство (блок "Скрещивание"). Созданные потомки формируют новую популяцию, причем часть потомков мутирует (используется генетический оператор мутации), что выражается в случайном изменении их генотипов (блок "Мутация"). Данная последовательность действий называется поколением. Эволюция популяции состоит из последовательности таких поколений. Результат эволюции — определение ГЭ.

В данной реализации ГА популяция — это случайным образом сгенерированные координаты взаим-

ного положения ТИ и ЭИ, которые кодируются битовыми строками с помощью кода Грея. Параметрами решения рассматриваемой задачи являются координаты взаимного положения ТИ и ЭИ, в которых ЦФ принимает максимальное значение.

В природе этап оценивания особей — это оценка того, насколько эффективен организм при конкуренции за ресурсы. В рассматриваемой задаче на этом этапе вычисляется значение ЦФ для полученных координат взаимного положения ТИ и ЭИ.

Селекция моделирует природный процесс естественного отбора. В рассматриваемой задаче селекция используется для выбора координат с лучшим значением ЦФ. В данной работе рассматриваются селекции трех видов: рулеточная, селекция усечением и турнирный отбор [1]. Рулеточная селекция считается для ГА классическим методом. Вероятность полученных координат принять участие в скрещивании пропорциональна полученной оценке, т. е. значению ЦФ. Селекция усечением использует отсортированные по возрастанию значения ЦФ. Для скрещивания выбираются  $l \cdot n$  лучших координат, где  $l$  — "порог отсечения",  $0 < l < 1$ ,  $n$  — размер популяции. Порог определяет, какие решения, начиная с координаты с максимальным значением ЦФ, будут принимать участие в отборе. Значение  $l$  определяет давление селекции. При турнирной селекции случайно выбираются  $t$  координат, и самая приспособленная из них допускается к скрещиванию. Формируется турнир из  $t$  координат, где  $t$  — размер турнира. Эта операция повторяется  $n$  раз ( $n$  — размер популяции). Чем больше значение  $t$ , тем больше давление селекции.

Оператор скрещивания (кроссовер) моделирует биологический процесс скрещивания особей, т. е. процесс обмена генетическим материалом между двумя особями. Применение скрещивания (кроссовера) необходимо для "освежения" популяции, для

того чтобы избежать преждевременной сходимости алгоритма и попадания в локальный экстремум. В данной работе используются одноточечный, двухточечный и однородный кроссовер. Одноточечный кроссовер работает аналогично операции перекреста хромосом при скрещивании биологических организмов. Пусть имеются две координаты. Выбирается произвольная точка разрыва и для создания новых координат проводится обмен битами родительских координат. Работу одноточечного кроссовера иллюстрирует рис. 3.

Для оператора двухточечного кроссовера выбираются две случайные точки разрыва, после чего для создания новых координат родительские обмениваются участками, лежащими между точками разрыва. При использовании одно-



Рис. 2. Общая схема генетического алгоритма



Рис. 3. Пример работы односточечного кроссовера

родного кроссовера разряды родительских координат наследуются независимо друг от друга. Для этого определяют вероятность  $P_0$  того, что  $i$ -й разряд координаты 1-го родителя попадет к первому потомку, а 2-го родителя — ко второму потомку. Вероятность противоположного события равна  $(1 - P_0)$ . Такая операция проводится для всех битов координаты.

Оператор мутации используется для внесения случайных изменений в хромосомы особей. В ГА же мутация (как и кроссовер) необходима для того, чтобы избежать попадания в локальный максимум и, как следствие, преждевременной сходимости алгоритма. Мутация изменяет отдельные разряды в закодированной координате. Для этого изменяемые разряды инвертируются с вероятностью  $P_m$ .

Степень оптимальности ГА во многом зависит от настройки его параметров, таких как число поколений, размер популяции, давление селекции, разновидность оператора кроссовера, вероятность кроссовера  $P_c$ , разновидность оператора мутации, вероятность мутации  $P_m$ .

В данной работе был проведен ряд исследований в целях получения оптимальных для решаемой задачи настроек. Исследования были разбиты на серии в зависимости от параметров ГА, размеров ЭИ и видов обработки ТИ. Число экспериментов в каждой серии — 300. Размер ЭИ одинаков для каждого запуска и имеет линейный размер 50 % от ТИ.

Доказано, что при данном размере ЭИ целесообразно выбирать размер популяции не менее 150 особей и число поколений не менее 20, чтобы вероятность попаданий в ГЭ превышала 95 %. При этом необходимо выбирать  $P_c$  и  $P_m$  более 25 % и 2 % соответственно.

Результаты исследований для различных видов селекции при односточечном, двухточечном и однородном кроссоверах приведены на рис. 4 (при размере популяции 200, числе поколений 20, вероятности  $P_c = 25 %$  и  $P_m = 2 %$ ). Из полученных результатов видно, что лучшие показатели имеет селекция усечением в совокупности с однородным кроссовером.

Исходя из полученных результатов исследования оптимальными настройками ГА для решения поставленной задачи при размере ЭИ 50 % от ТИ являются:

- длительность эволюции (число поколений) не менее 20;

- размер популяции не менее 150;
- интенсивность (давление) селекции 0,3...0,5;
- разновидность оператора кроссовера — однородный;
- вероятность кроссовера  $P_c$  более 25 %;
- разновидность оператора селекции — усечение;
- вероятность мутации  $P_m$  около 2 %.

При использовании этих настроек ГА обеспечивает вероятность попадания в ГЭ до 98 %. Однако трудоемкость данного метода [2] достаточно высока: более 2000 вычислений ЦФ. Поэтому совершенствование данного метода должно быть направлено на сокращение трудоемкости данного алгоритма при сохранении и даже увеличении вероятности и точности попадания в ГЭ.

При работе ГА на этапе скрещивания, т. е. получения новых координат взаимного положения ЭИ и ТИ, могут быть получены координаты, значения ЦФ в которых уже вычислялись ранее. Поэтому введение массива  $M$  уже вычисленных значений ЦФ может сократить число вычислений ЦФ.

Для увеличения вероятности и точности попадания в ГЭ предлагается использовать два этапа — глобальный и локальный [3]. Так как ГА в большинстве случаев находит приближенное значение ГЭ, то для его уточнения необходимо использовать уточняющие методы локальной оптимизации. В качестве локального метода выбран метод деформируемого многогранника (МДМ) [4], так как в этом методе сразу выбираются три точки для начального поиска экстремума, а не одна, как в других методах. Это позволяет лучше "чувствовать" пространство поиска и соответственно увеличивать вероятность попадания в ГЭ.

Оценка эффективности этих двух предложений иллюстрируется на рис. 5. Только использование массива  $M$  для хранения ранее вычисленных значений ЦФ позволяет уменьшить трудоемкость ГА более чем в 3 раза.

Дальнейшее уменьшение трудоемкости ГА можно осуществить за счет кластеризации области поиска. Допустимый размер кластера определяется расстоянием между локальными экстремумами.

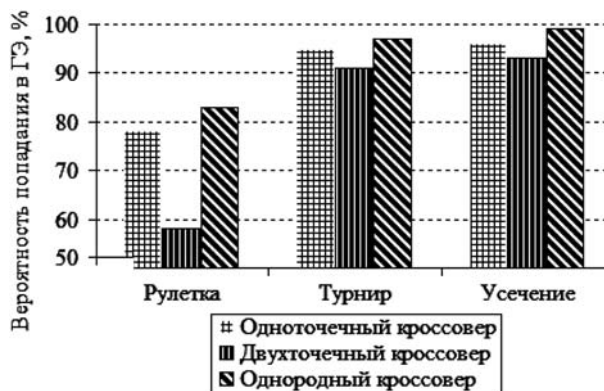


Рис. 4. Зависимость вероятности попаданий в глобальный экстремум от различных типов селекции и кроссовера

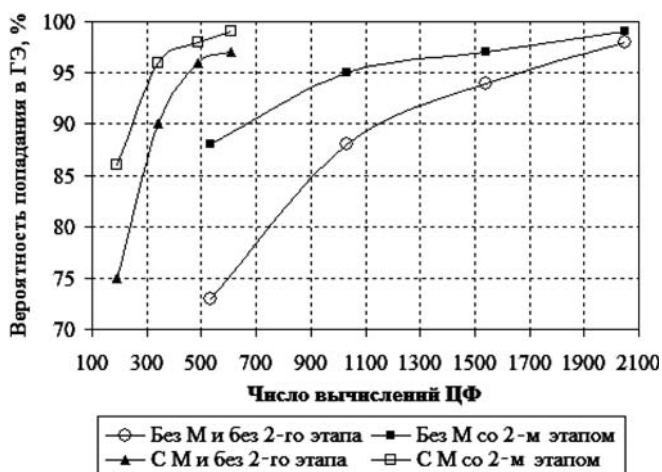


Рис. 5. Зависимость вероятности попаданий в глобальный экстремум от числа вычислений целевой функции при использовании 2-го этапа и массива значений целевой функции

Предварительные статистические исследования корреляционной функции, полученной в результате совмещений двух изображений, позволяют получить оценки среднего расстояния между локальными экстремумами  $d_{cp}$ :

$$d_{cp} = \frac{\sum_{i=1}^K \sum_{j=1, j \neq i}^K d_{ij}}{(K-1)^2}, \quad (3)$$

где  $K$  — число локальных экстремумов;  $d_{ij}$  — расстояние между  $i$ -м и  $j$ -м локальными экстремумами:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad (4)$$

где  $(x_i, y_i)$  и  $(x_j, y_j)$  — координаты  $i$ -го и  $j$ -го локальных экстремумов.

Сокращение трудоемкости осуществляется за счет того, что в каждом кластере значение ЦФ вычисляется только один раз.

Пусть для кластеров, имеющих форму квадрата с длиной стороны, равной  $a$ , его размер определяется параметром

$$r = (a - 1)/2. \quad (5)$$

В этом случае кластер, состоящий из одной точки (пикселя), имеет размер  $r = 0$ , т. е. кластеризация отсутствует. Кластер, который состоит из центрального пикселя и восьми пикселей, окружающих его одним слоем, имеет размер, равный  $r = 1$ , и т. д.

Для надежного выделения ГЭ необходимо, чтобы было вычислено не менее двух отсчетов значений ЦФ на интервале  $d_{cp}$ . В этом случае с учетом ортогональности сторон кластера  $a \leq d_{cp} \cdot \sqrt{2}/4$ . При этом допустимый размер кластера с учетом формулы (5) определяется отношением

$$r \leq (d_{cp} - 2) \cdot \sqrt{2}/8. \quad (6)$$

В данной работе исследуется эффективность кластеризации области поиска ГЭ ЦФ при использовании различных стратегий заполнения значений функции внутри кластера:

- интерполяция нулевого порядка (тиражирование) значений функции внутри кластера размером  $r > 0$ ;
- однократная интерполяция первого порядка значений функции внутри кластера размером  $r > 0$ ;
- многократная интерполяция первого порядка значений функции внутри кластера размером  $r > 0$ .

При интерполяции нулевого порядка предполагается тиражирование каждого заносимого в массив истинного значения ЦФ по всем неинициализированным элементам кластера вокруг него. Элементы кластера, которые уже были определены на предыдущих итерациях, переопределяются не подвергаются.

Интерполяция осуществляется для точек, которые являются общими для нескольких кластеров, с учетом весового влияния их центров. Однократная выполняется только один раз для каждой точки, ставшей общей впервые при пересечении двух кластеров.

Однократная интерполяция использует уже интерполяцию первого порядка и на каждой итерации затрагивает не только не определенные ранее элементы кластера, но и получившие свои значения путем тиражирования непосредственно вычисленного значения элемента массива на одной из прошлых итераций. Иллюстрация стратегии приведена на рис. 6, на нем область 1 — кластер на одной из прошлых итераций. В ходе его обработки непосредственно вычисленное значение ЦФ из центра кластера (точка  $A$ ) было растиражировано, в том числе и в неопределенную ранее точку  $B$ . Область 2 — кластер, обрабатываемый на текущей итерации. Ранее тиражированное в точку  $B$  значение уточняется.

Векторы  $AB$  и  $CB$  отражают влияние старого (точка  $A$ ) и нового (точка  $C$ ) центров на рассматриваемую точку  $B$ . Влияние каждого из центров обратно пропорционально длине соответствующего вектора:

$$f(B) = \frac{f(A)l_{CB} + f(C)l_{AB}}{l_{CB} + l_{AB}}, \quad (7)$$

где  $f(A), f(B), f(C)$  — значение ЦФ в соответствующих точках;  $l_{CB}, l_{AB}$  — длины соответствующих векторов.

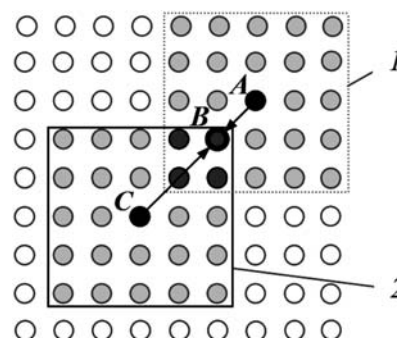


Рис. 6. Однократная интерполяция

Таким образом, получаем следующий алгоритм.

**Шаг 1.** Если очередное запрошенное значение ЦФ уже есть в массиве, оно берется оттуда. Выход.

**Шаг 2.** Если соответствующий элемент массива еще не определен, происходит вычисление ЦФ и ее значение записывается в соответствующий элемент массива.

**Шаг 3.** Перебор элементов массива в границах кластера, центром которого является элемент, определенный на шаге 2.

**Шаг 3.1.** Если очередной элемент массива в кластере не определен, в него записывается значение центрального элемента, координаты (индексы столбца и строки в массиве) этого центрального элемента записываются в соответствующие поля текущего элемента кластера.

**Шаг 3.2.** Если в очередном элементе кластера уже имеется значение, то оно уточняется в зависимости от соотношения длин двух векторов уточнения (7). Уточненное значение сохраняется в этом элементе кластера, и данное значение на последующих итерациях не уточняется. Это и дает методу название однократной интерполяции.

**Шаг 4.** Значение центрального элемента кластера возвращается в процедуру, реализующую алгоритм поиска экстремума.

Данный алгоритм выполняется каждый раз, когда алгоритм поиска экстремума запрашивает значение ЦФ в очередной поисковой точке.

Множественная интерполяция также основана на интерполяции первого порядка, но снимает ограничение на число уточнений неистинных значений в массиве. Пример работы алгоритма на первых двух итерациях показан на рис. 7, а. Область 1 — кластер на одной из прошлых операций. В ходе его обработки непосредственно вычисленное значение ЦФ из центра кластера (точка *A*) было растражировано, в том числе и в не определенную ранее точку *B*. Область 2 — кластер, обрабатываемый на текущей итерации. Ранее записанное в точке *B* значение уточняется.

Для каждого уточненного на некоторой итерации значения фиксируется точка-источник этого значения (точка *D*). Она находится суммированием векторов уточнения на двух последних итерациях, на которых данное значение изменялось (векторы

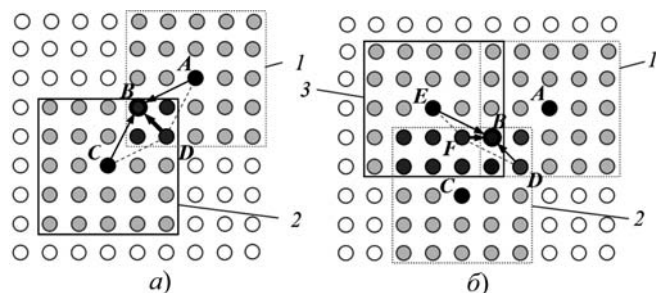


Рис. 7. Множественная интерполяция

*CB*, *AB* соответственно и найденный по принципу параллелограмма суммарный вектор *DB*).

При первом обращении к элементу (т. е. при тиражировании) точкой-источником являлась точка *A*, на более же поздних итерациях точка, заявленная в качестве источника, может даже не быть определенной. Она сохраняется лишь для продолжения алгоритма — на основе сохраненных координат этой точки вычисляется длина одного из двух векторов уточнения (*DB*), значение уточняется, определяется новая фиктивная точка-источник, и алгоритм продолжается (на рис. 7, б область 3 соответствует новому кластеру).

Алгоритмические отличия от стратегии однократной интерполяции проявляются на шаге 3.2.

**Шаг 3.2.** Если в очередном элементе кластера (точка *B* на рис. 7, а) уже имеется значение, то оно уточняется в зависимости от соотношения длин двух векторов уточнения. Началом одного из них является центр текущего кластера, другого — точка, координаты которой были сохранены в рассматриваемом элементе кластера, а общим концом векторов является точка, соответствующая рассматриваемому элементу кластера (на рис. 7, а это векторы *CB* и *AB* соответственно). По принципу параллелограмма находится сумма этих векторов, и координаты начала этого суммарного вектора сохраняются в специальных полях текущего элемента массива для использования на последующих итерациях. Уточненное значение также сохраняется в одном из полей этого элемента.

Для сравнения эффективности предложенных стратегий кластеризации области поиска экстремума ЦФ был проведен ряд модельных экспериментов для определения зависимости надежности и точности нахождения ГЭ ЦФ от числа вычислений ЦФ для различных значений размера кластера *r*.

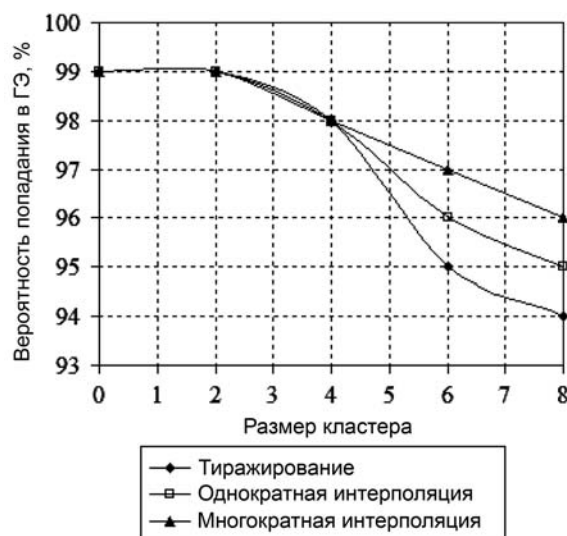


Рис. 8. Зависимость вероятности попадания в глобальный экстремум от размера кластера при различных видах интерполяции

Исследования проводились для ЭИ размером 50 % от ТИ. Число экспериментов в каждой серии 300.

На рис. 8 приведена зависимость вероятности попадания в ГЭ от размера кластера для различных стратегий. Все три стратегии дают практически одинаковый результат при  $r \leq 4$ . При больших размерах кластера  $r > 4$  небольшой выигрыш (приблизительно 1...2 %) имеет многократная интерполяция, при

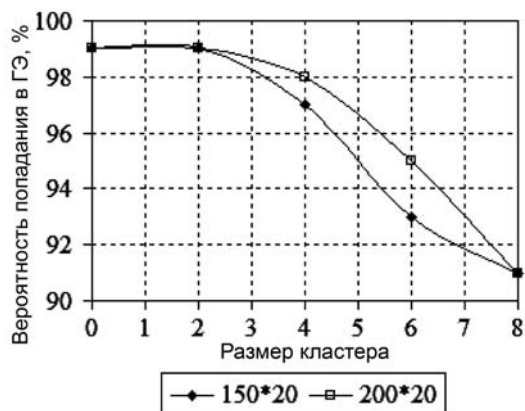


Рис. 9. Зависимость вероятности попадания в глобальный экстремум от размера кластера

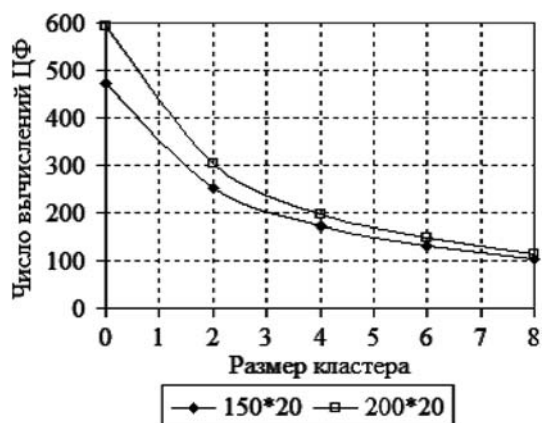


Рис. 10. Зависимость числа вычислений целевой функции от размера кластера

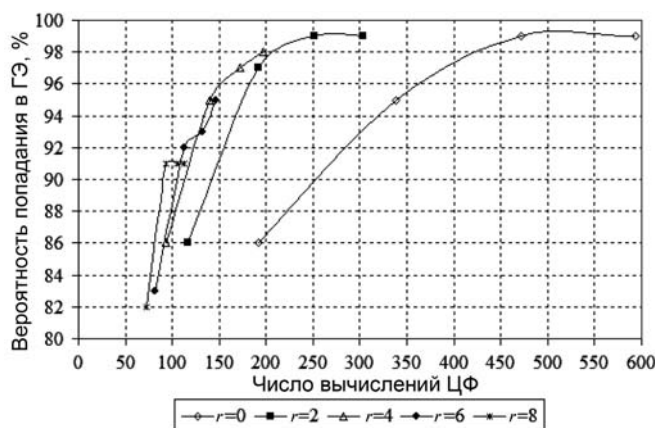


Рис. 11. Зависимость вероятности попадания в глобальный экстремум от числа вычислений целевой функции при различных размерах кластера

этом наблюдается падение вероятности попадания в ГЭ. Таким образом, наиболее рациональным является использование интерполяции нулевого порядка (тиражирования) как наименее трудоемкой при размере кластера  $r \leq 4$ .

На рис. 9 приведена зависимость вероятности попадания в ГЭ от размера кластера при числе поколений 20 и размере популяции 150 и 200. На рис. 10 приведена зависимость числа вычислений ЦФ от размера кластера.

На рис. 11 представлена зависимость вероятности попадания в ГЭ от числа вызовов ЦФ при различных размерах кластера.

Анализ приведенных графиков показывает, что при размере кластера  $r \leq 2$  вероятность попадания в ГЭ не уменьшается, а число вычислений ЦФ уменьшается в 2 раза. Оптимальный размер кластера  $r = 4$ , тогда при числе вызовов ЦФ, равном 200, вероятность попадания в ГЭ достигает 98 %.

\*\*\*

Применение кластеризации области поиска и позволяет сократить число вычислений ЦФ на порядок. При этом сохраняется вероятность попадания в ГЭ на уровне не менее 98 %. Оптимальный размер кластера для данного вида ТИ, в качестве которого используется радиолокационное изображение, равен 4.

Аналитическая оценка (6) размера кластера, равная 6, оказалась завышенной вследствие того, что поисковые методы нахождения поиска ГЭ неравномерно используют область поиска, а также вследствие того, что  $d_{ср}$  является случайной величиной с определенным среднеквадратическим отклонением координат локальных экстремумов для различных положений ЭИ. На основе результатов экспериментальных исследований следует применить более жесткое ограничение размера кластера

$$r \leq (d_{ср} - 2)/8. \quad (8)$$

Так как выигрыш от использования многократной интерполяции незначительный, но он требует дополнительных вычислений, то оптимальной стратегией можно считать интерполяцию нулевого порядка.

Введение локального этапа поиска ГЭ и использование на этом этапе поискового метода Нелдера—Мида (МДМ) позволяет увеличить надежность нахождения ГЭ.

#### Список литературы

1. Цой Ю. Р., Спицын В. Г. Генетический алгоритм: учеб. пособие. Томск: Изд-во ТПУ, 2006. 146 с.
2. Бабаев С. И., Елесина С. И., Костров Б. В. Сравнение модифицированного поискового и генетического алгоритмов нахождения глобального экстремума в системах навигации // Вопросы радиоэлектроники. Сер. ОТ. 2010. Вып. 1. С. 145—152.
3. Жиглявский А. А., Жилинскис А. Г. Методы поиска глобального экстремума. М.: Наука, Гл. ред. физ.-мат. лит., 1991. 248 с.
4. Пантелеев А. В., Летова Т. А. Методы оптимизации в примерах и задачах: учеб. пособие. М.: Высшая школа, 2002. 544 с.

**В. В. Васильев,**

д-р техн. наук, чл.-корр. НАН Украины,

**Л. А. Симак,** д-р техн. наук,

Отделение гибридных моделирующих  
и управляющих систем в энергетике ИПМЭ

им. Г. Е. Пухова НАН Украины,

e-mail: vsvv06@gmail.com

## **Полиномиальная аппроксимация сигналов как операционное исчисление: применение к моделированию динамических систем дробного порядка**

*Аналитическая аппроксимация непрерывных сигналов в виде полиномов наилучшего приближения с различными системами базисных функций рассматривается как операционное исчисление неклассического типа. Показан порядок применения такого исчисления к математическому моделированию динамических систем как целого, так и дробного порядков с использованием программных сред компьютерной алгебры. Приведены иллюстративные примеры аппроксимации сигналов и решения дифференциальных уравнений дробного порядка. Обсуждаются преимущества и недостатки предлагаемого подхода и перспективы дальнейших исследований.*

**Ключевые слова:** математическое моделирование, дифференциальные преобразования, аппроксимация сигналов, операционное исчисление, динамическая система

**Введение.** Математическое и компьютерное моделирование современных процессов и систем различной физической природы неразрывно связано с обработкой сигналов при исследовании нестационарных явлений. К аналоговой и цифровой обработке сигналов относятся такие проблемы как сжатие информации, фильтрация и выделение сигналов на фоне шумов и помех различного вида, оценивание информативных параметров сигналов, широкий круг задач, связанных с распознаванием образов. Многие из перечисленных выше задач требуют использования методов аналитической аппроксимации сигналов и обработки результатов наблюдений и измерений. Изучение нестационарных процессов в динамических системах при моделировании требует использования методов алгебраизации интегро-дифференциальных уравнений, составляющих основу математической модели исследуемой системы. К таким методам относятся широко известные методы интегральных преобразований, преобразование Лапласа, Фурье, дифференциальные преоб-

разования Пухова,  $z$ -преобразование, конечно-разностные методы, методы конечных элементов и др. [4, 6, 7]. Каждый из этих методов имеет свои достоинства, недостатки и области эффективного применения. Так, преобразование Лапласа эффективно при исследовании непрерывных линейных динамических систем. Преобразование Фурье используется в случае периодических процессов,  $z$ -преобразование применимо для изучения дискретных динамических систем. Между тем, сложность обратного преобразования Лапласа и невозможность его применения для нелинейных систем ограничивает область его использования. Дифференциальные преобразования Пухова сталкиваются со значительными трудностями при моделировании систем, сигналы которых сопровождаются помехами. Значительный интерес, который вызвали исследования в области дробной динамики и математического анализа дробных порядков [5, 8, 9], способствуют разработке новых численных и численно-аналитических методов моделирования. Одним из таких методов является метод  $S$ -преобразования [1], развиваемый в данной работе.

Работа построена следующим образом. Сначала изложены основы аппроксимации непрерывных сигналов обобщенными полиномами с различными системами базисных функций и сформулирована задача нахождения коэффициентов аппроксимирующего полинома наилучшего приближения по методу наименьших квадратов. Затем соотношения полиномиальной аппроксимации рассматриваются с позиций операционного исчисления неклассического типа, для которого используется термин " $S$ -преобразование", и рассматривается методика применения этого преобразования для алгебраизации интегро-дифференциальных математических моделей динамических систем, включающих дифференциальные операторы целых и дробных порядков. Изложение сопровождается иллюстративными примерами аппроксимации сигнала и решения дифференциального уравнения дробного порядка. Вычислительные эксперименты выполнены в программной системе Mathematica® [10].

**Аппроксимация сигналов.** Существо полиномиальной аппроксимации сигналов может быть представлено следующим образом. Задан сигнал  $x(t)$ , определенный на интервале изменения аргумента  $t \in [0, T]$ . Задана также система линейно независимых базисных функций:  $\mathbf{S}(t) = \{s_1(t), s_2(t), \dots, s_m(t)\}^*$ , определенных на том же интервале изменения аргумента. Под полиномиальной аппроксимацией сигнала понимают выражение

$$x_a(t) = \mathbf{X}^* \cdot \mathbf{S}(t) = \sum_{i=1}^m X_i \cdot s_i(t), \quad (1)$$

где  $\mathbf{X} = \{X_1, X_2, \dots, X_m\}^*$  — вектор коэффициентов аппроксимирующего полинома (1);  $\bullet$  — знак скалярного произведения векторов.

Полином наилучшего приближения выбирают исходя из минимизации нормы функции ошибки аппроксимации [3]. Одной из таких норм, применяемых в методе наименьших квадратов, является функционал в виде интеграла квадрата функции ошибки на интервале аппроксимации:

$$\mu = \int_0^T (x(t) - x_a(t))^2 dt = \int_0^T \left( x(t) - \sum_{i=1}^m X_i s_i(t) \right)^2 dt \Rightarrow \min. \quad (2)$$

Условием минимума функционала  $\mu$  является равенство нулю частных производных (2) по аргументам  $X_j$ :

$$\frac{\partial \mu}{\partial X_j} = 0, j := 1, \dots, m. \quad (3)$$

Выражение (3) порождает систему линейных алгебраических уравнений порядка  $m$  относительно искомым коэффициентов аппроксимирующего полинома, решение которой в векторно-матричной форме имеет вид:

$$\mathbf{X} = \mathbf{W}^{-1} \cdot \mathbf{Q}. \quad (4)$$

Элементы матрицы  $\mathbf{W}$  и вектора  $\mathbf{Q}$ , входящие в формулу (4), определяются выражениями:

$$w_{ij} = \int_0^T s_i(t) \cdot s_j(t) dt, (i, j := 1, \dots, m), \quad (5)$$

$$q_i = \int_0^T s_i(t) \cdot x(t) dt, (i := 1, \dots, m). \quad (6)$$

С учетом (5), (6), матрица  $\mathbf{W}$  и вектор  $\mathbf{Q}$  могут быть записаны в виде

$$\mathbf{W} = \int_0^T \mathbf{S}(t) \cdot \mathbf{S}^*(t) dt, \quad (7)$$

$$\mathbf{Q} = \int_0^T \mathbf{S}(t) x(t) dt. \quad (8)$$

В качестве систем базисных функций могут быть использованы различные системы (степенные и экспоненциальные, тригонометрические, ортогональные полиномы, локально-импульсные типа Уолша, Радемахера и т. п.) [1, 3]. Выбор подходящих систем базисных функций представляет непростую задачу и в значительной степени зависит от свойств и особенностей аппроксимируемых сигналов. В случае ортогональных систем базисных функций решение аппроксимационной задачи существенно упрощается, так как матрица  $\mathbf{W}$  стано-

вится диагональной, а для ортонормированных систем вырождается в единичную. Весьма перспективным для решения задач аппроксимации и моделирования динамических систем является использование в качестве систем базисных функций смещенных и преобразованных полиномов Лежандра [1, 3]. Вектора коэффициентов аппроксимирующих полиномов в этом случае позволяют оценить информативные параметры сигналов (средние значения сигналов и их производных различных порядков) [2], что в ряде случаев удобно для дискретизации и моделирования процессов обобщенного движения и оценивания различных параметров (обобщенных скоростей, ускорений и т. п.).

**Операционный метод "S-преобразование"**. Совокупность формул, определяющих полиномиальную аппроксимацию сигналов (1), (4–8), может трактоваться как пара операционных преобразований, называемых  $S$ -преобразованием [1]:

$$x(t) \Rightarrow \mathbf{X} = \left( \int_0^T \mathbf{S}(t) \cdot \mathbf{S}^*(t) dt \right)^{-1} \cdot \left( \int_0^T \mathbf{S}(t) \cdot x(t) dt \right) \Rightarrow \Rightarrow x_a(t) = \mathbf{X} \cdot \mathbf{S}(t). \quad (9)$$

Прямое операционное преобразование ставит в соответствие сигналу  $x(t)$  его изображение в виде вектора  $\mathbf{X}$  коэффициентов аппроксимирующего полинома наилучшего приближения. Обратное операционное преобразование восстанавливает оригинал сигнала в виде его аппроксимации. Каждая система базисных функций порождает свою разновидность преобразования.

Преимуществом  $S$ -преобразования по сравнению с известным преобразованием Лапласа является простота обратного преобразования. Существенным недостатком — приближенный характер и конечность интервала, на котором осуществляется аппроксимация сигналов. Для  $S$ -преобразования, как и для любого другого операционного исчисления, существуют правила операционной алгебры, определяющие основные свойства сигналов в области изображений. Так, линейной комбинации сигналов соответствует такая же линейная комбинация изображений:

$$z(t) = ax(t) \pm by(t) \Leftrightarrow \mathbf{Z} = a\mathbf{X} \pm b\mathbf{Y}. \quad (10)$$

Операции интегрирования сигнала с переменным верхним пределом соответствует операция матричного умножения изображения подинтегральной функции на матрицу, элементы которой зависят от системы базисных функций и порядка интегрального оператора:

$$y(t) = \int_0^T x(\tau) d\tau \Leftrightarrow \mathbf{Y} = \mathbf{P}_s^{(1)} \cdot \mathbf{X}. \quad (11)$$

Порядок применения  $S$ -преобразования для алгебраизации интегро-дифференциальных уравнений может быть проиллюстрирован на примере



простейшего обыкновенного дифференциального уравнения первого порядка (задача Коши):

$$\frac{dy(t)}{dt} + ay(t) = f(t), y(0) = y_0. \quad (12)$$

Преобразуем (12) к эквивалентному интегральному уравнению, введя замену переменных  $u(t) = \frac{dy(t)}{dt}$ :

$$u(t) + a \int_0^t u(\tau) d\tau = f(t) - ay_0. \quad (13)$$

Операционным аналогом интегрального уравнения (13) является следующая система линейных алгебраических уравнений в матричной форме:

$$\mathbf{U} + a\mathbf{P}^{(1)} \cdot \mathbf{U} = \mathbf{F} - ay_0 \cdot \mathbf{1}. \quad (14)$$

Разрешая (14) относительно неизвестного изображения первой производной искомого решения дифференциального уравнения, получим

$$\mathbf{U} = (\mathbf{E} + a\mathbf{P}^{(1)})^{-1} \cdot (\mathbf{F} - ay_0 \cdot \mathbf{1}). \quad (15)$$

Изображение решения дифференциального уравнения (12) определяется выражением

$$\mathbf{Y} = y_0 \cdot \mathbf{1} + \mathbf{P} \cdot \mathbf{U}. \quad (16)$$

Выполнив обратное операционное преобразование, получим

$$y_a(t) = \mathbf{Y} \cdot \mathbf{S}(t). \quad (17)$$

Еще одним существенным преимуществом  $\mathcal{S}$ -преобразования является то, что оно не меняет своих свойств в случае применения его к интегро-дифференциальным уравнениям, включающим наряду с обычными интегро-дифференциальными операторами дробного порядка. Поскольку изложение основ дробного исчисления не входит в задачу данной работы, ограничимся ссылкой на известные публикации [1, 8, 9] и приведем два определения интегралов и производных дробного порядка, которые будут далее использованы в иллюстративных примерах. Одним из наиболее употребительных определений операторов дробного порядка является определение интеграла Римана—Лиувилля [1, 8]:

$$y(t) = {}^{RL}J^\beta(x(t)) = \frac{1}{\Gamma(\beta)} \int_0^t (t - \tau)^{\beta-1} x(\tau) d\tau. \quad (18)$$

В выражении (18) приняты следующие обозначения:  ${}^{RL}J^\beta$  — интегральный оператор Римана—Лиувилля порядка  $\beta$ ;  $\Gamma(\beta)$  — гамма-функция.

Интеграл Римана—Лиувилля является обобщением известной интегральной формулы Коши и при целых значениях порядка превращается в определение повторного интеграла. Производная дробного порядка может вводиться различными способами. Укажем на два определения: Римана—Лиувилля и Капуто [8]. Производная дробного порядка  $\beta$ , удовлетворяющего условию  $n - 1 < \beta < n$  ( $n$  — целое),

вводится как производная целого порядка  $n$  от интеграла Римана—Лиувилля порядка  $n - \beta$ :

$$\begin{aligned} {}^{RL}D^\beta(x(t)) &= \frac{d^n}{dt^n} ({}^{RL}J^{n-\beta}(x(t))) = \\ &= \frac{d^n}{dt^n} \left( \frac{1}{\Gamma(n-\beta)} \int_0^t (t-\tau)^{n-\beta-1} x(\tau) d\tau \right). \end{aligned} \quad (19)$$

В технических приложениях используется производная дробного порядка по Капуто [8], которая определяется выражением:

$$\begin{aligned} {}^C D^\beta(x(t)) &= {}^{RL}J^{n-\beta} \left( \frac{d^n x(t)}{dt^n} \right) = \\ &= \frac{1}{\Gamma(n-\beta)} \int_0^t (t-\tau)^{n-\beta-1} \frac{d^n x(\tau)}{d\tau^n} d\tau. \end{aligned} \quad (20)$$

В формулах (19), (20) приняты следующие обозначения для операторов дифференцирования дробного порядка:

${}^{RL}D^\beta$  — оператор дифференцирования дробного порядка по Риману—Лиувиллю;

${}^C D^\beta$  — оператор дифференцирования дробного порядка по Капуто.

В области  $\mathcal{S}$ -преобразования важным является то, что операционное правило для интеграла дробного порядка по Риману—Лиувиллю сохраняет свой вид по сравнению с аналогичным правилом для интеграла целого порядка:

$$y(t) = \frac{1}{\Gamma(\beta)} \int_0^t (t - \tau)^{\beta-1} x(\tau) d\tau \Leftrightarrow \mathbf{Y} = \mathbf{P}_s^{(\beta)} \cdot \mathbf{X}. \quad (21)$$

Рассмотрим примеры, иллюстрирующие порядок решения задач аппроксимации непрерывного сигнала и применение  $\mathcal{S}$ -преобразования к решению дифференциального уравнения дробного порядка с производной по Капуто.

### Иллюстративные примеры

#### Пример 1. Аппроксимация непрерывного сигнала степенным полиномом

Исходные данные:

- сигнал  $x(t) = \sin(\pi t) + 2e^{-50(t-0,25T)^2}$ ;
- диапазон изменения аргумента:  $0 \leq t < T = 2$ ;
- порядок полинома  $m = 10$ .

#### Программа 1

- Задание исходных данных:

```
m := 10; T := 2;
x[t_] := Sin[π*t] + 2e-50(t - 0,25T)^2.
```

- Формирование базисной системы функций (степенной полином):

```
s[t_, T_, i_] := (t/T)i-1;
```

```
S = Table[s[t, T, i], {i, m}].
```

- Определение матрицы  $W$ :

$$W := \text{Table} \left[ \int_0^T s[t, T, i] * s[t, T, j] dt, \right. \\ \left. \{i, m\}, \{j, m\} \right]$$

- Определение вектора  $Q$ :

$$Q := \text{Table} \left[ \int_0^T s[t, T, i] * x[t] dt, \{i, m\} \right].$$

- Нахождение вектора коэффициентов аппроксимирующего полинома:

$$X = \text{Inverse}[W].Q.$$

- Формирование аппроксимирующего полинома:

$$xa = X.S.$$

- Аппроксимирующий полином:

$$xa \\ -0,105589 + 14,5258t - 207,307t^2 + \\ + 1294,3t^3 - 3630,1t^4 + 5396,08t^5 - \\ - 4591,16t^6 + 2251,29t^7 - 592,633t^8 + \\ + 64,9143t^9$$

- Визуализация сигнала и его аппроксимации (рис. 1):

$$\text{Plot}[\{x[t], xa\}, \{t, 0, T\}]$$

- Визуализация функции ошибки аппроксимации сигнала (рис. 2):

$$\text{Plot}[x[t] - xa, \{t, 0, T\}]$$

Как следует из анализа результатов аппроксимации, степенная система базисных функций в на-

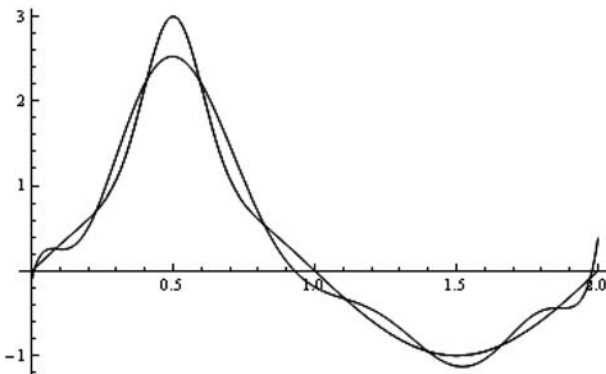


Рис. 1. Сигнал и его аппроксимация степенным полиномом

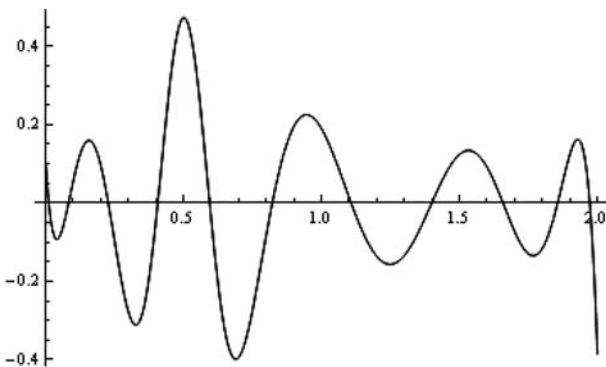


Рис. 2. Функция ошибки аппроксимации сигнала

шем случае не позволила получить удовлетворительное приближение к сигналу. Кроме того, большой разброс значений коэффициентов аппроксимирующего полинома ( $X_6/X_1 \approx 54000$ ) свидетельствует о плохой обусловленности обратной матрицы  $W^{-1}$ . В связи с этим рассмотрим пример использования локально-импульсной базисной системы на основе смещенных полиномов Лежандра для аппроксимации того же сигнала.

*Пример 2. Аппроксимация непрерывного сигнала на основе базисных систем с локальными полиномами Лежандра.*

Исходные данные:

- сигнал  $f(t) = \sin(\pi t) + 2e^{-50(t-0,25T)^2}$ ;
- диапазон изменения аргумента  $0 \leq t < T = 2$ ;
- порядок базисной системы функций  $m = 100$ ;
- шаг дискретизации аргумента  $h = 0,02$ .

### Программа 2

- Задание исходных данных:

$$m := 100; h := 0,02; T := 2; \\ f[t_] := \text{Sin}[\pi * t] + 2e^{-50(t - 0,25T)^2};$$

- Формирование подсистем базисных функций (смещенных полиномов Лежандра нулевого, первого и второго порядков:

$$v[t_, m_, h_] := \text{Table}[\text{If}[i - 1] * h \leq \\ \leq t < i * h, 1, 0], \{i, m\};$$

$$w[t_, m_, h_] := \text{Table}[(2t/h - 1 - \\ - 2(i - 1)) * v[t, m, h][[i]], \{i, m\};$$

$$u[t_, m_, h_] := \text{Table}[(6(t/h)^2 - \\ - (6/h) * (2i - 1) * t + 6i^2 - 6i + \\ + 1) * v[t, m, h][[i]], \{i, m\};$$

$$V = v[t, m, h]; W = w[t, m, h];$$

$$U = u[t, m, h];$$

- Визуализация некоторых составляющих подсистем базисных функций (рис. 3):

$$\text{Plot}[\{V[[1]], W[[3]], U[[5]]\}, \\ \{t, 0, 10h\}]$$

- Нахождение вектора коэффициентов аппроксимирующего полинома нулевого порядка:

$$F0 = \text{Table}[(1/h) * \int_{(i-1)*h}^{i*h} f[t] dt, \{i, m\};$$

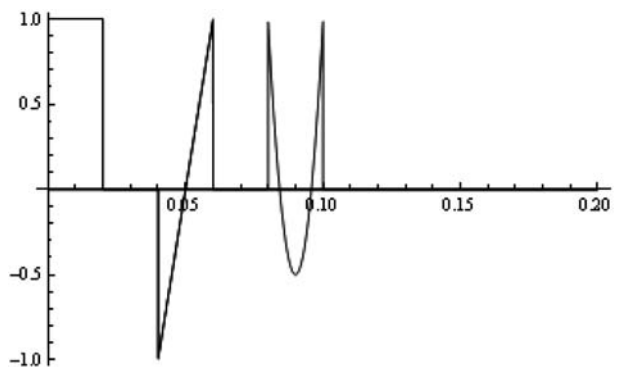


Рис. 3. Базисные функции  $v_1(t)$ ,  $w_3(t)$ ,  $u_5(t)$

- Нахождение фрагмента вектора коэффициентов аппроксимирующего полинома первого порядка:

$$F1 = \text{Table}\left[\left(\frac{3}{h}\right) \int_{(1-1)h}^{1h} f[t] \cdot (2t/h + 1 - 2i) dt, \{i, m\}\right];$$

- Нахождение фрагмента вектора коэффициентов аппроксимирующего полинома второго порядка:

$$F2 = \text{Table}\left[\left(\frac{5}{h}\right) \int_{(1-1)h}^{1h} f[t] \cdot (6(t/h)^2 - 6t \cdot (2i - 1)/h + 6i^2 - 6i + 1) dt, \{i, m\}\right];$$

- Формирование аппроксимаций сигнала соответственно нулевого, первого и второго порядков системы смещенных полиномов Лежандра:

$$fa0 = F0.V; \quad fa1 = fa0 + F1.W; \\ fa2 = fa1 + F2.U;$$

- Визуализация сигнала и его кусочно-постоянной аппроксимации (рис. 4):

$$\text{Plot}\left[\{f[t], fa0\}, \{t, 0, T\}, \text{PlotPoints} \rightarrow 200\right]$$

- Визуализация сигнала и его кусочно-линейной аппроксимации (рис. 5):

$$\text{Plot}\left[\{f[t], fa1\}, \{t, 0, T\}, \text{PlotPoints} \rightarrow 200\right]$$

Визуализация кусочно-параболической аппроксимации сигнала не приводится, так как отличия от сигнала очень малы и невидимы невооруженным глазом.

- Визуализация фрагментов функций ошибки кусочно-постоянной, кусочно-линейной и кусочно-параболической аппроксимаций (рис. 6–8):

$$\text{Plot}\left[f[t] - fa0, \{t, 0, 10h\}, \text{PlotPoints} \rightarrow 200\right]$$

$$\text{Plot}\left[f[t] - fa1, \{t, 0, 10h\}, \text{PlotPoints} \rightarrow 200\right]$$

$$\text{Plot}\left[f[t] - fa2, \{t, 0, T\}, \text{PlotPoints} \rightarrow 200, \text{PlotRange} \rightarrow \text{All}\right]$$

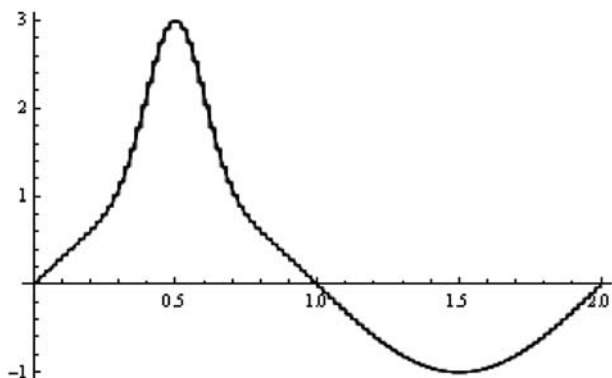


Рис. 4. Сигнал и его кусочно-постоянная аппроксимация

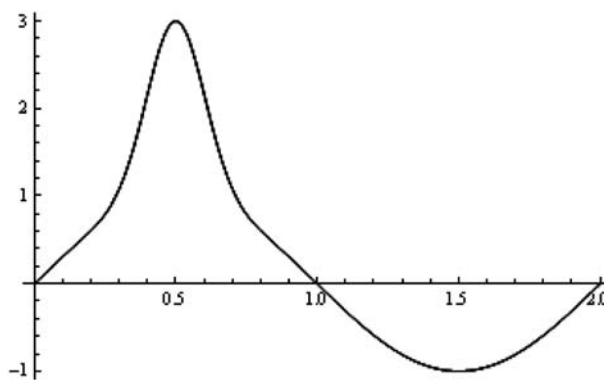


Рис. 5. Сигнал и его кусочно-линейная аппроксимация

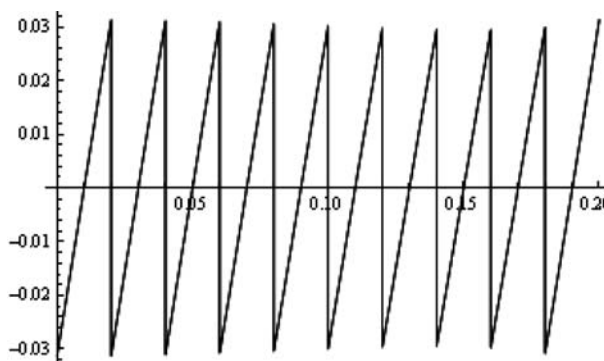


Рис. 6. Фрагмент функции ошибки кусочно-постоянной аппроксимации

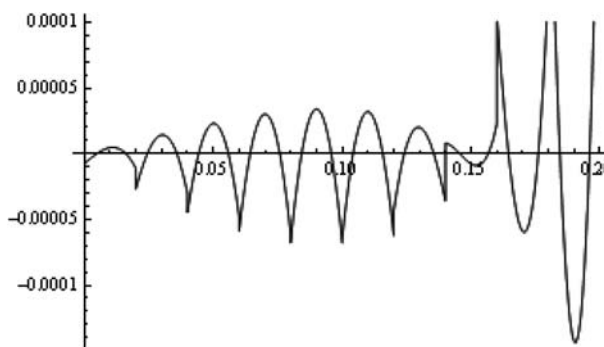


Рис. 7. Фрагмент функции ошибки кусочно-линейной аппроксимации

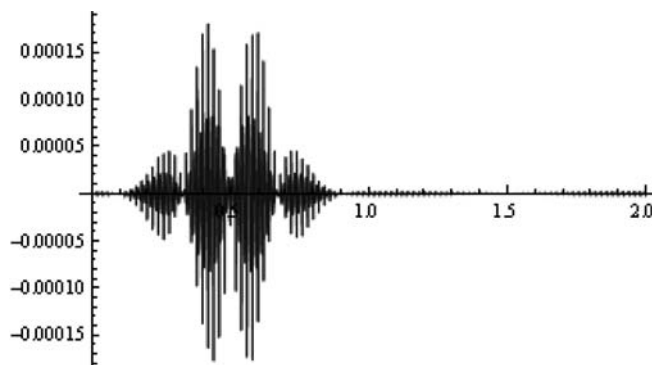


Рис. 8. Функция ошибки кусочно-параболической аппроксимации

**Пример 3. Решение дифференциального уравнения дробного порядка с производной по Капуто методом  $\mathcal{S}$ -преобразования с локальными полиномами Лежандра.**

Исходные данные:

- уравнение:  ${}^C D^{0,7}(y(t)) + 2y(t) = f$ ,  $y(0) = 2$ ,  
 $f = 4 + 1,114t^{0,3} + 2t - 5,1427t^{1,3} - 6t^2$ ;
- диапазон изменения аргумента  $0 \leq t < T = 1$ ;
- порядок базисной системы функций  $m = 50$ ;
- шаг дискретизации аргумента  $h = 0,02$ ;
- точное решение уравнения:  $y(t) = 2 + t - 3t^2$ .

Перед применением  $\mathcal{S}$ -преобразования введем

замену переменных:  $\frac{dy(t)}{dt} = u(t)$ .

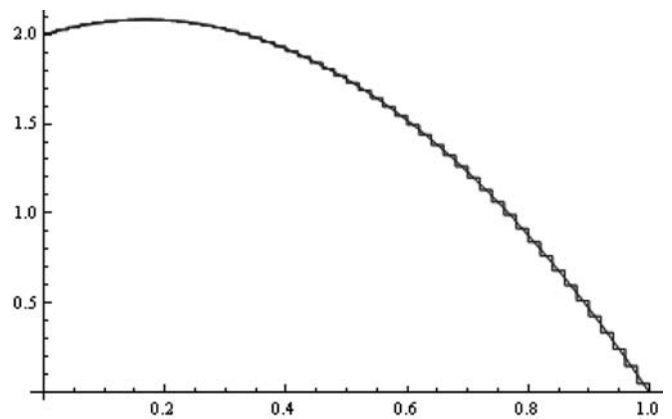
Используя определение дробной производной по Капуто, получим следующее интегральное уравнение:

$${}^{RL} J^{0,3}(u(t)) + 2 \int_0^t u(\tau) d\tau = f(t) - 2y(0).$$

Применяя к нему  $\mathcal{S}$ -преобразование, получим:  $\mathbf{P}^{(0,3)} \cdot \mathbf{U} + 2\mathbf{P}^{(1,0)} \cdot \mathbf{U} = \mathbf{F} - 2y(0) \cdot \mathbf{1}$ .

**Программа 3**

- Задание исходных данных:  
 $m := 50$ ;  $h := 0,02$ ;  $T := 1$ ;  $y_0 = 2$ ;  
 $f[t\_ ] := 4 + 1,114t^{0,3} + 2t - 5,1427t^{1,3} - 6t^2$ ;
- Определение базисной системы функций:  
 $v[t\_ , h\_ , i\_ ] := \text{If}[(i - 1) \cdot h \leq t < i \cdot h, 1, 0]$ ;  
 $S_v = \text{Table}[v[t, h, i], \{i, m\}]$ ;
- Определение операционных матриц интегрирования и задание необходимых для интегрального уравнения порядков (0,3 и 1,0):  
 $H_{11}[\beta\_ , h\_ , m\_ ] := \frac{h^\beta}{\Gamma[\beta + 2]} * \text{Table}[\text{Which}[i < j, 0, i == j, 1, i > j, (i - j + 1)^{\beta + 1} - 2(i - j)^{\beta + 1} + (i - j - 1)^{\beta + 1}], \{i, m\}, \{j, m\}]$ ;  
 $P_{03} = H_{11}[0.3, h, m]$ ;  $P_1 = H_{11}[1, h, m]$ ;
- Определение изображений правой части исходного дифференциального уравнения и константы  $\mathbf{1}$ :  
 $F = \text{Table}[\frac{1}{h} * \int_{(i-1) \cdot h}^{i \cdot h} f[t] dt, \{i, m\}]$ ;  
 $\text{One} := \text{Table}[1, \{i, m\}]$ ;
- Решение задачи в области  $\mathcal{S}$ -преобразования:  
 $\mathbf{U} = \text{Inverse}[\mathbf{P}_{03} + 2\mathbf{P}_1] \cdot (\mathbf{F} - 2y_0 * \text{One})$ ;  
 $\mathbf{Y} = y_0 * \text{One} + \mathbf{P}_1 \cdot \mathbf{U}$ ;



**Рис. 9. Точное и аппроксимационное решения дифференциального уравнения дробного порядка**

- Переход в область оригиналов (временного аргумента):  
 $y_a = Y \cdot S_v$ ;
- Визуализация точного и аппроксимационного решений уравнения (рис. 9):  
 $y[t\_ ] := 2 + t - 3t^2$ ;  $y_e = y[t]$ ;  
 $\text{Plot}[\{y_e, y_a\}, \{t, 0, T\}, \text{PlotPoints} \rightarrow 250]$

**Заключение.** Рассмотренные методы аппроксимации сигналов и алгебраизации математических моделей динамических систем могут быть использованы при исследовании нестационарных процессов в объектах с дробной динамикой и при обработке сигналов при проведении экспериментальных исследований. Компьютерная реализация методов допускает использование различных систем компьютерной алгебры, таких как MATLAB, MAPPLE и др.

**Список литературы**

1. **Васильев В. В., Симак Л. А.** Дробное исчисление и аппроксимационные методы в моделировании динамических систем. Киев: НАН Украины, 2008. 256 с.
2. **Васильев А. В.** Оценивание параметров и дискретизация непрерывных сигналов на основе полиномиальной аппроксимации в локальных базисах // Электроніка та системи управління. 2011. № 1 (27). С. 14–19.
3. **Гончаров В. Л.** Теория интерполирования и приближения функций. М.: ГИТТЛ, 1954. 328 с.
4. **Конторович М. И.** Операционное исчисление и нестационарные явления в электрических цепях. М.: ГИТТЛ, 1967. 568 с.
5. **Потапов А. А.** Фракталы и хаос как основа прорывных технологий в современных радиосистемах // Дополнение к книге Кроновер Р. Фракталы и хаос в динамических системах. М.: Техносфера, 2006. С. 374–484.
6. **Пухов Г. Е.** Комплексное исчисление и его применение. Киев: Изд-во АН УССР, 1961. 230 с.
7. **Пухов Г. Е.** Дифференциальные преобразования функций и уравнений. Киев: Наукова думка, 1980. 420 с.
8. **Учайкин В. В.** Метод дробных производных. Ульяновск: Артишок, 2008. 510 с.
9. **Oldham K. B., Spanier O.** The Fractional Calculus. — New York & London: Academic Press, 1974. 234 p.
10. **Wolfram Stephen:** The Mathematica book. — Wolfram Media / Cambridge University Press, 1996. 1403 p.

**ЖУРНАЛ В ЖУРНАЛЕ**



**НЕЙРОСЕТЕВЫЕ  
ТЕХНОЛОГИИ**

**№ 3**

**МАРТ**

**2012**

**Главный редактор:**

ГАЛУШКИН А.И.

**Редакционная коллегия:**

АВЕДЬЯН Э.Д.  
БАЗИАН Б.Х.  
БЕНЕВОЛЕНСКИЙ С.Б.  
БОРИСОВ В.В.  
ГОРБАЧЕНКО В.И.  
ЖДАНОВ А.А.  
ЗЕФИРОВ Н.С.  
ЗОЗУЛЯ Ю.И.  
КРИЖИЖАНОВСКИЙ Б.В.  
КУДРЯВЦЕВ В.Б.  
КУЛИК С.Д.  
КУРАВСКИЙ Л.С.  
РЕДЬКО В.Г.  
РУДИНСКИЙ А.В.  
СИМОРОВ С.Н.  
ФЕДУЛОВ А.С.  
ЧЕРВЯКОВ Н.И.

**Иностранные  
члены редколлегии:**

БОЯНОВ К.  
ВЕЛИЧКОВСКИЙ Б. М.  
ГРАБАРЧУК В.  
РУТКОВСКИЙ Л.

**Редакция:**

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

**Добротворский А. С.**

Оценка перспектив использования ассоциативных инструментальных платформ квантовых компьютеров, работающих в многозначном алфавите . . . . . 62

**Соломаха А. А., Милова К. А., Горбаченко В. И.**

Нейросетевая экспертная медико-технологическая система прогнозирования риска развития осложнений у больных с гнойно-деструктивными заболеваниями легких . . . . . 66

**Девятисильный А. С., Гриняк В. М.**

Нейросетевой алгоритм сопровождения судов в системах управления движением . . . . . 70

**Петрушан М. В., Самарин А. И.**

Метод контрастирования описаний изображений лиц для системы санкционированного доступа . . . . . 74

**А. С. Добротворский**, системный программист,  
e-mail: lexeich@nxt.ru  
Институт поведения —  
автономная некоммерческая организация  
Экспертный центр Государственной Думы  
и Федерального Собрания  
Российской Федерации по безопасности

## Оценка перспектив использования ассоциативных инструментальных платформ квантовых компьютеров, работающих в многозначном алфавите

*Рассмотрены возможности создания аппаратных эмуляторов для инструментальных платформ квантовых компьютеров, а также возможные пути повышения их производительности. Показано, что в первых поколениях квантовых вычислителей двоичная система счисления сохранится как наименее прецизионная с точки зрения систем идентификации результирующих состояний квантовых регистров и как наименее аппаратоемкая с позиций создания ассоциативных инструментальных платформ на основе традиционных вычислителей.*

**Ключевые слова:** квантовые компьютеры, ассоциативная память, эмулятор

### Введение

Основатели квантовых вычислителей [1—3] автоматически выбрали двоичную систему счисления, которая традиционна для "не квантовых" вычислителей с элементной базой, бистабильность которой поддерживается не на схемотехническом, а на физическом уровне. Сама по себе бистабильность в квантовых вычислителях не исключается, особенно если используемые физические эффекты основаны на модификациях спинов. Тем не менее, квантовые системы мультстабильны в своей основе, что позволяет использовать многозначный алфавит, который снижает длину квантового регистра, но повышает прецизионность измерительных систем, идентифицирующих его состояние. Но кроме прецизионности физических механизмов идентификации результирующих состояний квантовых регистров, работающих в  $k$ -значном алфавите, встают проблемы имитации квантовых вычислений на основе ассоциативной памяти традиционных компьютеров, используемых в инструментальных платформах квантовых компьютеров [4].

**Цель статьи** — оценить возможности использования ассоциативных инструментальных платформ квантовых компьютеров, работающих в многозначном алфавите.

### Постановка задачи

Инструментальная платформа для создания программ квантовых компьютеров должна обладать развитыми средствами отладки и тестирования. В частности, она должна включать в себя эмулятор, который позволил бы тестировать работу программы при отсутствии реального оборудования. Чтобы сэкономить память инструментального компьютера одной из традиционных архитектур и обеспечить приемлемое быстродействие эмулятора, желательно, чтобы последний был реализован аппаратно. Еще одной чертой эмулятора является то, что он должен выполнять моделирование квантовых вычислений по возможности "естественным" образом, т. е. имитируя суперпозицию состояний квантового регистра.

Покажем возможность создания и использования различимой суперпозиции состояний на примере схемы бит-поточкового [5, 6] суммирования, которая работает с данными в *символьном формате* и теоретически обеспечивает любую точность вычислений [7] (аналог обычного сложения столбиком).

Табл. 1 задает описание *суммирующей* машины Тьюринга, где:

- "единица переноса" от предыдущего разряда  $e^-$  однозначно идентифицирует предшествующее состояние  $q^-$  машины Тьюринга, а "единица переноса" в следующий разряд  $e^+$  — ее текущее состояние  $q^+$  по правилу  $q: = q_0$ , если  $e: = "0"$  и  $q: = q_1$ , если  $e: = "1"$ ;
- входное воздействие представлено не двумя, а тройкой символов, образующих триплет " $q^-, x_i, x_j$ ";
- действие однозначно представлено его результатом, т. е. *подставляемым* символом " $\Sigma_i$ ", а смещение и только влево по индексу  $i$  в преобразуемых операндах выполняется автоматически после перехода в новое состояние и контролируется условием останова.

Таблица 1

**Таблица машины Тьюринга  
при сложении посимвольной подстановкой**

$q^-$	" $x_i, x_j$ "	$\Sigma_i$	$q^+$	$q^-$	" $x_i, x_j$ "	$\Sigma_i$	$q^+$
$q_0$	0,0	0	$q_0$	$q_1$	0,0	1	$q_0$
$q_0$	0,1	1	$q_0$	$q_1$	0,1	0	$q_1$
$q_0$	1,0	1	$q_0$	$q_1$	1,0	0	$q_1$
$q_0$	1,1	0	$q_1$	$q_1$	1,1	1	$q_1$

Для упрощения описания работы ассоциативной машины Тьюринга ограничимся конкретным примером сложения двоичных чисел, заданных в символьном формате:  $100 + 110 = 1010$ . Тогда само сложение можно выполнить по следующему (микро)программному алгоритму.

**Шаг 1.** Присвоить индексу  $i = \overline{1, n}$  значение, отвечающее первому члену натурального ряда " $i$ " := "1", и выделить в слагаемых первые справа символы " $x_1^1$ " := "0" и " $x_1^2$ " := "0", где нижний индекс  $i$  соответствует положению суммируемых символов в упорядоченной последовательности (номер разряда). В таблице подстановки, отвечающей правилу суммирования (табл. 1), выбрать по содержанию первого триплета " $q_0, x_1^1, x_1^2$ " символы "0" и " $q_0$ ", где символ "0" соответствует значению первого разряда "суммы" (" $\Sigma_1$ " := "0"), а символ " $q_1$ " — значению "единицы переноса" (" $e^+$ " := "0") во второй разряд. Если " $i$ " — не последний член натурального ряда, задающий разрядную сетку преобразуемых операндов (" $i < n$ "), то выполнить шаг 2. В противном случае (при " $i \sim n$ ") перейти к шагу 3. Здесь символами :=, < и  $\sim$  обозначены соответственно оператор присваивания, отношение "больше—меньше" и "эквивалентность".

**Шаг 2.** Присвоить индексу " $i$ " значение следующего члена натурального ряда " $i$ " := " $i + 1$ ". (Здесь  $i + 1$  — не действие, а условное обозначение непосредственно следующего члена натурального ряда, стоящего за  $i$ ). Выделить в слагаемых символы из " $i + 1$ "-го разряда: " $x_2^1$ " := "0" и " $x_2^2$ " := "1". С учетом значения символа "единица переноса" от предыдущего разряда (" $e^-$ " := "1") в таблице подстановки по содержанию триплета " $q_1, x_2^1, x_2^2$ " выбрать символы "1" и " $q_0$ ", где символ "1" соответствует значению второго разряда "суммы" (" $\Sigma_2$ " := "1"), а символ " $q_0$ " — значению символа "единица переноса" (" $e^+$ " := "0") в разряд " $i + 1 + 1$ ". Если символ " $i + 1$ " — не последний член натурального ряда (" $i + 1 < n$ "), то повторить шаг 2. В противном случае (при " $i + 1 \sim n$ ") выполнить шаг 3.

В нашем примере повторяем шаг 2, т. е. " $i + 1$ " := " $i + 1 + 1$ ". В выделенном разряде слагаемых расположены символы " $x_3^1$ " := "1" и " $x_3^2$ " := "1", которым с учетом значения символа "единица переноса" от предыдущего разряда " $e^-$ " := "0" в таблице подстановки соответствуют триплет " $q_0, x_3^1, x_3^2$ " и символы

подстановки: " $\Sigma_3$ " := "0" и " $q_1$ ". При этом выполняется условие " $i + 1 + 1 \sim n$ ", что приводит к шагу 3.

**Шаг 3.** Подставить в следующий разряд суммы значение символа "единица переноса". Конец алгоритма ( $v = stop$ ) и сумма представлена упорядоченной последовательностью символов  $\Sigma$ : = "1", "0", "1", "0" (младший символ справа).

Простейшей аппаратной реализацией этого алгоритма может служить бит-поточковый ассоциативный символьный сумматор, выполненный по схеме рис. 1 [7].

Он включает в себя [7]:

- блок PD-ассоциативной памяти аргументов  $\{(x_i, x_j)\}$ , представленных в символьном виде: "1 десятичный символ — 1 полубайт", который содержит все возможные пары чисел, которые могут быть просуммированы;
- блок анализа содержимого ячеек PD-ассоциативной памяти аргументов, который формирует признак совпадения  $B_8$ : = 11111111 элемента памяти аргументов и пары входных операндов  $\{(x_i, x_j)\}$ ;
- блок памяти результатов сложения  $\Sigma_i$ , представленных в 8-битном символьном виде, в 5-м бите которого содержится значение "единица переноса" в следующий разряд  $e^+$ , причем на выход блока поступают одновременно, но по разным пространственным каналам, оба результата сложения как  $(\Sigma_i, e^+)/e^- = 0$ , так и  $(\Sigma_i, e^+)/e^- = 1$ ;
- блок выбора одной из "неоднозначных" реакций:  $\Sigma_i := (\Sigma_i/e^- = 0)$  или  $\Sigma_i := (\Sigma_i/e^- = 1)$ .

Главная особенность работы бит-поточкового ассоциативного символьного сумматора состоит в том, что на вход поступают пары символов аргументов  $\{(x_i, x_j)\}$ , а для однозначного выбора из таблицы машины Тьюринга требуется еще и единица переноса от предыдущего разряда  $e^-$ .

Отсутствие значения единицы переноса делает реакцию ассоциативной памяти неоднозначной:  $(\Sigma_i, e^+)/e^- = 0$  и  $(\Sigma_i, e^+)/e^- = 1$ . В результате приходится работать не с табл. 1, а с табл. 2, в которой число строк в 2 раза меньше, а выходная реакция неоднозначна, и избавиться от нее можно только на выходе ассоциативной памяти с помощью управляемого элемента задержки, изменяющего свое состояние  $q^- \rightarrow q^+$  адекватно  $e^+ \rightarrow e^-$ .



Рис. 1. Структурно-функциональная схема бит-поточкового ассоциативного символьного сумматора

Таблица 2

Таблица PD-ассоциативной машины Тьюринга при сложении посимвольной подстановкой

" $x_i, x_j$ "	$(\Sigma_i, q^+)/q^- = q_0$	$(\Sigma_i, q^+)/q^- = q_1$
0,0	0,0	1,0
0,1	1,0	0,1
1,0	1,0	0,1
1,1	0,1	1,1

Таким образом, как и в "кубите", приходится возбуждать "одновременно" две ячейки памяти, которые хранят два значения символа суммы  $\Sigma_i$ , отвечающие двум значениям признака "единица переноса" из предыдущего разряда:  $\Sigma_i/e^- = 0$  и  $\Sigma_i/e^- = 1$ .

В такой реализации неоднозначный отклик — вынужденная мера, так как "предыдущие разряды" разнесены в пространстве согласно правилу упорядочения строк в PD-ассоциативной памяти аргументов, а содержимое признака "единица переноса" должно учитываться в следующем разряде (цикле символьного сложения). Устранить такую неоднозначность можно только на выходе каналов ассоциативной памяти значений сумм  $\Sigma_i$ , где все результаты строго упорядочены по индексу разряда суммы. При этом следует иметь в виду, что в отличие от кубита, обе реакции PD-ассоциативной памяти не смешаны в линейную суперпозицию, а разнесены в пространстве. Благодаря этому удастся выделить и запомнить на два такта "единицу переноса" из предыдущего разряда  $e^-$  и в зависимости от ее содержимого выбрать верный результат при сложении следующей пары аргументов.

Структура схемы символьного суммирования с использованием ассоциативной памяти дает возможность ускорения вычислений. Первый путь увеличения производительности — это увеличение "значности" алфавита представления слагаемых. В частности, переход от двоичных к десятичным или шестнадцатеричным числам снижает разрядность слагаемых, а значит и число тактов, за которые будет выполнено суммирование. Второй путь повышения скорости — это работа одновременно с несколькими разрядами чисел. Имеется возможность аппаратно реализовать суммирование одновременно, например по два разряда каждого числа, что должно увеличить скорость вдвое. Однако каждый из путей оптимизации скорости ведет к увеличению размеров таблицы машины Тьюринга и необходимой памяти для реализации схемы ассоциативного суммирования.

### Оценка временных и аппаратных затрат на ассоциативную имитацию квантовых вычислений

Для оценки временных и аппаратных затрат рассмотрим процесс сложения 64-разрядных двоичных чисел, которые содержат максимум 20 разрядов при представлении в десятичной системе счисления и 16 разрядов в шестнадцатеричном представлении.

Скорость вычислений (операций в секунду) на аппаратном уровне определяется тактовой частотой работы ассоциативного арифметико-логического устройства, а также числом тактов, затрачиваемым на выполнение одной операции. В случае последовательных вычислений на базе рассмотренной выше схемы скорость вычислений определяется отношением тактовой частоты к числу тактов, затрачиваемых на выполнение одной операции: 64 такта для двоичных чисел, 20 тактов для десятичного представления тех же чисел и 16 тактов для шестнадцатеричной системы.

Поэтому прирост скорости (без учета дополнительных издержек на реализацию) при переходе от двоичной к десятичной системе составит по максимуму  $64/20 = 3,2$  раза, а при переходе от двоичной к шестнадцатеричной —  $64/16 = 4$  раза. Однако количество требуемых аппаратных ресурсов растет еще быстрее. Из данных табл. 2 видно, что для реализации схемы ассоциативного суммирования двоичных чисел требуется задать на аппаратном уровне четыре строки ассоциативной памяти, в то время как для десятичных чисел требуется уже 100 строк, а для шестнадцатеричных — 256. В итоге при переходе от двоичной системы к десятичной аппаратные затраты возрастают в  $100/4 = 25$  раз, а при переходе к шестнадцатеричной системе — в  $256/4 = 64$  раза.

Второй путь повышения производительности — это одновременная ассоциативная обработка в таблице Тьюринга нескольких и, в частности, двух разрядов. В идеале это должно повысить быстродействие схемы вдвое для каждой системы счисления, так как число тактов, необходимых для выполнения операции, сократится в половину. Однако чтобы хранить в ассоциативной памяти все возможные комбинации пар символов и отвечающих им результатов сложения, потребуется уже  $k^4$  строк таблицы, где  $k$  — основание системы счисления.

Таким образом, в случае суммирования одновременно двух разрядов входных операндов при переходе от двоичной системы счисления к десятичной аппаратные затраты вырастут в  $10^4/2^4 = 625$  раз, при переходе к шестнадцатеричной — в  $16^4/2^4 = 4096$  раз.

Поэтому платой за повышение скорости в 2 раза за счет ассоциативного суммирования дуплетов исходных операндов становится 4-кратное увеличение аппаратуры для двоичной системы счисления,



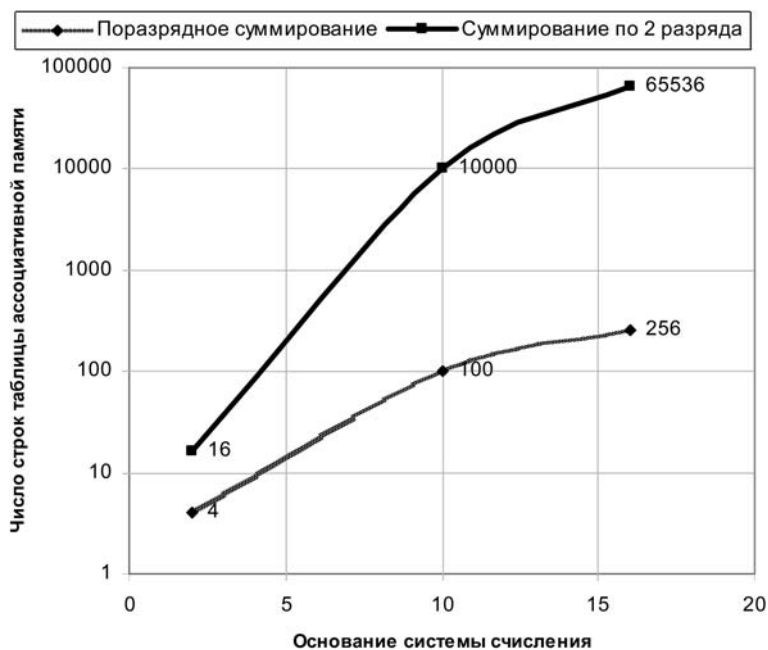


Рис. 2. Увеличение аппаратных затрат при увеличении системы счисления для случаев поразрядного суммирования и суммирования одновременно по 2 разряда

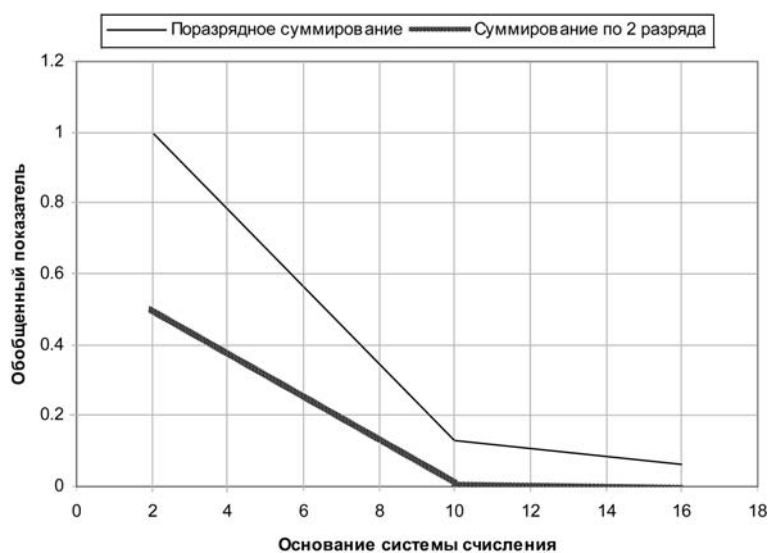


Рис. 3. Зависимость обобщенного показателя от системы счисления и типа операции

100-кратное — для десятичной и 256-кратное — для шестнадцатеричной системы счисления (рис. 2).

Если воспользоваться обобщенным показателем оценки  $\eta = \delta t / \delta l$  (рис. 3), то видна устойчивая тенденция падения КПД использования аппаратуры. Здесь  $\delta t$  — относительный выигрыш во времени

в разгах, а  $\delta l$  — относительный проигрыш в памяти в разгах.

Нужно отметить, что все приведенные выше расчеты справедливы для случая, когда работа ассоциативного арифметико-логического устройства с операндами в десятичной и шестнадцатеричной системе идет действительно в этой форме. Если ассоциативный сумматор работает в режиме интерпретации с двоичным представлением каждого символа (в виде двоичного кода), то в этом случае прироста скорости вообще не будет наблюдаться ни в случае изменения системы счисления, ни в случае обработки одновременно нескольких разрядов. Требования к аппаратуре, напротив, будут расти аналогично описанным выше данным.

## Выводы

По крайней мере, в первых поколениях квантовых вычислителей двоичная система счисления сохранится как наименее прецизионная с точки зрения систем идентификации результирующих состояний квантовых регистров и как наименее аппаратоемкая с позиций создания ассоциативных инструментальных платформ на основе традиционных вычислителей.

## Список литературы

1. Дойч Д. Квантовая теория принципа Черча—Тьюринга и универсальный квантовый компьютер: Пер. с англ. "Квантовые компьютеры & квантовые вычисления II" / Под ред. В. А. Садовниченко // Регулярная и хаотичная динамика. 1999.
2. Валиев К. А., Кокин А. А. Квантовые компьютеры: надежды и реальность // Регулярная и хаотичная динамика. 2001.
3. Фейнман Р. Моделирование физики на компьютерах: Пер. с англ. Квантовые компьютеры & квантовые вычисления II / Под ред. В. А. Садовниченко // Регулярная и хаотичная динамика. 1999.
4. Алакоз Г. М., Добротворский А. С. Бит-потокное ассоциативное суммирование с разделяемой пространственно-временной реакцией // Нейрокомпьютеры: разработка и применение. 2011. № 2.
5. Алакоз Г. М., Вольперт Л. А., Страутманис Г. Ф. Принципы MIMD-бит-потокной организации вычислений // Автоматика и вычислительная техника. 1997. № 2.
6. Алакоз Г. М., Попов А. А. MIMD-бит-потокные технологии для отказоустойчивых сверхпараллельных субпроцессорных трактов бортовых вычислительных систем // Авиакосмическое приборостроение. 2007. № 10.
7. Алакоз Г. М., Светлов Р. В. Машина Тьюринга как ассоциативная нейро-ЭВМ // Нейрокомпьютер: разработка и применение. 2009. № 5.

**А. А. Соломаха**, канд. мед. наук, доц.,  
Пензенский государственный университет,  
e-mail: solomakha7@mail.ru

**К. А. Милова**, ст. преподаватель,  
Пензенский государственный педагогический  
университет им. В. Г. Белинского,

**В. И. Горбаченко**, д-р техн. наук, проф.,  
Пензенский государственный педагогический  
университет им. В. Г. Белинского

## **Нейросетевая экспертная медико-технологическая система прогнозирования риска развития осложнений у больных с гнойно-деструктивными заболеваниями легких**

*Цель исследования — создание нейросетевой экспертной медико-технологической системы прогнозирования риска развития хирургических осложнений у больных с гнойно-деструктивными заболеваниями легких. Для достижения поставленной цели были проведены статистический анализ исходных данных, выбор критериев для построения модели, а также эксперименты по подбору типа и архитектуры нейронной сети для осуществления прогноза. Было применено квантование входных данных и их кодирование для улучшения качества работы нейронной сети. В результате экспериментов была построена модель, с помощью которой осуществляется прогнозирование риска развития хирургических осложнений с достоверностью 90 %. Поэтому следить за состоянием больных с гнойно-деструктивными заболеваниями легких целесообразно с использованием нейросетевой медико-технологической системы.*

**Ключевые слова:** нейросетевое прогнозирование, кровопотеря, гнойно-воспалительные осложнения, нейронная сеть, моделирование, управление, лечебный процесс, торакотомия

### **Введение**

Последнее десятилетие в России и за ее пределами характеризуется неуклонным интересом исследователей к изучению нейросетевых медицинских технологий с использованием различных технических и программных средств для решения клинических задач [3, 7].

При изучении доступных публикаций отечественных и зарубежных исследователей, а также ресурсов медицинских сайтов сети Интернет, нами не обнаружено работ, освещающих проблему прогнозирования риска развития послеоперационных осложнений у пациентов с гнойно-деструктивными заболеваниями легких после торакотомии, на

основе нейросетевого метода. Современная клиническая хирургия активно применяет математические методы для диагностики и прогнозирования состояния больных [2, 8, 9]. Это сегодня мотивируется целесообразностью получения новых знаний на основе традиционной медицинской информации: медико-статистических показателей деятельности хирургических, терапевтических отделений и вспомогательных служб лечебно-профилактических учреждений (ЛПУ), а также клинических и лабораторных данных, полученных при обследовании больных, для управления лечебным процессом в новых экономических условиях [6].

Целью исследования являлось создание нейросетевой экспертной медико-технологической системы прогнозирования риска развития хирургических осложнений у больных с гнойно-деструктивными заболеваниями легких. Для достижения поставленной цели решались следующие задачи: определение объективных критериев оценки тяжести состояния хирургических больных с гнойно-деструктивными заболеваниями легких, влияющих на риск развития хирургических осложнений; выбор архитектуры нейронной сети на основе эксперимента для создания медико-технологической нейросетевой системы прогнозирования риска развития осложнений у данной категории больных. Исследование выполнялось в отделении торакальной хирургии Пензенской областной клинической больницы им. Н. Н. Бурденко, кафедре "Хирургия" Пензенского государственного университета, кафедре "Информатика и вычислительные системы" Пензенского государственного педагогического университета им. В. Г. Белинского. Для экспериментов с архитектурой сети использовался пакет MATLAB.

### **Формальная постановка задачи**

Цель исследования — построить модель, позволяющую осуществить прогноз осложнений на основе сведений, содержащихся в историях болезни. Необходимо найти целевую зависимость, выражаемую отображением  $m: X \rightarrow O$ , где  $X$  — некоторое множество входных параметров, а  $O = \{0; 1\}$  — множество возможных исходов: 0 — отсутствие осложнения; 1 — наличие осложнения. Значения данной целевой зависимости известны только на множестве объектов конечной выборки:

$$S^m = \{(x_1^1, x_2^1, \dots, x_n^1), \\ (x_1^2, x_2^2, \dots, x_n^2), \dots, (x_1^m, x_2^m, \dots, x_n^m)\},$$

где  $S^m$  — выборочное множество;  $x_1^i, x_2^i, \dots, x_n^i$  — значения факторных переменных (анализы крови и другие медицинские показатели) для  $i$ -го пациента;  $n$  — число отобранных факторных признаков.

Из множества признаков  $X$  необходимо выделить подмножество  $X_1$  признаков, которые оказывают существенное влияние на результирующую переменную  $O$ . Исходя из имеющихся в историях болезни сведений на первом этапе были предложены четыре модели "Все показатели", "Общий анализ крови (ОАК)", "Белок", "Спирография".

Модель "Все показатели" использует показатели, по которым имеется достаточное количество наблюдений, а именно: длительность операции; возраст; данные спирографии; микробиология; уровень гемоглобина; лейкоциты; скорость оседания эритроцитов (СОЭ); общий билирубин; общий белок; мочевины; глюкоза.

Модель "ОАК" в качестве факторных признаков использует следующие показатели: длительность операции, возраст, данные спирографии, уровень гемоглобина; лейкоциты и СОЭ.

Модель "Спирография" построена на следующих признаках: длительность операции; возраст; данные спирографии; микробиология (модель, не содержащая данных об анализе крови).

Для модели "Белок" использовались такие признаки: длительность операции; возраст; данные спирографии; уровень гемоглобина; общий белок.

### Регрессионный анализ

Поскольку указанные выше наборы факторных признаков были предложены эмпирически, с целью выявления формальной зависимости между факторными переменными и результирующей переменной (наличие осложнений) в первую очередь был проведен множественный регрессионный анализ. Оказалось, что только в моделях "ОАК" и "Все показатели" факторные переменные оказывают влияние на результирующую переменную (наличие осложнения).

Поскольку множественная регрессия позволяет выявить лишь наличие или отсутствие множественной линейной зависимости, был проведен формальный отбор признаков с помощью инструмента пошаговой регрессии. Как известно, наиболее приемлемым способом отбора факторных признаков является шаговая регрессия (шаговый регрессионный анализ). Сущность метода шаговой регрессии заключается в последовательном включении факторов в уравнение регрессии и последующей проверке их значимости [1].

В результате применения процедуры шаговой регрессии было выяснено, что наибольшее влияние на результирующую переменную (наличие или отсутствие осложнения) оказывают две факторные переменные — продолжительность операции и СОЭ [4].

### Эксперименты с нейронной сетью

Поскольку регрессия выявляет лишь наличие или отсутствие множественной линейной зависимости, после формального отбора признаков были проведены эксперименты с нейронной сетью, по-

зволяющей выявить сложные нелинейные зависимости между переменными.

Была построена нейросетевая модель, использующая указанные факторные переменные для определения значения результирующей переменной. В модели использовалась многослойная сеть сигмоидального типа. Эксперименты по подбору оптимальной архитектуры сети проводились на двух- и трехслойной нейронной сети. Число нейронов в выходном слое определялось условиями задачи и равнялось одному. В результате экспериментов было выяснено, что на нескольких архитектурах достигается примерно одинаковый результат, однако наименьшая ошибка составила около 89 %. Это сеть, имеющая в первом слое 13 нейронов, во втором — 24 нейрона и в выходном слое 1 нейрон. Важно отметить, что ошибка нейросети второго рода равна 0, когда осложненный случай распознается как неосложненный.

### Использование дополнительных качественных параметров

Кроме формально установленных показателей на основе практического опыта врачей-экспертов, медико-статистического анализа осложнений риска развития несостоятельности ушитой культи бронха и гнойно-воспалительных осложнений нами были определены следующие факторы, позволяющие прогнозировать риск развития гнойно-воспалительных осложнений в торакальной хирургии: объем и длительность оперативного вмешательства; эндотрахеальный наркоз как фактор риска развития осложненной трахеобронхиальной дерева; гемоплазмотрансфузия и инфузия кровезаменителей, отражающие травматизм оперативного вмешательства; гистологические свойства ткани удаленного бронха по линии резекции; характер внутрибольничной инфекции; СОЭ, а также клиническая оценка тяжести состояния хирургического больного с гнойно-деструктивным заболеванием легких.

Аналитическое исследование медико-статистических клиничко-лабораторных данных историй болезни торакальных больных позволило установить, что информация о внутрибольничной инфекции, гистологических свойствах бронха по линии его резекции, сведения об особенностях проведения эндотрахеального наркоза не явились значимыми при проведении нейросетевого прогнозирования у больных с гнойно-деструктивными заболеваниями легких. Внутрибольничная инфекция сопровождает течение послеоперационного периода подавляющего большинства торакальных хирургических больных, поэтому данные о ней не являются ключевыми в нейросетевом прогнозировании. Гистологические свойства ткани удаленного бронха по линии его резекции не позволяют утверждать, в связи с отсутствием специальных морфологических данных, об их роли в развитии несостоятельности и гнойно-вос-

палительных осложнений у данной категории больных. Эндотрахеальный наркоз больным с гнойно-деструктивными заболеваниями легких в областной клинической больнице выполняется специально подготовленным врачом-анестезиологом, который имеет высшую квалификационную категорию и стаж практической работы свыше 25 лет. Поэтому осложнений наркоза в изучаемой когорте больных не было.

Таким образом, дополнительными входными параметрами для нейросетевого прогнозирования стали: объем оперативного вмешательства, гемоплазмотрансфузии и инфузии кровезаменителей, а также клиническая оценка тяжести состояния больного. Важно отметить, что нами специально в целях вовлечения и активного участия врача в процессе и результатах нейросетевого моделирования, введен параметр "клиническая оценка тяжести состояния больного". Поэтому разработанная нейросетевая медико-технологическая система, по своей сущности, относится к экспертным.

### Квантование данных

Одним из известных преобразований, используемых при подготовке данных к нейросетевому анализу, является квантование. Квантование подразумевает разбиение диапазона значений числового признака на заданное конечное число интервалов. Полученным интервалам присваиваются номера, именуемые уровнями квантования. Каждое значение заменяется номером интервала квантования, в который попало данное значение. Квантование позволяет представлять и хранить данные в более компактном и защищенном от искажений виде. Квантование используется для уменьшения числа разнообразных значений признака, что в некоторых случаях позволяет сделать работу модели более эффективной. Важным аспектом является выбор числа интервалов, поскольку в результате квантования осуществляется переход от точных данных к

некоторому набору интервалов, и при этом неизбежна потеря информации. В связи с тем, что два значения расположены в двух соседних интервалах, нет возможности точно определить, на сколько одно больше, чем другое. Следовательно, чем больше используется интервалов, тем точнее пред-

Таблица 1

#### Интерпретация значений квантования количественных медицинских данных

№	Наименование показателя	Интерпретация
1.	Длительность операции (общая сумма времени всех оперативных вмешательств)	1 — от 0, когда операция не выполнялась, до 100 мин; 2 — от 101 до 200 мин; 3 — от 201 до 300 мин; 4 — от 301 до 400 мин; 5 — более 400 мин
3.	Объем гемоплазмотрансфузий и кровезаменителей (общий объем перелитых сред больному)	1 — от 0 (не было трансфузий и инфузий) до 1 л; 2 — от 1 до 3 л; 3 — от 3 до 5 л; 4 — от 5 до 7 л; 5 — более 7 л
4.	СОЭ	1 — соответствует норме и не превышает 19 мм/ч; 2 — от 20 до 39 мм/ч; 3 — от 40 до 59 мм/ч; 4 — от 60 и более мм/ч
5.	Возраст больного	0 — от 1 до 30 лет включительно и старше 70 лет; 1 — от 31 до 40 лет; 2 — от 61 до 70 лет; 3 — от 41 до 60 лет
6.	Уровень гемоглобина	0 — соответствует норме; 1 — от 111 до 120 г/л; 2 — от 101 до 110 г/л; 3 — от 91 до 100 г/л; 4 — от 81 до 90 г/л; 5 — 70 до 80 г/л; 6 — менее 70 г/л
7.	Лейкоциты	1 — соответствует норме; 2 — от 9 до 10; 3 — от 11 до 15; 4 — от 16 до 20; 5 — более 20

Таблица 2

#### Интерпретация кодирования качественных медицинских показателей

№	Наименование показателя	Интерпретация
1.	Объем оперативного вмешательства	0 — оперативного вмешательства не было, выполнялось дренирование плевральной полости, торакоскопия или видеоторакоскопия; 1 — дренирование плевральной полости, торакотомия, резекция сегмента легкого или атипичная резекция сегмента легкого; 2 — дренирование плевральной полости, лоб-, билоб- или пневмонэктомия; 3 — торакотомия с декортикацией легкого, дренированием плевральной полости; 4 — выполнялось более трех оперативных вмешательств
2.	Спирография	0 — норма или не выполнялась ввиду недостаточной информативности; 1 — незначительное снижение вентилиционной функции; 2 — умеренное снижение; 3 — выраженное, соответствует 3-й степени
3.	Клиническая оценка состояния больного	1 — удовлетворительное состояние; 2 — состояние средней степени тяжести; 3 — состояние тяжелое; требуется подготовка к экстренной или отсроченной операции в условиях реанимации врачом-реаниматологом и торакальным хирургом

ставление исходных значений. В общем случае квантование упрощает алгоритм обучения нейронной сети [5].

При создании нейросетевой экспертной медико-технологической системы мы руководствовались стремлением к наиболее точному представлению исходных данных. В целях увеличения достоверности прогноза, выдаваемого нейронной сетью, а также в связи с включением в модель параметра "клиническая оценка тяжести состояния больного" было предложено использовать квантование входных данных.

Квантование первичных данных нами проведено для следующих показателей: общая сумма времени выполненных операций; общий объем перелитых трансфузионных сред и кровезаменителей; показатель СОЭ; возраст; гемоглобин; количество лейкоцитов (табл. 1).

Для качественных переменных (перечень операций, данные спирографии и клиническая оценка тяжести состояния торакального хирургического больного) была разработана система кодирования данных, представленная в табл. 2.

Закодированные и квантованные данные впоследствии использовались для экспериментов с нейронной сетью.

### Нейросетевые эксперименты с квантованными данными

После того, как было проведено квантование и кодирование данных по указанной схеме, полученные числа (номера интервалов и качественные оценки) были закодированы с помощью маски из двоичных цифр [4]. Затем с ними были проведены эксперименты с персептроном и нейронными сетями, использующими в качестве функции активации сигмоидальную функцию.

Персептрон (рис. 1, см. третью сторону обложки) представляет собой ставшую классической модель искусственного нейрона с пороговой функцией активации. Как известно, персептрон является простейшей нейронной сетью, которая, однако же, бывает эффективна для решения задач классификации в случае линейно-разделимых объектов [3].

Сигмоидальные сети используют в качестве функции активации различные сигмоидальные функции. В данном случае в качестве функции активации использовалась логарифмическая сигмоидная функция, поскольку ее диапазон чувствительности [0; 1].

Для эксперимента использовался как единственный сигмоидальный нейрон, так и двухслойная сигмоидальная нейронная сеть (рис. 2, 3, см. третью сторону обложки). В двухслойной сети число нейронов в выходном слое составляло 1 (определялось числом выходных переменных). Число нейронов в первом слое варьировалось.

Завершение экспериментов позволило проанализировать результаты и установить, что персепт-

рон позволяет прогнозировать риск развития хирургических осложнений с достоверностью до 75 %. Нейросеть, состоящая из одного нейрона с сигмоидальной функцией активации, логистической функцией, прогнозирует с достоверностью 85 %. Многослойная сигмоидальная нейросеть, состоящая из одного нейрона в первом слое и одного во втором, осуществляет прогноз с достоверностью 90 %. Дальнейшее усложнение структуры нейросети (увеличение числа нейронов в первом слое) не приводит к увеличению достоверности ее работы.

### Заключение

Таким образом, для решения поставленной цели были проведены статистический анализ исходных данных, выбор критериев для построения модели, а также эксперименты по подбору типа и архитектуры нейронной сети для осуществления прогноза. Было применено квантование входных данных и их кодирование для улучшения качества работы нейронной сети. В результате экспериментов была построена модель, осуществляющая прогнозирование риска развития хирургических осложнений с достоверностью 90 %. Поэтому управление состоянием больных с гнойно-деструктивными заболеваниями легких целесообразно осуществлять с помощью нейросетевой медико-технологической системы.

### Список литературы

1. Афффи А., Эйзен С. Статистический анализ: Подход с использованием ЭВМ: пер. с англ. М.: "Мир", 1982. 488 с.
2. Кислова Ю. В., Богомолов А. В., Солошенко Н. В. Математическое моделирование реакций сердечно-сосудистой системы человека при кровопотере и гипотермии // Медицинская техника. 2006. № 4. С. 10—13.
3. Лифшиц В. Б., Булдакова Т. И., Суятинов С. И., Колентьев С. В. Статистический и нейросетевой методы идентификации и прогнозирования в медицине // Информационные технологии. 2004. № 3. С. 60—63.
4. Милова К. А. Интеллектуальная система прогнозирования развития осложнений у хирургических больных // Нейрокомпьютеры. Разработка и применение. 2010. № 11. С. 20—23.
5. Паклин Н. Б., Орешкин В. И. Бизнес-аналитика: от данных к знаниям. М.: Питер, 2010. С. 146—148.
6. Соломаха А. А. Анализ медико-статистических показателей работы торакального отделения областной клинической больницы для выявления факторов риска развития хирургических осложнений после торакотомии // Вестник службы крови России. 2011. № 1. С. 22—26.
7. Lisboa P. J. G., El-Deredy W., Vellido A., Etchells T., Pountney D. C. Automatic Variable Selection and Rule Extraction Using Neural Networks // Proc. of the 15<sup>th</sup> IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics, Berlin. 1997. P. 461—466.
8. Schetinin V., Schult J. The Combined Technique for Detection of Artefacts in Clinical Electroencephalograms of Sleeping Newborns // IEEE Transaction on Information Technologies in Biomedicine. 2004. 8:1. 28—35. (ISI 1.54).
9. Van Calster B., Timmerman D., Lu C., Suykens J. A. K., Valentin L., Van Holsbeke C., Amant F., Vergote I., Van Huffel S. Preoperative diagnosis of ovarian tumors using Bayesian kernel-based methods // Ultrasound in Obstetrics and Gynecology. May 2007. Vol. 29, no. 5. P. 496—504.

**А. С. Девятисильный,**

д-р техн. наук, проф., гл. науч. сотр.,  
e-mail: devyatis@iacp.dvo.ru,

**В. М. Гриняк,** канд. техн. наук, ст. науч. сотр.,  
e-mail: Victor.Grinyak@vvsu.ru,

Институт автоматизации и процессов управления  
ДВО РАН, г. Владивосток

## Нейросетевой алгоритм сопровождения судов в системах управления движением

*Рассматривается задача сопровождения объекта в современных системах управления движением на море при обработке навигационных данных двухкоординатной радиолокационной станции кругового обзора. Обсуждается проблема срыва сопровождения малоразмерных маневрирующих судов. Предлагается нейросетевая интерпретация традиционных алгоритмов оптимальной фильтрации, повышающая устойчивость сопровождения и понижающая вероятность срыва.*

**Ключевые слова:** управление движением судов, сопровождение объекта, радиолокационные измерения, нейронная сеть, оценка траектории движения

### Введение

Современные системы управления движением судов (СУДС) представляют собой сложные в техническом и организационном отношении специализированные предприятия [1, 2]. Основной их задачей является обеспечение безопасности движения в районах интенсивного судоходства и стесненных условиях плавания (в проливах, в акваториях морских портов и т. п.).

Важнейшей составной частью СУДС, ее функциональным ядром, является подсистема сбора и обработки навигационной информации об объектах, находящихся в зоне ответственности СУДС. Первичной информационной базой этой подсистемы, несмотря на развитость и многообразие современных средств навигации, продолжают оставаться традиционные двухкоординатные радиолокационные станции (РЛС) кругового обзора. Вторичная обработка данных осуществляется с помощью специализированных программных комплексов, цель которых — предоставить такую информацию о навигационной обстановке, которая позволила бы обеспечить максимальный уровень безопасности движения [3, 4].

Проблема определения навигационных характеристик (координат и скоростей) судна с помощью

РЛС, сопряженной с ЭВМ, может быть дифференцирована на следующие подзадачи:

- получение радиолокационного образа навигационного пространства;
- преобразование радиолокационного образа с помощью аналого-цифровых устройств (АЦП) и "введение" его в память ЭВМ;
- обеспечение обнаружения, захвата и сопровождения судна;
- определение (оценка) координат судна и их производных.

Радиолокационный образ навигационного пространства представляется в памяти ЭВМ в виде двумерного массива  $a[m, n]$ , где  $m$  — число секторов по углу (линеек), формируемых за полный оборот антенны радара;  $n$  — число сегментов по дальности;  $a[i, j]$  — амплитуда отраженного эхо-сигнала в точке с азимутом  $i\Delta\varphi$  и с дальностью  $j\Delta r$  (здесь  $\Delta\varphi$  — разрешение радара по углу, соответствующее частоте посылок;  $\Delta r$  — разрешение по дальности, обусловленное частотой дискретизации при оцифровке отраженного сигнала). Массив обновляется по мере вращения антенны радара.

Для измерений координат движущегося объекта на протяжении всей его траектории необходима формализация исходной измерительной информации  $a[m, n]$  с помощью процедуры сопровождения. В обсуждаемом контексте сопровождением называется динамический процесс соотнесения условной точки, принятой за объект, с радиолокационным изображением объекта [5]. Инициацию этого процесса называют захватом объекта, а непроизвольное (аварийное) прекращение — срывом сопровождения (срывом захвата). С процессом сопровождения тесно связано понятие строба, а именно — с тем обстоятельством, что протяженный физический объект необходимо отождествить с точкой. Строб — область радиолокационного образа, используемая для преобразования в точку, отождествляемую затем с измеренными координатами объекта. В простейшем варианте стробирование представляет собой определение "центра масс" амплитуд отраженного сигнала в области строба; для описания линейных размеров этой области принят термин "ширина строба". Ширина строба зависит от размеров сопровождаемых объектов (она не должна быть меньше объекта), от характерного расстояния между объектами (в строб не должно попадать более одного объекта, т. е. он не может быть слишком большим), от маневренности объектов (ширина строба должна быть достаточной для устойчивого сопровождения без срывов).

Сопровождение в условиях насыщенных потоков маломерных судов, особенно интенсивных в летнюю навигацию, накладывает существенные ограничения на возможности выбора ширины строба.

Поэтому для такого класса судов актуальной становится разработка специальных алгоритмов сопровождения, сохраняющих устойчивость при сопровождении высокоманевренных объектов стробом небольшой ширины. В настоящей работе рассматривается модель задачи сопровождения, уменьшающая вероятность срыва и основанная на адаптации традиционных алгоритмов оптимальной фильтрации к модельным представлениям современной нейроинформатики.

### Модельные представления и постановка задачи

Рассмотрим следующую модель движения объекта:

$$\begin{aligned} x(k+1) &= x(k) + v_x(k)\tau + q_x(k); \\ y(k+1) &= y(k) + v_y(k)\tau + q_y(k), \end{aligned} \quad (1)$$

где  $k$  — идентификатор (порядковый номер) момента времени;  $x(k)$ ,  $y(k)$  — координаты объекта в момент времени  $t_k$ ;  $v_x(k)$ ,  $v_y(k)$  — компоненты вектора скорости объекта;  $q_x(k)$ ,  $q_y(k)$  — компоненты вектора случайных немоделируемых параметров движения;  $\tau$  — период оценивания, так что  $\tau = t_{k+1} - t_k$ .

Пусть измеряемыми параметрами являются декартовы координаты объекта. Тогда модель рассматриваемой задачи можно представить следующим дискретным матричным уравнением "состояние—измерение"

$$\begin{aligned} x_{k+1} &= \Phi x_k + q_k; \\ z_k &= Hx_k + r_k. \end{aligned} \quad (2)$$

Здесь  $x_k = (x(k), v_x(k), y(k), v_y(k))^T$  — вектор состояния объекта, включающий его координаты и их производные ( $\tau$  — символ транспонирования);  $q_k$  — вектор немоделируемых параметров;  $z_k$  — вектор измерений;  $r_k$  — вектор погрешностей измерений. С учетом модели (1) матричные коэффициенты  $\Phi$  и  $H$  равны соответственно

$$\Phi = \begin{bmatrix} 1 & \tau & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \tau \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Модель оценивания вектора состояния  $x_k$  по измерениям  $z_k$  может быть представлена следующим уравнением

$$\hat{x}_{k+1} = \Phi \hat{x}_k + K(z_{k+1} - H\hat{x}_{k+1}), \quad (3)$$

где  $\hat{x}_k$  — оценка вектора состояния;  $K$  — матричный коэффициент.

Известны несколько подходов к выбору матрицы  $K$ . Так, в практических приложениях популярен  $\alpha - \beta$ -алгоритм, основное достоинство которого — низкие требования к вычислительным ресурсам [6]. В этом алгоритме матрица  $K$  имеет вид

$$K = \begin{bmatrix} \alpha & 0 \\ \beta/\tau & 0 \\ 0 & \alpha \\ 0 & \beta/\tau \end{bmatrix}. \quad (4)$$

Сходимость алгоритма обеспечивается выполнением условия  $0 < \alpha \leq 1$ ,  $0 < \beta \leq 1$ . Коэффициенты  $\alpha$  и  $\beta$  выбираются исходя из требований чувствительности алгоритма к немоделируемым маневрам судна  $q_k$  и ошибкам измерений  $r_k$ . Показано [7], что соотношение коэффициентов  $\beta = \alpha^2/(2 - \alpha)$  является оптимальным.

Другой известный метод — алгоритм Калмана [8]. В нем матрица  $K$  выбирается таким образом, чтобы асимптотически обеспечить минимум функционала  $J = M[(x_k - \hat{x}_k)^T(x_k - \hat{x}_k)]$ , т. е. математического ожидания среднеквадратичной погрешности оценивания при априорно известных вероятностных характеристиках шумов  $q_k$  и  $r_k$ .

В обоих рассмотренных методах свойства  $q_k$  выбирают по представлениям о маневренности судов (оптимизируя, таким образом, алгоритм только для объектов одного класса). Свойства  $r_k$  определяют исходя из значений  $\Delta\varphi$  и  $\Delta r$  (разрешения радара), что не всегда корректно. Суть проблемы в том, что радиолокационный образ судна при высоком разрешении радара представляет собой не одиночное значение амплитуды, а целую матрицу таких амплитуд, сложно распределенных и зависящих от ориентации судна и расположения на нем локальных источников отраженного эхо-сигнала [9] (рис. 1).

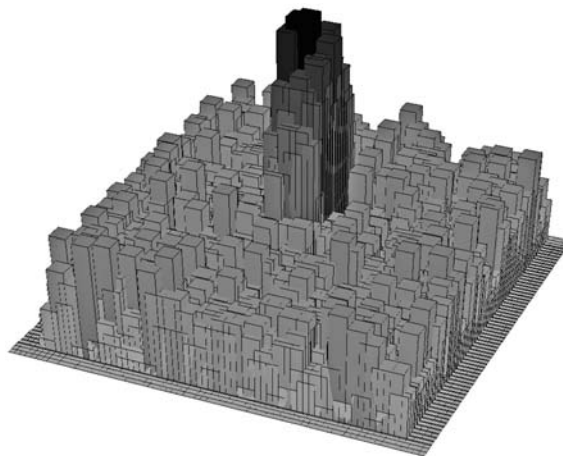


Рис. 1. Отраженный радиолокационный эхо-сигнал судна на фоне эхо-сигналов подстилающей поверхности (моря)

Соответственно и вероятностные характеристики ошибок измерений  $r_k$  можно априорно оценить только приблизительно, с точностью до порядка величин.

Недостоверность представлений о параметрах  $q_k$  и  $r_k$  способна существенно снизить эффективность алгоритмов, например, сделать их сходимость чересчур долгой или приводить к срыву сопровождения судна при его маневрировании.

Для преодоления указанных трудностей в настоящей работе предлагается особый подход к построению алгоритма сопровождения, основанный на идее обучения. Рассмотрим модель однослойной нейронной сети (рис. 2).

Здесь  $\delta z_x, \delta z_y$  — координаты вектора невязок измерений  $z_{k+1} - H\hat{x}_{k+1}$  из формулы (3);  $I_1, I_2$  — точки разветвлений;  $K_{ij}$  — весовые коэффициенты;  $\Sigma_i$  — сумматоры;  $f$  — функции активации;  $\delta x, \delta v_x, \delta y, \delta v_y$  — координаты вектора невязок  $\delta x_{k+1}$  для вектора состояния, так что

$$\hat{x}_{k+1} = \Phi \hat{x}_k + \delta x_{k+1}. \quad (5)$$

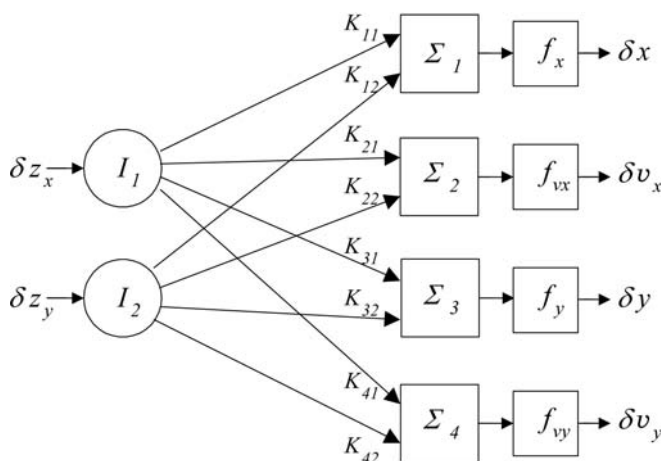


Рис. 2. Модель нейронной сети

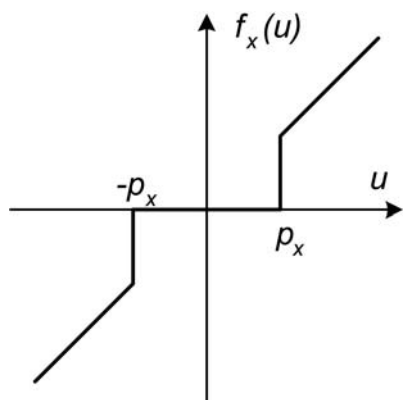


Рис. 3. Функция активации

Функции активации имеют вид (рис. 3)

$$f_x(u) = f_y(u) = \begin{cases} u, & u < -p_x; \\ 0, & -p_x \leq u \leq p_x; \\ u, & u > p_x; \end{cases}$$

$$f_{vx}(u) = f_{vy}(u) = \begin{cases} u, & u < -p_{vx}; \\ 0, & -p_{vx} \leq u \leq p_{vx}; \\ u, & u > p_{vx}. \end{cases} \quad (6)$$

Обучение сети (см. рис. 2) состоит в подборе коэффициентов  $K_{ij}$  и параметров функции активации  $p_x, p_{vx}$ .

### Метод решения задачи

Примем, что весовые коэффициенты сети (см. рис. 2) соответствуют значениям матрицы  $K$  в формуле (4). Таким образом, обучение сети будет состоять в выборе значений  $\alpha, \beta, p_x, p_{vx}$ .

В качестве обучающей выборки возьмем  $N$  измерений  $z_k, k = \overline{1, N}$ . Пусть на этих измерениях реализуется алгоритм сопровождения (5). Множество возможных значений кортежа  $\{\alpha, \beta, p_x, p_{vx}\}_i$  порождает множество оценок траектории объекта  $\{\hat{x}_k\}_i, i = \overline{1, U}$ . Оценка траектории, обеспечивающая минимум функционала

$$I = \sum_{k=1}^N [(z_{k+1} - H\hat{x}_{k+1})^T (z_{k+1} - H\hat{x}_{k+1})]$$

будет считаться окончательной оценкой. Соответствующие этой окончательной траектории значения  $\alpha, \beta, p_x, p_{vx}$  и будут определять конфигурацию сети (см. рис. 2). Значения  $\alpha, \beta, p_x, p_{vx}$  формируются прямым перебором с наперед заданным шагом изменения, так что  $0 < \alpha \leq 1, 0 < \beta \leq 1, 0 < p_x < d, 0 < p_{vx} < d_v$ , где  $d, d_v$  — соответственно характерный линейный размер судна и его характерная скорость.

Описанная процедура обучения может быть реализована однажды, с сохранением результатов в базе данных, например, при настройке системы для сопровождения судов различных классов на различных участках фарватера. Для тех зон акватории, где движение не регламентировано фарватерами, обучение системы может проводиться динамически — по мере движения судна. Последний случай требует значительных вычислительных ресурсов, однако высокая доля параллельных операций в задаче обучения открывает перспективы ее решения с помощью высокопроизводительных вычислительных систем.



## Результаты численного эксперимента

При моделировании задачи было принято, что информационной базой СУДС является двухкоординатный радар кругового обзора (например, типа Raytheon) с периодом обращения 3 с и разрешением по углу и дальности соответственно  $\Delta\varphi = 0,03^\circ$  и  $\Delta r = 6$  м.

На рис. 4 показана моделируемая траектория движения судна. Вначале судно движется прямолинейно и равномерно (скорость  $v = 20$  м/с), а затем совершает маневр — поворот с радиусом 300 м. (такие кинематические свойства вполне характерны для современных маломерных судов). Число последовательных измерений, участвующих в работе алгоритмов сопровождения,  $N = 10$ .

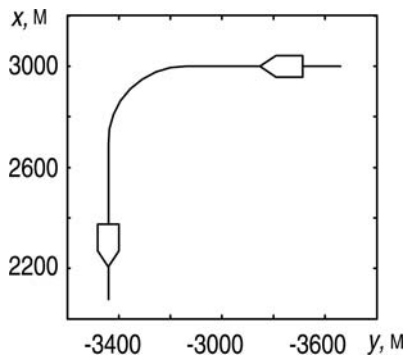


Рис. 4. Траектория движения судна

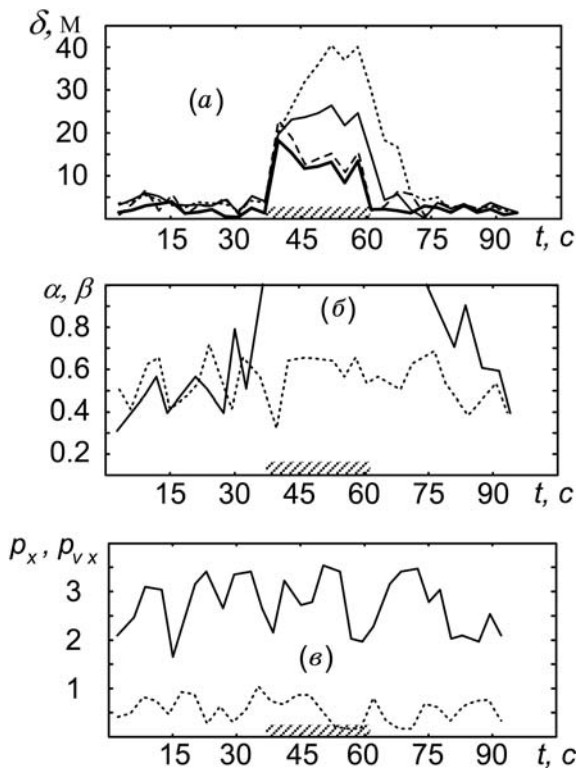


Рис. 5. Работа алгоритма сопровождения; штрихами по оси абсцисс показан участок маневрирования судна

На рис. 5, а показана погрешность оценивания  $\delta$  положения судна по мере его движения алгоритмом Калмана (пунктир);  $\alpha - \beta$  алгоритмом (3)—(4) при  $\alpha = 0,9$ ,  $\beta = 0,5$  (тонкая сплошная линия); нейросетевым алгоритмом (5) с функциями активации (6) (жирная сплошная линия); нейросетевым алгоритмом (5) с функциями активации  $f_x(u) = f_y(u) = u$ ,  $f_{vx}(u) = f_{vy}(u) = u$  (штриховая линия). В данном случае величина  $\delta$  дает представление о ширине строба, необходимого для устойчивого сопровождения объекта. Видно, что наименьшее значение максимума величины  $\delta$  реализует нейросетевой алгоритм (5) с функциями активации (6). Это свидетельствует о наибольшей устойчивости этого алгоритма по отношению к срыву сопровождения (минимально необходимый радиус строба для него  $\approx 20$  м). Алгоритм Калмана требует строб самого большого радиуса (не менее 40 м). Кроме того, оба варианта нейросетевого алгоритма быстро реагируют на изменение характера движения судна (начало и окончание маневрирования).

С изменением скорости объекта и радиуса его поворота при маневрировании соответственно изменяется и величина  $\delta$  — прямо пропорционально квадрату скорости и обратно пропорционально радиусу.

На рис. 5, б показаны значения коэффициентов  $\alpha$  (сплошная линия) и  $\beta$  (пунктир), а на рис. 5, в — коэффициентов  $p_x$  (сплошная линия) и  $p_{vx}$  (пунктир), найденных процедурой обучения нейросетевого алгоритма (5) с функциями активации (6). Видно, что коэффициент  $\alpha$  сохраняет высокие значения в процессе маневрирования судна и некоторое время после его окончания — пока измерения траектории, сделанные при выходе из маневра, принимают участие в вычислительной процедуре. Коэффициенты  $p_x$ ,  $p_{vx}$  достаточно стабильно и устойчиво характеризуют ошибки измерений, что подтверждает возможность предварительного обучения системы сопровождения при ее настройке.

## Заключение

При решении задачи сопровождения для насыщенных потоков маломерных маневрирующих судов повышается вероятность срыва сопровождения и, соответственно, нарушения оптимальных режимов работы СУДС при обеспечении безопасности коллективного движения. Предложенная в работе нейросетевая модельная интерпретация задачи автосопровождения судна позволяет, как показывают результаты эксперимента, существенно снизить вероятность срыва сопровождения стробом небольшого радиуса.

Результаты работы ориентированы на автоматизацию и расширение функций современных систем управления движением судов.

### Список литературы

1. **Модеев Р. Н.** СУДС — ядро информационной системы порта // Морские порты. 2010. № 8. С. 27—29.
2. **ОАО Норфес.** URL: <http://www.norfes.ru/>
3. **Девятисильный А. С., Дорожко В. М., Лоскутов Н. В.** Информационные модели систем управления безопасностью движения в насыщенных судопотоках // Проблемы безопасности и чрезвычайных ситуаций. 2007. № 1. С. 114—128.
4. **Девятисильный А. С., Дорожко В. М., Гриняк В. М.** и др. Система экспертных оценок состояния безопасности на морских акваториях // Информационные технологии. 2004. № 11. С. 48—53.
5. **Кузьмин С. З.** Основы теории цифровой обработки радиолокационной информации. М.: Сов. радио, 1974. 431 с.
6. **Тихонов В. И., Теплинский И. С.** Квазиоптимальное слежение за маневрирующими объектами // Радиотехника и электроника. 1989. Т. 34. № 4. С. 792—797.
7. **Benedict T. R., Bordner G. R.** Synthesis of an optimal set of radar track-while-scan smoothing equations // IRE Trans, on AC-1, July 1962. P. 27—32.
8. **Калман Р., Фалб П., Арbib М.** Очерки по математической теории систем. М.: Едиториал УРСС, 2010. 400 с.
9. **Дорожко В. М.** Имитационная модель радиолокационного эхо-сигнала // Дальневосточный математический журнал. 2001. № 1. С. 98—113.

УДК 004.93'1

**М. В. Петрушан**, науч. сотр.,  
e-mail: drn@bk.ru,

**А. И. Самарин**, ст. науч. сотр.,  
НИИ нейрокибернетики им. А. Б. Когана  
Южного федерального университета,  
Ростов-на-Дону,  
e-mail: samarin@krinc.ru

## Метод контрастирования описаний изображений лиц для системы санкционированного доступа

*Предлагается метод контрастирования описаний визуальных образов для задачи распознавания лиц на примере системы санкционированного доступа на охраняемую территорию. Метод основан на введении и настройке весовых коэффициентов к каждому компоненту описания, характеризующих устойчивые и уникальные признаки. Изначально неразделимые описания в рамках выбранной метрики признакового пространства становятся разделимыми после процедуры обучения, так как при распознавании больший вес в процедуру сравнения будут вносить уникальные компоненты, выявленные при контрастировании. Для каждого эталона формируется свой набор устойчивых и уникальных признаков, при предъявлении нового образа модифицируются только коэффициенты соответствующего эталона, позволяя дообучать систему распознавания в реальном времени без необходимости глобального переобучения.*

**Ключевые слова:** распознавание лиц, анализ изображений, машинное обучение, контрастирование описаний

### Введение

В любых задачах, связанных с биометрической идентификацией, можно выделить этапы формирования описаний и сравнения таких описаний. Проблемам формирования и сравнения описаний

посвящено множество работ [1, 2]. Как правило, эти две проблемы рассматриваются отдельно. На этапе формирования описания существенным требованием является инвариантность получаемого описания к ряду трансформаций [3, 4] или изменений внешних условий получения биометрического образа [1]. Процедуру сравнения описаний конечно нельзя считать независимой от вида описания, однако число методов сравнения векторных представлений образов более ограничено, чем число методов формирования описаний. Например, в работе [5] возможные нейросетевые алгоритмы сравнения зрительных образов делятся на две группы. К первой группе отнесены методы, связанные с разбиением пространства признаков секущими гиперплоскостями, например SVM-метод [6]. К характерным особенностям таких алгоритмов можно отнести отсутствие единственного эталонного описания определенного класса и необходимость использования достаточно больших обучающих выборок. Нейросетевая реализация такого метода представлена структурой перцептронного типа [7, 8]. Ко второй группе относятся алгоритмы сравнения, основанные на вычислении меры близости текущего и эталонного описаний. В некоторой степени нейросетевой реализацией такого метода можно считать сети Хопфилда [9].

Существует ряд ограничений, не позволяющих применять многие нейросетевые решения в задачах распознавания биометрических образов, особенно в тех, где требуется параллельное распознавание и обучение в режиме реального времени. Обычно для успешного решения задачи распознавания в рамках нейросетевого подхода требуется достаточно большая обучающая выборка, тогда как для ряда систем биометрической идентификации существенным требованием является возможность распознавания даже по одному единственному эталону. Кроме того, для многих нейросетевых решений характерна проблема сравнительно долгого переобучения сети при добавлении небольшого числа новых образов.

В данной работе рассматривается метод сравнения векторных описаний с параллельным обучением. Существенным требованием к такому методу является возможность дообучения в режиме реального времени при распознавании биометрических образов. В понятийной интерпретации такая способность биометрической системы формировать эталонное описание по единственному предъявлению эталонного образа и дообучаться по мере многократного прохождения идентифицируемых объектов в зоне контроля означает, что разработанные алгоритмы адаптации обеспечивают улучшение "узнавания" уже в процессе реальной рабочей эксплуатации биометрической системы. Процесс улучшения "узнавания" и различения изображений лиц в данной работе назван контрастированием описаний. В качестве исходных образов используются изображения лиц. Тестирование системы осуществлялось в процессе эксплуатации биометрической системы в режиме санкционированного доступа на охраняемую территорию.

### Структурно-контекстное описание изображений лиц

Описание изображения лица строится в рамках модели рассматривающих движений визуального сенсора как совокупность откликов контекстного фильтра, представленного на рис. 1. Точки позиционирования (точки фиксации взгляда) такого сенсора расставляются относительно предварительно найденных центров глаз. Экспериментально установлено, что достаточное число таких точек фиксации взгляда при описании изображения лица не превышает 20 с радиусом зоны внимания, равным одной трети расстояния между центрами глаз. Эти точки на изображении лица выбираются по критерию минимизации неопределенности в признаковом описании изображения. Подробно алгоритм описания изображений в модели рассматривающих движений визуального сенсора изложен в работе [1].

Сенсор имеет центральное (фовеальное) поле  $S_0$  и некоторую совокупность периферических радиальных рецептивных полей  $S(rad)_l$ ,  $l = 1...16$ . Поле зрения представляет собой круговую область с управ-



Рис. 1. Структура рецептивных полей визуального сенсора

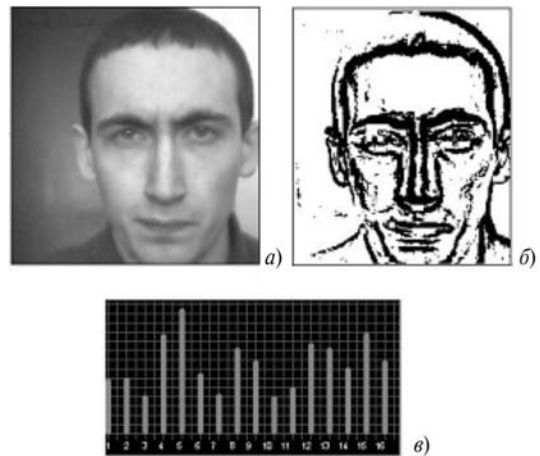


Рис. 2. Градиентное представление яркостного изображения: *a* — фрагмент изображения; *б* — градиентный препарат, черным цветом выделены точки высокоамплитудных перепадов яркости; *в* — интегральная гистограмма ориентаций перепадов яркости

ляемым радиусом  $R$  зоны внимания, центр которой фиксируется на произвольной точке изображения.

Каждое рецептивное поле визуального сенсора суммирует информацию в соответствующей области изображения и представляет ее в виде гистограммы квантованных (на 16 направлений) ориентаций перепадов яркости. Пример такой гистограммы для фрагмента изображения приведен на рис. 2, *в* (по оси абсцисс отложены значения квантованных направлений ориентаций, по оси ординат — доля представимости этой ориентации на изображении).

Для каждой точки фиксации взгляда фовеального сенсора записывается номер точки фиксации  $N$ , ее координаты  $(i, j)$ , радиус зоны внимания  $R$ , значение доминирующей ориентации  $\alpha$  в центральном рецептивном поле и матрица состояний радиальных рецептивных полей  $\mathbf{M}(l, k)$ , описывающая распределение  $l$  ориентаций перепадов яркости по  $k$  секторам фовеального фильтра:

$$\{N, (i, j)_N, R_N, \alpha_N, \mathbf{M}_N(l, k)\}.$$

На рис. 3 представлены четыре последовательных уровня формирования описания лица. Переход от яркостного изображения к градиентному дает инвариантность описания к вариации средней яркости. Описание в виде гистограмм ориентированных перепадов яркости позволяет получить инвариантность к линейным трансформациям анализируемого образа и устойчивость к небольшим нелинейным искажениям [1]. И, наконец, привязанная к системе координат объекта (центры глаз) совокупность локальных описаний в виде откликов визуального сенсора в максимально информативных областях лица дает возможность получить компактные описания многообразия изображений лиц достаточной информационной емкости для практических задач биометрической идентификации. Найденные с помощью интегральных ориента-

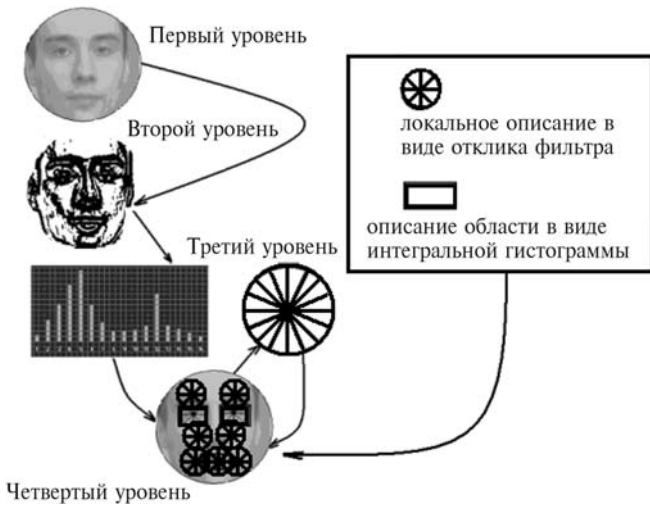


Рис. 3. Многоуровневое признаковое описание

ционных описаний области глаз задают характерные размеры лица на изображении и преднастраивают параметры визуального сенсора (обратные стрелки с четвертого на третий уровень на рис. 3).

Описание изображения лица в виде совокупности откликов визуального сенсора в точках позиционирования позволяет сравнивать не изображение с изображением, а описание с описанием.

### Сравнение и контрастирование описаний изображений лиц

Процедура распознавания заключается в сравнении сформированного признакового описания в виде совокупности откликов контекстных фильтров в точках позиционирования центров визуального сенсора на анализируемом изображении и на эталонном. Отклик фильтра в виде матрицы  $16 \times 16$  (шестнадцать ориентаций перепадов яркости в шестнадцати секторах визуального сенсора) можно рассматривать как точку в 256-мерном пространстве признаков. Тогда расхождение  $D_n$  откликов в  $n$ -й точке фиксации взгляда (позиционирования) можно интерпретировать как расстояние, например, — Евклидово, между одноименными точками в таком признаковом пространстве:

$$D_n = \sqrt{\sum_{k,l} (M_{1kl} - M_{2kl})^2}, \quad (1)$$

где  $M_1$  — компоненты признакового вектора описания эталонного образа;  $M_2$  — компоненты признакового вектора описания текущего образа;  $k, l$  — соответственно, номера строки и столбца матрицы отклика визуального сенсора.

Отклики считаются идентичными, если  $D_n$  меньше фиксированного порога  $D_t$ , определяемого экспериментально. Неравенство  $D_n < D_t$  ограничивает окрестность точки-отклика гиперсферой в пространстве признаков. Все точки-отклики, по-

падающие внутрь гиперсферы, считаются идентичными. Однако обычно совокупность признаков векторов одного класса в гиперпространстве не укладывается в гипершар, а имеет другую форму, вытянутую по одним осям и сжатую — по другим. Учет такой особенности отражается в следующих формулах:

$$D_n = \sqrt{\sum_{k,l} W_{kl} (M_{1kl} - M_{2kl})^2}; \text{ при } \sum_{k,l} W_{kl} = 256. \quad (2)$$

Весовые коэффициенты  $W_{kl}$  адаптивно подстраиваются по мере накопления базы изображений "своих" и "чужих" лиц. В данном подходе метод обучения заключается во введении и настройке весовых коэффициентов, основанной на анализе межклассовых и внутриклассовых вариаций компонентов описаний в пространстве откликов контекстных фильтров.

Коэффициенты  $W_{kl}$  меняются в соответствии с критериями устойчивости и уникальности, при этом всегда должно выполняться условие нормировки (2).

Критерий устойчивости описания, сформулированный для выделенной ориентации  $k$  в секторе  $l$ , имеет вид:

$$\frac{1}{m} \sum_m |M_{N_0}(kl) - M_{N_T}(kl)|_m < K_1, \quad (3)$$

где  $M_{N_0}(k, l)$ ,  $M_{N_T}(k, l)$  — отклики контекстных фильтров в одноименных точках эталонного и текущих (накопленных) изображений данного лица в разных условиях;  $m$  — число его тестовых предъявлений;  $K_1$  — порог допустимых изменений признака. Выражение (3) дает оценку варибельности данного признака в пространстве  $\{k, l\}$  при многократном распознавании данного лица в разных условиях. При выполнении этого условия признак считается устойчивым и его вес увеличивается.

Если (3) = true  $\rightarrow W_{kl} = s W_{kl}$ ;

Если (3) = false  $\rightarrow W_{kl} = \frac{1}{s} W_{kl}$ ,

где  $s$  — параметр обучения. При  $s = 1$  обучение отсутствует, при  $s > 1$  веса  $W_{kl}$  начинают меняться в соответствии с устойчивостью компонентов признакового вектора. Мультипликативное изменение весов, в отличие от аддитивного, позволяет добиться инвариантности модификации весов к последовательности образов, предъявляемых для обучения.

Процесс настройки весов может быть более гибким, если сделать параметр обучения не постоянным, а зависящим от вариации компонента признакового вектора. Определим усредненную вариацию компонента признакового описания в преде-

лах одного класса образов, она записана в левой части неравенства (3):

$$S_1(kl) = \frac{1}{m} \sum |M_{N_0}(kl) - M_{N_T}(kl)|.$$

Тогда правило модификации весов будет выглядеть следующим образом:

$$W_{kl} \Rightarrow \frac{K_1}{S_1(kl)} W_{kl},$$

где  $K_1$  — порог максимально допустимых изменений признака.

Далее проводится оценка вариабельности этого признака, устойчивого при формировании эталонного описания данного лица, с признаками  $\{k, l\}$  эталонных описаний других лиц, сформированных в одноименных точках фиксации взгляда. Если вариабельность рассогласования этого признака эталонного описания данного лица с эталонными описаниями других лиц велика, то признак считается уникальным, и его вес при аутентификации данной личности увеличивается.

Критерий уникальности описания:

$$\frac{1}{m_2} \sum |M_{N_0}(kl) - M_{N_{m_2}}(kl)|_{m_2} > K_2, \quad (4)$$

где  $M_{N_m}(k, l)$  — отклики контекстных фильтров в одноименных точках, соответствующие эталонам  $FD_m$  признакового описания  $m_2$ -х объектов разных классов;  $K_2$  — порог минимального разброса между классами. При выполнении этого условия признак считается уникальным и его вес увеличивается.

Если (4) = true  $\rightarrow W_{kl} \Rightarrow sW_{kl}$ .

Если (4) = false  $\rightarrow W_{kl} \Rightarrow \frac{1}{s} W_{kl}$ , где  $s$  — параметр обучения ( $s > 1$ ).

Параметр обучения можно связать с вариацией компонент между классами. Введем определение средней вариации компоненты признакового описания между классами:

$$S_2(kl) = \frac{1}{m_2} \sum |M_{N_0}(kl) - M_{N_{m_2}}(kl)|_{m_2}.$$

Тогда правило модификации весов будет выглядеть следующим образом:

$$W_{kl} \Rightarrow \frac{S_2(kl)}{K_2} W_{kl},$$

где  $K_2$  — порог изменений признака между классами.

Следует отметить, что при выполнении условия (3) и невыполнении условия (4) признак считается

устойчивым для всех лиц и должен иметь большой вес при сегментации изображения в процессе поиска области лица. Для большей эффективности этой процедуры вес признака увеличивается. Таким образом, для задач идентификации и сегментации определяются два разных массива весовых коэффициентов.

Метод контрастирования описаний не меняет признаковые вектора и эталонные описания, меняется метрика пространства признаков, причем индивидуально для каждого эталонного описания, в отличие от множества методов обучения, которые глобально меняют меру близости для всех эталонов (при глобальном обучении возникает так называемый эффект "забывания" прошлых обучающих выборок).

Предлагаемая процедура обучения позволяет постепенно модифицировать весовые коэффициенты в формуле (2).

#### Экспериментальные расчеты показателей эффективности метода контрастирования описаний

Экспериментальные исследования проводили на комплексной установке системы санкционированного доступа, установленной в НИИ нейрокибернетики им. А. Б. Когана. Для каждого человека (из 30 авторизованных) было представлено три эта-

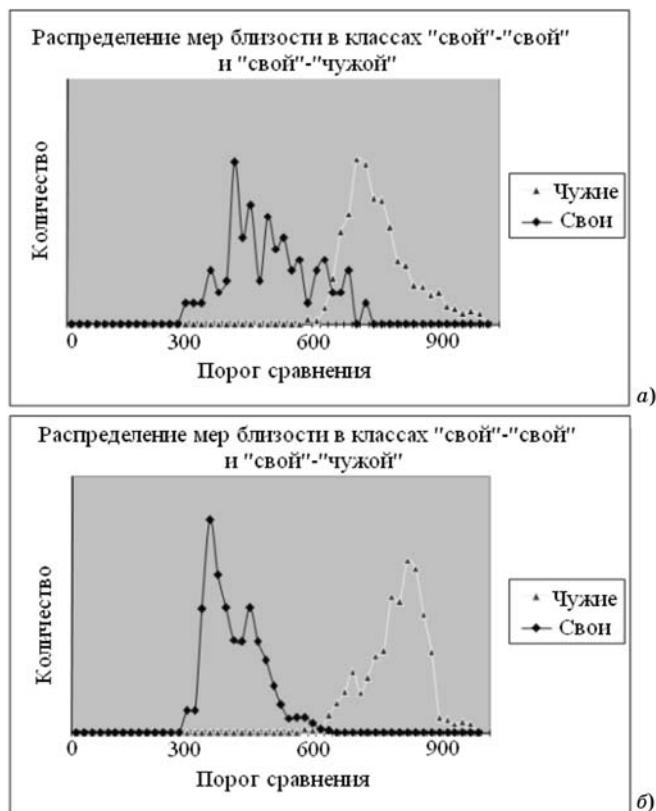


Рис. 4. Графики распределения мер близости для попарных сравнений "своих" и "чужих": а — до обучения; б — после обучения

лонных изображения его лица. Далее, каждый человек 4 раза проходил процедуру распознавания. Результат всех попарных сравнений на полученной выборке без процедуры обучения приведен на рис. 4, а.

На рис. 4, б приведен результат попарных сравнений на той же выборке при включенной процедуре обучения, изменяющей весовые коэффициенты, характеризующие устойчивые и уникальные компоненты описания изображений лиц каждого человека, при проходах через систему санкционированного доступа.

До процедуры обучения мера близости между изображениями лиц разных людей могла быть меньше, чем мера близости между изображениями лица одного человека. После процедуры обучения диапазоны мер близости попарных сравнений "своих" изображений данного человека и "чужих" по отношению к его эталонному описанию были существенно разделены. В исследование включены только те изображения, на которых в процедуре сегментации были корректно найдены положения центров глаз (98 % изображений).

До процедуры обучения при фиксированной ошибке второго рода ( $FAR < 10^{-4}$ ) ошибка первого рода составляла  $FRR = 0,2$ .

После процедуры обучения при фиксированной ошибке второго рода ( $FAR < 10^{-4}$ ) ошибка первого рода составляла  $FRR = 0,06$ .

### Заключение

Метод контрастирования векторных описаний изображений лиц, основанный на введении и настройке весовых коэффициентов, отражающих устойчивые и уникальные компоненты описания, позволяет устранить неопределенность при распознавании образов, выделив те признаковые компоненты, которые наиболее отличают его от остальных

описаний и при этом сами инвариантны к изменениям внешних условий. Такой набор весовых коэффициентов формируется для каждого эталонного образа, кусочно-линейно задавая функцию весов на множестве эталонов, что позволяет избежать проблемы "забывания" предыдущих образов при глобальном переобучении. Система распознавания изображений лиц для задачи санкционированного доступа, построенная на основе такого метода, способна улучшать качество распознавания в процессе эксплуатации.

### Список литературы

1. Самарин А. И., Шевченко М. А. Алгоритм автоматического построения описаний объектов сцены в базе рассматриваемых движений фовеального сенсора. "Нейроинформатика — 2002": Сборник научных трудов. Ч. 1. М.: МИФИ, 2002. С. 85—94.
2. Belongie S., Malik J., Puzicha J. Shape matching and object recognition using shape contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002. Vol. 24, no. 24. P. 509—522.
3. Dorko G., Schmid C. Selection of Scale-Invariant Parts for Object Class Recognition. // Proc. of the 9th IEEE International Conference on Computer Vision. October 2003. P. 634—640.
4. Leibe B., Schiele B. Scale Invariant Object Categorization Using a Scale-Adaptive Mean-Shift Search // Proc. of the DAGM Conference. Springer LNCS. 2004. Vol. 3175. P. 145—153.
5. Гаврилей Ю. К., Самарин А. И., Шепелев И. Е. Преднастройка нейронных сетей в задачах зрительного анализа // "Нейроинформатика — 2000": Сб. науч. тр. Ч. 1. М.: Изд-во МИФИ, 2000. С. 161—166.
6. Guo G., Li S. Z., Chan K. Face Recognition by Support Vector Machines // Proc. of the IEEE International Conference on Automatic Face and Gesture Recognition, 26—30 March 2000, Grenoble, France. P. 196—201.
7. Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain // Psychological Review. 1958. N 65. P. 386—407.
8. Dasgupta S., Kalai A. T., Monteleoni C. Analysis of Perceptron-Based Active Learning // Journal of Machine Learning Research. 2009. Vol. 10. P. 281—299.
9. Hopfield J. J. Neural networks and physical systems with emergent collective computational abilities // Proc. of National Academy of Sciences. 1982. Vol. 79, no. 8. P. 2554—2558.

## CONTENTS

### Afonin S. A., Bakhtin A. V. *Concept Hierarchy Construction by Means of Lexical Patterns* . . . . . 2

Internet search engines admit automated search for specified keywords and provide document snippets as long as number of pages containing these keywords. In this work an approach to automated concept hierarchy construction using an Internet search engine is proposed. It uses both lexical patterns and quantitative filtering techniques. Preliminary experimental results presented in this work show that proposed approach may be used real-life applications.

**Keywords:** lexical patterns, ontology, hyponymy relation

### Kostenko K. I. *Operations of the Digital Knowledge Spaces Unified Technology*. . . . . 8

In work the system of operations for construction of components of spaces of knowledge is defined. By means of such operations developments and transformations of formal models of different level in the models supposing effective digital realizations are realized. Schemes of application of operations make a basis of technology of construction of applied spaces of knowledge.

**Keywords:** knowledge space, algebraic system, homomorphism of models, integration of models

<b>Hodashinski I. A.</b> <i>Simulated Annealing and Methods Based on Derivatives for Fuzzy System Identification</i> . . .	14
<p>In this paper a methodology which identifies fuzzy systems is developed. By using input-output data, optimal parameter values are found by means of simulated annealing, Kalman filter, least-squares method, Levenberg—Marquardt method, gradient descent method, and hybrid method. Several numerical examples demonstrate the effectiveness of the proposed identification methods.</p> <p><b>Keywords:</b> fuzzy system identification, simulated annealing, Kalman filter, least-squares method, Levenberg—Marquardt method, gradient descent method</p>	
<b>Palamar I. N., Julin S. S.</b> <i>The Analysis of Applicability of Modern Methods of Statistical Modeling for Recognition and Understanding of Speech</i> . . . . .	20
<p>In article the review of modern methods of recognition and understanding of speech on the basis of statistical modelling is executed. The classification scheme of methods is offered. Representatives generative and discriminative models are considered and their comparative analysis is carried out. It is proved, that methods on the basis of conditional random fields are adequate base for system engineering of understanding of speech.</p> <p><b>Keywords:</b> speech recognition, generative and discriminative statistical models, Bayes classier and maximum entropy classier, hidden markov model, conditional random fields</p>	
<b>Andreichikov A. V., Chorychev A. A., Andreichikova O. N.</b> <i>Expert System for Initial Stages of Innovative Designing</i> . . . . .	26
<p>The paper describes a new approach to the creative engineering with use of inventions databases. This approach can be put in the basis of intelligent software for support of innovative designing.</p> <p><b>Keywords:</b> expert systems, heuristic classification, forecasting of the new markets, invention by analogy</p>	
<b>Bobkov S. G., Zadiabin S. O., Krinitski A. V., Novojilov E. E.</b> <i>USB 2.0 Host Controller Design Verification</i> . . . . .	33
<p>This article describes experience of applying VMM verification methodology for the block of SoC 1890VM61A — USB 2.0 host controller design, implementing ECHI register model.</p> <p><b>Keywords:</b> digital circuits, VLSI verification, SoC, black-box verification approach</p>	
<b>Kuznetsov E. S., Talicky E. N., Shumarin S. V.</b> <i>Automating the Design of Cells of Electronic Equipment Vibroprotection by Damping Links</i> . . . . .	37
<p>Propose an algorithm of designing and optimization of cells of electronic equipment vibroprotection by damping links. We consider systems of finite-element analysis for the implementation of this algorithm. Given the results of experimental research confirming the possibility of the application developed CAD for calculation and optimization of cells of electronic equipment vibroprotection by damping links.</p> <p><b>Keywords:</b> electronic equipment, vibroprotection, damping links, CAD ANSYS</p>	
<b>Kerimov S. G., Kerimova E. S.</b> <i>Repository in Information Systems and Software</i> . . . . .	43
<p>The role of repository in information systems and software is considered. The special features of the development of repositories for OLTP, OLAP systems and program packages placed in the Internet are presented. An example of repository construction of the database is shown. The survey of known repositories of software published in the Internet is given.</p> <p><b>Keywords:</b> repository, information system, database, data warehouse, software, Internet</p>	
<b>Elesina S. I., Nikiforov M. B.</b> <i>Improving the Efficiency of Genetic Algorithm</i> . . . . .	49
<p>The investigation of the genetic algorithm to obtain optimal settings for its use in correlation-extremal navigation systems. To reduce the complexity of finding a global extremum offered different strategies for clustering search area. Study their effectiveness. It is shown that the correlation-extremal navigation systems operating on the basis of radar images productivity increases on the order.</p> <p><b>Key words:</b> global extremum objective function, search, clustering, genetic algorithm, the correlation-extremal navigation systems</p>	
<b>Vasylyev V. V., Simak L. A.</b> <i>The Polynomial Approximation of Signals as Operational Calculus: Application to the Simulation of Dynamic Systems of Fractional Order</i> . . . . .	55
<p>Analytical approximation of continuous signals in the form of polynomials of best approximation with different systems of basis functions is considered as non-classical type of operational calculus. It is shows how to use this calculus to mathematical modeling of dynamic systems both a whole and fractional orders via using the systems of software of computer algebra. Illustrative examples of the approximation of signals and the solution of differential equations of fractional order are given. The advantages and disadvantages of the proposed approach and the prospects for further research are discussed.</p> <p><b>Keywords:</b> modeling and simulation, differential transform, signal approximation, operational calculus, dynamical system</p>	

**Dobrotvorskiy A. S.** *Estimation of Multivalued Alphabet Quantum Computers Associative Development Environment Usage Prospects* . . . . .62

The possibility of quantum computers development environments hardware emulators' creation as well as ways to improve their performance is considered. It is shown that in the first quantum computers' generation binary system will remain as the least precise with a view to quantum registers resulting states identification systems, and as the least hardware-intensive from a position of associative development environments based on traditional computers creation.

**Keywords:** quantum computers, associative memory, emulator

**Solomakha A. A., Milova K. A., Gorbachenko V. I.** *Neyroset Consulting Mediko-Technological Model of Prognostication of Risk of Development of Complications for Patients with the Festering-Destructive Diseases of Lights* . . . . .66

A research purpose was creation of neyroset consulting mediko-technological model of prognostication of risk of development of surgical complications for patients with the festering-destructive diseases of lights. For the decision of the put purpose were conducted statistical analysis of basic data, choice of criteria for the construction of model, and also experiments are conducted on the selection of type and architecture of neuron network for realization of prognosis. The quantum of datains and their encoding was applied for the improvement of quality of work of neuron network. As a result of experiments a model, carrying out prognostication of risk of development of surgical complications with authenticity 90 %, was built. Therefore management the state of patients with the festering-destructive diseases of lights it is expedient to carry out by the neyroset mediko-technological system.

**Keywords:** neural networks forecasting, pyoinflammatory complications, a neural network, modeling, management, medical process, torakotomia

**Devyatysilny A. S., Grinyak V. M.** *Neural Target Tracking for Vessel Traffic Systems* . . . . .70

Target tracking problem for vessel traffic systems with 2D radars is discussed. Conditions of tracking crash are watched. Neural algorithm for robust track-while-scan tracking is considered.

**Keywords:** vessel traffic system, target tracking, radar measurements, trajectory estimating

**Petrushan M. V., Samarin A. I.** *Contrast Enhancement Method for Face Image Descriptions* . . . . .74

The image descriptions contrast enhancement method for face recognition based authorized access system is suggested. The method is based on searching stable and unique components of descriptions and marking them with weights coefficients. A priori similar descriptions may become different after learning, because the most stable and unique components have more significant weights then others after contrasting. Each template has own coefficients set, new image description modifies coefficients of only correspondent template while recognizing without global relearning, so such system is able to learn in real time.

**Keywords:** face recognition, image analysis, machine learning

---

---

**Адрес редакции:**

107076, Москва, Стромынский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *Е. В. Конова*.

Корректор *М.Г. Джавадян*.

Сдано в набор 27.12.2011. Подписано в печать 16.02.2012. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 9,8. Заказ ИТ312. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".

105120, г. Москва, ул. Нижняя Сыромятническая, д. 5/7, стр. 2, офис 2.