

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

6(190)
2012

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Талалай М. С., Трушин К. В., Венгер О. В. Логический синтез булевой функции для проектирования интегральных схем на транзисторных шаблонах 2
Бибило П. Н. Экспертные знания в САПР цифровых устройств 11

СЕТИ И СИСТЕМЫ СВЯЗИ

- Тарасов В. Н., Мезенцева Е. М. Организация защиты компьютерных сетей. Метод многомодульной фильтрации спама на Web-сайтах. 18
Карутин С. Н. Метод автономного эфемеридно-временного обеспечения навигационных космических аппаратов системы ГЛОНАСС. 23

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Бочков М. В., Лалушкин Ю. П., Шкадов А. А. Имитационное моделирование процессов управления защищенностью ресурсов компьютерной сети на основе иерархических раскрашенных временных сетей Петри 29
Козленко А. В. Метод оценки защищенности информации от несанкционированного доступа в компьютерных сетях в условиях комплексных угроз 33

КОМПЬЮТЕРНАЯ ГРАФИКА И ГЕОМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ

- Бобков В. А., Роньшин Ю. И., Машенцев В. Ю. Построение карт глубин на множественной последовательности видов 39
Бимаков В. А., Бимаков Е. В. Принципы построения и функционирования устройств для высокоскоростной обработки n -мерных пиксельных геометрических моделей. 44

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Панин И. Г., Благовещенский В. В. Численное решение уравнения эволюции дислокационного источника Франка — Рида 50
Львович Я. Е., Каширина И. Л., Тузигов А. А. Генетический алгоритм решения многокритериальной задачи повышения надежности резервирования 56

КОДИРОВАНИЕ И ОБРАБОТКА СИГНАЛОВ

- Агиевич С. Н. Оценка параметров сигналов в базисах функций сплайн-характеров 61

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ФИНАНСОВО-ЭКОНОМИЧЕСКИХ СИСТЕМАХ

- Рындин А. А., Ульев В. П. Проектирование гибридной системы скоринговой оценки кредитоспособности юридических лиц 69
Архипова Я. А. Автоматическая периодизация динамики биржевых котировок на основе критерия минимума информационного рассогласования 73

ИНФОРМАЦИЯ

- 2-я Международная конференция по истории отечественной вычислительной техники и информатики SoRuCom—2011 77
Contents 79

- Приложение Кудинов Ю. И., Келина А. Ю. Методы синтеза и настройки нечетких ПИД регуляторов Мамдани

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.

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.

УДК 004.942

М. С. Талалай, инженер-исследователь,
e-mail: mikhail.s.talalay@intel.com,

К. В. Трушин, инженер-практикант,
e-mail: konstantin.trushin@intel.com,

О. В. Венгер, инженер-исследователь,
e-mail: oleg.v.venger@intel.com,
ЗАО "ИНТЕЛ А/О", Москва

Логический синтез булевой функции для проектирования интегральных схем на транзисторных шаблонах

Описан метод логического синтеза, предназначенный для проектирования интегральных схем с использованием современных технологических процессов, требующих регулярной топологии на базовых слоях. В качестве минимальных функциональных элементов для синтеза предлагается использовать набор транзисторных схем, имеющих регулярную топологию (транзисторные шаблоны). Предлагается набор минимальных функциональных элементов для синтеза схем с регулярной топологией и описывается метод декомпозиции булевой функции. Демонстрируются результаты применения предъявленных алгоритмов на примерах. Эксперименты показывают, что предложенный стиль проектирования потенциально способен сократить занимаемую площадь по сравнению с синтезом на стандартных ячейках.

Ключевые слова: регулярная топология, регулярная заготовка, логический синтез, транзисторный шаблон

Введение

Использование современных технологий производства интегральных схем (ИС) накладывает ряд дополнительных требований на процесс проектирования и производства. Одним из требований, обеспечивающим процесс литографии, является повышение регулярности фотошаблонов, т. е. наличие периодичности в топологии. Методология, основанная на топологических заготовках (Regular Fabrics), представляется наиболее привлекательной с точки зрения регулярности и эксплуатационных характеристик [1, 2]. Также к наиболее известным и широко применяемым способам построения схем с регулярной топологией относят проектирование

структурных ИС (structural ASICs), подходы с использованием стандартных логических матриц (Gate Array) и матриц, конфигурируемых межслойными переходами (VPGA). Сравнение и общее описание данных методов, а также других, представлено в работе [3].

При проектировании ИС с шириной затвора транзистора 32 нм и менее заметно проявляются различные квантовые эффекты, что вместе с неточностью аппаратуры для литографии неминуемо приводит к появлению большого числа правил проектирования. Данные правила формулируются в виде набора запрещенных конфигураций относительно размещения объектов на слоях. Критическим слоем на кристалле становится слой поликремния, так как именно на этом слое размещаются затворы транзисторов — топологические объекты, имеющие наименьшие размеры [3]. Для современных нанометровых технологических процессов нормой стало размещение одинаковых сегментов поликремния с постоянным шагом между ними (рис. 1). При этом невозможно использовать абсолютно все сегменты для реализации затворов тран-

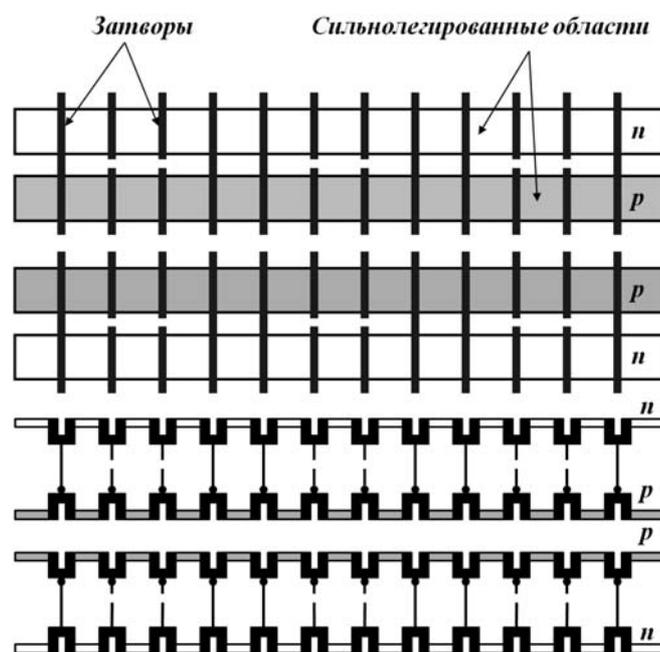


Рис. 1. Общий вид регулярной заготовки

зисторов. Неиспользуемые сегменты — нефункциональные затворы — являются накладным расходом площади кристалла.

В статье представлен новый подход синтеза ИС на регулярной заготовке, минимизирующий число нефункциональных и функциональных затворов. Отличие описываемого подхода от других заключается в том, что он использует в качестве минимальных функциональных элементов транзисторные шаблоны (см. раздел 1) — промежуточные структуры между стандартной ячейкой и отдельным транзистором. Выбор шаблона в качестве минимального функционального элемента для логического синтеза дает возможность избежать неоптимального использования затворов. Также появляется возможность состыковывать (конкатенировать) различные шаблоны, используя разделяемые сильнолегированные области одного типа, что приводит к уменьшению числа изоляционных затворов.

В статье описывается алгоритм синтеза для логической функции, использующий произвольный набор транзисторных схем в качестве множества минимальных функциональных элементов. Для проектирования схем с регулярной топологией выбирается множество транзисторных схем с подходящей топологией — регулярные транзисторные шаблоны. Предполагается, что каждый шаблон в данном множестве является частью регулярной заготовки, спроектированной в рамках технологических правил. Чтобы описать логическую функциональность шаблона, предлагается модель транзистора на уровне переключений. Потенциальные преимущества демонстрируются на примерах.

В статье обсуждается задача декомпозиции произвольной транзисторной подсхемы и определяется понятие транзисторного шаблона, описывается алгоритм декомпозиции логической функции на произвольной транзисторной схеме и алгоритм синтеза логической функции, а также исследуются свойства суперпозиции функций на шаблоне при синтезе "снизу вверх". Представлены экспериментальные результаты.

1. Постановка задачи декомпозиции, понятие регулярного шаблона

Под *комбинационной транзисторной схемой (подсхемой)* понимается древовидная структура, у которой множество вершин составляют транзисторы p - и n -типа, точки соединения проводников и особо выделенные точки входов и выхода, а в качестве ребер выступают соединительные проводники (рис. 2). Отметим, что функциональность произвольной комбинационной схемы вводится именно для древовидных структур, что является естественным ограничением. Также, не ограничивая общности, можно считать, что направление распространения логического сигнала от входов к выходу однозначно определено. Любая точка в схеме, подсоединенная

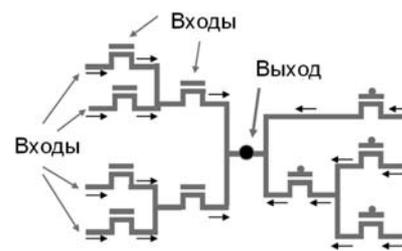


Рис. 2. Комбинационная транзисторная схема. Древовидная структура

к цепи питания или земле, также является входной точкой схемы. Чтобы подчеркнуть, что рассматриваемая схема является частью некоторой большей схемы, которая включает данный транзисторный граф, авторы используют термин *подсхема*.

Для описания функциональности представленной выше структуры предположим, что любой проводник может находиться в одном из следующих состояний: vcc , vss , z , $lowvcc$, $lowvss$. Состояние vcc означает, что проводник подсоединен к цепи питания напрямую или через цепочку открытых транзисторов p -типа. Аналогично, состояние vss на проводнике означает, что данный проводник подсоединен к цепи земли напрямую или через цепочку открытых транзисторов n -типа. Состояние z (состояние высокого импеданса) предписывается проводнику, который не имеет соединения через цепочку открытых транзисторов ни с цепью питания, ни с цепью земли. $Lowvcc$ — состояние на проводнике, которое возникает, если этот проводник подсоединен к цепи питания только через открытые транзисторы n -типа (на данном проводнике будет индуцировано напряжение, отличающееся от напряжения питания на некоторую значимую величину). $Lowvss$ — это состояние на проводнике, который подсоединен к цепи земли только через открытые транзисторы p -типа (напряжение на таком проводнике будет отличаться на значимую величину от нуля). Только состояния vcc и vss интерпретируются как логические 1 и 0 соответственно.

Предполагается, что для каждой входной точки заданы пять характеристических функций — одна функция для одного состояния. Данный набор функций называется *сигналом*. Не ограничивая общности, предположим, что все характеристические функции на всех входных точках зависят (возможно, несущественно) от единого множества булевых переменных.

В статье [4] детально описан способ пересчета характеристических функций для таких функциональных элементов (ФЭ), как транзисторы p - и n -типов, а также для точек соединения проводов. Авторы определяют понятие оператора как отображения сигналов на входах ФЭ в сигнал на выходе ФЭ. Каждый оператор выражается формулами следующего вида:

Оператор P (для транзистора p -типа).

$$\begin{aligned} bad_{cond} &= z_{gate} \vee lowvcc_{gate} \vee lowvss_{gate}; \\ vcc_{drain} &= vss_{gate} \wedge vcc_{source}; \\ vss_{drain} &= 0; \\ z_{drain} &= vss_{gate} \vee (vss_{gate} \wedge z_{source}); \\ lowvcc_{drain} &= vss_{gate} \wedge lowvcc_{source}; \\ lowvss_{drain} &= vss_{gate} \wedge (vss_{source} \vee lowvss_{source}). \end{aligned}$$

Здесь индексы *source*, *gate*, *drain* соответствуют сигналам на истоке, затворе и стоке соответственно. Функция *bad* определяет запрещенные в модели комбинации состояний, которые приводят либо к неоднозначной работе ФЭ, либо к электрическому замыканию.

Задача декомпозиции логической функции на произвольной транзисторной схеме T с множеством входов: $i_1, i_2, \dots, i_p \in I(T)$ и одним выходом o в общем случае формулируется следующим образом. Для схемы $T(i_1, i_2, \dots, i_p; o)$ и фиксированной булевой функции $f = f(x_1, x_2, \dots, x_n)$ требуется для каждого входа (индекс j) и каждого состояния (второй индекс, состояния $vcc, vss, z, lowvcc, lowvss$) найти набор характеристических функций $g_{j, vcc} = g(x_1, x_2, \dots, x_n), \dots, g_{j, lowvss} = g(x_1, x_2, \dots, x_n), j = 1, 2, \dots, p$, которые необходимо задать для входных точек так, чтобы на выходе o имели место сигналы вида

$$\begin{aligned} o_{vcc} &= f(x_1, x_2, \dots, x_n), \quad o_{vss} = \bar{f}(x_1, x_2, \dots, x_n), \\ o_z &= 0, \quad o_{lowvcc} = 0, \quad o_{lowvss} = 0. \end{aligned}$$

Для того чтобы иметь возможность поставить задачу декомпозиции для полученных сигналов в решении, для каждого входа добавляется ограничение вида:

$$\begin{aligned} g_{vcc} &= g(x_1, x_2, \dots, x_n), \quad g_{vss} = \bar{g}(x_1, x_2, \dots, x_n), \\ g_z &= 0, \quad g_{lowvcc} = 0, \quad g_{lowvss} = 0. \end{aligned}$$

Данное требование упрощает задачу и позволяет не рассматривать декомпозицию произвольного сигнала. Сигналы, удовлетворяющие данному свойству, будем называть *функциями*.

Также имеет смысл требовать более "простых" функций в решении по отношению к декомпозируемой функции, что обеспечит алгоритмическую сходимость процесса последовательного применения данной операции. Для этого, в качестве примера, авторы требуют, чтобы любая функция в решении зависела от меньшего числа существенных переменных, чем декомпозируемая функция. При этом факт того, что решение с такими ограничениями существует для произвольной логической функции, зависит от структуры транзисторной схемы, которая используется, и требует исследования.

Следующие понятия введены и детально описаны в статье [4] и определяют объекты, удобные для сведения задачи синтеза схем с регулярной топо-

логией к задаче декомпозиции функций на произвольных схемах.

В качестве *регулярной заготовки* рассматривается структура с выровненными полосами сильнолегированных областей p^+ - или n^+ -типа, фиксированной ширины (ширина может различаться для p^+ - и n^+ -частей) и с периодически меняющимися удлиненными и укороченными затворами. Получающиеся таким образом последовательно соединенные транзисторы формируют ряды; каждый ряд имеет инвертированный порядок p^+ - и n^+ -полос по отношению к соседнему ряду, что позволяет использовать один общий провод питания для двух соседних полос одного типа (см. рис. 1).

Транзисторный шаблон — это часть регулярной топологии с внутренней трассировкой и с отмеченными выходами и входами. Предполагается, что шаблон составляется из смежных пар транзисторов p - и n -типов. Соединение транзисторов в шаблоне получается либо посредством разделяемых участков сильнолегированных областей (для смежных транзисторов одного типа), либо посредством проводников на слое металла (в остальных случаях). В получившейся схеме отмечаются входные точки. Аналогично выделяются выходы схемы. На рис. 3 приведен пример шаблона с восемью входами и одним выходом.

Изоляционный затвор — это всегда закрытый затвор транзистора, который постоянно подсоединен к соответствующему напряжению. *Нефункциональный затвор* — это изоляционный затвор или затвор, который не подсоединен к слоям трассировки. Затвор транзистора будем называть *функциональным затвором*, если во время работы схемы к нему в разные моменты времени подводятся напряжения обоих типов (земля и питание), и это приводит к переходу транзистора из закрытого состояния в открытое, или наоборот.

Наиболее важным свойством шаблона является то, что любой транзисторный шаблон индуцирует комбинационную транзисторную схему, для которой оптимальная физическая реализация на регулярной заготовке уже определена (нет необходимости проводить размещение транзисторов и трассировать соединения).

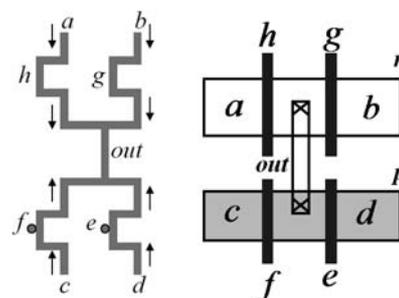


Рис. 3. Пример транзисторного шаблона с отмеченными входами и выходом

2. Алгоритм декомпозиции

Алгоритм строит решение методом динамического программирования (все множество решений выстраивается пошагово, при этом на каждом шаге поиск сужается путем откидывания заведомо неподходящих веток) и включает эвристику для выявления потенциально неподходящих решений (откидываются решения потенциально непротиворечивые, но эвристически оцененные как лишние).

Пусть требуется декомпонировать функцию $f(\vec{x}^n)$, заданную в векторном виде на комбинационной транзисторной схеме $T(i_1, i_2, \dots, i_p; o)$. При этом требуется выполнение ограничений, описанных в п. 1 (в решении должны быть сигналы, у которых характеристические функции для состояний $z, lowvcc, lowvss$ равны 0, и все функции в решении должны зависеть от меньшего числа существенных переменных).

Функциональность транзисторной схемы T определяется в виде оператора, аналогичного введенным выше для транзисторов p - и n -типов и точек соединения проводов. Все входные точки схемы рассматриваются как независимые переменные, которые могут принимать значения vcc и vss . Отсутствие состояний $z, lowvcc, lowvss$ на входных точках обусловлено наложенными выше ограничениями. Любая точка структуры, не являющаяся входной, может находиться в любом состоянии (т. е. соответствующая характеристическая функция не нулевая). Полученный оператор t осуществляет отображение из набора сигналов на входах схемы в сигнал на выходе и является суперпозицией последовательно примененных базовых операторов (рис. 4). Очевидно, что для произвольной указанной транзисторной схемы существуют формулы пересчета для каждого состояния на выходе, описывающие составной оператор. Рассмотрим только vss - и vcc -компоненты сигнала на выходе. Компонента vcc является булевой функцией от всех входных точек схемы, так как в указанных выше ограничениях на входном проводнике могут быть только состояния vcc и vss , что кодируется 1 и 0 соответственно. Далее для компоненты vcc строится множество наборов N_T , на которых эта компонента принимает значение 1. Аналогично вводится множество \bar{N}_T для vss -компоненты. Длина каждого набора равна числу входных точек:

$$N_T = \{\alpha | \alpha \in B^p, T(\alpha) = 1\};$$

$$\bar{N}_T = \{\alpha | \alpha \in B^p, T(\alpha) = 0\},$$

где B^p — двоичные наборы длины p ; $T(\alpha)$ — это отображение, которое в качестве аргумента принимает двоичный набор α значений на входах схемы. $T(\alpha)$ принимает значение 1, если на выходном проводнике схемы T индуцируется состояние vcc , и 0, если на выходном проводнике индуцируется состояние vss . Если на наборе α индуцируется на выходе

схемы T состояние $z, lowvcc, lowvss$ или bad , то значение $T(\alpha)$ считается неопределенным. Далее каждое характеристическое множество точек покрывается максимальными гранями, т. е. наборами $N', \bar{N}'_T \in \Gamma^n$, где $\Gamma = \{0, 1, -\}$. Отметим, что покрытие осуществляется именно максимальными гранями, так как их число в каждом из множеств определяет вычислительную сложность работы алгоритма (рис. 5, 6).

Дерево решений строится последовательно по уровню, начиная с пустой вершины на нулевом уровне. Вершина на i -м уровне представляет собой набор из множества N'_T либо \bar{N}'_T в зависимости от значения f на i -м наборе. Таким образом, любой путь от корня к листу в построенном дереве соответствует допустимому множеству векторов значений функций, которые необходимо получить. Вектор-столбец для i -й входной функции получается путем взятия i -й компоненты в каждой вершине построенного дерева (рис. 7).

Опишем процедуру отсекаания заведомо неподходящих ветвей в строящемся дереве. При построении

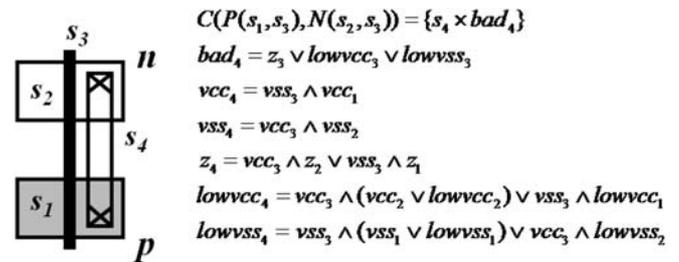


Рис. 4. Пример составного оператора: s_1, s_2, s_3, s_4 — сигналы; компоненты $vcc, vss, \dots, lowvcc$ сигналов отмечены соответствующими индексами

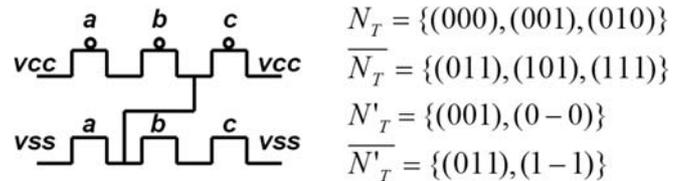


Рис. 5. Характеристические множества. Для простоты иллюстрация приведена для схемы с предопределенными константными значениями на некоторых входах (vcc, vss)

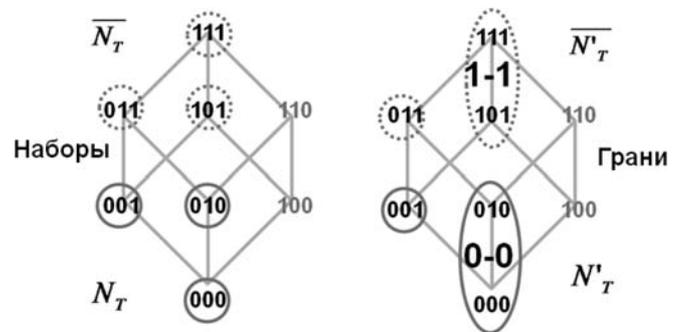


Рис. 6. Покрытие наборов $\beta \in B^3$ максимальными гранями $\gamma \in \Gamma^3$ на булевом кубе

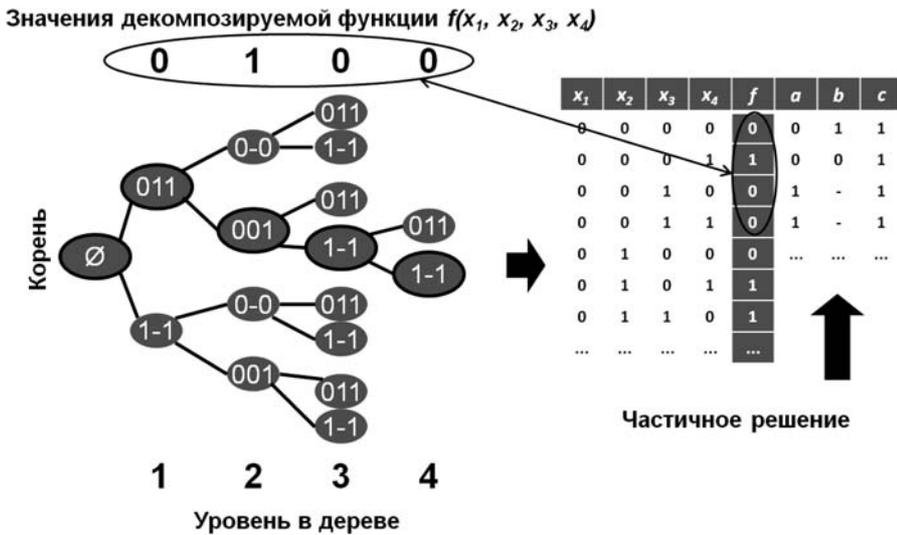


Рис. 7. Часть дерева вычислений при декомпозиции на примере функции $f = \bar{x}_1 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 x_3 \vee x_1 x_2 x_4$, выделен путь 1 в дереве и показано частично построенное решение

очередного листа выполняется проверка частично построенных вектор-столбцов, соответствующих указанному решению (пути в дереве), на предмет числа существенно зависимых переменных. Если это число превышает число существенных переменных функции f , то данная ветка не рассматривается.

Авторами предлагается быстрый способ определения множества существенных переменных для каждой из строящихся функций. Для этого используется связный список существенных переменных, выстраиваемый таким образом, что при появлении очередного листа достаточно провести одно сравнение для каждой еще независимой переменной. Ниже приведен способ быстрого вычисления номера бита k в вектор-столбце строящейся функции, значение в котором необходимо сравнить со значением в последнем построенном бите l :

$$k = l - 2^{vars - j}, \text{ если верно, что } \left\lfloor \frac{l-1}{2^{vars-j+1}} \right\rfloor \cdot 2^{vars-j+1} < l - 2^{vars-j},$$

где $vars$ — число переменных декомпозируемой функции, j — номер еще несущественной переменной; все нумерации начинаются с 1. Если условие не выполнено, то на данном шаге построения недостаточно информации для определения существенности переменной под номером j . Данная переменная считается существенной, если значения в k -м и l -м битах у анализируемой функции разные.

На примере (рис. 7) продемонстрируем работу описанной выше процедуры для выделенного пути в дереве вычислений. Здесь $vars = 4$; $j = 1, 2, 3, 4$; $l = 1, 2, \dots, 16$. При появлении узла $\{011\}$ l равняется 1 и условие-неравенство ложно для всех j . После добавления второго узла $\{001\}$, l равняется 2, условие-неравенство верно только для $j = 4$. Это

означает, что требуется сравнить значения функции на текущем втором бите со значением функции на первом бите. Действительно, для функции от четырех переменных сравнение значений на 1-м и 2-м лексикографически упорядоченных наборах необходимо для определения существенности четвертой переменной. Здесь (рис. 7) значения разные для функции b . Следовательно, в данной ветке дерева решений функция b уже точно зависит существенно от 4-й переменной. При $l = 3$ сравнение потребуется только для $j = 3$ со значением в первом бите, что для 3-й вершины $\{1-1\}$ выражается зависимостью функции a от 3-й переменной. При $l = 4$ понадобятся два сравнения для 3-й и 4-й переменных и т. д. Справедливость

формулы заключается в том, что набор под номером k отличается от l -го набора только значением в j -й позиции при выполненном условии-неравенстве. Следовательно, сравнение значений функций на данных наборах, дает возможность определить существенность переменной по определению.

Также авторами реализован подход, который позволяет контролировать весь поиск решений, начиная от решений, обладающих наилучшими характеристиками. К таковым решениям могут относиться, например, функции, которые зависят от меньшего числа переменных, или такие решения, в которых функции для входных точек на сильнолегированных областях являются константами (что соответствует подключению к цепи питания или земле) и т. д. Это достигается путем приостановки развития некоторых вычислительных веток, в которых соответствующие функции обладают плохими, по сравнению с другими ветками, характеристиками. Алгоритм выбирает то решение, которое было построено первым. Построенные вектор-столбцы для входных функций могут содержать символ "-". Эти позиции доопределяются таким образом, чтобы уменьшить число существенных переменных. На примере, приведенном на рис. 5–7, алгоритм строит следующее решение:

$$\begin{aligned} a &= \bar{x}_2 x_3 \\ b &= \bar{x}_1 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_4 \vee x_1 \bar{x}_3 x_4 \\ c &= \bar{x}_2 \vee x_2 \bar{x}_4 \\ f &= \bar{x}_1 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 x_3 \vee x_1 x_2 x_4. \end{aligned}$$

Алгоритм в силу его вычислительной сложности предполагается использовать для функций, зависящих не более чем от 6–7 логических переменных.

3. Общая схема синтеза логической функции

Общая предлагаемая схема синтеза логической функции в большей степени основана на операции декомпозиции (подход "сверху — вниз") на произвольной транзисторной подсхеме. Основываясь на результатах, описанных в работе [5], и для того, чтобы повысить качество решений, авторы предлагают также схему "снизу — вверх", которая заключается в подготовке заранее оптимальных решений для функций, зависящих от трех—четырёх переменных.

В качестве транзисторной подсхемы для декомпозиции используется понятие регулярного шаблона, введенного в п. 1 для фиксированной регулярной топологии. Для экспериментов авторы использовали регулярную топологию, изображенную на рис. 1. Перебором рассматривали части регулярной топологии, которые вместе с доопределенной внутренней трассировкой и различными вариантами выбора входных/выходных точек формируют шаблоны по определению. Таким образом строятся все регулярные шаблоны для данной регулярной заготовки, которые содержат не более восьми транзисторов. Данное множество шаблонов определяет набор доступных для декомпозиции транзисторных схем.

Все схемы, индуцированные регулярными шаблонами, упорядочены по возрастанию числа транзисторов. Алгоритм выбирает лучшее решение по числу существенных переменных, от которых зависят полученные функции, декомпозируя функцию на схемах в указанном порядке. Общая схема алгоритма проиллюстрирована на рис. 8. Процедура логического синтеза организована циклически. Для входной функции применяется операция декомпозиции, если оптимальная реализация, заранее заготовленная на этапе "снизу — вверх", отсутствует, иначе реализация берется из заготовленного списка. Далее операция повторяется для полученных функций на входных точках.

Алгоритмы проектирования на этапах физического синтеза, а также физические характеристики схемы (быстродействие и трассируемость) зависят в большей степени от того, в какие точки данного регулярного шаблона подключены выходы топологических конструкций, используемые для получения самих входных сигналов. Такие ограничения выставляются отдельно для проводников, подключаемых к затворам, и отдельно для проводников, подключаемых к сильнолегированным областям p^+ - или n^+ -типа. Например,

так как существует ограничение на максимальную длину цепочек последовательно соединенных транзисторов от источника питания до затвора транзистора, имеет смысл подключить к истокам транзисторов функции, которые можно реализовать, не используя длинные цепочки последовательно соединенных транзисторов, так как иначе данная цепочка еще больше увеличится. Поэтому актуально по-разному ограничивать типы сигналов, которые используются при синтезе. Для описания ограничений вводятся следующие определения.

Диффузионными сигналами будем называть сигналы, выставленные на входные вершины транзисторов схемы, являющиеся истоками транзисторов. Аналогично, *затворными сигналами* будем называть сигналы, выставленные на входные точки, являющиеся затворами транзисторов.

Сигнал называется *константой*, если его компоненты имеют вид

$$g_{vcc} = 1, g_{vss} = 0, g_z = 0, g_{lowvcc} = 0, g_{lowvss} = 0$$

(константа **1**);

$$g_{vcc} = 0, g_{vss} = 1, g_z = 0, g_{lowvcc} = 0, g_{lowvss} = 0$$

(константа **0**).

Сигнал называется *переменной*, если его компоненты имеют вид

$$g_{vcc} = x, g_{vss} = \bar{x}, g_z = 0, g_{lowvcc} = 0, g_{lowvss} = 0$$

(переменная **x**).

```

процедура логический_синтез (
  f (декомпозируемая функция),
  T (множество транзисторных шаблонов),
  L (список заготовленных реализаций) {
    Impls - множество реализаций для функций;
    S - стек для функций;
    f добавить в S;

    пока S непустой {
      cf := текущая функция с головы S;
      удалить cf из S;

      если реализация на шаблонах L(cf) для функции cf существует в L,
      то добавить реализацию L(cf) в Impls;
      иначе {
        реализация best_I := 0;
        множество функций newF := 0;
        множество функций для лучшей реализации best_newF := 0;
        для каждого транзисторного шаблона t из T {
          I := декомпозиция(f, t, newF); // newF - множество функций в решении
          если I «лучше» best_I, то {
            best_I := I;
            best_newF := newF;
          }
        }

        добавить best_I в Impls;
        добавить все функции из best_newF, зависящие > 1 переменной в S;
      }
    }
  }

```

Рис. 8. Общая схема алгоритма логического синтеза на регулярных шаблонах

Пусть \tilde{T} — некоторое множество комбинационных транзисторных схем и \tilde{S} — некоторое множество сигналов. Под перебором "снизу — вверх" по множеству \tilde{T} и множеству \tilde{S} понимается всевозможные подключения любого подмножества сигналов из \tilde{S} к входам любой схемы из \tilde{T} . Далее последовательно рассматриваются пары таких множеств для перебора. Каждая пара формирует класс.

Класс № 1. В качестве множества \tilde{T} используются конструкции, составленные из одного шаблона, а в качестве \tilde{S} — только константы и переменные. В результате построены множество F_1 — сигналы-функции, которые реализуются на выходе при переборе в данном классе, и множество S_1 — сигналы, не являющиеся функциями. Пример с некоторым шаблоном T_1 для реализации некоторой логической функции при переборе в рамках данного класса представлен на рис. 9.

Класс № 2. Множество конструкций \tilde{T} составляется из двух шаблонов, выход первого шаблона T_1

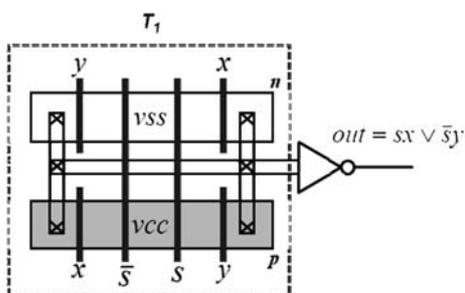


Рис. 9. Конструкция из класса № 1. Пример реализации мультиплексорной функции на регулярном шаблоне

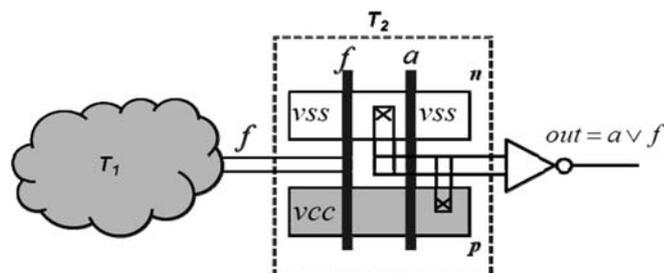


Рис. 10. Конструкция из класса № 2. Выход T_1 подключен к входу T_2

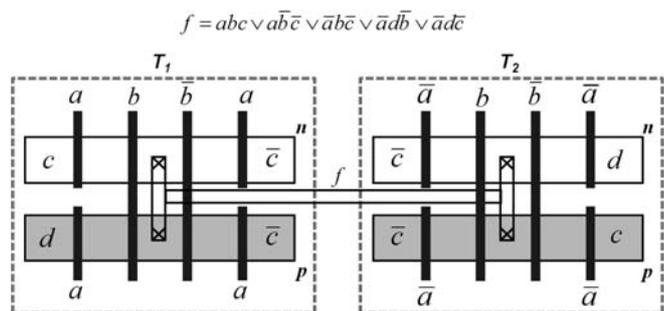


Рис. 11. Конструкция из класса № 3. T_1 и T_2 имеют общий выходной провод

подключается к некоторому числу входов второго шаблона T_2 . В качестве множества \tilde{S} используются только константы и переменные. При этом для входов T_1 выбираются только такие комбинации сигналов, которые приводят к реализации на выходе T_1 некоторой функции из F_1 , т. е. на некоторых входах T_2 реализуется функция из множества F_1 , а на остальных используются переменные и константы. При переборе в данном классе строится множество возможных функций F_2 на выходе T_2 (рис. 10).

Класс № 3. \tilde{T} — конструкции, составленные из двух шаблонов, соединенных общим выходным проводником. В качестве множества \tilde{S} используются только константы и переменные. Отметим, что подобного рода конструкция — соединенные по выходу два функциональных элемента — не имеет смысла для классического подхода на стандартных ячейках, что расширяет операцию суперпозиции для регулярных шаблонов. На входах каждого из шаблонов могут использоваться те же типы сигналов, что и в классе № 1, но при этом необходимо, чтобы на выходе каждого из этих шаблонов, рассмотренных в отдельности, реализовывался сигнал из S_1 . При этом рассматриваются только те варианты, которые приводят к появлению функций на выходе. Такие функции формируют множество F_3 (рис. 11).

Класс № 4. \tilde{T} — конструкции, составленные из трех шаблонов (T_1, T_2, T_3), в которых выходы T_1 и T_2 подключаются к входам T_3 . К входам T_1, T_2 подключаются переменные или константы. На входах T_1, T_2 рассматриваются только такие комбинации, которые приводят к реализации функций из F_1 на выходе. Выходные функции T_1, T_2 используются только в качестве затворных сигналов для T_3 . В качестве диффузионных сигналов на шаблоне T_3 используются только константы и переменные. Такое разделение на типы сигналов вызвано ограничением на длину цепей последовательно соединенных транзисторов, описанным выше, так как при подключении выхода шаблона к истоку транзистора в следующем шаблоне может возникнуть цепь из последовательно соединенных транзисторов, превышающая максимальную длину. Перебором в данном классе строится множество функций F_4 (рис. 12). На рис. 13 показан пример того, что на шаблоне могут возникать состояния *bad* при подключении независимых переменных к входам, в то время как существуют пары функций, которые не являются полностью независимыми и которые могут быть подключены к данной структуре (см. рис. 12). Этот пример иллюстрирует расширенную в сравнении со стандартными ячейками операцию суперпозиции, которая используется при синтезе на регулярных шаблонах.

Таким образом, множества F_1, F_2, F_3 и F_4 вместе с соответствующими конфигурациями образуют

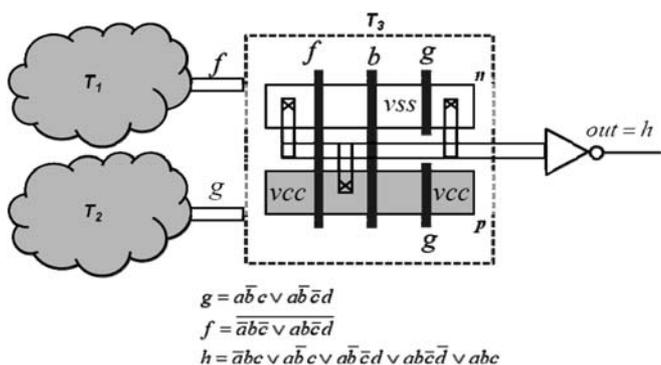


Рис. 12. Конструкция из класса № 4. К шаблону T_3 подключены выходы T_1, T_2

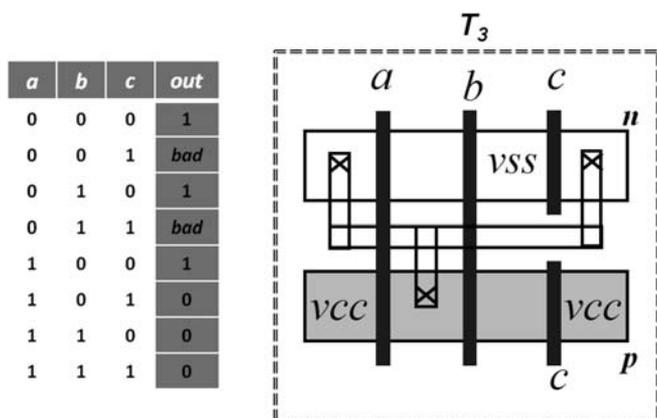


Рис. 13. Пример регулярного шаблона с таблицей истинности. Состояния *bad* возникают при подключении независимых входных переменных

список заранее подготовленных реализаций логических функций на шаблонах. Выявлено, что таким последовательным перебором "снизу — вверх", включая возможность вставки инвертера на проводник, соединяющий шаблоны, можно получить реализации для ~75 % всех функций, зависящих существенно не более чем от четырех переменных.

Важно отметить, что наличие реализации мультиплексорной функции (см. рис. 9) на шаблоне, обеспечивает сходимость всего процесса, что следует из разложения Шеннона.

4. Экспериментальные результаты

Проведено сравнение по площади после этапа логического синтеза на стандартных ячейках и на транзисторных шаблонах. В качестве входных задач для синтеза рассматривались логические блоки с 4—6 входами (логические переменные) и одним выходом (логическая функция). Выборка включает 17 000 различных функций с точностью до перестановки входных переменных. Занимаемая площадь оценивалась числом затворов как функциональных, так и нефункциональных.

Процедура синтеза на стандартных ячейках осуществлялась промышленными программами, ми-

нимизирующими площадь кристалла. Использовалась стандартная библиотека ячеек, включающая 25 различных логических функций с 2—5 входами, включая такие функциональные элементы как NAND, NOR, AOI/OAI (And-Or-Inverter/Or-And-Inverter), XOR (сумма по модулю 2), MUX (мультиплексор) и др.

Для того чтобы обеспечить более качественные электрические характеристики, требовалось получить топологию, в которой каждый входной проводник подсоединялся только к затворам транзисторов (не к истокам) и в которой отсутствуют цепи последовательно соединенных транзисторов с длиной более 3. Для получения оценки сверху на число транзисторов не использовалась операция конкатенации шаблонов, подробно описанная в работах [4, 5], которая потенциально способна уменьшить число нефункциональных транзисторов на этапах физического синтеза.

В п. 1 таблицы для каждой группы затворов: функциональные (Ф), нефункциональные (Н) и затворы всех типов (Ф + Н), указано среднее число транзисторов, полученное в результате логического синтеза на стандартных ячейках и на регулярных шаблонах по всей выборке. Таким образом, используя регулярные шаблоны для синтеза блоков с 4—6 входами, в среднем можно уменьшить площадь более чем на 10 % по сравнению с синтезом на стандартных ячейках.

В п. 2 таблицы показан процент логических блоков, для реализации которых требуется меньшее число затворов каждого типа. Наличие блоков (18 %), имеющих меньшее число транзисторов при проектировании на стандартных ячейках, объясняется ограниченным разнообразием типов графов соединений для регулярных шаблонов. Описанным алгоритмом чаще строятся древовидные структуры, что исключает возможность переиспользования эквивалентных логических точек. Чтобы нивелировать данный эффект при сравнении результатов, авторы проделали следующий эксперимент. Если при сравнении T_1 — реализации на регулярных шаб-

Сравнение по числу функциональных (Ф) и нефункциональных (Н) транзисторов, а также транзисторов любого типа (Ф + Н) при проектировании на стандартных ячейках и регулярных шаблонах

№ п/п	Характеристики сравнения	Ф	Н	Ф+Н	Ф	Н	Ф+Н
		Стандартные ячейки			Регулярные шаблоны		
1	Среднее число транзисторов для логического блока	23,7	25,1	48,8	21,7	21,7	43,4
2	Задачи с меньшим числом транзисторов указанного типа, %	17	14	18	64	68	75

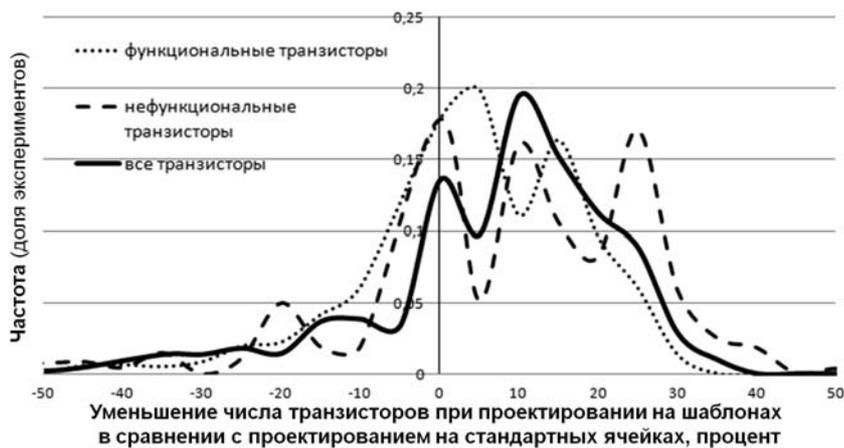


Рис. 14. Гистограмма, показывающая уменьшение числа транзисторов в процентах



Рис. 15. Гистограмма, показывающая уменьшение числа транзисторов

лонах с T_2 — реализацией на стандартных ячейках число транзисторов в T_1 было больше, чем в T_2 , то в качестве T_1 использовалась T_2 , но с заменой каждой стандартной ячейки на ее реализацию на регулярном шаблоне. Так как любая стандартная ячейка из библиотеки может быть реализована на регулярных шаблонах с не более чем таким же числом транзисторов (экспериментально установлено), то в построенной таким образом T_1 всегда число транзисторов будет не превосходить число транзисторов в T_2 . При таком подходе использование регулярных шаблонов уменьшает площадь в среднем на 17 % по сравнению с синтезом на стандартных ячейках.

Детальное описание результатов сравнения представлено на рис. 14, 15. Здесь иллюстрируется уменьшение числа транзисторов при проектировании на регулярных шаблонах в сравнении с проектированием на стандартных ячейках без использования описанной выше процедуры подмены графов соединений. Отсутствие монотонности для нефункциональных транзисторов и для всех транзисторов на рис. 15, а также наличие локальных экстремумов объясняются тем, что число нефункциональ-

ных транзисторов, как правило, кратно 4, так как каждый регулярный шаблон и стандартная ячейка ограничены 4 изоляционными затворами. Соответственно, разница (увеличение или уменьшение в площади) также кратно 4. При этом остаются некоторые случаи, которые вносят разницу, не кратную 4, в общую статистику. Такие ситуации случаются, если реализация стандартной ячейки включает дополнительные нефункциональные затворы, которые используются для внутренней изоляции. Из графиков следует, что 10–20 % уменьшения по площади являются наиболее ожидаемыми при использовании регулярных шаблонов.

Заключение

Представленный подход выгодно отличается от способов построения ИС, в которых используются отдельные транзисторы в качестве минимальных функциональных элементов (Sea-of-transistors), так как ряд задач, которые решаются на этапе физического синтеза, значительно упрощаются. Например, задача размещения произвольного графа транзисторных соединений на кристалле по высокотехнологичным правилам проектирования сводится к размещению уже заготовленных регулярных транзисторных шаблонов, имеющих готовое размещение внутренних транзисторов. Подобную аналогию также можно сформулировать и для задачи трассировки, так как регулярный шаблон имеет внутреннюю трассировку на транзисторном уровне, и задача сводится к трассировке меньшего числа точек по сравнению с подходами Sea-of-transistors.

При переходе на новые технологические процессы набор правил проектирования все больше сужает пространство возможных решений на физическом уровне для стандартной ячейки. Проектировщики все чаще используют максимально автоматизированные средства проектирования библиотек ячеек. Появляется возможность проектировать более сложные элементы, чем стандартные ячейки, автоматически и достаточно оптимально. В этих условиях регулярный транзисторный шаблон, как минимальный функциональный элемент, дает возможность расширить пространство возможных топологий на транзисторном уровне и тем самым получить более качественное решение как с точки зрения регулярности, так и с точки зрения занимаемой площади. Проведенные эксперименты

показали, что использование регулярных шаблонов при логическом синтезе блоков с шестью или менее входами способно сократить занимаемую площадь на 10–15 % по сравнению с синтезом на стандартных ячейках.

В настоящее время авторы исследуют электрические характеристики получаемых схем, а также развивают общие идеи и алгоритмы для физического синтеза логических функций на регулярных шаблонах. Также ведутся разработки в построении полной схемы логического и физического синтеза для ИС общего вида.

Список литературы

1. **Kuon I., Rose J., Rogers S.** Measuring the Gap Between FPGAs and ASICs // Proc. of FPGA. P. 23–30. 2006.
2. **Ran Y., Marek-Sadowska M.** An integrated Design Flow for a Via-Configurable Gate Array // Proc. of DAC. P. 582–589. 2004.
3. **Талалай М., Трушин К.** Обзор подходов к проектированию регулярных ИС // Информационные технологии. 2011. № 1. С. 2–7.
4. **Талалай М., Трушин К., Венгер О.** Логический синтез комбинационных схем на основе транзисторных шаблонов с регулярной топологией // Информационные технологии. 2011. № 4. С. 2–7.
5. **Talalay M., Trushin K., Venger O.** Between Standard Cells and Transistors: Layout Templates For Regular Fabrics // Proc. of EWDTS. P. 293–299. 2009.

УДК 621.38

П. Н. Бибилло, д-р техн. наук, проф.,
e-mail: bibilo@newman.bas-net.by,

Объединенный институт проблем информатики
Национальной академии наук Беларуси,
г. Минск

Экспертные знания в САПР цифровых устройств

Рассматривается проблема разработки экспертных систем автоматизированного проектирования (САПР) на примере проблемной области логического проектирования цифровых устройств. Анализируется организация систем проектирования, предлагается определение понятия "экспертные знания", сообщается о модели представления экспертных знаний.

Ключевые слова: цифровые устройства, САПР, экспертные знания, продукции, фреймы

Введение

Стремительное развитие микроэлектроники позволяет создавать сверхбольшие (СБИС), ультрабольшие (УБИС) схемы и системы на кристалле огромной функциональной сложности [1]. Повышение степени интеграции обуславливает совершенствование соответствующих систем автоматизированного проектирования на всех этапах — от архитектурного и алгоритмического этапов, до этапа проектирования топологии кристаллов. Центральным этапом является этап логического проектирования, результатом выполнения которого являются логические схемы в определенных технологических базисах и на котором определяются во многом основные параметры реализуемого проекта — сложность (площадь), быстродействие, энергопотреб-

ление, тестопригодность. При этом наблюдается следующая тенденция: эффективность систем сквозного проектирования возрастает, однако "внутренняя кухня" обычно скрывается от пользователей. Пользователям обычно предлагается один маршрут проектирования, а выбор решений, принимаемых при реализации проекта, не объясняется, а лишь протоколируется.

Однако для создания эффективных САПР требуется разработка комбинированных методов и маршрутов проектирования, их экспериментальная оценка и выбор "статистически лучших" для включения их в сквозные САПР. Кроме того, наличие конкурирующих САПР позволяет развивать методологию "горизонтальных" САПР, когда важно не только получать логические схемы в заданных технологических базисах, но и обеспечивать возможно лучшие проектные решения по указанным выше основным параметрам сложности, быстродействия и т. д. При горизонтальном проектировании важно не только унифицировать обмен данными между различными системами и подсистемами проектирования, но и уметь принимать рациональные (эффективные) решения при управлении процессами проектирования. В этой части незаменимыми являются знания экспертов — высококлассных специалистов проектировщиков и разработчиков систем проектирования. Использование экспертных знаний может значительно повысить эффективность проектирования, однако для этого требуется разработка соответствующих формализмов представления (описания) знаний и средств использования таких описаний в процессе проектирования.

В данной статье анализируется область логического проектирования цифровых устройств, выясняется назначение и роль экспертных знаний, сообщается о предлагаемом формализме их описания в САПР.

Задачи логического проектирования цифровых устройств

Дискретными называются устройства, функционирующие в дискретном времени и осуществляющие преобразование дискретной информации. Дискретная информация чаще всего представляется в цифровой (двоичной — 0, 1) форме, поэтому часто дискретные устройства называют также *цифровыми*. Объектами логического проектирования являются функционально-структурные схемы дискретных устройств, а основной задачей — синтез, т. е. получение по функционально-структурному (функциональному, структурному) описанию соответствующей логической схемы в заданном технологическом базисе [2].

Задача синтеза логической схемы формулируется следующим образом. Дано функциональное описание либо описание поведения будущей схемы и библиотека логических элементов (целевая библиотека синтеза, базис синтеза). Требуется построить логическую схему из элементов библиотеки, поведение (функционирование) которой эквивалентно поведению исходного описания. Трудность решения задачи синтеза заключается в том, что требуется найти оптимальную по тем или иным критериям (площадь кристалла, быстродействие, энергопотребление) схему.

Технологическим базисом синтеза могут выступать элементы, каждый из которых реализуется одним корпусом (микросхемой), либо элементы библиотеки интегральной технологии, когда схема реализуется в одном корпусе, например, в составе заказной, полузаказной, программируемой СБИС. В зависимости от технологического базиса критерии оптимизации схем довольно сильно различаются. Заметим, что рассматривается этап именно логического проектирования, когда после этапа алгоритмического проектирования исходными данными являются функционально-структурные, в том числе и иерархические, описания, а результирующими данными — логические схемы в технологических базисах.

На практике нужны не просто логические схемы, обеспечивающие нужное поведение, эквивалентное поведению исходного описания проектируемой цифровой системы, а схемы, оптимизированные по тем или иным критериям. Задачи получения оптимальной по сложности схемы даже в классе комбинационных логических схем, математическими моделями которых являются системы булевых функций, характеризуются комбинаторным ростом сложности вычислений. Часто говорят также о "комбинаторном взрыве" сложности вычислений [3], а задачи характеризуют как NP-трудные [4]. В связи с масштабностью решаемых задач в САПР речь о точных решениях, по существу, не идет, на практике используются приближенные (эвристические) алгоритмы, позволяющие решать задачи огромной

размерности, когда число логических элементов исчисляется миллионами. Совокупность правил выбора приемлемых в конкретной проектной ситуации эвристик является одной из компонент экспертных знаний. При технологическом отображении выбор соответствующей целевой технологической библиотеки является сложной задачей. Чаще всего такой выбор приходится делать проектировщику. Правила принятия подобных решений — еще один компонент знаний эксперта в области логического проектирования.

Задачами моделирования являются: вычисление реакции схемы на заданное входное воздействие (тест), поиск входного сигнала, вызывающего заданную реакцию, и др. Для синтезированных логических схем основная задача анализа — вычисление реакций схемы на заданные входные воздействия — решается в системах моделирования, например, в системе ModelSim с входным языком VHDL [5]. Заметим, что при решении задач моделирования исходное описание схемы не изменяется — моделируемая схема является "статичной".

Задача анализа логической схемы состоит в вычислении функций, которые реализует логическая схема. Если схема является комбинационной, то получаемые функции являются полностью определенными булевыми функциями. Если схема содержит элементы памяти, то под задачей анализа схемы с памятью понимается чаще всего вычисление функций комбинационной части схемы.

Задача верификации логических схем формулируется следующим образом. Даны две логические схемы. Требуется проверить, являются ли они функционально эквивалентными. В настоящее время разработка средств формальной верификации алгоритмических описаний логических схем представляет собой одну из важнейших проблем в проектировании.

Известны также *задачи построения тестов* для нахождения и диагностики неисправностей дискретных устройств [6].

Далее основное внимание будет уделено синтезу — получению структурного описания объекта проектирования. Задача получения структурного описания объекта проектирования и соответствующих характеристик является основной задачей САПР в различных проблемных областях. Уменьшение технологических норм приводит к необходимости учета на этапе логического проектирования топологических характеристик — длин соединений и площадей подсхем и элементов, а также решения проблем тестирования и снижения энергопотребления. Поэтому все чаще классические постановки задач логического проектирования увязываются воедино с постановками задач снижения энергопотребления, решаемых на всех этапах проектирования, задач тестирования, размещения элементов и трассировки соединений [7].

Организация систем логического проектирования

Развитие САПР в области логического проектирования цифровых устройств можно условно поделить на два больших периода.

Первый (начальный) период характеризуется отсутствием унифицированных языковых средств описания исходных данных для проектирования и отсутствием промышленных систем логического проектирования, которые были бы интегрированы в сквозные системы. Успехи микроэлектроники позволили создавать элементную базу — цифровые интегральные схемы — в виде БИС и СБИС. Из-за резкого увеличения сложности проектов ручное проектирование стало практически невозможным, появилась острая необходимость в унификации языковых средств и создании сквозных систем проектирования. Потребность в сквозных САПР еще более обострилась с появлением технологии производства программируемых логических схем, особую популярность среди которых приобрели FPGA.

Второй период развития САПР цифровых устройств характеризуется появлением таких стандартных языков описания поведения и структур дискретных устройств (цифровых систем), как VHDL, Verilog, и созданием сквозных промышленных (коммерческих) систем проектирования ПЛИС [8] и заказных СБИС. Системы проектирования заказных кристаллов большой сложности — микропроцессоров — являются технологическими секретами (собственностью) фирм — изготовителей кристаллов и не предназначаются для распространения, в отличие от доступных САПР ПЛИС. Следует сразу отметить, что устройство большинства систем проектирования ПЛИС типа FPGA и CPLD пользователям не открывается, примерами тому являются синтезатор XST системы проектирования WebPack ISE [8] и Synplify Pro [9]. Пользователи WebPack ISE и Synplify Pro могут только выбрать тип кристалла и, если это возможно, реализовать на нем свой проект, в противном случае приходится выбирать кристалл другого семейства. В отличие от таких САПР ПЛИС, синтезатор LeonardoSpectrum [5] предназначен для проектирования структур FPGA и CPLD, однако позволяет получать схемы в библиотеке логических элементов, сформированной непосредственно самим проектировщиком. Такие схемы могут быть реализованы в составе заказных СБИС.

Рассмотрим кратко эволюцию *языковых средств* описания цифровых устройств.

Первый период развития САПР цифровых устройств характеризуется большим разнообразием языков для описания классических формальных моделей дискретных устройств, к которым относятся в первую очередь булевы функции и конечные автоматы. Булевы функции представлялись в матричных либо скобочных формах, последние иногда называют логическими уравнениями. Пред-

ставления конечных автоматов также имели различную форму — это абстрактные автоматы, когда состояния их не закодированы, и структурные описания автоматов. Были предложены также самые разнообразные модификации автоматных описаний — микропрограммные автоматы, заданные граф-схемами алгоритмов (ГСА), секвенциальные автоматы и т. д.

Итак, базовыми языками исходного описания поведения и структур цифровых устройств и систем во *втором периоде* развития САПР в области логического проектирования являются языки VHDL и Verilog [10]. На языке VHDL могут быть представлены самые различные формальные модели булевых и конечнозначных функций, автоматов, параллельных процессов, регулярных схем и т. д. Исходное описание для синтеза должно быть проведено на специально выделенном подмножестве языка, называемом *синтезируемым подмножеством*. Только те описания, которые представлены конструкциями из такого подмножества, могут быть превращены синтезатором в логические схемы. Подобные возможности имеет и язык Verilog; о различиях языков VHDL, Verilog можно узнать в книге [9], в которой проводится подробное сравнение конструкций данных языков. Предпринимаются попытки создания и использования других языков, в первую очередь языка SystemC, прообразом которого являются языки C, C++, имеющие большую популярность при создании не аппаратных, а программных средств. Развиваются также новые языки типа PSL (Property Specification Language) [11] для целей верификации описаний цифровых систем, записанных на VHDL и Verilog, а также языки для решения задач на системном уровне проектирования цифровых СБИС и УБИС. Проблема функциональной верификации исходных спецификаций выдвигается на передний план несмотря на необходимость решения задач огромной размерности на нижних уровнях проектирования.

Результаты логического проектирования — логические схемы — представляются часто с использованием тех же языков, на которых представлены исходные для синтеза. Это связано с тем, что после синтеза обычно выполняется моделирование схем и сравнение их поведения с поведением исходных спецификаций, а для этого целесообразно использовать те же тестовые программы и тесты (наборы значений входных сигналов), которые были разработаны на этапе алгоритмического проектирования.

Организация САПР. Разработанные во время первого периода системы логического проектирования дискретных устройств обычно имеют развитые иерархически устроенные меню. Инициатива диалога принадлежит проектировщику, выполняющему последовательные преобразования описаний объектов проектирования. В случае возможности выполнения альтернативных действий промежу-

точные объекты сохраняются, поэтому может появиться много таких объектов, и пользователь должен помнить, какой именно объект получен в результате тех или иных действий.

Разработанные САПР представляли собой диалоговые системы, состоящие из модулей проектных операций и оболочки — монитора верхнего уровня, который осуществляет управление вычислительным процессом и поддерживает необходимый сервис. Если каждому программному модулю, выполняющему то или иное проектное действие, поставить в соответствие вершину графа, а ориентированные дуги пометить типами обрабатываемых объектов, то можно представлять процесс проектирования как движение фазовой точки по вершинам графа. Фазовая точка указывает объект проектирования, а вершины графа (модули) совершают преобразования объектов. Во многих системах логического проектирования, например, в системах программируемой логики ABEL [12], PLDshell Plus, CUPPL и др. принята файловая организация, где расширение имени файла указывает на тип объекта. Подсказка системы об очередных действиях пользователя сводится к информации о возможности перехода в то или иное состояние системы. Выбор самой системой конкретной операции (в случае наличия альтернативных) не предусматривается.

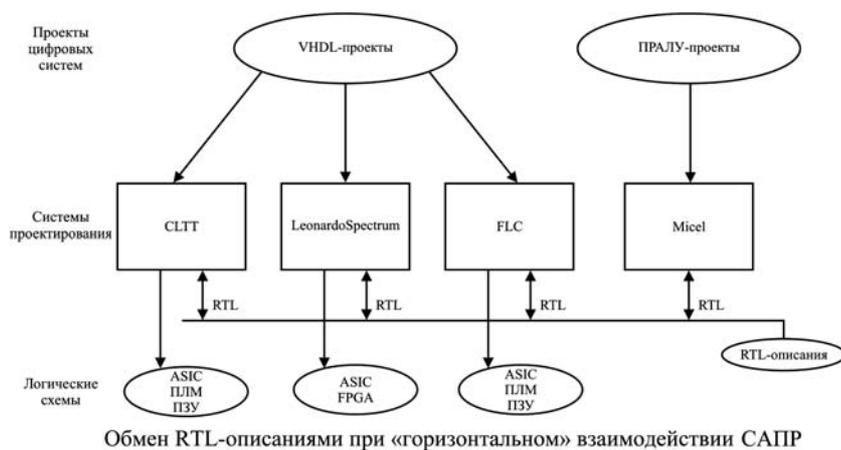
Внести ошибки в проект достаточно трудно — системы устроены так, что проект будет либо реализован правильно, либо не реализован вообще. Качество проекта зависит от качества решений, принятых проектировщиком. Однако тенденция исключить ручное проектирование с развитием САПР только усиливается. Это объясняется требованиями надежности проекта и трудностями верификации действий пользователя: при "ручном" вмешательстве в описания промежуточных объектов проектирования требуется решать задачу верификации — определения эквивалентности исход-

ного и скорректированного описания. Поэтому чаще всего допускается "ручное" изменение только исходных данных, технологических базисов, технологических библиотек.

Выбор того или иного проектного действия часто мотивируется "историей" сеанса проектирования, поэтому САПР снабжаются средствами ведения протокола сеанса работы; например, последовательно записываются в протокол сеанса имена обрабатываемых данных (проектов, схем) и имена обработавших программных модулей.

Во время первого периода предпринимались попытки использования методов искусственного интеллекта в логическом проектировании, чаще всего, пожалуй, при синтезе схем. В обзоре [13] сообщается о системе SOCRATES, в которой на верхних уровнях проектирования используются алгоритмические методы (минимизация систем функций в классе ДНФ, факторизация алгебраических выражений и др.), а при решении задачи технологического отображения (покрытия схемы элементами из библиотеки проектирования) используются правила. Полное описание базы знаний в данной работе, как, впрочем, и других работах, не приводится. Правила вида "если — то" предлагается [14] использовать для выбора вида декомпозиции булевых функций в зависимости от базиса синтеза и критерия оптимизации при решении многокритериальных задач синтеза. Таким образом, разработанные экспертные системы логического проектирования ориентированы в основном на компилятивное проектирование на этапе технологического отображения. В таких системах логический вывод (возможность применения правила) связан с комбинаторным поиском по базе данных. Роль эксперта при создании таких систем сводится к оформлению правил замены одних подсхем другими.

Функциональное назначение систем проектирования для второго периода развития САПР дискретных устройств представлено в различных публикациях [5, 6, 8 и др.]. Пользователям даются подробные описания, скриптовых команд, однако внутреннее (системное) устройство синтезаторов, верификаторов и систем моделирования обычно не раскрывается. Для оптимизации тех или иных параметров проекта все чаще организуется "горизонтальное" взаимодействие систем логического проектирования. На рисунке показано, как RTL-описания языка VHDL были успешно применены для обмена данными между отечественными САПР CLLT, MiCel, FLC [16] и промышленным синтезатором LeonardoSpectrum [5] (фирма Mentor Graphics).



Обмен RTL-описаниями при «горизонтальном» взаимодействии САПР

Обмен RTL-описаниями при "горизонтальном" взаимодействии САПР

Особенности логического проектирования

Кроме логико-комбинаторной природы задач логического проектирования их отличительными особенностями также являются: сравнительно небольшой объем памяти под исходные и результирующие данные, значительный объем памяти под промежуточные вычисления и большой набор формальных методов проектирования, базирующихся на различных моделях представления данных.

В отличие от других областей автоматизированного проектирования, например проектирования машиностроительных конструкций [16], объекты логического проектирования обладают *однородностью* — все комбинационные блоки проекта представляются различными формами задания булевых функций, так как функциональные описания триггеров, представляющие блоки памяти, не изменяются на этапе логического проектирования, следующем после этапа высокоуровневого проектирования.

Основные трудности в логическом проектировании доставляют: *большая размерность и связность* данных (взаимосвязи различных булевых переменных). Эти связи нельзя разрывать, но их можно *дублировать*, дублируя соответствующие логические элементы или подсхемы, либо *элиминировать* — устранять.

Важной особенностью логического синтеза является широкая возможность самых разнообразных *преобразований структуры* проекта, в отличие от проектирования машиностроительных конструкций. В машиностроительных конструкциях структура проектируемого объекта часто строго фиксирована и имеются значительные ограничения (в том числе и пространственные — геометрические) на соединения частей (деталей) конструкции [16]. Заметим, что при схемотехническом и топологическом проектировании заказных СБИС исчезают границы описаний на уровне логических элементов, вся логическая схема рассматривается как сеть из огромного числа транзисторов, число элементов и связей транзисторов только увеличивается по сравнению с логическим проектированием (там рассматриваются более "крупные" логические элементы), и эти связи, естественно, нельзя разрывать. Возможны различные подходы к соединению транзисторов: например, в макроблоке программируемой логической матрицы (ПЛМ) используются параллельные соединения транзисторов, при этом топологическая реализация ПЛМ из-за регулярности структуры упрощается, однако за это приходится "расплачиваться" площадью кристалла, так как топологическая упаковка нерегулярных схем из библиотечных элементов чаще всего дает выигрыш в площади по сравнению с площадью разреженных ПЛМ. Разреженность понимается как отсутствие транзисторов — точек коммутации вертикальных шин с горизонтальными. Таким образом, в логическом проектировании принципиальными явля-

ются другие трудности — это трудности решения оптимизационных задач большой размерности, результаты решения которых определяют характеристики проекта.

Проведя анализ промышленных систем проектирования цифровых устройств, можно заключить, что *областью знаний экспертов* являются знания по организации маршрутов проектирования в зависимости от целей проектирования, технологического базиса и характеристик реализуемого проекта. В синтезаторе LeonardoSpectrum *экспертными знаниями* можно считать последовательности скриптовых команд, позволяющих управлять процессами синтеза. Примеры таких последовательностей для синтеза логических схем приведены в работе [5]. В системе моделирования ModelSim [5] имеются также средства выполнения последовательностей команд, позволяющих управлять процессами моделирования. Для описания команд используется язык Tcl [17].

Данные и знания в логическом проектировании

Исходными данными являются иерархические описания логических схем, описания технологических базисов и параметры, требуемые для работы программных модулей, решающих конкретные задачи обработки иерархических описаний проекта и представлений листов проекта. Результирующими данными при этом считаются логические схемы в различных технологических базисах, если решаются задачи синтеза. Если же решаются задачи анализа, то исходными данными считаются логические схемы, а результирующими — функции, реализуемые схемами. Результирующими данными после решения задачи верификации двух логических схем либо верификации (сравнения) исходного описания с синтезированной схемой могут быть ответы "Да", если два описания эквивалентны по поведению, и "Нет", когда они не эквивалентны. В последнем случае могут быть предъявлены наборы значений входных сигналов, при которых значения выходных сигналов различаются.

Как уже говорилось, теория логического проектирования хорошо развита, на практике созданы промышленные системы синтеза, моделирования и верификации цифровых схем. Так как реализуемые логические функции очень разнообразны, то решение задачи синтеза одним методом может быть эффективным для задач из одного класса и неэффективным для задач из другого класса. Дальнейший прогресс и развитие логического проектирования будет происходить за счет организации эффективного взаимодействия различных методов и, естественно, реализующих эти методы программ. Это приводит к усложнению организации систем проектирования.

Общий взгляд на задачи логического проектирования для иерархических проектов состоит в том,

что при оптимизации, синтезе и верификации проектов могут изменяться "объемы" описаний отдельных переменных проекта (объемы гранул листовых описаний), вид деревьев иерархии описаний, способы обработки листовых описаний в зависимости от размерности проекта в целом, от размерностей листовых описаний и от уровня (функционального объема) гранул базисных элементов [15].

Принятие решений по выбору представлений и выбору уровней гранулярности листовых описаний, изменению дерева иерархии описания, т. е. по методам слияния и разбиения узлов и поддеревьев, является областью действий проектировщика — пользователя САПР при работе с конкретным проектом. Проектировщик устанавливает параметры, требуемые для работы программных модулей, принимает решения по выбору технологического базиса для блоков проекта, выполняет проектные операции и процедуры оптимизации и декомпозиции проекта, анализирует данные, оценивает получаемые результаты и т. д. В результате действий проектировщика в конкретных ситуациях получается последовательность проектных операций и процедур, приводящая к реализации исходных спецификаций — превращению исходного описания в результирующую логическую схему.

Знания проектировщика выражаются в способности принимать рациональные (эффективные) решения при управлении системой проектирования для достижения целей проектирования. При разработке САПР, содержащей средства интеллектуальной поддержки, предварительному анализу подвергается область действий проектировщика. Этот анализ опирается на понятие *проектной ситуации*, куда входят цели проектирования, параметры используемых в САПР описаний объектов и базисов проектирования и параметры программных модулей. Целью анализа при этом являются экспертные знания. Такие знания требуется выявить, формализовать и представить в терминах проблемной области, в нашем случае — в области проектирования функционально-логических схем.

Экспертными знаниями в САПР будем считать формализованную информацию о принятии решений и выполнении действий проектировщиком либо системой проектирования в конкретных проектных ситуациях для достижения заданных целей.

В данной формулировке подчеркивается роль понятия "знания" с точки зрения принятия решений. Конечно, это не единственные знания, которые могут быть "заложены" и используются в САПР. В данном определении косвенным образом учитываются и другие, т. е. "алгоритмические" знания (методы, алгоритмы), "встроенные" в программные модули, реализующие проектные операции. В области создания интеллектуальных систем вопрос об определении (и разграничении) понятий "данные" и "знания" подвергался многочисленным обсужде-

ниям. Есть основания полагать, что предложенное определение понятия "экспертные знания" приемлемо для использования и в САПР других объектов (машиностроительных конструкций, зданий, технологических процессов и т. д.), однако данный вопрос требует отдельного изучения.

Продукционно-фреймовая модель представления знаний

Чтобы можно было использовать экспертные знания, они должны быть формализованы. Для проблемной области логического проектирования в работе [15] используется продукционно-фреймовая модель представления знаний. Фреймы позволяют выделить "рамки" описания проектных ситуаций, продукции дают возможность определения последовательностей проектных действий.

Понятие *продукции* формально определим как запись

$$P^i: P^i \rightarrow V^i,$$

где P^i — условие применимости; V^i — заключение или действие, которое имеет место при истинности P^i . Условие применимости P^i представляет собой скобочное логическое *выражение*, в котором могут быть использованы только три логические операции: конъюнкция, дизъюнкция и отрицание (инверсии) над символами логических переменных $P_1^i, P_2^i, \dots, P_q^i$, являющихся предикатами. *Предикатом* называется логическая функция, которая при конкретных значениях своих аргументов превращается в высказывание со значением из множества {"истина", "ложь"}. Таким образом, левая часть продукции может принимать только одно из двух значений — "ложь" либо "истина". Аргументами каждого из предикатов могут быть только однотипные атрибуты. Допустимые типы атрибутов: символьный, числовой, логический. Каждый предикат задается своим выражением. Тождественно истинный предикат задает *безусловную* продукцию. В качестве правых частей V^i продукций используются некоторые выполняемые проектные действия. Упорядоченные системы пронумерованных продукций понимаются как *стратегии*.

Атрибуты задают набор существенных (с точки зрения организации процесса проектирования) свойств проектируемых объектов, технологических ограничений и целей проектирования. Совокупность задействованных в системе атрибутов определяет пространство признаков, необходимых для принятия проектных решений. Атрибуты представляются полным названием, сокращенным названием, типом и текущим значением. Полное название атрибута отражает его целевое назначение. Сокращенное название используется при записи определенных в системе продукций. Атрибуты могут быть трех

типов: логический (*bool*), числовой (*integer*) и символьный (*text*). Атрибуты могут быть классифицированы как атрибуты описаний объектов проектирования и базисов проектирования, атрибуты программных модулей, вспомогательные атрибуты для управления выполнением продукции. Атрибуты группируются по слотам, которые, в свою очередь, включаются во фреймы. В результате такого группирования признаковое пространство предлагается описывать четырьмя фреймами: *Блок проекта* (описание иерархически организованных данных), *Параметры* (описание базисов проектирования и параметров), *System* (пути к директориям и имена системных файлов), *Work* (атрибуты для управления стратегиями).

Заключение

Предложенная продукционно-фреймовая модель представления знаний была положена в основу продукционной системы FLC [15], предназначенной для проектирования логических схем. Комбинированные маршруты оптимизации, синтеза и верификации логических схем записываются в виде иерархически организованных стратегий. Широкие экспериментальные исследования подтвердили высокую эффективность совместного использования промышленного синтезатора LeonardoSpectrum и FLC.

Список литературы

1. Немудров В., Мартин Г. Системы-на-кристалле. Проектирование и развитие. М.: Техносфера, 2004. 216 с.

2. Бибило П. Н. Синтез логических схем с использованием языка VHDL. М.: Солон-Р, 2002. 384 с.
3. Закревский А. Д. Логический синтез каскадных схем. М.: Наука, 1981. 416 с.
4. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи: Пер. с англ. М.: Мир, 1982. 416 с.
5. Бибило П. Н. Системы проектирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, LeonardoSpectrum. М.: СОЛОН-Пресс, 2005. 384 с.
6. Хаханов В. И., Литвинова Е. И., Гузь О. А. Проектирование и тестирование систем на кристаллах. Харьков: ХНУРЭ, 2009. 484 с.
7. Рабаи Ж. М., Чандракасан А., Николич Б. Цифровые интегральные схемы, 2-е издание: Пер. с англ. М.: ООО "И.Д. Вильямс", 2007. 912 с.
8. Зотов Ю. В. Проектирование цифровых устройств на основе ПЛИС фирмы XILINX в САПР WebPack ISE. М.: Горячая линия—Телеком, 2003. 624 с.
9. Бухтеев А. Synplicity — синтез от А до Я // Электроника: Наука, Технология, Бизнес. 2004. № 7. С. 50—54.
10. Поляков А. К. Языки VHDL и VERILOG в проектировании цифровой аппаратуры. М.: СОЛОН-Пресс, 2003. 320 с.
11. Eisner C., Fisman D. A Practical Introduction to PSL. Springer, 2006. 240 p.
12. Уэйкерли Дж. Ф. Проектирование цифровых устройств. Т. 1. М.: Постмаркет, 2002. 544 с.
13. Trevillyan L. On Overview of Logic Synthesis Systems // Proc. 24th ACM/IEEE Design Automation Conference. 1987. P. 166—172.
14. Чебурахин И. Ф. Математические модели для интеллектуализации синтеза дискретных логических управляющих устройств // Изв. РАН. Теория и системы управления. 2008. № 1. С. 68—77.
15. Бибило П. Н., Романов В. И. Логическое проектирование дискретных устройств с использованием продукционно-фреймовой модели представления знаний. Минск: Беларус. навука, 2011. 279 с.
16. Евгеньев Г. Б. Интеллектуальные системы проектирования: учеб. пособие. М.: Изд-во МГТУ им. Н. Э. Баумана, 2009. 334 с.
17. Москвин В. П. Азбука TCL. М.: Горячая линия-Телеком, 2003. 216 с.

IV Международная конференция

"Проблемы кибернетики и информатики" (PCI'2012)

12—14 сентября 2012 года, г. Баку

Организаторы конференции:

Национальная академия наук Азербайджана,
Министерство связи и информационных технологий АР,
Министерство образования АР, Международный центр теоретической физики Абдуса Салама,
Национальная академия авиации, Институт кибернетики НАНА,
Институт информационных технологий НАНА, Бакинский государственный университет,
Азербайджанская государственная нефтяная академия, Азербайджанский технический университет,
Университет Кавказ, НИИ прикладной математики БГУ.

В работе конференции планируется участие ведущих ученых из США, России, Германии, Италии и других стран.

Доклады, прошедшие рецензирование, будут изданы в Трудах к началу конференции и размещены на сайте www.pci2012.science.az.

Вся необходимая информация размещена на сайте www.pci2012.science.az

УДК 004.7.056

В. Н. Тарасов, д-р техн. наук, проф., зав. каф.,
Е. М. Мезенцева, аспирант, ассистент,
Поволжский государственный университет
телекоммуникаций и информатики, г. Самара,
e-mail: katya-mem@psati.ru, vt@ist.psati.ru

Организация защиты компьютерных сетей. Метод многомодульной фильтрации спама на Web-сайтах

Описана модель фильтрации спама на сайтах в комментариях, формах обратной связи и других интерактивных разделах. Приведена общая схема работы фильтра, схема процесса фильтрации спама, обучение фильтра. Описан алгоритм идентификации спама.

Ключевые слова: спам, сообщение, фильтр, схема работы, токен, независимые редкие биграммы, формула Байеса

Введение

В современном обществе использование сети Интернет в качестве средства коммуникаций — широко распространенная практика. Сегодня уже не только электронная почта, но и блоги, социальные сети, форумы, Instant messaging (службы мгновенных сообщений — ICQ, skype и др.) являются привычными средствами общения.

По прогнозам исследовательской и консалтинговой компании Gartner, специализирующейся в информационных технологиях, к 2014 г. социальные сети станут основным средством электронного обмена в 20 % организаций. В то же время росту популярности социальных сервисов будут способствовать повышение их безопасности, создание "белых" сообществ и толерантное отношение к использованию персональных аккаунтов во время работы. В ходе дальнейшего развития принципиальные различия между электронной почтой и социальными сетями постепенно нивелируются, и почтовая служба перестанет играть главенствующую роль в осуществлении деловых операций [1].

Можно ожидать, что больше половины бизнес-структур будут поддерживать связь друг с другом

через подобие микроблогов, с улучшенной системой контроля и защиты информации.

Изначально данные виды коммуникаций оставались незамеченными спамерами, однако с ростом их популярности ситуация коренным образом изменилась. На сегодняшний день по различным подсчетам уже примерно 10 % IM (instant messenger) сообщений и 17 % комментариев в блогах, форумах и социальных сетях, представляет собой спам [2]. Заинтересованность спамеров в новых средствах коммуникации объясняется не только их растущей популярностью и увеличивающейся базой пользователей, но и тем, что средства борьбы со спамом в программах мгновенного общения и блогах находятся в настоящее время в зачаточном состоянии. Практически не существует универсальных решений, предназначенных для фильтрации спама на сайтах в комментариях, формах обратной связи и других интерактивных разделах.

Существует лишь небольшое число узкоспециализированных инструментов, препятствующих автоматическому размещению сообщений, рассчитанных на конкретную систему управления контентом, например, такую как WordPress. Эти модули имеют серьезные недостатки — либо они распространяются по принципу "как есть" без статистической базы, либо являются онлайн-сервисами, но ориентированы, главным образом, на западный рынок. В настоящее время нет универсальных решений, позволяющих выполнять фильтрацию на русскоязычных сайтах. К сожалению, подобные системы детектирования спама абсолютно беспомощны в случае, если спам распространяется вручную. Растет доля фишинга, целью которого является получение доступа к конфиденциальным данным пользователей — логинам и паролям. Сюда относятся кражи паролей, номеров кредитных карт, банковских счетов и другой конфиденциальной информации. Именно такой вид спама чаще всего распространяется вручную, и эта проблема носит серьезный характер.

Другие инструменты, препятствующие автоматическому размещению сообщений, — это тест Тьюринга. Наиболее популярной его реализацией является CAPTCHA (Completely automated public Turing test to tell computers and humans apart — Полностью автоматизированный публичный тест Тьюринга для различия компьютеров и людей) [2]. Суть этого теста заключается в том, что существуют

задачи, которые легко может исполнить человек, но которые очень тяжело реализовать с помощью программных средств. К ним относятся: распознавание образов, текста, человеческой речи, решение детских загадок и т. п. Существует несколько его разновидностей, применяемых в Интернете:

1) определение алфавитно-цифровой последовательности на картинке.

2) распознавание предмета на картинке или фотографии.

К плюсам первого метода можно отнести сравнительно простую реализацию и возможность отображения картинки на подавляющем числе пользовательских устройств. Минус в том, что владельцы блогов постоянно усложняют методы генерации текста на картинке и зачастую приходят к тому, что он становится трудно распознаваемым не только для компьютера, но и для человека, особенно имеющего проблемы со зрением.

Плюс второго метода заключается в том, что он наиболее легко и быстро распознается человеком, даже имеющим дефекты зрения. Минус этого метода — ограниченное число заранее проиндексированных изображений предметов, которые легко подвергаются перебору и составлению соответствия "картинка — значение" на стороне спамера.

В дальнейшем, безусловно, будет продолжение роста активности спамеров, так как трафик постоянно увеличивается. При этом в общем потоке будет возрастать доля ручного спама. Это произойдет за счет двух факторов: с одной стороны, за счет растущей популярности фишинга, с другой стороны, за счет начала использования этой технологии в рекламном спаме.

Поэтому разработка специализированной модели, методов и алгоритмов, предназначенных для фильтрации спама на форумах, блогах и в социальных сетях, актуальна и представляет научный и практический интерес в области защиты компьютерных сетей.

Постановка задачи

Ниже предлагается новый метод фильтрации спама на Web-сайтах, который, в отличие от известных подходов, позволит повысить качество оценки данных за счет учета следующих параметров: числа сообщений, в которых встречались слова определенной категории (спам/не спам); частоты использования слов в сообщениях определенной тематики; учета слов, впервые встретившихся в проверяемом сообщении и не существ-

вовавших до этого в базе данных, учета различных морфологических форм одного и того же слова [3].

Разрабатываемый антиспам-фильтр должен обеспечивать высокую скорость и стабильность работы, должны быть минимизированы ложные срабатывания, система фильтрации должна обеспечивать удобный интерфейс взаимодействия с любым Интернет-ресурсом.

Основными преимуществами системы станут гибкость изменения алгоритмов ее работы за счет многомодульности и простота надстройки новых средств обработки нежелательных сообщений для незамедлительного отражения новых видов атак спамеров сразу на всех подключенных Интернет-ресурсах, без каких-либо дополнительных действий со стороны администратора сайта.

1. Общая схема работы фильтра

В результате анализа поставленной задачи была получена спецификация разрабатываемого фильтра и на ее основе построена общая модель взаимодействия с пользователем и клиентом. Конкретизируем ее основные функции.

Фильтрация спама на сайтах-участниках системы работает по следующей общей схеме, приведенной на рис. 1.

1. Пользователь сайта, подключенного к сервису фильтрации спама, отправляет на сайт текстовое сообщение.

2. Поступившее сообщение сохраняется на сайте, и отправляется запрос с текстом сообщения фильтру.

3. Система фильтрации проводит необходимые преобразования, вычисления и принимает решение, является ли сообщение спамом.

4. В зависимости от внутренней политики программное обеспечение сайта либо скрывает сообщение пользователя, либо помечает его как спам, а также может приниматься решение об его удалении.

5. В случае если произошло ложное срабатывание фильтра, администратор сайта корректирует ре-



Рис. 1. Общая схема работы фильтра

зультат его работы. Результат корректировки отправляется на сервер спам-фильтра. Сервис фильтрации спама выполняет обучение фильтра по данному конкретному сообщению. Каждое новое сообщение сохраняется на сервере фильтра с присвоенным ему уникальным идентификатором.

2. Схема процесса фильтрации спама

В данной статье предлагается новый алгоритм фильтрации спама на основе семантического анализа сообщения и байесовского подхода, позволяющий в значительной степени уменьшить число ложных "тревог" и пропуска спама.

Для корректной работы фильтра необходимо провести его начальное обучение, т. е. вручную администратором сайта отсортировать спам от не спама. Это обучение должно проводиться на достаточно большой выборке существующих сообщений. Принцип работы фильтра после начального обучения показан на рис. 2.

В схеме процесса фильтрации блоки 3—7 как основные, реализованы в виде отдельных программных модулей. Рассмотрим основные этапы функционирования фильтра (рис. 2). В связи с тем, что сообщение обычно присылается в формате html, проводится предобработка текста, т. е. удаление всех html-тегов, в результате чего остается "чистый" текст, пригодный для анализа.

Основой метода статистического анализа сообщения является разбиение текста на отдельные слова с учетом морфологии, т. е. слова в разных словоформах будут определены как идентичные.



Рис. 2. Схема процесса фильтрации спама

Например, слова различных падежей будут приведены к именительному (в основную форму). Таким образом, если в тексте встречаются слова в разных формах они будут распознаны фильтром не как отдельные слова, а как одно слово.

В дополнение к разбивке на отдельные слова фильтр реализует алгоритм разбиения на словосочетания OSB (orthogonal sparse bigrams).

Для снижения вычислительной нагрузки принято решение не учитывать при анализе слова, имеющие коэффициент спамности 50 %, так как такие слова не оказывают влияния на конечный результат. Это в основном общеупотребительные, не несущие смысла слова — предлоги, союзы, частицы, местоимения.

Кроме того, целесообразно не учитывать слова, по которым собрано недостаточно статистики, т. е. слова, участвовавшие в сообщении менее 5 раз. Таким образом, если анализируемое слово ранее не встречалось, оно автоматически получит коэффициент 0,5, а по мере накопления статистики это значение будет выходить на свой естественный уровень.

Произведением вероятностей отдельных слов получаем вероятность совместного появления всех слов в сообщении. Вычисление вероятности соотнесения конкретного нового сообщения к той или иной категории выполняется по формуле Байеса. В зависимости от результата, если удалось классифицировать сообщение, мы помечаем его как спам или не спам и проводим автоматическое обучение. Если же сообщение классифицировать не удалось, администратор сайта сам принимает решение, к какой категории отнести сообщение. Таким образом, постоянно происходит непрерывное обучение фильтра на протяжении всего срока эксплуатации.

3. Обучение фильтра

От процесса обучения фильтра зависит качество его работы.

На вход подается выборка промаркированных (спам/не спам) сообщений. Из текста удаляется вся лишняя информация (общие слова, html-разметка, знаки препинания). Текст разбивается на отдельные слова — токены (от английского token — признак, лексема). Каждому токenu присваиваются следующие значения: статистическая вероятность наличия в группе спама; статистическая вероятность присутствия в разрешенных сообщениях. Эти значения и определяют вероятность того, что сообщение относится к категории спам или не спам. По мере накопления статистики программа составляет частотные словари — для каждой категории сообщения определяет, сколько раз и какой токен встретился в сообщениях этой категории.

При токенизации сообщения необходимо учесть морфологию, т. е. привести слова к базовой форме, что позволит идентифицировать повторяющиеся

слова в разных формах (падежах, склонениях, числах). Для поиска нормальной формы каждого слова используются специализированные словари.

Морфологические модули группы разработчиков АОТ (aot.ru) призваны решить указанные выше проблемы, обеспечив научные коллективы и вообще любых возможных энтузиастов-экспериментаторов системой морфологического анализа, которая:

- уже имеет словари достаточно большого объема, пополняется добровольцами, поэтому не должна устаревать;
- при поиске в словаре использует конечный автомат, что позволяет находить слово за линейное от его длины время;
- написана на C++, компилируется под Linux и под Windows;
- имеет развитую систему добавления новых слов;
- имеет в распоряжении русский, немецкий и английский лексиконы;
- распространяется бесплатно под лицензией LGPL в исходных кодах.

Все указанные выше свойства по отдельности можно встретить в существующих модулях морфологического анализа, однако именно данное сочетание свойств составляет новизну и актуальность представленной системы. Поэтому данное программное обеспечение нами было переписано на языке PHP.

Физически словарь представлен в виде нескольких файлов.

1. Key = value — хранилище всех словоформ. Хранилище использует модель минимального детерминированного конечного автомата (DAWG), позволяет искать слово за линейное от длины слова время $O(N)$, где N — длина слова.

2. Хранилище грамматической информации, таблицы окончаний, грамем.

3. Хранилище окончаний для предсказания, в котором используется структура, похожая на п. 1.

Для более полной идентификации одинаковых сообщений используется алгоритм разбиения на токены — OSB, который выбирает редко встречающиеся комбинации словосочетаний в тексте. Для этого каждое слово сопоставляется с последним словом текста. Для каждого сообщения мы получаем $N - 1$ дополнительных токенов, с наибольшей вероятностью присущих только данному тексту, где N — число слов в тексте. Например, если нам пришел текст: "Хорошо ли ты знаешь английский язык?", то мы получим следующие токены:

"Хорошо - - - - язык";

" - ли - - - - язык";

"- -ты - - язык";

"- - -знаешь - язык";

"- - - -английский язык".

На следующем этапе составляются частотные словари, т. е. для каждого токена вычисляются следующие параметры: частота использования токена

в письмах спама (не спама); частота появления данного токена в письме категории спам (не спам); общая частота встречаемости токена.

Вероятности принадлежности нового сообщения к тому или иному типу определяются по формуле Байеса. Причем вычисления проводятся для каждого признака нового сообщения. Вероятности для всего сообщения получаем суммированием и нормализацией полученных вероятностей каждого признака. В большинстве случаев вероятность принадлежности сообщения к одной из категорий намного выше, чем к остальным, в эту категорию сообщение и отправляется.

Вычисления, проведенные на этапе обучения фильтра, и кэширование результатов разгружают систему и ускоряют получение результата категоризации входящего сообщения. Также благодаря постоянному обучению фильтра значительно увеличивается точность определения спама с помощью формулы Байеса.

4. Алгоритм идентификации спама, основанный на применении формулы Байеса и априорного знания

Для того чтобы идентифицировать поступившее на сайт сообщение, будем исходить из словаря, используя байесовские оценки. Для любого сообщения введем две гипотезы:

H_A — спам;

H_B — не спам.

Отнесение сообщения, пришедшего на сайт, к спаму/не спаму будет осуществляться по приведенному ниже алгоритму.

Пусть

F_a — общее число сообщений спама;

F_{ai} — число сообщений с токеном i в группе спама;

F_b — общее число не спам-сообщений;

F_{bi} — число сообщений с токеном i в группе не спама.

Тогда

$$p_{ai} = \frac{F_{ai}}{F_a} \text{ — вероятность появления токена } i$$

в спам-сообщении;

$$p_{bi} = \frac{F_{bi}}{F_b} \text{ — вероятность появления токена } i \text{ в не}$$

спам-сообщении.

Для получения объединенных вероятностей всего сообщения введем следующие события: A — документ относится к спаму; B — документ не спам-сообщение. Мы предположили, что вероятности независимы, поэтому возможно их перемножение:

$F(A) = p_{a1} \times p_{a2} \times \dots \times p_{an}$ — вероятность совместного появления слов в спаме;

$P(B) = p_{b1} \times p_{b2} \times \dots \times p_{bn}$ — вероятность совместного появления слов в не спаме, где n — число слов в сообщении;

$p_a = \frac{F_a}{F_a + F_b}$ — априорная вероятность, что сообщение окажется спамом;

$p_b = \frac{F_b}{F_a + F_b}$ — априорная вероятность, что сообщение окажется не спамом;

$O_a = \frac{P_a}{1 - P_a}$ — априорные шансы, что сообщение окажется спамом;

$O_b = \frac{P_b}{1 - P_b}$ — априорные шансы, что сообщение окажется не спамом.

Тогда на основе теоремы Байеса с применением априорного знания получим:

$P(H_A) = \frac{P(A) \times O_a}{P(A) \times O_a + P(B) \times O_b}$ — апостериорная вероятность, что это спам-сообщение;

$P(H_B) = \frac{P(B) \times O_b}{P(A) \times O_a + P(B) \times O_b}$ — апостериорная вероятность, что это не спам сообщение [4].

Для окончательного принятия решений в методе классификации Байеса необходимо задать начальные значения нижнего и верхнего порогов. Пусть T и L — величины, измеряющиеся в процентах и определяющие соответственно верхний и нижний пороги принятия решений; H — одна из определенных ранее групп (спам/не спам сообщения); $P(H)$ — вероятность попадания сообщения в одну из определенных ранее групп.

Тогда:

- будем считать, что документ принадлежит группе H , если $P(H) \geq T$;
- документ не принадлежит группе H , если $P(H) \leq L$;
- если же $T \geq P(H) \geq L$, то нельзя принять никакого решения.

5. Экспериментальное исследование степени корректности фильтрации сообщений с помощью формулы Байеса и априорного знания

Для тестирования работы рассмотренного выше алгоритма классификации было проведено обучение нашей системы фильтрации спама на 300 сообщениях спама и 500 не спама.

Тестирование работы фильтра при различных порогах принятия решения показало, что самыми оптимальными являются:

- верхняя граница $T = 0,95$;
- нижняя граница $L = 0,4$.

Мы установили жесткие рамки по спаму и обычные для не спама. Это было сделано для того, чтобы

как можно меньше нормальных сообщений проходило в спам, т. е. избегаем ложных срабатываний.

После определения порогов принятия решения переходим к тестированию работы выбранного метода фильтрации спама.

Подаем на вход фильтра 400 спам-сообщений.

Метод Байеса классифицировал все сообщения как спам (100 % вероятность).

Тестируем на 200 не спам-сообщений:

- спам-сообщения — 11 сообщений (5,5 %);
- не спам-сообщения — 171 сообщение (85,5 %);
- неопределенные — 18 сообщений (9 %).

Проверяем работу фильтра на 200 новых перемешанных сообщениях (135 спам/65 не спам):

- спам сообщения — 137 сообщений;
- не спам сообщения — 57 сообщений;
- неопределенные — 6 сообщений.

Возникли два ложных срабатывания — нужные сообщения оценены как спам, в итоге все 135 спам-сообщений определены верно.

Процент ложных срабатываний (ошибочная классификация нужных сообщений как спам) и доля пропуска спама являются основными критериями при оценке работы алгоритмов фильтрации спама. Полностью избавиться от них невозможно, но необходимо максимально уменьшить их число. Этого мы и добились, применяя алгоритм фильтрации спама по методу Байеса. Даже при обучении на небольшой выборке сообщений процент ложных срабатываний не велик, а чувствительность нашего метода (доля выявленных спам сообщений) приближается к 100 %.

Заключение

Для того чтобы фильтр был точным, мы постарались проанализировать и избавиться от большинства недостатков, выявленных на этапе разработки, объединив несколько методов, разработали комплексный алгоритм работы.

В дальнейшем планируется глубокая доработка системы, а именно наполнение ее дополнительными сервисами, улучшение алгоритмов обучения и процесса принятия решения.

Список литературы

1. **Социальные** сети вытеснят почтовый сервис? [Электронный ресурс]. URL: http://www.securelist.com/ru/blog/32409/Sotsialnye_seti_vytesnyat_pochtovyy_servis
2. **Возможные** методы борьбы со спамом в блогах и мессенджерах [Электронный ресурс]. URL: <http://www.weblistpro.com/article/antispam-filters.html>
3. **Мезенцева Е. М.** Разработка специализированной модели, методов и алгоритмов, предназначенных для фильтрации спама // Интеллект. Инновации. Инвестиции. — 2011. Специальный выпуск по материалам международного форума "Россия как трансформирующееся общество: экономика, культура, управление". С. 73–77.
4. **Джарратано Д., Райли Г.** Экспертные системы, принципы разработки и программирование. 4-е изд. М.: Вильямс, 2007. 1147 с.

С. Н. Карутин,

канд. техн. наук, зам. начальника центра,
ОАО "Российские космические системы",
e-mail: sergey.karutin@rniikp.ru

Метод автономного эфемеридно-временного обеспечения навигационных космических аппаратов системы ГЛОНАСС

Проведен анализ существующих методов эфемеридно-временного обеспечения навигационных космических аппаратов космических комплексов ГЛОНАСС. Предложен метод автономного и оперативного определения эфемеридно-временной информации на борту навигационных космических аппаратов ГЛОНАСС за счет использования сети земных станций, формирующих радионавигационное поле с использованием навигационных радиосигналов навигационных космических аппаратов системы ГЛОНАСС. Приведены результаты проектирования системы оперативного эфемеридно-временного обеспечения навигационных космических аппаратов системы ГЛОНАСС. На основе проведенных исследований сделаны выводы о возможных направлениях развития элементов космического комплекса системы ГЛОНАСС.

Ключевые слова: глобальные навигационные спутниковые системы, ГЛОНАСС, эфемеридно-временное обеспечение

Введение

Одной из важнейших задач современного этапа развития глобальной навигационной спутниковой системы ГЛОНАСС является повышение точности навигационных определений, которая в основном зависит от погрешности эфемеридно-временной информации (ЭВИ) навигационных радиосигналов навигационных космических аппаратов (НКА) ГЛОНАСС, погрешности оценки значений задержек радиосигналов, вызванных ионосферной и тропосферной рефракциями, от аппаратурных погрешностей, а также погрешности, обусловленной многолучевым характером распространения сигнала НКА.

Анализ составляющих погрешности навигационных определений [1] показывает, что уменьшение погрешности оценивания ЭВИ с существующих значений в 1,4 до 0,3 м (среднего квадратического отклонения — СКО) является одной из важнейших задач следующего этапа развития системы ГЛОНАСС, которая должна быть решена в период 2012—2020 годов.

В настоящее время реализуется целый комплекс мероприятий, включающий модернизацию наземного комплекса управления, совершенствование бортовой аппаратуры НКА и совершенствование технологии расчета ЭВИ [2]. Наиболее важным результатом данной работы является переход НКУ системы ГЛОНАСС на запросно-беззапросную технологию эфемеридно-временного обеспечения (ЭВО), которая обеспечила уменьшение погрешности ЭВИ с 8 до 2 м (СКО) в 2007 г. Активные работы проводятся в направлениях внедрения технологии ЭВО на основе межспутниковых измерений [3] и запросно-беззапросных лазерных дальномерных измерений [4], которые подтверждают возможность обеспечения высокой точности ЭВО. Вместе с тем указанные средства имеют ряд недостатков — бортовая аппаратура межспутниковых измерений работает в импульсном режиме, а использование квантово-оптических средств ограничивается метеорологическими условиями.

В качестве возможного альтернативного варианта построения системы ЭВО НКА ГЛОНАСС можно рассматривать систему земных радиомаяков. В настоящее время существует положительный опыт использования доплеровской системы DORIS (Doppler Orbitography and Radiopositioning Integrated by Satellite) [5] и запросно-беззапросной системы PRARE (Precise Range and Range Rate Equipment) [6] на основе опорных земных станций (ЗС) для навигационного обеспечения космических аппаратов на низких орбитах (до 1500 км). Однако система DORIS позволяет осуществлять навигационные определения лишь с периодичностью 10 с и погрешностью 1 м (СКО) в реальном времени и 0,025 м (СКО) при апостериорной обработке, а при использовании системы PRARE погрешность апостериорной оценки ЭВИ составляет 0,05 м (СКО) [7]. При этом земная станция PRARE в состоянии обслуживать лишь один космический аппарат.

В связи с этим актуальной является задача разработки методов и технических средств, обеспечивающих оперативное высокоточное ЭВО НКА ГЛОНАСС в реальном масштабе времени с погрешностью менее 0,3 м (СКО) за счет использования ЗС.

Архитектура системы

Рассматриваемый метод ЭВО (рис. 1) основан на использовании измерений текущих навигационных параметров (ИТНП), полученных при определении задержки распространения радиосигналов на радиоперелиниях НКА — ЗС и ЗС — НКА. Поскольку координаты фазового центра приемной антенны ЗС априорно известны, то оценив время распространения радиосигналов от НКА до ЗС и в обратном

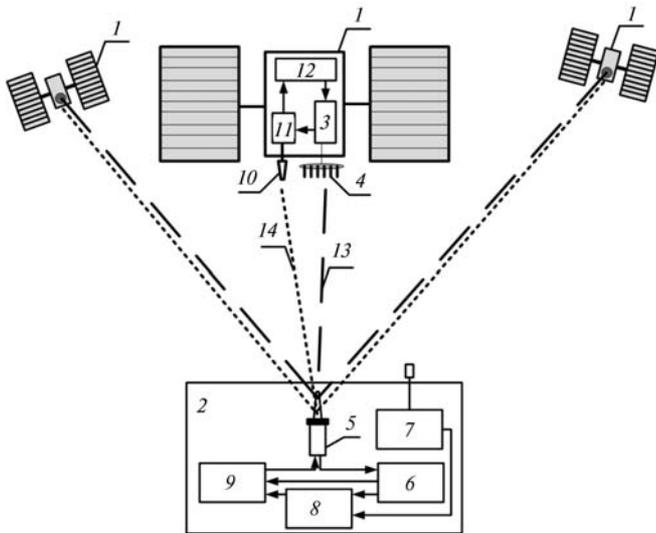


Рис. 1. Архитектура системы автономного ЭВО НКА ГЛОНАСС: 1 — НКА ГЛОНАСС; 2 — земная станция; 3 — бортовой источник навигационного радиосигнала; 4 — передающее антенно-фидерное устройство (АФУ) НКА; 5 — приемно-передающее АФУ земной станции; 6 — приемное устройство земной станции; 7 — метеостанция; 8 — вычислитель земной станции; 9 — источник навигационного сигнала земной станции; 10 — приемное АФУ НКА; 11 — бортовой приемник навигационного сигнала; 12 — бортовой цифровой вычислительный комплекс; 13 — навигационная радиолиния Космос — Земля; 14 — навигационная радиолиния Земля — Космос

направлении, бортовая аппаратура НКА может оценить параметры вектора состояния НКА.

Для оценки времени распространения радиосигналов от НКА до ЗС можно использовать навигационные радиосигналы НКА в L -диапазоне на частотах 1600,995 МГц и 1248,06 МГц, поскольку комбинированная обработка измерений навигационных параметров на данных частотах позволяет определить задержку радиосигналов в ионосфере. Кроме того, высокая стабильность данных колебаний позволяет применять их для формирования навигационного радиосигнала ЗС, который будет использоваться для измерения навигационных параметров радиолинии ЗС — НКА.

Для достижения требуемых высоких точностных характеристик ЭВО необходимо использовать ре-

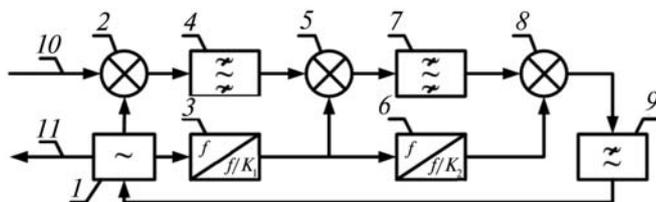


Рис. 2. Схема формирования несущих колебаний в ЗС: 1 — генератор; 2, 5, 8 — смеситель; 3, 6 — делитель; 4, 7 — полосовой фильтр; 9 — фильтр низких частот; K_1, K_2 — коэффициенты деления

зультаты измерения задержки распространения радиосигналов и фазы несущих колебаний. С учетом указанного обстоятельства при формировании навигационного радиосигнала ЗС необходимо учесть значение фазы принимаемого радиосигнала НКА. Общепринятой практикой формирования навигационных радиосигналов на борту НКА является создание сетки частот, имеющих общую опорную частоту колебаний. Для новых навигационных радиосигналов НКА ГЛОНАСС с кодовым разделением опорные колебания бортового синхронизирующего устройства (БСУ) имеют частоту $f_0 = 5,115$ МГц. Очевидно, что для простоты построения частотного плана приемного устройства на борту НКА навигационный радиосигнал ЗС также должен иметь частоту несущих колебаний, кратную опорным колебаниям БСУ НКА.

На рис. 2 приведена схема формирования несущих колебаний радиосигнала ЗС из принятого радиосигнала НКА, которая позволяет путем трехкратного преобразования входных колебаний с частотой $313 f_0 = 1600,995$ МГц получить на выходе фильтра низких частот 9 колебания с частотой f_0 , из которых генератор 1 формирует несущие колебания радиосигнала ЗС на частоте $390 f_0 = 1994,85$ МГц.

Приемо-передающее антенно-фидерное устройство (АФУ) ЗС и многоканальное приемное устройство ЗС позволяют параллельно сопровождать радиосигналы от всех видимых НКА, оценивать навигационные параметры и формировать опорные колебания для источника навигационного сигнала ЗС. Поскольку обработка радиосигналов каждого НКА проводится независимо, то и ответный навигационный радиосигнал для каждого НКА можно сформировать в цифровом виде в отдельном канале, а затем суммарный групповой сигнал излучить с использованием приемно-передающего АФУ ЗС. При этом ответный радиосигнал для каждого НКА будет иметь индивидуальные характеристики — уникальный дальномерный код из семейства периодических псевдослучайных последовательностей и собственный набор данных в составе цифровой информации. В составе цифровой информации на борт НКА должны быть переданы беззапросные измерения навигационных параметров, данные о координатах фазовых центров приемно-передающего АФУ ЗС, параметрах атмосферы в точке расположения ЗС и значениях аппаратурной задержки радиосигнала в ЗС.

Отличительной особенностью рассматриваемой системы является наличие высокой динамики движения НКА, который движется со скоростью около 4 км/с, и при высоте орбиты 19 100 км за время распространения радиосигнала местоположение НКА изменяется на $\sim 0,5$ км. Поэтому при разработке математических моделей первичных измерений навигационных параметров необходимо учитывать, что измерения проводят в различные моменты вре-

мени. Поэтому для упорядочивания дальнейших рассуждений определим три момента времени:

- t , момент излучения радиосигналов НКА;
- t_1 , момент излучения радиосигнала ЗС;
- t' , момент приема радиосигнала ЗС.

Указанные моменты времени связаны следующими соотношениями [3]:

$$\begin{aligned} t_1 - t &= \tau_{\text{НКА}}^{\text{И}}(t) = \tau_{\text{ЗС}}^{\text{П}}(t_1), \\ t' - t_1 &= \tau_{\text{НКА}}^{\text{П}}(t') = \tau_{\text{ЗС}}^{\text{И}}(t_1), \end{aligned} \quad (1)$$

где $\tau_{\text{НКА}}^{\text{И}}(t)$ — время распространения радиосигнала, излученного НКА в момент времени t ; $\tau_{\text{ЗС}}^{\text{П}}(t_1)$ — время распространения радиосигнала, принятого ЗС в момент времени t_1 ; $\tau_{\text{ЗС}}^{\text{И}}(t_1)$ — время распространения радиосигнала, излученного ЗС в момент времени t_1 ; $\tau_{\text{НКА}}^{\text{П}}(t')$ — время распространения радиосигнала, принятого НКА в момент времени t' .

Модели первичных измерений

В основе решения задачи ЭВО на борту НКА лежит использование первичных измерений навигационных параметров: беззапросных измерений псевдодальности S и псевдофазы Φ ; запросных дальности D и фазы Ψ по радиосигналам НКА и ЗС.

Модели беззапросных измерений хорошо известны и описаны в многочисленных источниках [1, 3] и в момент времени t_1 по шкале времени ЗС могут быть записаны в следующем виде:

$$S_i(t_1) = R(t_1, t) + \delta R(t) + c(\delta t(t_1) - \delta T(t) + \tau_{\text{троп}}(t_1) + \tau_{\text{ион}}(t_1) + \tau_{\text{рел}}(t_1) + \tau_{\text{мнл}}(t_1) + \tau_{\text{ап}}(t_1)) + e_S(t_1), \quad (2)$$

$$\begin{aligned} \lambda_i^{\text{НКА}} \Phi_i(t_1) &= R(t_1, t) + \delta R(t) + c(\delta t(t_1) - \\ &- \delta T(t) + \tau_{\text{троп}}(t_1) + \tau_{\text{ион}}(t_1) + \tau_{\text{рел}}(t_1) + \\ &+ \tau_{\text{мнл}}(t_1) + \tau_{\text{ап}}(t_1)) + \lambda_i^{\text{НКА}} N_i + e_{\Phi}(t_1), \end{aligned} \quad (3)$$

где $R(t_1, t) = \sqrt{(X_{\text{НКА}}(t) - x(t_1))^2 + (Y_{\text{НКА}}(t) - y(t_1))^2 + (Z_{\text{НКА}}(t) - z(t_1))^2}$ — геометрическая дальность между НКА с координатами $X_{\text{НКА}}(t) = (X_{\text{НКА}}(t), Y_{\text{НКА}}(t), Z_{\text{НКА}}(t))$ в момент времени t_1 и ЗС с координатами (x, y, z) в момент времени t ; δR — погрешность эфемеридной информации; c — скорость света; δt — расхождение между шкалами времени потребителя и используемой навигационной системы; δT — расхождение между шкалами времени НКА и его системы (погрешность частотно-временных поправок (ЧВП)); $\tau_{\text{ион}}$ — задержка распространения сигнала НКА вследствие ионосфер-

ной рефракции; $\tau_{\text{троп}}$ — задержка распространения сигнала НКА вследствие тропосферной рефракции; $\tau_{\text{рел}}$ — задержка распространения радиосигнала, обусловленная релятивистским эффектом и приливными эффектами; $\tau_{\text{мнл}}$ — задержка радиосигнала, обусловленная многолучевым характером распространения сигнала НКА; $\tau_{\text{ап}}$ — аппаратная задержка радиосигнала, обусловленная неидентичностью и неравномерностью характеристик группового времени запаздывания (ГВЗ) радиосигналов НКА в приемном тракте ЗС; $\lambda_i^{\text{НКА}}$ — длина волны несущих колебаний сигнала НКА, индекс i обозначает частотный диапазон $i = 1$ для диапазона L_1 и $i = 2$ для диапазона L_2 ; N_i — целочисленная неоднозначность измерения фазы; e_S, e_{Φ} — шумовые погрешности измерений псевдодальности и псевдофазы, соответственно.

Для построения модели первичных измерений запросных дальности и фазы рассмотрим процесс переноса фазы несущих и огибающих колебаний навигационного радиосигнала НКА при формировании ответного навигационного радиосигнала ЗС. Несмотря на то, что прием, обработка и формирование ответного радиосигнала осуществляются в цифровом виде, эквивалентом данных операций является обычная процедура переноса частоты в ЗС.

Отличительной особенностью рассматриваемой измерительной системы является наличие неоднородной среды распространения, включающей ионосферу, которая характеризуется различными значениями фазовой и групповой скоростей распространения радиосигналов, что приводит к нарушению взаимной синхронизации несущих колебаний и огибающей, при этом фаза несущих колебаний опережает фазу огибающей. Полный учет указанной особенности усложнит первоначальные выражения, поэтому рассмотрим вначале упрощенный случай, в котором средой распространения является вакуум.

При этом радиосигнал, сформированный в НКА в момент времени t , подвергается гетеродинированию в момент t_1 с помощью сигнала, сформированного по схеме, представленной на рис. 2, и принимается НКА в момент времени t' .

На выходе ЗС в момент времени t_1 навигационный радиосигнал имеет вид:

$$\begin{aligned} S_{\text{ЗС}}(t_1) &= S_{\text{НКА}}^{\text{И}}(t) S_{\text{ЗС}}^{\Gamma}(t_1) F_{\text{ЗС}}(w) = \\ &= A(t) \cos(w_{\text{НКА}} t) \cos(w_{\text{ЗС}}^{\Gamma} t_1) F_{\text{ЗС}}(w) = \\ &= A[t_1 - \tau_{\text{ЗС}}^{\text{П}}(t_1)] \cos[(w_{\text{НКА}} - w_{\text{ЗС}}^{\Gamma}) t_1 - w_{\text{НКА}} \tau_{\text{ЗС}}^{\text{П}}(t_1)] = \\ &= A[t_1 - \tau_{\text{ЗС}}^{\text{П}}(t_1)] \cos[w_{\text{ЗС}} t_1 - w_{\text{НКА}} \tau_{\text{ЗС}}^{\text{П}}(t_1)], \end{aligned} \quad (4)$$

где $A(t)$ — огибающие колебания; $F_{3C}(w) = \begin{cases} 2, & w > w_{\text{НКА}} \\ 0, & w < w_{\text{НКА}} \end{cases}$ — передаточная функция ЗС; $w_{\text{НКА}}$ — круговая частота несущих колебаний радиосигнала НКА, w_{3C}^{Γ} — круговая частота колебаний гетеродина ЗС; $w_{3C} = w_{\text{НКА}} - w_{3C}^{\Gamma}$ — круговая частота несущих колебаний радиосигнала ЗС.

Принятый НКА в момент времени t' навигационный радиосигнал имеет вид:

$$S_{\text{НКА}}^{\text{П}}(t') = A(t' - \tau_{3C}^{\text{П}}(t_1) - \tau_{3C}^{\text{И}}(t_1)) \cos[w_{3C}t' - w_{3C}\tau_{3C}^{\text{П}}(t_1) - w_{\text{НКА}}\tau_{3C}^{\text{И}}(t_1)]. \quad (5)$$

Используя выражения (1), (2), (3) и (5), запишем выражение для реального радиосигнала, прошедшего ионосферу и тропосферу, на входе приемного устройства НКА:

$$S_{\text{НКА}}^{\text{П}}(t') = A(t' - \tau_{\text{НКА}}^{\text{П}}(t') - \tau_{\text{НКА}}^{\text{И}}(t) - 2[\tau_{\text{троп}}(t_1) + \tau_{\text{ион}}(t_1) + \tau_{\text{рел}}(t_1)] - \tau_{\text{ап}}^{\text{П}}(t_1) - \tau_{\text{ап}}^{\text{И}}(t_1) - \tau_{\text{мнл}}(t_1)) \times \cos(w_{3C}t' - w_{3C}[\tau_{\text{НКА}}^{\text{И}}(t) + \tau_{\text{троп}}(t_1) - \tau_{\text{ион}}(t_1) + \tau_{\text{рел}}(t_1) + \tau_{\text{ап}}^{\text{И}}(t_1)] - w_{\text{НКА}}[\tau_{\text{НКА}}^{\text{П}}(t') + \tau_{\text{троп}}(t_1) - \tau_{\text{ион}}(t_1) + \tau_{\text{ап}}^{\text{П}}(t_1) + \tau_{\text{рел}}(t_1) + \tau_{\text{мнл}}(t_1)]), \quad (6)$$

где $\tau_{\text{ап}}^{\text{П}}$, $\tau_{\text{ап}}^{\text{И}}$ — аппаратные задержки при приеме и излучении радиосигнала в ЗС.

Исходя из выражения (6), запишем выражения для запросных измерений дальности D и фазы Ψ :

$$D(t') = R(t_1, t) + R(t_1, t') + 2c(\tau_{\text{троп}}(t_1) + \tau_{\text{ион}}(t_1) + \tau_{\text{рел}}(t_1)) + c(\tau_{\text{мнл}}(t_1) + \tau_{\text{ап}}^{\text{П}}(t_1) + \tau_{\text{ап}}^{\text{И}}(t_1)) + e_d(t_1, t); \quad (7)$$

$$\lambda^{3C}\Psi(t') = R(t_1, t') + c(\tau_{\text{троп}}(t_1) - \tau_{\text{ион}}(t_1) + \tau_{\text{рел}}(t_1) + \tau_{\text{ап}}^{\text{И}}(t_1)) + \lambda^{3C}N_3 + \frac{\lambda^{3C}}{\lambda_{\text{НКА}}} (R(t_1, t) + c(\tau_{\text{троп}}(t_1) - \tau_{\text{ион}}(t_1) + \tau_{\text{рел}}(t_1) + \tau_{\text{мнл}}(t_1) + \tau_{\text{ап}}^{\text{П}}(t_1)) + \lambda_{\text{НКА}}^{\text{НКА}} N_1) + e_{\Psi}(t, t_1), \quad (8)$$

где λ^{3C} — длина волны несущих колебаний радиосигнала ЗС; N_3 — неоднозначность измерения фазы несущих колебаний радиосигнала ЗС; e_d , e_{Ψ} — шумовые погрешности измерения дальности и фазы соответственно.

Задержка радиосигнала в тропосфере может быть исключена из числа оцениваемых параметров, поскольку в состав ЗС входит метеостанция,

обеспечивающая непрерывную оценку температуры, давления и влажности, на основании которых можно оценить влажную составляющую тропосферной задержки.

Уравнения вида (2), (3), (7) и (8) образуют систему нелинейных уравнений относительно искомым координат НКА $\mathbf{X}_{\text{НКА}}(t)$ и $\mathbf{X}_{\text{НКА}}(t')$, ухода шкалы времени НКА $\delta t(t)$, неоднозначности измерений псевдофазы и фазы, а также задержки радиосигнала в ионосфере. Указанная система уравнений может быть линеаризована путем разложения в ряд Тейлора с удержанием первых членов разложения относительно искомым параметров. Полученная система линеаризованных уравнений для измерений по одному НКА и одной ЗС будет иметь вид:

$$\mathbf{Y}_i^j(t', t_1, t) = \mathbf{H}_i^j(t', t_1, t)\Delta(t', t) + \mathbf{e}, \quad (9)$$

$$\text{где } \mathbf{Y}_i^j(t', t_1, t) = \begin{bmatrix} S_1(t_1) - R(t_1, t) \\ \lambda_1^{\text{НКА}} \Phi_1(t_1) - R(t_1, t) \\ S_2(t_1) - R(t_1, t) \\ \lambda_2^{\text{НКА}} \Phi_2(t_1) - R(t_1, t) \\ D(t') - R(t_1, t') - R(t_1, t) \\ \lambda^{3C}\Psi(t') - R(t_1, t') - R(t_1, t) \end{bmatrix}$$

вектор невязок измеренных и расчетных значений беззапросных псевдодальностей и псевдофаз, запрос-

$$\text{ных дальностей и фаз; } \Delta(t', t)^{\text{T}} = \begin{bmatrix} \Delta X_{\text{НКА}}(t) \\ \Delta Y_{\text{НКА}}(t) \\ \Delta Z_{\text{НКА}}(t) \\ \delta t(t) \\ \Delta X_{\text{НКА}}(t') \\ \Delta Y_{\text{НКА}}(t') \\ \Delta Z_{\text{НКА}}(t') \\ N_1^{\text{НКА}} \\ N_2^{\text{НКА}} \\ N^{3C} \\ \tau_{\text{ион}}(t) \end{bmatrix}$$

вектор искомым поправок к априорно известным значениям искомым параметров, $\mathbf{H}_i^j(t', t_1, t) =$

$$= \begin{bmatrix} C_x(t) & C_y(t) & C_z(t) & C_t(t) & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ C_x(t) & C_y(t) & C_z(t) & C_t(t) & 0 & 0 & 0 & \lambda_1^{\text{НКА}} & 0 & 0 & 1 \\ C_x(t) & C_y(t) & C_z(t) & C_t(t) & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ C_x(t) & C_y(t) & C_z(t) & C_t(t) & 0 & 0 & 0 & 0 & \lambda_2^{\text{НКА}} & 0 & 1 \\ C_x(t) & C_y(t) & C_z(t) & 0 & C_x(t') & C_y(t') & C_z(t') & 0 & 0 & 0 & 2 \\ C_x(t) & C_y(t) & C_z(t) & 0 & C_x(t') & C_y(t') & C_z(t') & \lambda_1^{\text{НКА}} & 0 & \lambda^{3C} & 2 \end{bmatrix}$$

— матрица частных производных выражений (2), (3), (7) и (8) по искомым параметрам; \mathbf{e} — вектор погрешностей ИТНП.

Учитывая, что в число оцениваемых параметров системы уравнений (9) входят вещественные и целочисленные параметры, для получения координат может быть использован трехэтапный алгоритм, предложенный в работе [9]. Указанный алгоритм обеспечивает получение максимально-правдоподобных оценок координат, только, если система уравнений (9) является определенной, т. е. число неизвестных меньше числа уравнений. В общем случае, когда число ЗС, от которых НКА принимает радиосигналы, равно M , при выполнении условия непрерывного слежения за фазой в ЗС и НКА при обработке измерений за L моментов времени число измерений равно $6ML$. При этом число оцениваемых параметров будет равно $M(5L + 3)$. Условие определенности системы уравнений (9) имеет вид:

$$6ML \geq M(5L + 3) \Rightarrow L \geq 3. \quad (10)$$

Таким образом, необходимым условием реализации предложенного метода оценки ЭВИ является непрерывное формирование ИТНП не менее чем тремя станциями.

При математическом моделировании с целью построения сети ЗС, удовлетворяющей условию (10), необходимо учитывать следующие ограничения:

- ЗС должна находиться на суше;
- максимально возможно должна использоваться территория Российской Федерации и станции Российской антарктической экспедиции.

Результаты моделирования (рис. 3, см. третью сторону обложки) показывают, что построение глобальной сети ЗС требует размещения не менее 12 станций по всему земному шару.

Вторым важным аспектом реализации предложенной системы в интересах ЭВО НКА ГЛОНАСС является решение системы уравнений (9) на борту НКА. Реальный опыт использования трехэтапной процедуры получения оценок координат, предложенной в работе [9], для решения аналогичных задач определения пространственной ориентации различных объектов показывает, что для быстрой сходимости необходима значительная избыточность измерений, которая достигается за счет либо увеличения числа станций, либо использования ИТНП, полученных на длительных интервалах времени.

Экспериментальные исследования

Оценить реализуемость предложенной системы на борту НКА ГЛОНАСС можно математическим моделированием алгоритма оценки координат с привлечением реальных ИТНП, полученных глобальной сетью из 24 беззапросных станций. Увеличение в 2 раза числа ЗС по отношению к минимально необходимому числу, определенному ранее, связано с необходимостью учета запросных

измерений и повышения сходимости оценок ЭВИ на коротких интервалах времени.

Эксперимент проводился с использованием компьютера на базе процессора Xeon E5620, загрузка которого не превышала 10 %. Таким образом, можно предположить, что аналогичные показатели времени расчета будут достигнуты при использовании процессора с тактовой частотой не менее 300 МГц.

Оценка времени, необходимого для обработки ИТНП, полученных на интервалах от 1 до 4 сут. по орбитальной группировке из 23 НКА ГЛОНАСС, приведена на рис. 4. Рис. 5 и 6 содержат результаты оценки СКО прогнозирования ЭВИ на интервалах до 6 ч по орбитальной группировке из тех же НКА ГЛОНАСС. Прогнозные значения ЭВИ сравнивались с эталонной ЭВИ, полученной в центре международной службы ГНСС (IGS) на основе апостериорной обработки данных на интервале 14 сут.

Полученные результаты показывают, что современные вычислительные средства позволяют решать задачу оперативного оценивания ЭВИ при

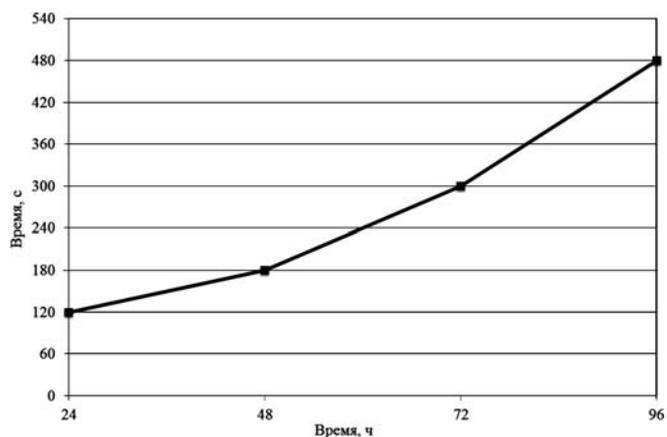


Рис. 4. Зависимость времени обработки от длительности интервала накопления измерений текущих навигационных параметров

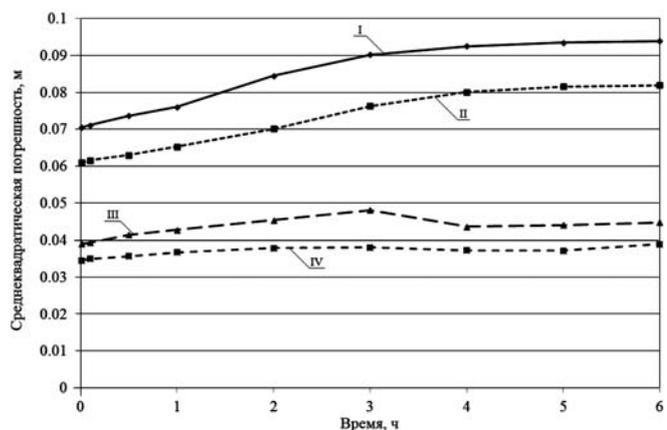


Рис. 5. Зависимость погрешности прогнозирования эфемерид НКА ГЛОНАСС от времени прогнозирования при различной длительности интервала накопления ИТНП: I — 1 сутки; II — 2 суток; III — 3 суток; IV — 4 суток

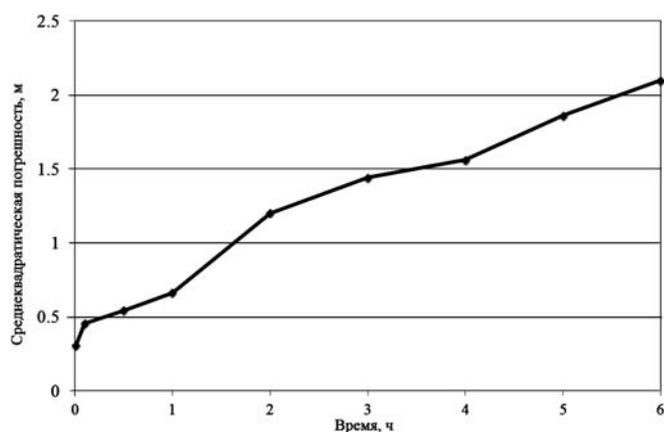


Рис. 6. Зависимость погрешности прогнозирования ухода шкалы времени НКА ГЛОНАСС от времени прогнозирования

обработке ИТНП на длительных интервалах времени. Использование ИТНП, полученных на интервалах 3–4 сут., позволяют получить прогноз эфемерид НКА с погрешностью менее 0,04 м при прогнозировании на интервал времени, требуемый для выполнения очередного цикла обработки. Вместе с тем результаты экспериментального исследования показывают, что погрешность оценки ухода бортовой шкалы времени НКА не зависит от длительности интервала накопления ИТНП и при прогнозировании на время 36 с (0,01 ч) не превышает требуемого значения в 0,3 м. Поскольку в реальности для достижения высоких точностей оценки эфемерид требуется от 300 до 480 с, погрешность прогнозирования ЧВП на интервале 360 с (0,1 ч) составит 0,45 м (1,5 нс).

Заключение

Повышение точности эфемеридно-временного обеспечения в соответствии с перспективными требованиями к системе ГЛОНАСС возможно за счет реализации автономной технологии оценки

ЭВИ на борту НКА ГЛОНАСС. Современные технические средства позволяют реализовать предложенную в данной статье технологию и обеспечить прогнозирование эфемерид НКА ГЛОНАСС с погрешностью менее 0,4 м (СКО) и ЧВП с погрешностью порядка 0,3 м (СКО).

В заключение необходимо отметить, что полученные результаты позволяют проводить дальнейшие исследования по совершенствованию технических средств наземного комплекса управления и бортовой аппаратуры НКА в целях улучшения модели прогнозирования ЧВП и реализации предложенной технологии ЭВО НКА ГЛОНАСС.

Список литературы

1. Дворкин В. В., Карутин С. Н. Возможность построения системы высокоточного определения местоположения потребителей глобальных навигационных спутниковых систем // Измерительная техника. 2011. № 5. С. 30–34.
2. Пасынков В. В., Данилюк А. Ю., Забокрицкий А. В. Эфемеридно-временное обеспечение ГНС ГЛОНАСС // Труды Института прикладной астрономии РАН. 2009. Вып. 20. С. 90–103.
3. Харисов В. Н., Перов А. И., Болдин В. А. и др. Глобальная спутниковая радионавигационная система ГЛОНАСС. М.: ИПРЖР, 1999. 560 с.
4. Рой Ю. А., Шаргородский В. Д., Садовников М. А., Сумерин В. В., Чубыкин А. А., Федотов А. А. Лазерные информационно-измерительные средства для системы ГЛОНАСС // Аэрокосмический курьер. 2010. № 2. С. 32–35.
5. Wills P., Fagard W., Ferrage P. and etc. The International DORIS Service (IDS): Toward maturity // Advances in Space Research. 2010. N 45. P. 1408–1420.
6. Reigber Ch., Massmann F., Fletchner F. The PRARE System and the Data Analysis Procedure // CSTG Bull. 1997. Vol. 14. P. 11–19.
7. Hoffmann-Wellenhof B., Lichtenegger H., Wastle E. GNSS — Global Navigation Satellite Systems. GPS, GLONASS, Galileo and more. Wien: Springer-Verlag, 2008. 516 p.
8. Urlichich Y., Subbotin V., Stupak G., Dvorkin V., Povaliaev A., Karutin S. GLONASS Modernization // Proc. of the 24rd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011), Portland, OR, September 2011. P. 3125–3128. URL: <http://www.ion.org>
9. de Jounge P., Tiberius C. The LAMBDA method for integer ambiguity estimation: implementation aspects. Delft Geodetic Computing Centre, 1996. 50 p.

Новая книга

И. И. Абрамов.

Лекции по моделированию элементов интегральных схем микроэлектроники. LAP LAMBERT Academic Publishing, Saarbrücken, Germany. 2012. 116 с.

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

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

УДК 004.056.53

М. В. Бочков, д-р. техн. наук,
зам. директора по научной работе,
НОУ "Центр предпринимательских рисков",
Ю. П. Лалушкин, д-р. техн. наук, консультант,
УИС Спецсвязи ФСО России,
А. А. Шкадов, канд. техн. наук, зам. нач. отделения,
НИИ "Энергия" ФСО России,
e-mail: shkadline@mail.ru

Имитационное моделирование процессов управления защищенностью ресурсов компьютерной сети на основе иерархических раскрашенных временных сетей Петри

Предложена модель управления защищенностью компьютерной сети на основе иерархических раскрашенных временных сетей Петри. Проведен анализ существующих подходов к управлению защищенностью компьютерной сети. Рассмотрены особенности применения модели на примере реализации угроз, направленных на web-приложения.

Ключевые слова: раскрашенные временные сети Петри, политика безопасности, уязвимости, управление защищенностью

Анализ существующих подходов к управлению защищенностью компьютерной сети

Анализ существующих решений [1, 2] показывает, что на сегодняшний день можно выделить два основных подхода к управлению защитой ресурсов компьютерной сети в условиях локальных или сетевых атак:

- безопасность на основе распространяемых политик;
- предотвращение вторжений путем нейтрализации уязвимостей.

В рамках первого подхода пользователь получает набор политик (ограничений), которые позволяют защититься от новой угрозы до появления сигнатур, однозначно идентифицирующих соответствующую атаку, или обновлений, исключающих последнюю.

Отличительной особенностью второго подхода является предупреждение локальных и удаленных

атак путем анализа журналов регистрации событий на территориально распределенных узлах компьютерной сети и реагирования на них в реальном времени.

Управление защитой реализуется путем активизации соответствующего признакам атаки набора правил. В рамках защиты сетевого периметра от внешних злоумышленников данный подход предусматривает использование технологии предотвращения вторжений, которая позволяет избежать попадания на компьютер вирусов и червей, а также защититься от хакерских атак. Достигается это посредством своевременной блокировки отдельных портов (например, тех, через которые на компьютер попадает опасный на данный момент червь), запрета доступа к определенным файлам и папкам.

Вышеперечисленные подходы имеют ряд существенных недостатков. При первом защищенность сети ставится в зависимость от качества услуг внешнего сервиса. Также возникает проблема смены политик вследствие их большого числа. Недостатком второго подхода является отсутствие возможности обеспечения баланса между защищенностью и функциональностью компьютерной сети в условиях воздействий злоумышленника.

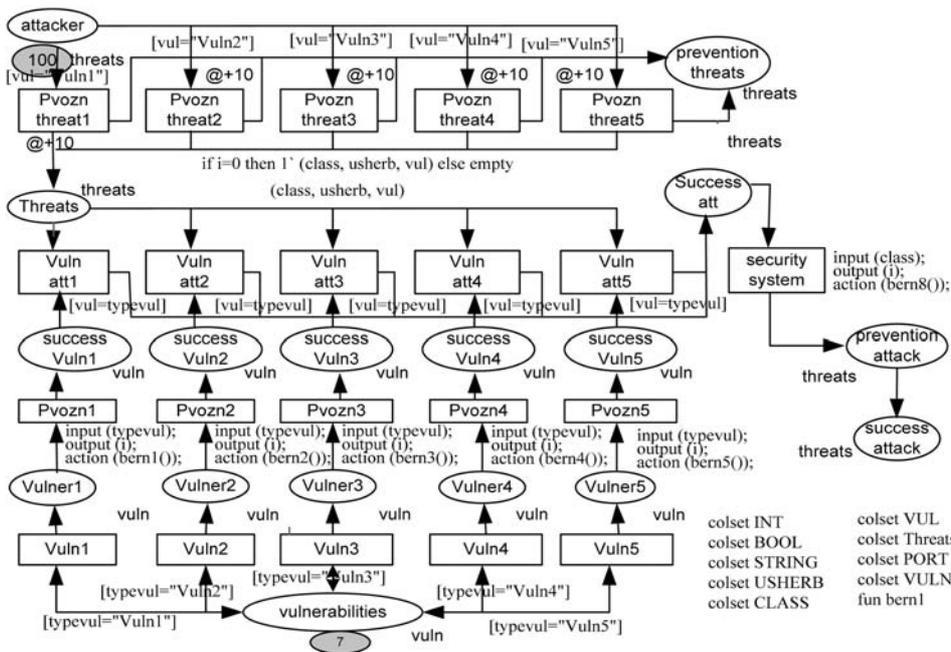
На основе вышеизложенного можно сделать вывод об актуальности создания модели, согласно которой в зависимости от возникающих уязвимостей и воздействующих на них угроз путем динамической смены политик безопасности достигался бы необходимый уровень защищенности.

Имитационная модель управления защищенностью ресурсов компьютерной сети на основе иерархических раскрашенных сетей Петри

Для формального представления имитационной модели будем использовать аппарат раскрашенных сетей Петри.

Сеть Петри [4] — это двудольный ориентированный граф с динамическими элементами — фишками. Динамика сети представляет собой процесс перемещения фишек в результате срабатывания переходов. Переход разрешен, если все его входные позиции имеют фишки. При срабатывании переход изымает фишки из своих входных позиций и помещает в свои выходные позиции. Срабатывание перехода происходит мгновенно.

Поскольку возможностей классических сетей Петри при спецификации сложных систем оказы-



Модель оценки защищенности компьютерной сети на основе иерархических раскрашенных временных сетей Петри

вается недостаточно, в аппарате сетей Петри имеются их расширения. В данной работе использован класс раскрашенных иерархических временных сетей Петри. С точки зрения наглядности восприятия сети Петри и качественного анализа моделируемых процессов, что особенно важно при разработке спецификаций как инструмента проектирования, особое место занимает графическое представление сетей Петри. Такой подход (визуального представления) является основой для графических редакторов в специализированных инструментальных программных системах моделирования сетей Петри. В комментариях к графическому представлению и в самом графическом представлении будем придерживаться системы обозначений, используемых в монографии К. Jensen [5], и инструментальной программной системы моделирования раскрашенных сетей Петри Design/CPN [6]. На рисунке представлена имитационная модель оценки защищенности сети.

Проведенную моделирования должна предшествовать начальная разметка сети, заключающаяся в определении маркировки состояний attacker и vulnerabilities.

Состояние vulnerabilities содержит фишки цвета vuln, а состояние attacker — фишки цвета threats. Фишки в состояниях attacker и vulnerabilities являются составными, содержащими несколько параметров.

Фишки цвета vuln включают в себя два параметра: тип уязвимости (typevul) и порт (port), используемый для реализации уязвимости данного типа. Фишки цвета threats включают в себя три параметра: класс угрозы, значение риска при реализации данного типа угрозы и уязвимость, используемую для реализации угрозы нарушителем. В классе уг-

роз рассматриваются угрозы нарушения целостности (cel), угрозы нарушения конфиденциальности (conf), угрозы нарушения доступности (dost) и угрозы отказа в обслуживании (DoS). Значения риска делятся на: высокий (high), средний (medium) и низкий (low).

Описание позиций и переходов данной модели отражено в табл. 1.

В соответствии с начальной маркировкой позиция attacker содержит фишки цвета threats. Переходы Pvoznatt1, Pvoznatt2, Pvoznatt3, Pvoznatt4, Pvoznatt5 являются временными. Временные задержки представляют собой равномерно распределенные случайные величины.

Данные переходы моделируют процесс возникновения угрозы со стороны нарушителя.

В случае отсутствия уязвимости в системе в результате срабатывания одного из переходов Pvoznatt фишка из позиции attacker перейдет в позицию prevention_attack. Вероятности обнаружения и предотвращения конкретного типа угроз системой защиты информации для каждого перехода задаются с помощью функции Bernoulli.

Основное внимание в модели уделено случаю, когда нарушителем обнаружена и начинает эксплуатироваться конкретная уязвимость. Данный процесс моделируется путем перевода фишки из позиции attacker в позицию Threats. Процесс выбора уязвимости конкретного типа из общего перечня уязвимостей, заданных в позиции vulnerabilities и определяемых цветом vuln, осуществляется с помощью переходов Vuln1, Vuln2, Vuln3, Vuln4 и Vuln5. В результате срабатывания перехода фишка из позиции vulnerabilities изымается и помещается в одну из позиций Vulner1, Vulner2, Vulner3, Vulner4 или Vulner5 в зависимости от типа выбранной уязвимости. Переходы Pvozn1, Pvozn2, Pvozn3, Pvozn4 и Pvozn5 моделируют процесс возникновения уязвимости конкретного типа в системе. Вероятности возникновения уязвимости конкретного типа в системе для каждого перехода задаются с помощью функции Bernoulli. Числовые значения функции Bernoulli определяются исходя из используемой политики безопасности. В результате срабатывания перехода Pvozn1 фишка из позиции Vulner1 изымается и помещается в позицию success_Vuln1. Нахождение фишки в состоянии success_Vuln1 означает наличие в системе уязвимости 1-го типа. В результате срабатывания перехода Pvozn2 фишка из позиции Vulner2 изымается и помещается в пози-

Описание состояний и переходов модуля оценки защищенности

№	Состояния/ переходы	Описание	Цвета маркеров (фишек), действие
1	2	3	4
1	attacker	Содержит множество угроз, реализуемых нарушителем	threats
2	vulnerabilities	Содержит множество уязвимостей компонентов сети	vuln
3	vuln1	Моделирует процесс выбора уязвимости 1-го типа	typevul="viln1"
4	vuln2	Моделирует процесс выбора уязвимости 2-го типа	typevul="viln2"
5	vuln3	Моделирует процесс выбора уязвимости 3-го типа	typevul="viln3"
6	vuln4	Моделирует процесс выбора уязвимости 4-го типа	typevul="viln4"
7	vuln5	Моделирует процесс выбора уязвимости 5-го типа	typevul="viln5"
8	Vulner1	Содержит уязвимость 1-го типа	vuln
9	Vulner2	Содержит уязвимость 2-го типа	vuln
10	Vulner3	Содержит уязвимость 3-го типа	vuln
11	Vulner4	Содержит уязвимость 4-го типа	vuln
12	Vulner5	Содержит уязвимость 5-го типа	vuln
13	Pvozn1	Моделирует процесс возникновения уязвимости 1-го типа в системе	input (typevul); output (i); action (bern1());
14	Pvozn2	Моделирует процесс возникновения уязвимости 2-го типа в системе	input (typevul); output (i); action (bern2());
15	Pvozn3	Моделирует процесс возникновения уязвимости 3-го типа в системе	input (typevul); output (i); action (bern3());
16	Pvozn4	Моделирует процесс возникновения уязвимости 4-го типа в системе	input (typevul); output (i); action (bern4());
17	Pvozn5	Моделирует процесс возникновения уязвимости 5-го типа в системе	input (typevul); output (i); action (bern5());
18	success_Vuln1	Наличие в системе уязвимости 1-го типа	vuln
19	success_Vuln2	Наличие в системе уязвимости 2-го типа	vuln
20	success_Vuln3	Наличие в системе уязвимости 3-го типа	vuln
21	success_Vuln4	Наличие в системе уязвимости 4-го типа	vuln
22	success_Vuln5	Наличие в системе уязвимости 5-го типа	vuln
23	Pvoznatt1	Моделирует процесс возникновения угрозы 1-го типа со стороны нарушителя	vul="vuln1"
24	Pvoznatt2	Моделирует процесс возникновения угрозы 2-го типа со стороны нарушителя	vul="vuln2"
25	Pvoznatt3	Моделирует процесс возникновения угрозы 3-го типа со стороны нарушителя	vul="vuln3"
26	Pvoznatt4	Моделирует процесс возникновения угрозы 4-го типа со стороны нарушителя	vul="vuln4"
27	Pvoznatt5	Моделирует процесс возникновения угрозы 5-го типа со стороны нарушителя	vul="vuln5"
28	threats	Содержит угрозы, реализуемые нарушителем	threats
29	prevention_threats	Угрозы предотвращенные системой защиты информации	threats
30	Vuln_att1	Моделирует процесс перехода атаки, использующей уязвимость 1-го типа в стадию активной реализации	vul=typevul
31	Vuln_att2	Моделирует процесс перехода атаки, использующей уязвимость 2-го типа в стадию активной реализации	vul=typevul
32	Vuln_att3	Моделирует процесс перехода атаки, использующей уязвимость 3-го типа в стадию активной реализации	vul=typevul
33	Vuln_att4	Моделирует процесс перехода атаки, использующей уязвимость 4-го типа в стадию активной реализации	vul=typevul
34	Vuln_att5	Моделирует процесс перехода атаки, использующей уязвимость 5-го типа в стадию активной реализации	vul=typevul
35	Success_att	Содержит атаки, находящиеся в стадии активной реализации	threats
36	security_system	Моделирует процесс обнаружения атак на уровне системы	input (class); output (i); action (bern8());
37	prevention_attack	Содержит предотвращенные атаки	threats
38	success_attack	Содержит успешные атаки	threats

цию success_Vuln2. Нахождение фишки в состоянии success_Vuln2 означает наличие в системе уязвимости 2-го типа. В результате срабатывания перехода Pvozn3 фишка из позиции Vulner3 изымается и помещается в позицию success_Vuln3. Нахождение фишки в состоянии success_Vuln3 означает наличие в системе уязвимости 3-го типа. В результате срабатывания перехода Pvozn4 фишка из позиции Vulner4 изымается и помещается в позицию success_Vuln4. Нахождение фишки в состоянии success_Vuln4 означает наличие в системе уязвимости 4-го типа. В результате срабатывания пере-

хода Pvozn5 фишка из позиции Vulner5 изымается и помещается в позицию success_Vuln5. Нахождение фишки в состоянии success_Vuln5 означает наличие в системе уязвимости 5-го типа. Переход Vuln_att1 срабатывает в случае наличия фишки в позициях Threats и success_Vuln1.

Необходимо отметить, что обязательным условием для срабатывания перехода является наличие уязвимости 1-го типа в системе и использование данной уязвимости нарушителем при реализации угрозы. Срабатывание перехода означает, что атака перешла в стадию активной реализации. В резуль-

тате срабатывания перехода фишки из позиций Threats и success_Vuln1 изымаются, а позиция Success_att получает фишку. Позиция Success_att является аккумулялирующей и содержит все атаки, находящиеся в стадии активной реализации. Переход Vuln_att2 срабатывает в случае наличия фишки в позициях Threats и success_Vuln2. Переход срабатывает при условии наличия уязвимости 2-го типа в системе и использовании данной уязвимости нарушителем при реализации угрозы. В результате срабатывания перехода фишки из позиций Threats и success_Vuln2 изымаются, а позиция Success_att получает фишку. Переход Vuln_att3 срабатывает в случае наличия фишки в позициях Threats и success_Vuln3. Переход срабатывает при условии наличия уязвимости 3-го типа в системе и использовании данной уязвимости нарушителем при реализации угрозы. В результате срабатывания перехода фишки из позиций Threats и success_Vuln3 изымаются, а позиция Success_att получает фишку. Переход Vuln_att4 срабатывает в случае наличия фишки в позициях Threats и success_Vuln4. Переход срабатывает при условии наличия уязвимости 4-го типа в системе и использовании данной уязвимости нарушителем при реализации угрозы. В результате срабатывания перехода фишки из позиций Threats и success_Vuln4 изымаются, а позиция Success_att получает фишку. Переход Vuln_att5 срабатывает в случае наличия фишки в позициях Threats и success_Vuln5. Переход срабатывает при условии наличия уязвимости 5-го типа в системе и использовании данной уязвимости нарушителем при реализации угрозы. В результате срабатывания перехода фишки из позиций Threats и success_Vuln5 изымаются, а позиция Success_att получает фишку.

Возможность обнаружения атаки на уровне системы описана с помощью перехода security_system. В результате срабатывания перехода фишка из позиции success_att может перейти в позицию prevention_attack или в позицию success_attack. Переход фишки в позицию prevention_attack означает, что атака, реализуемая нарушителем, успешно отражена. В том случае, когда атака прошла успешно, фишка перейдет в позицию success_attack.

В представленной модели отражены наиболее часто встречающиеся уязвимости web-приложений, распределение которых по типам и вероятности появления $P_{возн}$ представлено в табл. 2.

Вероятности появления уязвимостей моделируются с помощью переходов Pvozn1, Pvozn2, Pvozn3, Pvozn4, Pvozn5, при этом числовые значения задаются с помощью функции Bernoulli. Вероятности $P_{угр}$ возникновения угроз, использующих перечисленные типы уязвимостей, представлены в табл. 3.

Моделирование системы защиты информации (СЗИ) ограничено одним из наиболее распространенных классов — сканером защищенности и представлено в модели переходом security_system.

Вероятность обнаружения угрозы современными сканерами составляет 0,7 [8].

Множество политик безопасности, используемых для защиты от моделируемых угроз, включает следующие:

- политики нейтрализации уязвимости SQL injection;
- политики нейтрализации уязвимости Cross-Site Scripting;
- политики нейтрализации уязвимости Directory Indexing;
- политики нейтрализации уязвимости Information Leakage;
- политики нейтрализации уязвимости HTTP Response Splitting.

Политики безопасности для каждого класса представляют собой правила по предотвращению возможности использования вышеперечисленных типов уязвимостей.

Политики безопасности меняются в зависимости от реализуемой нарушителем в данный момент времени угрозы. Для предложенной модели адекватное угрозе применение политики безопасности приводит к изменению вероятности успешной реализации угрозы, использующей соответствующую уязвимость. Основные результаты моделирования представлены в табл. 4.

В табл. 4 представлены экспериментальные значения следующих показателей, отражающих уровень защищенности ресурсов сети:

- $t_{\text{мод}}$ — время моделирования;
- $N_{\text{общ}}$ — общее число проведенных атак;

Таблица 2

Тип уязвимости	$P_{\text{возн}}$
Cross-Site Scripting	0,47
Information Leakage	0,23
SQL injection	0,19
HTTP Response Splitting	0,08
Directory Indexing	0,03

Таблица 3

Тип уязвимости	$P_{\text{угр}}$
Cross-Site Scripting	0,46
Information Leakage	0,23
SQL injection	0,15
HTTP Response Splitting	0,11
Directory Indexing	0,05

Таблица 4

Политики	$t_{\text{мод}}$	$N_{\text{общ}}$	$N_{\text{СЗИ}}$	$N_{\text{усп}}$	P_p
Отсутствуют	60 мин	100	83	17	0,17
Присутствуют	60 мин	100	96	4	0,04

$N_{СЗИ}$ — суммарное число атак отраженных системой защиты информации (включая сканер защищенности);

$N_{усп}$ — число успешных атак;

P_p — вероятность успешной реализации угрозы.

Таким образом, в работе предложен подход к управлению защищенностью компьютерной сети на основе политик безопасности. Разработана имитационная модель, позволяющая проводить оценку защищенности ресурсов web-приложений в условиях возникающих угроз при использовании динамически изменяемых политик безопасности. Направлением дальнейших исследований является адаптация механизма шаблонов безопасности операционных систем семейства Windows для реализации изложенного в статье подхода к управлению защищенностью.

Список литературы

1. Доля А. Проактивные технологии для борьбы с вирусами // Экспресс Электроника. 2006. № 5. С. 56—61.
2. Котенко И. В., Юсупов Р. М. Перспективные направления исследований в области компьютерной безопасности // Защита информации. Инсайд. 2006. № 2. С. 46—57.
3. Шишкин В. М. Метамоделю анализа, оценки и управления безопасностью // Сб. тр. Института системного анализа Российской академии наук. М.: Едиториал УРСС 2002. С. 92—105.
4. Котов В. Е. Сети Петри. М.: Наука, 1984. 160 с.
5. Jensen K. Colored Petri Nets — Basic Concepts, Analysis Methods and Practical Use. Springer—Verlag, 1997. Vol. 1—3.
6. Albert K., Jensen K. Design/CPN: A Tool Package Supporting the use of Colored Nets // Petri Net Newsletter, April, 1989. P. 22—35.
7. www.securitylab.ru Статистика уязвимости web-приложений за 2007 г.
8. Мак-Клар С., Скембрей Дж., Курц Дж. Секреты хакеров. Безопасность сетей — готовые решения. М.: Вильямс, 2004.

УДК 621.391.18:004.056.53

А. В. Козленко, адъюнкт,
Военная академия связи, г. Санкт-Петербург,
e-mail: et-ak@yandex.ru

Метод оценки защищенности информации от несанкционированного доступа в компьютерных сетях в условиях комплексных угроз

Показана актуальность проблемы количественной оценки защищенности информации от несанкционированного доступа в компьютерных сетях. Предложен метод оценки защищенности информации от несанкционированного доступа в компьютерных сетях на основе графа защищенности. Предложенный метод обеспечивает повышение эффективности управления защитой информации в компьютерных сетях за счет комплексного показателя защищенности, а также применением графа защищенности, учитывающего реальную структуру компьютерной сети и системы защиты информации.

Ключевые слова: оценка защищенности, несанкционированный доступ, компьютерная сеть, коэффициент защищенности, граф защищенности

Введение. Нарушение нормального функционирования компьютерных сетей большинства промышленных, правительственных, банковских, оборонных и правоохранительных организаций вследствие несанкционированного доступа (НСД) к обрабатываемой в них информации влечет за собой

значительный ущерб. Достижение научно обоснованного уровня защищенности информации от НСД обуславливает необходимость проведения оценки защищенности на всех этапах жизненного цикла компьютерных сетей при различной степени полноты и достоверности имеющейся информации.

Цель работы — разработка метода оценки защищенности информации от НСД в компьютерных сетях, обеспечивающего повышение эффективности управления защитой информации в организациях за счет комплексного показателя, учитывающего как характеристики процесса нарушений безопасности, так и характеристики процесса защиты, а также за счет применения графа защищенности, учитывающего реальную структуру компьютерной сети и системы защиты информации (СЗИ).

Основная часть. На современном этапе становление научного направления, связанного с исследованием и оценкой защищенности информации от НСД в компьютерных сетях, сдерживается отсутствием единого понятийного аппарата в области защиты информации от НСД, преобладанием в руководящих документах качественных подходов к оценке защищенности, ориентацией на статические условия функционирования систем защиты.

Аттестация АС, сертификация СВТ согласно требованиям действующих в РФ нормативных документов (руководящих документов Федеральной службы по техническому и экспортному контролю РФ, ГОСТ Р ИСО/МЭК 15408—2008, ГОСТ Р ИСО/МЭК 17799—2005) требует высокой квалификации персонала, обработки больших объемов данных и значительных временных затрат.

В условиях стремительного развития информационных технологий постоянно возникают новые

виды угроз, что, в свою очередь, приводит к появлению новых уязвимостей программного и аппаратного обеспечения как самих компьютерных сетей, так и средств защиты информации. В таких условиях более перспективными являются количественные методы оценки защищенности информации от НСД. Среди известных отечественных и зарубежных методик количественной оценки защищенности информации от НСД наибольшее распространение получил подход на базе анализа информационных рисков [1] (например модель системы обеспечения безопасности Клементса [2, 3]). На основе анализа рисков созданы такие средства оценки защищенности как Microsoft Baseline Security Analyzer (MBSA) [4], CRAMM, Counter Measures, VCM-Analyser [5], а также отечественные "Триф" и "РискМенеджер" [5]. Однако у данного подхода есть ряд существенных недостатков, а именно: не учитывается реальная структура компьютерной сети; оценивается стоимость потерь от несанкционированного доступа к информации в денежных единицах, что приемлемо не для всех организаций; не полностью учитывается вариативность сценариев реализации НСД и динамические характеристики процесса защиты информации.

Защищенность информации в компьютерной сети от НСД определяется защищенностью каждого защищаемого ресурса. Для решения задачи оценки защищенности информации от НСД целесообразно использовать комплексные показатели защищенности, учитывающие как процессы нарушения безопасности ресурсов в компьютерной сети, так и процессы контроля и восстановления их защищенного состояния. В качестве такого показателя будем

использовать коэффициент защищенности информации от НСД, аналогичный используемому в теории надежности коэффициенту готовности [6]. Аналогичный подход был предложен А. Ю. Щегловым [7], однако в этом подходе не учитывается структура компьютерной сети, а также характеристики различных типов угроз безопасности информации и характеристики соответствующих средств защиты. В частности, параметры потоков нарушений безопасности ресурсов компьютерной сети (уязвимостей) и характеристики восстановления защищенного состояния в одной компьютерной сети будут отличаться для конфиденциальности, целостности и доступности ресурсов компьютерной сети.

В этот же период подобный подход был предложен А. И. Костокрызовым (ГОСТ РВ 51987—2002), имеющий схожие недостатки.

Предлагаемый метод оценки защищенности информации от НСД в компьютерных сетях состоит из следующих этапов (рис. 1):

- 1) сбор и анализ исходных данных;
- 2) расчет показателей защищенности отдельных ресурсов;
- 3) построение графов защищенности по типам угроз безопасности информации;
- 4) расчет показателей защищенности информации в компьютерной сети;
- 5) оценка защищенности информации от НСД в компьютерной сети (по выбранному критерию);
- 6) выработка рекомендаций по обеспечению требуемого уровня защищенности информации от НСД в компьютерной сети должностным лицам, ответственным за защиту информации в организации.

Исходными данными для оценки защищенности информации от НСД в компьютерной сети являются:

- перечень защищаемых ресурсов (какие ресурсы являются критически важными для выполнения задач в компьютерной сети, их количество, на каких серверах и (или) автоматизированных рабочих местах (АРМ) они расположены);
- состав и параметры функционирования средств защиты ресурсов (количество средств защиты ресурсов, на каких серверах и (или) АРМ они расположены, какие конкретно ресурсы они защищают, каким образом организована защита);
- интенсивность нарушений безопасности ресурсов. В расчете на худший случай, когда нарушитель "идеальный" (имеет высокую квалификацию, постоянно отслеживает появление новых уязвимостей, а также имеет возможность мгновенно



Рис. 1. Метод оценки защищенности информации от НСД в компьютерных сетях

использовать их для осуществления НСД к информации, обрабатываемой в компьютерной сети организации), интенсивность нарушений безопасности ресурсов компьютерной сети соответствует интенсивности появления уязвимостей в программно-аппаратном обеспечении компьютерной сети. Интенсивность нарушений безопасности ресурсов целесообразно рассчитывать отдельно для трех основных типов угроз безопасности информации — угроз конфиденциальности, целостности и доступности;

- интенсивность восстановления защищенности ресурсов. Данная интенсивность зависит от числа должностных лиц, ответственных за защиту информации в организации, и от уровня их квалификации. Интенсивность восстановления защищенности ресурсов также целесообразно рассчитывать отдельно для трех основных типов угроз безопасности информации.

Также на этапе анализа исходных данных проводится анализ структуры, параметров и алгоритмов функционирования компьютерной сети.

После анализа исходных данных выполняется расчет показателей защищенности отдельных ресурсов. При этом для расчета коэффициентов защищенности ресурса можно использовать следующие выражения:

$$K_{зщ}^к = \frac{\mu_{в.з}^к}{\lambda_{н.б}^к + \mu_{в.з}^к}, K_{зщ}^ц = \frac{\mu_{в.з}^ц}{\lambda_{н.б}^ц + \mu_{в.з}^ц},$$

$$K_{зщ}^д = \frac{\mu_{в.з}^д}{\lambda_{н.б}^д + \mu_{в.з}^д}, \quad (1)$$

где $K_{зщ}^к$, $K_{зщ}^ц$, $K_{зщ}^д$ — коэффициенты защищенности ресурсов от угроз конфиденциальности, целостности и доступности соответственно; $\lambda_{н.б}^к$, $\lambda_{н.б}^ц$, $\lambda_{н.б}^д$ — интенсивности нарушений конфиденциальности, целостности и доступности ресурсов соответственно; $\mu_{в.з}^к$, $\mu_{в.з}^ц$, $\mu_{в.з}^д$ — интенсивности восстановления защищенности для конфиденциальности, целостности и доступности ресурсов соответственно.

Запишем выражения (1) в виде

$$K_{зщ}^{к,ц,д} = \frac{\mu_{в.з}^{к,ц,д}}{\lambda_{н.б}^{к,ц,д} + \mu_{в.з}^{к,ц,д}}, \quad (2)$$

и в дальнейшем будем пользоваться такой формой записи.

Для расчета показателей защищенности всей информации в компьютерной сети необходимо построить графы защищенности по типам угроз безопасности информации. Под графом защищеннос-

ти компьютерной сети понимается совокупность связанных элементов, характеризующих состояние защищенности ее ресурсов [8].

Граф защищенности строится на основе анализа структуры, параметров и алгоритмов функционирования компьютерной сети, проведенном при анализе исходных данных. Данный этап требует высокой квалификации персонала и относительно больших временных затрат, поэтому основной объем работ по построению графов защищенности по типам угроз безопасности информации рекомендуется проводить на этапе проектирования компьютерной сети. В этом случае на этапе эксплуатации компьютерной сети для получения графов защищенности по типам угроз безопасности информации необходимо уточнить графы защищенности, полученные на этапе проектирования, в соответствии с изменениями структуры, параметров и алгоритмов функционирования компьютерной сети, произошедшими к данному моменту эксплуатации (изменение серверов и (или) АРМ, изменение состава и параметров функционирования средств защиты ресурсов и т. д.). При анализе компьютерной сети на предмет защищенности, в первую очередь, следует учитывать наличие альтернативных средств защиты, а также наличие эквивалентных по пригодности для процесса достижения цели функционирования компьютерной сети ресурсов и (или) способов их использования, при этом следует руководствоваться следующими правилами [8].

1. Если для достижения цели функционирования компьютерной сети требуется защищенное состояние всех ресурсов (нарушение функционирования компьютерной сети происходит при нарушении безопасности хотя бы одного ресурса) и при этом для защиты ресурсов не используются дополнительные альтернативные средства защиты (используемое средство или несколько средств защиты преодолеваются за счет использования уязвимости одного типа в одном или нескольких средствах защиты), а также отсутствуют альтернативные с точки зрения защищенности резервные ресурсы (например альтернативные по уязвимостям резервные программные и технические средства автоматизации), то такой компьютерной сети ставится в соответствие последовательный граф защищенности, элементы которого соответствуют защищаемым ресурсам.

2. Если для защиты ресурса кроме основного используются дополнительные средства защиты, альтернативные основному по способу защиты от реализации угроз, то такому ресурсу соответствует параллельный граф защищенности, число элементов в котором соответствует общему числу средств защиты данного ресурса.

3. Если при нарушении безопасности ресурса существуют (предусмотрены) варианты решения соответствующей задачи компьютерной сети с ис-

пользованием другого альтернативного ресурса (защищаемый ресурс компьютерной сети резервирован структурно, функционально и т. д.), то при допущении о нарушении безопасности в один момент времени только в отношении одного ресурса основной и резервные ресурсы образуют резервированную группу, им ставится в соответствие параллельный граф защищенности.

В общем случае граф защищенности компьютерной сети может содержать как последовательные, так и параллельные соединения. Графы защищенности целесообразно строить отдельно для основных типов угроз безопасности обрабатываемой в компьютерной сети информации — угроз конфиденциальности, целостности и доступности.

На основе построенных графов защищенности по типам угроз безопасности информации и рассчитанных показателей защищенности отдельных ресурсов проводится этап расчета показателей защищенности информации в компьютерной сети.

Выражения для расчета показателей защищенности информации в компьютерной сети для общего случая при допущении о том, что нарушения безопасности отдельных ресурсов являются независимыми событиями, можно получить, используя теорему умножения для независимых событий. При этом для последовательного соединения графа защищенности

$$K_{\text{зщ КС}}^{К,Ц,Д} = \prod_{i=1}^{N_{3,п}} K_{\text{зщ } i}^{К,Ц,Д}, \quad (3)$$

где $K_{\text{зщ КС}}^{К,Ц,Д}$ — коэффициенты защищенности компьютерной сети (КС); $N_{3,п}$ — число защищаемых ресурсов; $K_{\text{зщ } i}^{К,Ц,Д}$ — коэффициенты защищенности i -го ресурса.

Для параллельного соединения графа защищенности

$$K_{\text{зщ КС}}^{К,Ц,Д} = 1 - \prod_{i=1}^{N_{3,п}} (1 - K_{\text{зщ } i}^{К,Ц,Д}). \quad (4)$$

Для последовательного соединения при неограниченных ресурсах на восстановление защищенности для расчета коэффициентов защищенности может быть использована следующая формула:

$$K_{\text{зщ КС}}^{К,Ц,Д} = \prod_{i=1}^{N_{3,п}} \left(\frac{\mu_{В.3}^{К,Ц,Д}}{\lambda_{Н.б}^{К,Ц,Д} + \mu_{В.3}^{К,Ц,Д}} \right).$$

При ограниченных ресурсах на восстановление защищенности (в один момент времени восстанавливается защищенность только одного ресурса)

для последовательного соединения коэффициенты защищенности рассчитываются по формуле

$$K_{\text{зщ КС}}^{К,Ц,Д} = \frac{1}{\sum_{i=1}^{N_{3,п}} \frac{N_{3,п}!}{(N_{3,п} - i)!} \left(\frac{\lambda_{Н.б}^{К,Ц,Д}}{\mu_{В.3}^{К,Ц,Д}} \right)^i}.$$

При резервировании ресурсов (осуществляемого, например, в целях обеспечения целостности и (или) доступности) и применении альтернативных средств защиты основных и резервных ресурсов граф защищенности имеет последовательно-параллельную структуру. При постоянном использовании одинакового числа альтернативных средств защиты ($N_{а.с.3}$) дополнительно к основным ($N_{с.3}$), а также при одинаковых интенсивностях нарушения безопасности ресурсов и интенсивностях восстановления защищенности ресурсов расчетная формула имеет следующий вид:

$$K_{\text{зщ КС}}^{К,Ц,Д} = \left(1 - \left(1 - \frac{\mu_{В.3}^{К,Ц,Д}}{\lambda_{Н.б}^{К,Ц,Д} + \mu_{В.3}^{К,Ц,Д}} \right)^{N_{с.3} + N_{а.с.3}} \right)^{N_{3,п}}. \quad (5)$$

После расчета показателя защищенности информации в компьютерной сети проводится оценка защищенности информации от НСД по принятому критерию, после чего выявляются "узкие" места в системе защиты. Затем на основе полученных результатов разрабатываются рекомендации по обеспечению требуемого уровня защищенности информации в компьютерной сети должностным лицам, ответственным за защиту информации в организации.

Приведем пример оценки защищенности информации от НСД на основе предлагаемого метода. Рассмотрим вариант типичной компьютерной сети организации, построенной на основе локальной вычислительной сети. Каждый сотрудник организации имеет рабочую станцию, функционирующую под управлением операционной системы Microsoft Windows 7, на которой находятся его пользовательские данные. В организации имеется 50 таких рабочих станций. Рабочие станции объединены в компьютерную сеть с тремя серверами на базе операционной системы Microsoft Windows 2008 Server, на которых функционирует общесистемное и прикладное ПО организации (почтовый сервер, СУБД, Web-сервер предприятия, мгновенная система обмена сообщениями для сотрудников и т. д.). Два сервера объединены в отказоустойчивый кластер, третий сервер — сервер единого времени. Для выхода в Интернет используется компьютер-шлюз с установленным межсетевым экраном (МСЭ) Microsoft FireFront. Компьютерная сеть функционирует в условиях комплексных угроз, следовательно, будем рассчитывать коэффициенты защищенности для внутренних и внешних нарушителей отдельно.

Проведем анализ исходных данных. Критичные с точки зрения безопасности информации ресурсы расположены следующим образом. Каждая рабочая станция (АРМ) содержит один критически важный защищаемый ресурс — пользовательские данные. Два сервера (С-1 и С-2), объединенных в отказоустойчивый кластер, обеспечивают функционирование СУБД (Microsoft SQL Server 2008), почтового сервера организации (Microsoft Exchange Server 2010), Web-сервера организации и мгновенной системы обмена сообщений для сотрудников, следовательно, они содержат четыре защищаемых ресурса. Третий сервер (С-3) выделен для функционирования системы единого времени — один защищаемый ресурс. Для защиты указанных ресурсов от НСД используют одно основное средство защиты (СЗ) информации (для каждого сотрудника и администратора сервера предусмотрен личный пароль для входа в операционную систему) и одно дополнительное альтернативное (АСЗ) (аутентификация сотрудников и администраторов серверов по смарт-карте). Для защиты компьютера-шлюза от НСД используется одно средство защиты — межсетевой экран (МСЭ).

Для определения интенсивности нарушений безопасности ресурсов компьютерной сети организации был проведен анализ общедоступной статистики по обнаружению уязвимостей в ОС Windows и МСЭ MS FireFront за последний год. Использовалась база уязвимостей Open Source Vulnerability Database [9], база уязвимостей National Vulnerability Database [10] и система оценки уязвимостей CVSS [11]. Учитывались только те уязвимости, для которых предложены оценки CVSS, рассматривались показатели "ConfImpact" (влияние на конфиденциальность ресурса при успешном использовании уязвимости), "IntegImpact" (влияние на целостность), "AvailImpact" (влияние на доступность) для каждой учитываемой уязвимости. При этом считалось, что два нарушения безопасности ресурса, для которых значение одного из показателей "ConfImpact", "IntegImpact" или "AvailImpact" равно "Partial" (частичное), эквивалентны одному нарушению безопасности ресурса, для которого соответствующее значение показателя "ConfImpact", "IntegImpact" или "AvailImpact" равно "Complete" (полное). Анализ показал, что для ОС Windows интенсивность нарушений конфиденциальности информации составляет приблизительно 9,96 нарушений в месяц ($\lambda_{н.б}^k = 0,0138/ч$), интенсивность нарушений целостности информации — 9,75 нарушений в месяц ($\lambda_{н.б}^ц = 0,0135/ч$), интенсивность нарушений доступности информации — 10,04 нарушений в месяц ($\lambda_{н.б}^д = 0,0139/ч$). Для МСЭ значения интенсивностей нарушений безопасности ресурсов сле-

дующие: $\lambda_{н.б}^k = 2,028 \cdot 10^{-4}/ч$, $\lambda_{н.б}^ц = 3,75 \cdot 10^{-4}/ч$ и $\lambda_{н.б}^д = 2,028 \cdot 10^{-4}/ч$. Предположим, что должностные лица организации, ответственные за защиту информации, имеют численный состав и квалификацию, позволяющие восстанавливать конфиденциальность, целостность и доступность любого числа ресурсов за 3 ч.

Для расчета показателей защищенности отдельных ресурсов будем использовать формулу (2). Например, коэффициенты защищенности для пользовательских данных на рабочих станциях (РС) — $K_{зщ\ РС}^k = 0,96$; $K_{зщ\ РС}^ц = 0,961$; $K_{зщ\ РС}^д = 0,96$.

Построим графы защищенности компьютерной сети организации от угроз конфиденциальности, целостности и доступности информации по указанным правилам. Полученные графы защищенности представлены на рис. 2, 3 и 4 соответственно. Графы защищенности от внутренних нарушителей следует рассматривать без пунктирных линий, от внешних — с пунктирными линиями.

Применяя правила расчета показателя защищенности информации в компьютерной сети с использованием значений показателей защищенности отдельных ресурсов, и на основе построенных графов защищенности по типам угроз безопасности информации (выражения (3), (4) и (5)) получим следующие коэффициенты защищенности компьютерной сети: от внутренних нарушителей — $K_{зщ\ КС}^k = 0,842$, $K_{зщ\ КС}^ц = 0,847$, $K_{зщ\ КС}^д = 0,999$; от внешних нарушителей — $K_{зщ\ КС}^k = 0,999$, $K_{зщ\ КС}^ц = 0,999$, $K_{зщ\ КС}^д = 0,998$. Контроль защищенности информации, обрабатываемой в компью-

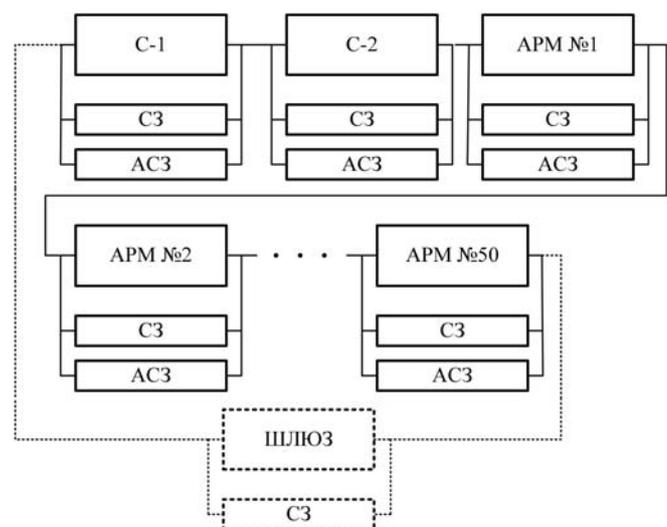


Рис. 2. Граф защищенности компьютерной сети организации от угроз конфиденциальности информации

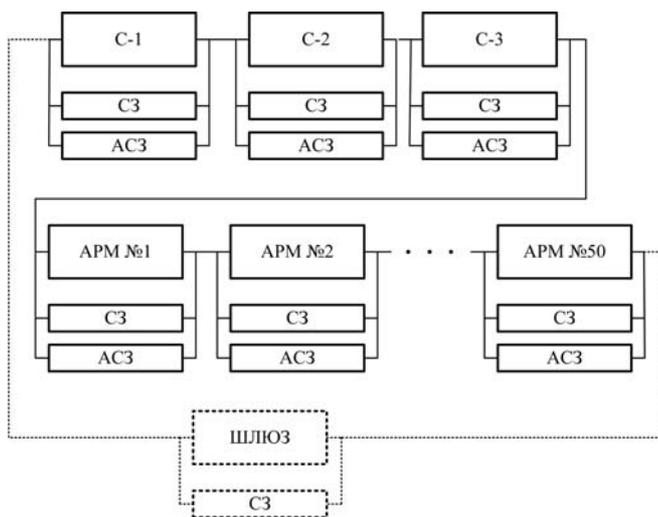


Рис. 3. Граф защищенности компьютерной сети организации от угроз целостности информации

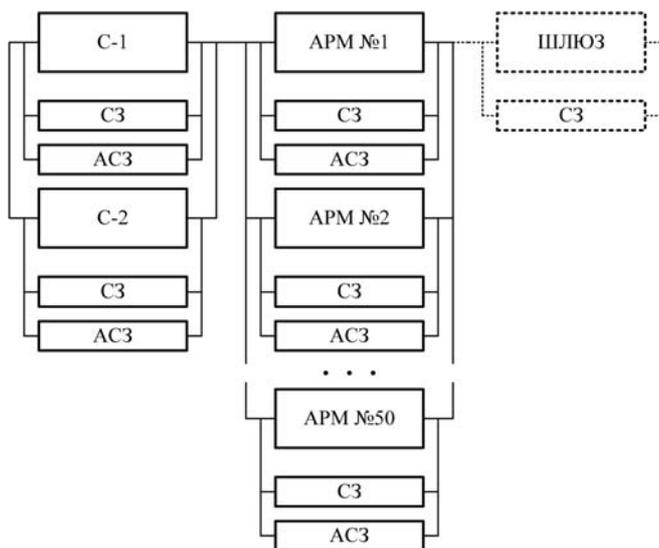


Рис. 4. Граф защищенности компьютерной сети организации от угроз доступности информации

терной сети организации, по критерию пригодности

$K_{\text{зщ}}^{\text{к,ц,д}} > 0,99$ позволяет сделать вывод о том, что система защиты информации от НСД в компьютерной сети организации не позволяет обеспечить требуемый уровень защищенности от внутренних нарушителей (требуемый уровень защищенности достигнут лишь для одного типа угроз безопасности информации). Для достижения требуемого уровня защищенности необходимо провести комплексные организационно-технические мероприятия. При добавлении дополнительного альтернативного средства защиты информации (например, контроль доступа по отпечатку пальца отдельным программно-аппаратным средством) коэффициенты защищенности от внутренних нарушителей соответственно равны: $K_{\text{зщ}}^{\text{к}} = 0,993$, $K_{\text{зщ}}^{\text{ц}} = 0,994$,

$K_{\text{зщ}}^{\text{д}} = 0,999$. А при повышении квалификации должностных лиц, ответственных за защиту информации, до уровня, при котором они смогут восстанавливать защищенность ресурсов за 30 мин, коэффициенты защищенности от внутренних нарушителей повышаются до следующих значений:

$$K_{\text{зщ}}^{\text{к}} = 0,995, K_{\text{зщ}}^{\text{ц}} = 0,995, K_{\text{зщ}}^{\text{д}} = 0,999.$$

Выводы. Для решения задачи количественной оценки защищенности информации от НСД в компьютерных сетях разработан метод на основе графа защищенности, который в отличие от известных учитывает динамику и стохастическую неопределенность всех основных процессов защиты информации, а также реальную структуру компьютерной сети и системы защиты информации при расчете показателей защищенности; обеспечивает возможность получения количественных оценок защищенности для различных типов угроз безопасности информации как на этапе разработки компьютерных сетей, так и на этапе эксплуатации при условии решения задачи количественной оценки времени восстановления защищенности ресурсов. Решение данной задачи можно определить в качестве направления дальнейших исследований

По мнению авторов, цель работы достигнута. Данный подход может представлять интерес как для разработчиков систем защиты информации, так и для должностных лиц, ответственных за защиту информации от НСД в организациях.

Список литературы

1. Шумский А. А., Шелупанов А. А. Системный анализ в защите информации: учеб. пособие для студентов вузов, обучающихся по специальностям в обл. информ. безопасности. — М.: Гелиос-АРВ, 2005. 224 с.
2. Анищенко В. В., Криштофик А. М. О необходимости разработки моделей защищенности объектов информационных технологий // Информатика. 2005. № 1.
3. Астахов А. Анализ защищенности корпоративных автоматизированных систем // URL: http://www.cobit.ru/security/Pubs/Publ_AAM_SecEval.htm/
4. Microsoft Baseline Security Analyzer (MBSA) [электронный ресурс]. URL: <http://www.microsoft.com/technet/security/tools/mbsahome.mspx>
5. Сафонов А. Практическое применение методов и средств анализа рисков // Информационная безопасность. 2010. № 3. С. 42–43.
6. Черкесов Г. Н. Надежность аппаратно-программных комплексов: учеб. пособие. — СПб.: Питер, 2005. 479 с.
7. Щеглов А. Ю. Защита компьютерной информации от несанкционированного доступа. СПб.: Наука и техника, 2004. 384 с.
8. Авраменко В. С. Модели защищенности информации от несанкционированного доступа в многорежимных автоматизированных системах и методы ее контроля в условиях неопределенности угроз // Информация и космос. 2008. № 2. С. 87–95.
9. Open Source Vulnerability Database (OSVDB) [электронный ресурс]. URL: <http://osvdb.org/>
10. National Vulnerability Database (NVD) [электронный ресурс]. URL: <http://nvd.nist.gov/>
11. Common Vulnerability Scoring System (CVSS) [электронный ресурс]. URL: <http://www.first.org/cvss>

КОМПЬЮТЕРНАЯ ГРАФИКА И ГЕОМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ

УДК 004.93

В. А. Бобков, д-р. техн. наук, зав. лаб.,

e-mail: bobkov@iacp.dvo.ru,

Ю. И. Роншин, вед. инж.,

e-mail: ronshin@iacp.dvo.ru,

В. Ю. Машенцев, аспирант,

e-mail: v.mashentsev@gmail.com,

Институт автоматики

и процессов управления ДВО РАН

Построение карт глубин на множественной последовательности видов¹

Предложена модификация ранее разработанного авторами алгоритма построения карт глубин по множественной последовательности изображений с реализацией параллелизма на языке CUDA. Алгоритм основывается на применении sparse-sweep-стратегии, адаптивной методики сопоставления окон и учета связности точек. Получены оценки эффективности алгоритма по результатам вычислительных экспериментов.

Ключевые слова: множественная последовательность изображений, карта глубин, сопоставление точек, CUDA

В настоящей статье представлена усовершенствованная версия алгоритма построения карты глубин по изображениям с использованием графических процессоров, ранее описанного авторами в работах [1, 2]. Там же был дан и обзор предшествующих работ в этой области. Более подробный обзор по методам построения карт глубин можно найти, например, в работе [3]. Под термином "карта глубин" понимаются дальностные данные, связанные с изображением, т. е. расстояния до видимых на изображении объектов в узлах плотной сетки, накладываемой на изображение. В литературе на них также ссылаются как на range images, depth images, xyz maps, surface profiles and 2.5D images. Эти данные, полученные или прямым образом лазерными измерителями, или косвенным образом с помощью алгоритмов обработки двух и более изобра-

жений, используются уже непосредственно для построения триангуляционной модели объектов сцены.

Статья организована следующим образом. В разделе "Методика построения 3D-модели на длинной последовательности видов" описывается применяемая схема обработки видов для получения 3D-модели, оптимизирующая достигаемую точность реконструкции и общее время вычислений. В разделе "Базовая версия алгоритма построения карт глубин" дается краткое описание основных алгоритмических решений начальной версии алгоритма [1, 2]. В разделе "Модификация алгоритма" обсуждаются некоторые недостатки базовой версии алгоритма, выявленные в экспериментах, и предлагаются дополнительные алгоритмические решения, направленные на повышение точности получаемых карт глубин и минимизацию вычислительных трудозатрат. Здесь же представлена конечная схема алгоритма с распределением вычислений между центральным процессором и графическими процессорами (параллельные вычисления на языке CUDA). Дано описание настраиваемых параметров и возможных режимов работы алгоритма, позволяющих регулировать соотношение "точность — время расчета". В разделе "Эксперименты" обсуждаются результаты тестирования системы.

Методика построения 3D-модели на длинной последовательности видов

Для построения 3D-модели объектов сцены используется длинная последовательность калиброванных видов (изображений/кадров) с короткой базой между видами, которая обеспечивает детальный обзор сцены. 3D-модель строится в два этапа. Вначале строятся карты глубин для опорных видов, которые выбираются из последовательности с некоторым интервалом. Для построения карты глубины опорного вида используют несколько примыкающих к нему кадров, которые будем называть *последовательностью опорного вида*. В зависимости от соотношения выбранной длины интервала и выбранной длины последовательности опорного вида две смежные последовательности могут смыкаться или перекрываться (рис. 1). На втором этапе карты глубин опорных видов объединяются, создавая плотное облако 3D-точек, по которому строится полная триангуляционная модель сцены. Преимущество такой схемы обработки в отличие, например,

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 11-07-00088-а) и Программы № 18 Президиума РАН.

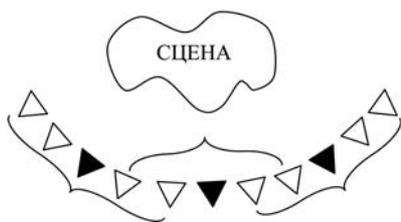


Рис. 1. Последовательности опорных видов из пяти изображений с перекрытием. Темным цветом выделены опорные виды

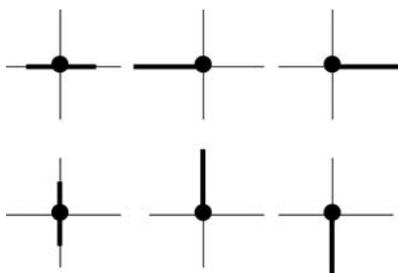


Рис. 2. Сдвигаемые линейные окна

от неплотной последовательности кадров (длинная база между видами), где каждый кадр является опорным, заключается в том, что, с одной стороны, в короткой и плотной последовательности опорного вида число полувидимых (видимых не на всех кадрах) точек невелико. А с другой стороны, за счет объединения множества опорных видов число невидимых точек в сцене при полном обзоре тоже минимально. Таким образом, число дефектов, связанных с полувидимостью точек, снижается.

Базовая версия алгоритма построения карт глубин

Подробное описание ранее реализованного алгоритма можно найти в работах [1, 2], поэтому здесь мы ограничимся его краткой характеристикой с анализом недостатков, которые преодолеваются в предлагаемой новой версии алгоритма. Алгоритм базируется на авторской модификации метода sparse-sweep [4], согласно которому поиск правильной глубины для точки (пикселя) осуществляется перебором глубин на луче (из центра проекций через рассматриваемую точку) с выбором наилучшей вычисленной оценки сходства (текстурной близости) проекций 3D-точки на имеющиеся виды. Оценкой служит часто применяемая квадратичная оценка разностей интенсивностей цветов соответствующих точек (пикселей) окна на опорном виде и образа этого окна на другом виде:

$$v = \Sigma(rp_{1i} - rp_{2i})^2 + (gp_{1i} - gp_{2i})^2 + (bp_{1i} - bp_{2i})^2,$$

где rp_{1i} , gp_{1i} , bp_{1i} — компоненты цвета i -й точки 1-го окна; rp_{2i} , gp_{2i} , bp_{2i} — компоненты цвета соответствующей точки 2-го окна.

Особенностями реализованной алгоритмической схемы являются:

а) алгоритм работает в два этапа — на первом этапе поиск правильной глубины осуществляется с грубым шагом по лучу, на втором этапе выполняется проверка с мелким шагом в малой окрестности найденной глубины. Это позволяет уменьшить объем вычислений;

б) проверка множества гипотез-нормалей к поверхности объекта (в отличие от аналогов, где обычно предполагалась одна или три нормали, например, в [5]), что позволяет выбрать наиболее оптимальную нормаль (близкую к истинному наклону поверхности);

в) использование "линейного" окна (отрезка) вместо традиционного прямоугольного окна, что существенно сокращает время вычислений без заметной потери качества. Используются два взаимно ортогональных направления для линейных окон;

г) использование плоской гомографии для вычисления правильной геометрии окна на других видах. Существенно, что в случае "линейного" окна гипотезы-нормали берутся на полукружности вместо полушеры при прямоугольном окне, что также сокращает объем вычислений;

д) применение методики сдвигаемых окон [6]. Методика заключается в проверке оценок для нескольких окон в окрестности рассматриваемой точки (рис. 2) и выборе наилучшей из них. Предполагается, что лучшей оценке соответствует гладкий участок поверхности (без разрывности глубины).

Методика направлена на повышение эффективности алгоритма на участках разрывности глубин (ребра зданий и т. п.);

е) темпоральный отбор кадров. Предположительно, этот механизм должен повышать эффективность сопоставления точек на полувидимых участках. Суть его заключается в том, что при формировании интегральной (по всем видам) сравнительной оценки применяется не усреднение локальных оценок по всем парам (опорный + другой вид), а отбор наилучших по локальным оценкам кадров. В основе такого отбора лежит предположение, что наилучшим кадрам соответствует ситуация видимости точки;

ж) реализованы два режима работы алгоритма. В первом режиме алгоритм обрабатывает каждую точку независимо от других, т. е. делается полный проход по лучу. Во втором режиме обработка новой точки делается с учетом вычисленной глубины предыдущей точки, что позволяет резко сократить диапазон поиска глубин на луче для очередной точки.

В результате работы алгоритма получается массив оценок для всех гипотез-вариантов (глубина, нормаль), удовлетворяющих пороговому критерию. В качестве найденного решения берется вариант с наилучшей оценкой.

Модификация алгоритма

Анализ результатов тестирования показал, что ошибочные решения (неправильный выбор глубины) в описанной выше версии алгоритма возникают по нескольким причинам. Во-первых, это известные причины — повторяемость или однородность текстуры, наличие полувидимых точек. Как следствие, в силу локальности текстурной оценки и дискретности варьируемых параметров (шаг по лучу, изменение нормали) возможны ситуации, когда для ошибочной гипотезы-варианта вычисляемая интегральная по всем видам оценка лучше, чем для правильной глубины. Во-вторых, ошибоч-

ные решения могут порождаться применяемым механизмом темпорального отбора. При темпоральном отборе "плохие" локальные оценки для отдельных пар исключаются, и, соответственно, интегральная оценка получается усреднением по всем оставшимся "хорошим" оценкам. Такая логика направлена на обработку полувидимых точек. Но, поскольку алгоритм не различает видимые и полувидимые точки, этот механизм работает и для видимых точек. А в случае видимых точек это приводит к тому, что интегральная оценка для ошибочных решения искусственно завышается и может оказаться лучше оценки для правильного решения. И в результате мы получаем ошибочную глубину для видимой на всех кадрах точки.

Для исключения указанных ситуаций предлагаются два дополнительных алгоритмических решения. Первое решение заключается в реализации режима "жесткого" отбора пар при формировании интегральной оценки, второе — в реализации критерия "связности", учитывающего тот факт, что поверхности объектов кусочно-гладкие.

Режим "жесткого" отбора пар означает, что интегральная оценка при обработке текущего варианта формируется только при условии, что все локальные оценки для пар проходят порог. Если же хотя бы одна локальная оценка не удовлетворяет установленному порогу, то вариант исключается из дальнейшей обработки, т. е. этот режим является альтернативой темпоральному отбору. Он обеспечивает высокую достоверность (точность) получаемой карты глубин, поскольку исключаются точки с "сомнительными" оценками безотносительно к причинам их порождающих. Как следствие, полувидимые точки в этом случае определяться не будут, однако этот минус компенсируется за счет используемой вышеописанной методики обработки видов и возможностей применения комбинированных режимов работы алгоритма (см. ниже).

Критерий связности применяется для повышения достоверности получаемого решения за счет учета непрерывности (гладкости) наблюдаемой поверхности для двух соседних точек на опорном виде. Критерий справедлив, если наблюдаемая пространственная точка не находится на границе разрывности глубин. В этом случае подразумевается сходство оценок двух соседних точек как на опорном виде, так и на других видах. Для реализации этого критерия необходимо решить три задачи. Первая — обеспечить условие гладкости для применения критерия связности, т. е. выбрать соседнюю точку в направлении, где нет разрыва глубин. Вторая задача — выбрать подходящую метрику для оценки и сформулировать критерий для его алгоритмической реализации. Третья задача возникает в связи с тем, что в вышеописанном базовом алгоритме в качестве решения выбирается вариант с наилучшей оценкой, в то время как правильному решению не всегда соответствует лучшая оценка по причинам,

обозначенным выше. Поэтому задача состоит в нахождении правильного варианта из полученного множества возможных вариантов с использованием оценок текстурного сходства и критерия связности.

Для решения первой задачи используются результаты работы методики сдвигаемых окон, а именно — сохраняется информация о направлении выбранного окна при вычислении оценки. Предположительно именно на этом направлении в ближайшей окрестности нет разрыва глубин, поскольку этому окну соответствует наилучшая оценка. Данные о направлении используются для выбора соседней точки. Для вычисления оценки связности в предшествующих работах использовались данные об интенсивности закраски точек на опорном и другом виде, т. е. использовалась та же оценка текстурного сходства, которой оперирует основной алгоритм.

В данной работе предлагается другой способ вычисления оценки связности. Его суть заключается в оценке расстояния на изображении другого вида между двумя положениями соседней точки (рис. 3), вычисляемыми независимыми способами. Первое положение соседней точки получаем непосредственно в результате ее обработки базовым алгоритмом. Второе положение вычисляется с применением гомографии, полученной для анализируемой (первой) точки. Оба положения должны быть близкими, поскольку обе пространственные точки предположительно находятся на гладкой поверхности в близком соседстве друг от друга. Заметим, что такая оценка не зависит от наклона наблюдаемой поверхности. Например, нельзя в качестве оценки связности использовать расстояние между образами основной и соседней точки на другом виде, поскольку при большом наклоне поверхности (маленький угол между лучом зрения и касательной плоскостью в наблюдаемой точке) это расстояние может превышать любой заданный порог. Таким образом, оценку связности точки $p1$ и соседней с ней точки $p2$ можно записать следующим образом:

$$D^j = |p2^j - p2_{G1}^j|,$$

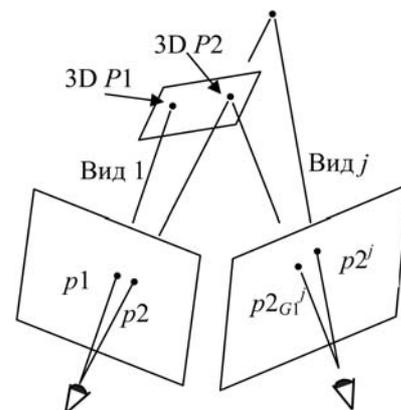


Рис. 3. Оценка связности точек $p1$ и $p2$

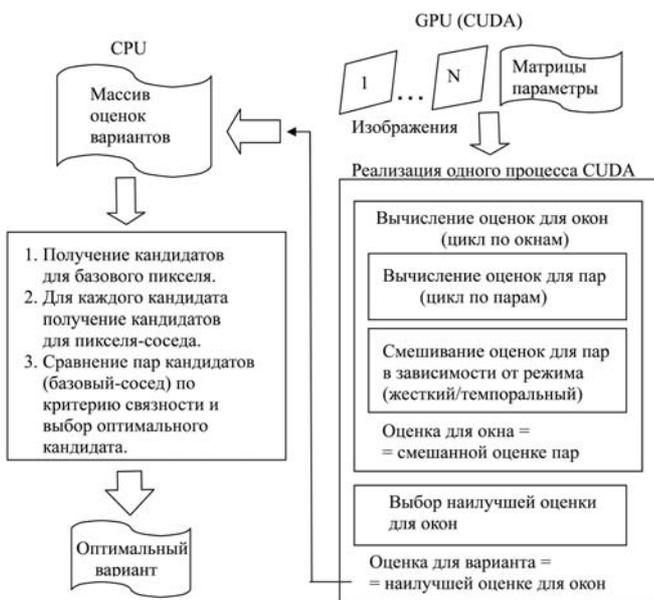


Рис. 4. Схема алгоритма

где $p2^j$ — координаты соседней точки на виде j , полученные в результате независимой обработки базовым алгоритмом соседней точки (задаваемой на опорном изображении); $p2^j_{G1}$ — координаты соседней точки на виде j , вычисленные преобразованием плоской гомографии, полученным для $p1$. Исходной информацией для решения третьей задачи является множество вариантов для точки $p1$ и множество вариантов для точки $p2$, выбранной согласно вышеописанному способу.

Далее по тексту для обозначения варианта решения (глубина, нормаль с оценкой сходства) будем использовать термин "кандидат". Будем искать такую пару кандидатов (кандидат из множества для $p1$, кандидат из множества для $p2$), которая, во-первых, удовлетворяет критерию связности $D < \epsilon$, а, во-вторых, имеет лучшую суммарную оценку сходства. Зафиксируем число рассматриваемых кандидатов q для обрабатываемой точки. Для каждого кандидата точки $p1$ имеется q кандидатов точки $p2$, выбранной в нужном направлении. Для вычисления оценки связности пары кандидатов используется "гомография" рассматриваемого кандидата точки $p1$. Таким образом, обрабатываются q^2 пар кандидатов. Из них исключаются пары, не удовлетворяющие критерию связности. Для каждой оставшейся пары подсчитывается суммарная оценка сходства ($v1 + v2$), где $v1$ — оценка кандидата $p1$, $v2$ — оценка кандидата $p2$. В качестве искомого решения выбирается пара с наилучшей суммарной оценкой. Таким образом, получаем глубины для точки $p1$ и соседней точки $p2$.

Еще одной особенностью применяемой схемы вычисления оценок сходства является использование аппаратной субпиксельной интерполяции, что позволяет оперировать не с пикселями, а с точками.

Схема модифицированной версии алгоритма с реализацией параллельных вычислений на графических процессорах средствами языка CUDA представлена на рис. 4. Параллельные вычисления организованы для обработки одной точки. Результат — множество вариантов решения для точки. Остальные вычисления осуществляются на центральном процессоре.

Настроечные параметры и режимы работы алгоритма. В системе реализован достаточно большой набор возможностей по интерактивной настройке параметров и выбору режимов построения карт глубин. Во-первых, это число видов и тип изображений (черно-белые или цветные). Наряду с основной реализацией режима работы с использованием многопроцессорной обработки на языке CUDA есть возможность обработки только на центральном процессоре (при отсутствии соответствующей графической платы или для получения сравнительных оценок по скорости работы алгоритмов).

Вначале оператором указывается прямоугольная область на опорном изображении, для которой будет получена карта глубин. Выбирается тип окна (прямоугольное или линейное) для вычисления оценки сходства. Также есть возможность использования различных метрик для получения оценок сходства — кроме основной метрики "квадратичные разности интенсивностей" возможны и другие (кросс-корреляция, энтропия). К основным, интерактивно настраиваемым параметрам относятся: шаг по лучу и число нормалей-гипотез при поиске глубины для точки, порог для оценки сходства, размер окна сравнения, режим применения окон сравнения (только центральное окно или сдвигаемые окна), а также режим "жесткого" отбора гипотез-вариантов или режим "темпорального" отбора.

Интерактивно выбирается и режим работы алгоритма: с применением критерия связности или без него. При использовании "связности" задаются два параметра: порог для связности и число кандидатов. Интерактивная настройка перечисленных параметров и режимов работы алгоритма позволяет оптимизировать соотношение времени обработки и точности получаемых карт глубин. Например, при использовании центрального окна время обработки примерно в 6 раз меньше, чем с использованием сдвигаемых окон. Для минимизации вычислительных затрат также была реализована интерактивная возможность задания многоэтапной обработки. То есть после выполнения первого этапа можно переуставлять вышеперечисленные параметры и режимы и продолжать обработку первоначально установленной области на опорном изображении. На этапе продолжения обрабатываются точки, для которых не были получены глубины. Например, на первом этапе можно задать режим "жесткого" отбора с применением центрального окна, а на продолжении применить сдвигаемые окна. В этом случае для обработки подавляющего числа точек будет

затрачено минимальное время, а более трудоемкая обработка со сдвигаемыми окнами будет выполнена только для точек вблизи границ разрывности по глубине.

Эксперименты

Вычислительные эксперименты проводились на PC CPU Intel® Core™ i5 CPU760 с видеокартой NVIDIA GeForce GTX 470 на модельных сценах и на наборе данных (<http://vision.middlebury.edu/mview/>) [7] (рис. 5, см. третью сторону обложки). Поскольку в предыдущей работе [2] было отмечено преимущество линейного окна в сравнении с прямоугольным, во всех экспериментах использовалось линейное окно сравнения. Оценивалось ускорение за счет применения многопроцессорной обработки и сравнительные времена получения карты глубин и достигаемые точности при различных описанных выше режимах работы системы. На сцене 1 расстояние от камеры до объектов составляло 140...260 м (единицы измерения условные, поскольку сцена модельная). Для сцены 2 эти расстояния 700...1100 м.

Оценки точности вычислялись по отношению к используемым полигональным моделям сцен в условных сантиметрах и в процентах по отношению к глубине сцены. Применение вышеуказанной видеокарты для алгоритма с линейным окном дало ускорение по отношению к CPU до 200 раз. В экспериментах с данными сценами оптимальный размер грубого шага 0,1...0,25 % от глубины сцены. При увеличении числа нормалей-гипотез наблюдался небольшой прирост в точности. Эксперименты показали, что применение "жесткого" режима отбора всегда дает большое преимущество в точности по сравнению с темпоральным отбором. Типичный пример, иллюстрирующий этот вывод, показан на рис. 6 (см. третью сторону обложки). В случае б) точность составила 1 м, определилось 1730 точек из 1840 за 1,37 с, в случае в) точность 2,75 см, определилось 1457 точки за 1,31 с. Однако следует учесть, что "жесткий" режим целесообразно использовать только при применении описанной выше методики построения 3D-модели.

Предложенный в работе механизм учета "связности" дает дополнительное преимущество и в точности, и в числе сопоставленных точек, но скорость обработки при этом, естественно, снижается. В соответствии с логикой алгоритма, реализующего учет "связности", при числе кандидатов = 1 он работает как фильтр. В многочисленных экспериментах эффект от учета критерия связности был различным в зависимости от характера сцены и соотношения значений других параметров и режимов. При обработке всей сцены 2 (27652 точек) учет связности дает увеличение точности на 17 %. Эксперименты подтвердили целесообразность применения описанной выше двухэтапной методики работы алгоритма, позволяющей использовать режим сдвигаемых окон только вблизи границ разрывности

Зависимость эффективности алгоритма от числа используемых видов

Число видов	Точность, м (глубина, %)/число найденных точек из 27 652, без учета связности	Точность, м (глубина, %)/число найденных точек из 27 652, с учетом связности
3	1,52 (0,2 %)/21 657	1,12 (0,16 %)/21 064
4	0,82 (0,12)/20 332	0,56 (0,08)/19 773
5	0,53 (0,075)/19 055	0,44 (0,06)/18 502
11	0,29 (0,04)/17 066	0,24 (0,034)/16 520

глубин. Эксперименты также показали, что работа алгоритма в режиме 2 (см п. ж раздела "Базовая версия алгоритма построения карт глубин") дает более высокую скорость обработки, чем в первом режиме при одинаковых значениях других параметров.

Однако сравнимая точность получаемой карты глубин достигается только на сценах с гладкими поверхностями. В среднем скорости обработки отличаются в два раза. На рис. 7 (см. четвертую сторону обложки) представлена построенная по карте глубин текстурированная триангуляционная сетка для сцены 2. Карта глубин получена по пяти видам с параметрами: 700 грубых шагов (1 м) и 40 мелких (0,025 м); угол между нормальями-гипотезами 45°; линейное окно сравнения 13 пикселей; жесткий режим, центральное окно, учет связности. Время обсчета 10,4 с, точность по 17 841 точек из 27 652 — 51 см или в процентах по отношению к глубине сцены — 0,06 %. Наблюдаемые на сетке "дыры" являются результатом постфильтрации, направленной на повышение достоверности определяемых глубин. Аналогичный результат для сцены 3 показан на рис. 8 (см. четвертую сторону обложки). Зависимость точности получаемой карты глубин от числа используемых видов для сцены 2 представлена в таблице.

Заключение

Представлена усовершенствованная версия ранее разработанного авторами алгоритма вычисления карт глубин с реализацией параллельных вычислений на графических процессорах средствами языка CUDA. В данной версии реализованы дополнительные пороговые оценки по учету связности точек видимой поверхности и фильтрации ложных решений, направленные на повышение достоверности определяемой карты глубин. Интерактивные возможности по настройке параметров и переключению режимов работы алгоритма с возможностью многоэтапной обработки повысили гибкость использования системы, позволяя регулировать соотношение точность вычисления глубин/время счета. Поскольку в текущей версии алгоритма параллелизм реализован на уровне обработки одной точки, в дальнейшем предполагается его модификация с параллельной обработкой всего массива точек, что возможно позволит обеспечить режим реального времени.

Список литературы

1. Bobkov V. A., Ronshin Y. I. GPU Implementation of Depth Map Algorithm // The First Russia and Pacific Conference on Computer Technology and Applications, Vladivostok, 6–9 September. 2010. P. 382–387.
2. Бобков В. А., Роншин Ю. И. Реализация алгоритма построения карт глубин на графических процессорах // Информационные технологии. 2011. № 8. С. 17–22.
3. Campbell N. D. F., Vogiatzis G., Hernández C., Cipolla R. Using multiple hypotheses to improve depthmaps for multi-view stereo // Proc. 10th ECCV. Part 1. 2008. P. 766–779.
4. Collins R. T. A space-sweep approach to true multi-image matching // Int. Conf. on Computer Vision and Pattern Recognition. 1996. P. 358–363.

5. Mordohai P., Frahm J.-M., Akbarzadeh A., Clipp B., Engels C., Gallup D., Merrell P., Salmi C., Sinha S., Talton B., Wang L., Yang Q., Stewenius H., Towles H., Welch G., Yang R., Pollefeys M., Nister D. Real-time video-based reconstruction of urban environments // ISPRS Working Group V/4 Workshop 3D-ARCH 2007: 3D Virtual Reconstruction and Visualization of Complex Architectures (ETH Zurich, Switzerland), July 2007.
6. Kang S., Szeliski R. and Chai J. Handling occlusion in dense multi-view stereo // Int. Conf. on Computer Vision and Pattern Recognition. 2001. P. 103–110.
7. Seitz S., Curless B., Diebel J., Scharstein D., Szeliski R. A comparison and evaluation of multi-view stereo reconstruction algorithms // Proc. of the IEEE international conference on computer vision and pattern recognition. 2006. P. 519–526.

УДК 004.925.8

В. А. Бимаков, директор,
Е. В. Бимаков, вед. инженер,
Общество с ограниченной ответственностью
"Робинт", г. Ижевск,
e-mail: robint@mail.ru

Принципы построения и функционирования устройств для высокоскоростной обработки n -мерных пиксельных геометрических моделей

Излагаются общие принципы построения и функционирования устройств, предназначенных для применения в интеллектуальных системах технического зрения, в системах управления высокоманевренными автономными роботами, в системах высокореалистичной компьютерной графики и анимации, в системах автоматизированного проектирования. Метод обработки сцен и изображений, реализуемый в данных устройствах, позволяет предложить новые сверхскоростные алгоритмы обработки, не зависящие от геометрической формы обрабатываемых объектов.

Ключевые слова: скоростная обработка сцен и изображений, геометрические модели

1. Отличительные признаки устройств параллельной обработки пиксельных моделей

Пиксельный (иначе точечный или дискретный) метод геометрического моделирования имеет следующие исключительные преимущества по глубокому распараллеливанию основных процедур обработки сцен (ОПОС) [1]:

- теоретико-множественные операции (ТМО) над объектами сцен;

- геометрические преобразования (ГП);
- анализ объектов на пересекательность;
- вычисление объемов и площадей объектов;
- определение положения объекта в пространстве.

Реализовать эти преимущества призваны специальные вычислительные устройства — устройства параллельной обработки пиксельных моделей (УОПМ) [1, 3, 4, 5]. УОПМ имеют следующие отличительные признаки:

1) все v -мерные геометрические модели, обрабатываемые в УОПМ с максимальной скоростью, являются пиксельными и представляются в виде одноцветного содержимого n -мерных бинарных изображений (где $v \geq n \geq 2$ — целые числа);

2) адресуемыми информационными единицами УОПМ являются не только числовые двоичные переменные (иначе "одномерные бинарные изображения"), но также и n -мерные бинарные изображения (БИ) некоторого заданного формата $F^{(n)}$ [1];

3) в состав УОПМ помимо обычного арифметико-логического узла, предназначенного для обработки чисел, входят дополнительные аппаратные средства, позволяющие выполнять ОПОС над одноцветным содержимым адресуемых n -мерных БИ формата $F^{(n)}$ за время, сравнимое с одним элементарным тактом устройства;

4) система команд УОПМ помимо арифметико-логических команд обработки чисел содержит дополнительный набор команд, оперирующих $F^{(n)}$ -форматными БИ, каждая из таких команд реализует одну ОПОС над одноцветным содержимым адресуемых БИ, этот набор является функционально полным, позволяющим выполнить весь цикл высокоскоростной обработки сцены на основе использования только n -мерных пиксельных моделей объектов.

К наиболее близким аналогам УОПМ можно отнести:

- матричные препроцессоры, используемые для предварительной обработки (фильтрации) оцифрованных фотоизображений сцены;

- видеопроцессоры, предназначенные для визуализации (рендеринга) синтезированных полигональных сцен на экране растрового дисплея.

Данные аналоги позволяют максимально распараллелить ТМО и некоторые ГП (одношаговые сдвиги, ортогональные повороты, различные виды симметрий и т. п.) над пиксельными геометрическими моделями, представленными в виде одноцветного содержимого БИ. Однако набор параллельных ОПОС (осуществляемых над пиксельными моделями), аппаратно-реализованный в данных аналогах, крайне мал и не является функционально полным. В результате указанные аналоги можно применять только на начальном (фильтрация) и конечном (визуализация) этапах высокоскоростной обработки сцены, а для выполнения ОПОС на других (наиболее сложных и трудоемких) этапах обработки (этапах анализа и синтеза сцен) вынуждены применять другие устройства (однопроцессорные и конвейерные системы), оперирующие уже моделями-многогранниками. При этом время обработки сложных геометрических сцен резко возрастает.

Напротив, набор параллельных ОПОС, аппаратно реализованный в УОПМ, является функционально полным, пиксельные модели используются здесь на всех этапах обработки сцены, необходимости в применении моделей-многогранников нет. Время обработки резко сокращается, появляется возможность применения более коротких алгоритмов обработки, время выполнения которых не зависит от геометрической формы обрабатываемых объектов.

2. Принципы построения УОПМ

УОПМ представляет собой многопроцессорную ОКМД-систему (рис. 1), в которой функцию управления выполняет блок программного управления (БПУ). Назначение БПУ (выполняемого по одной из известных схем) — хранение команд и числовых данных, организация их последовательного извлечения из памяти программ и данных (ППД), арифметико-логическая обработка чисел, хранящихся в ППД.

Но главным компонентом УОПМ, обеспечивающим рекордно высокую производительность данного вида техники при обработке сцен и изображений, должен стать блок обработки бинарных изображений (БОБИ).

2.1. Назначение, состав, архитектура и конструкция БОБИ

БОБИ служит для реализации ОПОС над n -мерными пиксельными объектами, образованными одноцветным содержимым $F^{(n)}$ -форматных БИ; ОПОС выполняются

с предельной степенью параллелизации (т. е. за время, сравнимое с одним элементарным тактом управления). В состав БОБИ входят:

- 1) сверхоперативная память бинарных изображений (СПБИ) произвольной выборки для хранения адресуемых $F^{(n)}$ -форматных БИ-операндов;
- 2) N_F — разрядные двоичные регистры (РГБИ) для хранения обрабатываемых $F^{(n)}$ -форматных БИ-операндов (в одном РГБИ размещается один БИ-операнд) на время выполнения текущей ОПОС (где N_F — число пикселей в $F^{(n)}$);
- 3) геометрические преобразователи (ГП) одноцветного содержимого $F^{(n)}$ -форматного БИ-операнда (ГПБИ);
- 4) теоретико-множественные преобразователи одноцветного содержимого $F^{(n)}$ -форматных БИ-операндов (ТМПБИ);
- 5) вычислитель значения меры $\mu^{(n)}$ (ВМБИ) одноцветного содержимого $F^{(n)}$ -форматного БИ-операнда (понятие меры $\mu^{(n)}$ является n -мерным обобщением понятий длины ($\mu^{(1)}$), площади ($\mu^{(2)}$), объема ($\mu^{(3)}$) [2]);

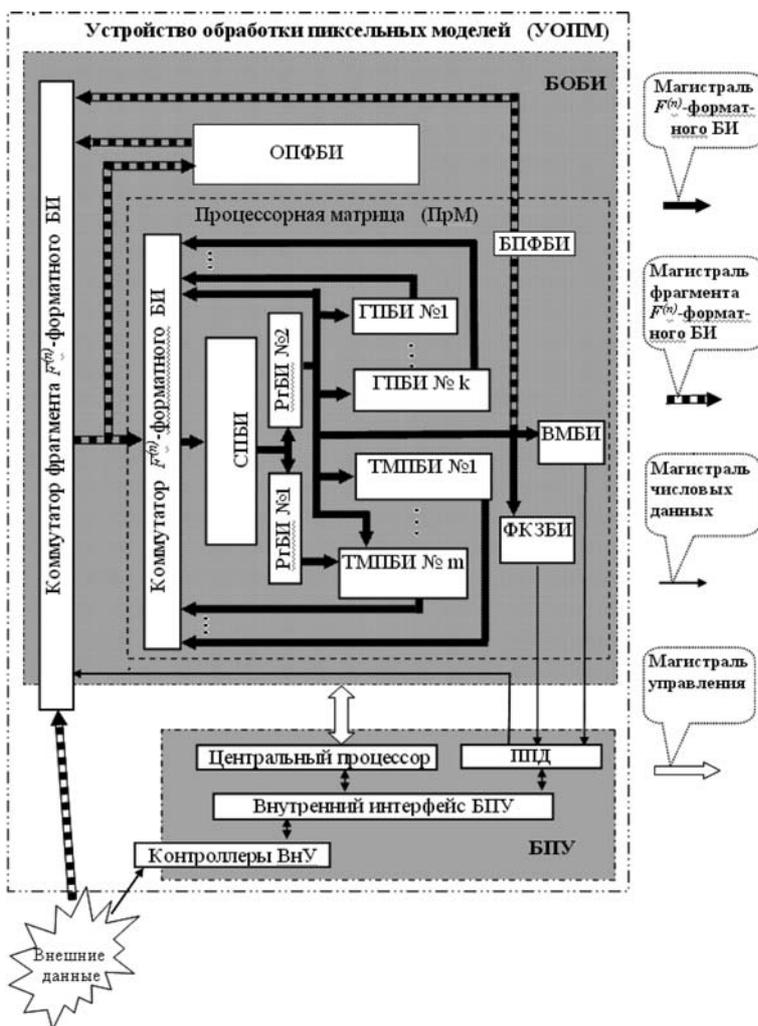


Рис. 1. Общая структура УОПМ

б) формирователь кода заполнения БИ, интегрально описывающего характер заполнения $F^{(n)}$ -форматного БИ одноцветным содержимым (ФКЗБИ);

7) оперативная память фрагментов (ОПФБИ) для оперативного хранения адресуемых фрагментов $F^{(n)}$ -форматных БИ (в отличие от СПБИ память ОПФБИ имеет большую емкость и является наращиваемой);

8) буферная память фрагментов (БПФБИ) для временного (сверхоперативного) хранения фрагментов $F^{(n)}$ -форматных БИ.

Перечисленные первый, второй и третий компоненты БОБИ вместе могут быть представлены в виде единой процессорной матрицы (ПрМ), состоящей из N_F однотипных процессоров (ОПр), связанных сетью межпроцессорных связей и имеющих собственную локальную память (ЛП) [1]. Одному пикселю $p_i^{(n)} \subset F^{(n)}$ в ПрМ соответствует один ОПР. ЛП всех ОПР матрицы ПрМ вместе образуют СПБИ.

Для реализации функционально-полного набора ТМО над БИ-операндами в составе БОБИ должно быть не менее двух РгБИ. Информационный вход каждого j -го разряда РгБИ соединяется с j -м выходом регистра данных СПБИ (т. е. с информационным выходом j -й ЛП). Перед выполнением каждой ОПОС участвующий в ней $F^{(n)}$ -форматный БИ-операнд читается из СПБИ в нужный РгБИ одновременно по всем пикселям этого БИ-операнда.

Для увеличения степени распараллеливания ОПОС целесообразно увеличивать N_F . С этой целью ОПР выполняются одноразрядными. Однако неограниченный рост N_F приведет к неограниченному росту аппаратуры и к снижению тактовой частоты управления. Другая проблема связана с тем, что, несмотря на конечность спектра дискретизирующих отображений (ДО) [1], моделирующих ГП, общее число ГП (даже в рамках одного вида моделируемых ГП) весьма велико, и все они не могут быть реализованы аппаратно. В этой связи возникает проблема выбора оптимального $F^{(n)}$, а также задача выбора оптимального базиса аппаратно-реализуемых ДО (базиса АРДО). Обе проблемы тесно взаимосвязаны [1].

ДО моделирует ГП, в общем случае, с погрешностью, а последовательное выполнение нескольких ДО приведет к возрастанию погрешности, вносимой отдельным ДО. Для оценки точности моделирования геометрического преобразования $f: R^n \rightarrow R^n$ (где R^n — n -мерное евклидово пространство) дискретизирующим отображением f^p введем следующие определения.

Определение 1. Погрешностью $\sigma(p_i^{(n)})$ моделирования пикселя-образа $f(p_i^{(n)})$ пикселем-моделью $f^p(p_i^{(n)})$ при моделировании преобразования $f: R^n \rightarrow R^n$ дискретизирующим отображением f^p назовем расстояние между центром образа $f(p_i^{(n)})$ пикселя-прообраза $p_i^{(n)}$ и центром пикселя-модели $f^p(p_i^{(n)})$.

Для пикселей $p_i^{(n)}$, рассматриваемых ниже (кубиков $1 \times 1 \times 1$, квадратиков 1×1), и аффинных преобразований f (переносов, поворотов, сжатий) дополнительное определение центра $f^p(p_i^{(n)})$ и центра $f(p_i^{(n)})$ не требуется.

Определение 2. Погрешностью σ моделирования преобразования f дискретизирующим отображением f^p над одноцветным содержимым $F^{(n)}$ -форматного БИ назовем максимальное значение погрешности $\sigma(p_i^{(n)})$ для всех одноцветных пикселей $p_i^{(n)} \subset F^{(n)}$.

Погрешность моделирования f дискретизирующим отображением f^p обусловлена двумя причинами: погрешностью дискретизации и выходом содержимого преобразованного изображения за пределы формата $F^{(n)}$. Учитывая это, главными претендентами на аппаратную реализацию должны стать собственные $F^{(n)}$ -форматные ГП (моделируемые без какой-либо погрешности) и дискретно-инвариантные ГП (моделируемые без погрешности дискретизации). Примеры этих преобразований приведены в работе [1].

Основу оборудования ГПБИ составляют многочисленные проводные связи. Высокая насыщенность ГПБИ оборудованием остро ставит вопрос о выборе оптимального набора АРДО, каждое из которых реализуется в отдельном ГПБИ и моделирует ГП над одноцветным содержимым БИ, размещенным в РгБИ. Пути оптимизации требуемого набора ГПБИ рассмотрим на примере набора ГПБИ, воспроизводящих повороты одноцветного содержимого БИ кубического формата $A \times A \times A$ вокруг центра этого БИ.

В R^3 угловое положение объекта в пространстве может быть задано тремя углами Эйлера: углом φ собственного вращения; углом θ нутации; углом ψ прецессии. Повороты одноцветного содержимого БИ могут отличаться не только значениями углов φ , θ , ψ , но и центром поворота. Для переноса повернутых БИ минимально необходимый набор АРДО должен содержать хотя бы одно ДО, воспроизводящее одношаговый перенос изображения $A \times A \times A$ в одном из шести направлений (**Ox**, **Oy**, **Oz**, **-Ox**, **-Oy**, **-Oz**) [1], параллельных ребрам куба $A \times A \times A$ бинарного изображения.

Таким образом, в минимальный набор АРДО должны войти:

- 1) АРДО, моделирующее поворот БИ формата $A \times A \times A$ на угол 90° вокруг центральной оси **Ox**;
- 2) АРДО, моделирующее поворот БИ формата $A \times A \times A$ на угол 90° вокруг центральной оси **Oy** (**Oz**);
- 3) АРДО, моделирующее параллельный перенос изображения $A \times A \times A$ на единицу в направлении вектора **Ox** (**-Ox**, **Oy**, **-Oy**, **Oz**, **-Oz**).

Все другие дискретно-инвариантные ГП, осуществляющие поворот и перенос одноцветного содержимого БИ формата $A \times A \times A$, могут быть

воспроизведены с помощью композиции (суперпозиции) этих трех АРДО.

Но не все повороты, которые необходимо моделировать в УОПМ с высокой степенью параллелизации, являются дискретно-инвариантными. Набор АРДО должен быть расширен ДО, воспроизводящими дискретно-неинвариантные повороты. Моделирование дискретно-неинвариантных ГП может быть выполнено с различной точностью. Из определения σ и правила № 2 [1] выбора соответствия между пикселем-моделью $f^p(p_i^{(3)})$ и его пикселем-прообразом $p_i^{(3)}$ следует, что погрешность σ моделирования поворота содержимого БИ формата $A \times A \times A$ вокруг центральной оси (направленной параллельно ребру куба $A \times A \times A$) одним дискретно-неинвариантным ДО не превышает значения $0,5\sqrt{2} \approx 0,7$ [4]. Моделирование поворота того же БИ путем последовательного выполнения двух дискретно-неинвариантных ДО: ДО, моделирующего поворот на угол θ ; ДО, моделирующего поворот на угол ψ , — приведет к погрешности $\sigma \leq \sqrt{(0,5)^2 + (0,5\sqrt{2} + 0,5)^2} \approx 1,3$ [4].

В БОБИ изменение угловой ориентации содержимого БИ $A \times A \times A$ вокруг центра этого изображения сразу на три произвольных угла Эйлера (φ, θ, ψ) можно воспроизвести тремя различными способами:

а) моделированием поворота одним дискретно-неинвариантным ДО;

б) моделированием двумя дискретно-неинвариантными ДО, выполняемыми в следующей последовательности:

б-1) выполняется ДО, воспроизводящее поворот на угол φ ;

б-2) выполняется ДО, воспроизводящее пространственный поворот, эквивалентный повороту сначала на угол θ , а затем на угол ψ ;

в) моделированием тремя дискретно-неинвариантными ДО, выполняемыми в следующей последовательности:

в-1) ДО, воспроизводящим поворот на угол φ ;

в-2) ДО, воспроизводящим поворот на угол θ ;

в-3) ДО, воспроизводящим поворот на угол ψ .

Способ "а" детально описан в работе [1]. Этот способ отличается максимальным быстродействием, минимальной погрешностью $\sigma \leq 0,5\sqrt{3} \approx 0,87$ [4], но и чрезвычайно высокими аппаратными затратами. Способ "б" характеризуется умеренной погрешностью $\sigma \leq \sqrt{0,5 + (0,5\sqrt{2} + 0,5)^2} \approx 1,39$ [4], умеренны-

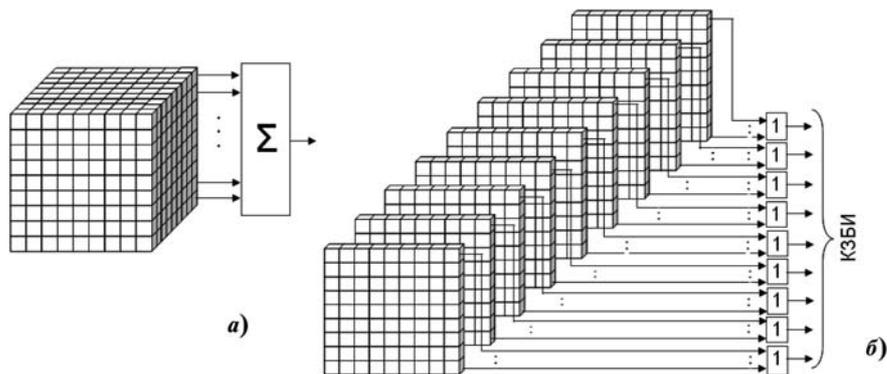


Рис. 2. Формирование интегральных характеристик $F^{(3)}$ -форматного БИ-операнда $9 \times 9 \times 9$ в БОБИ

ми аппаратными затратами и достаточно высоким быстродействием. Способ "в" отличается большей погрешностью $\sigma \leq \sqrt{1^2 + (0,75\sqrt{2} + 0,5)^2 + (\sqrt{2}/4)^2} \approx 1,89$, относительно большие временные затраты, но этот способ может быть реализован минимальным объемом оборудования.

Способы моделирования поворотов двумя и тремя дискретно-неинвариантными ДО являются наиболее приемлемыми для реализации в БОБИ большинства пространственных поворотов (ввиду относительно невысоких требований к объему используемого оборудования).

Преобразования сжатия одноцветного содержимого $F^{(n)}$ -форматного БИ-операнда могут быть реализованы в БОБИ либо аппаратно, либо программно. Аппаратный способ отличается максимальным быстродействием, но ввиду больших аппаратных затрат его использование оправдано только в УОПМ, проектируемых под конкретные алгоритмы и под заранее известные коэффициенты сжатия. Программный способ моделирования сжатия не требует дополнительных аппаратных затрат, при его реализации используются средства ТМПБИ и описанный выше минимальный набор дискретно-инвариантных АРДО [3]. Благодаря предельному распараллеливанию в УОПМ одношагового переноса и ТМО над содержимым БИ $A \times A \times A$ программный способ имеет достаточно высокое быстродействие и его использование для большинства коэффициентов сжатия является предпочтительным.

Интегральными характеристиками бинарных изображений $A \times A \times A$ (или $A\sqrt{A} \times A\sqrt{A}$), вычисляемыми в УОПМ аппаратно, являются:

- объем (площадь) одноцветного содержимого БИ-операнда;
- код заполнения БИ-операнда (КЗБИ).

Функцию ВМБИ выполняет A^3 -входовой параллельный сумматор (рис. 2, а), каждый i -й вход которого соединен с i -м выходом одного из РГБИ. Сумматор выполняется по одной из известных схем.

Вариант ФКЗБИ показан на рис. 2, б, состоит из A независимых A^2 -входных схем ИЛИ, каждый i -й вход j -й схемы ИЛИ соединен с i -м выходом j -й группы выходов РгБИ, соответствующих в РгБИ j -му слою $A \times A \times 1$.

2.2. Особенности конструкции УОПМ

Исключительно высокая аппаратная насыщенность БОБИ оказывает определяющее влияние на выбор конструкции УОПМ. Для обеспечения высокой тактовой частоты, уменьшения габаритных размеров, повышения надежности УОПМ проектирование УОПМ должно начинаться с проектирования элементной базы БОБИ, учитывающей все особенности конкретной архитектуры БОБИ. В качестве элементной базы БОБИ целесообразно использовать специально конфигурируемые ПЛИС (при небольших сериях производства УОПМ) или заказные СБИС (при больших сериях производства УОПМ).

ПрМ в БОБИ может быть выполнена в однокристалльном и многокристалльном исполнении. Однокристалльный вариант позволяет резко поднять надежность и тактовую частоту УОПМ, однако разрешение $F^{(n)}$ здесь невысоко. Многокристалльный вариант позволяет значительно поднять разрешение $F^{(n)}$. При разработке многокристалльных вариантов ПрМ следует в качестве несущей конструкции БОБИ применять радиоэлектронные блоки специальной разъемной конструкции [6], которые существенно снижают общую длину проводных связей и общее число разъемных контактов при сохранении высокой ремонтопригодности устройства.

3. Принципы функционирования УОПМ

3.1. Понятие составного изображения

Максимальное распараллеливание ОПОС достигается в УОПМ только при обработке $F^{(n)}$ -форматных



Рис. 3. Вариант представления полутоновых изображений в УОПМ

матных БИ. Разрешающая способность (N_F) формата $F^{(n)}$ ограничена возможностями элементной базы (числом и размером кристаллов, плотностью размещения электронных компонентов на кристалле) и не может быть очень большой (в настоящее время примерно от 500 до 20 000 пикселей). В то же время разрешение обрабатываемых изображений, встречающихся в большинстве прикладных задач, значительно превышает разрешение $F^{(n)}$ -форматных БИ (1 млн и более). Поэтому изображения высокого разрешения представляются в УОПМ в виде изображений, составленных из нескольких $F^{(n)}$ -форматных БИ.

Определение 3. Бинарное изображение, формат которого представляется в виде объединения нескольких взаимно непересекающихся БИ формата $F^{(n)}$, назовем *бинарным $F^{(n)}$ -составным изображением* или просто бинарным составным изображением (БСИ).

БСИ обрабатываются в УОПМ последовательно (БИ-фрагмент за БИ-фрагментом). Полутоновые и цветные изображения представляются в УОПМ тоже в виде нескольких $F^{(n)}$ -форматных БИ и обрабатываются последовательно ($F^{(n)}$ -форматными фрагментами). Вариант представления $F^{(2)}$ -форматного полутонового изображения в виде нескольких $F^{(2)}$ -форматных БСИ показан на рис. 3.

Очевидно, что чем меньше число $F^{(n)}$ -форматных фрагментов в БСИ, тем выше скорость обработки данного БСИ. Поскольку это число зависит от размерности БСИ и от размерности фрагмента $F^{(n)}$, то для увеличения скорости обработки БСИ целесообразно повышать разрешение $F^{(n)}$.

УОПМ помимо высокоскоростного решения геометрических задач можно использовать для параллельной обработки числовых массивов (значений яркости пикселей, значений сенсорных сигналов, значений сигналов регулирования сервоприводами и т. п.). Это делает УОПМ почти идеальным инструментом для применения его в качестве вычислителя в системах управления автономным роботом [5].

3.3. Обработка бинарных составных изображений в УОПМ

БСИ, обрабатываемые в УОПМ, могут отличаться числом $F^{(n)}$ -форматных фрагментов и положением $F^{(n)}$ -форматных фрагментов относительно друг друга. Кроме того, одно и то же БСИ на разных этапах обработки может по-разному отображаться внутри УОПМ, т. е. иметь различную форму внутреннего машинного представления. Выбор формы представления БСИ внутри УОПМ определяется способом хранения информации. Поскольку в УОПМ адресуемые единицы накопителей СПБИ, ППД, ОПФБИ и внешней памяти различны, то и

форма машинного представления БСИ внутри указанных накопителей также будет различной.

Для ускорения обработки БСИ целесообразно в алгоритмах решения прикладных задач пользоваться только типовыми (стандартными) формами представления БСИ. Подобная типизация видов представления БСИ в УОПМ несколько напоминает типизацию числовых данных в обычных компьютерах, когда на основе адресуемых информационных единиц (байтов, слов — аналогов адресуемого $F^{(n)}$ -форматного БИ) создается множество других, более сложных типов числовых данных (длинных целых, вещественных и т. п.). Аналогично тому, как на множестве типовых числовых данных определяются типовые арифметико-логические операции, так и на множестве типовых форм представления БСИ могут быть определены типовые процедуры обработки сцены (ТПОС), для которых известны (вычислены еще на этапе проектирования УОПМ) результаты промежуточных расчетов, необходимых для сборки преобразованных $F^{(n)}$ -форматных БИ в синтезируемое БСИ. Благодаря исключению этих промежуточных расчетов из ТПОС скорость их выполнения будет превышать скорость обработки нетиповых процедур обработки сцены.

Все типовые формы представления БСИ в УОПМ можно отнести к *пиксельным и непиксельным формам представления БСИ*. Пиксельные формы представления используются на этапах сверхоперативной и оперативной обработки БСИ. Непиксельные формы представления служат для длительного хранения БСИ в памяти ППД или внешних носителях.

Типовые пиксельные формы представления БСИ. Типовой способ пиксельного представления БСИ предполагает размещение в накопителях УОПМ следующих информационных компонентов:

- 1) *тела* БСИ, хранящегося в памяти БОБИ (СПБИ, ОПФБИ) и состоящего из адресуемых БИ-фрагментов соответствующего формата;
- 2) *паспорта* БСИ, хранящегося в ППД и состоящего из нескольких служебных слов (байт);
- 3) *адреса* паспорта БСИ, хранящегося в ППД и состоящего из слова.

Паспорт БСИ содержит слова, обозначающие: *код типовой формы* представления БСИ; *начальный адрес тела* БСИ в памяти БОБИ. Адрес первого слова паспорта БСИ является *адресом паспорта* БСИ в ППД.

Непиксельные формы представления БСИ. Непиксельная форма представления БСИ является экономным способом хранения БСИ в ППД или на внешних носителях. Помимо небольшого объема используемой памяти эта форма должна позволять быстро восстанавливать по ней типовую (сверхоперативную или оперативную) пиксельную форму представления данного БСИ (соответственно в СПБИ или в ОПФБИ).

Так как наивысшая скорость формирования пиксельных моделей достигается только в БОБИ, любое непиксельное описание БСИ — это программа УОПМ, содержащая в себе и команды БПУ, и команды БОБИ. Выполнение этой программы приводит к синтезу типовой пиксельной формы данного БСИ в одном из накопителей БОБИ (либо в СПБИ, либо в ОПФБИ). Типовой способ непиксельного представления БСИ состоит из следующих информационных компонентов:

- 1) *программы синтеза* пиксельной формы представления БСИ;
- 2) *паспорта* непиксельной формы представления БСИ;
- 3) *адреса паспорта* непиксельной формы представления БСИ.

Паспорт непиксельной формы представления БСИ содержит:

- 1) код формы представления БСИ;
- 2) адрес точки входа программы для синтеза пиксельной формы БСИ.

Адрес первого слова паспорта непиксельной формы БСИ является ППД-адресом паспорта непиксельной формы представления этого БСИ.

Сверхоперативная и оперативная формы представления БСИ. В УОПМ наибольшую разрядность адресуемой единицы и наибольшую скорость выборки имеет СПБИ. По каждому адресу СПБИ хранится один $F^{(n)}$ -форматный БИ-фрагмент составного изображения. Порядок следования адресов $F^{(n)}$ -форматных БИ-фрагментов, образующих тело БСИ, в адресном пространстве СПБИ должен жестко соответствовать пространственному положению этих $F^{(n)}$ -форматных БИ-фрагментов внутри БСИ.

В отличие от СПБИ адресуемой единицей ОПФБИ является бинарный фрагмент формата $P^{(n)} \subset F^{(n)}$ (или иначе субфрагмент БСИ). Порядок следования адресов $P^{(n)}$ -форматных БИ-субфрагментов, образующих тело БСИ, в адресном пространстве ОПФБИ должен жестко соответствовать пространственному положению этих $P^{(n)}$ -форматных БИ-субфрагментов в БСИ.

Список литературы

1. **Бимаков В. А.** Высокоскоростная обработка n -мерных пиксельных геометрических моделей // Информационные технологии. 2009. № 3. С. 29—36.
2. **Колмогоров А. Н., Фомин С. В.** Элементы теории функций и функционального анализа. М.: Наука, 1981.
3. **Бимаков В. А., Бимаков Е. В.** Устройство обработки двумерных и трехмерных изображений: пат. РФ 2289161, опубл. Бюл. № 34 от 10.12.2006.
4. **Бимаков В. А., Бимаков Е. В.** Устройство обработки двумерных и трехмерных изображений: пат. РФ 2376637, опубл. Бюл. № 35 от 20.12.2009.
5. **Бимаков В. А., Бимаков Е. В.** Устройство управления автономным роботом: пат. РФ 2424105, опубл. Бюл. № 20 от 20.07.2011.
6. **Бимаков В. А., Бимаков А. В.** Радиоэлектронный блок: пат. РФ 2192716, опубл. Бюл. № 31 от 10.11.2002.

УДК 548.4+539.37

И. Г. Панин, канд. техн. наук, доц. каф. ИТ,
e-mail: igpanin@list.ru,

В. В. Благовещенский, д-р физ.-мат. наук,
проф. каф. физики,
Костромской государственной
технологической университет

Численное решение уравнения эволюции дислокационного источника Франка — Рида

Дано описание принципов построения модели дислокационного источника Франка — Рида, основанной на решении линейного уравнения в частных производных параболического типа. Проведен выбор наиболее подходящей схемы численного решения уравнения. Показаны устойчивость, сходимость, точность разностной схемы и метода прогонки решения системы линейных уравнений, составленной на основании решаемого уравнения. Выбран алгоритм решения задачи с минимальными временными затратами.

Ключевые слова: кристалл, дислокация, дислокационный источник, модель, численное решение, сходимость, устойчивость, точность

1. Принцип работы и уравнение эволюции источника

Дислокации, находящиеся и движущиеся в кристаллических материалах, оказывают значительное влияние на их механические свойства. При нагружении материала дислокации могут не только двигаться, но и размножаться, тем самым увеличивая свое влияние. Одним из известных методов размножения дислокаций является дислокационный источник Франка — Рида (ФР), принцип работы которого хорошо известен [1—3]. Начальная форма дислокационного сегмента (ДС) представляет из себя прямолинейный отрезок длиной l между точками закрепления A и B (рис. 1). При приложении внешнего напряжения σ ДС начинает прогибаться и через некоторое время принимает критическую форму, близкую к дуге полуокружности с радиусом $R = l/\pi$, где l — длина сегмента. Минимальное напряжение $\sigma_{кр}$, при котором ДС преодолевает критическое положение, называется критическим, и может быть вычислено из условия баланса сил, действующих на ДС: $\sigma_{кр} = Gb/l$, где G —

модуль сдвига; b — модуль вектора Бюргерса. Если внешнее напряжение $\sigma < \sigma_{кр}$, ДС не доходит до критического положения и принимает равновесную конфигурацию с радиусом $R = Gb/(2\sigma)$. В том случае, если $\sigma > \sigma_{кр}$, ДС проходит критическое положение и стремительно расширяется до момента самопересечения ("отшнуровывания"), при котором происходит аннигиляция ДС, при этом сегмент делится на две части. Внешняя часть образует замкнутую петлю, а внутренняя часть принимает исходное положение. Этот процесс может повторяться неоднократно, источник испускает все новые дислокационные петли, что приводит к увеличению плотности дислокационных линий (ДЛ) в кристалле.

Впервые динамическое (с учетом дислокационной вязкости) уравнение эволюции источника ФР было предложено в работах В. Д. Нацика и К. А. Чишко [4] при изучении характеристик его акустического излучения. При выводе уравнения были сделаны следующие упрощения.

1. В плоскости скольжения отсутствуют точечные центры, препятствующие движению ДЛ.
2. Сила трения имеет вязкий характер: $F_{тр} = -Bv$, где B — коэффициент динамического торможения ДЛ; v — скорость единицы длины ДЛ.
3. Сила инерции пренебрежима мала по сравнению с силой трения.
4. При записи сил самодействия краевые и винтовые участки ДЛ принимаются неразличимыми, среда считается неограниченной, однородной и изотропной.
5. Концы ДС шарнирно закреплены полубесконечными дислокациями, перпендикулярными плоскости скольжения, — "возжами", которые воздей-

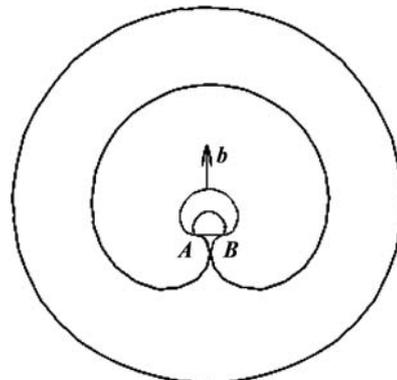


Рис. 1. Источник Франка — Рида

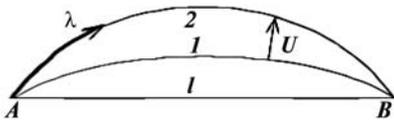


Рис. 2. Последовательные конфигурации дислокационного сегмента

ствуют на элемент дислокации с силой F_b , причем этой силой, ввиду ее малости, можно пренебречь.

6. Направление смещения в произвольной точке дислокации перпендикулярно касательной в этой точке.

7. Максимальное смещение точек ДС из положения 1 (рис. 2) в положение 2 за промежуток времени δt мало по сравнению с длиной сегмента L_0 .

В итоге было получено следующее уравнение для малых смещений ДС [4–6]:

$$\frac{B}{b} \frac{\partial U}{\partial t} - Gb \frac{\partial^2 U}{\partial \lambda^2} = \sigma - \frac{Gb}{R(\lambda, t)}, \quad (1)$$

$$U(0, t) = U(L, t) = 0 \quad \forall t,$$

где t — время; G — модуль сдвига; b — модуль вектора Бюргерса; B — коэффициент динамической вязкости; $R(\lambda, t)$ — радиус кривизны ДС; σ — значение внешнего напряжения; U — искомое смещение произвольной точки; λ — ее криволинейная координата вдоль ДС; L — длина сегмента в криволинейных координатах (рис. 2). Силой взаимодействия с другими дислокациями и силой действия со стороны "возжей" пренебрегаем ввиду их малости. Уравнение (1) является уравнением в частных производных второго порядка параболического типа при заданных граничных условиях закрепления крайних точек ДС

$$U(0, t) = U(L, t) = 0 \quad (2)$$

и начальных условиях

$$U(\lambda, 0) = 0. \quad (3)$$

Уравнение (1)–(3) допускает решение в виде ряда Фурье [6], при этом Фурье-образ сил, стоящих в правой части уравнения выражается через интегралы, которые могут быть вычислены только численно, поскольку численно задается функция $R(\lambda, \tau)$. В работе [7] такое решение использовали для построения математической модели эволюции сегмента, однако при реализации этой модели обнаруживается явление Гиббса [8, 9], которое даже при использовании сглаживающих множителей не позволило выполнить расчеты для всей области значений физических постоянных.

2. Построение численных уравнений

В данной работе предлагается численное решение уравнения (1)–(3) с помощью конечных разностей, при котором на каждом этапе осуществляется переход из положения 1 кривой AB в соседнее положение 2 (рис. 2). Решение строится в криво-

линейной системе координат, причем одна ось λ совпадает с кривой AB , а вторая ось — перпендикулярна к ней, точка A — начало координат. Для построения сеточной области по координате λ первоначально прямолинейный отрезок AB разбивается N точками на равные части. При движении сегмента координаты узлов по λ меняются, могут меняться и расстояния между узлами. Таким образом, задача решается на неравномерной сетке, меняющейся с течением времени.

Для решения поставленной задачи вводятся дополнительные безразмерные переменные и размерные коэффициенты:

$$V = U/l_{кр}; \quad \mu = \lambda/l_{кр}; \quad \Psi = R/l_{кр}; \quad \tau = \eta_1 t; \quad P = \eta_2 \sigma;$$

$$\eta_1 = \frac{Gb^2}{Bl_{кр}^2}, \quad [\eta_1] = \frac{1}{c}; \quad \eta_2 = \frac{l_{кр}}{Gb}, \quad [\eta_2] = \frac{M}{H}.$$

С помощью данного преобразования порядок величин, входящих в уравнение, существенно сблизится (например, порядок t увеличится с 10^{-9} до 10^{-2} , а порядок σ уменьшится с 10^6 до 10^0), что повысит точность решения. Подставим их в исходные уравнения (1–3) и получим эквивалентное безразмерное уравнение

$$\frac{\partial V}{\partial \tau} = \frac{\partial^2 V}{\partial \mu^2} + P - \frac{1}{\Psi},$$

которое необходимо перевести в разностное уравнение. Для перевода используется двухслойная схема с одним из трех шаблонов: явным четырехточечным, неявным четырехточечным (с опережением) и неявным шеститочечным (симметричным) [10]. Пусть i — номер точки на кривой AB ($i = 1, N$), j — номер временного слоя ($j = 1, \infty$); $\Delta \tau$ — значение временного шага; V_i^j — значение смещения i -й точки на j -м временном слое; μ_i^j — криволинейная координата i -й точки; $h_i^j = \mu_{i+1}^j - \mu_i^j$ — переменный шаг по μ между i -й и $(i+1)$ -й точками; Ψ_i^j — радиус кривизны в i -й точке кривой AB j -го временного слоя. Для сокращения записи h_i^j обозначается через h_i ; $(h_i^j + h_{i+1}^j)/2$ — через \tilde{h}_i , а Ψ_i^j — через Ψ_i . Для трех перечисленных шаблонов с использованием введенных обозначений получаются три разностных уравнения для поиска смещения i -й точки j -го временного слоя:

$$\frac{V_i^{j+1} - V_i^j}{\Delta \tau} = P - \frac{1}{\Psi_i} + \frac{1}{\tilde{h}_i} \left(\frac{V_{i+1}^j}{h_{i+1}} - \frac{V_i^j}{h_{i+1}} - \frac{V_i^j}{h_i} + \frac{V_{i-1}^j}{h_i} \right);$$

$$\begin{aligned} & \frac{V_i^{j+1} - V_i^j}{\Delta\tau} = \\ & = P - \frac{1}{\Psi_i} + \frac{1}{\tilde{h}_i} \left(\frac{V_{i-1}^{j+1}}{h_i} - \frac{V_i^{j+1}}{h_i} - \frac{V_i^{j+1}}{h_{i+1}} + \frac{V_{i+1}^{j+1}}{h_{i+1}} \right); \\ & \frac{V_i^{j+1} - V_i^j}{\Delta\tau} = P - \frac{1}{\Psi_i} + \frac{1}{2\tilde{h}_i} \left(\frac{V_{i-1}^{j+1}}{h_i} + \frac{V_{i-1}^j}{h_i} - \right. \\ & \left. - \frac{V_i^{j+1}}{h_i} - \frac{V_i^j}{h_i} - \frac{V_i^{j+1}}{h_{i+1}} - \frac{V_i^j}{h_{i+1}} + \frac{V_{i+1}^{j+1}}{h_{i+1}} + \frac{V_{i+1}^j}{h_{i+1}} \right), \end{aligned}$$

которые после введения величин $k_{1i} = \Delta\tau/(\tilde{h}_i h_i)$, $k_{2i} = \Delta\tau/(\tilde{h}_i h_{i+1})$, преобразуются в уравнения следующего вида соответственно:

$$\begin{aligned} V_i^{j+1} &= \Delta\tau \left(P - \frac{1}{\Psi_i} \right) + k_{1i} V_{i-1}^j + \\ &+ (1 - k_{1i} - k_{2i}) V_i^j + k_{2i} V_{i+1}^j, \end{aligned} \quad (4)$$

где $i = 2, \dots, N-1$,

$$\begin{aligned} -k_{1i} V_{i-1}^{j+1} + (1 + k_{1i} + k_{2i}) V_i^{j+1} - k_{2i} V_{i+1}^{j+1} &= \\ = V_i^j + \Delta\tau \left(P - \frac{1}{\Psi_i} \right), \end{aligned} \quad (5)$$

где $i = 2, \dots, N-1$;

$$\begin{aligned} -k_{1i} V_{i-1}^{j+1} + (2 + k_{1i} + k_{2i}) V_i^{j+1} - k_{2i} V_{i+1}^{j+1} &= \\ = k_{1i} V_{i-1}^j + (2 - k_{1i} - k_{2i}) V_i^j + \\ + k_{2i} V_{i+1}^j + \Delta\tau \left(P - \frac{1}{\Psi_i} \right), \end{aligned} \quad (6)$$

где $i = 2, \dots, N-1$.

Ко всем уравнениям (4)–(6) в качестве граничных условий добавляется жесткое закрепление первой и последней точек для любого временного слоя $V_1^j = V_N^j = 0 \forall j$, получают замкнутые системы N линейных уравнений относительно N величин V_i^{j+1} . В начальный момент времени ($j = 0$) во всех точках принимается $V_i^0 = 0 \forall i$. В первой системе все величины смещения на $(j+1)$ -м временном слое V_i^{j+1} выражаются через смещения V_i^j на j -м слое явно. Вторая и третья неявные системы уравнений решаются методом прогонки [11, 12]. Результатом решения любой из этих систем являются значения смещения V_i^{j+1} каждой i -й точки для ее перехода с j -го временного слоя на $(j+1)$ -й слой.

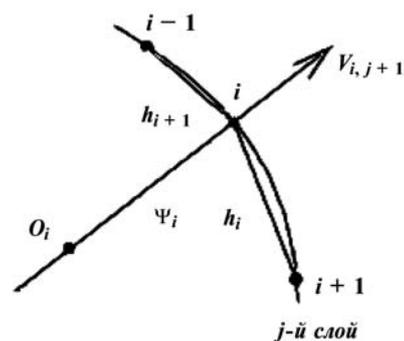


Рис. 3. Нахождения радиуса кривизны и направления смещения

В исходном уравнении (1)–(3) существуют две отличительные особенности, которые вносят в него нелинейность. Во-первых, из принятых предположений, направление смещения V_i^{j+1} i -й точки должно быть по нормали к кривой AB j -го временного слоя. Во-вторых, при составлении уравнений системы необходимо знать радиус кривизны Ψ_i^j (нелинейный член) для каждой i -й точки. Оба эти вопроса решаются одновременно следующим образом (рис. 3). По любым трем точкам $(i-1, i, i+1)$ j -го временного слоя кривой AB строится окружность с центром в точке O_i :

$$\begin{aligned} x_{O_i} &= \left(\frac{x_{i-1}^2 - x_i^2}{2(y_{i-1} - y_i)} - \frac{x_i^2 - x_{i+1}^2}{2(y_i - y_{i+1})} - \right. \\ &\left. - \frac{y_{i-1} - y_{i+1}}{2} \right) / \left(\frac{x_{i-1} - x_i}{y_{i-1} - y_i} - \frac{x_i - x_{i+1}}{y_i - y_{i+1}} \right); \\ y_{O_i} &= \frac{x_{i-1}^2 - x_i^2}{2(y_{i-1} - y_i)} - \frac{x_{i-1} - x_i}{y_{i-1} - y_i} x_{O_i} + \frac{y_{i-1} + y_i}{2}. \end{aligned}$$

За радиус кривизны Ψ_i^j принимается радиус этой окружности на j -м временном уровне:

$$\Psi_i = \sqrt{(x_{O_i} - x_i)^2 - (y_{O_i} - y_i)^2},$$

а за направление смещения i -й точки принимается направление от центра O_i к этой точке. Таким образом, в результате решения уравнения для каждой i -й точки j -го временного слоя рассчитываются значение и направление смещения и, следовательно, новая конфигурация точек ДС на следующем временном слое.

3. Выбор разностной схемы, исследование ее сходимости, устойчивости, точности

Из трех разностных схем необходимо выбрать схему, которая является устойчивой и с достаточ-

ной точностью и скоростью сходится к решению. Для этого необходимо рассмотреть неоднородное уравнение теплопроводности с постоянным коэффициентом в следующем виде:

$$\frac{\partial U}{\partial t} = k \frac{\partial^2 U}{\partial x^2} + f(x, t).$$

Вид уравнения и обозначения в данном разделе принимаются как в работе [10]. Значение коэффициента k не должно выходить из некоторого интервала $[c_1, c_2]$, а поскольку в данном случае k — константа ($k > 0$), то можно положить, что $c_1 = c_2 = k$. Применительно к решению задачи движения дислокационного сегмента значения параметров уравнения примут следующие значения: $k = 1, f(x, t) = P - 1/\Psi(x, t)$.

При построении разностной схемы необходимо учитывать, что расчетная сетка неравномерна по x и перемещение точек кривой происходит вне сеточной области, поэтому оценку устойчивости и сходимости необходимо проводить не в среднем, по сеточной норме, а в области непрерывных функций (равномерная сходимость). В работе [10] показано, что схема с весами (к которой относятся все три выбранных шаблона) на любой последовательности неравномерных сеток устойчива и равномерно сходится со скоростью $O(\tau^{m_\sigma} + |h|^2)$, если f — гладкая функция и выполняется соотношение $\sigma \geq \sigma_\varepsilon$. Здесь h — шаг; τ — шаг по времени; m_σ — параметр, значение которого равно либо 2 (при $\sigma = 0,5$), либо 1 (при $\sigma \neq 0,5$); σ — параметр схемы с весами, отвечающий за выбор шаблона ($\sigma = 0$ — явный четырехточечный шаблон, $\sigma = 0,5$ — неявный шеститочечный шаблон, $\sigma = 1$ — неявный четырехточечный шаблон); $|h| = \max_{1 \leq i \leq N-1} h_i$; $\sigma_\varepsilon = 0,5 - (1 - \varepsilon)/(\tau \|A\|)$;

$0 < \varepsilon \leq 1$ — произвольное число, независимое от h и τ ; A — линейный самосопряженный, положительно определенный оператор $A: H \rightarrow H$ с нормой $\|A\| \leq 4c_2/h_0^2$; $h_0 = \min_{1 \leq i \leq N-1} h_i$. Исходя из оценки нормы оператора A можно оценить значение величины σ_ε : $\sigma_\varepsilon \geq 0,5 - 0,25(1 - \varepsilon)/kh_0^2/\tau$, а так как $\sigma \geq \sigma_\varepsilon$, то

$$\sigma \geq 0,5 - \frac{1 - \varepsilon}{4k} \frac{h_0^2}{\tau}. \quad (7)$$

Из неравенства видно, что при $\sigma = 0$ выполняется соотношение $\tau/h_0^2 \leq 0,5(1 - \varepsilon)/k$, из которого следует, что между τ и h существует зависимость ($\tau/h_0^2 \rightarrow 0$ при $\varepsilon \rightarrow 1$), т. е. явная четырехточечная

схема условно устойчива и сходится со скоростью $O(\tau + |h|^2)$.

При $\sigma = 0,5$ из неравенства (7) вытекает, что $0,25(1 - \varepsilon)/kh_0^2/\tau \geq 0$, — выполняется при любых значениях параметров, и, следовательно, симметричная неявная шеститочечная схема абсолютно устойчива и сходится со скоростью $O(\tau^2 + |h|^2)$.

При $\sigma = 1$ из выражения (7) вытекает, что $0,5(1 - \varepsilon)/kh_0^2/\tau \geq -1$, — выполняется при любых значениях параметров, и, следовательно, неявная четырехточечная схема с опережением также абсолютно устойчива и сходится со скоростью $O(\tau + |h|^2)$.

Таким образом, для решения задачи можно выбрать любую из трех схем, поскольку все они устойчивы и сходятся к решению. Неявные схемы предпочтительны, так как сходятся при любом соотношении значений τ и h . Погрешность шеститочечной схемы по τ меньше, чем четырехточечной, за счет этого при вычислениях можно взять больший шаг по времени и, как следствие, увеличить скорость расчетов.

Вместе с тем, поскольку обе неявные схемы устойчивы и граничные условия нулевые ($U(0) = U(l_0) = 0$), то для них должно выполняться следующее неравенство [10]:

$$\|U_i^{j+1}\|_C \leq \|F_i^j\|_C. \quad (8)$$

Здесь F_i^j — правая часть уравнения (5) или (6).

Другими словами, смещения $\max_{1 < i < N} |U_i^{j+1}|$ точек на $(j + 1)$ -м слое не должны превышать значения $\max_{1 < i < N} |F_i^j|$. Оценим величину F_i^j для четырехточечной и шеститочечной схем. Для схемы с опережением имеем

$$F_i^j = V_i^j + \left(P - \frac{1}{\Psi_i^j} \right).$$

Оба слагаемых данного выражения имеют один и тот же порядок малости: в том случае, если значение P больше $1/\Psi$, смещение сегмента V в данной точке по сравнению с предыдущим шагом увеличивается, причем это увеличение становится более значимым с ростом Ψ ; в том случае, если значение P меньше $1/\Psi$, смещение сегмента в данной точке уменьшается; если же $P \approx 1/\Psi$, то точка сегмента остается без движения. Поскольку значение радиуса Ψ вдоль сегмента изменяется плавно, то значение F от точки к точке сегмента также изменяется без скачков.

Для симметричной схемы в выражении появляется третье слагаемое:

$$F_i^j = V_i^j + \left(P - \frac{1}{\Psi_i^j} \right) + \frac{\Delta\tau}{\tilde{h}_i} \left(\frac{V_{i+1}^j - V_i^j}{h_{i+1}} - \frac{V_i^j - V_{i-1}^j}{h_i} \right),$$

имеющее такой же порядок малости, что и первые два. Для третьего слагаемого характерно, что оно зависит и от шага h и от соотношения $\Delta\tau$ и h . При нормальном развитии дислокации в каждой точке сегмента третье слагаемое должно быть одного знака, это означает, что данная линия без перегибов с отрицательной второй производной (выпуклая линия). В связи с тем, что шаг h переменный, мал и во время движения сегмента может принимать разные значения, существует вероятность того, что это слагаемое в какой-либо точке изменит знак, тогда на дислокационной кривой появится точка излома, вследствие чего нарушится условие (8), что может привести к сбою в работе алгоритма и программы. При нормальной работе обеих схем разность между ними практически отсутствует, но с точки зрения надежности (и это подтверждается при испытаниях программы), предпочтительнее использовать для решения исходной задачи четырехточечную схему с опережением.

4. Исследование сходимости, устойчивости, точности метода прогонки

Решение системы линейных алгебраических уравнений единственно, сходится и устойчиво по входным данным, если для данной системы выполняется условие диагонального преобладания [11, 12]. Для выбранной системы эти условия выглядят следующим образом:

$$|1 + k_{1i} + k_{2i}| \geq |k_{1i}| + |k_{2i}| \text{ при } i = 3, N - 2;$$

$$|1 + k_{1i} + k_{2i}| \geq |k_{2i}| \text{ при } i = 2;$$

$$|1 + k_{1i} + k_{2i}| \geq |k_{1i}| \text{ при } i = N - 1,$$

поэтому выполняются всегда.

Точность решения системы, в том числе и методом прогонки, оценивается мерой обусловленности ν исходной матрицы C , составленной по уравнениям (5): $\nu = \text{cond}(C) = \|C\| \|C^{-1}\|$. Поскольку процесс вычисления точного значения C^{-1} трудоемкий, то можно дать приближенную оценку точности. Предположим, что все значения шагов h_i^j одинаковы и равны максимальному значению шага $h = \max_{1 \leq i \leq N-1} h_i$, в этом случае матрица C после эквивалентных преобразований будет выглядеть следующим образом:

$$C = \begin{pmatrix} -2 - \frac{h^2}{\Delta\tau} & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 - \frac{h^2}{\Delta\tau} & 1 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 1 - 2 - \frac{h^2}{\Delta\tau} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 - \frac{h^2}{\Delta\tau} \end{pmatrix}.$$

Если обозначить коэффициенты главной диагонали матрицы C через $q = -2 - h^2/\Delta\tau$, то норма матрицы C будет равна [13]

$$\|C\| = 2 + |q| = 2 - q, \quad (9)$$

поскольку значение h и $\Delta\tau > 0$, а значение $q < -2$ всегда.

Для нахождения нормы обратной матрицы C^{-1} рассмотрим процесс вычисления этой матрицы. Обратная матрица [13] может быть найдена из условия $C \times C^{-1} = E$, где E — единичная матрица размерности $M = N - 2$. Обозначая матрицу C^{-1} через S , можно записать следующее выражение:

$$\begin{pmatrix} q & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & q & 1 & \dots & 0 & 0 & 0 \\ 0 & 1 & q & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & q & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & q & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & q \end{pmatrix} \begin{pmatrix} s_{1,1} & s_{1,2} & s_{1,3} & \dots & s_{1,M-2} & s_{1,M-1} & s_{1,M} \\ s_{2,1} & s_{2,2} & s_{2,3} & \dots & s_{2,M-2} & s_{2,M-1} & s_{2,M} \\ s_{3,1} & s_{3,2} & s_{3,3} & \dots & s_{3,M-2} & s_{3,M-1} & s_{3,M} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ s_{M-2,1} & s_{M-2,2} & s_{M-2,3} & \dots & s_{M-2,M-2} & s_{M-2,M-1} & s_{M-2,M} \\ s_{M-1,1} & s_{M-1,2} & s_{M-1,3} & \dots & s_{M-1,M-2} & s_{M-1,M-1} & s_{M-1,M} \\ s_{M,1} & s_{M,2} & s_{M,3} & \dots & s_{M,M-2} & s_{M,M-1} & s_{M,M} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix}. \quad (10)$$

Учитывая специфику данной системы, поиск элементов матрицы S можно разбить на M несвязных между собой подсистем, в каждой из которых неизвестными будут элементы какого-либо столбца матрицы S : $C \times S_i = E_i$, где S_i, E_i — i -е столбцы матриц S и E соответственно ($i = 1, 2, 3, \dots, M$), или

$$\begin{pmatrix} q & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & q & 1 & \dots & 0 & 0 & 0 \\ 0 & 1 & q & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & q & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & q & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & q \end{pmatrix} \begin{pmatrix} s_{1,i} \\ s_{2,i} \\ s_{3,i} \\ \dots \\ s_{M-2,i} \\ s_{M-1,i} \\ s_{M,i} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 1 \\ \dots \\ 0 \\ 0 \end{pmatrix}. \quad (11)$$

Решая данную i -ю подсистему по методу Крамера [13], получаем элементы i -го столбца матрицы S . Каждое j -е неизвестное ($j = 1, 2, 3, \dots, M$) является дробью, в знаменателе которого стоит главный определитель системы, а в числителе — определитель системы с заменой j -го столбца на столбец свободных членов (j -й определитель). Очевидно, что главный определитель системы представляет из себя полином от q степени M , причем коэффициент при старшем члене равен 1 (произведение членов главной диагонали). Каждый j -й определитель системы (числитель) также представляет из себя полином от q , причем наибольшая степень у j -го определителя будет при $j = i$. В этом случае элемент (i, i) главной диагонали становится равным 1, и степень полинома будет равна $M - 1$ с коэффициентом 1 при старшей степени. В предположении, что q по абсолютной величине является достаточно большой, с помощью i -го определителя находим максимального элемента системы (10):

$$s_{i,i} = \left| \frac{q^{M-1} + \dots}{q^M + \dots} \right|.$$

Таким образом, порядок максимального элемента решения любой i -й системы (11) равен $|1/q|$, этой же величине равен и порядок максимального элемента в каждом столбце матрицы C^{-1} . В этом случае порядок нормы обратной матрицы

$$\|C^{-1}\| = |1/q| = -1/q, \quad (12)$$

и, следовательно, получаем итоговую обусловленность матрицы C из (9), (12):

$$\nu = \frac{q-2}{q} = \frac{h^2/\Delta\tau + 4}{h^2/\Delta\tau + 2}.$$

Для того чтобы обеспечить достаточную точность решения, необходимо, чтобы $\nu \rightarrow 1$, другими словами должно выполняться следующее неравенство: $\Delta\tau \ll h^2$.

5. Алгоритм решения уравнения

Переход ДС в новое положение (см. рис. 2) может осуществляться двумя разными способами. По первому способу (рис. 4, а) после проведения только одного шага решения уравнения осуществляется и расчет нового положения 2 дислокационной кривой, т. е. после нахождения значения смещения v_i точек сразу определяются новые координаты точек кривой. При этом на каждом шаге решения уравнения начальные условия, т. е. начальные смещения точек, будут считаться нулевыми.



Рис. 4. Два способа перехода дислокационной кривой в новое положение

При втором, аддитивном, способе выполняются несколько последовательных шагов решения уравнения движения (рис. 4, б), при этом только на первом шаге начальные условия нулевые, а при расчете всех последующих шагов учитываются смещения, найденные на предыдущем шаге. Смещения v_i каждой i -й точки пошагово суммируются и для всех суммарных смещений должно выполняться условие применимости уравнения (1)–(3), т. е. $\sum v_i^{j+1} \ll L^j$. Расчет нового положения точек кривой происходит только после нарушения данного условия, при этом для вычисления нового положения точек используются начальные координаты и кривизна точек кривой в положении 1, а также найденные суммарные смещения $V_i^{j+1} = \sum v_i^{j+1}$. Результаты, полученные с помощью обоих способов, как показывают сравнительные расчеты, примерно одинаковые, но второй способ дает существенный выигрыш по времени.

Выводы

Таким образом, показано, что при использовании четырехточечного шаблона конечно-разностный метод решения задачи эволюции дислокационного источника Франка — Рида сходится и абсолютно устойчив. Метод прогонки решения системы алгебраических уравнений также сходится, устойчив и достаточно точен. Для ускорения работы алгоритма решения задачи применяют аддитивный способ нахождения нового положения точек дислокационного сегмента.

Список литературы

1. Frank F. C., Read W. T. // Phys. Rev. 1953. 89. P. 663.
2. Фридель Ж. Дислокации. М.: Мир, 1967.
3. Предводителев А. А., Тяпунина Н. А., Зиненкова Г. М., Бушуева Г. В. Физика кристаллов с дефектами. М.: Изд-во МГУ, 1986.
4. Нацик В. Д., Чижко К. А. Динамика и звуковое излучение дислокационного источника Франка — Рида. II. Формирование дислокационного скопления. Препринт. Харьков: ФТИИТ АН УССР, 1976.
5. Tyapunina N. A., Blagoveshchenskii V. V. Dislocation Behaviour and Multiplication under Ultrasound // Phys. Stat. Sol. (a). 1982. V. 69. P. 77–83.
6. Тяпунина Н. А., Благовещенский В. В. Особенности работы источника Франка — Рида под действием ультразвука // ДАН СССР. 1980. 254, № 4. С. 869.
7. Благовещенский В. В. Эволюция дислокационной структуры под действием ультразвука и неупругость кристаллов: дис. ... доктора физ.-мат. наук. Кострома, КГТУ, 2001. 237 с.
8. Тихонов А. А., Самарский А. А. Уравнения математической физики. М.: Наука, 1966.
9. Хемминг Р. В. Численные методы. М.: Наука, 1968.
10. Самарский А. А. Теория разностных схем. М.: Наука, 1983.
11. Самарский А. А., Николаев Е. С. Методы решения сеточных уравнений. М.: Наука, 1978.
12. Райс Дж. Матричные вычисления и математическое обеспечение. М.: Мир, 1984.
13. Корн Г., Корн Т. Справочник по математике. М.: Наука, 1978.

УДК 519.873

Я. Е. Львович, д-р техн. наук, проф., зав. каф.,
И. Л. Каширина, канд. техн. наук, доц.,
А. А. Тузиков, аспирант,
e-mail: Kash.irina@mail.ru,
Воронежский государственный
технический университет

Генетический алгоритм решения многокритериальной задачи повышения надежности резервирования

Процесс автоматизированного проектирования систем является задачей многокритериальной оптимизации, один из критериев которой — обеспечение заданного уровня показателей надежности. Актуальность научных работ данной тематики состоит в том, что не существует универсальных формализованных методов, алгоритмов, программ, позволяющих автоматизировать процесс управления надежностью для любой сложной технической системы на всех этапах ее жизненного цикла. В рамках данной работы предлагаются математическая модель и генетический алгоритм для решения задачи повышения надежности резервирования компонентов стендовой информационно-управляющей системы для проведения огневых испытаний жидкостных ракетных двигателей.

Ключевые слова: жидкостный ракетный двигатель, система управления испытаниями, резервирование, многокритериальная математическая модель, генетический алгоритм

Общие сведения о надежности аппаратных средств и методах резервирования

При проведении огневых испытаний жидкостных ракетных двигателей (ЖРД) предъявляются особые требования к безотказности технических

средств автоматизированных систем управления технологическими процессами (АСУ ТП). Одними из основных показателей безотказности технических средств АСУ ТП являются среднее время наработки на отказ T и вероятность безотказной работы $P(t)$.

Для объектов промышленной автоматизации вероятность безотказной работы можно определить как [1]

$$P(t) = e^{-\lambda t}, \quad (1)$$

где $\lambda = \text{const}$ — интенсивность отказов.

При этом частота отказов по определению [1] является плотностью распределения времени до отказа $f(t)$:

$$f(t) = \frac{d[1 - P(t)]}{dt} = \lambda e^{-\lambda t}. \quad (2)$$

Среднее время наработки на отказ T определяется как математическое ожидание длительности безотказной работы t :

$$T = \int_0^{\infty} t f(t) dt = \frac{1}{\lambda}. \quad (3)$$

Одним из самых радикальных способов повышения безотказности (надежности) является резервирование. В промышленной автоматизации наибольшее распространение получили следующие методы резервирования: резервирование замещением "один из двух" (1oo2 — 1 out of 2) и метод мажоритарного голосования "два из трех" (2oo3) [2]. Системы без резервирования классифицируются как 1oo1.

Обозначим A_i — событие, означающее безотказную работу i -го элемента системы, а отказ — как \bar{A}_i . При этом очевидно, что вероятность безотказной работы нерезервированной системы будет определяться выражением (1) $P_{1oo1}(t) = P$, а среднее время наработки на отказ согласно (3) $T_{1oo1} = 1/\lambda = T$.

Рассмотрим резервированную систему, состоящую из двух элементов (1oo2). События A_1 и A_2

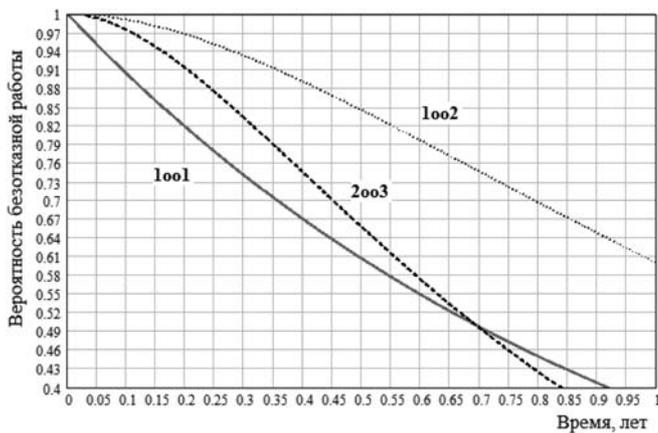


Рис. 1. Вероятность безотказной работы 1оо1, 1оо2 и 2оо3 при $\lambda = 1 \text{ год}^{-1}$

означают безотказную работу этих элементов, следовательно, событие A , означающее безотказную работу всей системы, определяется как

$$A = A_1 A_2 + \bar{A}_1 A_2 + A_1 \bar{A}_2 = A_1 + \bar{A}_1 A_2. \quad (4)$$

Учитывая, что элементы резервированной системы идентичны и вероятности событий A_1 и A_2 равны P , то вероятность безотказной работы системы 1оо2:

$$P_{1оо2}(t) = P(A) = P(A_1) + P(\bar{A}_1 A_2) = P(A_1) + [1 - P(A_1)]P(A_2) \quad (5)$$

или

$$P_{1оо2}(t) = 2P - P^2 = 2e^{-\lambda t} - e^{-2\lambda t}. \quad (6)$$

Продифференцировав (6) в соответствии с (2) и проинтегрировав в соответствии с (3), получим:

$$T_{1оо2} = 2\lambda \int_0^{\infty} t(e^{-\lambda t} - e^{-2\lambda t}) dt = \frac{3}{2\lambda} = \frac{3}{2} T. \quad (7)$$

Аналогично для резервированной схемы 2оо3 можно получить:

$$A = A_1 A_2 A_3 + \bar{A}_1 A_2 A_3 + A_1 \bar{A}_2 A_3 + A_1 A_2 \bar{A}_3; \quad (8)$$

$$P_{2оо3}(t) = 3P^2 - 2P^3 = 3e^{-2\lambda t} - 2e^{-3\lambda t}; \quad (9)$$

$$T_{2оо3} = 6\lambda \int_0^{\infty} t(e^{-2\lambda t} - e^{-3\lambda t}) dt = \frac{5}{6\lambda} = \frac{5}{6} T. \quad (10)$$

На рис. 1 показаны вероятности безотказной работы систем 1оо1, 1оо2 и 2оо3 при $\lambda = 1 \text{ год}^{-1}$.

Приравняв уравнения (1) и (9), можно узнать критическое время $T_{2оо3кр} = T \ln 2 \approx 0,7T$, после которого вероятность безотказной работы системы 2оо3 становится меньше, чем у системы 1оо1, а использование резервирования становится неэффектив-

ным. Проанализировав соотношения (1) и (6), можно сказать, что вероятность безотказной работы системы 1оо2 всегда больше, чем у системы 1оо1. Однако системе резервирования 1оо2 присущ ряд недостатков, рассматриваемых далее.

Архитектура систем управления огневыми испытаниями ЖРД

Системой управления двигателя называется комплекс устройств, посредством которых осуществляется запуск, останов, изменение режимов работы и контроль параметров двигателя. В основу системы управления положена релейная автоматика [3], основными элементами которой являются дискретные ключевые элементы, обеспечивающие подачу и снятие команд управления и имеющие состояния: "замкнут", "разомкнут". Ключевой элемент является одним из звеньев канала (тракта) управления.

Высокие требования, предъявляемые к безотказности систем управления испытаниями ЖРД, относятся в первую очередь к надежности релейной автоматики. Так как ключевой элемент имеет только два состояния, то и видов отказов может быть только два: короткое замыкание (КЗ) и обрыв. Ключевой элемент обычно представляет собой электромагнитное реле или полупроводниковый прибор (транзистор).

На рис. 2 представлена структурная схема резервирования 1оо2 для ключевых элементов. Условные обозначения: КУ — команда дискретного управления; ИП — источник питания; K_1 — ключ; ОУ — объект управления.

На рис. 2, а ключи объединены по схеме ИЛИ. Недостатком данной схемы является то, что при возникновении отказа типа КЗ невозможно снять команду с объекта управления. Схема, представленная на рис. 2, б, нечувствительна к отказам типа КЗ из-за наличия блока переключения на резерв. Однако данной схеме также присущи недостатки: блок переключения на резерв должен быть абсолютно надежным (а он в общем случае также состоит из ключей), требуется достаточно информативный сигнал диагностики для определения момента возникновения отказа, имеется ограничение по времени переключения на резерв. Для некоторых резервируемых элементов такой блок переключения на резерв может просто отсутствовать.

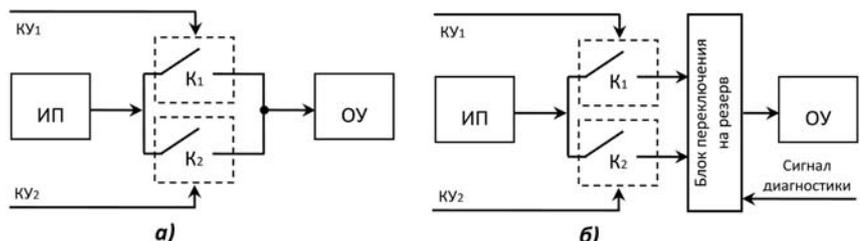


Рис. 2. Резервирование ключевых элементов по схеме 1оо2

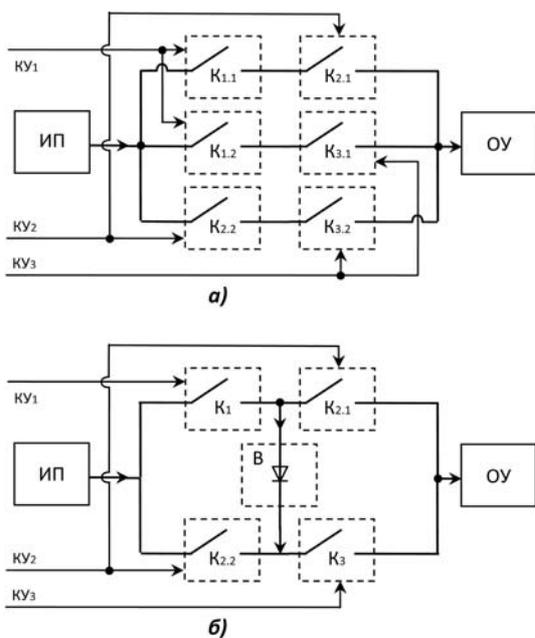


Рис. 3. Резервирование ключевых элементов по схеме 2oo3

На рис. 3, а представлена классическая структурная схема резервирования 2oo3 для ключевых элементов.

Как видно из рис. 3, а, при классическом резервировании 2oo3 требуется в 6 раз больше ключей, чем в схеме 1oo1. Этот недостаток можно частично устранить, применив упрощенную схему рис. 3, б. В схеме на рис. 3, б всего четыре ключа и один вентиль (В). Назначение вентиля В заключается в том, чтобы разрешать подачу команды на ОУ только по направлению, указанному стрелкой. Вентиль обеспечивает совместную работу ключей K_1 и K_3 . В качестве вентиля широкое применение нашли полупроводниковые диоды, надежность которых как минимум на порядок выше надежности ключей, поэтому при анализе надежности всего ключевого элемента их можно вообще не учитывать. Тем не менее, эта схема остается самой дорогостоящей (по сравнению с 1oo1 и 1oo2).

Модель выбора вариантов резервирования компонентов стендовой информационно-управляющей системы

Для выбора вариантов резервирования компонентов стендовой информационно-управляющей системы предлагается оптимизационная модель. Каждому i -му компоненту может быть назначен один из трех вариантов резервирования:

1 — элемент ставится без резервирования (1oo1);
2 — резервирование замещением "один из двух" (1oo2);

3 — метод мажоритарного голосования "два из трех" (2oo3).

Пусть n — общее число резервируемых компонентов.

Введем переменные:

$x_{ij} = 1$, если j -му компоненту назначается i -й вариант резервирования;

$x_{ij} = 0$, в противном случае; $i = \overline{1, 3}$, $j = \overline{1, n}$.

Важным ограничением является фиксированное среднее время безотказной работы системы (наработка до отказа).

Как следует из формулы (7), среднее время безотказной работы элемента, резервируемого по схеме 1oo2, составляет $3/2$ от среднего времени работы нерезервируемого элемента, а согласно формуле (10) среднее время безотказной работы элемента по схеме 2oo3 соответственно равно $5/6$ от среднего времени работы 1oo1.

Исходя из того, что $T = 1/\lambda$, наработку до отказа всей системы $T_{\text{ср}}$ можно представить в виде

$$\frac{1}{T_{\text{ср}}} = \sum_{j=1}^n \frac{1}{T_j},$$

где T_j — наработка до отказа j -го элемента.

Тогда первое ограничение имеет вид

$$\sum_{j=1}^n \left(\frac{1}{T_j} x_{1j} + \frac{2}{3T_j} x_{2j} + \frac{6}{5T_j} x_{3j} \right) \leq \frac{1}{T_{\text{ср}}}, \quad (11)$$

где T_j — среднее время наработки до отказа j -го элемента системы без резервирования.

Поскольку каждому элементу назначается ровно один метод резервирования, вторая группа ограничений задачи имеет вид

$$x_{1j} + x_{2j} + x_{3j} = 1, \quad j = \overline{1, n}. \quad (12)$$

В качестве критериев оптимизации могут рассматриваться общая стоимость системы и вероятность безотказной работы всех ее компонентов.

Стоимость элемента, резервируемого по схеме 1oo2 в два раза выше стоимости элемента без резервирования, а стоимость элемента, резервируемого по схеме 2oo3, как показано выше, в четыре раза выше стоимости нерезервируемого элемента. Однако схема 1oo2 не всегда реализуема, так как для нее необходим абсолютно надежный блок переключения на резерв, который для некоторых резервируемых элементов может отсутствовать. Если же такой блок присутствует, его стоимость может увеличивать общую стоимость данного варианта резервирования.

Поэтому первая целевая функция имеет вид

$$\sum_{j=1}^n (S_j x_{1j} + 2S_j G_j x_{2j} + 4S_j x_{3j}) \rightarrow \min. \quad (13)$$

Здесь S_j — стоимость j -го элемента резервирования; $G_j \geq 1$ — коэффициент, увеличивающий стоимость схемы 1oo2 в случае, если для данного резервируемого элемента существует надежный блок переключения на резерв, и $G_j = S$ в случае, если

такой блок отсутствует (S — максимально возможная суммарная стоимость резервируемых элементов, выполняет роль штрафного коэффициента). Заметим, что если стоимости резервируемых элементов примерно равны, то критерий (13) можно интерпретировать как минимизацию общего числа элементов системы (а следовательно, и ее габаритные размеры).

В качестве второго критерия оптимизации может рассматриваться вероятность безотказной работы всех компонентов системы.

Как следует из формул (6) и (9), если P_j — вероятность безотказной работы j -го элемента без резервирования, то в схеме 1002 вероятность безотказной работы имеет вид $2p_j - p_j^2$, а в схеме 2003 эта вероятность равна $3p_j^2 - 2p_j^3$. Таким образом, вторая целевая функция имеет вид

$$P_{\text{общ}} = \prod_{j=1}^n (p_j x_{1j} + (2p_j - p_j^2)x_{2j} + (3p_j^2 - 2p_j^3)x_{3j}) \rightarrow \max. \quad (14)$$

В итоге, модель выбора вариантов резервирования компонентов стандовой информационно-управляющей системы принимает следующий вид:

$$S_{\text{общ}} = \sum_{j=1}^n (S_j x_{1j} + 2S_j G_j x_{2j} + 4S_j x_{3j}) \rightarrow \min;$$

$$P_{\text{общ}} = \prod_{j=1}^n (p_j x_{1j} + (2p_j - p_j^2)x_{2j} + (3p_j^2 - 2p_j^3)x_{3j}) \rightarrow \max;$$

$$\sum_{j=1}^n \left(\frac{1}{T_j} x_{1j} + \frac{2}{3T_j} x_{2j} + \frac{6}{5T_j} x_{3j} \right) \leq \frac{1}{T_{\text{сп}}};$$

$$x_{1j} + x_{2j} + x_{3j} = 1, j = \overline{1, n},$$

$$x_{ij} \in \{0, 1\}.$$

Данная задача может решаться как с помощью специализированных методов векторной оптимизации, так и сведением к скалярной задаче путем преобразования одного из критериев в ограничение. Далее предлагается генетический метод ее решения.

Генетический метод решения

Основной проблемой, касающейся большинства традиционных методов решения задач многокритериальной оптимизации, является необходимость перезапускать алгоритм несколько раз для получения аппроксимации множества эффективных точек (число прогонов метода, как минимум, равно мощ-

ности предполагаемой аппроксимации множества Парето). Это и само по себе не очень удобно при решении практических задач, однако имеется и еще более существенный недостаток — получаемая аппроксимация множества недоминируемых решений может оказаться нерепрезентативной, так как генерируемые эффективные точки будут неравномерно распределены как в пространстве альтернатив, так и в пространстве критериев. Зачастую можно получить набор очень близких друг к другу решений при потенциально большом множестве Парето. Основной причиной такой ситуации является то, что в каждом из запусков метода задача многокритериальной оптимизации сводится к задачам однокритериальной оптимизации. Таким образом, теряется суть решаемой задачи — одновременный учет многих критериев.

В отличие от классических подходов к многокритериальной оптимизации, генетические алгоритмы (ГА) принадлежат к разряду многоточечных поисковых методов. При решении многокритериальных задач ГА способны находить множество Парето-оптимальных решений за один прогон благодаря заложенному в них полимодальному поиску. Однако для обеспечения репрезентативной аппроксимации необходимо принимать специальные меры [4].

Рассмотрим подробно основные этапы ГА, разработанные применительно к данной многокритериальной задаче.

Генетический алгоритм решения многокритериальной задачи

1-й этап. Представление данных

Хромосома, представляющая неизвестную матрицу X , задается с помощью строкового кодирования. Суть кодирования заключается в следующем: экземпляр популяции — это строка длиной n (n — размерность задачи), в которой на i -м месте стоит $j \in \{1...3\}$, если $x_{ij} = 1$. Таким образом, решение

$$X = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \text{ (фенотип) будет записано как}$$

"троичная" строка: $x = (1, 1, 2, 3, 2, 3, 2)$ (генотип).

2-й этап. Генерация начальной популяции.

Первая популяция создается случайным образом из L различных "троичных" строк длиной n . Дополнительно (в целях улучшения генетического материала) в первую популяцию можно ввести строки, представляющие собой приближенные решения задач по обоим критериям в отдельности.

3-й этап. Оценка особей популяции по критерию приспособленности

1. Для каждого решения в популяции вычисляется вектор целей $(S_{\text{общ}}, P_{\text{общ}})$, по которому далее

определяется скалярное значение функции приспособленности.

2. Из текущей популяции выбирается множество недоминируемых внутри этой популяции решений, они запоминаются и временно выбрасываются из рассмотрения.

3. Далее ищется множество недоминируемых вариантов в усеченном множестве, и они исключаются. Эта процедура продлевается до тех пор, пока все варианты не будут исключены из популяции.

4. Теперь все строки ранжируются: принадлежащие последнему исключенному множеству получают ранг 1, предпоследнему — ранг 2. Строки, первыми выброшенные из рассмотрения, получают самый высокий ранг. Внутри каждого исключенного множества все варианты решения имеют одинаковый ранг.

5. Далее, в отдельности для каждой группы строк одного ранга происходит назначение оценок приспособленности. Предположим, ранг k имеет m строк. Тогда строка, сумма евклидовых расстояний от которой до остальных строк данного ранга максимальна, получит оценку $k + (m - 1)/m$.

Строка со второй по величине суммой расстояний получит оценку $k + (m - 2)/m$. Строка с минимальной суммой расстояний до остальных особей данного ранга будет иметь оценку k .

Такой подход к оцениванию экземпляров популяции настраивает алгоритм не только на поиск недоминируемых решений (любая недоминируемая строка будет иметь оценку выше любой доминируемой), но и на поддержание разнообразия популяции (удаленные точки получают более высокие оценки), что обеспечивает лучшую аппроксимацию Парето-оптимального множества.

4-й этап. Отбор (селекция)

В качестве процедуры селекции будем использовать стандартные механизмы пропорционального или турнирного отбора, и чем выше у индивида оценка приспособленности, тем вероятнее она попадет в родители следующего поколения.

5-й этап. Скрещивание и мутация

Для создания новых особей-потомков может использоваться любая стандартная мутация и любой

из классических операторов скрещивания (одноточечный, двухточечный, равномерный кроссовер), дополненные процедурой исправления недопустимых решений. Недопустимыми решения могут оказаться только за счет невыполнения ограничения (11). Это можно подкорректировать, исправив несколько координат в соответствующей строке (например, случайным образом значение 3 исправить на 2 или 1, либо значение 2 исправить на 1). Такие исправления позволят уменьшить левую часть ограничения (11). Они проводятся до тех пор, пока это ограничение не будет выполнено.

6-й этап. Критерий прекращения работы

В качестве рекорда хранится множество недоминируемых за все время работы алгоритма вариантов. Как критерий останова вычислений используется следующая проверка: если за последние 100 поколений рекордное множество не изменилось, то дальнейшая работа алгоритма прекращается.

Заключение

Эффективность алгоритма исследовалась на представительном множестве тестовых задач с различным числом переменных. Анализ эффективности работы метода проводился на основании сравнения качества аппроксимации множества Парето по следующим показателям: равномерность распределения генерируемых решений и число Парето-оптимальных решений в итоговой популяции. На рис. 4 для иллюстрации результатов приведены распределения недоминируемых точек, полученные после решения задачи генетическим алгоритмом. На рис. 4, а вероятности безотказной работы всех элементов без резервирования полагались равными $p = 0,7$. На рис. 4, б вероятности безотказной работы всех элементов без резервирования брались $p = 0,9$. По оси абсцисс откладывалось значение критерия $S_{\text{общ}}$, по оси ординат — $P_{\text{общ}}$.

Отмечено, что при решении тестовых задач предложенным методом обеспечивается достаточно хорошая представительность Парето-оптимального множества.

Список литературы

1. Половко А. М., Гуров С. В. Основы теории надежности. СПб.: БХВ-Петербург, 2008. 704 с.
2. МЭК 61508. Функциональная безопасность электрических, электронных и программируемых электронных систем, связанных с безопасностью.
3. Испытания жидкостных ракетных двигателей: учебное пособие для авиац. специальностей вузов / Под. ред. В. З. Левина. М.: Машиностроение, 1981. 199 с.
4. Каширина И. Л., Семенов Б. А. Генетический алгоритм решения многокритериальной задачи о назначениях // Информационные технологии. 2007. № 5. С. 62–68.

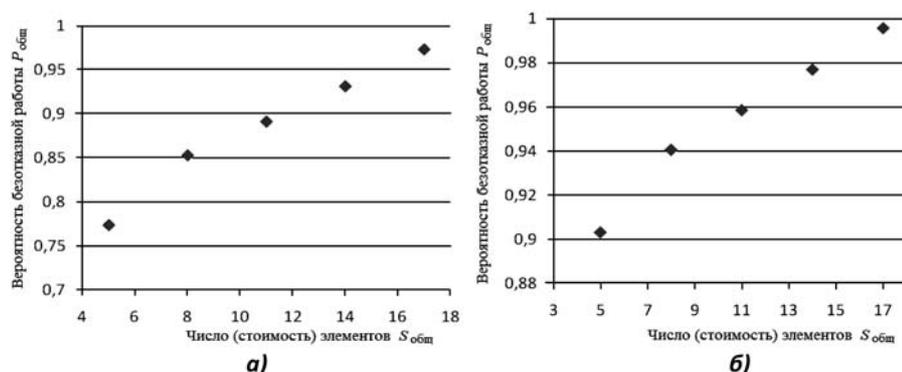


Рис. 4. Распределения недоминируемых точек

УДК 621.391

С. Н. Агиевич,
канд. техн. наук, ст. науч. сотр., вед. специалист,
ООО "Специальный технологический
центр", г. Санкт-Петербург,
e-mail: aserzhnic@mail.ru

Оценка параметров сигналов в базисах функций сплайн-характеров

Предлагаются методы и реализующие их алгоритмы оценки параметров сигналов в базисах функций сплайн-характеров (БФСХ). Описывается быстрое преобразование сигналов в БФСХ. Обосновывается его высокая вычислительная эффективность. Демонстрируется выигрыш в объеме вычислений при переходе от дискретных экспоненциальных функций к частному случаю БФСХ — базису сплайн-Виленкина — Крестенсона функций.

Анализируется эффективность метода оценки несущей частоты сигналов с позиций точности, скорости и вычислительных затрат.

Ключевые слова: оценивание параметров сигналов, функции сплайн-характеров, быстрые преобразования сигналов, базис сплайн-Виленкина — Крестенсона функции

Введение

В системах связи основой передачи информации являются радиосигналы, континуальная природа которых позволяет излучать их в эфир. Между тем, учитывая, что в настоящее время в большинстве задач приема и передачи радиосигналов активно применяются цифровые методы, целесообразно использовать дискретные формы их описания. Таким образом, возникает необходимость поиска компромиссных решений, позволяющих совмещать полезные свойства непрерывного и дискретного представления сигналов. Одно из них видится в использовании сплайнов, являющихся согласно [1] непрерывными функциями, но построенных на основе дискретных данных.

По сравнению с аппроксимацией дискретных данных на основе различных полиномов, сплайн-функции обладают следующими очевидными преимуществами:

- улучшенными свойствами аппроксимации;

- незначительностью влияния ошибок округления, возникающих в ходе вычислений;
- свойством сглаживания мест разрывов.

Таким образом, целесообразно исследовать возможность применения данного аппарата к решению задач алгоритмов цифровой обработки сигналов (ЦОС).

Определение сплайн-характеров

Пусть имеется пространство функций, заданных на абелевой группе H и принимающих значения в некотором кольце K . Это пространство обозначим $L(H, K)$. Аналогами комплексных экспонент в $L(H, K)$ являются характеры $\chi(n, k)$, образующие ортонормированный базис в пространстве $L(H, K)$ [1].

Характеры $\chi(n, k)$, определенные на конечном отрезке, называются χ -функциями. Здесь n — номер функции в базисе характеров (аналог понятия номера гармоники для гармонического базиса), k — текущее значение дискретного отсчета функции характера (аналог временных отсчетов для гармонических функций). В дальнейшем для общности будем использовать понятие характеры, а из контекста будет ясно, о каком случае (конечном или бесконечном) будет идти речь.

Одним из достоинств характеров $\chi(n, k)$ является их многообразие, определяемое многообразием групп H и колец K . Однако все они являются функциями дискретными, поэтому реализация на их основе сигналов для непосредственного излучения в эфир невозможна. Получить гладкие ортонормированные базисные функции оказалось возможным, объединив свойства $\chi(n, k)$ и сплайнов.

В результате в полученном из $L(H, K)$ пространстве G_n^p периодических сплайнов произвольный сигнал $S^p(t)$ можно разложить следующим образом:

$$S^p(t) = \frac{1}{N} \sum_k q_k M^p(t \ominus t_k) = \sum_n c_n^* \lambda_n^p(t), \quad (1)$$

где p — порядок сплайна; N — объем выборки (длина функции); q_k — коэффициенты сплайна (здесь и далее k — текущий номер); $M^p(t)$ — базисный B -сплайн; t — текущая координата времени; t_k — дискретные отсчеты времени; μ — модуль представления чисел; c_n^* — спектральные коэффициенты в базисе $\lambda_n^p(t)$, которые будем называть ортонор-

мированными сплайн-характерами; * — указывает на принадлежность к ортонормированному базису; \ominus — сдвиг по модулю μ .

Значение c_n^* будем рассчитывать по формуле

$$c_n^* = F_n(z) \sqrt{u_n^{2p}} / u_n^p,$$

где $F_n(z)$ — преобразование Фурье в базисе характеров над множеством z отсчетов сигнала; u_n^{2p} — дискретное преобразование Фурье от базисного B -сплайна $M^p(t)$; $2p$ — порядок B -сплайна, в 2 раза больший, чем p , причем коэффициент 2 указывает превышение порядка числителя над знаменателем.

Здесь $\lambda_n^p(t)$ представляют ортонормированный базис пространства G_n^p :

$$\lambda_n^p(t) = m_n^p(t) / \sqrt{u_n^{2p}},$$

где $m_n^p(t)$ — базисные функции в пространстве G_n^p .

Базисные функции $m_n^p(t)$ будем определять следующим образом:

$$m_n^p(t) = \frac{1}{N} \sum_k \bar{\chi}(n, k) M^p(t \ominus t_k),$$

где $\chi(n, k)$ — характеры группы H ; $\bar{\chi}(n, k)$ — значение, комплексно-сопряженное с $\chi(n, k)$.

Дискретное преобразование Фурье от базисного B -сплайна определим как

$$u_n^p = F_n(M^p) = \frac{1}{N} \sum_k \bar{\chi}(n, k) M^p(t_k),$$

где значение t_k определяется из формулы $t_k = (\frac{p}{2} + k) / N$.

Преобразования сигналов в базисах функций сплайн-характеров

Для пространств G_n^p (при $t = t_k$) введем пары прямых и обратных преобразований Фурье (ПФ) в базисах функций сплайн-характеров (БФСХ):

$$S^p(k) = \sum_n F_n(q^*) \lambda_n^p(k) = \sum_n F_n(z) \frac{\sqrt{u_n^{2p}}}{u_n^p} \lambda_n^p(k), \quad (2)$$

где $t_k = k$ — текущее значение дискретного временного отсчета; q^* — коэффициенты сплайна при использовании ортонормированного базиса.

Дискретное ПФ от коэффициентов сплайна при использовании ортонормированного базиса будем находить по формуле, которая по своей сути представляет обратное ПФ в базисе характеров

$$F_n(q^*) = \frac{1}{N} \sum_k \bar{\chi}(n, k) q_k^*$$

и сплайн-характеров

$$F_n(q^*) = \frac{1}{N} \sum_k \bar{\lambda}_n^p(k) z_k. \quad (3)$$

Отметим, что при $p = 1$, частными случаями пары выражений (2) и (3) являются выражения прямого и обратного преобразований в базисах функций-характеров, полученных в работе [1].

Метод быстрых преобразований сигналов в БФСХ

Согласно (2), для нахождения спектральных коэффициентов в базисе λ_n^p необходимо вычислить дискретное ПФ $F_n(q^*)$. Один из вариантов его вычисления — через $F_n(z)$. Для этого необходимо полученную последовательность $F_n(z)$ поэлементно

умножить на последовательность $\sqrt{u_n^{2p}} / u_n^p$. Элементы последней последовательности могут быть вычислены заранее, а для вычисления $F_n(z)$ существуют алгоритмы [1]. Следовательно, переход из базиса функций характеров (БФХ) в БФСХ на выборке длиной N увеличивает число операций преобразования на N . Это касается как ПФ в БФСХ, так и его быстрого алгоритма.

Таким образом, быстрое преобразование в БФСХ существует и его основа — алгоритм быстрого преобразования Фурье (БПФ) в БФХ. При этом вычислительная сложность БПФ в БФСХ на выборке длиной N увеличивается на N операций умножения по сравнению с алгоритмом БПФ в БФХ.

Сравнение вычислительных затрат классического алгоритма БПФ со всеми алгоритмами БПФ в БФСХ — задача невыполнимая по причине бесконечного числа последних. Поэтому остановимся на некоторых из них.

В частности, сравним вычислительные затраты, требуемые для реализации алгоритмов БПФ для частного случая БФСХ — базисов сплайн-Виленикина — Крестенсона (СВКФ). Для достижения этой цели воспользуемся подходом, предложенным в работе [2]. Так, если принять, что на операцию умножения и сложения тратится одинаковое время, то предельный выигрыш по скорости обработки ξ при использовании быстрого преобразования Уолша относительно дискретных экспоненциальных функций (ДЭФ) в алгоритме БПФ будет достигать $\xi = 5$. В то же время применение функций Виленикина — Крестенсона (ВКФ) по модулю 4 обеспечивает $\xi = 3,25$. В рассматриваемом случае для получения спектра использовался алгоритм БПФ в базисах СВКФ [3]. Согласно этому алгоритму на N входных точек преобразования дополнительно к стандартному объему вычислений необходимо добавить N операций умножения. Это приводит к увеличению объема вычислений по сравнению

с классическими алгоритмами быстрого преобразования Уолша и БПФ в базисе ВКФ по модулю 4. Однако получаемый выигрыш в этих случаях по сравнению с использованием классического алгоритма БПФ все равно оказывается существенным.

Данные об объеме вычислений для алгоритма БПФ [4] и результаты, полученные в работе [2], позволили получить оценку эффективности [5] (рис. 1), где ξ — выигрыш в количестве необходимых вычислительных операций, N — длина обрабатываемой выборки.

Анализ полученных результатов показал, что выигрыш в числе необходимых вычислительных процедур при переходе к базису СВКФ может достигать 2...3 раз. Таким образом, реализация операций ЦОС с использованием алгоритма БПФ в базисе СВКФ ведет к существенному сокращению вычислительных затрат. Аналогичные результаты можно получить и при рассмотрении алгоритма в базисе сплайн-Рейдера [1].

Алгоритм сплайн-БПФ в базисах функций сплайн-характеров

Довольно часто при выполнении стандартных операций ЦОС (фильтрации, вычислении корреляционных функций, определении несущей частоты) возникает необходимость осуществления интерполяции обрабатываемых дискретных данных о непрерывных функциях. Следовательно, возникает необходимость разработки вычислительно эффективного алгоритма интерполяции сигналов. В теории сплайн-гармонического анализа таким алгоритмом является алгоритм сплайн-БПФ в ДЭФ [6]. Его вычислительная эффективность базируется на том, что он построен на основе классического БПФ, поэтому по свойствам сравним с известным алгоритмом интерполяции, основанном на добавлении нулевых коэффициентов в спектральной области. Основное отличие состоит в том, что вместо добавления нулей в спектральной области используется информация о степени гладкости. Это позволяет осуществлять не линейную интерполяцию, как в классическом случае, а соответствующую выбранному порядку сплайна (например, кубическому). Рассматриваемый подход реализации быстрых преобразований позволяет разработать алгоритм сплайн-БПФ сигналов в БФСХ.

Рассмотрим особенности этого алгоритма. Так как в основе алгоритма сплайн-БПФ в базисе ДЭФ лежат процедуры БПФ, то разумно положить процедуры БПФ в БФСХ и в основу алгоритма сплайн-БПФ в БФСХ.

Исходными данными для алгоритма (рис. 2) являются: объем выборки N ; порядок сплайна p ; тип группы H и кольца K , модуль представления числа μ ; отсчеты сигнала $\{z_k\}$ и шаг сетки интерполяции h .

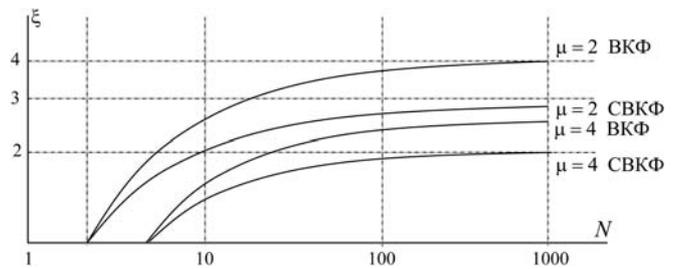


Рис. 1. Выигрыш в количестве операций вычисления преобразования Фурье при переходе от базиса ДЭФ к базисам ВКФ и СВКФ при $\mu = 2, 4$

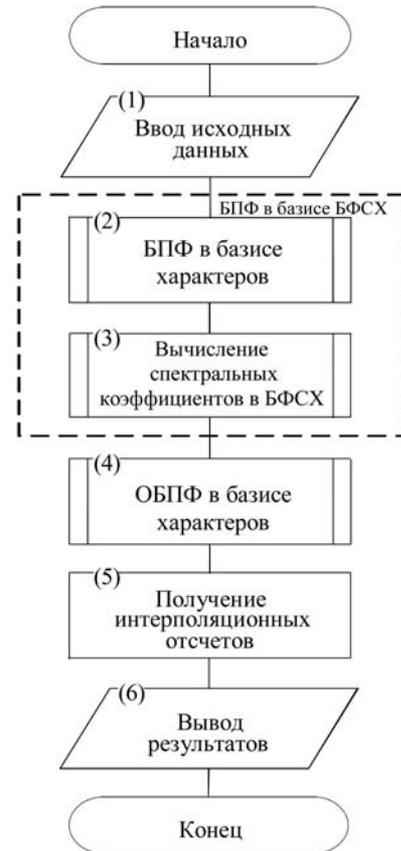


Рис. 2. Алгоритм сплайн-БПФ в БФСХ

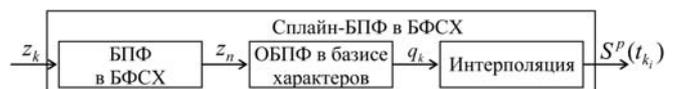


Рис. 3. Принцип реализации процедур интерполяции в БФСХ

Разработанный алгоритм отличается от известного не только переходом от ДЭФ к БФСХ, но и отказом от увеличения числа спектральных коэффициентов перед осуществлением операции обратного БПФ (ОБПФ) в соответствующем базисе, поскольку получение интерполяционных значений происходит уже во временной области. Структурная схема, реализующая указанные процедуры, представлена на рис. 3.

Таблица 1

Число операций на интерполяцию различными методами в базисах СВКФ (квадратичный сплайн)

N	Модуль числа	Алгоритм интерполяции	Всего операций	Значение выигрыша
64	ДЭФ	Добавление нулей	4992	—
64	ДЭФ	Сплайн-БПФ	2944	1,69
64	2	Сплайн-БПФ	2496	2
64	4	Сплайн-БПФ	2688	1,85
512	ДЭФ	Добавление нулей	53 760	—
512	ДЭФ	Сплайн-БПФ	26 624	2,01
512	2	Сплайн-БПФ	19 776	2,71
512	4	Сплайн-БПФ	21 658	2,48

В табл. 1 представлены сравнительные затраты на интерполяцию при использовании классического, в котором для интерполяции функций в спектральной области добавляются нулевые коэффициенты, и предлагаемого методов на примере частного случая БФСХ — СВКФ. Здесь индекс i при пара-

метре k отсчета t_k указывает на интерполяцию между отсчетами t_k и t_{k+1} . Величина выигрыша в вычислительных операциях рассматривается по отношению к классическому методу (см. строки 1 и 5 табл. 1).

Анализ полученных результатов показывает, что выигрыш в объеме вычислений может достигать от 1,69 до 2,71 раза даже на относительно коротких выборках сигнала. Таким образом, предложенный алгоритм обладает вычислительной эффективностью и может быть использован для осуществления фильтрации, вычисления корреляционных функций, определения несущей частоты.

Метод оценки несущей частоты в БФСХ

Большую роль в ходе осуществления радиоприема играет процедура оценки несущей частоты. Для классических моделей сигналов, синтезированных в базисе ДЭФ, для ее получения используют ПФ. Однако для сигналов, сформированных в БФСХ, применение базиса ДЭФ не позволяет получить

достоверную оценку значения несущей частоты. Естественно, что данную операцию необходимо выполнять в БФСХ.

Так, на рис. 4 представлены эпюры, поясняющие принцип оценки несущей частоты в БФСХ.

Аналоговый сигнал $z(t)$ (рис. 4, а) дискретизируют $z(t_k)$ (рис. 4, б). Затем вычисляют последовательность комплексных спектральных коэффициентов $F_n(z)$ (рис. 4, в) методом преобразования в выбранном базисе характеристик. Одновременно путем ПФ В-сплайн заданной степени $p - 1$ преобразуется в последовательность комплексных дискретных отсчетов u_n^p , представленных на рис. 4, г. Порядок В-сплайна определяется пользователем и зависит от степени гладкости анализируемого сигнала. Затем последовательность комплексных спектральных коэффициентов $F_n(z)$ умножают поэлементно на последовательность комплексных дискретных отсчетов $\sqrt{u_n^{2p}}/u_n^p$ для ортонормированных базисов В-сплайнов $\tilde{\chi}_n^p(t)$ (рис. 4, д). Далее вычисляют компоненты спектральной плотности мощности $\tilde{S}(n)$ в БФСХ заданной степени $p - 1$ с помощью выражения, представлен-

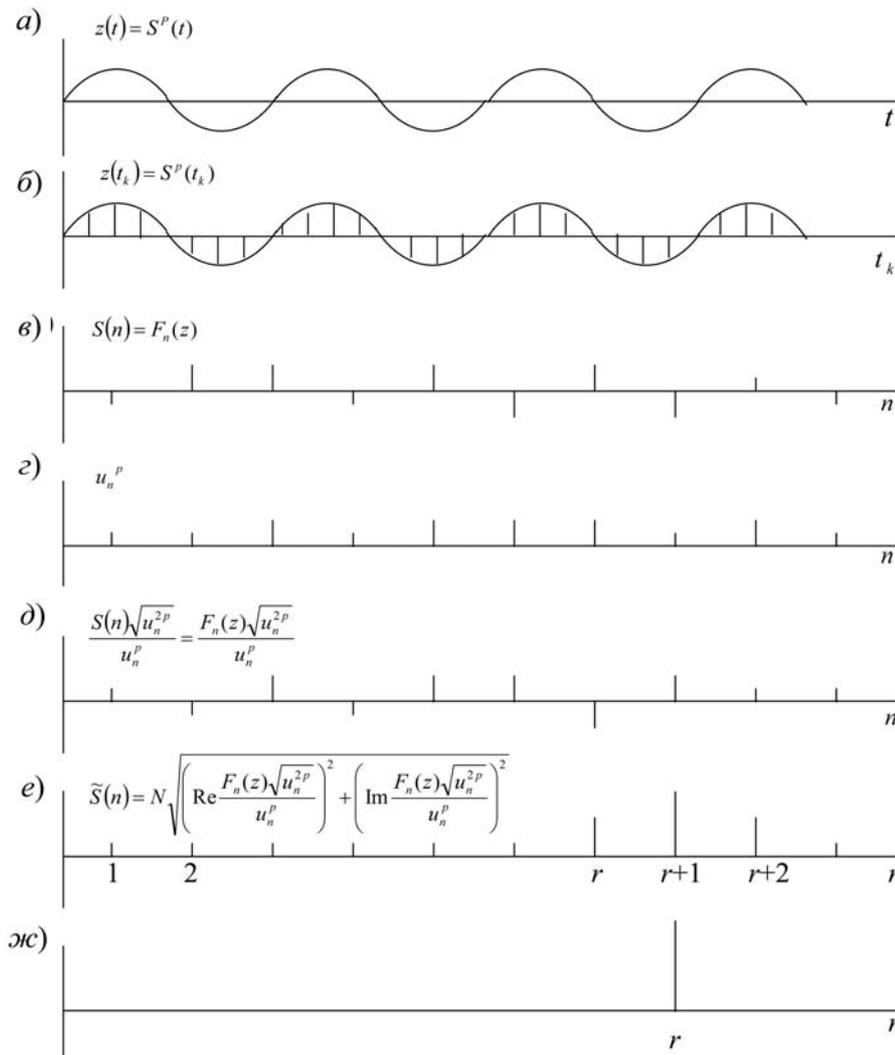


Рис. 4. Принцип реализации процедуры оценки несущей частоты в БФСХ

Число операций на интерполяцию при определении несущей частоты различными методами в базисах СВКФ (кубический сплайн)

N	Модуль числа	Алгоритм интерполяции	Число операций на интерполяцию			Всего операций	Выигрыш
			$\lambda \times 5 \times N/2$	$\lambda \times 5 \times N$	$\lambda \times 5 \times 2$		
64	ДЭФ	Нулями	—	—	—	13 824	—
64	ДЭФ	Сплайн-БПФ	1280	—	—	2240	6,17
64	ДЭФ	Сплайн-БПФ	—	—	80	1040	13,26
64	2	Сплайн-БПФ	—	2560	—	3082	4,48
64	2	Сплайн-БПФ	—	—	80	602	22,96
64	4	Сплайн-БПФ	1280	—	—	1896	7,29
64	4	Сплайн-БПФ	—	—	80	696	19,86
128	ДЭФ	Нулями	—	—	—	30 720	—
128	ДЭФ	Сплайн-БПФ	2560	—	—	4736	6,48
128	ДЭФ	Сплайн-БПФ	—	—	80	2256	13,61
128	2	Сплайн-БПФ	—	5120	—	6194	4,95
128	2	Сплайн-БПФ	—	—	80	1154	26,62
128	4	Сплайн-БПФ	2560	—	—	3846	7,98
128	4	Сплайн-БПФ	—	—	80	1366	22,48

Примечание. Для указанных алгоритмов число точек интерполяции $\lambda = 8$.

ного на рис. 4, *e*, где N — объем выборки. На новом массиве компонент спектральной плотности мощности находится максимум r (рис. 4, *ж*), а значение несущей частоты сигнала определяется по формуле $f_n = r\Delta f$, где Δf — расстояние между спектральными компонентами $\lambda_n^p(t)$.

Оценим вычислительные затраты рассмотренного подхода. Его основу составляют процедуры БПФ в БФСХ. Сокращение объема вычислений данной процедуры возможно за счет формирования сигналов в базисах сплайн-Рейдера, в СВКФ с использованием модуля 2...4. В этом случае вычислительные затраты сократятся в 2...3 раза.

Методы оценки несущей частоты сигналов с использованием алгоритма сплайн-БПФ в БФСХ

Дальнейшее повышение вычислительной эффективности и точности оценки возможно за счет учета информации о гладкости спектральной плотности мощности сигналов. Поскольку точность оценки определяется частотным разрешением, то необходимо повысить разрешающую способность. При классическом подходе это достигается за счет увеличения длины реализации сигнала, в том числе и путем добавления нулевых значений. При этом существенно растет объем вычислений, а интерполяция осуществляется по линейному закону, без использования информации о степени гладкости спектральной плотности мощности сигнала. В связи с этим предлагается использовать сплайн-интерполяцию в частотной области, и осуществлять ее с помощью вычислительно эффективного алгоритма сплайн-БПФ в БФСХ. Возможны два варианта интерполяции [7].

Таблица 3

Интерполяционные и сглаживающие свойства глобальных сплайнов

№ п/п	Метод интерполяции	Потенциальная точность интерполяции	Требования к функции гладкости p	Выигрыш в помехоустойчивости, дБ
1	Добавление нулей	$1/2h$	C^1	—
2	Квадратичная интерполяция	Не хуже $o(h^4 z_{\max}^{(4)})$	C^4	—
	Квадратичное сглаживание	$<o(h^4 z_{\max}^{(4)})$	C^4	0,3
3	Кубическая интерполяция	$<o(h^5 z_{\max}^{(5)})$ $5/384(h^4 z_{\max}^{(4)})$	C^5	—
	Кубический сглаживающий сплайн	$<o(h^5 z_{\max}^{(5)})$ $5/384(h^4 z_{\max}^{(4)})$	C^5	0,5
4	Интерполяционный сплайн 4-й степени	$<o(h^6 z_{\max}^{(6)})$	C^6	—
	Сглаживающий сплайн 4-й степени	$<o(h^6 z_{\max}^{(6)})$	C^6	0,9
5	Интерполяционный сплайн 5-й степени	$<o(h^7 z_{\max}^{(7)})$	C^7	—
	Сглаживающий сплайн 5-й степени	$<o(h^7 z_{\max}^{(7)})$	C^7	1,1

Примечание. Для указанных алгоритмов число точек интерполяции $\lambda = 8$.

h — интервал интерполяции; $z_{\max}^{(l)}$ — максимальные значения производной l -го порядка сигнала z ; C^p — пространство функций заданной степени гладкости; $o(*)$ — оценка точности остатка интерполяции функции.

Число операций на интерполяцию при определении несущей частоты различными методами с использованием локальных сплайнов для $N = 64$

Модуль числа	Алгоритм интерполяции	Число операций на интерполяцию				Всего операций	Выигрыш
		Одной точки	$\lambda \times 5 \times N/2$	$\lambda \times 5 \times N$	$\lambda \times 5 \times 2$		
ДЭФ	Добавление нулей	—	—	—	—	13 824	—
ДЭФ	КВСМШ	21	5376	—	—	5920	2,33
ДЭФ	КВСМШ	21	—	—	336	880	15,7
ДЭФ	КСМШ	28	7168	—	—	7712	1,79
ДЭФ	КСМШ	28	—	—	448	992	13,93
ДЭФ	СМШ 4-й степени	80	20 480	—	—	21 024	0,66
ДЭФ	СМШ 4-й степени	80	—	—	1280	1824	7,58
ДЭФ	СМШ 5-й степени	102	26 112	—	—	26 656	0,52
ДЭФ	СМШ 5-й степени	102	—	—	1632	2176	6,35
ДЭФ	КИС 5-й степени	175	44 800	—	—	45 344	0,3
2	КИС 5-й степени	175	—	—	2800	3344	4,13
ДЭФ	СМС 5-й степени	175	44 800	—	—	45 344	0,3
ДЭФ	СМС 5-й степени	175	—	—	2800	3344	4,13

Примечание. Для указанных алгоритмов число точек интерполяции $\lambda = 8$.

В первом случае используются коэффициенты всей реализации, во втором — интерполяция проводится только вблизи исходного максимального спектрального коэффициента.

Эффективность указанных методов оценивалась с позиций скорости процедур ЦОС (табл. 2). Характеристика методов по точности определения несущей частоты сигналов представлена в табл. 3.

Согласно табл. 3, чем выше степень гладкости интерполируемого процесса p , соответственно и степень гладкости сплайна, тем выше σ^* — точность интерполяции [8]. Так, например, если $h = 0,1$, то точность кубической интерполяции с использованием глобальных сплайнов будет пропорциональна значению 0,0001. В то же время при классическом подходе точность интерполяции пропорциональна интервалу интерполяции $h = 0,05$, что равносильно увеличению требуемого объема производимых вычислений.

Важно отметить, что использование глобальных сглаживающих сплайнов может повысить помехоустойчивость определения несущей частоты. Согласно табл. 3, при выборе кубического сплайна выигрыш в помехоустойчивости может достигать 0,5 дБ.

Оценка несущей частоты с использованием локальных сплайнов

Основное отличие локальных сплайнов от глобальных состоит в том, что для интерполяции используется значительно меньший объем информации. Например, для поиска экстремума достаточно данных лишь о вблизи расположенных отсчетах сигнала. Это относится и к простейшим сплайнам, таким как сплайны минимального шаблона (СМШ) квадратичные (КВСМШ) и кубические (КСМШ), квазиинтерполяционные сплайны (КИС), сплайны максимального сглаживания (СМС) [9].

Таблица 5

Интерполяционные и сглаживающие свойства локальных сплайнов для различных методов оценивания несущей частоты

№ п/п	Метод интерполяции	Потенциальная точность интерполяции	Требование к функции	Выигрыш в помехоустойчивости, дБ
1	Добавление нулей	$1/(2h)$	C^1	—
2	КВСМШ	$0,047h^3 z_{\max}^{(3)}$	C^4	—
	Квадратичный КИС	Менее $0,047h^3 z_{\max}^{(3)}$	C^4	—
	Квадратичный СМС	$0,047h^3 z_{\max}^{(3)}$	C^4	0,1
3	КСМШ	$35/1152h^4 z_{\max}^{(4)}$	C^5	—
	Кубический КИС	Менее $35/1152h^4 z_{\max}^{(4)}$	C^5	—
	Кубический СМС	$35/1152h^4 z_{\max}^{(4)}$	C^5	0,22
4	СМШ 4-й степени	$13\ 2677/13\ 271\ 040 \times h^5 z_{\max}^{(5)}$	C^5	—
	КИС 4-й степени	Менее $132\ 677/13\ 271\ 040 \times h^5 z_{\max}^{(5)}$	C^6	—
	СМС 4-й степени	$132\ 677/13\ 271\ 040 \times h^5 z_{\max}^{(5)}$	C^6	0,34
5	СМШ 5-й степени	$59/5120h^6 z_{\max}^{(6)}$	C^7	—
	КИС 5-й степени	Менее $59/5120h^6 z_{\max}^{(6)}$	C^7	—
	СМС 5-й степени	$59/5120h^6 z_{\max}^{(6)}$	C^7	0,5

Число операций на вычисление корреляционных функций в базисах СВКФ

№	Мо- дуль числа	БПФ в базисе ВКФ (операций)	Вычисле- ние спек- тральных коэффи- циентов	Вычис- ление спектра	ОБПФ в базисе ВКФ от N точек	ОБПФ в базисе ВКФ от $N \cdot \lambda$ точек	Число операций на интерполяцию		Всего операций	Выигрыш
							на $\lambda \times 4 \times N$ точках	на $\lambda \times 4 \times 2$ точках		
64	ДЭФ	384	—	384	—	4608	—	—	5376	—
64	2	192	64	64	192	—	2048	64	2560/576	2,1/9,33
64	4	256	64	384	256	—	2048	64	3008/1024	1,78/5,25
128	ДЭФ	896	—	768	—	10 240	—	—	11 904	—
128	2	407	128	128	407	—	4096	64	4806/1134	2,48/10,5
128	4	535	128	768	535	—	4096	64	6062/2030	1,96/5,86
256	ДЭФ	1024	—	1536	—	22 528	—	—	25 088	—
256	2	410	256	256	410	—	8192	64	9124/996	2,75/25,19
256	4	552	256	1536	552	—	8192	64	11 088/2960	2,26/8,48
512	ДЭФ	4608	—	3072	—	49 152	—	—	56 832	—
512	2	1440	512	512	1440	—	16384	64	20 288/3968	2,8/14,32
512	4	2355	512	3072	2355	—	16384	64	24 678/8358	2,3/6,8

Примечание. Для указанных алгоритмов число точек интерполяции $\lambda = 8$.

Однако обеспечиваемая ими точность и помехоустойчивость ниже по сравнению с глобальными сплайнами, а вычислительная эффективность существенно изменяется в зависимости от длины реализации.

Представление о вычислительной эффективности простейших сплайнов на примере СМШ с использованием экспоненциальных функций и СВКФ с модулем 2 и 4 дает табл. 4.

Характеристика точности определения несущей частоты сигналов представлена в табл. 5.

Анализ полученных результатов показал, что при выборе наиболее часто используемого кубического СМС выигрыш в помехоустойчивости может достигать 0,22 дБ.

Метод вычисления корреляционных функций сигналов

Предлагаемый метод базируется на свойствах сплайнов и предложенного алгоритма СБПФ. Его вычислительная эффективность обеспечивается при условии расчета корреляционной функции при предельно низкой частоте дискретизации. Вариант структурной схемы, реализующий метод в соответствии с указанными требованиями, представлен на рис. 5.

Сравнительные затраты на вычисление корреляционных функций при использовании различных базисов представлены в табл. 6.

В табл. 6 в столбце "Всего операций" через дробь указано число операций на интерполяции на $\lambda \times 4 \times N$ точках и число операций на $\lambda \times 4 \times 2$ точках.

При этом интерполяция на $\lambda \times 4 \times N$ точках проводится для получения дополнительных значений корреляционной функции на всей ее области определения, а интерполяция на $\lambda \times 4 \times 2$ точках — только в районе максимума. Здесь λ — число точек интерполяции между двумя узловыми точками, $N(2)$ — число интервалов интерполяции, 4 — число операций для получения одного интерполируемого значения. Полученные результаты подтверждают вычислительную эффективность предложенного алгоритма.

Метод фильтрации сигналов в базисах функций сплайн-характеров

Вариант структурной схемы метода фильтрации сигналов в базисах функций сплайн-характеров представлен на рис. 6.

На рис. 6 $L^P(t \Theta t_k)$ — импульсная характеристика фильтра в БФСХ. Результаты оценки вычислительной эффективности метода фильтрации представлены в табл. 7.

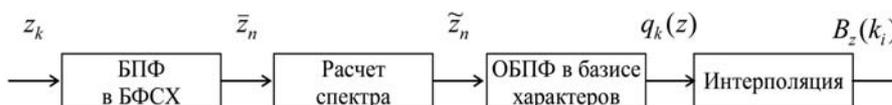


Рис. 5. Вычисление автокорреляционной функции в БФСХ

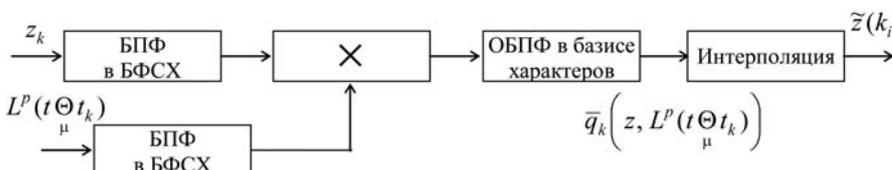


Рис. 6. Фильтрация в базисах СВКФ

Число операций на фильтрацию в различных базисах (без интерполяции)

<i>N</i>	Модуль числа	БПФ в базисе ВКФ (операций)	Вычисление спектральных коэффициентов сигнала, прошедшего через фильтр	ОБПФ в базисе ВКФ <i>N</i> точек	Всего операций	Выигрыш
64	ДЭФ	384	64	384	832	—
64	2	192	64	192	448	1,86
64	4	256	64	256	575	1,45
512	ДЭФ	4608	512	4608	9728	—
512	2	1440	512	1440	3392	2,86
512	4	2355	512	2355	5222	1,86

В основе разработанного метода лежит использование алгоритма сплайн-БПФ в БФСХ. Анализ полученных результатов показывает, что даже без интерполяции выигрыш в объеме вычислений достигает от 1,45 до 2,86 раза на коротких выборках сигнала. Данный факт подтверждает вычислительную эффективность разработанного метода.

Заключение

Разработана аналитическая основа методов оценки спектральных и временных параметров сигналов в БФСХ.

Предложен метод быстрых преобразований сигналов в БФСХ, обладающий высокой вычислительной эффективностью. Выигрыш в объеме вычислений при переходе от ДЭФ к частному случаю БФСХ — базису функций сплайн-Виленкина — Крестенсона достигает 2...3 раз.

На основе разработанного алгоритма сплайн-БПФ в БФСХ предложены методы оценки несущей частоты сигналов с использованием глобальных сплайнов. Вычислительная эффективность указанных методов для СВКФ достигает 2,54 раза и более, даже на коротких выборках сигнала. А выигрыш в помехоустойчивости для сглаживающих глобальных сплайнов может достигать 1 дБ.

Использование методов оценки несущей частоты сигналов на основе локальных сплайнов может обеспечить выигрыш в помехоустойчивости до 0,5 дБ при достаточно высокой вычислительной эффективности.

При расчете корреляционных функций сигналов вычислительные затраты, по сравнению с класси-

ческим подходом для СВКФ, как частного случая БФСХ, сокращаются в 2,1 и более раза.

Выигрыш в объеме вычислений при фильтрации сигналов достигает от 1,45 до 2,86 раза даже на коротких выборках сигнала по отношению к классическому подходу.

Таким образом, применение разработанного аналитического аппарата обработки сигналов обеспечивает повышение помехоустойчивости при общем снижении вычислительных затрат.

Список литературы

1. **Вариченко Л. В., Лабунец В. Г., Раков М. А.** Абстрактные алгебраические системы и цифровая обработка сигналов // Киев: Наукова Думка, 1986. 247 с.
2. **Трахтман А. М., Трахтман В. А.** Основы теории дискретных сигналов на конечных интервалах. М.: Советское радио, 1975. 208 с.
3. **Агиевич С. Н.** Сплайн-Виленкина — Крестенсона функции в представлении сигналов // Научное приборостроение. 2002. Том 12, № 1. С. 79—89.
4. **Рабинер Л., Голд Б.** Теория и применение цифровой обработки сигналов / Пер. с англ. М.: Мир, 1978. — 848 с.
5. **Агиевич С. Н., Беспалов В. Л.** Цифровая обработка сигнала: скорость и экономия ресурса // Мобильные системы. 2007. № 3. С. 26—29.
6. **Желудев В. А.** Периодические сплайны и быстрое преобразование Фурье // Журнал вычислительной математики и математической физики. 1992. Т. 32, № 2. С. 179—198.
7. **Агиевич С. Н., Малышев С. Р., Подымов В. А.** и др. Способ (варианты) и устройство (варианты) оценивания несущей частоты. Патент № 2168759 RU, МПК6 G 06 F 17/14, G 01 R 23/ 00, № 99126680/09; Заявл. 16.12.99; Опубл. 10.06.01, Бюл. № 16. 132 с.
8. **Завьялов Ю. С., Квасов Б. И., Мирошниченко В. Л.** Методы сплайн-функций. М.: Наука, 1980. 352 с.
9. **Желудев В. А.** Локальные сплайны с регулирующим параметром // Журнал вычислительной математики и математической физики. 1991. Т. 31. № 2. С. 193—211.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ФИНАНСОВО-ЭКОНОМИЧЕСКИХ СИСТЕМАХ

УДК 004.891

А. А. Рындин, д-р техн. наук, проф.,
В. П. Ульев, соискатель,
e-mail: u_vitalii@mail.ru,
Воронежский государственный
технический университет

Проектирование гибридной системы скоринговой оценки кредитоспособности юридических лиц

Рассматривается задача проектирования гибридной системы скоринговой оценки кредитоспособности юридических лиц. Рассмотрены классификация гибридных систем и проблематика их проектирования, предложена компенсационная модель оценки интеграции различных методик. Предложен подход построения гибридной нечеткой нейрогенетической экспертной системы принятия решений. Приводится общая блок-схема алгоритма гибридной системы, рассмотрены алгоритм основного решающего блока и принципы взаимодействия между подсистемами.

Ключевые слова: кредитный скоринг, гибридные системы, нечеткие множества, нейронные сети, генетические алгоритмы

Принятие решения о кредитовании потенциального заемщика основывается, как правило, на оценке такого показателя заемщика, как его кредитоспособность. Этот показатель определяет способность потенциального заемщика полностью и в срок рассчитаться по долговым обязательствам и является важнейшим фактором в системах принятия решений по кредитованию. Основная задача скоринговой системы оценки кредитоспособности юридических лиц состоит в определении оценки, или балла, кредитоспособности, позволяющей провести его категорирование и выработать решение о кредитовании. Оценка кредитоспособности является нелинейной функцией от ряда параметров. Под параметрами функции кредитоспособности следует понимать набор финансовых и иных показателей, характеризующих деятельность организации. Виды и число показателей или факторов в целом определяют скоринговую модель. Таким образом, задача

скоринговой системы сводится к построению нелинейной функции кредитоспособности на заданном интервале времени и определению ее значения для заданных значений выбранных показателей кредитоспособности потенциального заемщика.

Проблематика выбора скоринговой модели заключается в определении оптимального набора показателей, имеющих наибольшее влияние в заданных условиях на итоговую оценку. Каждый из показателей может оказывать соответственно сильное или слабое влияние либо не оказывать его вовсе. Использование же избыточного набора показателей в целом усложняет процесс анализа, непосредственно приводит к увеличению времени анализа и вносит определенную степень ошибки. Таким образом, целевая функция определения лучшей скоринговой модели определяется следующим критерием: минимальное число используемых показателей, выбранных исходя из их наибольшей значимости, которые определяют минимальную погрешность оценки относительно эталонного результата.

С математической точки зрения скоринг есть процедура определения нелинейной функции и вычисления ее значений при заданных исходных данных. В настоящее время известно множество методик исследования вариации зависимой переменной от набора независимых. В целом, следует отметить, что у каждого из методов исследования вариации результата от исходных данных имеются в определенной степени как положительные качества, так и отрицательные. Поэтому проектирование гибридной скоринговой системы продиктовано перспективой получения более качественной скоринговой оценки за счет взаимной компенсации отрицательных и положительных свойств применяемых методик.

Рассмотрим основные проблемы проектирования гибридных систем:

- выбор интегрируемых методик;
- порядок взаимодействия разнородных подсистем;
- формирование итогового решения гибридной системы посредством объединения результатов составных разнородных подсистем.

Решение проблемы выбора интегрируемых методик лежит в плоскости определения целей и задач, поставленных перед гибридной системой, а также в контексте фундаментального подхода компенсации недостатков одних методов за счет преимуществ других. Рассмотрим основные задачи, предъявляе-

мые скоринговой системе оценки кредитоспособности юридических лиц:

- расчет скоринговой оценки кредитоспособности юридических лиц;
- построение и обработка различных скоринговых моделей оценки кредитоспособности;
- обеспечение самоконтроля корректности результирующей оценки;
- обучение на основе исторических данных;
- ввод и учет экспертной оценки кредитоспособности, а также общей политики условий кредитования;
- обеспечение самостоятельного эволюционирования в целях более качественной оценки кредитоспособности и выявления слабых связей между финансовыми показателями.

Исходя из поставленных задач и возможностей их реализации предлагается спроектировать гибридный математический аппарат, сочетающий в себе такие области изучения искусственного интеллекта, как нейронные сети, теория нечетких множеств и генетические алгоритмы. Нейронные сети способны к обучению и решают задачу категорирования данных, генетические алгоритмы применимы в рамках изучения возможности развития системы, а механизм нечетких множеств позволяет получить альтернативную оценку кредитоспособности с учетом экспертного участия и в целом возможности влияния на итоговый результат. При использовании совокупности разных научных подходов в решении задачи появляется возможность перекрытия слабых мест одних подходов за счет сильных сторон других. Так, например, проблему построения и развития эффективной топологии нейронной сети можно возложить на механизм генетических алгоритмов. Результатом работы блока нечеткой логики можно оценивать корректность выхода нейросетевого блока, тем самым обеспечивая возможность экспертного участия в процессе и определения глобальной политики оценки кредитоспособности.

Согласно наиболее общей классификации гибридных систем по уровню интеграции можно выделить следующие типы систем:

- автономные;
- трансформационные;
- слабосвязанные;
- сильносвязанные;
- полностью интегрированные.

Автономные системы состоят из независимых программных компонентов, обрабатывающих один набор данных разными методами. Главным недостатком данных гибридных систем является необходимость модификации всех компонентов системы в случае изменения исходных данных. Трансформационные гибридные системы по существу являются автономными, с тем отличием, что используемый компонент фактически подбирает наилучший метод обработки данных по заданному критерию. Слабосвязанные системы характеризуются

набором отдельных компонентов, взаимосвязь между которыми осуществлена off-line (например, через файлы данных). Сильносвязанные гибридные системы отличаются on-line взаимодействием между компонентами и, исходя из своего названия, имеют достаточно быструю природу межкомпонентного взаимодействия. Полностью интегрированные системы совместно используют общие структуры данных и представлений знаний, а взаимосвязь между компонентами достигается посредством двойственной природы их структур.

При проектировании нечеткого нейрогенетического аппарата воспользуемся методикой интеграции, применяемой в сильносвязанных системах, известной как "доска объявлений". Данный подход проектирования интересен тем, что позволяет использовать различные компоненты (процессоры), совместно решающие одну задачу и использующие "доску объявлений" как рабочее место для выработки решения. При этом использование в системе каждого компонента влияет на качество результата, а не на его наличие в целом. При данном подходе обеспечивается модульность системы и независимость ее отдельных компонентов.

Далее рассмотрим упрощенную компенсационную модель предложенных методик.

1. *Нейронные сети:*

— недостатки: отсутствие оперативной гибкости результирующей оценки обусловлено актуальностью эталонной статистической выборки; качество вывода может быть оценено косвенно испытанием на эталонной выборке исходных данных; нейронная сеть не способна к самостоятельному развитию; проблема выбора эффективной топологии нейросети;

— компенсация: оперативная гибкость может быть улучшена за счет использования методики нечетких множеств, реализующих текущую экспертную субъективную модель; эволюция нейросети, поиск оптимальной топологии реализуется методикой генетических алгоритмов.

2. *Нечеткие множества:*

— недостатки: нечеткие правила носят субъективный характер, существует проблема формирования базы нечетких знаний ввиду необходимости учета всех возможных факторов;

— компенсация: субъективность метода компенсируется объективностью статистических методов, в частности, технологией нейронных сетей; мультивариантность нечеткой базы компенсируется технологией генетических алгоритмов, способных к поиску и выделению наиболее значимых правил.

3. *Генетические алгоритмы:*

— недостатки: путь эволюции может быть непрерывным, ветвь развития может оказаться тупиковой, функция приспособленности — не актуальной;

— компенсация: актуальность функции принадлежности может быть поддержана методикой нечетких множеств.

Таким образом, все компенсационные схемы интеграции методик в единой системе можно категорировать следующим образом:

- линейная (каждая из применяемых методик компенсирует недостатки последующей);
- радиально-линейная (вспомогательные методики компенсируют недостатки основной);
- замкнутые (методики взаимно компенсируют друг друга в той или иной степени).

Модель скоринговой нечеткой нейрогенетической системы (далее ННГ-системы) предполагает использование указанных выше методик и реализует следующую компенсационную схему:

- основной решающий модуль основан на технологии нейронных сетей и осуществляет обработку данных на основе имеющейся статистики;
- дополнительный модуль экспертной оценки, основанный на теории нечетких множеств, вносит субъективный локальный (региональный, отраслевой) момент смещения вектора результирующей оценки;
- дополнительный модуль эволюции системы, основанный на технологии генетических алгоритмов (ГА), выполняет поиск оптимальной топологии нейросети;
- в результате оценки варианта топологии нейросети (НС) подсистемой настройки основного решающего модуля модуль эволюции получает оперативные сведения, необходимые для дальнейшей селекции.

Предложенная компенсационная схема в большей степени является радиально-линейной, где за основу решающего блока выбрана нейросетевая технология.

На рис. 1 представлена схема скоринговой ННГ-системы.

Для реализации блока оценки кредитоспособности на основе экспертных правил обратимся к теории нечетких множеств. С учетом того, что степень принадлежности определена на отрезке $[0, 1]$, аппарат нечеткой логики, фактически, предлагает механизм приведения различных по сути единиц измерений к единой области определения. Нечеткая база знаний представляет совокупность нечетких правил. Результат обработки нечеткой базы является нечетким логическим выводом (fuzzy logic inference), представляющим аппроксимацию зависимости $y = f(x_1, x_2, \dots, x_n)$ результата от набора критериев. Так, для каждого потенциального заемщика набор показателей кредитоспособности, выраженный в виде функций принадлежности к нечетким множествам, обрабатывается на не-

четкой базе. Каждое нечеткое правило определяет совокупность необходимых операций, таких как конъюнкция и дизъюнкция, над показателями кредитоспособности, принятыми в скоринговой модели. Для представления правил используется операция импликации, для которой предложены различные способы нечеткой реализации. В случае если нечеткие правила сформированы логично и не находятся во взаимном противоречии, то результатом импликации каждого правила является результирующий терм удовлетворенности альтернативы на единичном интервале. Следующим и заключительным шагом является сопоставление альтернатив на основе точечных оценок. Лучшей считается альтернатива с наибольшей точечной оценкой. Определение пороговых значений точечных оценок проводится также экспертным путем.

Технология нейронных сетей позволяет решать задачи по категорированию исследуемых объектов, используя статистическую информацию. Тем самым, нейросетевой блок формирует объективную оценку на основе статистической информации. Используя статистику по финансовым показателям и оценкам кредитоспособности юридических лиц, обученная нейросеть с определенной степенью точности способна прогнозировать кредитоспособность исследуемого заемщика. Достоверность вывода нейросетевого блока оценивается в результате выполнения следующих условий:

- корректное завершение обучения нейросети, т. е. переход системы в установившийся режим, когда весовые коэффициенты перестают изменяться в процессе обучения или квадратичная ошибка достигла приемлемого уровня;
- соответствие оценки кредитоспособности на контрольной выборке ожидаемым результатам;
- соответствие оценки результатам работы экспертного блока в ожидаемых пределах.



Рис. 1. Схема скоринговой ННГ-системы

Рассмотрим основные задачи, возлагаемые на нейросетевой блок:

- оценка кредитоспособности потенциального заемщика на основе статистической информации;
- настройка и тестирование новой топологии нейросети и проверка ее качества.

Исходя из общей компоновочной схемы ННГ-системы, можно считать, что нейросетевой модуль является основным решающим блоком. Способность системы к процессу эволюции характеризуется способностью к саморазвитию ее составных элементов. Тем самым, эволюция нейросетевого блока, основного решающего элемента системы, позволяет в полной мере выделить это свойство у всей скоринговой системы в целом. Учитывая основные задачи нейросетевого модуля, очень важно разделить основные процессы: рабочий процесс выработки скоринговой оценки и внутренний процесс настройки нейросети. Рассмотрим каждый процесс в отдельности.

Процесс определения скоринговой оценки. Выработка решения по скоринговой оценке происходит в результате работы настроенной и обученной нейросети принятой топологии. На вход в решающую нейросеть подается массив входных данных: финансовые и статистические показатели потенциального заемщика, на выходе формируется скоринговая оценка.

Процесс настройки нейросети. Основной задачей процесса является настройка нейронной сети, обеспечивающей ее наиболее качественную реакцию. Данный процесс предполагает решение таких задач, как обучение нейронной сети, проверка качества реакции на контрольной выборке, определение наиболее удачного варианта нейросети. При этом оба процесса в рамках нейросетевого модуля должны функционировать независимо друг от друга, разделяя бизнес и сервисные функции.

Для решения поставленных задач наиболее очевидным представляется реализация описанных основных процессов в виде независимых блоков, представленных на рис. 2.

Таким образом, в рамках нейросетевого модуля скоринговой системы в режиме реального времени должны функционировать две нейронные сети: одна нейросеть находится в режиме настройки и оценки качества реакции, вторая в промышленном режиме определяет скоринговую оценку кредитоспособности потенциального заемщика. При этом блоку развития нейросети отводится важная роль взаимодействия с генетическим модулем эволюции нейросети. Тем самым, инициализация блока развития должна происходить при следующих событиях:

- первый старт скоринговой системы (настройка базовой топологии нейросети);
- изменение обучающей выборки;
- изменение базовой топологии нейросети (при поступлении новой топологии от модуля эволюции).

Настройка нейронной сети промышленного блока проводится путем загрузки варианта топологии и настройки весовых коэффициентов, переданного от блока развития и признанного им наиболее оптимальным.

Генетический модуль скоринговой системы реализует важнейший сервисный процесс — эволюцию инструмента определения скоринговой оценки кредитоспособности потенциальных заемщиков. Наличие данного модуля в составе системы позволяет ей развиваться и более качественно решать поставленную задачу.

Процесс развития нейросетевого модуля нацелен главным образом на решение задачи поиска оптимальной топологии нейронной сети, обеспечивающей максимально точную оценку за минимальное время. Учитывая комплексность задачи, процесс развития следует рассматривать в тесном взаимодействии с описанным выше сервисным процессом обучения нейросети. Моделирование новой топологии нейросети предполагает решение таких задач, как определение числа слоев, числа нейронов в каждом из них, вида и числа межнейронных связей, применение той или иной функции активации нейрона, формирование блочных структур нейронов в рамках слоя. Результатом совместного функционирования процесса эволюции и обучения является определение оптимальной топологии и настройки нейронной сети.

Далее рассмотрим взаимодействие подсистем гибридной скоринговой системы. Взаимодействие аппарата генетических алгоритмов с блоком настройки нейросетевого модуля двустороннее: эволюционный блок вырабатывает решение и предлагает его нейросетевому блоку, в ответ получает некоторую оценку предложенному решению, необходимую в процедуре оценки приспособленности.

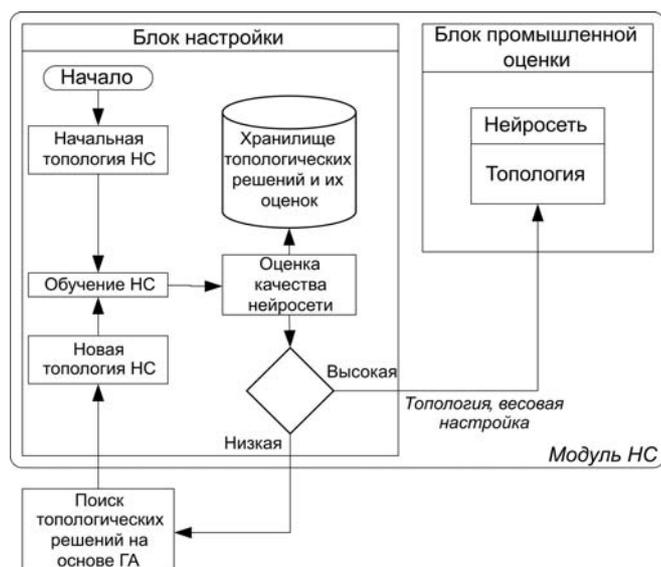


Рис. 2. Схема реализации нейросетевого модуля ННГ-системы

Взаимодействие экспертного блока и основного решающего сведено к сравнению итоговой оценки кредитоспособности. Для каждой оценки можно определить рамки или область определения и рассчитать среднее взвешенное значение или так называемый центр тяжести. Соответственно, степень доверия снижается на границах области определения категорий кредитоспособности. Итоговая оценка может быть рассчитана как средняя величина или взята исходя из коэффициента доверия. Результат принятия решения прямым образом влияет на инициацию процессов изменения в рассматриваемых подсистемах скоринговой гибридной системы.

Блок сравнительного анализа выводов решающих блоков реализует смещение оценки кредитоспособности в заданном направлении. При этом предусматривается обратная связь, влияющая на обучающий алгоритм нейросетевого модуля и его аппарат эволюционирования — генетический модуль развития топологии нейросети.

Таким образом, при комбинировании имеющихся методик построения экспертных систем открывается возможность моделирования гибкой системы принятия решений, развивающейся в процессе обработки данных.

Список литературы

1. Рутковская Д., Пилинский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. М.: Горячая линия-Телеком 2006. 383 с.
2. Ярушкина Н. Г. Основы теории нечетких и гибридных систем: учеб. пособие. М.: Финансы и статистика, 2009. 320 с.
3. Мэйз Э. Руководство по кредитному скорингу. Гревцов Паблшер, 2008. 464 с.
4. Борисов А. П., Крумберг О. А., Федоров И. П. Принятие решений на основе нечетких моделей. Рига: Зинатне, 1990. 184 с.
5. Нечеткие множества в моделях управления и искусственного интеллекта / Под ред. Д. А. Поспелова. М.: Наука, 1986. 312 с.
6. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Мир, 1976. 165 с.

УДК 519.72:621.372

Я. А. Архипова, аспирант,
e-mail: yand5883@yandex.ru,
Нижегородский государственный
лингвистический университет

Автоматическая периодизация динамики биржевых котировок на основе критерия минимума информационного рассогласования

Отталкиваясь от общей формулировки задачи о разладке случайного процесса, ставится и решается проблема автоматической периодизации случайного временного ряда биржевых котировок на однородные отрезки данных (кластеры) на основе авторегрессионной модели наблюдений и критерия минимума информационного рассогласования. Ставится и решается проблема выбора оптимального порога разладки. Приведены результаты экспериментальных исследований динамики фондовых рынков США и России.

Ключевые слова: случайный временной ряд, линейная авторегрессионная модель, критерий минимума информационного рассогласования

Введение

Задача прогнозирования случайных временных рядов (СВР) по конечным выборкам наблюдений является одной из центральных в статистическом анализе. Решение данной задачи возможно мето-

дами технического анализа, использующими авторегрессионные модели (АР-модели) наблюдений [1] с применением быстрых вычислительных процедур с высокой скоростью сходимости, таких как метод Берга и др., при исследовании однородных выборок наблюдений [2]. На практике, особенно при анализе социально-экономических процессов, границы однородности выборок наблюдений не всегда известны, что существенно понижает точность прогнозирования. Тогда перед исследователем ставится задача проверки однородности данных и периодизации СВР [3]. Одним из наиболее перспективных инструментов ее решения является метод обеляющего фильтра (МОФ) с использованием принципа минимального информационного рассогласования (МИР) в метрике Кульбака—Лейблера [4]. Применительно к проблеме автоматической периодизации данных задача сводится к проверке статистических гипотез о разладке случайного процесса по конечным (малым) выборкам наблюдений [5, 6]. При этом на первый план выходит выбор оптимального порога разладки на выходе обеляющего фильтра.

Постановка задачи

Известно, что для дискретных распределений вероятностей критерий МИР эквивалентен общесистемному критерию максимального правдоподобия, т. е. является оптимальным в байесовском смысле [7]. Для непрерывных сигналов свойство оптимальности критерия МИР сохраняется в гауссовском семействе распределений. Возьмем центрированный временной ряд $X \subset N(K)$ (где K — его

автокорреляционная матрица), разобьем некоторую выборку из него $X_0 = (X_1, X_2)$ на последовательные части $X_1 = \text{col}(x_{1,1}, x_{1,2}, \dots, x_{1,M_1})$ и $X_2 = \text{col}(x_{2,1}, x_{2,2}, \dots, x_{2,M_2})$ объемов M_1 и M_2 соответственно. Здесь $x_{i,j} = \text{col}(x_{i,j}(1), x_{i,j}(2), \dots, x_{i,j}(n))$ — это n -вектор (столбец) отсчетов сигнала в j -м цикле наблюдений в пределах выборки X_i , $i = 1; 2$, со свойством $M_x(x_{i,j} x_{i,j}^T) = K_i$ (M_x — символ математического ожидания, τ — транспонирование векторов). В общем случае число циклов наблюдений в выборках M_1, M_2 может быть различным. Полагаем выполнение соотношения $M_1 \gg M_2$ как очевидное условие для оперативного принятия решений о разладке в выборке X_1 по наблюдениям X_2 . Задача проверки однородности их распределений формируется, в общем случае, в терминах проверки статистических гипотез в отношении автокорреляционных матриц (АКМ) K_1 и K_2 [6]:

проверяется гипотеза

$$W_0: K_1 = K_2 \triangleq K_0$$

против альтернативы

$$W_1: K_1 \neq K_2.$$

Воспользуемся для ее решения оптимальным правилом принятия решений об обнаружении разладки в объединенной выборке X_0 :

$$W_1: \lambda(X_0) = M_1 \gamma_{1,0} + M_2 \gamma_{2,0} > \lambda_0, \quad (1)$$

где

$$\gamma_{k,0} = \frac{1}{2} [\text{tr}(S_k S_0^{-1}) - \ln |S_k S_0^{-1}| - n] \quad (2)$$

— величина информационного рассогласования (по Кульбаку — Лейблеру) гипотетического гауссова процесса с автокорреляционной матрицей S_k , $k = 1; 2$, по отношению к гауссову же процессу, но с матрицей S_0 ; $S_k = \frac{1}{M_k} \sum_{i=1}^{M_k} x_{k,i} x_{k,i}^T$, $k = 1; 2$ — классифици-

рованная выборка наблюдений; $S_0 = \frac{M_1}{M_0} S_1 + \frac{M_2}{M_0} S_2$ —

выборочная оценка максимального правдоподобия для АКМ K_0 объединенной выборки $X_0 = (X_1, X_2)$ ($|\cdot|$ — определитель $(n \times n)$ -матрицы, $\text{tr}(\cdot)$ — ее след).

Пороговый уровень λ_0 выбирается из следующего условия:

$$P\{\lambda(X_0) > \lambda_0 / W_0\} \leq \alpha_0 = \text{const}. \quad (3)$$

В случае $M_1 \gg M_2$ выражение для оптимальной решающей статистики (1) существенно упрощается:

$$W_1: \lambda(X_0) = \gamma_{2,1} = \frac{1}{2} [\text{tr}(S_2 S_1^{-1}) - \ln |S_2 S_1^{-1}| - n] > \tilde{\lambda}_0. \quad (4)$$

Для решения данной задачи используется принцип минимума информационного рассогласования между двумя рассматриваемыми выборками (стандартная формулировка критерия МИР) [4].

Метод обесляющего фильтра

Для частотной области обработки сигналов алгоритм (4) эквивалентен [8]:

$$W_1: \lambda(X_0) = \frac{1}{F} \sum_{f=1}^F \left(\frac{G_x(f)}{G_1(f)} + \ln \frac{G_x(f)}{G_1(f)} \right) - 1 > \tilde{\lambda}_0, \quad (5)$$

где $G_x(f)$ — выборочная оценка спектральной плотности мощности (СПМ) анализируемого сигнала $X_2(t)$ в функции дискретной частоты f ; $G_1(f)$ — аналогичная оценка СПМ "опорного" сигнала $X_1(t)$; F — верхняя граница частотного диапазона сигнала или используемого для его передачи канала связи.

При дополнительном условии нормировки АР-модели сигналов по дисперсии их порождающего шума слагаемое под знаком логарифма (5) оказывается практически равным нулю, и выражение для решающей статистики принимает вид

$$W_1: \lambda(X_0) = \frac{1}{F} \sum_{f=1}^F \frac{\left| 1 + \sum_{m=1}^p a_x(m) \exp(-j\pi m f / F) \right|^2}{\left| 1 + \sum_{m=1}^p a_1(m) \exp(-j\pi m f / F) \right|^2} - 1 > \tilde{\lambda}_0, \quad (6)$$

где $\{a_1(m)\}, \{a_x(m)\}$ — k -векторы коэффициентов линейной среднеквадратической авторегрессии сигналов $X_1(t)$ и $X_2(t)$ соответственно, полученных с применением МОФ [4]. При этом $X_2(t)$ делится на сегменты одинаковой длительности.

Выражение (6) описывает выборочную оценку величины информационного рассогласования (ВИР) между сигналом X на входе и опорным сигналом в частотной области [8]. Последовательно применяя правило (6) к выделенным сегментам данных, мы либо фиксируем продолжение однородности данных, если сигнал сохраняет свой закон распределения, либо начало нового сегмента данных в противном случае.

Выражение (6) позволяет рассчитать решающую статистику λ_0 в виде порогового уровня $\tilde{\lambda}_0$ и выбрать оптимальный порог разладки на выходе обесляющего фильтра. При наличии массивов данных однородные участки можно объединить в кластер по принципу минимума информационного рассогласования [4].

Результаты экспериментальных исследований

Для практической апробации предложенного алгоритма был выбран фондовый рынок США, который может быть взят за основу при выявлении

закономерностей рыночных колебаний, а также развивающийся фондовый рынок России. Анализ проводился по основным индексам S & P 500 и Nasdaq, а также акциям компаний Cabot Oil & Gas Corporation и Ford для американского рынка и акциям НК "Лукойл" и "Газпром" для российского рынка. В качестве исследуемых временных рядов использовались последовательности ежедневных цен закрытия торгов $c(t)$, $t = 1, 2, \dots$, полученные из электронных баз данных с Интернет-сайта <http://www.finam.ru/analysis/export/default.asp?id=1> на бирже NYSE в США и с Интернет сайта <http://quote.rbc.ru/exchanges/> на бирже ММВБ в России за период с 2007 по 2011 гг. Все вычисления проводились в среде MatLab 7.11.

Были рассмотрены ежедневные изменения котировок акций как непрерывный сигнал $x = x(t)$, $t = 1, 2, \dots$, в функции дискретного времени t , разделенные на сегменты $x_n = \{x_1, \dots, x_c\}$ по N отсчетов каждый. Применительно к последовательности приращений ежедневных цен закрытия биржи, вычисляемых по первой разности, $x(t) = c(t) - c(t-1)$, где $c(t)$ — цена закрытия, $t = 1, 2, \dots$. Длительность сегмента данных N была определена в 20...100 рабочих дней. При этом значение порога разладки варьировалось в пределах $\tilde{\lambda}_0 = 0,00 \dots 1,0$. Коэффициенты авторегрессии определялись с использованием рекуррентной процедуры Берга—Левинсона с высокой скоростью сходимости [4]:

$$a_m(i) = a_{m-1}(i) + c_m a_{m-1}(m-i), \quad i = \overline{1, m};$$

$$c_m = S_{m-1}^{-2} \sum_{t=m}^{N-1} \eta_{m-1}(t) v_{m-1}(t-1);$$

$$S_{m-1}^2 = 0,5(n-m)^{-1} \sum_{t=m}^{N-1} [\eta_{m-1}^2(t) + v_{m-1}^2(t-1)]; \quad (7)$$

$$\eta_m(t) = \eta_{m-1}(t) - c_m v_{m-1}(t-1);$$

$$v_m(t) = v_{m-1}(t-1) - c_m \eta_{m-1}(t), \quad t = 0, 1, \dots, N-1;$$

$$\sigma_m^2 = (1 - c_m^2) \sigma_{m-1}^2, \quad \sigma_0^2 = S_0^2, \quad m = \overline{1, p},$$

при ее инициализации системой равенств $v_0(n) = \eta_0(n-1) = x(n)$. Здесь a_m — вектор коэффициентов линейной авторегрессии сигнала $x(n)$; c_m — коэффициенты линейного предсказания (коэффициенты отражения); v_m и η_m — ошибки линейного предсказания вперед и назад; S_m — классифицированная выборка наблюдений; S_0 — выборочная оценка максимального правдоподобия; n — номер шага. Порядок АР-модели был установлен постоянным $p = 30$. Полученные результаты отображены на рис. 1–4.

На рис. 1 изображена экспериментальная зависимость числа кластеров в динамике цен на акции компании Cabot Oil & Gas Corporation от порога разладки и длины сегмента, где значение порога

разладки $\tilde{\lambda}_0$ указано в относительных единицах, а длина сегмента N — в отсчетах. На рис. 1 видно, что число получаемых при этом кластеров меняется в зависимости от значения предъявляемого порога по ВИР $\tilde{\lambda}_0$ и объема сегмента данных N . При максимальных значениях обоих параметров число кластеров минимально. Исходя из полученных закономерностей значения параметров были выбраны следующим образом: $N^* = 80$ рабочих дней, или примерно 3...4 месяца, и $\tilde{\lambda}_0^* = 0,6$.

На рис. 2 представлена временная диаграмма приращения цены закрытия $\Delta C(t)$ для акций компании Cabot Oil & Gas Corporation за период с 12.10.07 по 28.02.11. За указанный период было выявлено 10 кластеров: 7 продолжительных кластеров длиной до 6 месяцев (кластеры 1, 2, 5–8, 10) и 3 коротких кластера длиной приблизительно 1...1,5 месяца (кластеры 3, 4, 9). Короткие кластеры отражают пограничные (неустойчивые) состояния рынка.

На рис. 3 представлены основные характеристики кластера № 10. Он состоит из 128 отсчетов, имеет продолжительность 26 недель или 6 месяцев.

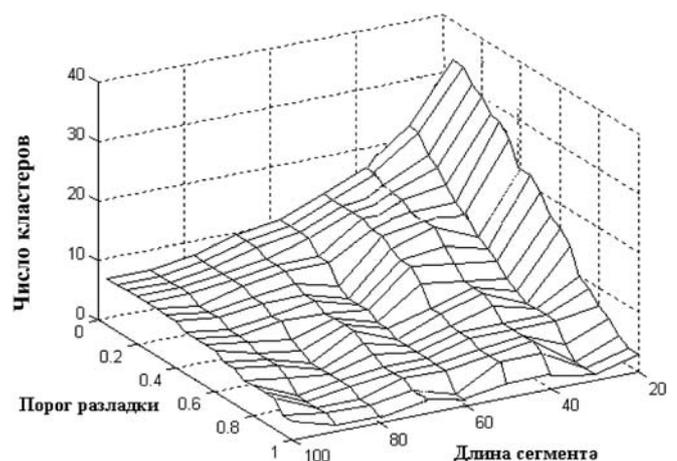


Рис. 1. Экспериментальная зависимость числа кластеров в динамике цен на акции компании Cabot Oil & Gas Corporation от порога разладки и длины сегмента

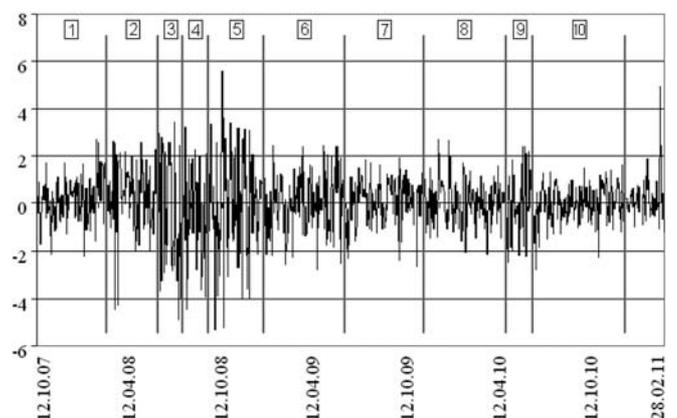


Рис. 2. Временная диаграмма приращения цены закрытия для акций Cabot Oil & Gas Corporation с выделенными границами кластеров

Выводы

Были проведены исследования рынков США и России на примере основных индексов S & P 500 и Nasdaq, а также акций компаний Cabot Oil & Gas Corporation и Ford, НК "Лукойл" и "Газпром" согласно новому методу автоматической периодизации динамики биржевых котировок на основе критерия МИР [3]. Были также выявлены участки реализаций данных, аналогичные для обоих рынков.

Предложенный метод обладает высокими динамическими свойствами и широкими функциональными возможностями. Он позволяет выделить периоды во временном ряду, в течение которых использование методов прогнозирования цены является малоэффективным — короткие (переходные) кластеры или высокоэффективным — продолжительные кластеры, что даст возможность снизить степень риска при биржевой игре и вовремя предупредить биржевых игроков о наступлении нестабильного периода изменения цены.

Применение данного метода в экономической теории можно отнести к ряду приоритетных направлений исследований, в частности, решению задачи кластерного анализа динамических процессов в экономике и связанную с ней общую задачу экономической диагностики.

Список литературы

1. Савченко В. В. Прогнозирование социально-экономических процессов на основе адаптивных методов спектрального оценивания // Автометрия. 1999. № 3. С. 99.
2. Савченко В. В. Проверка однородности выборочных данных в задачах спектрального оценивания // Радиотехника и электроника. 1999. Т. 44, № 1. С. 65—69.
3. Савченко В. В. Информационная теория колебаний биржевых котировок в динамике // Информационные технологии. 2011. № 3. С. 57—63.
4. Акатьев Д. Ю., Савченко В. В. Обнаружение разладки случайного процесса по выборке на основе принципа минимума информационного рассогласования // Автометрия. 2005. № 2. С. 68—74.
5. Марпл С. Л. Цифровой спектральный анализ и его приложения. М.: Мир, 1990.
6. Савченко В. В., Губочкин И. В. Фонетический анализ речи методом переменного дерева // Изв. вузов России. Радиоэлектроника. 2008. Вып. 3. С. 14—20.
7. Савченко В. В. Различение случайных сигналов в частотной области // Радиотехника и электроника. 1997. Т. 42, № 4. С. 426—431.
8. Савченко В. В. Обнаружение и прогнозирование разладки случайного процесса на основе спектрального оценивания // Автометрия. 1996. № 2. С. 77—84.



Рис. 3. Основные характеристики кластера № 10

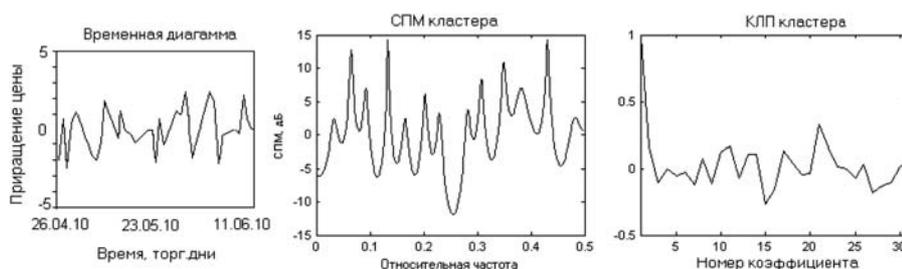


Рис. 4. Основные характеристики кластера № 9

Оценка спектральной плотности мощности (СПМ) кластера имеет сложный спектр и находится в диапазоне от -5 до 7 дБ, относительная частота варьируется от 0 до $0,5$. Даны также коэффициенты линейного предсказания (КЛП), значения которых от $-0,3$ до $0,9$ при порядке АР-модели $p = 30$.

Имеющаяся информация о форме СПМ, временная диаграмма и значения КЛП позволяют выявить временные интервалы, когда поведение рынка является малопредсказуемым, и точность прогнозов значительно снижается.

На рис. 4 показаны основные характеристики короткого (переходного) кластера № 9, который предшествует рассмотренному. Он состоит из 34 отсчетов, имеет продолжительность 7 недель или 1,5 месяца. СПМ характеризуется еще большей нестабильностью спектра и находится в диапазоне от -13 до 14 дБ при относительной частоте от 0 до $0,5$. Значения КЛП для 30 коэффициентов варьируются от $-0,4$ до 1 .

Последний кластер временного ряда характеризует динамику изменения цены в настоящее время. Полученные в результате эксперимента кластеры в динамике биржевых котировок США в целом имеют устойчивую продолжительность 3...6 месяцев. Это справедливо и для фондового рынка России, который имеет ярко выраженный развивающийся характер и является более нестабильным по сравнению с фондовым рынком США, поскольку число переходных кластеров на 35 % больше для рассматриваемых инструментов анализа. Можно также говорить о сильной зависимости российского рынка от "настроений" рынка США и его относительной стабилизации в течение второй половины 2009 — начале 2010 г., особенно по сравнению с 2008 г.

2-я Международная конференция по истории отечественной вычислительной техники и информатики SoRuCom—2011

С 12 по 16 сентября 2011 г. в Великом Новгороде на базе Новгородского государственного университета им. Ярослава Мудрого прошла 2-я Международная конференция "Развитие вычислительной техники и ее программного обеспечения в России и странах бывшего СССР (SoRuCom—2011)". Конференция проводилась под эгидой Международной федерации по обработке информации (IFIP) и при поддержке Российского фонда фундаментальных исследований (РФФИ).

Первая такая конференция состоялась в г. Петрозаводске в 2006 г. Вызванный ею огромный интерес, важность и актуальность тематики, а также то обстоятельство, что многие заметные вехи развития вычислительной техники, программирования и информационных технологий в нашей стране на первой конференции затронуты не были, предопределили необходимость проведения очередной конференции SORUCOM—2011. Это тем более актуально, что на 2010—2011 гг. пришлось заметные юбилейные даты: в 2010 г. исполнилось 90 лет со дня рождения Анатолия Ивановича Китова, в 2011 г. — 100 лет со дня рождения Алексея Андреевича Ляпунова и 80 лет со дня рождения Андрея Петровича Ершова. В 2011 г. также исполнилось 60 лет с момента создания ЭВМ "М-1" и 50 лет Институту кибернетики им. В. М. Глушкова в Киеве.

В программный комитет конференции вошли представители России, США, Украины, Белоруссии, сотрудники московских НИИ, ОАО и учреждений РАН, ведущих вузов и музеев, учреждений Сибирского отделения РАН, Калифорнии (США) и Массачусетского технологического института (США), Средиземноморского института прикладных наук. Программа конференции включала пленарные заседания ("Мемориальная сессия памяти российских ученых А. П. Ершова, А. И. Китова и А. А. Ляпунова") — 15 докладов с показом фильма и работу по секциям: "История создания ЭВМ" (17 докладов), "Программирование" (16 докладов), "Сети ЭВМ" (6 докладов) и "Музеи вычислительной техники" (5 докладов).

Как было отмечено выше, основные доклады Мемориальной пленарной сессии были посвящены памяти российских ученых А. П. Ершова, А. И. Китова и А. А. Ляпунова. Здесь основные акценты были сделаны на становлении в нашей стране вычислительной техники, информатики и кибернетики (в первой половине 1950-х гг. считавшейся "буржуазной лженаукой"), на непростые этапы их признания в СССР, на личный вклад (а порой и личные трагедии) лидеров этих научных направлений. В докладе Г. И. Марчука, Президента АН СССР в 1986—1991 гг., был дан развернутый анализ первых и последующих шагов становления отечественной вычислительной техники, роли не только ведущих советских ученых, но и дипломатов, например посла СССР в США А. Ф. Добрынина. Несомненный интерес участников конференции вызвало выступление директора Института системного программирования РАН В. П. Иванникова. В докладах других выступавших (россиян и иностранцев) акценты очевидны из их заголовков: "Точка отсчета отечественной кибернетики" (В. А. Китов и В. В. Шилов), "Плодотворное сотрудничество пионеров кибернетики (А. И. Берг, А. И. Китов, А. А. Ляпунов)" (С. Б. Оганджян и С. П. Прохоров); "Информатика от Силиконовой Долины до Золотой Долины: Андрей Ершов и Джон Маккарти" (К. Тамарченко, Принстонский университет, США); "У истоков отечественной военной информатики" (А. Я. Приходько); "The Institute and three socio-technological transformations of IT" (Р. Балодис и И. Опмане, Рига); "A Review of Select Periodicals. What Did the Americans Know?" (Алекс Боханек, Computer History Museum, Mountain View, Калифорния, США); "Человеческие качества А.А. Ляпунова" (Я. И. Фет); "История развития вычислительной техники в Армянской ССР" (С. Б. Оганджян) и др.

Содержание секционных докладов по истории создания отечественных ЭВМ также можно проследить по их заголовкам: "60 лет первой отечественной ЭЦВМ М-1", "Тройчатая ЭВМ "Сетунь 70", "История развития микроэлектроники в ленинградских КБ", "Первая Отечественная многопроцессорная ЭВМ М-10", "Перспективы развития цифровых управляющих вычислительных систем реального времени (ЦУВС РВ)", "Вычислительные средства для систем противоракетной и противосамолетной обороны страны. Роль С. А. Лебедева и В. С. Бурцева", "К истории разработки БЦВМ для противоракет дальнего перехвата А925". Как бы

в дополнение к военной тематике прозвучали доклады "Мини-ЭВМ "Электроника-100" — шаг к гражданским системам реального времени" (В. М. Трояновский), а также "Проблемы управления электрофизическими установками как предпосылка создания оригинальных ЭВМ" и "Применение универсальных процессоров АП-20/АП-32 для обработки физической информации", сделанные сотрудниками Института ядерной физики СО РАН.

На секции "Программирование" обсуждались вопросы создания алгоритмических языков программирования и системного и прикладного программного обеспечения, теоретико-графовые методы в программировании и др. Значительный интерес представляют опубликованные в сборнике трудов конференции доклады "ФОРТ в СССР и в России"; "Проект КЛАСТОС"; "Системное программное обеспечение вычислительной системы "Электроника ССБИС"; "Становление отечественного программирования в период с 1950 по 1960 г."; "Истоки алгоритмического языка АЛГЭМ и его место в творчестве А. И. Китова"; "Первые реализации языка Lisp в СССР"; "О работах Валентина Федоровича Турчина по кибернетике и информатике"; "Школа программирования ИПМ им. акад. М. В. Келдыша" и др. Наряду с докладами о языках программирования прозвучал интересный доклад о первых расчетах на ЭВМ "М-2" нагрева баллистических ракет в атмосфере. Оживленную дискуссию вызвала демонстрация анимационно-документального фильма о выдающемся российском конструкторе ЭВМ Б. И. Рамееве, снятого его внучкой А. Белкиной (Emerson College, Бостон, США).

В докладах, сделанных на секции "Сети ЭВМ", отмечалась роль акад. Г. И. Марчука в создании вычислительных центров Сибири и развитии сетевых технологий в СО АН СССР в 1970-е годы; рассматривались проблемы разработки проекта ОГАС (А. В. Кутейников); значение АСУ "Львов" и АСУ "Кунцево" (Е. П. Стрюкова). Проректор Новгородского госуниверситета им. Ярослава Мудрого Н. В. Курмышев детально рассказал об информатизации своего университета.

На секции "Музеи ВТ" прозвучали доклады о коллекциях музея ИТМ и ВТ РАН, московского Политехнического музея, Казанского музея вычислительной техники. Известный специалист по истории российских счетных приборов Тимо Лейпяля (Финляндия) сделал доклад о производстве арифмометров Однера в России, Швеции и СССР в 1912—1928 гг.; о забытых счетных машинах русско-польского изобретателя XIX в. Яна Барановского рассказал В. В. Шилов.

Культурная программа конференции включала экскурсии по Великому Новгороду, одному из

древнейших и красивейших городов России и теплоходную прогулку по реке Волхов.

Следует отметить широкую географию участников конференции (Россия, Латвия, Финляндия, США, Канада, Узбекистан), а также широкий спектр учреждений и организаций. Среди последних значатся 17 (отечественных и зарубежных) университетов, 6 (отечественных и зарубежных) музеев, четыре отечественных НИИ по созданию вычислительных комплексов, 14 институтов РАН, 8 (отечественных и зарубежных) IT-компаний, 4 издательства. Рабочими языками были русский и английский.

Не менее важны и персоналии участников конференции — от известных ученых и руководителей до разработчиков первых ЭВМ. О ярком выступлении на пленарном заседании Г. И. Марчука уже было сказано. Конференцию украсило присутствие Т. М. Александриды, Я. А. Хетагурова, Р. И. Подловченко, А. Н. Томилина, А. А. Берса, Л. В. Городней, Ю. Л. Кеткова, И. М. Лисовского и других глубокоуважаемых ветеранов отечественных информационных технологий.

В Решении Международной конференции "SoRuCom—2011" отмечается высокий уровень большинства представленных докладов, их актуальность и большое значение для истории отечественной науки. Тексты представленных на конференции докладов опубликованы в виде отдельного тома [1] (здесь стоит заметить, что сборник избранных докладов предыдущей конференции недавно был опубликован на английском языке издательством Springer [2]). Ответственный редактор сборника, д-р физ.-мат. наук А. Н. Томилин, а также Э. М. Пройдаков, Н. А. Черемных, И. А. Крайнева, Н. В. Курмышев и их коллеги потратили очень много времени и сил для того, чтобы конференция состоялась и прошла столь успешно. Следующую конференцию по данной тематике планируется провести в 2014 г.

Список литературы

1. **Труды** Международной конференции по истории вычислительной техники и информатики (SoRuCom 2011). Великий Новгород: ЗАО "Новгородский технопарк", 2011. 365 с.
2. **Perspectives** on Soviet and Russian Computing. First IFIP WG 9.7 Conference, SoRuCom 2006. Petrozavodsk, Russia, July 3—7, 2006. Revised Selected Papers. J. Impagliazzo, E. Proydakov (Eds.). Springer, 2011. 272 p.

В. А. Китов, Институт истории естествознания и техники им. С. И. Вавилова РАН
e-mail: vladimir.kitov@mail.ru
В. М. Трояновский, Национальный исследовательский университет "МИЭТ"
e-mail: troy40@mail.ru

CONTENTS

Talalay M. S., Trushin K. V., Venger O. V. <i>Logic Synthesis of a Boolean Function on Regular Layout Templates</i>	2
--	----------

In this paper we describe the method of logic synthesis that can be applied to design ICs within the limits of strongly regular layout structures. Proposed technique skips the hierarchy of standard cells and uses regular layout templates as minimal functional elements. We describe the method to decompose logic functions using combinational circuits of general view. We propose a set of transistor circuits that have regular layout for usage as the set of functional elements for regular layout synthesis and demonstrate examples of template-based synthesis solutions. We show that the template-based approach can significantly reduce design area in comparison with standard cells approach.

Keywords: regular layout, regular fabrics, logic decomposition, layout templates

Bibilo P. N. <i>Expert Knowledge in CAD for Digital Devices</i>	11
--	-----------

The problem of development of expert systems for CAD is considered in an example of the problem area of logical design of digital devices. Organization of design systems is analysed, and a model of expert knowledge representation is reported.

Keywords: digital devices, CAD systems, expert knowledge, productions, frames

Tarasov V. N., Mezentseva E. M. <i>Organization of the Computer Networks Security. Multimodular Filtering Method for Spam in Web-Sites</i>	18
---	-----------

We show the model of spam filtering in comments, feedback forms, and other interactive sections. Also given the general scheme of the filter and filtering process, learning of the filter and algorithm of spam identification.

Keywords: spam, message, filter, principal scheme, token, orthogonal sparse bigrams, Bayes formula

Karutin S. N. <i>Autonomous GLONASS Navigation Satellites Ephemeris and Clocks Estimation Method</i>	23
---	-----------

Analysis of existing GLONASS navigation satellite orbits and clocks estimation methods had been done. Autonomous and real time on-board of GLONASS navigation satellites orbits and clocks determination method, based on a use of radio navigation field created by ground stations with GLONASS satellites navigation signal support, is proposed. Results of GLONASS navigation satellites real-time orbits and clocks determination system design are presented. Conclusions about possible GLONASS space complex components development directions had been made based on these studies.

Keywords: GLONASS, orbits and clocks determination

Bochkov M. V., Laluskin Yu. P., Shkadov A. A. <i>The Model of Conditions of Protection Computer Network in Based Hierarchical Colored Petri Nets</i>	29
---	-----------

In the article approach to the management of secure computer network, based on the policy of security, is described. The model of definition of secure computer network has been offered. The formal model is designed compositionally using colored Petri nets (CPN), which can serve as a high-level design for penetration testing.

Keywords: colored Petri nets, security policy, vulnerabilities, computer network

Kozlenko A. V. <i>Information Security Estimation Method from Unauthorized Access in Computer Networks in the Conditions of Complex Threats</i>	33
--	-----------

In this article the urgency of information security quantitative estimation problem from unauthorized access in computer networks is shown. Information security estimation method from unauthorized access in computer networks on a basis of security graph is offered. The offered method provides rise of a information security management efficiency in computer networks due to complex security metric, and also application of the security graph which considering real structure of a computer network and information security system.

Keywords: security estimation, unauthorized access, computer network, security coefficient, security graph

Bobkov V. A., Ronshin Yu. I., Mashentsev V. Yu. <i>Depth-Maps Generation from Multiple Images</i>	39
--	-----------

A modification of the previously developed algorithm for the creation of depth maps from multiple images with the implementation on CUDA is suggested. The algorithm is based on the use of space-sweep strategy, the adaptive window technique and consideration of the connection points. The results of computational experiments on efficiency algorithm are presented.

Keywords: multiple images, depth map, point matching, CUDA

Bimakov V. A., Bimakov E. V. <i>Principles of Building and of Functioning of Devices for High Speed Processing of Dot Geometrical Models</i>	44
---	-----------

The general principles of building and functioning of the devices intended for application in intellectual systems of technical vision, in control systems by high-maneuverable autonomous robots, in systems of high-realistic computer graphics and animation, in systems of the automated designing are being described.

The method, being used in these devices, to process the scenes and images, allows to offer the new superfast processing algorithms, which are not dependent on geometrical forms of objects being processed .

Keywords: high-speed processing of scenes and images, geometrical models

Panin I. G., Blagoveshchenskii V. V. Numerical Solution of Equations of Evolution Dislocation Sources Frank — Read 50

This paper describes the principles of constructing a model of dislocation Frank — Read source, based on the solution of linear partial differential equations of parabolic type. Produced by choosing the most suitable schemes for the numerical solution of the equation. Studied problems of stability, convergence, accuracy difference scheme and the sweep method solving the system of linear equations, compiled on the basis of the equation being solved. The choice made solution algorithm with minimum time spent.

Keywords: crystal, dislocation, dislocation source model, the numerical solution, convergence, stability, accuracy

Lvovich Ya. E., Kashirina I. L., Tuzikov A. A. Genetic Algorithm Solution the Problem of Increasing the Reliability Multicriterial Reservation 56

The process of automated design systems is the multi-criteria optimization problem, which is one of the criteria — to ensure a given level of reliability. The relevance of scientific publications of this theme is that there is no universal formal methods, algorithms, programs to help automate the management of reliability for any complex technical system at all stages of its life cycle. In this paper the mathematical model and genetic algorithm for solving reliability redundancy components Clay information and control systems for fire tests of liquid rocket engines.

Keywords: liquid rocket engine, control system testing, reservation, multicriteria mathematical model, the genetic algorithm

Agievich S. N. Parameter Estimation for Signals in the Bases of Spline-Characters Functions 61

The methods and implement their algorithms for the evaluation of signal parameters in the bases of spline-characters functions (BFSH). Described by the rapid transformation of the signals in BFSH. Justified by its high computational efficiency. Demonstrates how to win in the volume of calculations in the transition from discrete exponential functions to the particular case BFSH — the basis of spline-Vilenkin — Chrestenson functions.

Analyzes the effectiveness of the method of evaluation of the carrier frequency signals from the standpoint of accuracy, speed and computational resources.

Keywords: estimation parameter signal, functions spline-character, quick transformations signal, the base of spline-Vilenkina — Krestensona functions

Ryndin A. A., Ulyev V. P. Development a Hybrid System of Scoring Credit Rating for Corporate Clients 69

The article reviews the main challenges of designing a hybrid scoring systems for evaluating the credit-worthiness of legal entities. We consider the classification of hybrid systems and the problems of their design, proposed compensation model estimates the integration of different techniques. The approach of constructing a hybrid fuzzy neurogenetical system of decision-making. Presented a general hybrid system flow chart, preview the main module algorithm considered and principles of interaction between subsystems.

Keywords: credit scoring, hybrid systems, fuzzy sets, neural networks, genetic algorithms

Arkhipova Ya. A. Automatic Periodization of Exchange Quotation Dynamics on the Basis of Minimum Information Divergence Criterion 73

According to the general enunciation of the disorder problem set and solved the task automatic segmentation of exchange quotations stochastic time series into homogeneous data segments (clusters) on the basis of the autoregressive model and minimum of an information mismatch criterion. The problem of the selection and validation of the admissible level of disorder of time series raised and resolved. The results of the experimental investigations of stock market dynamics in the USA and Russia are shown.

Keywords: stochastic time series, linear autoregressive model, minimum of information mismatch criterion

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *Е.В. Конова*.

Корректор *Е.В. Комиссарова*.

Сдано в набор 02.04.2012. Подписано в печать 23.05.2012. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 9,8. Заказ ИТ612. 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.