



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ**

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ТАГАНРОГСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ**

**ШТУТТГАРТСКИЙ УНИВЕРСИТЕТ**

**ИНЖЕНЕРНАЯ АКАДЕМИЯ УКРАИНЫ**

**ООО „Технопарк ДонНТУ „УНИТЕХ”**



**""МОДЕЛИРОВАНИЕ  
И КОМПЬЮТЕРНАЯ ГРАФИКА""**

**Материалы первой международной  
научно-технической конференции**

**г. Донецк**

**04-07 октября 2005 года**

**Донецк 2005  
ДонНТУ**

ББК 33.31  
УДК 004.925.8: 681.5: 658.5  
М 74

М 74 **Моделирование и компьютерная графика:** Материалы 1-й международной научно-технической конференции, г Донецк, 04-07 октября 2005 г. — Донецк, ДонНТУ, Министерство образования и науки Украины, 2005. — 285 с.

Доклады ученых и специалистов по проблемам разработки и исследования систем моделирования и компьютерной графики. Приводятся результаты работ по построению аппаратного и программно-алгоритмического обеспечения средств компьютерной графики, по параллельному моделированию сложных динамических систем и по моделированию и проектированию средств вычислительной техники. Материалы предназначены для ученых, преподавателей, аспирантов и студентов высших учебных заведений, которые занимаются вопросами разработки и использования современных компьютерно - информационных систем.

Доповіді вчених і фахівців з проблем розробки і дослідження систем моделювання і комп'ютерної графіки. Приводяться результати робіт з побудови апаратного й програмно-алгоритмічного забезпечення засобів комп'ютерної графіки, з паралельного моделювання складних динамічних систем і з моделювання і проектування засобів обчислювальної техніки. Матеріали призначені для вчених, викладачів, аспірантів і студентів вищих навчальних закладів, які займаються питаннями розробки й використання сучасних комп'ютерно-інформаційних систем.

Редакционная коллегия:

д-р техн. наук Минаев А.А.,  
д-р техн. наук Башков Е.А.,  
д-р техн. наук Зори А.А.,  
д-р техн. наук Ли В.Г.,  
д-р техн. наук Пьявченко О.Н.,  
д-р техн. наук Святный В.А.,  
д-р техн. наук Скидан И.А.,  
д-р техн. наук Фельдман Л.П.

Рекомендовано к печати Ученым советом Донецкого национального технического университета Министерства образования и науки Украины  
Протокол № 6 от 02 сентября 2005 р.

© Донецкий национальный технический университет Министерства образования и науки Украины, 2005

## СОДЕРЖАНИЕ

### Пленарное заседание

	<b>Приветственное слово</b>	<b>5</b>
<b>П.01</b>	<b>Палагин А.В., Опанасенко В.Н.</b> МОДЕЛИРОВАНИЕ РЕКОНФИГУРИРУЕМЫХ КОМПЬЮТЕРНЫХ СИСТЕМ	<b>6</b>
<b>П.02</b>	<b>Валькман Ю. Р., Быков В. С.</b> О МОДЕЛИРОВАНИИ ВИЗУАЛЬНОГО МЫШЛЕНИЯ В КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЯХ	<b>11</b>
<b>П.05</b>	<b>Аноприенко А.Я., Башков Е.А., Самойлова Т.А.</b> ПОРТАЛ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ: ЦЕЛИ, ЗАДАЧИ И ОСОБЕННОСТИ ОРГАНИЗАЦИИ	<b>16</b>

### Секция 1. «Компьютерная графика»

<b>С1.02</b>	<b>Ли В.Г.</b> КОМПЬЮТЕРНЫЙ СТЕНД ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ОПЕРАТОРСКОЙ ДЕЯТЕЛЬНОСТИ	<b>21</b>
<b>С1.03</b>	<b>Башков Е.А., Карабчевский В.В.</b> ГРАФИЧЕСКАЯ ПОДГОТОВКА СТУДЕНТОВ СПЕЦИАЛЬНОСТИ «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ»	<b>27</b>
<b>С1.04</b>	<b>Гетманова Е.Е.</b> ИСПОЛЬЗОВАНИЕ КОМПЬЮТЕРНОЙ ГРАФИКИ ПРИ ИЗУЧЕНИИ ФИЗИКИ	<b>31</b>
<b>С1.05</b>	<b>Гаврилова З.П., Зулкарнеева В.В.</b> РАЗРАБОТКА ЭЛЕКТРОННОГО УЧЕБНИКА «ИНЖЕНЕРНАЯ И КОМПЬЮТЕРНАЯ ГРАФИКА В ПРИМЕРАХ»	<b>38</b>
<b>С1.06</b>	<b>Фатгахова Н.Г.</b> ОБУЧАЮЩАЯ ПРОГРАММА ДЛЯ GEOMEDIA PROFESSIONAL	<b>41</b>
<b>С1.07</b>	<b>Быков В. С.</b> ПРИНЦИПЫ ПОСТРОЕНИЯ БАЗ ЗНАНИЙ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ	<b>44</b>
<b>С1.08</b>	<b>Костюкова Н.С.</b> СОВРЕМЕННЫЕ СИСТЕМЫ ПОИСКА ИЗОБРАЖЕНИЙ НА ОСНОВЕ ИХ СОДЕРЖИМОГО	<b>49</b>
<b>С1.09</b>	<b>Вовк О.Л.</b> ПРИМЕНЕНИЕ ПРОЦЕДУРЫ КЛАСТЕРИЗАЦИИ ДЛЯ СОДЕРЖАТЕЛЬНОГО ПОИСКА ИЗОБРАЖЕНИЙ	<b>55</b>
<b>С1.10</b>	<b>Башков Е.А., Пауков Д.П.</b> МАТЕМАТИЧЕСКИЕ МОДЕЛИ ПОВЕРХНОСТЕЙ СЛОЖНОЙ ФОРМЫ В СИСТЕМАХ МОДЕЛИРОВАНИЯ И ВИЗУАЛИЗАЦИИ	<b>59</b>
<b>С1.11</b>	<b>Аббасов И.Б.</b> ИСПОЛЬЗОВАНИЕ СИСТЕМ МОДЕЛИРОВАНИЯ ДЛЯ СОЗДАНИЯ ИНТЕРЬЕРА	<b>66</b>
<b>С1.12</b>	<b>Ковальский С.В., Зори С.А.</b> МОДИФИЦИРОВАННЫЙ АЛГОРИТМ РЕКОНСТРУКЦИИ ТРЕХМЕРНЫХ СЦЕН ГОРОДСКИХ ЛАНДШАФТОВ НА ОСНОВЕ ФОТОИЗОБРАЖЕНИЙ	<b>70</b>
<b>С1.13</b>	<b>Григорьев А.В.</b> МЕТОДЫ И СРЕДСТВА РАБОТЫ С ГРАФИЧЕСКИМИ МОДЕЛЯМИ В САПР ПАРОГАЗОВЫХ УСТАНОВОК СПРУТ	<b>77</b>

### Секция 2. «Моделирование и проектирование вычислительной техники»

<b>С2.01</b>	<b>Белецкий В.Н., Резникова С.А., Чемерис А.А.</b> ПЛАНИРОВАНИЕ ЗАДАЧ ПРИ МОДЕЛИРОВАНИИ ЦИФРОВЫХ УСТРОЙСТВ НА МНОГОПРОЦЕССОРНЫХ СИСТЕМАХ	<b>87</b>
<b>С2.02</b>	<b>Алишов Н.И.</b> КОЭФФИЦИЕНТ ПОЛЕЗНОЙ ПЕРЕДАЧИ ДАННЫХ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ	<b>92</b>
<b>С2.03</b>	<b>Андрюхин А.И.</b> МОДЕЛИРОВАНИЕ И ДИАГНОСТИРОВАНИЕ МОП-СТРУКТУР	<b>97</b>
<b>С2.04</b>	<b>Труб И.И.</b> О ПАРАДОКСЕ НАРУШЕНИЯ СТАЦИОНАРНОГО РЕЖИМА ПРИ МОДЕЛИРОВАНИИ WEB-КЛАСТЕРА С ДИСЦИПЛИНОЙ ОБСЛУЖИВАНИЯ SITE-E	<b>105</b>
<b>С2.05</b>	<b>Григорьев А.В., Кошелева Д.А.</b> ИНТЕЛЛЕКТУАЛИЗАЦИЯ ПРОЦЕССА ПРОЕКТИРОВАНИЯ АППАРАТУРЫ СРЕДСТВАМИ ЯЗЫКА VHDL	<b>110</b>
<b>С2.06</b>	<b>Жуков I.A., Гуменюк В.О., Пашенко Н.В.</b> ПРЕДСТАВЛЕНИЯ ЦИФРОВОЙ ИНФОРМАЦИИ У ПАРАЛЛЕЛЬНЫХ ОБЧИСЛЮВАЛЬНЫХ СИСТЕМАХ НЕПОДЛЬНИМИ КОДАМИ “M N”	<b>117</b>
<b>С2.07</b>	<b>Ладыженский Ю.В., Мирецкий А.В.</b> МОДЕЛИРОВАНИЕ МУЛЬТИАГЕНТНЫХ АЛГОРИТМОВ МАРШРУТИЗАЦИИ СООБЩЕНИЙ В КОМПЬЮТЕРНЫХ СЕТЯХ	<b>122</b>
<b>С2.08</b>	<b>Ладыженский Ю.В., Куркчи В.А.</b> МОДЕЛИРОВАНИЕ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ ПОИСКА НЕЗАВИСИМЫХ МНОЖЕСТВ НА ГРАФАХ	<b>127</b>
<b>С2.09</b>	<b>Бельков Д.В.</b> КЛАССИФИКАЦИЯ МЕТОДОВ БАЛАНСИРОВКИ НАГРУЗКИ ПРОЦЕССОРОВ ПРИ РАСПРЕДЕЛЕННОМ МОДЕЛИРОВАНИИ ЦИФРОВЫХ УСТРОЙСТВ	<b>134</b>
<b>С2.10</b>	<b>Михайлова Т.В.</b> ОЦЕНКА ЭФФЕКТИВНОСТИ КЛАСТЕРНЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ ВЕРОЯТНОСТНЫХ МОДЕЛЕЙ	<b>141</b>

C2.11	<b>Возняк Н.О.</b> АНАЛІЗ ЕПІДЕМІОЛОГІЧНОГО ПОРОГУ ПОШИРЕННЯ КОМП'ЮТЕРНОГО ВІРУСУ В ІНТЕРМЕРЕЖАХ	146
C2.12	<b>Костин В.И., Краснокутская М.В.</b> МОДЕЛИРОВАНИЕ РАСПАРАЛЛЕЛИВАНИЯ ЗАДАЧИ С ПОМОЩЬЮ ГРАФА ПОТОКОВ ДАННЫХ	151
C2.13	<b>Аноприенко А.Я. Рычка С.В., Хасан Аль Абабнех</b> СПОСОБЫ И СРЕДСТВА МОДЕЛИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ С ЦЕЛЬЮ ОБЕСПЕЧЕНИЯ ЭФФЕКТИВНОСТИ ФУНКЦИОНИРОВАНИЯ WEB-СЕРВИСОВ	156
C2.14	<b>Мамедова Т.А., Костин В. И.</b> МОДЕЛЬ ОЦЕНКИ ЭФФЕКТИВНОСТИ РИСК-МЕНЕДЖМЕНТА В INTERNET-КОМПАНИИ	160
C2.15	<b>Ладыженский Ю.В., Попов Ю.В.</b> ИССЛЕДОВАНИЕ КОМБИНИРОВАННОГО ПРОТОКОЛА СИНХРОНИЗАЦИИ ЛОГИЧЕСКИХ ПРОЦЕССОВ ПРИ РАСПРЕДЕЛЕННОМ СОБЫТИЙНОМ МОДЕЛИРОВАНИИ ЦИФРОВЫХ УСТРОЙСТВ	165

### **Секция 3. «Моделирование сложных динамических систем»**

C3.01	<b>Бигдан В.Б., Пепеляев В.А., Сахнюк М.А.</b> О МЕТОДОЛОГО-ТЕХНОЛОГИЧЕСКИХ АСПЕКТАХ ПОСЛЕДОВАТЕЛЬНОГО ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ	170
C3.02	<b>Фельдман Л.П.</b> ОЦЕНКИ И КОНТРОЛЬ ПОГРЕШНОСТИ РЕШЕНИЯ ЗАДАЧИ КОШИ ДЛЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ПАРАЛЛЕЛЬНЫМИ РАЗНОСТНЫМИ МЕТОДАМИ.	177
C3.03	<b>Назарова И.А.</b> ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ ЧИСЛЕННОГО МОДЕЛИРОВАНИЯ СЛОЖНЫХ ДИНАМИЧЕСКИХ СИСТЕМ С СОСРЕДОТОЧЕННЫМИ ПАРАМЕТРАМИ НА ОСНОВЕ НЕЯВНЫХ МЕТОДОВ РУНГЕ-КУТТЫ	192
C3.04	<b>Дмитриева О.А.</b> ПАРАЛЛЕЛЬНЫЕ МЕТОДЫ РЕШЕНИЯ СИСТЕМ С РАЗРЕЖЕННЫМИ МАТРИЦАМИ КОЭФФИЦИЕНТОВ	196
C3.05	<b>Саух С.Е.</b> БЛОЧНЫЕ ЧИСЛЕННЫЕ ОПЕРАТОРНЫЕ МЕТОДЫ АЛГЕБРАИЗАЦИИ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ	200
C3.06	<b>Зайцев Д.А., Шмелёва Т.Р.</b> ОСНОВЫ ПОСТРОЕНИЯ ПАРАМЕТРИЧЕСКИХ МОДЕЛЕЙ ПЕТРИ КОММУТИРУЕМЫХ СЕТЕЙ	207
C3.07	<b>Селевко Г.О.</b> СЕМІОТИЧНА МОДЕЛЬ УЗАГАЛЬНЕНОГО ГЕНЕТИЧНОГО АЛГОРИТМУ ДЛЯ ПРОЕКТУВАННЯ ДИНАМІЧНИХ ОБ'ЄКТІВ	215
C3.08	<b>Коломойцева И.А.</b> АЛГОРИТМ АВТОМАТИЗИРОВАННОГО ПОСТРОЕНИЯ БАЗЫ ЗНАНИЙ НА ОСНОВЕ АНАЛИЗА ЕСТЕСТВЕННО-ЯЗЫКОВЫХ МЕДИЦИНСКИХ ТЕКСТОВ	222
C3.09	<b>Скобцов Ю.А., Родин Ю.В., Оверко В.С.</b> МОДЕЛИРОВАНИЕ И ВИЗУАЛИЗАЦИЯ ПОТОКОВ КРОВИ В СОСУДАХ С РАЗЛИЧНОЙ ПАТОЛОГИЕЙ	227
C3.10	<b>Зори А.А., Вовна А.В, Григор Е.А.</b> ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ФУНКЦИОНИРОВАНИЯ КАНАЛА ИЗМЕРЕНИЯ ТЕМПЕРАТУРЫ	233
C3.11	<b>Клевцов С.И.</b> МОДЕЛИРОВАНИЕ ГРАДУИРОВОЧНЫХ ХАРАКТЕРИСТИК ИНТЕЛЛЕКТУАЛЬНЫХ ДАТЧИКОВ ПРОСТРАНСТВЕННЫМИ ЭЛЕМЕНТАМИ	238
C3.12	<b>Рябкин Ю.В.</b> ОБОСНОВАНИЕ АЛГОРИТМА ОЦЕНИВАНИЯ ЧАСТОТНЫХ ПАРАМЕТРОВ РАДИОИМПУЛЬСОВ БОЛЬШОГО УРОВНЯ ПО ИНФОРМАТИВНЫМ СИГНАЛАМ АКУСТООПТИЧЕСКОГО СПЕКТРОАНАЛИЗАТОРА	243
C3.13	<b>Зори А.А., Савкова Е.О.</b> МОДЕЛИРОВАНИЕ ПАРАМЕТРОВ ТОНКОЙ СТРУКТУРЫ ГИДРОФИЗИЧЕСКИХ ПОЛЕЙ ПРИ ЗОНДИРОВАНИИ ВОДНОЙ СРЕДЫ	248
C3.14	<b>Киктев Н.А., Чичикало Н.И.</b> АППАРАТНАЯ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИРОВАНИЯ ЭЛЕКТРОХИМИЧЕСКОГО ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА	252
C3.15	<b>Ларина Е.Ю., Ларин В.Ю., Данько Д.А.</b> МОДЕЛИРОВАНИЕ И АППАРАТНАЯ ОРГАНИЗАЦИЯ СИСТЕМЫ КОНТРОЛЯ МАГНИТНОЙ НЕСИММЕТРИИ ЭЛЕКТРИЧЕСКОЙ МАШИНЫ	259
C3.16	<b>Солодкий А.М., Тамыев А.Н.</b> ИСПОЛЬЗОВАНИЕ МОБИЛЬНЫХ ДИАГНОСТИЧЕСКИХ СИСТЕМ ДЛЯ ВИЗУАЛЬНОГО КОНТРОЛЯ И МОДЕЛИРОВАНИЯ СЦЕНАРИЕВ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ ГАЗОПЕРЕРАБАТЫВАЮЩИХ ПРЕДПРИЯТИЙ	267
C3.17	<b>Теплинский К.С., Трубаров В.А., Святный В.А.</b> ПОДСИСТЕМА ОПТИМИЗАЦИИ НА БАЗЕ ЭВОЛЮЦИОННЫХ ВЫЧИСЛЕНИЙ ДЛЯ ПАРАЛЛЕЛЬНОЙ МОДЕЛИРУЮЩЕЙ СРЕДЫ	273
C3.18	<b>Ларин В.Ю.</b> МОДЕЛИРОВАНИЕ РАБОТЫ СХЕМЫ ИСТОЧНИКА СТАБИЛИЗИРОВАННОГО НАПРЯЖЕНИЯ ПИТАНИЯ МАГНИТОУПРУГИХ ДАТЧИКОВ НАГРУЗКИ	281

## ПРИВЕТСТВЕННОЕ СЛОВО

Донецкий национальный технический университет приветствует гостей – участников 1-й Международной научно-технической конференции «Моделирование и компьютерная графика».

Сегодня Донецкий национальный технический университет, основанный в 1921 году, крупнейший научно-технический и учебный центр, где обучается более 27 тысяч студентов и работает свыше ста двадцати профессоров, докторов наук и 650 доцентов, кандидатов наук. Широко известны научные школы ДонНТУ, авторитет которых связан с именами профессоров В.Г.Гейера, В.А.Сорокина, В.М.Клименко, В.А.Гольцова, Б.С.Панова, Л.П.Фельдмана, В.В.Пака и других известных ученых.

Без создания компьютерно-информационных систем различного назначения невозможно инновационное развитие Украины. На кафедрах университета сформировались научные направления по разработке и исследованию аппаратного, алгоритмического и программного обеспечения систем, ориентированных на моделирование сложных динамических процессов, построению САПР средств вычислительной техники и компьютерной визуализацию процессов различной природы. Здесь нашими учеными получены достаточно интересные и перспективные результаты, которые выносятся на обсуждение научной общественности.

С вниманием и благодарностью будут приняты доклады ученых, приехавших к нам, чтобы поделиться своими результатами и наработками.

Хотелось бы выразить надежду, что настоящая конференция будет полезной для всех ее участников. Приветствую Вас в стенах Донецкого национального технического университета и желаю Вам плодотворной работы и интересных встреч.

Председатель оргкомитета конференции  
Минаев А.А.

## МОДЕЛИРОВАНИЕ РЕКОНФИГУРИРУЕМЫХ КОМПЬЮТЕРНЫХ СИСТЕМ

А.В. Палагин, В.Н. Опанасенко

Институт кибернетики им. В.М. Глушкова НАН Украины

### *Анотація*

*Запропоновано підхід до проектування обчислювальних систем, побудованих на основі таких пристроїв, що реконфігуруються, який заснований на створенні добре структурованої бібліотеки методів і відповідних архітектур проектованої обчислювальної системи. Виконано формалізацію задачі оптимального синтезу структури обчислювальної системи, критерієм якості якої є сумарні апаратні витрати на реалізацію, час виконання всіх алгоритмів.*

### *Введение*

Внедрение в практику кристаллов ПЛИС и HDL — технологии [1] для выполнения проектов в этом элементном базисе интенсифицировало разработку широкого спектра цифровых модулей, представляющих собой готовые технические решения, существенно сокращающие время проектирования, моделирования, верификации и выхода на рынок новых изделий. Эти решения, называемые IP (Intellectual Property) — блоками, ядрами (Core), параметрическими модулями, могут быть настроены под конкретные требования нового проекта. Они относятся к soft cores или "мягким" заготовкам, описанным с помощью HDL-языка. Такие заготовки можно легко настраивать в соответствии с требованиями нового проекта и они, как правило, независимы от технологии изготовления ПЛИС и позволяют выполнять их моделирование и синтез с использованием инструментальных средств различных фирм. Данная работа выполнена в рамках ДНТП "Нові вітчизняні інтелектуальні комп'ютерні засоби" Министерства образования и науки Украины.

### *Постановка проблемы*

В известных методах формализованного проектирования вычислительных систем сам процесс представляется в виде последовательности этапов, на каждом из которых проект системы представлен совокупностью математических моделей, описывающих различные ее части. Модель проектируемой вычислительной системы можно представить следующей системой:

$$S = \langle M, A, B, P \rangle ,$$

где:  $M$  — множество математических методов для предметной области, лежащих в основе функционирования системы;

$A$  — множество алгоритмов реализации метода;

$B = \{b\}$  — алфавит конструктивов, из которых синтезируется структура. Алфавитом будем называть набор всех компонентов  $\{b\}$ , на основе которых строятся допустимые описание и реализация проекта;

$P$  — процедура описания проекта (описание объекта).

Таким образом, процесс проектирования состоит в решении задачи синтеза структуры на основе конструктивов  $\{b\}$  алфавита  $B$  для выполнения определенного алгоритма  $A$ , реализующего метод  $M$ , лежащий в основе функционирования структуры, в соответствии с требованиями спецификаций. Результатом процедуры  $P$  является описание проекта во входном языке САПР.

Отметим, что (в отличие от традиционных подходов) процесс разработки метода и алгоритма решения исходной задачи носит итеративный характер. Критериями эффективности искомого метода (алгоритма) являются обобщенные характеристики производительности, аппаратных затрат, точности решения задачи, сложности алгоритма, надежности проектируемой системы либо специальные критерии, такие как работа в реальном времени, трудоемкость разработки соответствующих метода и алгоритма и др.

Очевидно, сегодня представляется затруднительным предложить общую методологию оптимального решения произвольной задачи в произвольной предметной области. Более реалистичным на настоящем этапе представляется создание хорошо структурированной библиотеки методов и соответствующих архитектур проектируемой вычислительной системы, хранимых в виде программных файлов во внешней памяти базовой вычислительной подсистемы, и выбор подходящей пары (метод-архитектура) для конкретной проблемной ситуации.

Таким образом, задача оптимального синтеза сводится к задаче оптимального выбора на предварительно сформированном (и постоянно расширяемом) множестве решений из библиотеки файлов конфигураций (БФК). Этот процесс можно достаточно строго формализовать. Предложенный подход позволяет не только получить оптимальное решение задачи, но облегчить ее постановку и взаимодействие пользователя с компьютерной системой.

**Постановка задачи.** Процесс проектирования вычислительных устройств на ПЛИС полностью поддерживается инструментальными средст-

вами САПР, которые позволяют выполнять: описание проекта, логическое моделирование, размещение и трассировка проекта в заданном кристалле ПЛИС, временная верификация и формируется стандартный файл для конфигурации кристалла. Процесс проектирования основан на некотором алфавите, в качестве которого могут использоваться:

набор сложных проектных решений —  $V^4 = \bigcup_{\gamma} b_{\gamma}^4, (\gamma = 1 \div p)$ ;

набор примитивов и макроэлементов стандартной библиотеки —  $V^3 = \bigcup_z b_z^3, (z = 1 \div k)$ ;

набор макроэлементов, формируемых разработчиком —  $V^2 = \bigcup_j b_j^2, (j = 1 \div m)$ ;

набор параметрических макромодулей (LogiCore) —  $V^1 = \bigcup_i b_i^1, (i = 1 \div l)$ .

В конечном итоге конкретный проект будет использовать алфавит конструктивов  $V$ :

$$V = \bigcup_s V^s, (\forall s = 1 \div 4),$$

где часть подмножеств  $V^s$  могут являться пустыми множествами:

$$V^1 = \emptyset (\forall i = 0); V^2 = \emptyset (\forall j = 0); V^3 = \emptyset (\forall z = 0); V^4 = \emptyset (\forall \gamma = 0).$$

**Задача оптимального синтеза структуры.** Рассмотрим синтез структурной реализации последовательности алгоритмов, когда БФК содержит алгоритм в виде одного библиотечного элемента. Пусть метод/задача ( $M$ ) представляется последовательностью алгоритмов ( $A_i, \forall i = \overline{1 \div n}$ ) —  $M = \bigcup_i A_i$ .

Имеем хорошо структурированную библиотеку БФК структурных реализаций методов (алгоритмов), выполняющих отображение алгоритма в структурную реализацию ( $F: A_i \Rightarrow V_i$ ). Каждый алгоритм имеет отображение  $F: A_i \Rightarrow V_i$  в структурную реализацию ( $V_i$ ), которая представляет собой файл конфигурации для кристалла ПЛИС.

В общем случае имеется несколько вариантов реализации алгоритма (например, последовательная, последовательно параллельная и параллельная):

$$V_i = \bigcup_z B_{iz}, (z = 1 \div k),$$



Каждый вариант характеризуется параметрами быстродействия (время выполнения —  $t_{iz}$ ) и аппаратными затратами ( $q_{iz}$ ). Причем предполагаем, что мощность множества  $B$  является достаточной для реализации широкого набора алгоритмов. В том случае, если требуемая реализация  $i$ -го алгоритма в библиотеке отсутствует ( $B_i = \emptyset$ ), то необходимо с помощью инструментальных средств САПР ПЛИС создать этот проект и включить его в качестве стандартного элемента в библиотеку.

Таким образом, задача оптимизации сводится к упорядоченному назначению каждому  $i$ -му алгоритму ( $B_{iz}$ )-го элемента библиотеки для реализации задачи с целью получения экстремального значения некоторого критерия качества. Т.е. любой алгоритм отображается только одним вариантом реализации из библиотеки. В результате определяется структура, реализующая заданный метод/задачу. Тогда решение задачи может быть получено методами целочисленного математического программирования и, в зависимости от требуемого критерия качества, можно предложить следующие варианты постановки задачи оптимизации, которая состоит в определении минимума целевой функции.

1. *Критерием качества является суммарные аппаратные затраты* на реализацию всех алгоритмов:

$$Q = \sum_i \sum_z q_{iz} x_{iz} = \min, \quad (\forall i = \overline{1 \div n}, \forall z = \overline{1 \div k}),$$

$$\sum_{z=1}^k x_{iz} = 1, \quad i = \overline{1 \div n}, z = \overline{1 \div k},$$

$$it_r + \sum_i \sum_z (t_{iz} x_{iz}) \leq T_0,$$

где:  $t_r$  — время реконfigurирования кристалла ПЛИС с помощью файла конфигурации ( $t_r = \text{const}$ ) и зависит от типа кристалла ПЛИС.

$T_0$  — допустимое время выполнения всех алгоритмов.

2. *Критерием качества является суммарное время выполнения* всех алгоритмов:

$$T = \sum_i \sum_z t_{iz} x_{iz} = \min, \quad (\forall i = \overline{1 \div n}, \forall z = \overline{1 \div k}),$$

$$\sum_{z=1}^k x_{iz} = 1, \quad i = \overline{1 \div n}, z = \overline{1 \div k},$$

$$\sum_i \sum_z q_{iz} x_{iz} \leq Q_0,$$

где  $Q_0$  — допустимые аппаратные затраты.

3. *Комплексный критерий качества, учитывающий суммарное время выполнения всех алгоритмов и затраты оборудования:*

$$\alpha \sum_i \sum_z t_{iz} x_{iz} + \beta \sum_i \sum_z q_{iz} x_{iz} = \min, \quad (\forall i = \overline{1 \div n}, \forall z = \overline{1 \div k}),$$

$$\sum_{z=1}^k x_{iz} = 1, \quad i = \overline{1 \div n}, z = \overline{1 \div k}.$$

$$\sum_i \sum_z q_{iz} x_{iz} \leq Q_0,$$

$$\sum_i \sum_z t_{iz} x_{iz} \leq T_0,$$

где  $\alpha, \beta$  — весовые коэффициенты.

Методы решения таких задач достаточно хорошо разработаны и позволяют за допустимое время их решать [3].

**Выводы.** Предложенный подход к постановке задачи оптимального синтеза структуры в зависимости от требуемых критериев оптимизации позволяет, в отличие от известных подходов, решать такие задачи путем выбора подходящей пары (метод-архитектура) для конкретной проблемной ситуации на предварительно сформированном (и постоянно расширяемом) множестве решений из библиотеки файлов конфигураций (БФК).

### *Литература*

1. Палагин А.В., Опанасенко В.Н., Сахарин В.Г. Опыт проектирования цифровых устройств на базе ПЛИС с использованием HDL-технологии // УСиМ. — 2004. — №6. — С. 11–20.
2. Палагин А.В., Опанасенко В.Н., Сахарин В.Г. Вычислительные системы с реконфигурируемой (программируемой) архитектурой // Проблеми інформатизації та управління: Зб. наукових праць НАУ. — Вип. 10. — Київ, 2004. — С.5–13.
3. Сергиенко И.В., Шило В.П. Задачи дискретной оптимизации. Проблемы, методы решения, исследования. — Киев: Наук. Думка, 2003. — 261 с.

## О МОДЕЛИРОВАНИИ ВИЗУАЛЬНОГО МЫШЛЕНИЯ В КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЯХ

Ю.Р. Валькман, В.С. Быков

Международный научно-учебный центр ЮНЕСКО информационных технологий и систем НАН Украины и МОН Украины

### **Анотація**

*Розглядаються проблеми і принципи моделювання візуального мислення в комп'ютерних технологіях. Визначаються операції візуального мислення. Аналізуються відносини «абстрактне-конкретне», «візуальний-графічний образ», «загальне-приватне» і т.д. Планується розробка апарата візуального (графічного) виводу, за аналогією з дедуктивними й індуктивними виводами в логіці.*

**Введение.** Визуальное мышление характерно для процессов распознавания ситуаций, диагностики, принятия решений, планирования, управления и других творческих операций интеллектуальных систем.

Моделирование визуального мышления в компьютерных технологиях обеспечит значительное повышение эффективности соответствующих систем. Более того, мы считаем, что в настоящее время разработанные в рамках этой проблематики методы, модели и средства поставят процессы интеллектуализации систем управления на качественно новую ступень развития. В настоящее время решением соответствующих задач занимаются в большей мере психологи. Однако, современное развитие компьютерных технологий, с нашей точки зрения, позволяет некоторые процессы визуального погружать в вычислительную среду.

**Постановка проблемы.** С визуальным мышлением психологи обычно связывают творческие процессы, образное мышление. Операции мышления в настоящее время трудно моделируются в вычислительной среде. Заметим, процессы продуктивного мышления (и гештальты) Вертгеймер, в основном, демонстрировал [2] посредством операций визуального мышления. Когнитивную графику [3] также можно рассматривать как разновидность визуального мышления.

Почти полное отсутствие работ в этой предметной области и сложность проблемы приводит к необходимости на первом этапе разработки базовых принципов компьютерного моделирования визуального мышления.

**Принципы моделирования визуального мышления.** Данный термин был введен известным американским психологом Рудольфом Арнхеймом [1]: «Визуальное мышление — это мышление посредством визуальных операций». Визуальное мышление является разно-

видностью образного. Поэтому операции, характерные для образного мышления (синтез, анализ, сравнение, обобщение, классификация, абстрагирование, систематизация и распознавание образов правомерно рассматривать в качестве процедурной компоненты и визуального мышления.

Для моделирования визуального мышления мы должны уточнить данное определение в нескольких направлениях. Эти уточнения и являются принципами компьютерного моделирования визуального мышления.

1. В роли операндов в этих операциях выступают визуальные образы. Здесь мы будем говорить, в основном, о графических образах (ГО). При этом, на данном этапе исследования под ГО будем понимать графическое представление данных в векторном формате. Это обусловлено тем, что операции с видеоданными, определенными в растровом формате (AVI, MPEG и т.п.), мы не можем, пока, относить к визуальному мышлению. Поэтому, быть может лучше говорить о графическом мышлении.

2. Необходимо конкретизировать визуальные операции. В настоящее время мы рассматриваем процессы:

- *трансформации (модификации) ГО;*
- *интеграции, обобщения ГО (объединения нескольких ГО);*
- *детализации ГО (фрагментации их, выделения в данном ГО отдельных, локальных образов).*

При этом, в операциях трансформации можно выделить три разновидности: преобразование данного ГО, его модификация с использованием других ГО и с использованием информации, не представленной в форме ГО.

3. При моделировании визуального мышления в вычислительной среде (и не только) необходимо выделить специальный класс операций — конвертации данных, информации, знаний (мыслей), определенных в различных форматах (табличном, аналитическом, текстовом и т.д.) в форму ГО.

Операция «первичного» синтеза ГО часто носит творческий характер, и, скорее, относится к искусству, чем к технологии или науке. Заметим, в искусственном интеллекте есть целое направление, называемое «переформулированием». Его суть заключается в поиске таких визуальных средств представления условий задачи, которые сразу приводят к ее решению (без аналитических преобразований). Это направление близко проблематике и методам когнитивной графики.

4. Представляется вполне очевидным, что невозможно создание компьютерных технологий моделирования визуального мышления для всех предметных областей. Поэтому, любая компьютерная систе-

ма моделирования визуального мышления должна проблемно-ориентированной.

5. Чтобы «передать компьютеру» хотя бы некоторые операции визуального мышления, необходимо обеспечить его соответствующей базой знаний. Как обычно, в этой БЗ выделяются декларативная и процедурная компоненты.

В «декларативной БЗ» хранятся, систематизируются и структурируются ГО с их вербальной интерпретацией. В рамках этой БЗ представляет интерес многоуровневая система графемных (неделим на соответствующем уровне) конструкции.

Заметим, что структура БЗ ГО значительно сложнее традиционных БЗ. Это обусловлено тем, что между локальными и/или интегрированными ГО отношения более многоаспектны, т.к. в них описываются помимо классических связей (род–вид, целое–часть, причина–следствие и т.д.) еще и пространственные отношения местоположения одних ГО в других.

В «процедурной БЗ» выделяются две компоненты: стандартные операции синтеза-анализа ГО и специальные правила вывода. В специальных правилах вывода накапливается и систематизируется опыт эффективного использования некоторых операций визуального мышления в данной проблемной области.

6. Некоторые из этих операций, имеющие стереотипный характер, должны быть представлены в процедурной компоненте БЗ в форме продукционных систем.

Эта компонента в традиционных БЗ интеллектуальных систем называется подсистемой логического вывода. В ней отражаются все стереотипные операции вывода моделируемой предметной области, носящие характер специальных и общих закономерностей (синтеза, анализа, обобщения, детализации и т.д. для образного мышления).

В дальнейшем целесообразно говорить и о подсистеме визуального, графического вывода и даже образного вывода, поскольку здесь дедуктивные (иногда и индуктивные) цепочки вывода осуществляются на ГО или гибридно (с привлечением выводов на вербальных признаках — силлогизмах).

В этой подсистеме должны отражаться и общие закономерности мышления: константность восприятия (независимость интерпретации объекта от его величины, расстояния от него, отражаемой в ГО проекции), неразрывность операций синтеза–анализа (обобщенность процедур мышления), определения по части (проекции ГО объекта) целого и т.д. С помощью такой БЗ становится возможным моделировать как левополушарное, так и правополушарное мышления. Как следствие можно говорить о моделировании понимания.

Таким образом, БЗ визуального мышления становится в большей степени похожа на онтологию моделируемой предметной области, чем традиционные интеллектуальные системы. Вспомним тезис Аристотеля о том, что онтология должна быть внеязыковой системой.

7. Форм и форматов представления ГО в памяти компьютеров в настоящее время наработано весьма много, и они продолжают разрабатываться не только общих, но и специальных, например, для хранения карт в геоинформационных системах. Поэтому, одним из базовых принципов создания БЗ визуального мышления является “*принцип разумного ограничения*”. Формы и форматы ГО в БЗ должны ориентироваться на операции их обработки и использования.

Особой операцией в моделировании образного мышления является процедура оцифровки трехмерных сцен (объектов, интерьеров) в робототехнике, компьютерных играх и т.д.

Вообще посредством ГО можно представлять самые различные данные, информацию, знания. Так, например, графики аналитических зависимостей в математике, диаграммы таблично организованных данных, блок-схемы работы различных устройств, мнемосхемы, графические образы семантических сетей и т.д. Более того, мы можем строить самые различные условные графические представления своих мыслей — графические метафоры.

Поэтому второе использование принципа разумного ограничения заключается в определении требований моделируемой предметной области к видам и разновидностям графических образов.

8. Поскольку визуальное мышление носит параллельно-пространственный характер (а не линейно-последовательный), то и редактирование (модификация компонентов и структур) информации в БЗ значительно сложнее, чем в традиционных БЗ. Вообще структуры БЗ визуального мышления более динамичны. И, конечно, эта БЗ по определению является открытой системой. Замкнутость, дедуктивность, аксиоматичность характерна лишь для некоторых ее фрагментов.

9. В БЗ ГО также поддерживаются обычно несколько уровней графемных (неделимых) конструкций. Например: для UML-диаграмм — это условные диаграммы объектов и отношений между ними, для других языков — это изображения базового алфавита знаков. Или, графемы «/», «\» и «-» являются строительным материалом (1-й уровень) буквы «А». Буква «А» — один из модулей (2-го уровня) для построения слов, из слов (3-й уровень) синтезируются предложения и т.д. И, поскольку целостность любого ГО определяется возможностью

проведения над ними некоторых операций, то на втором уровне буква «А» не делится на фрагменты.

10. Но, помимо графической компоненты в БЗ визуального мышления желательно (а часть и необходимо) поддерживать БЗ семантической интерпретации ГО. Эта информация представляет собой вербальные характеристики (структуры признаков) ГО.

В некотором отношении в БЗ интерпретации копирует структуру БЗ ГО, но могут быть ГО без вербальной трактовки, вербальные описания без графической интерпретации. Может одному ГО соответствовать несколько интерпретаций (в зависимости от контекста) и/или одному вербальному элементу — несколько ГО.

Сложность отношений «ГО — вербальные характеристики» можно назвать «проблемой мультikonтекстности»: «один ГО → много контекстов», «один текст → много ГО». Можно привести множество примеров: как изобразить понятия мебель, оружие, чувство, стол (обеденный, рабочий, журнальный и т.п.). Эти проблемы близки к операциям обобщения, абстрагирования, классификации.

Поэтому весьма сложны процедуры именования (идентификации синтеза вербальных структур) ГО и, наоборот, построения ГО по вербальным характеристикам.

В докладе приводится множество примеров визуального мышления, когнитивной графики, эффективных графических образов (и их систем) с детальным анализом их специфики, выразительности и функциональности.

**Выводы.** На основе принципов, изложенных в докладе, в Международном научно-учебном центре информационных технологий и систем разрабатываются теоретико-методологические основы моделирования визуального мышления.

Далее, на основе апробированных принципов и методов будет разрабатываться система управления базами знаний для моделирования визуального мышления.

### **Литература**

1. Арнхейм Р. Визуальное мышление // Зрительные образы: феноменология и эксперимент. Ч. 2. — Душанбе, 1973.
2. Вертгеймер М. Продуктивное мышление — М.: Прогресс, 1987.
3. Зенкин А.А. Когнитивная компьютерная графика. — М.: Наука, 1991.

## **ПОРТАЛ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ: ЦЕЛИ, ЗАДАЧИ И ОСОБЕННОСТИ ОРГАНИЗАЦИИ**

А.Я. Аноприенко, Е.А. Башков, Т.А. Самойлова  
Донецкий национальный технический университет

### ***Анотація***

*В доповіді обґрунтовується необхідність спеціального web-порталу комп'ютерного моделювання і розглядаються основні особливості його організації. В якості прикладу типового модуля порталу представлено його початковий розділ, присвячений археомоделюванню.*

Компьютерное моделирование в ДонНТУ как приоритетное направление исследований и разработок восходит еще к временам аналоговых вычислительных машин, работа с которыми началась в вузе в начале 60-х годов. Возможности сетевых технологий в исследованиях по моделированию динамических процессов впервые начали использоваться с начала 90-х годов [1]. В настоящее время акцент в такого рода исследованиях перенесен на максимальное использование возможностей web-технологий, в частности разработана и успешно апробирована концепция WEB-ориентированной среды для интеграции моделирующих, вычислительных и информационных сервисов [2]. В связи с этим возникла также и необходимость в создании специального информационного web-портала, который в перспективе мог бы также стать и основой для формирования соответствующей среды для интеграции моделирующих сервисов.

Опыт успешного создания такого рода порталов с активным привлечением студентов старших курсов накоплен в ДонНТУ при разработке и эксплуатации портала магистров ДонНТУ [3], который начиная с 2000 года объединяет персональные тематические страницы магистров практически всех специальностей университета. В настоящее время на портале размещено более тысячи оригинальных тематических сайтов научно-образовательного направления, благодаря чему данный портал вышел в число лидеров по посещаемости среди всех ресурсов такого на постсоветском пространстве [4].

Целесообразность создания специализированного портала компьютерного моделирования связана также и с тем, что в настоящее время наблюдается определенный дефицит такого рода информационных источников. Из существующих в стране ресурсов



можно выделить только сайт Центра имитационного моделирования НТУУ «КПИ» [5], а также ряд российских ресурсов, таких как «GPSS World» [6] и «Математическое моделирование в естественнонаучных дисциплинах» [7]. Однако, имеющиеся на сегодня ресурсы, как правило, или имеют узкоспециализированный характер (как, например, [6]), или не в полной мере используют потенциал современных сетевых и компьютерных технологий. В любом случае, следует признать актуальность создания такого ресурса по компьютерному моделированию, который, с одной стороны, в максимальной степени учитывал бы потребности исследовательской работы и учебного процесса в высшей школе, а с другой – охватывал бы максимально широкий круг вопросов компьютерного моделирования и ориентировался на самые различные современные компьютерные технологии.

В связи с этим в текущем году в ДонНТУ было инициировано создание специального портала компьютерного моделирования [8], целью которого является концентрация и стимулирование различного рода исследований и разработок в данном направлении. При этом предполагается, что с помощью данного портала будут решаться следующие основные задачи:

- концентрация и систематизация информации, посвященной прошлому, настоящему и будущему компьютерного моделирования;
- размещение различных оригинальных средств компьютерного моделирования, предоставление доступа к ним, организация их апробации и документирование опыта эксплуатации;
- организация коллективных проектов в области компьютерного моделирования и моделирующих сервисов различного назначения;
- предоставление систематизированного доступа к основным ресурсам всемирной сети, связанным с компьютерным моделированием;
- обеспечить возможность привлечения и координации взаимодействия широкого круга ученых, аспирантов и студентов к разработкам и исследованиям в области компьютерного моделирования.

Естественно, что предполагается постепенное расширение круга решаемых задач и, соответственно, функциональности, портала.

К числу главных особенностей организации портала следует отнести его мультиязычность и существенную модульность структуры, что предполагает возможность относительно независимой разработки, эволюции и текущей эксплуатации отдельных разделов, что позволит в перспективе привлечь к участию в развитии ресурса достаточно широкий круг ученых, специалистов и студентов.

Первым реализованным модулем такого рода стал раздел «Археомоделирование», введенный в эксплуатацию еще в первой половине 2005 года. Выбор именно данного модуля в качестве начального имеет определенный символический характер, смысл которого заключается в наглядной демонстрации того, что современное компьютерное моделирование имеет чрезвычайно глубокие цивилизационные корни и является естественным продолжением пока еще малоизвестной, но чрезвычайно важной и интересной традиции докомпьютерной эволюции вычислительного моделирования.

Основой данного раздела стали исследования, проведенные в конце 90-х годов и впервые опубликованные в работах [9-12]. В начальном варианте в разделе на двух языках представлена информация о трех известных артефактах, самому древнему из которых почти 20 тысяч лет, а самому «молодому» почти 2 тысячи лет (рис. 1).

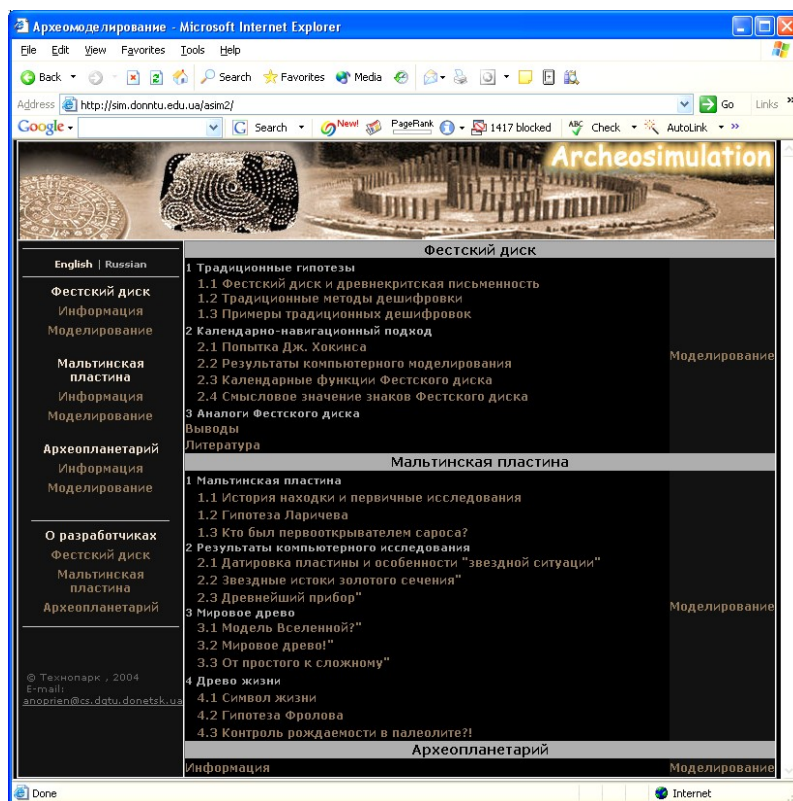


Рисунок 1 – Главная страница раздела «Археомоделирование»

На сайте впервые представлены не только интерпретации данных артефактов как специфических вычислительных моделей, но и представлены их действующие модели, реализованные на базе flash-технологии фирмы Macromedia (рис. 2).

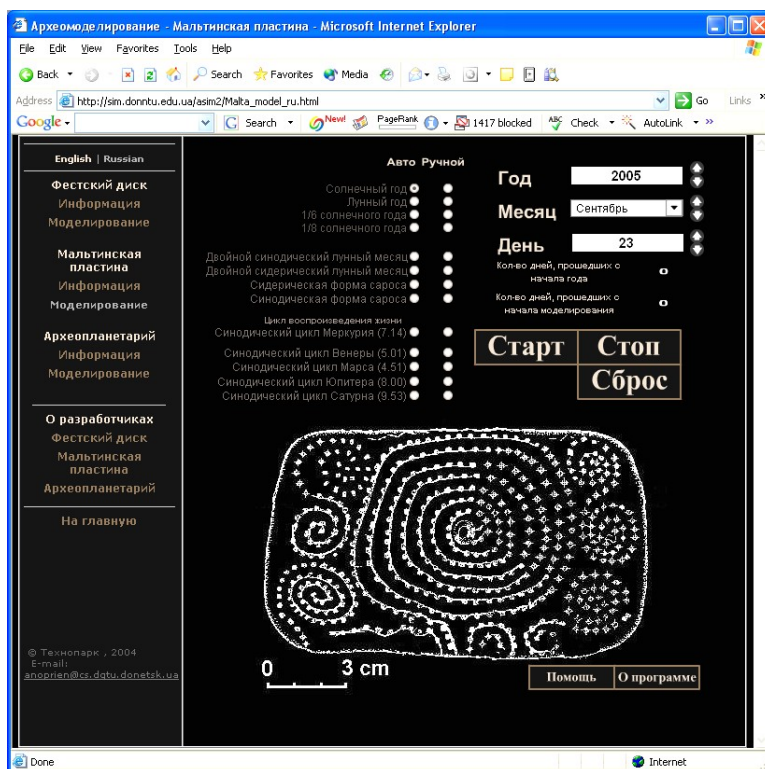


Рисунок 2 – Пример страницы моделирования: модельная интерпретация т.н. «Мальтинской пластины» - древнейшего из известных на сегодня модельных артефактов.

В представленных моделях имеется возможность как автоматического, так и ручного управления процессом вычислительного моделирования в различных режимах. Все три модели реализованы в едином стиле, что в общем случае является не обязательным для портала в целом, но для раздела, являющегося своего рода основополагающим началом, было признано целесообразным и необходимым.

В целом предполагается также, что одним из основных стимулов и источников развития портала в дальнейшем станут

международные научно-технические конференции по моделированию и компьютерной графике в ДонНТУ, начало которым положено в 2005 году.

#### Литература:

1. Святный В.А., Анопrienко А.Я. Опыт реализации системы моделирования динамических процессов на параллельной ЭВМ в среде сетевого графического интерфейса // Тезисы докладов Всесоюзной научно-технической конференции “Перспективы развития и применения средств вычислительной техники для моделирования и автоматизированного исследования”. - Москва. - 1991. - С. 190-191.
2. Svjatnyj V., Anoprijenko A., Potapenko V., Zabrovsky S. The universal WEB-based distributed environment for simulation services integration // EUROSIM 2004: 5th EUROSIM Congress on Modeling and Simulation. 06–10 September 2004. ESIEE Paris, Marne la Vallée, France. Book of abstracts. S. 63-64.
3. Портал магистров ДонНТУ, <http://masters.donntu.edu.ua>
4. Minaev A., Bashkov E., Anoprienko A. Master’s Portal as the Practical Result of Studying Web-technologies // Proceedings of the International Conference on Engineering Education ICEE 2005 “Global Education Interlink”, Silesian University of Technology. – Gliwice, 2005, Vol. I. S. 727-732.
5. Имитационное моделирование систем. Центр имитационного моделирования НТУУ «КПИ» <http://simulation.org.ua>
6. GPSS World: Имитационное моделирование на языке GPSS <http://www.gpss.ru>
7. Математическое моделирование в естественнонаучных дисциплинах <http://mathmod.exponenta.ru/>.
8. Портал компьютерного моделирования <http://sim.donntu.edu.ua>
9. Anoprienko A. Interpretation of some artefacts as special simulation tools and environments / “Short Papers Proceedings of the 1997 European Simulation Multiconference ESM'97. Istanbul, June 1-4, 1997” - Istanbul, SCS, 1997, p. 23-26
10. Анопrienко А.Я. От вычислений к пониманию: когнитивное компьютерное моделирование и опыт его практического применения на примере решения проблемы Фестского диска // Научные труды Донецкого государственного технического университета. Выпуск 6. Серия “Информатика, кибернетика и вычислительная техника” (ИКВТ-99). - Донецк: ДонГТУ. - 1999. - С. 36-47.
11. Анопrienко А. Я. Восхождение интеллекта: эволюция монокодовых вычислительных моделей // Научные труды Донецкого государственного технического университета. Выпуск 15. Серия “Информатика, кибернетика и вычислительная техника» (ИКВТ-2000). - Донецк: ДонГТУ. – 2000. - С. 36-47
12. Anoprijenko A. The early history of simulation in Europe: scale planetariums and astromorphic models // EUROSIM 2004: 5th EUROSIM Congress on Modeling and Simulation. 06–10 September 2004. ESIEE Paris, Marne la Vallée, France. Book of abstracts. S. 146-147.

## КОМПЬЮТЕРНЫЙ СТЕНД ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ОПЕРАТОРСКОЙ ДЕЯТЕЛЬНОСТИ

В.Г. Ли

Таганрогский государственный радиотехнический университет

### ***Анотація***

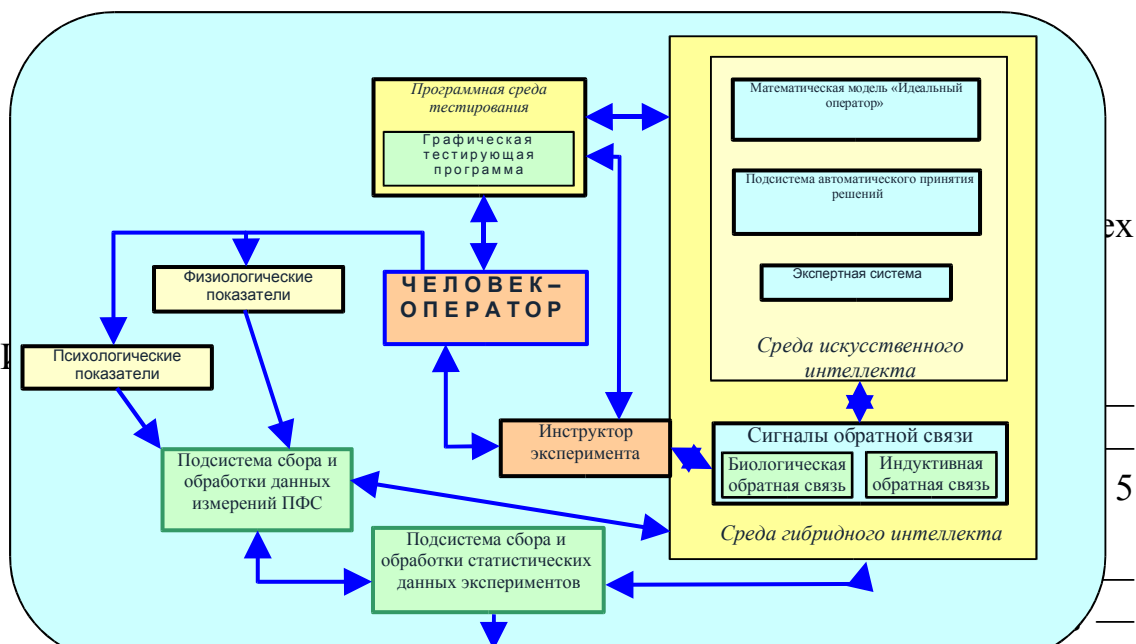
*Стаття присвячена розробці програмного й апаратного забезпечення розподіленого імітаційного моделювання діяльності людини-оператора на багатомашинному комп'ютерному стенді. Стенд призначений для проведення експериментальних досліджень надійності й ефективності діяльності оператора.*

Актуальность изучения психофизиологических состояний (ПФС) определяется их вкладом в обеспечение эффективности деятельности и надежности человека, а также увеличением количества профессий и изменением условий труда, когда психофизиологические возможности человека становятся определяющими. Данные психофизиологических исследований в сочетании с психологическими методами обусловили исследования ПФС, выделив их в отдельный предмет исследования.

На рисунке 1 представлена структурно-функциональная схема стенда, обеспечивающего измерение параметров ПФС человека-оператора в процессе исполнения профессиональной деятельности. Стенд представляет собой человеко-машинный комплекс с элементами искусственного и гибридного интеллекта.

***Основные функциональные параметры тестирующей графической программы:***

- Количество цветов (по умолчанию — 7 цветов радуги)
- Количество форм фигур (по умолчанию — 4 формы: пирамида, сфера, цилиндр, куб)
- Подложка (например, звездное небо; небо/земля с линией горизонта)
- Частота порождения целей в сеансе (от 5 в сек. до 30 в сек.)
- Координаты точек на граничной сфере (порождения) целей задаются случайным образом
- Координаты точек на граничной сфере исчезновения целей задаются случайным образом на противоположной (относительно точки порождения) половине граничной сферы сцены
- Звуковое сопровождение каждой из целей начиная с момента порождения и заканчивая моментом ее исчезновения (или уничтожения) различное по частоте для каждой из 21 типов целей. Сила звука зависит от дистанции наблюдения



К подсистеме формирования моделей деятельности оператора человеко-машинных систем

- Скорость движения целей: низкая (5–15 м/сек); средняя 15–35 м/сек); высокая (35–50 м/сек)
- Частота порождения целей (в сек): низкая — до 5; средняя — до 15; высокая — до 30
- Время сеанса: короткое (1–2 мин. — 1 тестирующий цикл); среднее (8–10 мин. — до 4–5 тестирующих циклов с условными переходами на другие уровни сложности и мотивации); большое — свыше 5 тестирующих циклов с условными переходами на другие уровни сложности и мотивации); заведомо неограниченное с наивысшей степенью текущей и отложенной мотивации
- Указатель задаваемого диапазона испытания по цвету
- Указатель задаваемого диапазона испытания по форме
- Маркер текущего состояния испытания по цвету (включен/выключен)
- Маркер текущего состояния испытания по форме (включен/выключен)
- Помехи:
  - ❖ визуальные: вибрации изображения; качка изображения; затемнение экрана; туман, снег и т.п.; отключение функции отображения факта уничтожения цели.
  - ❖ звуковые: отключение сигнала о факте уничтожения цели; отключение звукового сопровождения целей; возможность голосового управления прицелом и командой уничтожения.

- ❖ функциональные: запаздывание срабатывания сигнала о факте уничтожения цели; уменьшение размеров прицела (точности требуемого позиционирования); отключение маркера текущего результата по цвету; отключение маркера текущего результата по форме; отключение таймера; отключение табло текущего результата; уменьшение геометрических размеров объектов-целей.
- Структура рабочего окна программы представлена на рисунке 2.

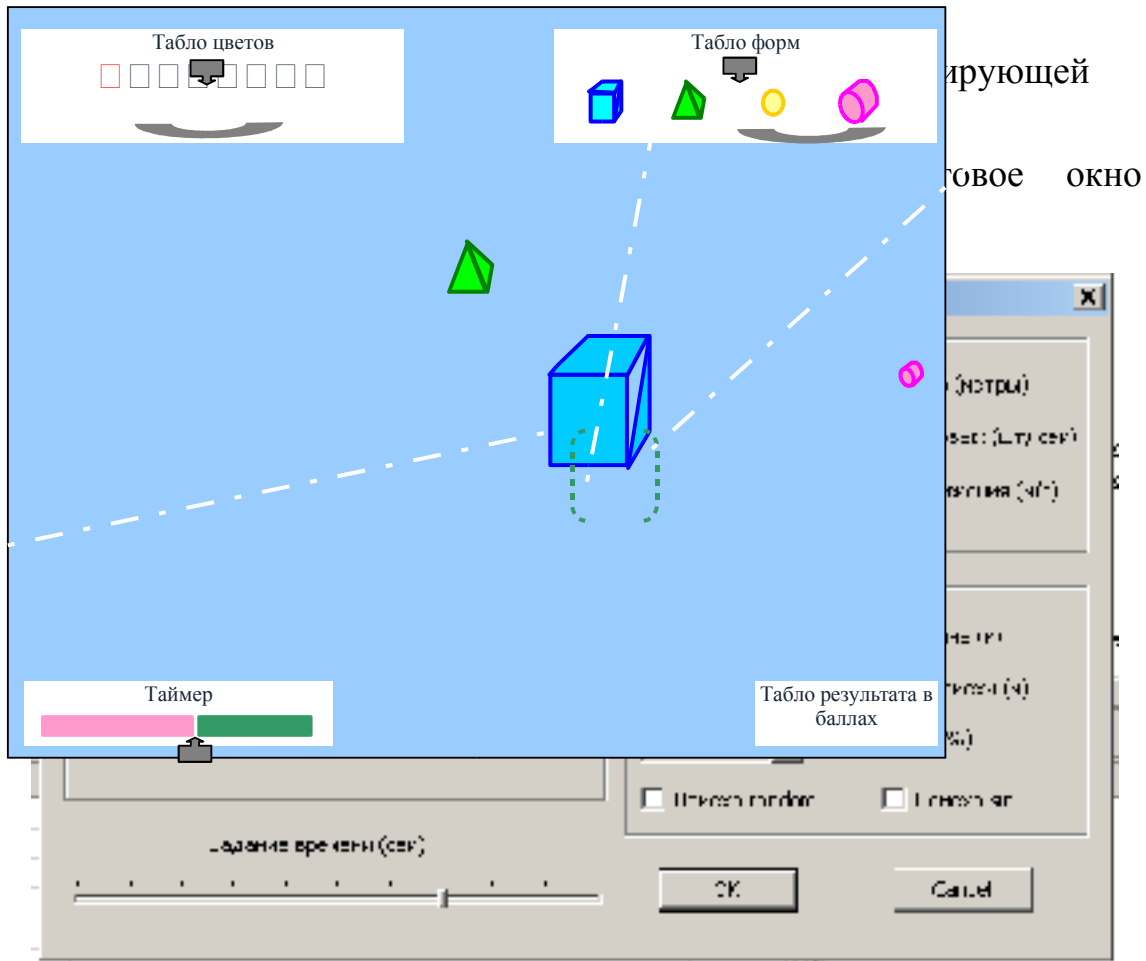


Рисунок 3 — Окно «Диалог» для ввода исходных данных

**Стратегия деятельности оператора** в среде предложенной тестирующей программы определяется главной целью — к моменту запланированного окончания сеанса добиться наименьшего отклонения верхних маркеров на табло цветов и табло форм целей от срединных линий заданных в сеансе на этих табло интервалов.

**Итоговая интегральная оценка деятельности (ИОД)** оператора определяется в баллах и вычисляется нарастающим итогом по формуле

$$ИОД = \sqrt{PЦ^{(ОЦ)} \times P\Phi^{(ОФ)}},$$

где:  $PЦ$  — результат деятельности на табло цветов целей (в баллах);  
 $OЦ$  — индекс отклонения маркера от срединной линии табло цветов (показатель степени). Индекс — положительное число в интервале от 0,0 до 1,0;

$PФ$  — результат деятельности на табло форм целей (в баллах);  
 $OФ$  — индекс отклонения маркера от срединной линии табло форм (показатель степени). Индекс — положительное число в интервале от 0,0 до 1,0.

Результаты  $PЦ$  и  $PФ$  определяются по формулам

$$PЦ = (KK_{д1} \times KK_{Hi} \times 10 \times N_1 + KK_{д2} \times KK_{Hi} \times 3 \times N_2 - (1 - KK_{д3}) \times (1 - KK_{Hi}) \times 5 \times N_3 + KK_{д4} \times KK_{Hi} \times 100 \times N_4) \times KK_{ц} \times KK_{с} \times KK_{л},$$

$$PФ = (KK_{д1} \times KK_{Hi} \times 10 \times N_1 + KK_{д2} \times KK_{Hi} \times 3 \times N_2 - (1 - KK_{д3}) \times (1 - KK_{Hi}) \times 5 \times N_3 + KK_{д4} \times KK_{Hi} \times 100 \times N_4) \times KK_{ф} \times KK_{с} \times KK_{л},$$

где:  $KK_{дi} = (T_{oi} - T_{vi}) / T_{oi}$  — корректирующие коэффициенты, отражающие скорость реакции оператора при уничтожении каждой из целей; действительные числа в интервале от 0,0 до 1,0. Чем раньше сбивается цель, тем больше величина коэффициента.

$T_o$  — общее время возможного наблюдения цели в рабочем окне оператором;  $T_v$  — интервал времени от начала наблюдения до момента уничтожения цели;

$KK_{Hi} = (N_{oi} - N_{vi}) / N_{oi}$  — корректирующие коэффициенты, отражающие насыщенность текущей сцены; действительные число в интервале от 0,0 до 1,0.  $N_o$  — общее количество наблюдаемых целей;  $N_{л}$  — суммарное количество целей, подлежащих уничтожению по заданным признакам цвета и формы;

$10 \times N_1$  — «цена» правильно сбитых целей, т.е. если они относятся к обоим заданным интервалам и по цвету и по форме ( $N_1$  — количество таких целей);

$3 \times N_2$  — «цена» частично правильно сбитых целей, т.е. если они относятся только к одному заданному интервалу ( $N_2$  — количество таких целей);

$5 \times N_3$  — «цена» неправильно сбитых целей, т.е. если они не относятся ни к одному из заданных интервалов и по цвету и по форме ( $N_3$  — количество таких целей);

$100 \times N_4$  — «цена» сбитых целей-«убийц», независимо от того, относятся ли они к заданным интервалам ( $N_4$  — количество таких целей). Количество таких целей не учитывается ни в одной из величин  $N_1, N_2, N_3$ );

— корректирующие коэффициенты  $KK_{ц}$  (по цвету) и  $KK_{ф}$  (по форме), отражающие сложность сценария в зависимости от длины



задаваемых интервалов на табло цветов и форм целей. Значения принимаются эвристически (по аналогии с законом нормального распределения) в табличной форме в виде

а) для табло цветов:

- при длине задаваемого интервала — 1 цвет,  $KK_{ц} = 0,4$ ;
- при длине задаваемого интервала — 2 цвета,  $KK_{ц} = 0,6$ ;
- при длине задаваемого интервала — 3 цвета,  $KK_{ц} = 0,8$ ;
- при длине задаваемого интервала — 4 цвета,  $KK_{ц} = 0,9$ ;
- при длине задаваемого интервала — 5 цветов,  $KK_{ц} = 0,8$ ;
- при длине задаваемого интервала — 6 цветов,  $KK_{ц} = 0,6$ ;
- при длине задаваемого интервала — все 7 цветов,  $KK_{ц} = 0,3$ ;

б) для табло форм:

- при длине задаваемого интервала — 1 форма,  $KK_{ф} = 0,4$ ;
- при длине задаваемого интервала — 2 формы,  $KK_{ф} = 0,9$ ;
- при длине задаваемого интервала — 3 формы,  $KK_{ф} = 0,7$ ;
- при длине задаваемого интервала — все 4 формы,  $KK_{ф} = 0,3$ ;

–  $KK_{с}$ , отражающие сложность сценария в зависимости от задаваемой скорости движения целей — положительные относительные числа (отношение заданной скорости к максимально возможной) в интервале от 0,0 до 1,0. В настоящей версии программы скорость может задаваться на интервале от 2 до 30 м/сек.);

–  $KK_{п}$  — коэффициенты, отражающие сложность сценария в зависимости от наличия помех; целое положительное число, соответствующее количеству задействованных в текущем сценарии помех.

На основе ИОД может быть определена относительная оценка (ООД) (в процентах) деятельности оператора по формуле

$$ООД = (ИОД/ОДИО) \times 100\%,$$

где ОДИО — максимально возможное (теоретически) количество баллов, которое могло быть получено, если бы деятельность оператора выполняла математическая модель «идеального оператора».

Помимо основной оценки в виде ИОД, может быть количественно определена косвенная **оценка эффективности деятельности** (ОЭД), которая также вычисляется в %-тах нарастающим итогом по формуле

$$ОЭД = \sqrt{\{1 - [(N_{\text{всего}} - N_{\text{заплан}}) / N_{\text{всего}}]\}} \times \{1 - [(N_{\text{заплан}} - N_{\text{сбитых}}) / N_{\text{заплан}}]\}} \times 100\%,$$

где:  $N_{\text{всего}}$  — общее количество наблюдаемых целей, порожденных программой к текущему моменту оценки;

$N_{\text{заплан}}$  — количество порожденных программой целей, которые

должны быть уничтожены в соответствии с заданием на сеанс (включая цели-«убийцы»);

$N_{сбитых}$  — количество правильно «сбитых» оператором целей (включая цели-«убийцы»).

К примеру, пусть:  $N_{всего}=200$  целей;  $N_{заплан}=50$  целей;  $N_{сбитых}=20$  целей, тогда  $ОЭД = \sqrt{(1-((200-50)/200)) \times (1-((50-20)/50))} \times 100\% = 32\%$ . При тех же параметрах сценария, но при  $N_{сбитых}=40$  целей, имеем эффективность **45%**.

Если бы эти же результаты были показаны оператором при  $N_{всего}=100$  целей, то результаты соответственно имели оценки **45%** и **63%**.

## ГРАФИЧЕСКАЯ ПОДГОТОВКА СТУДЕНТОВ СПЕЦИАЛЬНОСТИ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Е.А. Башков, В.В. Карабчевский

Донецкий национальный технический университет

### *Анотація*

*Студенти засвоюють методи і засоби створення двовимірних і тривимірних моделей, відповідний інструментарій AutoCAD. Особлива увага приділяється створенню параметризованих геометричних моделей, як за допомогою мови AutoLISP, так і шляхом розробки графічних додатків з самостійною реалізацією операцій над тривимірними полігональними моделями.*

В учебном плане специальности «Программное обеспечение автоматизированных систем» (ПО) важное место занимает подготовка в области компьютерной графики и геометрического моделирования. Это обусловлено ролью, которую играет компьютерная графика в современном программировании, САПР, мультимедийных системах, игровых приложениях. Графическая подготовка студентов специальности ПО обеспечивается несколькими учебными дисциплинами.

Двумерные построения в экранной системе координат и средства спрайтовой анимации изучаются в курсе «Объектно-ориентированное программирование». Курсовая работа по этой дисциплине предусматривает обязательное использование этих средств при выполнении задания.

Средства матричной алгебры, необходимые для изучения основ аналитической геометрии, рассматриваются в курсе «Высшая математика».

Программа курса «Инженерная и компьютерная графика» предусматривает изучение основ начертательной геометрии и инструментальных средств компьютерного черчения; курс направлен на формирование у студентов представлений о соотношении между геометрическими объектами в пространстве и их изображениями на плоскости и должен способствовать развитию пространственного воображения и навыков логического мышления при изучении геометрических моделей (рис. 1). Практическая часть курса предусматривает решение задач классическими методами, но инструментом для выполнения соответствующих лабораторных работ является система AutoCAD (рис. 2). В курсе изучаются ортогональные проекции геометрических фигур, методы решения

основных позиционных и метрических задач, многогранные и линейчатые поверхности, нелинейчатые кривые поверхности, задание точек и линий на поверхностях, методы преобразования чертежа, развертки, сечения поверхностей плоскостью, отыскание точек пересечения прямой и поверхности, построение линий пересечения поверхностей.

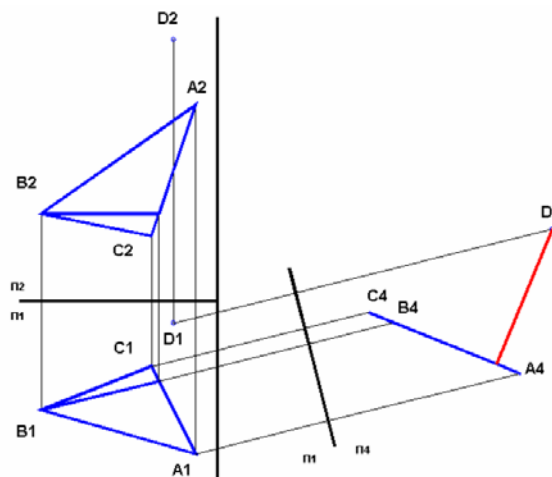
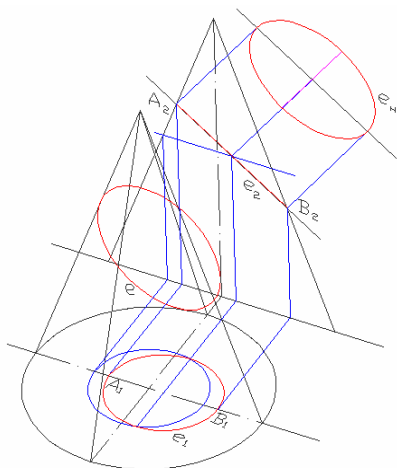


Рисунок 1 — Сечение конуса

Рисунок 2 — Отыскание расстояния от точки до плоскости

Студентам предлагается изданный на компакт-диске учебно-методический комплекс по курсу «Инженерная и компьютерная графика».

Состав комплекса:

1. Конспект лекций по курсу «Инженерная и компьютерная графика»;
2. Учебно-методическое пособие для выполнения лабораторных работ по курсу «Инженерная и компьютерная графика»;
3. Электронный учебник по курсу «Инженерная и компьютерная графика», содержащий теоретические материалы, рекомендации по выполнению лабораторных работ в системе AutoCAD и динамические модели геометрических объектов и процессов решения задач (рис. 3). Учебник разработан с применением средств HTML [1];
4. Система, позволяющая выполнять построения на эюре Монжа и просматривать их результаты в аксонометрии (рис. 4). Система предназначена для повышения наглядности построений и, в перспективе, может заменить AutoCAD в качестве среды для выполнения лабораторных работ [2] – [3].

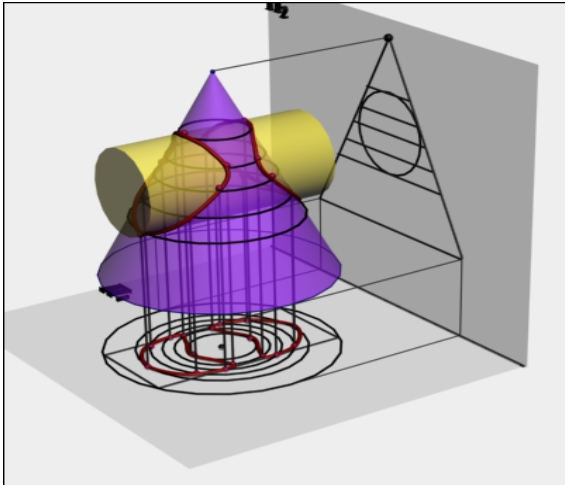


Рисунок 3 — Пересечение конуса и цилиндра

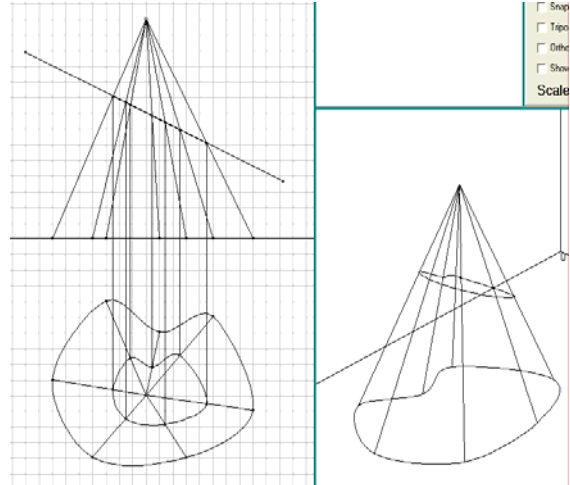


Рисунок 4 — Сечение конической поверхности

Такой подход к содержанию курса позволяет использовать начертательную геометрию как основу для изучения методов и алгоритмов геометрического моделирования и визуализации трехмерных моделей, которые изучаются в курсе «Графическое и геометрическое моделирование и интерактивные системы». Предварительное ознакомление с AutoCAD позволяет быстро изучить средства поверхностного и твердотельного моделирования этой системы (рис. 5), освоить методы создания параметризованных моделей с применением AutoLISP (рис. 6).

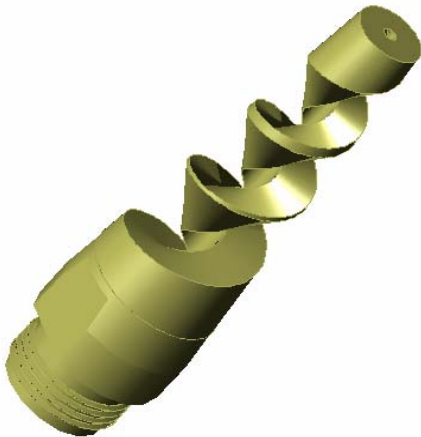


Рисунок 5 — Модель детали, содержащая винтовые поверхности

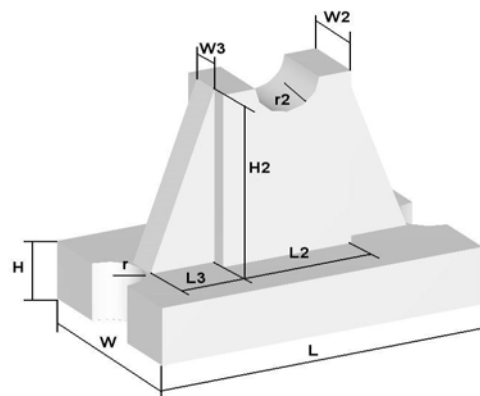


Рисунок 6 — Параметризованная твердотельная модель

При изучении этой дисциплины студенты выполняют курсовой проект, стандартное задание состоит в разработке графического

редактора для создания и визуализации параметризованных моделей заданного вида, в ходе разработки могут быть использованы как средства Open GL и Direct 3D, так и инструментарий низкого уровня.

В курсе «Компьютерный синтез и обработка изображений» изучаются математические и алгоритмические основы построения систем синтеза и обработки изображений. Рассматриваются различные способы представления объектов сцены, средства описания поверхностей, изучаются основные понятия и особенности цветового восприятия, различные цветовые пространства, модели освещенности и функции закрашки. В разделе «Обработка изображений» освещаются основные моменты, связанные с хранением и воспроизведением изображений, их обработкой (псевдотонирование, оптимизация палитры и др.), а также наиболее популярные алгоритмы сжатия изображений.

Лекции по всем этим курсам читаются с применением оборудования, которое обеспечивает передачу изображения с лекторского компьютера или телекамеры на мониторы, размещенные в аудитории. Это позволяет использовать презентации, а также демонстрировать работу различных графических приложений.

Научно-исследовательская работа многих студентов связана с созданием трехмерных моделей различных объектов, новых алгоритмов и программных средств геометрического моделирования, обучающих систем, средств поиска, распознавания и синтеза изображений, разработкой игровых приложений.

Такой уровень графической подготовки студентов специальности ПО позволяет им выступать как в роли квалифицированных пользователей современных систем геометрического моделирования, так и в качестве разработчиков специализированных приложений, что повышает их конкурентоспособность на рынке труда.

### *Литература*

1. Карабчевский В.В. Электронный учебник по курсу инженерная графика// Труды конференции «Информационные технологии в науке, образовании, телекоммуникации, бизнесе». Ялта-Гурзуф, 20–30 мая 2002 года. — С. 285–287.
2. Карабчевский В.В. Повышение качества преподавания инженерной графики путем разработки и применения обучающих систем. Научные труды Донецкого государственного технического университета. Серия: Информатика, кибернетика и вычислительная техника, (ИКВТ-99) выпуск 6: — Донецк: ДонГТУ, 1999. — С. 294–299.
3. Карабчевський В.В. Комп'ютерні технології в викладанні інженерної графіки. Прикладна геометрія та інженерна графіка. Праці Таврійської державної агротехнічної академії. — Вип. 4, т. 14. — Мелітополь: ТДАТА, 2001. — С. 83–90.

УДК 517 11  
УДК 53(075 8)

## ИСПОЛЬЗОВАНИЕ КОМПЬЮТЕРНОЙ ГРАФИКИ ПРИ ИЗУЧЕНИИ ФИЗИКИ

Е.Е. Гетманова

Харьковский национальный университет радиоэлектроники  
Харьков, Украина

### *Анотація*

*Використання 2D та 3D комп'ютерної графіки при вивченні фізики дає можливість поліпшити викладання. Такий підхід покращує професійні здібності студентів та сприяє їх конкурентоспроможності.*

Движение объектов в окружающем мире описывается физическими законами. Для создания реалистичного движения в компьютерной анимации необходимо применять законы физики. Именно поэтому общепризнанной практикой [1,2] является, включение в программу подготовки специалистов по компьютерной анимации в качестве базовых знаний законов Ньютона, теории колебаний, момента импульса, тензора инерции и т.д.

Использование физических законов при создании web-сайта [3], с использованием Flash презентаций, позволяет создать реализм, который было бы невозможно достичь иным способом.

Моделирование физических процессов в 3D графике с помощью VPython достаточно широко используется в образовании ([www.vpython.org](http://www.vpython.org)). 2D графика при изучении физики не применяется, но создание компьютерных игр на Flash основано на физических законах[2]. Поэтому изучение физики с применением графических пакетов является актуальным.

В статье показано применение программ Flash MX и VPython для моделирования процессов кинематики и динамики. Лабораторные работы по физике [4,5], выполняемые студентами в Харьковском национальном университете радиоэлектроники, вырабатывают навыки использования физических законов в компьютерном моделировании, обучают созданию компьютерных игр.

После изучения курса физики с использованием программирования, студенты вполне способны написать простые игры и реалистичные презентации с использованием ActionScript.

Моделирование на Flash начинается с создания движущегося с постоянной скоростью объекта

```

xmov=3;
_root.onEnterFrame = function(){
    car._x += xmov;
}

```

Небольшое усложнение кода позволяет ввести ускорение, а также воспроизвести движение по оси у.

Применение второго закона Ньютона позволяет моделировать движение тела под действием сил:

```

var ymov=0;
var mass=10;
var force1=50;
var force2=-31;
var netForce=force1+force2;
var yaccel=netForce/mass;
_root.onEnterFrame=function(){
    ymov+=yaccel;
    ball._y+=ymov;
}

```

После изучения приведенных выше простых кодов, студенты могут изменять значения массы движущихся объектов и сил. Далее вводится коэффициент трения и кнопки, которые регулируют движение объектов.

При нажатии одной кнопки объект движется равномерно, второй — с ускорением, третьей — останавливается. При таком подходе обучение носит игровой характер.

Приведены коды программ, которые позволяют имитировать упругое и неупругое столкновения объектов.

После освоения простых кодов, рассматриваются компьютерные игры. Например, игра, в которой шарик, имеющий начальную скорость, движется под действием приложенной силы. При каждой новой загрузке амплитуды силы и начальной скорости, а также углы, под которым эти векторные величины приложены к шарiku, задаются произвольно. Для наглядности направления скорости и силы показаны векторами. Движение шарика описывается физическими законами. Требуется поймать шарик в корзину, которая перетаскивается с помощью мыши. Если шарик пойман, то он становится невидимым. Вид игрового поля приведен на рис.1.



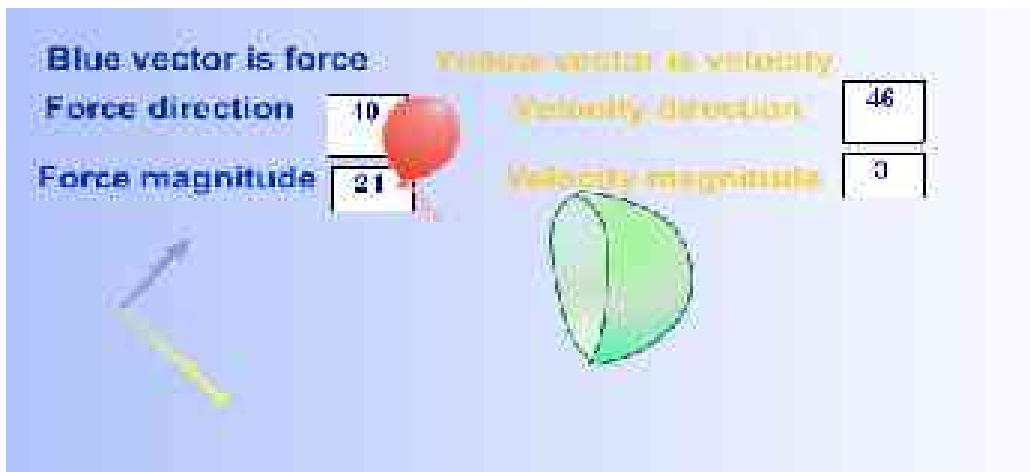


Рисунок 1

Моделировать физические процессы путем программирования 3D графики достаточно сложно. Использование языка Python позволяет преодолеть эту сложность. Python — интерпретируемый, объектно-ориентированный, высокоуровневый язык программирования. Модуль VPython подключается для создания **3D объектов**.

Во многих университетах мира студенты используют VPython для моделирования физических процессов. Использование данной программы при изучении физики ставит образование на уровень передовых стран. Синтаксис языка достаточно понятен, трехмерные модели строятся при подключении графического модуля. Подключается также модуль для создания графиков.

В пособие по лабораторным работам «Моделирование физических процессов в VPython» включены основные сведения по подключаемому модулю VPython. Оно содержит подробное описание синтаксиса языка, а также примеры по созданию 3D объектов. Студенты вначале повторяют простые короткие коды программ, данные в качестве примеров, и создают геометрические объекты. Затем учатся изменять параметры этих объектов (размеры, цвет, координаты). Далее идет усложнение кода программ, и объектам задают скорость и ускорение. Последовательно подключается модуль, при помощи которого строятся графики изменения пути или скорости от времени. При таком поэтапном усложнении кода, студенты осваивают программирование на VPython достаточно быстро. Поэтому основное внимание уделяется изучению физического явления.

Например, при изучении поступательного движения, набирается следующий код

```
from visual import *
```

```

from visual.graph import *
win=500
scene=display(title="TRANSLATION MOTION", width=win,
height=win, x=0,y=0)
floor=box(length=50, height=0.1, width=10, color=color.red)
body=box(length=1,height=1, width=1, color=color.blue)
body.pos=vector(0,0.5,0)
body1=box(length=1, height=1, color=color.green)
body1.pos=vector(0,0.5,-1)
body.velocity=vector(-1,0,0)
body1.velocity=vector(1,0,0)
ppos=gdisplay(x=0,xmax=1,y=win,ymax=5,width=win,
height=win/2,xtitle='time',
ytitle='coordinate x')
path=gcurve(color=color.cyan)
path1=gcurve(color=color.red)
dt=0.01
for t in arange(0,1.01,0.01):
    path1.plot(pos=(t,-1*t+5*t*t/2))
    path.plot(pos=(t,1*t+3*t*t/2))
while 1:
    rate(50)
    body1.pos.x=body1.pos.x+body1.velocity.x*dt
    body.pos.x=body.pos.x+body.velocity.x*dt
    body1.velocity.x=body1.velocity.x+7*dt
    body.velocity.x=body.velocity.x+3*dt

```

При запуске программы, создается поверхность, по которой двигаются два объекта (рис.2).

Код программы подключает графический модуль, модуль для создания графиков, создает объекты и задает законы движения. Одновременно с движением объектов, строятся графики зависимости пройденного пути от времени (рис.2). **Сочетание запрограммированных физических формул, визуального наблюдения движения объектов, графиков позволяет понять суть физического явления.** Когда объекты встречаются, графики пересекаются, и определяется время и место встречи. Данная лабораторная работа помогает понять физический смысл отрицательных ускорений и скоростей. Можно поставить задачу о нахождении скорости, при которой один объект догонит второй. **Во многих задачах кинематики требуется найти время и место встречи двух движущихся объектов. Умение студентов решать данный тип**

задач может быть проверено с помощью лабораторной работы. Если задача решена правильно, то время и место встречи, показанное на графике, будет совпадать с полученным при решении.

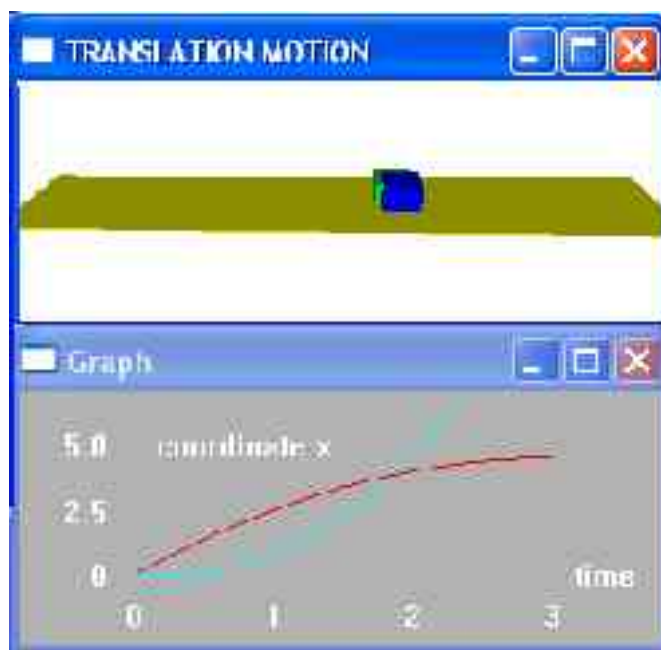


Рисунок 2

Векторные величины в VPython могут трактоваться как векторные физические величины (сила, скорость, ускорение и т.д.). Изменение компонент вектора рассматривается как изменение проекций векторных физических величин на соответствующие оси декартовой системы координат. Это дает возможность понять суть векторных физических величин. Меняя компоненты векторной величины, можно увидеть их влияние на характер протекания физического процесса. Это позволяет представить, например, как изменится движение тела при изменении компонент начальной скорости. Такое ясное понимание, основанное на визуальном восприятии 3D графики, практически невозможно осуществить при натуральных экспериментах.

В одной из лабораторных работ изучается движение тел под действием приложенной силы по соответствующим траекториям (рис.3).

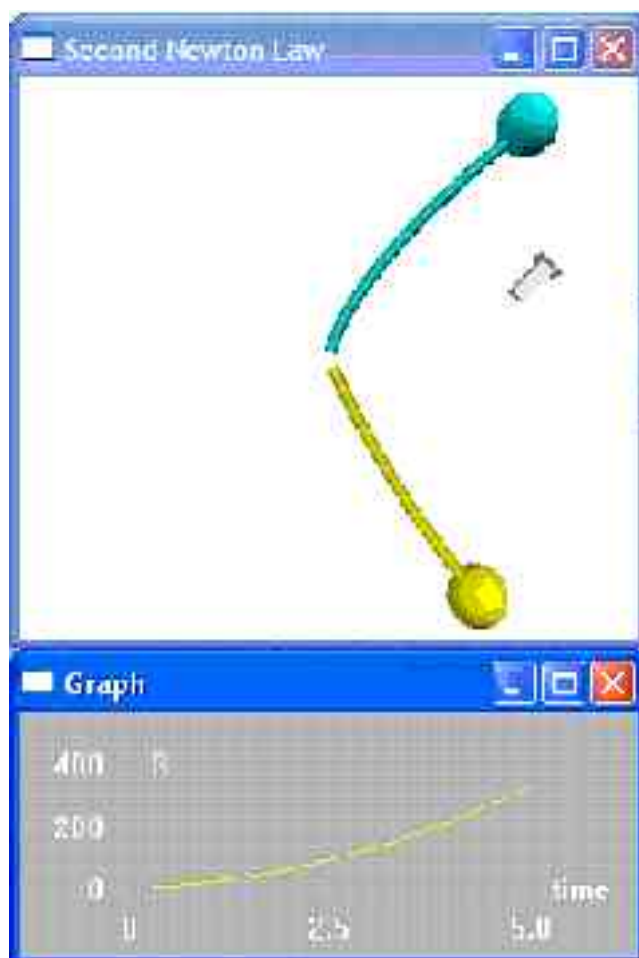


Рисунок 3

График показывает изменение расстояния между объектами в зависимости от времени (рис.3). В качестве самостоятельной работы предлагается изменить компоненты силы. **На основе данной лабораторной работы можно проверить знания студентов по кинематике и динамике. Можно предложить студентам решить данную задачу аналитически, и найти расстояние между объектами в определенный момент времени. Затем запустить программу на выполнение и проверить полученный результат.**

Использование ползунков и переключателей применяется для изменения скоростей вращательного и поступательного движения тела. Комбинация положений ползунков имитирует плоское движение тела. При этом студенты учатся моделировать плоское движение без проскальзывания.

Одним из сложных является о движении твердого тела, брошенного под углом к горизонту.

Подобного рода задачи часто возникают при создании игр. В общем случае, как известно, такое движение представляет собой поступательное и вращательное движения. В зависимости от точки приложения силы, характер движения тела меняется. Объяснить вид движения без привлечения 3D графики достаточно трудно. И в данном случае, 3D графика «помогает» быстрому и ясному пониманию физического явления.

Таким образом, изучение физики, как основы для создания компьютерных игр представляется динамичным и эффективным. Данный подход к изучению сочетает в себе традиционные лабораторные работы и практические занятия, но при этом вызывает значительно больший интерес у студентов, способствует получению более профессиональных и востребованных рынком знаний и умений.

### *Литература*

1. Рик Пэрент Компьютерная анимация. Кудиц-образ, Москва. 2004.
2. Джоб Макара. Секреты разработки игр в Macromedia Flash MX, Кудиц-образ, Москва. 2004.
3. Джоди Китинг Flash MX Искусство создания web-сайтов, DiaSoft, Москва, 2003.
4. Е.Е.Гетманова, Hint for laboratory physics computer works, based by Flash MX, 3D MAX, MathCAD, ХНУРЭ, Харьков, 2004.
5. Е.Е.Гетманова, Моделирование физических процессов в VPython, ФинАрт, Харьков, 2004.

УДК 681.5

**РАЗРАБОТКА ЭЛЕКТРОННОГО УЧЕБНИКА  
ПО «ИНЖЕНЕРНОЙ И КОМПЬЮТЕРНОЙ ГРАФИКЕ  
В ПРИМЕРАХ»**

З.П. Гаврилова, В.В. Зулкарнеева  
Факультет высоких технологий РГУ  
zinaida78@mail.ru

Для повышения эффективности образования вводятся активные методики обучения, основным преимуществами которых, является сокращение числа лекционных занятий, при обеспечении получения студентами основных знаний посредством самостоятельного ознакомления с учебными материалами в доступной форме. Проведение занятий с применением новых информационных технологий показывает, что такой подход повышает интерес студентов к изучаемой дисциплине и объективность самоконтроля знаний, улучшает качество подготовки, облегчает контакт студентов с новой областью знаний. Использование электронного учебника, как в локальной сети класса, так и в Интернет, обеспечивает возможность самостоятельно освоить учебный курс или его большой раздел и позволяет достигнуть существенного результата в обучении. Применение электронного учебника облегчает работу самого преподавателя на занятиях и может использоваться для самостоятельной работы студентов. Предположительно электронный учебник соединяет в себе свойства обычного учебника, справочника, задачника и лабораторного практикума.

В соответствии с учебным планом по курсу «Инженерной и компьютерной графики», преподаваемого студентам дневного отделения факультета высоких технологий и других инженерных специальностей РГУ, изучающих, нормативный курс инженерная и компьютерная графика был разработан электронный учебник «Инженерная и компьютерная графика в примерах». Для создания электронного учебника по курсу «Инженерной и компьютерной графики» использовались язык программирования HTML и среда Macromedia Flash MX.

Процесс формирования данного электронного учебника состоял из следующих этапов:

- Разработки оболочки для содержательной части учебника,
- Составление и компоновка текста учебника
- Разработка наборов контрольных вопросов и упражнений для каждой главы учебника

- Создание компьютерных анимаций.

В состав данного электронного учебника входят следующие компоненты:

- образовательный минимум по дисциплине;
- программа курса;
- литература;
- вопросы;
- практические занятия по темам с мультимедийными упражнениями и вопросами к каждой теме.

Данный электронный учебник имеет удобный интерфейс и руководство пользователю в виде схемы учебника (рис. 1), с помощью которой можно попасть на любую страницу учебника. Каждый пункт в учебнике представляет собой кнопку, которая выводит на экран текст или пример. Указатели на каждой странице позволяют легко «путешествовать» по структуре учебника, при этом каждый из обучающихся может индивидуально определять объем учебной информации по той или иной теме. Лекционный курс и другие текстовые материалы электронного учебника представляются в виде статических HTML-страниц — это обеспечивает «прокрутку» текста стандартными средствами.

Электронный учебник разбит на две главы: компьютерная графика и инженерная графика. Следующие страницы содержат краткое содержание и примеры, необходимые для усвоения графического материала, выполненного с помощью программ CorelDRAW и AutoCAD — это чертежи, схемы, графики, рисунки, механизмы построения которых представлены цветными клипами, подготовленными в среде Macromedia Flash MX. Пространственное построение деталей и рисунков подкреплено трехмерными цветными изображениями, которые можно передвигать и вращать на экране.

Предполагается использовать электронный учебник не только в локальной сети факультета, но и в Интернет в связи с малым занимаемым размером файлов.

Работа над электронным учебником продолжается, увеличивается количество заданий тренировочного характера, адаптируются методические указания к выполняемым заданиям (детальное описание порядка выполнения построения), обновляется курс лекций. Планируется дополнить учебник заданиями для дистанционного и очно-заочного обучения.

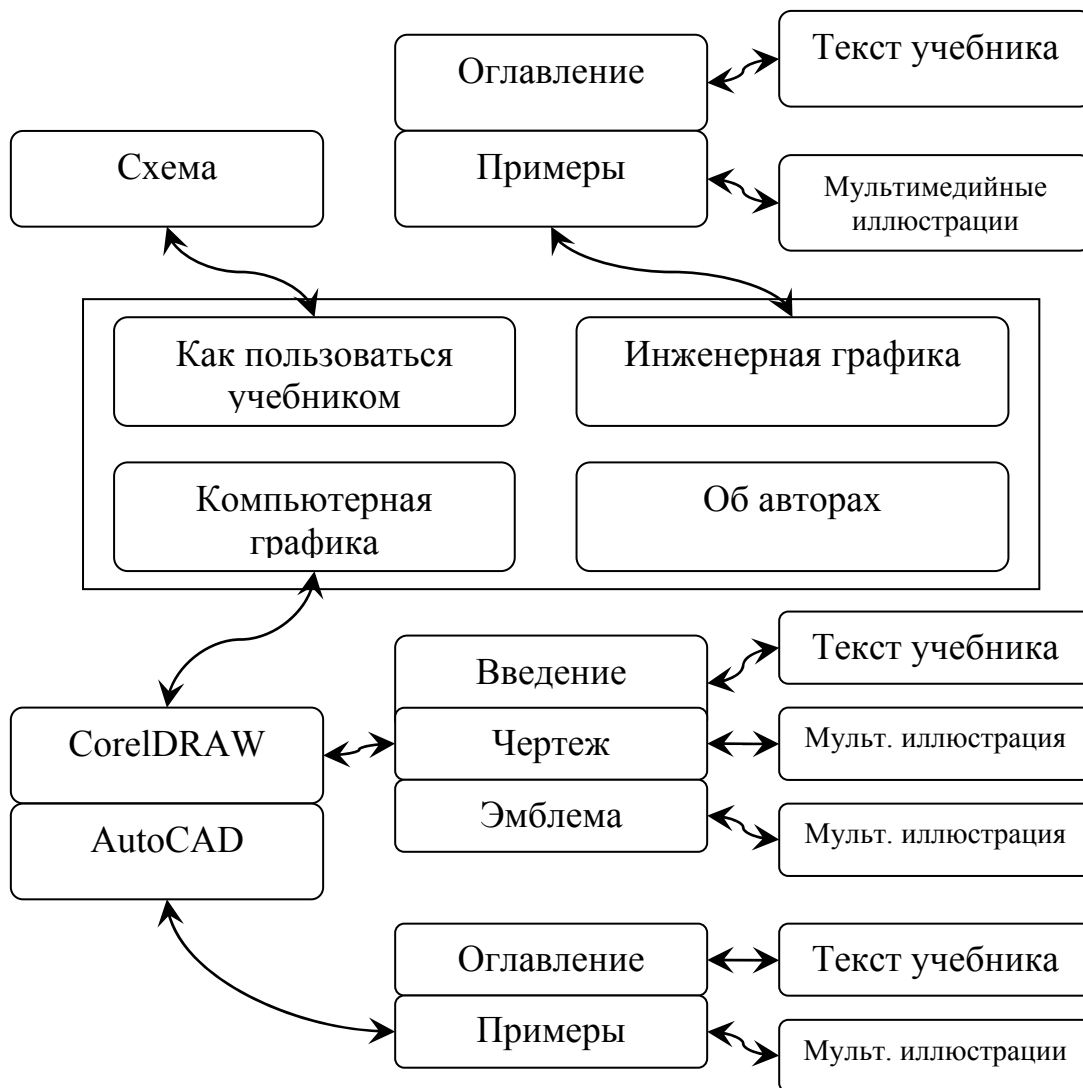


Рисунок 1 — Схема электронного учебника «Инженерная и компьютерная графика в примерах»



## ОБУЧАЮЩАЯ ПРОГРАММА ДЛЯ GEOMEDIA PROFESSIONAL

Н.Г. Фаттахова

Ростовский государственный университет, Россия

### ***Анотація***

*Навчальна програма реалізована на FlashMX і являє собою курс відеолекцій, який дозволяє за короткий час одержати навички роботи з GeoMedia Professional, а на закінчення провести тестування й одержати об'єктивну оцінку отриманим знанням.*

GeoMedia Professional — продукт корпорации Intergraph, позволяющий обрабатывать географические данные и совершать над ними такие операции как: оцифровка аналоговых карт, анализ местности, разбиение цифровой карты местности на слои, которые, например, позволяют проследить населенность территорий, количественные и качественные характеристики зеленых насаждений местности, автотрассы, железные дороги и др.

Актуальность создания обучающей программы связана, с одной стороны, с расширением круга пользователей, а, с другой стороны, с отсутствием методических пособий или подробных руководств пользователя GeoMedia Professional на русском языке. Встроенный в программу обучающий курс «Tutorial» (написанный на английском языке) представляет собой лишь поверхностное описание программы, не являющееся полным и подробным руководством по работе с GeoMedia Professional.

Средой для создания программы выбрана программа FlashMX, со встроенным объектно-ориентированным языком программирования Action Script, которая позволяет создавать полноценные интерактивные программные приложения, итоговые файлы которых получаются очень «легкими» (обучающая программа со встроенными видеолекциями «весит» всего тридцать шесть мегабайт).

Уроки создавались следующим образом: для записи видеоуроков использовалась программа Quick Screen Recorder, которая записывает в видеофайлы все происходящее на мониторе компьютера. В видеофайлы записывался процесс работы в GeoMedia Professional. Затем видео импортировалось во FlashMX и производилось озвучивание уроков. Для удобства пользователя, чтобы можно было интерактивно управлять воспроизведением урока, при запуске урока, к нему подгружается панель управления, которая также реализована на Action Script, и которая позволяет пользователю

управлять уроком: воспроизвести, остановить, перемотать вперед или назад.

Каждая из кнопок является объектом типа Button, к которому привязан скрипт, реализующий соответствующее действие программы, как, например, для кнопки «Оглавление»:

```
... on (press) {loadMovieNum("Оглавление.swf", 0);} ...
```

При нажатии на эту кнопку, пользователь прерывает воспроизведение урока и переходит к оглавлению, и обратно, при наведении курсора мыши на какой-либо из уроков, пользователь, нажатием левой кнопки вызывает воспроизведение урока. Ниже на рис.1 приведен фрагмент воспроизведения видеурока:

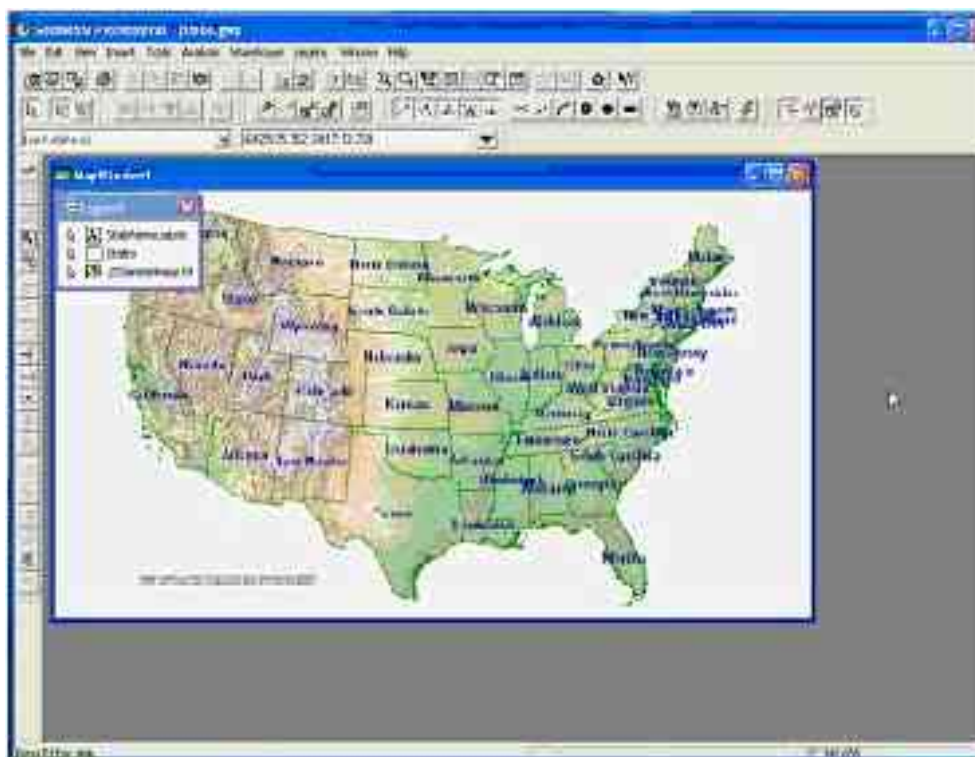


Рисунок 1 — Фрагмент воспроизведения видеурока

При создании уроков возникали следующие проблемы: звуковая дорожка отставала от видеоизображения. Эта задача решалась с помощью встроенного во FlashMX кодека Sorenson Spark, и поиска соответствия и точки равновесия между количеством импортируемых «ключевых кадров» и качества изображения (Quality). При импорте клип перекодируется с использованием алгоритма Sorenson Spark и разработчик софта имеет возможность задавать его параметры. Для задания параметров качества использовался компонент Compression Profile (Профиль сжатия). В нем представлены несколько так называемых профилей — predetermined, поименованных

наборов параметров. Каждый такой профиль задает параметры сжатия, наиболее подходящие для передачи готового фильма по различным сетям. С помощью расширенных настроек задавались дополнительные параметры клипа: яркость, цветовая насыщенность и пр. Процесс перекодировки клипа занимает много времени и зависит от мощности компьютера. Далее для нормального воспроизведения внедренного видеоклипа, для него создавалась последовательность кадров соответствующей длины. Озвучивание уроков шло отдельно, после импорта видеофайлов. Импорт звука происходило по схеме подобной импорту видео.

Логическим завершением программы является, реализованный на Action Script, тест, который позволяет пользователю проверить полученные знания и получить оценку или рекомендации по дальнейшей работе. При запуске теста пользователю предлагаются правила работы с тестом, правила заполнения. Тестирование состоит из 20 вопросов/заданий, которые пользователь проходит последовательно. Программа подсчитывает количество правильных ответов и на основании количества заработанных баллов формирует результирующую оценку.

К преимуществам созданной обучающей программы следует отнести то, что информация, представленная визуально и озвученная, гораздо лучше и быстрее усваивается, чем текст; программа снабжена гипертекстовым меню, позволяющим переходить от одного урока к другому; уроки тематически сгруппированы; у пользователя есть возможность проверить и закрепить полученные во время обучения знания; программа имеет удобный, интуитивно понятный интерфейс пользователя и занимает очень мало дискового пространства.

Разработанная обучающая программа была одобрена представителями компании Intergraph. В данный момент наращивание идей по усовершенствованию программы и по ее дальнейшей разработке продолжается: планируется ввести словарь терминов, разработать общий дизайн программы, а также создать английскую версию программы.

## ПРИНЦИПЫ ПОСТРОЕНИЯ БАЗ ЗНАНИЙ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

В.С. Быков

Международный научно-учебный центр ЮНЕСКО информационных технологий и систем НАН Украины и МОН Украины

### *Анотація*

*Розглядаються проблеми побудови баз знань графічних образів. Аналізуються й обґрунтовуються принципи їх побудови і практичного використання. Приводяться архітектурні і проектні рішення. Досліджуються і будуються класифікатори графічної інформації. Розглядається безліч прикладів збереження, структуризації, пошукових процедур графічних образів.*

**Введение.** Общеизвестно, что эффективность любой интеллектуальной системы в первую очередь определяется мощностью ее базы знаний (БЗ). В моделировании образного мышления [1–3] это тем более значимо, так как наша мыследеятельность существенно зависит от наших знаний. Вместе с тем, специалисты по созданию интеллектуальных технологий больше внимания уделяют разработке различных методов, алгоритмов обработки данных, информации, чем систематизации необходимых знаний. Данная работа является продолжением исследований, результаты которых, в частности, изложены в [1–4].

**Проблема.** К образному мышлению мы относим [1] операции, в которых в качестве операндов и результатов используются, в настоящее время чаще, графические образы (ГО). В рамках проекта «*Образный компьютер*» разрабатывается программно-информационный инструмент для осуществления этих процессов. В частности, одной из основных проблем реализации такого инструмента является создание систем управления базами знаний (СУБЗ) графической информации.

**Принципы построения БЗ графической информации.** В настоящее время усилия проектировщиков графических интерфейсов в основном ориентированы на разработку новых форматов ГО и эффективных методов и средств их синтеза-анализа. Но, часто, в различных ГО отражаются разные стороны (аспекты) исследуемого информационного объекта, или один ГО характеризует несколько подобных (по определенным признакам) объектов, необходима интерпретация графической информации, ее структуризация, классификация по самым различным критериям и т.д. (см., например [1–4]). Поэтому в интеллектуальных системах целесообразно строить и поддерживать в актуальном состоянии соответствующую БЗ графической информации. Введем ряд рабочих определений.

Под графической информацией будем понимать системы взаимосвязанных графических образов, с их трактовкой, отображающих некоторый объект. Как правило, этот объект представлен в соответствующей БЗ или БД (в форме реляционной или гипертекстовой или другой модели в компьютерной среде). Под ГО понимаем любые данные, представленные в графическом формате. Здесь, под базой знаний графической информации будем понимать систему взаимосвязанных ГО с их интерпретацией и отношениями с моделируемыми информационными объектами и операциями синтеза-анализа ГО.

Любой ГО строится с какой либо целью и является отражением некоторых (исходных) данных, знаний, информации. Как правило, целью синтеза ГО являются анализ соответствующего процесса, явления, объекта, их трактовка, интерпретация, объяснение, передача понимания, принятие решений.

Как обычно в БЗ выделим три взаимосвязанных макрообъекта: декларативные, процедурные и концептуальные знания. Здесь рассмотрим декларативный макрообъект. В нем выделяются четыре взаимосвязанные компоненты: БЗ метаданных информационных объектов — прообразов для синтеза соответствующих ГО (**META**); БЗ проблем и целей интеллектуального анализа данных, в которых используются ГО (**PROB**); БЗ методов и средств генерации и анализа ГО (**REFL**); БЗ графических образов (**PATR**).

Кроме этого, поддерживается специальный ассоциатор отношений между БЗ **META**, **REFL**, **PATR** и **PROB**.

1. В БЗ **META** содержится информация обо всех объектах предметной области интеллектуальной системы, для которых целесообразно использовать графические интерфейсы. По сути, **META** — это репозиторий используемого хранилища данных или словарь-справочник метаданных соответствующей системы, но ориентированный на синтез-анализ ГО. Поэтому, с одной стороны, в нем должно быть достаточно информации для поиска прообраза для генерации ГО, с другой — форматы представления исходных данных должны соответствовать методам, используемым для синтеза ГО. Если соответствия нет, то строится или используется специальный конвертор. Таким образом, БЗ **META** связана с соответствующей системой конверторов-преобразователей форматов хранения данных графических образов. Заметим, в роли прообразов для синтеза ГО могут выступать и другие графические образы. И эти графические прообразы могут поступать как из внешних источников, так и порождаться рассматриваемой системой. Поэтому репозиторий **META** должен включать описание форматов хранения и указателей путей доступа не только к информационным объектам БЗ соответствующей системы, но и к объектам «соб-

ственной» базы знаний PATR. БЗ МЕТА — необходимая компонента БЗ образов, независимо от форм (звук, текст, графика, текстура и т.д.) и форматов (таблицы, ГО, аналитические формулы и т.д.) их представления и предметной/проблемной области, образное мышление которой моделируется.

2. В БЗ PROB описываются характеристики проблемных областей информационно-аналитических систем, в которых используются графические методы исследования и с какими целями генерируются соответствующие ГО. В этой БЗ выделяются две взаимосвязанные структуры: сеть «проблем-задач» и сеть «целей-средств» синтеза ГО. Их взаимное отображение обеспечивает процессы анализа эффективности использования графики в интеллектуализации функций соответствующей системы. Здесь, кроме этого, накапливается и систематизируется положительный и отрицательный опыт синтеза-анализа ГО. Отношения «проблем-задач» с «целями-средствами» необходимо отражать в БЗ образного мышления независимо от формата и формы представления образа и от компьютеризируемой предметной области.

3. В БЗ REFL описываются характеристики методов и средств получения ГО из прообразов (информационных объектов предметной области). 1) В методах выделяются *классические* математические (построение графиков аналитических, статистических и других зависимостей в двумерном или трехмерном видах), *традиционные* компьютерные (синтеза диаграмм, гистограмм, кругорам и т.п.) и *специальные*. К последним относятся различные оригинальные методы, вновь разработанные или, по различным причинам, не нашедшие в настоящее время широкого применения. В частности, например, это система ТЕКРИС — синтеза рисунка по тексту, ранее разрабатываемая в ВЦ РАН. 2) Средства делятся на: аппаратные (и «ручные»), аппаратно-программные и программные. а) Отличительной особенностью первых является то, что результатом их работы является образ «вне вычислительной среды» (например, ручной эскиз, фотография и т.п.). Поэтому необходимо проведение предварительной операции (оцифровки) перед погружением ГО в БЗ. б) Цифровые видеокамеры, фотоаппараты и другие измерительные приборы и системы — аппаратно-программные средства. в) Третий класс средств включает самые разнообразные инструментальные компьютерные комплексы, ориентированные на синтез графических образов. Это могут быть статические и динамические ГО, растровые и векторные и т.п. Классификации форм и форматов ГО посвящена отдельная работа. БЗ REFL описываются также средства и методы анализа ГО. Они связаны с инструментами (в широком смысле) синтеза ГО, так как существенно зависят от форм и форматов представления образов. В анализе также выделяются методы и средства. Заметим, один и тот же метод может быть реализован в различных средствах. Мето-

ды также делятся на классические, традиционные и специальные. Например, традиционными методами являются операции поворотов, увеличения/уменьшения масштаба и т.д. В средствах анализа мы рассматриваем только программные комплексы, так как с их помощью осуществляются операции с ГО, представленными в вычислительной среде. Описание характеристик и свойств методов и средств синтеза-анализа образов различных форм представления и их форматов для БЗ образного мышления более значимы, ввиду их множества и разнообразия.

4. БЗ PATR — центральная компонента базы знаний графической информации интеллектуальной компьютерной технологии. В этой БЗ хранятся все графические образы, генерируемые и анализируемые в процессе функционирования соответствующей интеллектуальной системы. Особый интерес представляют отношения между образами. Это обусловлено тем, что в БЗ ГО эти отношения, как правило, более «семантически насыщены» относительно традиционных БЗ. Например, в отношении «части–целое» необходимо отражать местоположение («выше», «ниже», «вне», «внутри», «слева», «справа», «на», «над», «под» и т.п. — для моделей пространственных отношений), ориентацию и прочие характеристики соответствующих ГО–фрагментов в данном целом ГО. Поддержка этих отношений требует использования в БЗ лингвистических шкал и методов нечеткой математики. Заметим, что отношения ассоциации, подобия, аналогии должны снабжаться весовыми характеристиками. Также сложно описывать родо-видовые отношения между ГО или ассоциативные. В этой БЗ отражаются отношения между ГО, обусловленные принадлежностью некоторому информационному объекту. Вообще в отношениях между ГО должны отражаться структуры отношений между информационными объектами соответствующей интеллектуальной системы. Поскольку любой образ должен иметь прообраз, то у любого ГО должны быть «родители» или в среде БД ГО, или в среде информационных объектов. В БЗ ГО также поддерживаются некоторые специальные отношения: «абстрактное–конкретное», «общее–частное» и т.д. В БЗ PATR также поддерживаются обычно несколько уровней графемных (неделимых) конструкций. Для поддержки этих отношений в БЗ ГО используются фреймовые структуры и инструментарий многомерных семантических сетей. Заметим, с нашей точки зрения, интеллектуализация графического обеспечения процессов анализа данных предполагает постепенное выделение стереотипных графических конструкций (часто и эффективно применяемых), из которых можно будет построить алфавит (словарь) графем для синтеза более сложных ГО. Это отношение аналогично отношению «часть–целое».

**Выводы.** На основании проведенных исследований можно сформулировать следующие основные принципы создания БЗ ГО.

1. БЗ ГО должны строиться как открытая система. 2. База знаний должна включать пять макрообъектов, описанных выше. 3. Архитектура БЗ ГО должна поддерживать многослойность представления графического материала (графемных структур). 4. Отношения в этой БЗ значительно сложнее традиционных БЗ, что является, в частности, отражением параллельно-пространственности процессов графического мышления. 5. Помимо БЗ ГО необходимо строить и БЗ семантической (в частности, вербальной) интерпретации ГО. Эти БЗ тесно взаимосвязаны, как правополушарное и левополушарное мышление. 6. В БЗ необходимо формировать (динамически) множество контекстов ассоциативных и других отношений ГО между собой. 7. Процедурная компонента БЗ должна включать как логический (традиционный) вывод, так и графический (визуальный), образный. Необходимо моделировать и взаимодействие между ними. 8. Структуры БЗ ГО в значительной степени более динамичны относительно традиционных интеллектуальных систем. Форматы хранения ГО должны обеспечивать возможность различных модификаций.

В дальнейшем в МНУЦИТиС предполагается создание инструментального комплекса построения БЗ ГО на базе определенных здесь принципов.

### **Литература**

1. Валькман Ю.Р., Исмагилова Л. Р. О языке образного мышления // Труды Международного семинара Диалог'2004 «Компьютерная лингвистика и интеллектуальные технологии». — Тверь, 2004. — С. 131–142.
2. Быков В.С. Проблема и принципы построения баз знаний для моделирования образного мышления // Тезисы докладов Международной конференции «Интеллектуальные системы принятия решений и прикладные аспекты информационных технологий (ISDMIT'2005)» — Евпатория-18.V–21.V.2005 — (в печати).
3. Быков В.С. Базы знаний моделирования визуального мышления в интеллектуальных системах // Тезисы докладов 12-ой Международной конференции по автоматическому управлению «Автоматика–2005» — Харьков –30. V–3.VI.2005 — (в печати).
4. Валькман Ю.Р., Книга Ю.Н. Анализ понятия «графический образ» // Труды Международного семинара Диалог'2002 «Компьютерная лингвистика и интеллектуальные технологии». — Протвино, 2002. — С. 41–52.



## **СОВРЕМЕННЫЕ СИСТЕМЫ ПОИСКА ИЗОБРАЖЕНИЙ НА ОСНОВЕ ИХ СОДЕРЖИМОГО**

Н.С. Костюкова  
Кафедра ПМиИ, ДонНТУ

### ***Анотація***

*У статті розглядаються питання, пов'язані з побудовою систем пошуку зображень за їх змістом, розглядаються алгоритми, що використовуються в процесі пошуку, виконується класифікація таких систем за рядом ознак.*

### ***Введение***

Разработка новых технических средств, позволяющих представлять информацию в виде изображений, привела к тому, что работа с графическими данными оказалась в центре внимания многих исследователей. С распространением Internet наблюдается тенденция к накоплению информации, представленной в виде изображений, и к созданию баз данных (БД) изображений. В зависимости от области применения в такие БД включаются как произвольные изображения, так и изображения ограниченного класса. Общей проблемой для таких БД является эффективный поиск.

Наиболее простым и исторически первым способом решения задачи поиска изображения был поиск по текстовому описанию. Главным недостатком такого поиска остается субъективность описания, поскольку оно составляется человеком, что в итоге ухудшает качество поиска. Более предпочтительным в данной ситуации является поиск, основанный на автоматическом сравнении содержимого изображений. Примером успешной реализации такого поиска могут служить биометрические системы, в которых обрабатываемые изображения принадлежат к ограниченному классу. Например, в работах по распознаванию человеческих лиц учитывается тот факт, что цвет человеческой кожи зависит от концентрации меланина, а также то, что число вариантов геометрии человеческого лица ограничено. Системы, обрабатывающие изображения отпечатков пальцев или сетчатки, используют ограничения другого рода. Это позволяет уменьшить размерность хранимых данных без потери общего зрительно воспринимаемого качества изображений [1]. Алгоритмы, используемые в современных системах такого рода, чувствительны к колебаниям в освещении, изменению положения камеры, а также к применению камер различных типов. Что же касается

произвольных цветных изображений, то задача их успешного сопоставления до сих пор успешно не решена, хотя имеется насущная потребность в ее решении. В настоящее время существует большое количество библиотек оцифрованных изображений, содержащих до 1 млн. картинок, среди которых виртуальная библиотека Нового Южного Уэльса [2], цифровая коллекция Национального Русского Эрмитажа [3], коллекция живописи Национальной Галереи Искусств США [4], цифровая библиотека Vcl Antropomorphic Image Library [5], цифровая астрономическая библиотека [6]. Поиск по содержимому в настоящее время реализован только в цифровой коллекции Эрмитажа.

Ключевым моментом при решении задачи контекстного поиска изображений является сравнение изображений, под которым понимается автоматическая установка соответствия между визуальными характеристиками двух или более изображений. Первой попыткой решения этой задачи было сравнение пикселей, образующих изображение, однако этот подход оказался практически нереализуемым из-за значительной пространственно-временной сложности: во-первых, объем БД изображений, как правило, очень велик, и, во-вторых, изображения состоят из миллионов пикселей, из-за чего для их попарного сравнения необходимо слишком много времени. Кроме того, при попиксельном сравнении изображений мелкие детали (в частности, отдельные различающиеся пиксели) могут сильно повлиять на результаты сравнения.

В настоящее время сформировался подход к решению данной задачи, называемый контекстным поиском изображений. Указанный подход использует дескриптивную теорию изображений [7], согласно которой первоначально создается формальное описание изображения, а впоследствии выполняется анализ и обработка этого описания, а не самого изображения. Таким образом, доступ к изображению осуществляется только в момент его занесения в БД и при визуализации результатов поиска, а сам поиск осуществляется на основе метаданных — характеристик изображения, вычисленных при его занесении в базу и хранящихся вместе с ним. Архитектуры систем контекстного поиска изображений и сравнение систем контекстного поиска изображений с точки зрения их функциональных возможностей рассматривались автором ранее в работе [8].

### *Алгоритмы поиска изображений в базах данных*

Одна из наиболее значимых воспринимаемых человеком характеристик изображения — цвет, поэтому все существующие системы контекстного поиска обязательно используют его вместе с другими признаками. Наиболее простые реализации поиска используют значения наиболее часто используемого цвета, наиболее яркий цвет, цвет фона изображения [9]. Такие оценки содержимого изображения называют точечными, и их существенным недостатком является неточность описания содержимого изображения.

Популярным способом представления цветового содержимого изображения являются цветовые гистограммы [9, 10], характеризующиеся простотой построения и сравнения. Для сравнения двух цветовых гистограмм используются различные метрики (евклидово, квадратичное, косинусное расстояния, конъюнкция гистограмм), описанные в литературе [9,10]. Кроме этого, в литературе [11] в качестве способа представления информации о цветовом содержимом изображений описаны кумулятивные гистограммы, для сравнения которых используются те же величины, а также первый и второй центральный моменты [11].

Кроме подходов, основанных на глобальных характеристиках изображений, используют также поиск, учитывающий признаки отдельных областей изображения. Выделение областей может быть выполнено различными способами. Наименее сложным является ручное выделение областей, когда изображение обрабатывается человеком и вся необходимая информация определяется визуально. В частности, этот подход использовался разработчиками системы QVIC [12]. Однако ручное выделение областей и объектов крайне громоздко и для больших наборов изображений требует значительных временных затрат. Другой подход заключается в том, что области изображения получают в результате разбиения изображения на фиксированные фрагменты. Этот прием был предложен в [13]. Ограничением на его использование является то, что в общем случае сложно выбрать для каждого изображения масштаб разбиения, которое, к тому же, не является инвариантным относительно сдвига и масштабирования. Третий подход включает в себя сегментацию изображений — разбиение изображения на неперекрывающиеся гомогенные сегменты. Препятствием в использовании данного подхода является то, что для произвольных изображений не существует способа

определения правильной сегментации или оценки корректности ее выполнения и то, что задача сегментации изображения не имеет единственного решения [14].

Кроме указанных выше, для учета пространственного расположения цветов используются и другие методы: использование коэффициентов дискретного косинусного преобразования, обратное проецирование цветовых гистограмм [15], построение матрицы корреляции [16], разбиение изображения на прямоугольные области и построение локальных ЦГ. Все рассмотренные подходы обладают как достоинствами, так и недостатками.

Еще одним средством представления цветового содержимого изображения является цветовая коррелограмма (далее просто коррелограмма) — пространственная корреляция пар изменений цветов с расстоянием [17]. В отличие от гистограммы, коррелограмма охватывает не только распределение цветов, но и учитывает пространственное распределение цветов в изображении, позволяя определить различие изображений, которые при использовании точечных оценок и цветовых гистограмм считались бы одинаковыми. В качестве меры различия коррелограмм принимается их конъюнкция.

Многие изображения представляют собой проявление повторяющейся структуры (аналогично кирпичной кладке, структуре ткани или узору паркета). Такие изображения называют текстурами, и существует ряд задач, в рамках которых необходимо сравнивать текстурные характеристики изображений [18]. Из рассмотренных ранее систем поиска изображений поиск по текстурным признакам реализован в системах Virage и Visualseek. В системе VisualSeek изображение текстуры рассматривается как сигнал, имеющий девять частотных составляющих. Для представления текстурных характеристик в этой системе используются текстурные гистограммы и бинарные векторы [14].

### ***Классификация систем контекстного поиска изображений***

Проведенный анализ существующих реализаций контекстного поиска изображений позволил выделить ряд признаков, на основе которых может быть выполнена классификация систем контекстного поиска изображений. Так, все системы, решающие рассматриваемую задачу, различаются по следующим признакам:

- 1) по сфере применения:
  - универсальные (VisualSeek, QBIC, Virage);

- для специальных классов изображений (FaceIt, TrueFace, Viisage);
- 2) по цветовой гамме обрабатываемых изображений:
  - системы для поиска цветных изображений (VisualSeek, QBIC, Virage, FaceIt, TrueFace, Viisage);
  - системы для поиска полутоновых изображений (VisualSeek, Virage — при поиске по текстурным признакам, биометрические системы, использующие изображения отпечатков пальцев и черно-белые фотографии);
  - системы для поиска черно-белых изображений;
- 3) по используемым признакам содержимого изображения:
  - системы поиска по цветовому содержимому (QBIC);
  - системы поиска по текстурному содержимому;
  - комбинированные системы (VisualSeek, Virage)
  - системы поиска с учетом пространственной информации (VisualSeek, QBIC, Virage, FaceIt, TrueFace, Viisage);
- 4) по структуре признаков содержимого изображения:
  - системы с точечными оценками;
  - системы с гистограммными оценками (VisualSeek);
  - системы с комбинированным представлением (Virage, Qbic);
  - системы со специфическими (проблемно-ориентированными) оценками (FaceIt, TrueFace, Viisage);
- 5) по учету пространственной информации:
  - системы, учитывающие пространственную информацию;
  - системы, не учитывающие пространственную информацию;
- 6) по типу используемой базы данных:
  - системы, работающие в WWW;
  - системы, работающие с локальной базой данных.

### *Литература*

1. Пентланд А., Чаудхари Т. Распознавание лиц для интеллектуальных сред. Открытые системы, №3. — 2000, с.28–33.
2. Виртуальная библиотека Нового Южного Уэльса: <http://cdromweb.snsweb.gov.au/picman/welcome.html>.
3. Цифровая коллекция Эрмитажа: <http://www.hermitagemuseum.org>.
4. Коллекция живописи Национальной галереи искусств США: <http://www.nga.gov>.
5. Цифровая библиотека VCL Antropomorphic Image Library: <http://us.vclart.net/vcl>
6. Цифровая астрономическая библиотека: [www.astronomy.ca/images/](http://www.astronomy.ca/images/).
7. Параллельная обработка информации: В 5 т./ т.5: Проблемно-ориентированные и специализированные средства обработки информации: Под ред. Б.Н.Малиновского и В.В.Грицька. — 1990. — 504 с.

8. Шозда Н.С. Сравнительный анализ систем контекстного поиска изображений. Наукові праці Донецького державного технічного університету. Серія: "Інформатика, кібернетика та обчислювальна техніка". Випуск 6. — Донецьк, 1999. — С.151–156.
9. Прэтт У. Цифровая обработка изображений: Пер. С англ. — М.: Мир, 1982. — Кн. 2 —310 с., ил.
10. John R. Smith and Shih-Fu Chang. Tools and Techniques for Color Image Retrieval ACM Multimedia 1996: Boston, MA.
11. M. Striker, M. Orengo. Similarity of color images// SPIE. — Vol. 2420, NO. 1, 1995. — pp. 381–392.
12. IBM Almaden Research Center. Query by Image and Video Content: the QBIC System. Computer, September 1995, 23–31.
13. M. Stricker and A. Dimai. Color indexing with weak spatial constraints. In Symposium on Electronic Imaging: Science and Technology - Storage & Retrieval for Image and Video Databases IV, volume 2670, pages 29–41. IS&T/SPIE, 1996.
14. John R. Smith Integrated Spatial and Feature Image Systems: Retrieval, Analysis and Compression. Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Graduate School of Arts and Sciences, Columbia University, 1997.
15. V. Vinod, Hiroshi Murase, Object Location Using Complementary Color Features: Histogram and DCT. NTT Basic Research Labs, 3–1 Morinosato Wakamia, Atsugishi, 243–01 Japan.
16. Ліпанов О.В. Алгоритми слідкуючої нормалізації у системах технічного зору. Автореферат дисертації на здобуття наукового ступеня кандидата технічних наук. — Харків, 1999. — 18 стр.
17. J. Huang. Color Spatial Image Indexing and Application. a Dissertation Presented to the Faculty of the Graduate School of Cornell University to Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy, 1998.
18. Шозда Н.С. Поиск изображений по текстурным признакам в больших базах данных. Наукові праці Донецького державного технічного університету. Серія: "Інформатика, кібернетика та обчислювальна техніка". Випуск 39. — Донецьк, 2002. — С.182–187.

## ПРИМЕНЕНИЕ ПРОЦЕДУРЫ КЛАСТЕРИЗАЦИИ ДЛЯ СОДЕРЖАТЕЛЬНОГО ПОИСКА ИЗОБРАЖЕНИЙ

О.Л. Вовк

Донецкий национальный технический университет

### *Анотація*

*В роботі розглядаються можливості використання процедури кластеризації для пошуку зображень в електронних колекціях. Наводяться переваги застосування кластеризації зображень у порівнянні з іншими технологіями контекстного пошуку.*

Тенденция постоянного увеличения количества изображений, доступных в мировой паутине Internet, повышает интерес исследователей к разработке эффективных механизмов поиска изображений. На данный момент разработано множество систем поиска, основанных на текстовых описаниях изображений. Примерами могут служить такие системы как система поиска коллекция живописи Национальной галереи искусств США [1], поисковая система AltaVista [2, 3]. Однако, технология поиска, основанная на текстовых описаниях, ассоциированных с изображением, может быть применена только при поиске внутри ограниченной коллекции, а не в вычислительной сети Internet, так как неоднозначность соответствия между визуальным содержанием и текстовым описанием снижает показатели точности и полноты поиска [4]. Поэтому для решения задачи поиска изображений во всемирной сети принято использовать механизм контекстного поиска изображений.

Поиск изображений по контекстному содержимому [5] — набор технологий для извлечения семантически связанных изображений из базы данных, основанный на автоматическом выделении характеристик изображения. Т.е. в базе данных выполняется поиск изображений визуально сходных с заданным изображением (изображением-шаблоном, изображением-образцом) по заданному набору свойств (характеристик, визуальных примитивов). Стоит отметить, что при поиске изображений в Internet, анализируемая база данных формируется путем индексирования ресурсов мировой паутины.

Визуальные примитивы — это характеристики изображения, которые автоматически вычисляются по оцифрованным визуальным данным, позволяют эффективно индексировать их и обрабатывать запросы с использованием визуальных свойств изображения. Поисковый образ изображения, сгенерированный из визуальных примитивов, невелик по размеру в сравнении с самим изображением и удобен для организации

поиска [4]. Примерами наиболее распространенных визуальных примитивов служат примитивы цвета, текстуры и формы [5].

Общая схема содержательного поиска изображений состоит из следующих этапов [1, 5]:

- индексирование изображений базы данных путем автоматического расчета визуальных характеристик;
- вычисление визуальных характеристик изображения-запроса;
- сравнение (вычисление степени подобия) характеристик всех изображений базы данных и характеристик изображения-запроса;
- сортировка результатов запроса по уменьшению меры подобия изображению-образцу.

Разработано несколько групп методов контекстного поиска, основные из них:

- группа методов цветовых гистограмм [1, 5]. В основе методов данной группы: дискретизация цветового пространства и построение гистограммы распределения дискретизированного пространства цветов изображения;
- группа методов поиска по “цветовой планировке” [5]. Отличительной чертой этой группы алгоритмов является предварительная сегментация изображений на блоки заданного размера или в заданном количестве. Этот этап позволяет учитывать местоположение того или иного цвета, что делает методы поиска по “цветовой планировке” более совершенными по сравнению с предыдущей группой методов;
- группа методов, основанных на кластеризации изображений [5]. При поиске изображений методами этой группы сравнение визуальных примитивов осуществляется на уровне отдельных объектов (регионов, областей) изображений, которые выделяются в процессе кластеризации.

Стоит отметить, что корректное удовлетворение запросов возможно только при использовании методов из третьей группы в виду следующих недостатков методов двух других групп [5]:

- невозможность выделения характеристик формы объектов изображений без выделения самих объектов;
- “чувствительность” к масштабированию, повороту и переносу объектов внутри изображений в случае использования группы методов цветовых гистограмм, и к масштабированию и повороту — группы методов поиска по “цветовой планировке”;
- зависимость результатов поиска от вариации цвета и освещенности изображений.



Кластеризация [6] — общее название множества вычислительных процедур, используемых при создании классификации объектов. В результате кластеризации образуются “кластеры” — группы похожих по различным характеристикам объектов. Кластерный метод [6] — многомерная статистическая процедура, выполняющая сбор данных, содержащих информацию о характеристиках объектов, и затем упорядочивающая объекты в сравнительно однородные группы. Исходные данные для процедуры кластеризации — набор объектов, каждый из которых задается вектором своих характеристик.

В случае решения проблемы распознавания образов изображений, заданный набор точек кластеризируемого изображения рассматривается в качестве многомерной совокупности объектов, заданных цветовыми характеристиками, согласно которым требуется рассортировать эти объекты на несколько классов-образов изображения (распределить по регионам). Пример кластеризации изображения приведен на рис. 1.

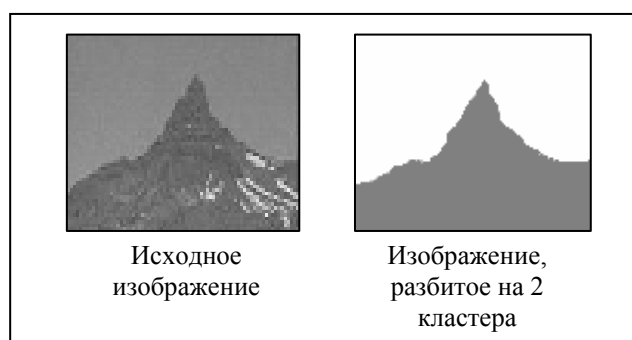


Рисунок 1 — Пример кластеризации изображений

Статистические методы кластеризации наиболее часто применяются для выделения объектов изображений, причем, наиболее популярный статистический метод — метод k-means кластеризации [5].

Кроме статистических методов кластеризации, для распознавания объектов изображений используют методы кластеризации, основанные на выделении перепадов яркости [4]. При кластеризации данными методами проблема выделения границ объектов решается путем локализации на изображении резких перепадов яркости цвета [4]. С этой целью вычисляется градиент функции интенсивности в каждой точке изображения, после чего подавляются значения, не превышающие установленного порога. За основу принято брать метод Собеля.

Для увеличения скорости работы алгоритмов кластеризации изображений производится предварительное уменьшение количества цветов классифицируемого изображения — квантование (дискретизация) пространства цветов [7].

При поиске изображений в электронных коллекциях процедуру кластеризации используют не только для распознавания объектов изображений, но и для разделения изображений анализируемой коллекции на группы со сходными визуальными характеристиками [8, 9]. Настоящее применение позволяет увеличить скорость поиска заданного изображения базы данных, так как поиск осуществляется не среди всех изображений, а только среди изображений группы с подобными визуальными свойствами. Такое применение кластеризации особенно актуально, если электронная коллекция изображений сформирована из изображений сети Internet, количество которых неограниченно увеличивается с каждым днем.

### *Литература*

1. Башков Е.А., Шозда Н.С. Поиск изображений в больших БД с использованием коэффициента корреляции цветовых гистограмм // GraphiCon'2002. — Нижний Новгород. — 2002. — С. 458–460.
2. Qui G., Palmer R.D. WISE: A Content-based Web Image Search Engine // Proc. Multimedia Computing and Networking 2001, January 21–26, 2001, San Jose, CA, USA
3. <http://images.altavista.com/>.
4. Байгарова Н.С., Бухштаб Ю.А., Евтеева Н.Н. Современная технология содержательного поиска в электронных коллекциях изображений. — Институт прикладной математики им. М. В. Келдыша РАН, <http://artinfo.ru/eva/eva2000M/eva-papers/200008/Baigarova-R.htm>.
5. Wang J. Z., Li J. Wiederhold G. SIMPLicity: Semantics-Sensitive Integrated Matching for Picture Libraries // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2001. — vol. 23, №9. — P. 947–963.
6. Ким Д. О., Мьюллер Ч. У., Клекка У. Р. Факторный, дискриминантный и кластерный анализ. — М.: Финансы и статистика, 1989. — 215 с.
7. Schettini R., Ciocca G., Zuffi S. A Survey of Methods for Colour Indexing and Retrieval in Image Databases , [www.ivl.disco.unimib.it/Teaching/CorsoEI-2003-04Complementi/articoli%20e%20dispense/methodsCIIR-9-1-01a.pdf](http://www.ivl.disco.unimib.it/Teaching/CorsoEI-2003-04Complementi/articoli%20e%20dispense/methodsCIIR-9-1-01a.pdf)
8. Chen J.Y., Bounman C.A., Dalton J.C. Hierarchical Browsing and Search of Large Image Databases // IEEE Transactions on Image Processing. — 2000. — vol. 9, №3. — P. 442–455.
9. Abdel-Mottaleb M., Krishnamachari S., Mankovich N.J. Performance Evaluation of Clustering Algorithms for Scalable Image Retrieval // IEEE Computer Society Workshop on Empirical Evaluation of Computer Vision Algorithms. — Santa Barbara. — 1998.

## **МАТЕМАТИЧЕСКИЕ МОДЕЛИ ПОВЕРХНОСТЕЙ СЛОЖНОЙ ФОРМЫ В СИСТЕМАХ МОДЕЛИРОВАНИЯ И ВИЗУАЛИЗАЦИИ**

Е.А. Башков, Д.П. Пауков

Донецкий национальный технический университет

### ***Анотація***

*Моделювання та візуалізація об'єктів складної форми широко використовує математичні моделі поверхонь складної форми, серед яких важливе місце посідає криві та поверхні в формі Без'є, сплайни та NURBS. Стаття присвячена огляду цих способів подання геометричних даних в системах моделювання та візуалізації.*

### ***Введение***

Моделирование и визуализация объектов сложной формы с большим объемом вычисляемых данных является в последнее время важной задачей. Проблема привлечения средств моделирования и визуализации возникает в таких областях как геоинформационные системы, медицина, отображение результатов научных экспериментов, реставрация объектов искусства, системы виртуальной реальности. В связи с постоянным развитием этих направлений научных исследований актуальность работ, связанных с исследованием моделирования и визуализации сложных объектов будет в дальнейшем только возрастать.

Для моделирования и визуализации таких объектов в системах компьютерной графики привлекаются, как правило, методы математического моделирования поверхностей пространства. Важность этих методов состоит и в том, что они обеспечивают необходимый математический аппарат для построения систем моделирования и визуализации, обеспечивающих достаточную точность представления для решения имеющихся и возникающих задач.

Перечисленные выше предметные области, как правило, используют в своем анализе объекты сложной формы, которые требуют обработки больших объемов данных и при этом высокую точность представления. Обычно поверхности таких объектов представляют собой сложные образования, которые можно моделировать или кривыми поверхностями или графическими моделями. При визуализации таких объектов широко применяются триангуляционные сетки. Таким образом, особое внимание в компьютерной графике уделяется именно кривым гладким поверхностям и триангуляционным сеткам. Поэтому существует два направления исследований в области представления сложных объектов. Во-первых, усовершенствование существующих и

разработка новых математических способов представления таких объектов; во-вторых, разработка методов управления и усовершенствования сеток. Такая работа уже ведется. Математические модели представления поверхностей в компьютерной графике, а также геометрия кривых и поверхностей описаны в [1–4, 10, 11]. Методы и алгоритмы усовершенствования триангуляционных сеток представлены в [5–9]. В этих работах излагаются идеи, позволяющие существенно упростить триангуляционные сетки при визуализации.

### ***Формы представления кривых***

Рассмотрим математические основы представления кривых и поверхностей. Существует три формы наиболее распространенных способа представления кривых и поверхностей: явная, неявная и параметрическая форма.

Явная форма представления предусматривает наличие уравнения (или совокупность уравнений), в левой части которого стоит зависимая переменная, а в правой — функция от независимой переменной. Например, для описания кривой в пространстве имеем два уравнения от одной независимой переменной  $x$ :

$$\begin{cases} y = f(x) \\ z = g(x) \end{cases} \quad (1)$$

Для описания поверхности в явном виде используют явную функцию от двух независимых переменных:

$$z = f(x, y). \quad (2)$$

Важно отметить, что нет никакой гарантии, что для определенной линии или поверхности существует явное уравнение, поэтому часто используется неявная форма представления. Уравнение линии на плоскости в этом случае будет иметь вид

$$f(x, y) = 0. \quad (3)$$

Поверхность может быть описана следующим уравнением трех переменных в неявной форме:

$$f(x, y, z) = 0. \quad (4)$$

Представить линию в пространстве в неявной форме не так просто: необходимо использовать систему уравнений, представляющих поверхности, пересечение которых (если оно существует) образует кривую.

$$\begin{cases} f(x, y, z) = 0 \\ g(x, y, z) = 0 \end{cases} \quad (5)$$

Параметрическая форма представления наиболее удобна для задач моделирования и визуализации. Ее суть состоит в том, что

кривая в пространстве описывается совокупностью функций одной переменной, например  $t$ . Поверхность — совокупностью функций двух независимых переменных, например,  $u$  и  $t$ .

$$\begin{cases} x = x(t), \\ y = y(t), \\ z = z(t) \end{cases} \quad \begin{cases} x = x(u, t), \\ y = y(u, t), \\ z = z(u, t) \end{cases} \quad (6)$$

Как правило, функции  $x(t)$ ,  $y(t)$ ,  $z(t)$  и  $x(u, t)$ ,  $y(u, t)$ ,  $z(u, t)$  выбираются в виде полиномов. Параметрическая форма является наиболее гибкой и удобной, она позволяет легко манипулировать функциями, вычислять производные для осуществления требования гладкости кривых и поверхностей.

### ***Методы моделирования кривых и поверхностей***

При моделировании и визуализации исходная информация об объекте часто представляется в виде набора опорных точек, описывающих кривую или поверхность. В зависимости от интерпретации этих точек, а также от требований к гладкости поверхности, методы моделирования кривых и поверхностей можно разделить на три группы: интерполяционные кривые и поверхности, кривые и поверхности в форме Эрмита и Безье, B-сплайны и NURBS.

### ***Интерполирование***

Интерполирование предусматривает требование: кривая или поверхность должна проходить через заданные точки. Это условие используется для вычисления коэффициентов полинома.

Наиболее широкое распространение в классе интерполяционных кривых и поверхностей получили кубические интерполяционные полиномы. Для построения кубического интерполяционного полинома для кривой в пространстве необходимо иметь 4 опорных точки, через которые кривая будет проходить, а, следовательно, 12 уравнений с 12 неизвестными, что позволяет определить все коэффициенты полинома. Для кубической интерполяционной поверхности необходимо задать 16 опорных точек, чтобы вычислить 48 коэффициентов, определяющих поверхность (см. рис. 1, 2). Более подробно интерполяционные кривые и поверхности описаны в [1, 2, 10].

Существенным недостатком интерполяционных кривых и поверхностей часто выступает то, что количество опорных точек определяет степень полинома. Для преодоления этого недостатка на больших наборах точек строят сегменты интерполяционных кривых, при этом возникает проблема управления гладкостью кривой и поверхности в месте соединения сегментов.

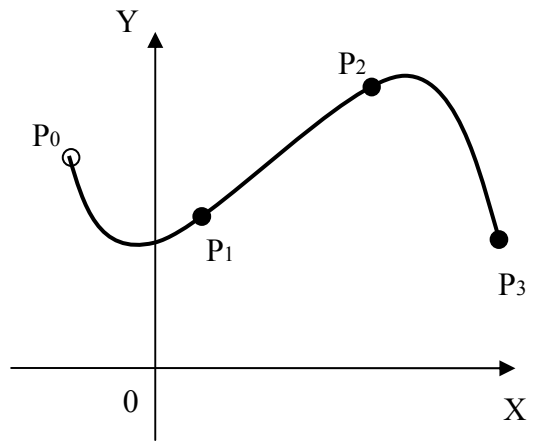


Рисунок 1 — Кубическая интерполяционная кривая

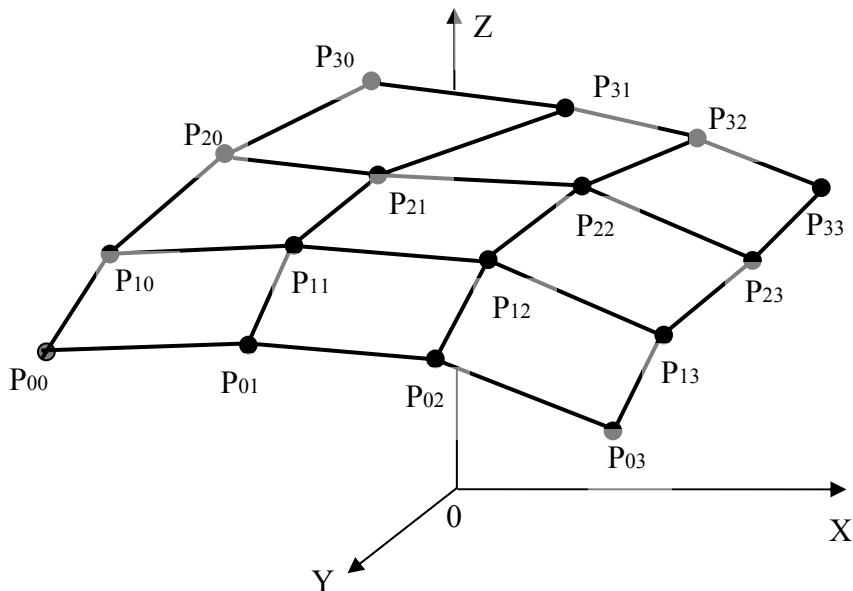


Рисунок 2 — Набор опорных точек для построения порции кубической интерполяционной поверхности

### **Форма Эрмита и Безье**

Для управления гладкостью кривых и поверхностей в местах соединения сегментов интерполяционных кривых Эрмитом было предложено задавать в начальной и конечной точке сегмента значения производных (см. рис. 3а) [10]. Форма Эрмита требует использовать для каждого сегмента 2 опорные точки и задавать для каждой из них значения производных. Форма Безье не предусматривает задания производных и так же как форма Эрмита позволяет управлять гладкостью кривых. Форма кривых Безье использует аппроксимацию

векторов касательных (см. рис. 3б), что позволяет использовать для задания кривых Безье только опорные точки без производных.

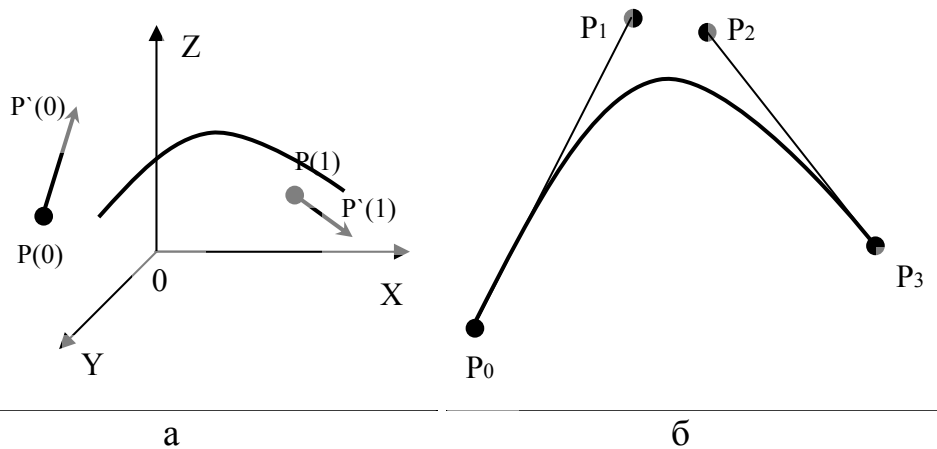


Рисунок 3 — Определение кривой в форме Эрмита и форме Безье

Кривые и поверхности Безье в настоящее время получили широкое распространение в системах компьютерной графики, визуализации и моделирования: пользователь может задать положение 4 опорных точек и, изменяя их положения, корректировать форму кривой. Подробное описание кривых и поверхностей в форме Безье приводится в [1, 2]. В [10] кроме описания кривых и поверхностей в форме Безье имеется алгоритм рекурсивного разбиения кривых и поверхностей, который является основой аппроксимации кривых поверхностей полигональной моделью.

### ***В-сплайны***

Если отказаться от требования, чтобы формируемая кривая или поверхность проходила через опорные точки (проходила близко к ним), то можно обеспечить гладкость и непрерывность не только самой кривой или поверхности, но и ее первой и второй производных в точках сопряжения сегментов (см. рис. 4).

Если по одному и тому же набору точек построить интерполяционную кривую, кривую Безье и В-сплайн, то окажется, что для В-сплайна необходимо выполнить в 3 раза больше вычислений, чем для интерполирования и кривой Безье. Аналогично порция поверхности В-сплайна требует в 9 раз больше вычислений, чем интерполяционные поверхности и поверхности Безье. Однако метод В-сплайнов позволяет построить более плавную кривую или поверхность, чем кривые и поверхности в форме Безье. Для подробного знакомства с В-сплайнами см. [1–3, 10, 11].

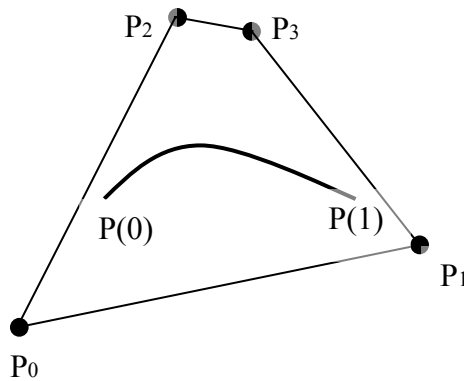


Рисунок 4 — Сегмент В-сплайна

Важной разновидностью В-сплайнов считается неравномерный рациональный В-сплайн (NURBS, nonuniform rational B-spline). Суть NURBS состоит в том, что для построения В-сплайна в трехмерном пространстве точки  $p_i = (x_i, y_i, z_i)$  представляют в пространстве однородных координат

$$q_i = w_i \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} \quad (7)$$

Коэффициенты  $w_i$  используют для увеличения или уменьшения влияния конкретной опорной точки в наборе. Эти опорные точки используются для формирования четырехмерного В-сплайна, первые три компоненты которого будут представлять собой В-сплайновое представление опорных точек [1, 2, 10]. Таким образом, NURBS кривые и поверхности сохраняют все свойства трехмерных В-сплайнов, но, кроме этого, обладают еще важным свойством корректного перспективного преобразования. В последнее время NURBS в компьютерной графике получили широкое распространение, потому что на их основе можно эффективно моделировать практически все распространенные типы криволинейных геометрических объектов.

### **Выводы**

В статье рассмотрены наиболее распространенные математические модели поверхностей объектов сложной формы, используемые в современных системах моделирования и визуализации.



При визуализации, как правило, применяются полигональные модели, которые получаются путем аппроксимации математических моделей поверхностью триангуляционной сетки. Такие триангуляции могут быть построены, например, на основе имеющегося представления поверхности в форме Безье и рекурсивного его деления на мелкие составляющие [10], которые затем заменяют плоскими поверхностями (треугольниками). Полученную полигональную модель упрощают [5–9], а затем визуализируют.

Таким образом, моделирование и визуализация объектов сложной формы является сложной задачей, связанной, как правило, с обработкой больших объемов данных. Поэтому исследования связанные с разработкой оптимального представления поверхностей таких объектов с целью уменьшения объемов вычисляемых данных являются актуальными и в настоящее время требуют рассматривать не только эффективность представления и моделирования поверхностей, но и работать над усовершенствованием методов аппроксимации и улучшения триангуляционных сеток, а также непосредственно над алгоритмами визуализации.

### *Литература*

1. Farin G. Curves and Surfaces for Computer Aided Geometric Design. — Academic Press, New York, 1988.
2. Роджерс Д., Адамс Дж. Математические основы машинной графики. — М.: Мир, 2001 — 604 с.
3. Фокс А., Пратт М., Вычислительная геометрия: применение в проектировании и производстве — М.: Мир, 1982 — 302 с.
4. Watt A. 3D Computer Graphics, Second Edition. — Addison-Wesley, Workingham, England, 1993.
5. Н. Hoppe “Progressive Meshes”, ACM Siggraph 96 Conf. Proc., pp. 99–108, 1996.
6. Н. Hoppe “Efficient Implementation of Progressive Mesh”, Computer & Graphics, vol. 22, 1998.
7. D. Cohen-Or, D. Levin, and O.Remez, “Progressive Compression of Arbitrary Triangular Meshes”, Proc. IEEE Visualization 99, pp. 67–72, 1999.
8. Z. Karni, A. Bogomjakov, and C. Gotsman, “Efficient Compression and Rendering of Multi-Resolution Meshes”, Proc. IEEE Visualization, 2002.
9. S.Valette, R. Prost "Wavelet-Based Multiresolution Analysis of Irregular Surface Meshes", IEEE Transactions On Visualization And Computer Graphics vol. 10, No. 2, march/april 2004.
10. Эйнджел Эдвард Интерактивная компьютерная графика. Вводный курс на базе OpenGL, 2 изд.: Пер. с англ. — М.: Издательский дом «Вильямс», 2001. — 592 с.: ил.
11. Никулин Е. А. Компьютерная геометрия и алгоритмы машинной графики. — СПб.: БХВ-Петербург, 2003. — 560 с.: ил.

## ИСПОЛЬЗОВАНИЕ СИСТЕМ МОДЕЛИРОВАНИЯ ДЛЯ СОЗДАНИЯ ИНТЕРЬЕРА

И.Б. Аббасов

Таганрогский государственный радиотехнический университет

### **Анотація**

*Стаття присвячена моделюванню інтер'єра приміщення бара з використанням графічної системи тривимірного моделювання 3 ds max. Розглядаються питання моделювання предметів інтер'єра, присвоєння матеріалів і текстур. Для створення фотореалістичного зображення використовуються різного типу джерела висвітлення.*

В данной работе рассматриваются вопросы трехмерного моделирования интерьера бара. Для трехмерного компьютерного моделирования используется графическая система 3 ds max 5. Графическая система 3 ds max 5 позволяет создавать как трехмерные сцены, так анимацию. В нашем случае для моделирования объектов интерьера используются двумерные формы, модификаторы создания трехмерных объектов и стандартные трехмерные примитивы.

Для создания стен, рам, стульев, барной стойки и крыши использовались двумерные формы. Фасадная стена (справа) была создана методом выдавливания, боковые стены, барная стойка и набор с бутылкой были созданы методом лофтинга. Стулья были созданы методом выдавливания по профилю. Настенные светильники собраны из четырех частей, при моделировании которых использовались методы лофтинга, выдавливания и вращения.

Диван и настенные полки созданы на основе каркасной решетки трехмерного примитива параллелепипеда. Кресла смоделированы также из параллелепипеда методом полигонального выдавливания. Затем к ним применялись модификаторы пространственной деформации. В собранном виде интерьер помещения бара представлен на рис.1. При сборе интерьера немаловажную роль играет вопрос соблюдения пространственной соразмерности отдельных объектов.

Рассмотрим, например, создание бутылки методом лофтинга. Для использования метода лофтинга требуется две плоские формы. В нашем случае нам потребуются следующие двумерные примитивы: линия, окружность и восьмиконечная звезда.

На основе исходной линии сначала создается объект по начальному сечению в виде окружности. В результате получаем цилиндр, потом, поднимаясь на некоторую высоту, меняем сечение на восьмиконечную звезду, еще через некоторое расстояние возвращаемся к цилиндру. В результате этой операции получаем ис-

ходную заготовку, далее к этой заготовке применяем операции деформирования по высоте и кручения. В дальнейшем для создания набора по тем же сечениям методом лофтинга были созданы рюмки (рис.2).

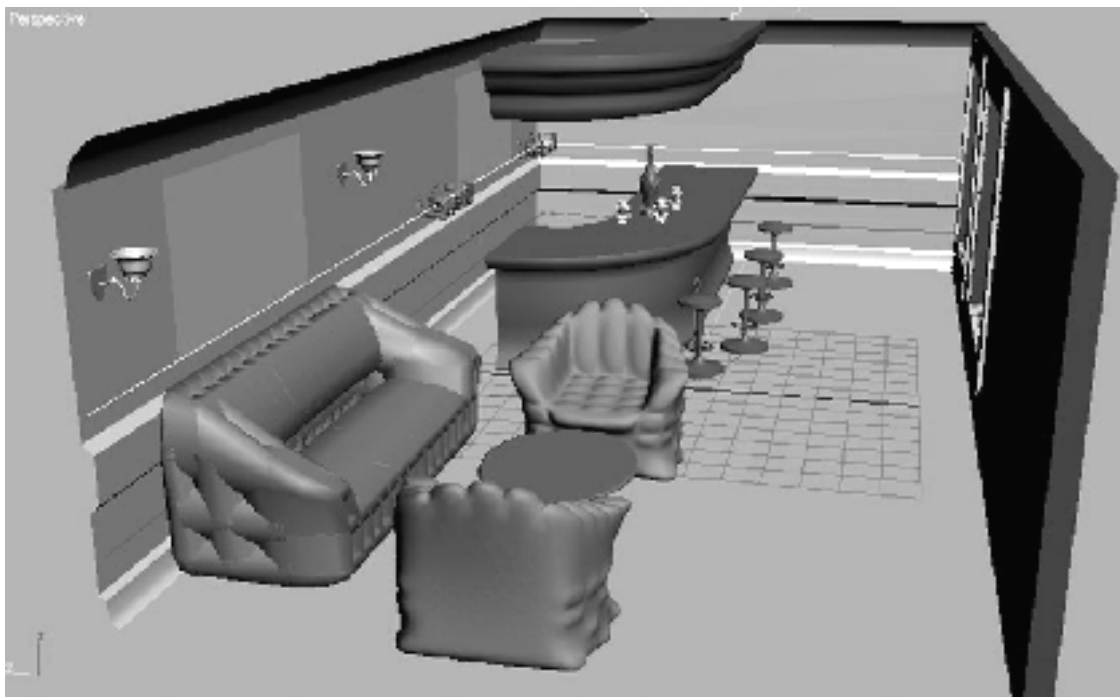


Рисунок 1 — Сбор интерьера бара

После создания всех объектов трехмерной сцены необходимо перейти к раскраске поверхностей. Эта операция осуществляется с помощью редактора материалов. Для раскраски сцены использовались стандартные материалы и текстуры. Стенам присвоены стандартные материалы на основе раскраски по Блинну, с соответствующей настройкой диффузного цвета и параметров зеркального блика.

Реальные материалы мы первую очередь определяем по форме и интенсивности зеркального блика. Пол был застелен терракотовой плиткой, обладающей свойством зеркального отражения. К барной стойке был присвоен многокомпонентный материал, на основе трассировки лучей имитирующий нержавеющую сталь, а столешница была покрашена под красную деревянную плиту. Все остальные объекты сцены были раскрашены исходя из общей концепции — в теплых тонах.



Рисунок 2 — Набор из бутылки с рюмками

Далее были установлены источники света на основе искусственного освещения. На потолок были установлены лампы дневного света и были созданы соответствующие световые пятна на полу. Над барным столиком были установлены прожекторы для направленного освещения. Для них был установлен режим реального затухания с расстоянием.

В настенные светильники были установлены всенаправленные источники света. В целях реалистичности источников освещения, плафонам были присвоены материалы со свойством сияния. После присвоения материалов и установки освещения в окончательном виде визуализация сцены представлена на рис.3. Представленная сцена является достаточно реалистичным, и дает наглядное представление об интерьере помещения. Удобство компьютерного моделирования заключается в том, что после построения трехмерных моделей объектов интерьера, цветовой тон и освещение бара можно менять достаточно быстро без особых материальных затрат, и потом анализируя все варианты можно выбрать наиболее подходящий.



Рисунок 3 — Окончательная визуализация сцены бара

### *Литература*

1. Аббасов И.Б.. Основы трехмерного моделирования в графической системе 3 ds max: Учебное пособие. — Таганрог: Изд-во ТРТУ. — 2005. — 207с.
2. Бордман Т. 3 ds max 5. — СПб.: Издательство «Питер», 2004. — 443 с.
3. М. и Дж. Эйвис. Интерьер. Выбираем цветовой дизайн. — М.: Ниола, 2003. — 288 с.
4. Соловьев М.М. Трехмерный мир 3D Studio Max 5. — М.: Издательство «СО-ЛОН-Пресс», 2003. — 224 с.

УДК 681.3

## МОДИФИЦИРОВАННЫЙ АЛГОРИТМ РЕКОНСТРУКЦИИ ТРЕХМЕРНЫХ СЦЕН ГОРОДСКИХ ЛАНДШАФТОВ НА ОСНОВЕ ФОТОИЗОБРАЖЕНИЙ

С.В. Ковальский, С.А. Зори

Донецкий национальный технический университет

### *Анотація*

*У роботі розглянуті основні підсумки модифікацій алгоритму реконструкції тривимірних сцен по їх фотозображеннях з метою підвищення швидкості і якості процесу*

### **1. Обзор современного состояния решения задачи СИД**

При построении и моделировании трехмерных виртуальных сред, используемых в навигационных системах, летных тренажерах и других системах виртуальной и дополненной реальности, возникает задача создания трехмерной модели существующих в реальности сцен. Наиболее доступным источником информации о реальном окружающем мире в таких случаях являются фотоизображения. Задача реконструкции трехмерной сцены по имеющимся фотоизображениям была впервые поставлена в [1] и получила название Structure-from-Motion — SFM (Структура из движения — СИД). Выполненный обзор литературы по этому вопросу позволяет утверждать — до настоящего времени результаты исследований не позволяют полностью автоматизировать процесс решения задачи СИД и получать гарантированно качественный результат [1–7]. Это дает возможность говорить об актуальности таких исследований.

Алгоритмы решения задачи СИД классифицируются по типам опорных элементов (точки, отрезки прямых), моделям камеры (перспективная, аффинная), особенностям структурной организации (отдельные этапы могут группироваться и выполняться итеративно [6] или рекурсивно [4]), и другим признакам. Подробно классификация алгоритмов СИД была рассмотрена в [7].

В задаче СИД можно выделить ряд этапов (рис. 1).

Этап реконструкции геометрической модели по набору опорных элементов является наиболее трудоемким для ручного выполнения, что делает поиск быстрого и качественного его решения приоритетным направлением в области СИД. В ряде источников [2–5] определение задачи СИД ограничивается рамками данного этапа.

## 2. Предложения по модификации алгоритма реконструкции

В [2] представлен алгоритм реконструкции трехмерной геометрической модели сцены по набору опорных элементов, размеченных и приведенных в соответствие на наборе фотоизображений. Исходными опорными элементами являются отрезки прямых, а результатом реконструкции — набор прямых в пространстве, соответствующих опорным элементам. Алгоритм построен на основе минимизации целевой функции, характеризующей отклонение реконструируемой модели от опорных элементов. Минимизация выполняется по набору параметров, характеризующих ориентации ( $R$ ) и положения ( $t$ ) камер в пространстве при получении исходных изображений, а также характеризующих ориентации ( $v$ ) и положения ( $d$ ) прямых в пространстве, образующих реконструируемую модель. На рисунке 2 приведена схема работы алгоритма.

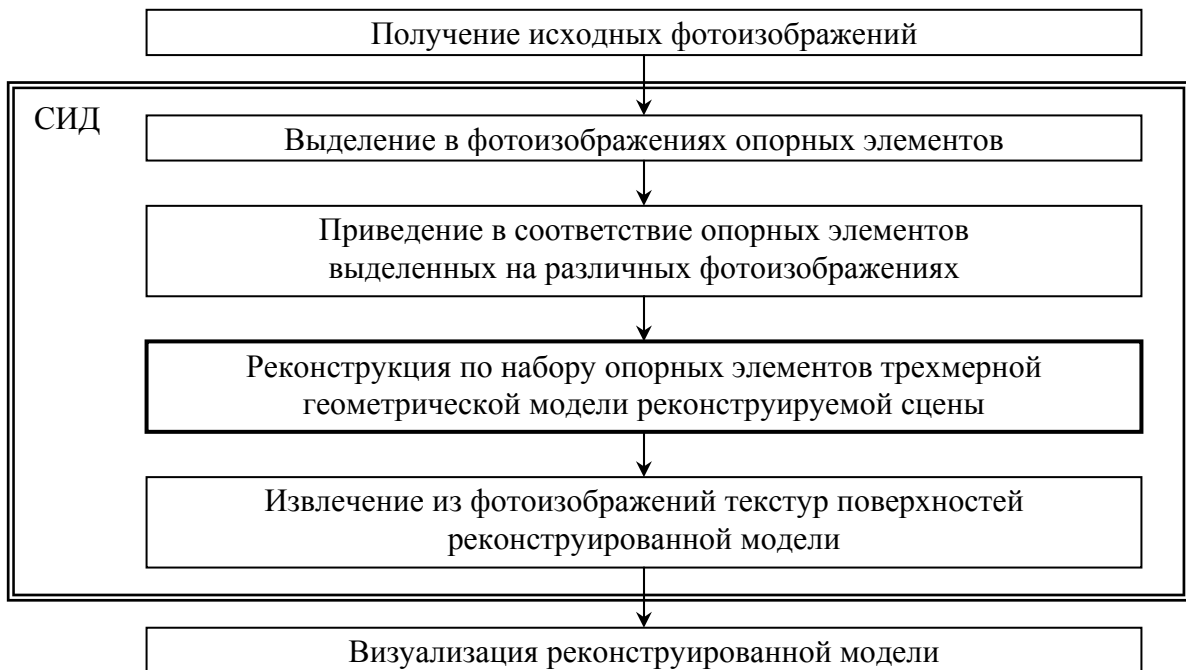


Рисунок 1 — Этапы решения задачи СИД

Описанный базовый алгоритм может требовать значительных затрат машинного времени при большом количестве элементов, образующих реконструируемую модель. Проведенное имитационное моделирование показало, что при количестве опорных элементов более 100 и исходных изображений более 10 для реконструкции на самых современных ПЭВМ требуется более 30 мин, и не обеспечиваются гарантированно качественные результаты реконструкции.

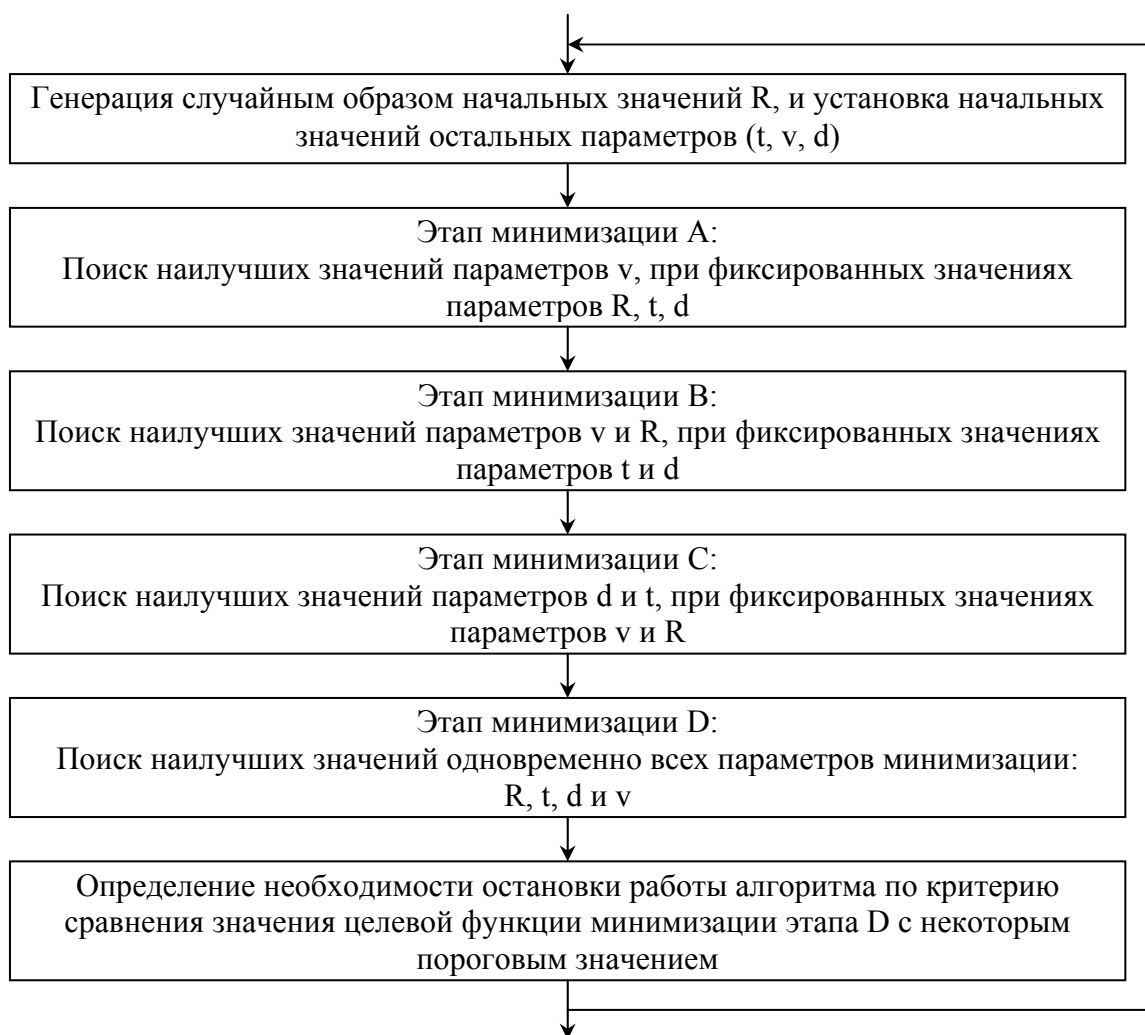


Рисунок 2 — Схема работы алгоритма

На рисунке 3 представлена приблизительная зависимость времени и качества реконструкции от количества опорных линий для количества исходных изображений (позиций камеры), равного 8. В настоящее время проводятся исследования получения зависимостей для различного количества снимков, позиций камер, и конфигурации реконструируемой сцены.

Авторами предлагается ряд модификаций базового алгоритма, повышающих его временные и качественные характеристики.

Часть параметров минимизации целевой функций базового алгоритма описывает преобразование вращения в пространстве. Вращение в пространстве представимо рядом способов, и выбор лучшего из них позволяет значительно снизить вычислительную сложность базового алгоритма. Подробный анализ и определение



лучшего с вычислительной точки зрения способа представления вращения в пространстве был сделан в [8].

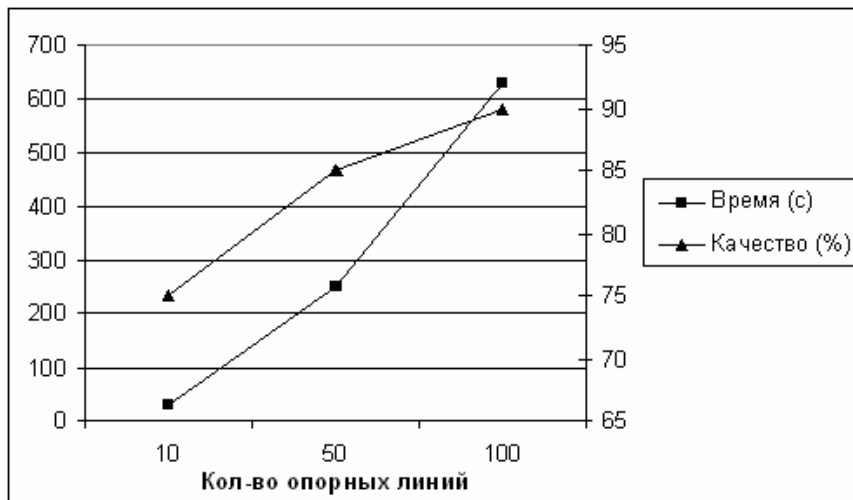


Рисунок 3 — Зависимость времени и качества реконструкции от количества опорных линий

Авторами был проанализирован вид зависимости целевой функции минимизации базового алгоритма от параметров, непосредственно характеризующих отклонение реконструируемой модели от опорных элементов, и обнаружена ошибка в используемой в алгоритме формуле функции минимизации [2], приводящая к неадекватным результатам реконструкции (рис. 4). На рисунке 4 красным цветом обозначены реконструируемые отрезки, голубым цветом — реконструированные отрезки (концы отрезков выделены перекрестиями), серым цветом — не участвующие в реконструкции отрезки. По результатам анализа были сделаны выводы о возможности модификации целевой функции с целью повышения чувствительности алгоритма реконструкции к изменениям параметров отклонения, а также отмечены особенности формы поверхности целевой функции, которые необходимо учитывать при построении критерия останова итераций алгоритма.

Трехмерная модель, получаемая в результате работы базового алгоритма, требует дальнейшего преобразования в представление, традиционное для современных систем визуализация трехмерной графики. В [9] предложены несколько вариантов алгоритма получения трехмерной модели сцены в виде набора отрезков прямых в пространстве на основе результатов работы базового алгоритма (набора прямых в пространстве).

Также, в [9] был выполнен анализ и определение лучшего варианта предлагаемого алгоритма с точек зрения вычислительной сложности и качества реконструкции, и был предложен критерий остановки итераций алгоритма, позволяющий устанавливать требуемый уровень качества реконструкции.

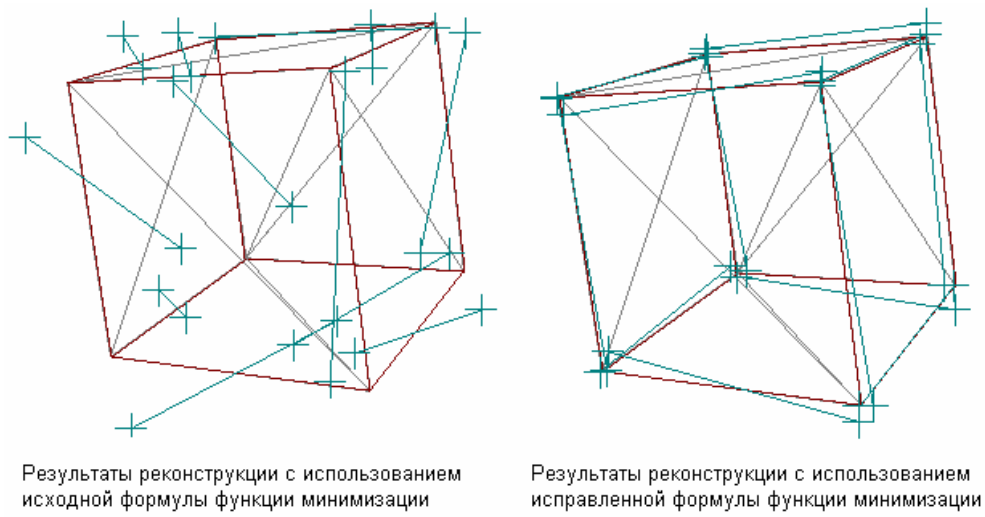


Рисунок 4 — Качественное изменение результатов реконструкции за счет исправления ошибки в формуле функции минимизации

Структура базового алгоритма позволяет распараллелить вычисления, выполняемые на этапах итераций минимизации. Оценка коэффициента ускорения и предлагаемая схема распараллеливания базового алгоритма представлены авторами в [10]. Коэффициент ускорения алгоритма при распараллеливании вычислений зависит от порядка используемого метода минимизации и является наибольшим при использовании методов 0-го порядка. Результаты исследований модификаций приведены в таблице 1.

Таблица 1 — Ускорение базового алгоритма при распараллеливании вычислений в зависимости от порядка метода минимизации

Порядок метода минимизации	Коэффициент ускорения
0-й (использование в процессе минимизации только значений целевой функции)	$C_1 * N * M$
1-й (использование градиента — вектора 1-х производных целевой функции)	$C_2 * N + C_3 * M$
2-й (использование матрице Гессе — матрицы вторых производных целевой функции)	$((C_2 * N + C_3 * M)^2)$

В таблице 1:  $N$  — количество исходных фотоизображений,  $M$  — количество реконструируемых линий,  $C_1$  — к-т, определяющий ускорение в расчете на 1 проекцию линии в 1-м фотоизображении,  $C_2$  — к-т, определяющий ускорение в расчете на 1 исходное фотоизображение,  $C_3$  — к-т, определяющий ускорение в расчете на 1 реконструируемую линию.

### **Выводы**

В данной работе были рассмотрены итоги исследований возможности модификаций базового алгоритма реконструкции трехмерных сцен по их фотоизображениям, повышающих временные и качественные характеристики алгоритма [8, 10]:

- 1) определен лучший с вычислительной точки зрения способ представления вращения в пространстве, позволяющий повысить производительность алгоритма;
- 2) определена возможность модификации целевой функции с целью повышения чувствительности алгоритма к изменениям параметров отклонения реконструируемой модели от опорных элементов, и, соответственно, улучшения качества реконструкции;
- 3) предложены варианты алгоритма преобразования модели, получаемой базовым алгоритмом, в традиционный для современных систем визуализации вид, и, среди них, определен лучший с вычислительной и качественной точек зрения алгоритм;
- 4) предложен критерий остановки итераций алгоритма, позволяющий устанавливать требуемый уровень качества реконструкции;
- 5) предложена общая структура параллельной вычислительной системы для реализации этапов локальных итераций алгоритма с целью ускорения работы алгоритма.

Следует отметить, что целесообразно продолжить исследования в следующих направлениях:

- 1) исключение случайной составляющей из процесса генерации начальных значений параметров минимизации;
- 2) группировка этапов итераций алгоритма и выполнение итераций на группах этапов;
- 3) разработка архитектуры вычислительной системы на основе результирующего алгоритма и возможности распараллеливания вычислений и исследование ее характеристик.

### *Литература*

1. S. Ulman. The Interpretation of Visual Motion. The MIT Press, Cambridge, MA, 1979.
2. C.J. Taylor and D.J. Kriegman. Structure and Motion from Line Segments in Multiple Images. IEEE Trans. Pattern Anal. Machine Intell., 17(11), November 1995.
3. A. Bartoli, P. Sturm. Multiple-View Structure and Motion From Line Correspondences. Proceedings of the IEEE International Conference on Computer Vision, Nice, France, October 2003.
4. A. Azarbayejani and A. P. Pentland, Recursive estimation of motion, structure, and focal length, IEEE Trans. Pattern Analysis and Machine Vision, 17(6): 562–575, 1995.
5. E. Grossmann and J. Santos-Victor. Maximum likelihood 3D reconstruction from one or more images under geometric constraints. BMVC 2002, Cardiff.
6. Z. Zhang and Y. Shan. Incremental Motion Estimation Through Local Bundle Adjustment. Technical Report MSR-TR-01-54, Microsoft Research, Microsoft Corporation One Microsoft Way, Redmond, WA 98052, May 2001.
7. Ковальский С.В., Зори С.А. Современные алгоритмы извлечения геометрической структуры объектов из фотографических снимков для пространственного моделирования городских ландшафтов. В кн.: Наукові праці Донецького національного технічного університету. Серія "Проблеми моделювання та автоматизації проектування динамічних систем" (МАП–2004). Випуск: 74 — Донецьк: ДонНТУ, 2004 — 7 с.
8. С.В. Ковальский, С.А. Зори. Исследование алгоритма извлечения трехмерной структуры объектов из их фотоизображений для реконструкции геометрических моделей сцен городских ландшафтов. В кн. Наукові праці Донецького державного технічного університету. Серія: Інформатика, кібернетика та обчислювальна техніка (ИКВТ–2005). — Донецьк: ДонНТУ, 2005 (находится в процессе публикации).
9. С.В. Ковальский, С.А. Зори. Реконструкция отрезков в алгоритме извлечения трехмерной структуры сцены из ее фотоизображений. Практика и перспективы развития партнерства в сфере высшей школы: Материалы шестого научно-практического семинара, г. Донецк, 27–29 апреля 2005 г. — Донецк, ДонНТУ, 2005.
10. С.В. Ковальский, С.А. Зори. Исследование возможностей распараллеливания вычислений в алгоритме извлечения трехмерной структуры сцены из ее фотоизображений. Роботи VII Міжнародної молодіжної науково-практичної конференції “Людина і космос”, м. Дніпропетровськ, НЦЛОМУ, 2005.

## МЕТОДЫ И СРЕДСТВА РАБОТЫ С ГРАФИЧЕСКИМИ МОДЕЛЯМИ В САПР ПАРОГАЗОВЫХ УСТАНОВОК СПРУТ

А.В. Григорьев  
Кафедра ПМиИ, ДонНТУ  
grigorie@pmi.donetsk.ua

### **Анотація**

*У роботі приведені методи і засоби роботи з графічними моделями, реалізовані в САПР паро-газових установок СПРУТ. Дані засоби дозволяють розширити концепцію інструментальної оболонки для побудови інтелектуальних САПР.*

### **Введение**

Среди многочисленных проблемно-ориентированных САПР, имеющих в настоящее время, можно выделить интеллектуальные САПР, обладающими теми или иными интеллектуальными возможностями, повышающими эффективность труда проектировщиков. Можно назвать для примера комплексы T-FLEX [1], AUTOCAD, семейство пакетов САПР фирмы Autodesk, ряд архитектурных систем и прочие. Анализ достоинств рассмотренных пакетов позволяет сформировать следующий набор требований к САПР, обеспечивающий необходимый уровень стандартных возможностей:

- необходима внутренняя математическая модель;
- средства формирования различных видов и срезов;
- проектирование на основе "интеллектуальных" элементов;
- параметрическое проектирование;
- взаимосвязанность моделей и рисунков;
- поиск следующего примитива;
- регулирование видимости примитивов (скрытый/видимый);
- моделирование поверхностей;
- тонирование поверхностей;
- необходимость избежать возможных коллизий, учесть необходимые зоны изоляции, проделать прочностные расчеты и т.д.

Вместе с тем, главный недостаток рассмотренных выше систем с точки зрения "интеллектуализации" процесса проектирования и создания новых проблемно-ориентированных САПР заключается в том, что они не учитывают наличие ряда эвристических методик проектирования, в том числе и реконструкции, не позволяют создавать базы знаний для любой

предметной области, обучаться на небольшом числе сложных апробированных решений — прототипов и т.д.

Автором в работах [2–7] предложена концепция мета-эвристической оболочки (МЭО), ориентированной на автоматизацию построения интеллектуальных САПР, направленной на устранение данных недостатков.

Следует отметить также, что ранее автором был разработан интеллектуальный проблемно-ориентированный САПР, предназначенный для проектирования и реконструкции паро-газовых установок (ПГУ) в энергетике — СПРУТ [8–14]. СПРУТ по своим возможностям являлся одновременно как прототипом интеллектуального САПР, создаваемого в МЭО, так и — полигоном для приложения возможностей МЭО. Т.о., система СПРУТ может рассматриваться как:

1) САПР, построенная на общих принципах, на которых создаются САПР в МЭО, в частности по таким пунктам:

- создание системы постпроцессоров для связи с другими САПР, для выполнения задач документирования, редактирования, модельного анализа моделей ПГУ, построенных в СПРУТ;
- использование программирования в ограничениях (вычислительные недоопределенные сети) для автоматизации формирования моделей объектов проектирования в МЭО;

2) САПР, используемая для построения над ним интеллектуальной надстройки, т.е. системы синтеза программ на внутреннем языке системы СПРУТ, т.е. языке ЛИСП, в частности:

- имеется одна из версий МЭО, ориентированная на синтез моделей ПГУ на ЛИСП;
- имеется расширение СПРУТ, ориентированное на синтез моделей ПГУ на ЛИСП;

3) САПР, имеющая ряд возможностей, не описанных ранее в работах по теории построения МЭО, т.е. дополняющих общую концепцию МЭО, в частности:

- работа с полным комплексом модельных уровней представления объектов проектирования.
- решение задач реконструкции;
- методы и средства работы с графическими объектами.

В связи с этим, СПРУТ может быть исследован на эффективность:

- 1) методов организации САПР, принятый в МЭО;
- 2) методов организации интеллектуальных надстроек над проблемно-ориентированными САПР.

Целью статьи является изложение ряда положений, доопределяющих концепцию мета-оболочек методами и средствами, имеющимися в СПРУТ, т.е.: графика, чертежи; вычислительные модели; интерфейсы, постпроцессоры, комплекс проблемно-ориентированных САПР.

### ***1. Общая характеристика САПР СПРУТ***

Перечислим важнейшие элементы концепции МЭО, реализованные в СПРУТ:

- наличие ряда интерпретаторов (постпроцессоров) внутренней семантической модели во внешний язык представления моделей систем моделирования различных САПР;
- представление внутренней модели семантической сетью фреймов, рассматриваемой в комплексе с библиотекой базовых соответствий (ББС) как вычислительная модель, обеспечивающая возможность редуцирования неопределенностей во внутренней модели за счет работы вычислительной сети;
- наличие грамматики языка внутреннего представления объекта проектирования, т.е. ПГУ, заданной как вычислительной модели на основе средств ЛИСПА;
- использование распознающей системы для построения моделей ранее спроектированных объектов по разработанному варианту структурно-лингвистического метода.

Общая структура базы данных (БД) системы СПРУТ (рис.1) включает в себя: БД чертежей, БД элементов данного чертежа, БД типов элементов. БД типов элементов чертежа включает в себя отдельные библиотеки для различных классов элементов, в частности, это: трубы, арматура для труб, оборудование, строительные конструкции. Компонент базы данных элементов чертежа включает в себя две составляющие: вычислительную топологическую модель (ВТМ) и графическую модель.

Модель проектируемого объекта в СПРУТ представляет собой два файла, собственно графическую модель чертежа, т.е. DWG-файл, и математическую модель объекта. Для формального языкового представления видеообразов элементов и чертежа в целом используется подмножество языка АВТОЛИСП.

### ***2. Особенности представления графических моделей***

Структура графических моделей для различных типов элементов, входящих в различные библиотеки, может быть показана на примере моделей труб. Графическая модель трубы любого типа включает в себя такие составляющие:

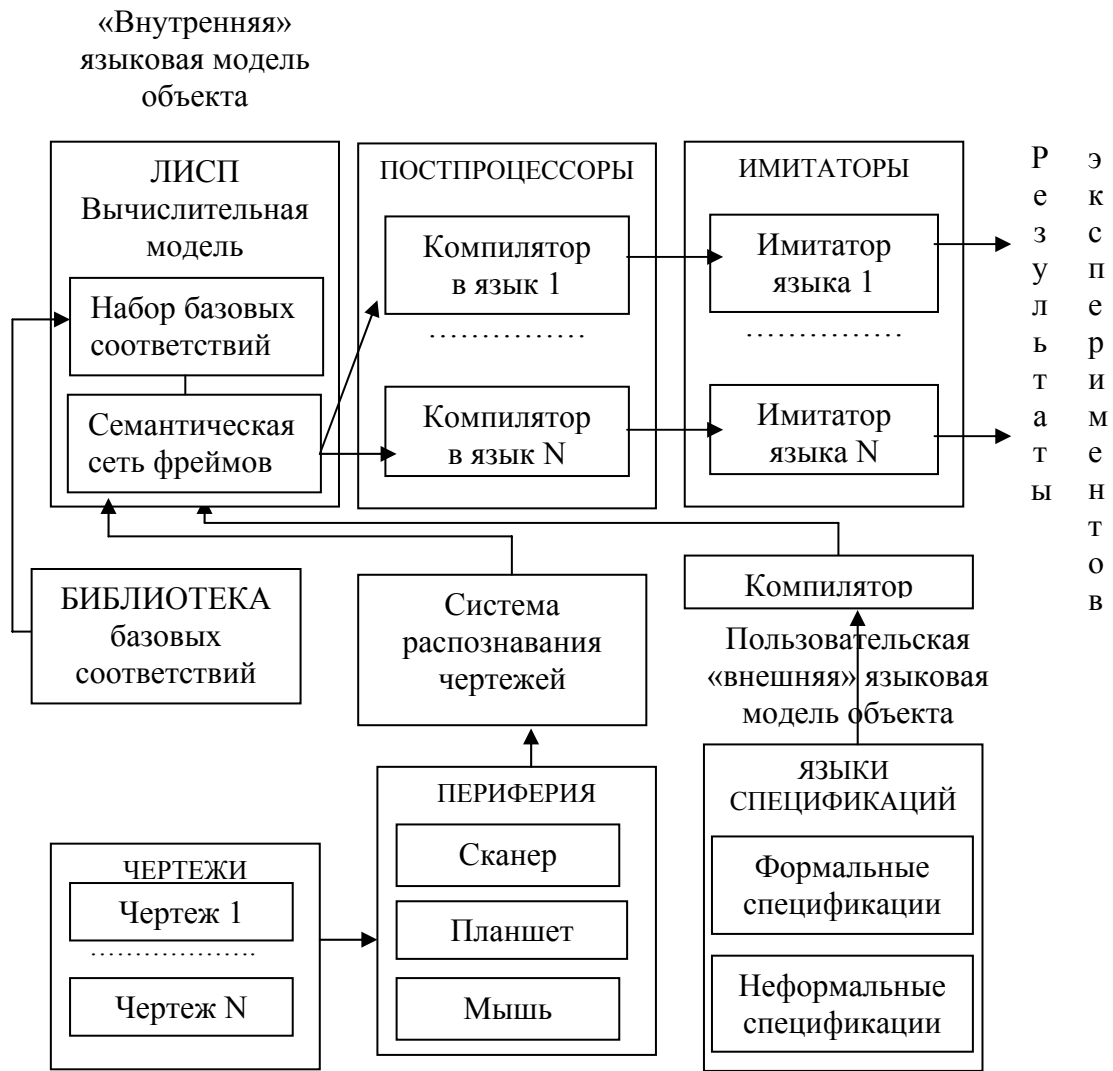


Рисунок 1 — Общая структура САПР СПРУТ

- список из списков свойств характерных точек изображения, в данном случае — концов труб; список свойств включает: трехмерные координаты, нормаль и радиус конца, наличие фланца (часть топологической модели);
- список плоских 4-угольных поверхностей, составляющих изображение; каждая поверхность представлена списком своих свойств, куда входят трехмерные координаты углов и признаки видимости/невидимости отдельных ребер;
- функциональную модель зависимости количества 4-гранных поверхностей и их свойств от значений элементов списка свойств характерных точек;
- механизм отрисовки поверхностей по их свойствам.
- вектор направления точки зрения на чертеж;



– и вектора точки зрения на объект.

Имеется библиотека модулей синтеза проекций трехмерных объектов с заданной точки зрения по характерным точкам изображения и определенным между ними связям. "Синтез изображений осуществляется на основе "сборки" проекции из плоских четырехугольников с видимыми и невидимыми краями. Кривые аппроксимируются отрезками, представляющими собой видимые грани четырехугольников, с некоторым заданным шагом.

### *3. Язык представления чертежей*

ЛИСП-файл, в котором хранятся данные, представлен в виде S-выражений. Имеется 4 раздела, которые содержат информацию о трубах, оборудовании, арматуре и точках, которые характеризуют координаты объектов. Каждый раздел начинается с двух строк: заголовок раздела; максимальный номер объекта. Заголовок раздела указывает, о каких объектах хранятся данные в этом разделе:

трубы ----- "TRUB'S-->";  
точки ----- "POINT---->";  
оборудование ----- "OBR----->";  
арматура ----- "ARM----->".

Структура S-выражения для хранения данных о точке задана следующим образом: (**num** (**X** **Y**) **Z** **DIAM**), где

**num** — номер точки;  
**X, Y, Z** — координаты точки в пространстве;  
**DIAM** — диаметр трубы в этой точке.

Структуру S-выражения для хранения данных о трубе можно представить следующим образом:

(**num type** (**num\_end1 n\_point** (**con trub**)) (**num\_end2 n\_point** (**con trub**))),

где **num** — номер трубы;  
**type** — тип трубы;  
**num\_end1, num\_end2** — номера концов трубы;  
**n\_point** — номер точки для конца трубы;  
**con** — тип связи в этой точке;  
**trub** — номер следующей трубы.

Если тип связи и номер следующей трубы на каком-либо конце не заданы ('nil'), то считается что труба на этом конце не имеет соединения.

Данные об оборудовании и арматуре хранятся в виде S-выражения следующего вида: (**num** "type" **point\_1 point\_2 point\_3 ..... point\_n**),

где **num** — номер элемента;

**type** — тип элемента;

**point\_x** — номер точки соединения.

Признаками или характерными свойствами видеообразов считаются координаты точек стыков между отдельными элементами чертежа, нормали направления стыков, тип стыка (для труб это сварка или фланец), размер стыка (для труб это диаметр трубы). Стыки одного элемента объединяются в целое логической связью. Для координирования признаков и связей используется механизм вычислительных моделей. Это позволяет косвенно распознавать признаки одних элементов чертежа через признаки других.

#### ***4. Особенности представления вычислительных топологических моделей***

Рассмотрим методы параметризации чертежей и математических моделей, принятые в СПРУТ. Различаются такие типы вычислительных топологических моделей для труб: 1) соответствия стыков двух труб, связанных между собой; 2) соответствия стыков одной трубы. Вычислительная *топологическая* модель труб включает: 1) список из списков свойств концов труб, таких как: координаты, нормаль и радиус конца, наличие фланца; 2) функциональную модель взаимной зависимости элементов списков.

Для случая использования вычислительной сети для расчета геометрических и территориальных свойств элементов структурной модели имеются отличия в способе ее трактовки и применения.

Вычислительная модель используется нестандартным путем — не для поиска одного возможного пути решения задачи от данных условий до цели и последующего синтеза программы по структуре и набору базовых соответствий, а для доопределения модели. Условие задачи ("дано") в данном случае объединяет частично определенные параметры элементов трассы труб, которые могут располагаться как в начале трассы, так и в любом ее месте. Общей целью ("получить"), входящей в постановку задачи, в этом случае, является доопределение всех неопределенных свойств модели до конца. Это позволит перейти в дальнейшем к формированию программы.

Программа, которую надо построить в нашем случае, это описание структуры объекта на языке графических образов, т.е. языке описания чертежа.

Язык описания чертежа представляет собой некоторый обобщенный список, включающий в себя в качестве подсписков такие списки: 1) список элементов; 2) список свойств элементов; 3) список связей между элементами. Данный язык списков ссылается на

библиотеку базовых соответствий для различных типов связей между элементами. Программа на данном языке по существу и является основой вычислительной модели, поскольку задает все возможные пути определения неопределенных параметров по ранее определенным.

Процесс моделирования на вычислительной сети в такой форме описания представляет собой итерационный процесс многократного прохождения по трассе трубопровода, который заканчивается в том случае, если при прохождении по трассе больше не возможно доопределить не одно свойство элементов. Возможна постановка и частичной цели решения задачи, т.е. отрисовка только части изображений для элементов, имеющих полностью определенный набор геометрических и территориальных соответствий. То есть предполагается, что степень доопределения модели может быть любая. Момент вызова доопределения может специально выбираться пользователем по мере его желания. Имеется интерпретатор с этого языка в командный язык AUTOCADa. Интерпретатор использует библиотеку графических модулей для элементов трассы.

### 5. Программная реализация СПРУТ

Общее меню программы и состав команд для работы с внутренней математической моделью объекта показаны на рис. 2 и 3.

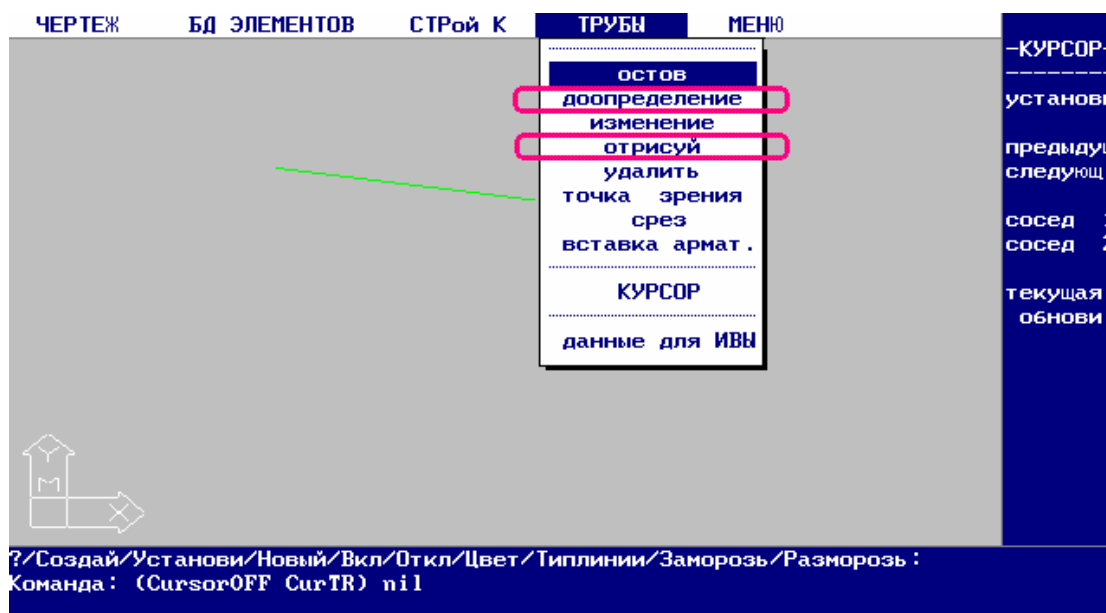


Рисунок 2 — Командное меню системы СПРУТ

«Отрисовать» — группа команд синтеза изображения трубопровода по заданному остову. «Доопределить» — группа команд

доопределения типа и параметров отдельных труб и арматур, составляющих остов. Доопределение вызывается автоматически, если невозможно отрисовать трубу или по вызову пользователя. В случае использование вычислительной сети для расчета функциональных свойств элементов сети целью является сам процесс прослеживания (трассировки) изменения свойств.

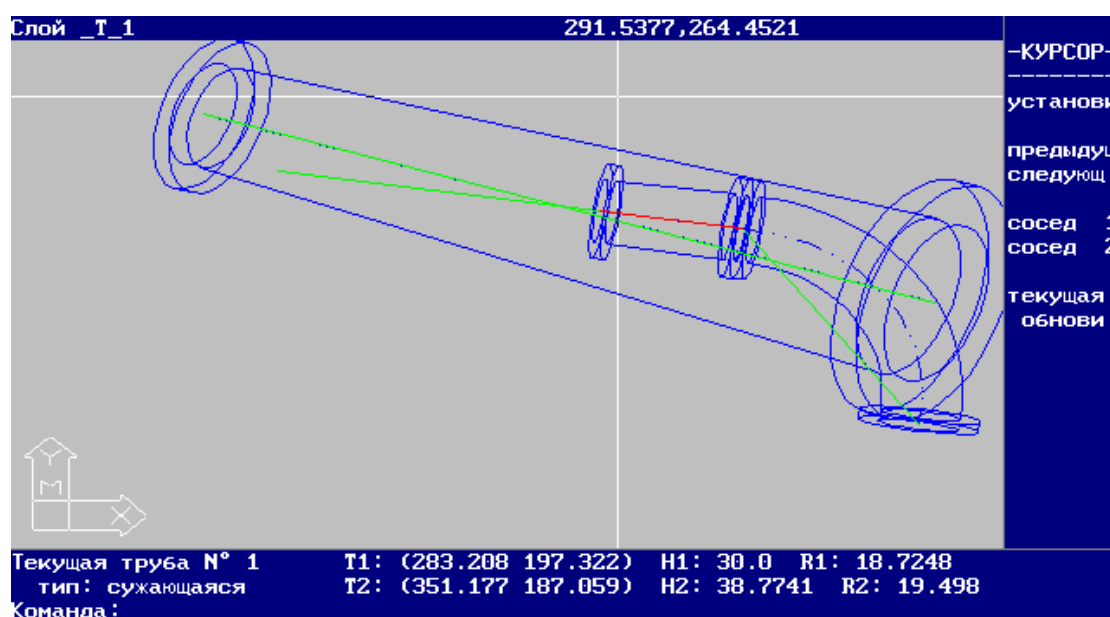


Рисунок 3 — Процесс отрисовки трубопровода по остову

### ***Заключение***

В системе СПРУТ реализовано:

- 1) предлагаемая концепция МЭО реализована на количественном модельном уровне представления знаний об объекте за счет организации ряда интерфейсов с другими проблемно-ориентированными САПР;
- 2) доказана возможность применения данного подхода для структурного и логического уровней представления моделей данной ПО;
- 3) реализован механизм семантических сетей фреймов на языке ЛИСП как внутренняя форма представления моделей объектов, а так же механизм вычислительных сетей, определенный над внутренней формой представления моделей объектов, необходимый для независимого от процессов продукционного вывода редуцирования неопределенностей в модели;
- 4) реализована та часть предлагаемой общей структуры БЗД, которая связана с библиотекой БС, как части внутренней модели объекта;
- 5) на основе данного подхода дано новое решение задачи реконструкции сложных технических объектов;

6) предложен вариант нового структурно-лингвистического метода распознавания видеобразов, применяемый для построения моделей ранее спроектированных объектов, ориентированный на работу в условиях погрешностей выполненных вручную чертежей и их большого количества, отличающийся грамматикой формального языка для описания признаков изображений и их связей, выбором набора признаков и подходом к трактовке семантических образов элементов;

7) на примере организации интерфейса с системами ИВА/ASTRA реализован метод создания ряда постпроцессоров с языка внутреннего представления модели в языки внешних систем моделирования для анализа синтезированной модели объекта.

8) предложен новый тип вычислительных моделей, отличающийся:

- формой представления исходных и выходных параметров задачи,
- способом формирования и назначением синтезируемой программы, ориентированных на доопределение геометрических и территориальных свойств элементов моделей объектов количественного уровня САПР по ранее определенным свойствам с целью формирования описания их чертежей.

Предлагаемые решения позволяют расширить концепцию МЭО за счет методов и средств работы с графическими объектами.

### *Литература*

1. Профессиональная система автоматизированного проектирования, черчения и твердотельного моделирования с мощными параметрическими возможностями // [www.tfex.ru](http://www.tfex.ru).
2. Григорьев А.В. Унифицированная концептуальная модель предметной области. В кн. Информатика, кибернетика и вычислительная техника (ИКВТ-97). Сборник трудов ДонГТУ, Выпуск 1. — Донецк: ДонГТУ, 1997. — С.218–224.
3. Григорьев А.В. Семиотическая модель базы знаний САПР. Научные труды Донецкого государственного технического университета Серия: Проблемы моделирования и автоматизации проектирования динамических систем, выпуск 10: — Донецк, ДонГТУ, 1999. — С. 30–37.
4. Григорьев А.В. Теоретико-множественные операции над грамматиками как механизм работы со знаниями в интеллектуальных САПР. Вісник Східноукраїнського національного університету імені Володимира Даля, N 2(48). — Луганск, ВУТУ, 2002. — С. 186–194.
5. Григорьев А.В., Каспаров А.А. И/ИЛИ-дерево как средство абстрактного представления знаний. Наукові праці національного технічного університету.

- Серія «Інформатика, кібернетика та обчислювальна техніка». Випуск 39: — Донецьк: ДонНТУ, 2002. — С.36–42.
6. Григорьев А.В. Адаптивная система ограничений на сложность при синтезе новых решений в интеллектуальных САПР. Искусственный интеллект. N 2, 2001. — С. 152–167.
  7. А.В. Григорьев. Организация пространства поиска решений в специализированной оболочке для создания интеллектуальных САПР. Вісник ТРТУ-ДонГТУ. Матеріали 2-го міжд. н.-т. семінара "Практика и перспективы институционального партнерства". — Донецк, ДонГТУ, 2001, N1. — С.57–68.
  8. Григорьев А.В. Организация проблемно-ориентированной базы знаний в САПР трубопроводов. Международная научно-методическая конференция "Автоматизация проектирования и производства в машиностроении". — Киев.: ИСИ ОУ, 1995. — С. 49–50.
  9. Григор'ів О. Один підхід до організації баз даних і знань у проблемно-орієнтованих САПР. 5-та Міжнародна науково-практична конференція "Україномовне програмне забезпечення УкрСофт-95". — Львів.: ДУ "Львівська політехніка", 1995. — С. 100.
  10. Григорьев А.В. Автоматизированная система проектирования трубопроводов. Доклады региональной научной конференции "Творческое наследие В.И. Вернадского и современность". Секция 4 "Актуальные проблемы вычислительной техники, информатики и энергетики". Часть 1. — Донецк: ДонГТУ, 1995. — С.49–50.
  11. Григорьев А.В. Метод распознавания образов для задач реконструкции сложных технических систем. В кн. "Современные проблемы машиностроения и технический прогресс. Тезисы докладов международной научно-технической конференции. 10–13 сентября 1996 г., г. Севастополь. — Донецк: ДонГТУ. — С. 59.
  12. Григорьев А.В., Базалей А.О., Юрченко С.В. Особенности реализации системы автоматизации построения интеллектуальных САПР и АСНИ. В кн. "Современные проблемы машиностроения и технический прогресс. Тезисы докладов международной научно-технической конференции. 10–13 сентября 1996г., г. Севастополь. — Донецк: ДонГТУ. — С. 60.
  13. Григорьев А.В. Распознавание образов в задачах реконструкции сложных технических систем. В кн. "Сборник трудов факультета вычислительной техники и информатики. Выпуск 1. Донецкий государственный технический университет. — Донецк: ДонГТУ, 1996. — С. 153–156.
  14. Григорьев А.В. Инвариантные интеллектуальные средства САПР и АСНИ. В кн. "Сборник трудов факультета вычислительной техники и информатики. Выпуск 1. Донецкий государственный технический университет. — Донецк: ДонГТУ, 1996. — С. 157–160.

**ПЛАНИРОВАНИЕ ЗАДАЧ ПРИ  
МОДЕЛИРОВАНИИ ЦИФРОВЫХ УСТРОЙСТВ  
НА МНОГОПРОЦЕССОРНЫХ СИСТЕМАХ**

В.Н. Белецкий, С.А. Резникова, А.А. Чемерис  
Ин-т проблем моделирования в энергетике им. Г.Е. Пухова  
НАН Украины, г. Киев

***Анотація***

*Розглянуто алгоритм горизонтального планування завдань при моделюванні поведінки цифрових пристроїв, що проектується, на багатопроцесорних комп'ютерах і який спрямовано на підтримку принципів рівномірного завантаження процесорів системи і мінімізації передач даних.*

Проектирование цифровых схем в современных технологических процессах осуществляются на основе моделей, которые представляют на языках описания аппаратуры высокого уровня, например, VHDL, Verilog, SystemC и т.д. Известно, что процесс моделирования занимает значительную часть процесса проектирования, выполняется на различных уровнях проекта и носит итерационный характер. Таким образом, для уменьшения времени выхода на рынок проектируемого устройства процесс моделирования представляет большой резерв экономии. Одним из подходов ускорения проектирования есть использование многопроцессорных вычислительных систем, в которых моделирование проектируемого устройств распределяется по нескольким процессорам и в идеальном случае время моделирования уменьшается в  $P$  раз, где  $P$  — число процессоров в системе.

Задача планирования вычислений, состоящая в назначении заданий на процессоры многопроцессорной системы таким образом, чтобы свести время выполнения алгоритма к минимуму, остается актуальной, несмотря на быстрое развитие средств вычислительной техники. Существуют точные и эвристические алгоритмы планирования заданий. Первые, являясь разновидностями методов перебора, требуют большого количества времени. Вторые, являясь достаточно быстрыми, дают решение с некоторой погрешностью, величина которой зависит от качества алгоритма планирования. В общем случае задача планирования является NP-полной за исключением некоторых отдельных случаев [1].

В этой статье мы фокусируем внимание на статических методах планирования. Известны эвристические методы планирования, которые основаны на упрощении описаний структур алгоритма и вычислительной системы [2,3]. Однако на практике применяется несколько простых алгоритмов, таких как Round robin [4].

Представляемый метод планирования есть частью системы автоматизированного проектирования цифровых устройств, которая рассчитана на использование языка VHDL. Компилятор с VHDL генерирует систему булевых функций, которая описывает проектируемое устройство и является входом для программ моделирования. Процесс моделирования состоит в по-тактном определении значений переменных, которые входят в систему булевых функций. Для реализации процесса моделирования на многопроцессорном компьютере строится граф системы функций и производится разбиение по числу процессоров. При этом производится попытка совместить два принципа эффективного планирования задач: 1) равномерно загрузить процессоры системы и 2) минимизировать пересылки данных.

#### ***Алгоритм горизонтального планирования***

Рассмотрим алгоритм горизонтального планирования, который относится к классу эвристических методов. Идея планирования состоит в том, что для каждого яруса графа вершины планируются таким образом, чтобы нагрузка на каждый процессор системы была приблизительно равной с учетом минимизации межпроцессорных обменов. При этом полагаем, что программа представлена графом, вершины которого распределены по ярусам и каждой определена вычислительная стоимость. Перед выполнением планирования необходимо задать топологию многопроцессорной системы.

#### ***Алгоритм:***

- Распределение вершин по ярусам
- Назначение вершин первого яруса графа процессорам
- Далее для каждого яруса, начиная со второго
  - Определение средней вычислительной нагрузки для яруса
  - Последовательный просмотр всех вершин яруса
    - Определение всех вершин графа, которые связаны с рассматриваемой вершиной по входу
      - Определение номера процессоров, на которые назначены эти вершины



- Назначение рассматриваемой вершины последовательно на процессоры системы
- Для каждого назначения вычисление цены межпроцессорной связи  $C$ , зависящей от архитектуры многопроцессорной системы
- Определение минимальной цены межпроцессорных связей  $C_{min}$
- Назначение вершины на процессор с минимальной ценой межпроцессорных связей и, если вычислительная нагрузка вершины станет равной или большей средней по ярусу, исключение данного процессора из дальнейшего рассмотрения
- Конец.

Особенности предлагаемого алгоритма состоят в следующем. Во-первых, ярусы алгоритма рассматриваются отдельно друг от друга. Однако ярус  $j > i$  не может быть рассмотрен раньше яруса  $i$ . Во-вторых, при вычислении минимальной цены межпроцессорного обмена рассматриваются только те процессоры, которые не добрали вычислительную нагрузку.

### **Пример.**

В качестве примера рассмотрим модель двухразрядного сдвигового регистра. Система булевых функций имеет вид:

$$\begin{array}{ll}
 Q(t,1)=S(1)+(!R(1)*Q(t-1,1)); & Q(t,2)=S(2)+(!R(2)*Q(t-1,2)); \\
 S(1)=C*Q1(t-1,1); & S(2)=C*Q1(t-1,2); \\
 R(1)=C*!Q1(t-1,1); & R(2)=C*!Q1(t-1,2); \\
 Q1(t,1)=S1(1)+(!R1(1)*Q1(t-1,1)); & Q1(t,2)=S1(2)+(!R1(2)*Q1(t-1,2)); \\
 S1(1)=D(1)*!C; & S1(2)=D(2)*!C; \\
 R1(1)=!C*!D(1); & R1(2)=!C*!D(2); \\
 D(2)=Q(t-1,1); & D(1)=!Q(t-1,2);
 \end{array}$$

Здесь введены следующие обозначения:  $*$ ,  $+$ ,  $!$  — соответственно, операции булевой алгебры И, ИЛИ, НЕ;  $Q(t)$ ,  $Q(t-1)$  — соответственно, значения переменной  $Q$ , которая представляет выход триггера, на текущем и на предыдущем тактах. В круглых скобках указывается номер разряда (по аналогии с массивами в языках высокого уровня).

Обработка компилятором с учетом планирования заданий дает результат, представленный в табл. 1. Стрелками обозначены пересылки данных.

Таблица 1

№ ячейки	Процессор 1		Процессор 2
1	2	3	4
1	I: IN C		I: IN C
2	IN Q1[t-1,1]		IN NOT Q1[t-1,1]
3	AND S[1]		AND R[1]
4	I: IN C		I: IN C
5	IN NOT Q1[t-1,2]		IN Q1[t-1,2]
6	AND R[2]		AND S[2]
7	IN Q[t-1,1]		IN NOT Q[t-1,2]
8	AND D[2]		AND D[1]
9	IN R[2]		noop
10	SR: AND R[2]		noop
11	noop		RL: GET *R[2]
12	noop		IN *R[2]
13	noop		AND R[2]
14	noop		IN R[1]
15	noop		SL: AND R[1]
16	RR: GET *R[1]		IN NOT R[2]
17	IN *R[1]		IN Q[t-1,2]
18	AND R[1]		AND #TEMP1
19	IN NOT R[1]		IN S[2]
20	IN Q[t-1,1]		IN #TEMP1
21	AND #TEMP1		OR Q[t,2]
22	IN S[1]		O: OR Q[t,2]
23	IN #TEMP1		IN D[1]
24	OR Q[t,1]		I: IN NOT C
25	O: OR Q[t,1]		AND S1[1]
26	IN D[2]		I: IN NOT C
27	I: IN NOT C		IN NOT D[1]
28	AND S1[2]		AND R1[1]
29	I: IN NOT C		IN NOT R1[1]
30	IN NOT D[2]		IN Q1[t-1,1]
31	AND R1[2]		AND #TEMP1
32	IN NOT R1[2]		IN S1[1]
33	IN Q1[t-1,2]		IN #TEMP1
34	AND #TEMP1		F: OR Q1[t,1]
35	IN S1[2]		noop

Таблица 1 — Продолжение

36	IN #TEMP1		noop
37	F: OR Q1[t,2]		noop
38	IN Q1[t,2]		noop
39	SR: AND Q1[t,2]		noop
40	noop		RL: GET *Q1[t,2]
41	noop		IN *Q1[t,2]
42	noop		AND Q1[t,2]
43	noop		IN Q1[t,1]
44	noop		SL: AND Q1[t,1]
45	RR: GET *Q1[t,1]		IN Q[t,2]
46	IN *Q1[t,1]		AND Q[t-1,2]
47	AND Q1[t,1]		IN Q1[t,1]
48	IN Q[t,1]		AND Q1[t-1,1]
49	AND Q[t-1,1]		IN Q1[t,2]
50	IN Q1[t,2]		AND Q1[t-1,2]
51	AND Q1[t-1,2]		noop
52	IN Q1[t,1]		noop
53	AND Q1[t-1,1]		noop
54	END		END

Представленный алгоритм планирования используется для разбиения систем булевых уравнений, которые описывают модель проектируемого устройства. При планировании учитывается архитектура используемого компьютера и принципы равномерности загрузки процессоров и минимизации обменов данными.

### *Литература*

1. Gonzalez, "Deterministic Processor Scheduling," *ACM Computing Surveys*, vol. 9, no. 3, Sep. 1977. — pp. 173–204.
2. T.L. Adam, K. Chandy and J. Dickson, "A Comparison of List Scheduling for Parallel Processing Systems", *Communications of the ACM*, vol. 17, no. 12, Dec. 1974. — pp. 685–690.
3. S.H. Bokhari, "On the Mapping Problem", *IEEE Trans. on Computers*, vol. C-30, Mar. 1981. — pp. 207–214.
4. B. Wilkinson, M. Allen "Parallel programming. Techniques and applications using networked workstations and parallel computers." 2d edition, Prentice Hall, NJ, USA, 2004. —467 p.

## КОЭФФИЦИЕНТ ПОЛЕЗНОЙ ПЕРЕДАЧИ ДАННЫХ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

Н.И. Алишов

Институт кибернетики НАН Украины

Теоретически, время доставки массива информации будет минимальным, когда между источником и приемником данных отсутствуют транзитные узлы (коммутаторы, маршрутизаторы) и пакет не имеет заголовка. В таком случае время доставки данных объемом  $D$  байтов равно  $D$  байт-тактам. Если же пакет данных содержит заголовок, состоящий из  $h$  байтов, то минимальное время его доставки составляет  $D+h$  байт-тактов. Таким образом, при отсутствии транзитных узлов коэффициент полезной передачи данных ( $k_{пнд}$ ) определяется как  $\Xi_0 = \frac{D}{D+h}$ . Когда массив  $D$  данных

транспортируется к адресату посредством  $n$  пакетов, то  $\Xi_n^h = \frac{D}{\left(\frac{D}{n} + h\right)n} = \frac{D}{D+nh}$ .

Для рассматриваемой системы коммутации пакетов  $k_{пнд}$  будем определять как  $\Xi_r^h = \frac{D}{T_n^D} = \frac{D}{\left(\frac{D}{n} + h\right)(r+n)}$ . Действительно, если полагать,

что  $\frac{D}{\left(\frac{D}{n} + h\right)(r+n)} = \frac{D}{D+nh}$ , и вычислить значение величины  $r$ , при

котором данное равенство соблюдается, то получим

$\frac{D+nh}{\left(\frac{D}{n} + h\right)(r+n)} = 1$ ;  $\frac{n}{r+n} = 1$ , т.е.  $r=0$ . Когда массив данных объемом  $D$

байтов передается по транспортной системе за минимальное время,

$\Xi^{n_0} = \frac{D}{\left(\sqrt{hr} + \sqrt{D}\right)^2}$ . Поэтому величина  $\Xi$  характеризует степень потерь

во времени доставки данных из-за таких необходимых непроизводительных затрат, как передача заголовка пакета и коммутация самих пакетов. Поскольку  $rh$ -оптимизация подразумевает выбор приемлемых для приложений значений величины  $n$ , то отношение

$\frac{\left(\sqrt{hr} + \sqrt{D}\right)^2}{\left(\frac{D}{n} + h\right)(r+n)}$  будет наилучшим критерием для практических оценок

$k_{пнд}$  (см. рис. 1).

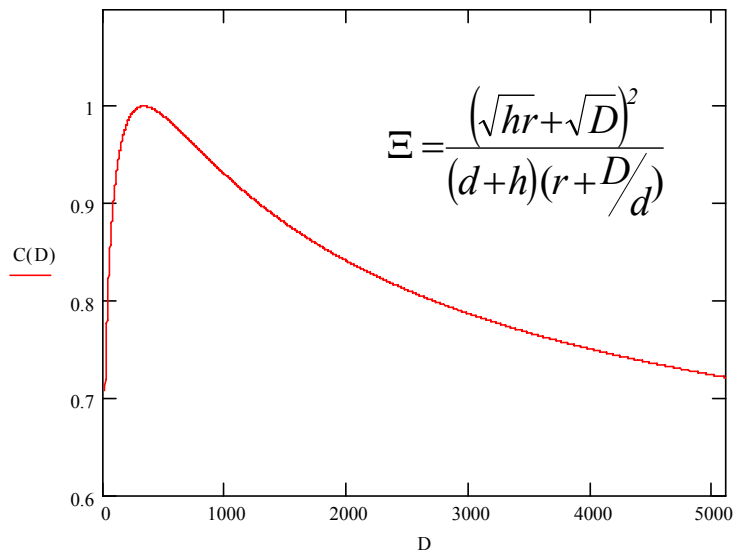


Рисунок 1 — Характеристика степени потерь во времени доставки данных

Это связано с тем, что как в формуле  $\Xi_n^h = \frac{D}{D+nh}$  значение величины  $D$

является предельно достижимым, так же в формуле  $\Xi = \frac{(\sqrt{hr} + \sqrt{D})^2}{(D/n + h)(r + n)}$

значение выражения  $(\sqrt{hr} + \sqrt{D})^2$  является теоретическим пределом для сетей с коммутацией пакетов. Следует отметить, что в выражении  $(D/n + h)(r + n)$  значение величины  $n$ , которое должно обеспечивать

$\Xi = 1$ , определяется как  $n = \sqrt{\frac{Dr}{h}}$ . На практике это равенство не

всегда выполняется, поскольку в реальных системах возникает необходимость реализовать округление в следующих ситуациях.

1. Если значение  $n$  — дробное число, то оно округляется до ближайшего целого.

2. Если  $D/n$  — не целое, тогда необходимо округлять значение  $n$  до ближайшего целого числа, при котором  $D/n$  становится целым числом.

3. Форматы пакетов в конкретных сетях могут обуславливать заполнение полей данных до границы двоичных слов (8-, 16- или 32-разрядных). В таких случаях к значению  $D/n$  добавляется несколько байтов (до 7-ми, 15-ти и 31-го байтов, соответственно).

Подобное округление приводит к тому, что  $\Xi \neq 1$ .

Рассмотрим пример. Пусть  $D = 1024$ ,  $r = 5$ ,  $h = 5$ . Формат пакета имеет 16-разрядную границу. Тогда  $n = 32$ . Поэтому  $(D/n + h)(r + n) = (\sqrt{hr} + \sqrt{D})^2$  и, следовательно,  $\Xi = 1$ . Если же  $D = 1020$ ,  $r = 5$ ,  $h = 16$ , то  $n = 17,85$ . Выполняем соответствующие округления.

1. До ближайшего целого, т.е.  $n = 18$ .

2.  $1020/18 = 56,66$  – не целое число. Находим ближайшее (к 18!) целое число, на которое величина  $D$  делится нацело, т.е.  $n = 20$ .

3.  $1020/20 = 51$ . Из 64-х необходимых байтов заняты 51, т.е. в каждый пакет добавляем еще по 13 байтов данных.

Таким образом, исходный массив объемом 1020 байтов будет доставляться 16-ю пакетами, 15 из которых содержит по 64 байта полезных данных, а в 16-м пакете — 60 байтов.

Тогда  $(D/n + h)(r + n) = (960/15 + 16)(5 + 15) + (64 + 16) = 1680$  байт-

тактов и при этом  $(\sqrt{hr} + \sqrt{D})^2 = (\sqrt{16 * 5} + \sqrt{1020})^2 = 1671,31$ . Итак,

$\Xi = 1671,31 / 1680 = 0,994$ . А если бы формат пакета не обуславливал

выравнивания поля данных по краю 32-разрядного двоичного слова, получили бы значение  $\Xi = 0,997$ .

При создании средств интеллектуального управления сетями коммутации пакетов с оптимизацией времени доставки массивов данных необходимо реализовать подсистему коррекции параметров с учетом необходимости округления значений величины  $n$  в вышеперечисленных случаях.

В большинстве современных сетей с коммутацией форматы передаваемых пакетов предполагают наличие фиксированного служебного поля ( $h$ ) и поля данных ( $D$ ). Как правило, длина поля данных определяет минимальный и максимальный размер передаваемого пакета. Поэтому в существующих сетях возможности обеспечения минимального времени доставки массивов данных могут быть ограничены.

Попытаемся определить кпд для пакетов данных, в которых размеры полей данных и служебной информации имеют фиксированные значения.

Пусть поле данных в конкретном пакете имеет размер  $f_d = \overline{1, \dots, d}$ . Тогда время передачи массива данных объемом  $D$  байтов  $D/d$  пакетами равно

$$T_{D/d}^D = (d + h) \left( r + \frac{D}{d} \right) \text{ байт-тактам.}$$

Если  $d$  байтов передается  $d/n$  пакетами, то  $T_{d/n}^D = (d/n + h)(r + Dn/d)$ . При этом минимальное время доставки  $d$  байтов составляет  $T_{d/n_0}^D = (d/n_0 + h)(r + Dn_0/d)$  байт-тактов, где  $n_0 = \sqrt{dr/h}$ .

Вычислим значение  $D$ , при котором  $T_{n_0}^D = T_{d/n_0}^D$ . Из равенства  $(d+h)(r+D/d) = (d/n_0+h)(r+Dn_0/d)$  следует, что  $dr + D + hr + hD/d = dr/n_0 + D + hr + hDn_0/d$ . Тогда

$$dr\left(1 - \frac{1}{n_0}\right) = D\left(\frac{hn_0}{d} - \frac{h}{d}\right); \quad dr \frac{n_0 - 1}{n_0} = Dh \frac{n_0 - 1}{d};$$

$$D = \frac{d}{n_0} \frac{dr}{h}.$$

А поскольку  $\frac{dr}{h} = n_0^2$ , то  $D = dn_0$ .

Это означает, что массив данных объемом  $D$  байтов может быть доставлен за минимальное время, если время доставки массива из  $d$  байтов также будет минимальным. Вычислим диапазон значений величины  $D$ , для которых достижима возможность доставки массива информации за минимальное время при условии, что для передачи используется конкретный формат пакетов с ограниченным полем данных ( $d$ ). Для этого необходимо найти такие значения  $D$ , при

которых  $(d+h)(r+D/d) = (\sqrt{hr} + \sqrt{D})^2$ . Если учесть, что  $n = \sqrt{\frac{Dr}{h}}$  и

при этом  $\frac{D}{n} \leq d$ , то из  $D \leq nd \leq \sqrt{\frac{Dr}{h}}$  находим  $D \leq \frac{d^2 r}{h}$ . Таким

образом, для значений  $D = 1, \dots, \frac{d^2 r}{h}$  справедливо равенство

$(d+h)(r+D/d) = (\sqrt{hr} + \sqrt{D})^2$  при  $d = D/n$ . В результате получаем

следующую формулу для  $knpd$ :

$$\Xi = \begin{cases} 1 & \text{для } D = 1, \dots, d^2 \frac{r}{h}; \\ \frac{(\sqrt{hr} + \sqrt{D})^2}{(d+h)(r+D/d)} & \text{для } D > d^2 \frac{r}{h}. \end{cases}$$

Определим минимальное значение  $\Xi$  для массива информации объемом  $D$ . Для этого необходимо вычислить

$$\lim_{D \rightarrow \infty} \frac{(\sqrt{hr} + \sqrt{D})^2}{(d+h)(r+D/d)} = \frac{hr + 2\sqrt{hrD} + D}{dr + D + hr + hD/d} = \frac{\frac{hr}{D} + \frac{2\sqrt{hrD}}{D} + 1}{\frac{dr}{D} + 1 + \frac{hr}{D} + h/d} = \frac{d}{d+h}.$$

$$\text{Таким образом, } \min \Xi \geq \frac{d}{d+h}.$$

Для практических расчетов, а также для алгоритмической реализации вариантов оптимизации коммутации пакетов в транспортной среде целесообразным будет определение минимальной границы  $knpd$  по формуле

$$\Xi^{n_0} = \frac{D}{(\sqrt{hr} + \sqrt{D})^2} = \frac{d^2 r/h}{hr + 2dr + d^2 r/h} = \frac{d^2}{h^2 + 2dh + d^2} = \frac{d^2}{(d+h)^2} = \frac{d}{d+h}.$$

При  $h=0$  необходимо установить  $n := D$ . Тогда

$$T_D^D = r + D; \quad \Xi = \frac{D}{r+D} = \frac{d^2 r/h}{r + d^2 r/h} = \frac{d^2}{d^2 + h}.$$

При  $r=0$  следует установить  $n := 1$ . В этом случае

$$T_D^D = D + h; \quad \Xi = \frac{D}{D+h} = \frac{d^2 r/h}{d^2 r/h + h} = \frac{d^2 r}{d^2 r + h^2}.$$

Вычислим значение величины  $d$ , при котором выполняется равенство  $(d+h)(r+D/d) = (\sqrt{hr} + \sqrt{D})^2$ .

$$dr + D + hr + hD/d = hr + 2\sqrt{hrD} + D; \quad \frac{d^2 r + hD}{d} = 2\sqrt{hrD};$$

$$d^4 r^2 + 2d^2 rhD + h^2 D^2 = 4d^2 rhD; \quad (d^2 r - hD)^2 = 0; \quad d = \sqrt{\frac{hD}{r}}.$$

### Литература

1. Н.И. Алишов. Адаптивный стековый алгоритм универсального множественного доступа в распределенных системах и сетях компьютеров УСиМ, 2004, № 1 — с. 59–72
2. Н.И. Алишов. Оптимизация коммутации пакетов в распределенных системах. Комп'ютерні засоби, мережі та системи, 2004, № 3, с. 87–94.



## МОДЕЛИРОВАНИЕ И ДИАГНОСТИРОВАНИЕ МОП-СТРУКТУР

А.И. Андрюхин  
Кафедра ПМиИ, ДонНТУ  
andr@r5.dgtu.donetsk.ua

### *Анотація*

*У роботі розглядаються питання моделювання й діагностування основних дефектів МОП-структур.*

### *Введение*

Известно, что модель одиночных константных неисправностей является неадекватным приближением дефектов современных МОП-структур и физические дефекты КМОП-схем отображаются тремя классами неисправностей: константные (*SA*), обрывы (*SOP*), короткие замыкания (*SON*). Сведения о распределении дефектов и неисправностей в МОП-структурах приведены в [1]. Доминирующим видом дефектов при производстве ИС являются замыкания (short) между узлами схемы [1–2]. Эти замыкания могут быть двух типов: замыкания внутри вентиля и замыкания между внешними полюсами различных вентилях [3–4]. Согласно данным [5–6] внутренние замыкания (bridging fault-BF) представляют около 90% всех замыканий.

Доля физических дефектов, которые представляются неисправностями вида «обрыв транзистора» оценивается примерно 15–20% всего объема производимых МОП-структур [1–2], что и предопределяет актуальность диагностирования этих неисправностей [1,3–5].

Целью работы является классификация моделей неисправностей вида «замыкания» и обрывов и соответствующих им методов диагностирования. В частности рассматривается авторский подход при моделировании этих неисправностей.

### *Модели замыканий*

Точность моделирования замыканий и, следовательно, автоматическое построение теста сильно зависят от точности и адекватности моделей замыканий. Основная трудность построения модели замыканий заключается в необходимости рассмотрения логической интерпретации поведения не только вентиля с неисправностью, но и вентилях с которыми он связан, в частности учитывать различие их порогов срабатывания. Более того,

необходимо учитывать пороги срабатывания любого внешнего полюса вентиля с неисправностью. Известно, что понижение питающего напряжения обнаруживает замыкания с высоким сопротивлением, которые не обнаруживаются при более высоком питающем напряжении. Отсюда следует необходимость логического тестирования устройств при различных значениях напряжения. Для точной оценки поведения замыкания, мы должны определить напряжение на узлах, соединенных с местоположением неисправности, т.е. зафиксировать каждый входной вектор воздействий. Используя эту информацию, мы можем определить, базируясь на знании логических порогов вентиля-последователей, когда неисправность обнаруживается на выходах этих вентилях.

Для прогноза поведение неисправности с высокой степенью доверия, мы должны изучать его специфические черты, определить диапазон флуктуаций пороговых значений, так как обычные вариации параметров производства БИС являются причиной широкого разброса их логических порогов срабатывания. В [6] указано, что вариации коэффициента усиления транзисторов превышают 20% для одной партии.

Добавим, что известное «византийское» поведение неисправности определяется тем, что различные значения в определенном интервале могут быть интерпретированы как различные логические значения различных вентилях, принадлежа вариациям пороговых напряжений между различными вентилями типами [7–8]. На вентилях уровне можно различать одиночные и кратные замыкания, которые соединяют две или более чем две линии соответственно. В [9] показано, что большинство кратных неисправностей обнаруживаются тестами, построенными для одиночных замыканий, входящих в кратные. Поэтому большинство работ в данной области ориентировано на одиночные замыкания. Если замыкание случается между двумя линиями, которые соединены другим путем, то мы имеем замыкание с обратной связью (ОС). Ясно, что замыкания с ОС трансформируют комбинационные схемы в последовательностные. Если число инверсий сигнала между двумя замкнутыми линиями нечетно, тогда такие замыкания могут вызвать осцилляции, которые трудны для тестирования [10].

Простейшая модель замыкания — это монтажные модели, которые адекватно отражают ее для технологий (РТЛ, ДТЛ, ЭСЛ) [11–12]. Используются две модели: монтажное-И и монтажное-ИЛИ в зависимости от вида логики. Они представлены на рис.1б. Функция

$Z(x,y)$  определяется выражением И(x,y) или ИЛИ(x,y) соответственно и в любом случае  $Z(x,x)=x$ .

Однако эти модели являются упрощенными и неадекватными при их применении в КМОП-структурах. Более точной является модель на рис.1с где значение  $Z(x,y)$  при  $x \neq y$  определяется отношением силы переходов  $1 \rightarrow 0$ ,  $0 \rightarrow 1$  замкнутых линий. Сопротивление этих переходов зависит от технологии изготовления, геометрических характеристик транзисторов (ширина и длина канала) и количества транзисторов и поэтому значение  $Z(x,y)$  может быть неопределенным. Необходимо отметить, что и эта достаточно точная модель является упрощенной, так как нелинейные транзисторы аппроксимируются линейными резисторами и сопротивление замыкания предполагается незначительным по сравнению с сопротивлением транзисторов [13].

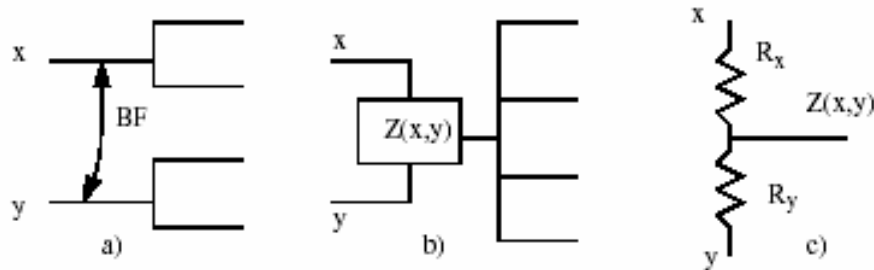


Рисунок 1 — Монтажные модели замыканий

В работах [14–16] автором был предложен метод моделирования исправных и неисправных МОП-схем, позволяющем генерировать тесты для МОП-структур на переключательном уровне. Для его применения необходимо структурное преобразование исправной схемы согласно рис.2. При флаге моделирования  $F=1$  моделируем исправную схему, при  $F=0$  имеем на линиях L1(L2) значения сигналов  $S1=(H1,G1)$  ( $S2=(H2,G2)$ ) соответственно. Здесь H,G значения силы тока и напряжения в выбранном алфавите моделирования.

### **Модели «обрывов»**

Неисправность «обрыв транзистора» (*stuck-open*) означает, что неисправный транзистор никогда не переключается в активное состояние (постоянно отключен), либо неисправный транзистор никогда не переключается в неактивное состояние (постоянно открыт-*stuck-on*) [17]. Для *n*-МОП транзистора *stuck-open* и *stuck-on* неисправности эквивалентны *stuck-at=0* и *stuck-at=1* на затворе транзистора соответственно. Обнаружение *stuck-open*-неисправности

обычно требует двухнаборного теста. Если есть *stuck-open*-неисправность, то выход не может установиться в лог.1 (*stuck-open*-неисправность для *p*-МОП транзистора) или он не может быть установлен в лог.0 (*stuck-open*-неисправность для *n*-МОП-транзистора). Тест для *stuck-open*-неисправности *p*-МОП-транзистора должен своим первым набором установить выход транзистора в лог.0 и следующим вторым тестовым воздействием попытаться установить выход МОП-транзистора в лог.1. Если *p*-МОП-транзистор неисправен, то выход остается в лог.0, в противном случае выход устанавливается в лог.1.

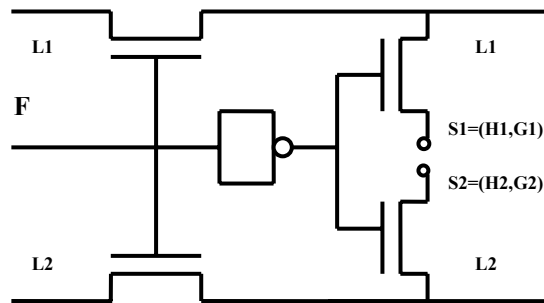


Рисунок 2 — Структурное преобразование схем для моделирования замыканий

К примеру для вентиля И-НЕ на рис.3. для тестирования *stuck-open*-неисправности транзистора *P1* мы используем 2-наборный тест (*11, 01*). Выход исправного вентиля устанавливается в лог.1 после приложения второго набора. В случае *stuck-open*-неисправности транзистора *P1* он остается в лог.0. Для обнаружения *stuck-open*-неисправности транзистора *N1* мы применяем двухнаборный тест (*00, 11*).

Тестирование *stuck-on*-неисправности требует большего знания о транзисторе и электрических характеристиках соединения. Если *p*-МОП-транзистор имеет неисправность *stuck-on*, мы можем потенциально обнаружить эту неисправность переключением соответствующего *n*-МОП-транзистора в активное состояние. Для обнаружения *stuck-on*-неисправности транзистора *P1* на рис.1 можем использовать набор *AB=11*. Если переход положительного фронта *0→1* *p*-МОП-транзистора сильнее, чем переход фронта *1→0* *p*-МОП-транзистора, какими являются транзисторы *N1, N2*, то выход будет иметь значение лог.1 и неисправность будет обнаружена. Здесь мы имеем случай *p*-доминирования. Если схема является *n*-

доминирующей, т.е. переход  $1 \rightarrow 0$   $n$ -МОП-транзистора сильнее, чем переход  $1 \rightarrow 0$   $p$ -МОП-транзистора, тогда *stuck-on*-неисправность не может быть обнаружена измерением напряжения на выходе. *Stuck-on*-неисправность может быть обнаружена измерением силы тока ( $I_{DDQ}$ ), связанного с построением статического токового пути между узлами «земля» и «питание».

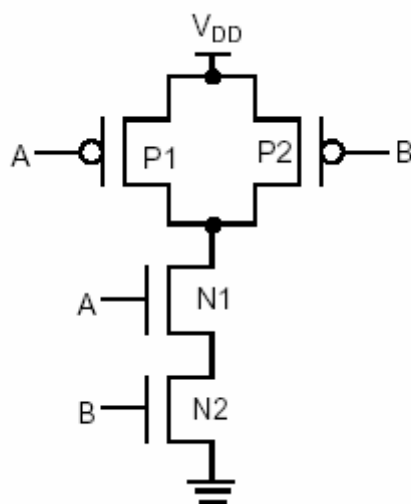


Рисунок 3 — Базовый вентиль И-НЕ на переключательном уровне

Обрыв сети — это обрыв в  $p$ -сети или в  $n$ -сети ячейки, который прерывает один или более транзисторных путей между выходами устройства и узлами «питание» или «земля» [18]. Под транзисторным путем понимаем последовательность транзисторов, физически связанными своими полюсами, т.е. стоками и истоками. В [19–20] показано, что тест для открытого транзистора может обнаруживать некоторые из 4 типов обрывов, которые возникают вследствие одиночного плавающего затвора транзистора.

Обрыв схемы (непреднамеренный разрыв в ней) разделяет узел на два или более узла. С точки зрения прохождения тока различают сильный (ток не может течь между концами обрыва при приложении напряжения к ним) и слабый (малый ток утечки вследствие туннельного эффекта течет через обрыв) разрывы [21]. Обнаружение обрыва сети на основе измерения напряжения требует двухнаборного теста, однако показано, что транзитный путь к узлам „питание” или „земля” может делать двухнаборные тесты недействительными при тестировании неисправности “*open*”. Анализ возможных ситуаций выполнен в [22–23].

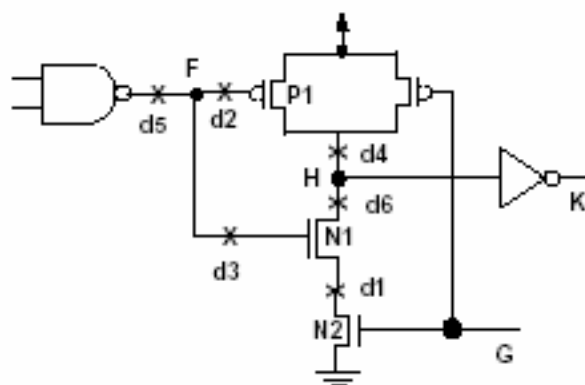


Рисунок 4 — Виды обрывов МОП-ячейки

Так для базовой ячейки на рис.4 [21] дефекты  $d_2$ ,  $d_3$  являются обрывами затворов для транзисторов  $P1$ ,  $N1$  соответственно. Дефекты  $d_4$ ,  $d_6$  классифицируем как обрыв  $p$ -сети ( $n$ -сети) соответственно. Дефект  $d_5$  —неисправность обрыв выхода, а дефект  $d_1$  есть неисправность обрыва истока (стока) для транзисторов  $N1(N2)$  соответственно.

Для моделирования обрывов используют переключательные модели, где МОП-транзистор моделируется переключателем, управляемым напряжением. Существуют переключательные модели схем, построенных на базе эмиттерно-связанной логики (ECL), в которых транзисторы рассматриваются, как переключатели управляемые током [23]. В [24–25] рассматриваются вопросы построения переключательных моделей для биполярных транзисторов. Обычно биполярный транзистор рассматривается как управляемый током переключатель и расширенная модель, в которой он рассматривается как два переключателя, управляемые напряжением, необходима для совместимости с переключательной моделью МОП-транзистора. Эта аппроксимация, адекватная для BiCMOS вентилях, анализируется в [23–27] и позволяет расширить возможности метода в [16], где приведен пример моделирования неисправности “обрыв транзистора” на переключательном уровне. Для диагностирования этих неисправностей необходимо: 1) определить алфавит моделирования; 2) построить соответствующую алфавиту систему булевых уравнений для описания соотношений между значениями сигналов в устойчивом состоянии МОП-схемы[15]; 3) выполнить структурное преобразование МОП-схемы согласно приемам, изложенным в [16].

### **Выводы**

На основании выполненного обзора делаем вывод, что разнообразие характеристик различных типов современных МОП-структур стимулирует развитие рассмотренных методов моделирования и диагностирования.

### **Литература**

1. Вейцман И.Н., Кондратьева О.М. Тестирование КМОП-схем // Автоматика и телемеханика. — 1991. — N 2. — с.3–34.
2. R. Rodriguez-Montanes, E. M. J. G. Bruls, and J. Figueras, “Bridging Defect Resistance Measurements in a CMOS Process”, Int. Test Conf., 1992, pp. 892–899.
3. M. Renovell, P. Huc, and Y. Bertrand, “CMOS Bridging Fault Modeling”, VLSI Test Symp., 1994, pp. 392–397.
4. M. Renovell, P. Huc, and Y. Bertrand, “The Concept of Resistance Interval: A New Parametric Model for Realistic Resistive Bridging Fault”, VLSI Test Symp., 1995, pp. 184–189.
5. J. J. T. Sousa, F. M. Goncalves, and J. P. Teixeira, “IC Defects-Based Testability Analysis,” Int. Test Conf., 1991, pp. 500–509.
6. V. Sar-Dessai and D. M. H. Walker, “Accurate Fault Modeling and Fault Simulation of Resistive Bridges,” Int. Symp. Defect and Fault Tolerance in VLSI Systems, 1998, pp. 102–107.
6. V. S. Rangavajjhata, B. Bhara and S. E. Kerns, “Statistical degradation analysis of digital CMOS IC’s”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, June 1993, Vol. 12, pp. 837–843.
7. B. Chess, C. Roth and T. Larrabee, “On evaluating competing bridge fault models for CMOS ICs”, Proc. 12th IEEE VLSI Test Symposium, 1994, pp. 446–451.
8. P. C. Maxwell and R. C. Aitken, “Biased voting: A method for simulating CMOS bridging faults in the presence of variable gate logic thresholds”, in Proc. IEEE International Test Conference, Oct. 1993, pp. 63–72.
9. C.Y. Mei, “Bridging and Stuck-At Faults,” IEEE Trans. on Computers, Vol. C–23, No.7, pp. 720–727 July, 1974.
10. Kristian Wiklund, Tomas Magnusson and Peter Dahlgren. Switch-Level Test Generation of Competing Bridging Faults in the Presence of Feedback, Gothenburg, Sweden, Chalmers University of Technology, TR 00–02, 2002,
11. M. Abramovici and P.R. Menon. “A practical approach to Fault Simulation and Test Generation for Bridging Faults”//IEEE Trans. on Computers, Vol. C–34, No.7, pp. 658–663, July, 1985.
12. Y. Gong, S. Chakravarty, “A Diagnosis Algorithm for Bridging Faults in Combinational Circuits,” Technical Report, Department of Computer Science, University of Buffalo, State University of New York, 1992.
13. V. Sar-Dessai and D. M. H. Walker, “Accurate Fault Modeling and Fault Simulation of Resistive Bridges,” Int. Symp. Defect and Fault Tolerance in VLSI Systems, 1998, pp. 102–107.
14. Андрюхин А.И. Алгоритмы параллельного логического моделирования и псевдослучайной генерации тестов для МОП-структур // Микроэлектроника. —1995, N 5. —с. 331–336.

15. Андрюхин А.И. Параллельное многозначное логическое моделирование исправных и неисправных псевдобулевых схем. // Электронное моделирование — 1997, N 1. — с. 58–63.
16. Андрюхин А.И. Параллельное моделирование неисправностей МОП-структур// Научные труды Донецкого государственного технического университета. Серия: Проблемы моделирования и автоматизации проектирования динамических систем. Выпуск 29. 2001 г., С.205–211.
17. R.L.Wadsack. Fault modeling and logic simulation of CMOS and MOS integrated circuits//Bell Syst. Tech.J., vol.57,May–June 1978, , pp.1449–1473.
18. H.Konuk .Testing for Opens in Digital CMOS Circuits, University of California, Santa Cruz, dissertation 1996., p.109.
19. M.Renovell, G.Cambon. Electrical analysis and modeling of floating –gate fault// IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Nov. 1992 , pp. 1450–1458.
20. V.H.Champac, A.Rubio, J.Figuera. Electrical model of the floating gate defect in CMOS ICs:Implications on IDDQ testing// Transactions on Computer-Aided Design of Integrated Circuits and Systems, March 1994, pp. 359 –369.
21. S.Chakravarty. Defect Detection Capability of Delay Tests for Path Delay Faults //TR 96–20, State University of New York, Buffalo, 1996, 23 p.
22. C.Di, J.A.G.Jess. On accurate modeling and efficient simulation of CMOS opens // Proceedings of Int. Test Conf.Oct.1993, p.875–882.
23. M.Favalli, M.Dalpasso, P.Olivo, B.Ricco. Modeling of broken connections faults in CMOS ICs. Proceedings of European Design and Test Conf., 1994.
23. Yang, A.T.;Chang, Y.-H.;Saab, D.G.;Hajj, I.N.Switch-level timing simulation of bipolar ECL circuits // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol.: 12, № 4, April 1993, pp.516 –530.
24. Ma S.C., McCluskey E.J. Open faults in BiCMOS gates //Там же, Vol.: 14, № 5, May 1995, pp. 567 –575.
25. J. C-Mo Li.Test and Diagnosis of Open Defects in Digital CMOS Integrated Circuits/Ph.D.thesis, Stanford University,Stanford,California,2002.
26. S. Ma “Testing BiCMOS and dynamic CMOS logic”, CRC Technical Report No. 95–1, Stanford University, Stanford, 1995,p.55.
27. Li, J. C.M and E.J. McCluskey, "Testing for Tunneling Opens," Proceeding of International Test Conference, pp. 85–94, 2000.



## **О ПАРАДОКСЕ НАРУШЕНИЯ СТАЦИОНАРНОГО РЕЖИМА ПРИ МОДЕЛИРОВАНИИ WEB-КЛАСТЕРА С ДИСЦИПЛИНОЙ ОБСЛУЖИВАНИЯ SITA-E**

И.И.Труб

ООО «Сургутгазпром», г.Сургут Тюменской области, Россия

### ***Анотація***

*Розглянуто явище порушення стаціонарного режиму при імітаційному моделюванні дисципліни обслуговування SITA-E для WEB-кластеру при гарантованому виконанні умови його існування для дисципліни FIFO. Установлено зв'язок цього явища з масштабуванням модельного часу і вироблені рекомендації з подолання виявленого парадокса.*

За последние годы веб-кластеры стали объектом пристального изучения специалистов самой различной направленности — IT-менеджеров, инженеров-сетевиков, ученых-математиков. Об этом свидетельствуют нарастающий из года в год поток публикаций, в том числе монографий [2], [4].

Организация распределенного веб-сервера весьма сложна и требует детальной проработки множества вопросов, например:

- изучение и анализ сетевого трафика и ожидаемой загрузки сервера, распределения длительности обслуживания запросов;
- выбор архитектуры распределенной системы;
- маршрутизация запросов внутри системы;
- функциональность распределителя нагрузки;
- дисциплина обслуживания запросов и выбора сервера;
- распределение хранимой информации.

Подробные обзоры текущего состояния вопросов, связанных с изучением веб-кластеров, можно найти в [1] и [6].

### ***1. Поток данных в Веб***

На протяжении нескольких десятилетий при анализе моделей, возникающих в теории вычислительных систем, исследователи привыкли предполагать входные потоки пуассоновскими, а распределения длин заявок — экспоненциальными. Эти предположения позволяют легко строить марковский процесс и получать наглядные аналитические результаты, которые носят если и не предсказательный, то по крайней мере объясняющий характер. Можно ли предположить то же самое относительно Всемирной паутины?

Основополагающей здесь явилась работа [5], с тех пор регулярно цитируемая. Обработав большой статистический материал, авторы убедительно показали, что потоки в Веб — не пуассоновские, а описываются другими законами распределения — с «тяжелым хвостом» (heavy-tailed), представителями которых являются степенные распределения. Эти распределения подчиняются зависимости  $\Pr(X > x) \sim x^{-\alpha}$ , где  $0 < \alpha < 2$ ,  $\Pr(X > x)$  — вероятность того, что значение случайной величины  $X$  превысит заданное число  $x$ .

Типичным примером распределения с «тяжелым хвостом» является распределение Парето

$$\Pr(X \leq x) = \begin{cases} 0, & x < s \\ 1 - s^\alpha x^{-\alpha}, & x \geq s, \alpha < 2 \end{cases}$$

Основными свойствами этого распределения являются:

- с увеличением продолжительности накопленного интервала времени ожидаемое остаточное время увеличивается;
- бесконечное значение дисперсии, а при  $\alpha < 1$  — бесконечное математическое ожидание;
- большая доля нагрузки приходится на очень малую (менее 1%) долю «длинных» интервалов времени.

## 2. Дисциплина маршрутизации запросов SITA-E

Улучшить показатели производительности по сравнению с использованием дисциплины FIFO можно только в том случае, если при маршрутизации принять во внимание трудоемкость и ресурсоемкость поступившего запроса.

Функциональность современных распределителей нагрузки позволяет это. Уже стала общепринятой их классификация на устройства уровня 4 (layer-4) и уровня 7 (layer-7). Это транспортный и прикладной уровни согласно модели сетевых функций OSI. Устройства layer-4 анализируют пакеты только на уровне TCP/IP, не доходя до уровня приложения. Они не принимают во внимание содержимое HTTP-запроса при перенаправлении пакета на сервер. Такой механизм называется *слепым к содержанию* (content-blind), или *немедленным связыванием* (immediate binding). Устройства же layer-7 принимают решение о маршрутизации запроса только при получении HTTP-пакета. При этом они могут, проведя анализ URL, принять во внимание тип запроса, оценить его трудоемкость и определить, на каком сервере находится запрашиваемый ресурс. Этот механизм называется *content-aware*, или *отложенное*

*связывание* (delayed binding). Применение устройств layer-7 оправдано, если его возможности используются дисциплиной обслуживания. Каким же образом можно это сделать?

Важным шагом в этом направлении явилась работа [3]. В ней рассмотрена новая дисциплина SITA-E (Size Interval Task Assignment - equal). Основная идея — путем учета длины заявки при маршрутизации снизить коэффициент вариации длин заявок на каждом из серверов. Введено понятие *точек разбиения* (cutoff points): если требуется разделить входной поток заявок между  $N$  серверами,  $N - 1$  точек разбиения длин заявок на интервалы находятся из условия

$$\int_0^{x_1} xf(x) dx = \int_{x_1}^{x_2} xf(x) dx = \dots = \int_{x_{N-1}}^x xf(x) dx,$$

где  $f(x)$  — плотность распределения длин заявок. Заявки, длина которых лежит в пределах интервала  $[x_{i-1}; x_i]$ , направляются на  $i$ -й сервер. Интегральное соотношение означает равенство интервалов, взвешенных по вероятности.

Для практической реализации или имитационного моделирования дисциплины SITA-E необходимо предварительно вычислить точки разбиения  $x_i$ . В зависимости от вида  $f(x)$  для одних функций можно получить аналитическое решение, для других — нет. Приведем несколько примеров:

1. Равномерное распределение на интервале  $[a; b]$ :

$$x_i = \sqrt{a^2 + i((b^2 - a^2)/N)}, i = 0, 1, \dots, N.$$

2. Распределение Парето:  $x_i = s \left(1 - \frac{i}{N}\right)^{\frac{1}{1-\alpha}}$ ,  $1 < \alpha < 2$ .

Для экспоненциального распределения аналитического решения не существует.

Может ли теоретически SITA-E привести к нарушению стационарного режима в системе? Ответ на этот вопрос дает следующая

**Теорема.** *Если случайный процесс изменения длины очереди в системе массового обслуживания GI/G/N-FIFO описывается стационарным режимом, то в случае реализации дисциплины SITA-E длина очереди к каждому из серверов также описывается стационарным режимом.*

Доказательство теоремы здесь не может быть приведено ввиду ограниченности объема публикации.

### 3. Парадокс SITA-E

Программную реализацию SITA-E следует, тем не менее, проводить с достаточной осторожностью, так как простое следование описанию может привести к парадоксальным результатам. Рассмотрим простой пример. Пусть интенсивность входного потока равна  $\lambda = 0,21$ , а средняя длина заявки равномерно распределена на интервале  $[20; 40]$ , то есть  $\mu = 1/30 \approx 0,033$ . Разумеется, один сервер не может в этом случае обеспечить стационарный режим, так как  $\mu < \lambda$ . Минимально необходимое количество серверов для этого определяется из условия  $n > \lambda/\mu$  и равно 7, но для уверенности пусть их будет 8. Предположим также, что модельное время не масштабируется, то есть при потактовом продвижении модельного времени длина заявки генерируется как целое число от 20 до 40 с равной вероятностью  $1/21$  каждое. Если моделировать такую многоканальную систему при условии общей очереди, стационарный режим будет существовать. Если разделить очереди, используя дисциплину FIFO-Random, заявки в статистически равных количествах «разойдутся» по восьми очередям, опять-таки обеспечив существование стационарного режима. А вот попытка применить SITA-E приводит к «астрономическим» значениям для счетчиков общего числа заявок и времени пребывания их в системе и, как следствие, к аварийному завершению программы по причине переполнения. В чем же дело?

Вычислим с точностью до двух знаков значения точек разбиения. Они равны: 20; 23,45; 26,46; 29,15; 31,62; 33,91; 36,06; 38,08; 40. Таким образом, на серверы попадают заявки следующей длины, создавая тем самым нагрузку:

1. Сервер 1 — 20, 21, 22, 23.  $\lambda_1 = 0,21 \cdot (4/21) = 0,04$ ,  $\mu_1 = 1/21,5 \approx 0,0465 > \lambda_1$ .
  2. Сервер 2 — 24, 25, 26.  $\lambda_2 = 0,21 \cdot (3/21) = 0,03$ ,  $\mu_2 = 1/25 \approx 0,04 > \lambda_2$ .
  3. Сервер 3 — 27, 28, 29.  $\lambda_3 = 0,21 \cdot (3/21) = 0,03$ ,  $\mu_3 = 1/28 \approx 0,036 > \lambda_3$ .
  4. Сервер 4 — 30, 31.  $\lambda_4 = 0,21 \cdot (2/21) = 0,02$ ,  $\mu_4 = 1/30,5 \approx 0,033 > \lambda_4$ .
  5. Сервер 5 — 32, 33.  $\lambda_5 = 0,21 \cdot (2/21) = 0,02$ ,  $\mu_5 = 1/32,5 \approx 0,031 > \lambda_5$ .
  - 6. Сервер 6 — 34, 35, 36.  $\lambda_6 = 0,21 \cdot (3/21) = 0,03$ ,  $\mu_6 = 1/35 \approx 0,0286 < \lambda_6$ .**
  7. Сервер 7 — 37, 38.  $\lambda_7 = 0,21 \cdot (2/21) = 0,02$ ,  $\mu_7 = 1/37,5 \approx 0,027 > \lambda_7$ .
  8. Сервер 8 — 39, 40.  $\lambda_8 = 0,21 \cdot (2/21) = 0,02$ ,  $\mu_8 = 1/39,5 \approx 0,025 > \lambda_8$ .
- Итак, на шестом сервере стационарный режим не обеспечивается.

Объяснение парадокса заключается в следующем. Дело в том, что значение  $\lambda_6$  при моделировании отличается от своего истинного

значения, равного 0.0226. Искажение возникло из-за того, что единица модельного времени была принята равной единице реального времени, иными словами, коэффициент масштабирования времени  $T=1$ . Такое значение в данном случае является слишком грубым. Ситуацию легко исправить, если принять, например,  $T=2$ . В этом случае  $\lambda_6 = (0,21/2) \cdot (5/41) = 0,0128$ ,  $\mu_6 = 1/70 \approx 0.0143 > \lambda_6$ .

В общем случае при равномерном распределении длины заявки коэффициент масштабирования времени  $T$  должен удовлетворять следующему условию:

$$\frac{\lambda}{T} \frac{Z(Tx_{k-1}; Tx_k)}{Z(Ta; Tb)} < \frac{2}{\lceil Tx_{k-1} \rceil + \lfloor Tx_k \rfloor}, \quad k=1, \dots, N,$$

где  $Z(x; y)$  — количество целых чисел, принадлежащих отрезку  $[x; y]$ , а и  $b$  — минимальное и максимальное значения длины заявки. В случае иных распределений условие имеет более сложный характер, однако вычислять  $T$  аналитически нет необходимости, т.к. коэффициент масштабирования может быть подобран и экспериментальным путем.

**Вывод:** при моделировании дисциплины SITA-E с помощью потактового продвижения модельного времени недостаточно большое значение коэффициента масштабирования может привести к потере стационарного режима на одном или нескольких серверах и, как следствие, к серьезному искажению результатов моделирования. Это следует учесть при планировании имитационного эксперимента.

### *Литература*

1. Труб И. Алгоритмическое обеспечение распределенных WEB-серверов // Открытые системы. — 2003. — № 5. — С.49–54.
2. Bourke T. Server Load Balancing. — O'Reilly, 2001.
3. Harchol-Balter M., Crovella M., Murta C. To queue or not to queue?: When FCFS is better than PS in a distributed system // Technical Report, BUCS-TR-1997-017, October, 31, 1997.
4. Koppurapu C. Load Balancing Servers, Firewalls and Caches. — New York: John Wiley&Sons, 2002.
5. Paxson V., Floyd S. Wide-area Traffic: The Failure of Poisson Modeling // IEEE/ACM Transactions on Networking. — 1995. — June. — P. 226–244.
6. The State of the Art in Locally Distributed Web-server Systems / V. Cardellini, E. Casalicchio, M. Colajanni, P. S. Yu // ACM Computing Surveys, vol.34, No.2, June, 2002. — P. 263–311.

## ИНТЕЛЛЕКТУАЛИЗАЦИЯ ПРОЦЕССА ПРОЕКТИРОВАНИЯ АППАРАТУРЫ СРЕДСТВАМИ ЯЗЫКА VHDL

А.В. Григорьев, Д.А. Кошелева  
ДонНТУ

### **Анотація**

*Розглядається задача інтелектуалізації САПР проектування апаратури на базі мови VHDL засобами спеціалізованої інструментальної оболонки в умовах представлення знань на основі графік, що породжують, з визначеними над ними продукціями.*

### **Введение**

Язык проектирования аппаратуры HDL является основным инструментальным средством многочисленных САПР — Active HDL, OrCAD и т.д. [1–4]. HDL наряду с Verilog является одним из двух наиболее популярных средств проектирования аппаратуры в мире и — самым популярным в Европе.

Процесс проектирования аппаратуры с использованием языка HDL состоит в разработке программы, реализующей необходимый алгоритм функционирования проектируемого устройства. Имеющиеся компиляторы обеспечивают перевод алгоритмов на HDL в схемы, соответствующие выбранной аппаратной базе реализации, например — в ПЛИС. Представление проектов в форме текста в значительной мере свободно от недостатков графических и табличных форм представления проектов, таких как трудоёмкость процедуры ввода, трудность поиска ошибок. Вместе с тем, особо значимыми в этом случае становятся процедуры отладки и верификации проектных решений. К достоинствам описания проекта в текстовом виде (на языках Verilog, VHDL и др.) можно отнести [2]:

- компактность представления очень сложных логических проектов;
- легкость понимания и большая скорость создания проектов, включая относительную простоту поиска ошибок и внесения модификации в проект;
- лёгкость переноса проекта в различные приборные среды;
- простота настройки или перенастройки параметров разрабатываемых устройств или их фрагментов, например, разрядности или подмножества реализуемых функций;
- сближение методов проектирования аппаратуры и программных средств;

- улучшение взаимопонимания между проектировщиками различных подсистем вычислительных комплексов;
- возможность перевода программ пользователя, построенных на базе стандартных универсальных языков (например, С), в языки описания аппаратуры и обратно;
- наличие стандартных комментариев в текстах программ.

Главное достоинство языка HDL — это возможность накопления и повторного использования экспертных знаний об апробированных решениях в форме, удобной для хранения, передачи и редактирования. Однако, в настоящее время нет средств обучения САПР, позволяющих обеспечить процесс передачи экспертных знаний о методиках проектирования либо напрямую — от эксперта, либо на основе ряда примеров, т.е. накопленных апробированных решений. Обеспечение такой возможности позволит повысить эффективность использования САПР на базе языка HDL за счет повышения уровня автоматизации процесса проектирования.

Цель данной работы — описание средств обучения САПР, позволяющих обеспечить процесс передачи экспертных знаний о методиках проектирования путем создания на основе обучения интеллектуальной надстройки в среде специализированной оболочки для создания интеллектуальных САПР.

### ***1. Анализ существующих методов автоматизации проектирования в САПР на базе HDL***

В настоящее время в САПР-ах используется версия языка HDL, ориентированная на возможность графического отображения структурной компоненты программ — язык VHDL. Такой язык позволяет совместить достоинства графических моделей устройств и текстовых описаний алгоритмов. Среди имеющихся средств автоматизации создания моделей на языке VHDL в различных САПР можно назвать [2]:

- использование шаблонов, макросов, мастеров для синтеза текстов программ;
- формирование языковых описаний по графической модели алгоритма, представленной в виде блок-схемы;
- использование наборов подключаемых библиотек программ.

*Макросы* служат средством автоматизации интерфейса самой системы проектирования и интеграции её в проект создаваемых HDL программ. Они включают: А) описания, заключаемые в системные ключевые слова, исходного набора файлов проекта; Б) входные конкретные данные для программы; В) время симуляции в системе. Во время запуска макроса происходит передача параметров в систему

проектирования. У разработчика появляются предпосылки расширить возможности языка за счёт введения собственных макроопределений, ориентированных на класс реализуемых проектов.

Набор подключаемых библиотек констант и функций, применяемых в текстах программ на VHDL, может быть как стандартным, так и определённым пользователем. В нём есть стандартизированные константы и готовые функции, применяемые в текстах программ на VHDL.

*Мастера* помогают быстро подготовить шаблон для заданной архитектуры программируемой системы и даже определить количество начальных параметров и типов. Но все остальные дальнейшие заботы по написанию кода программ ложатся, естественно, на программиста-разработчика.

Собственно текстовые модельные описания используется в современных САПР очень широко. Можно назвать, для примера, некоторые новые возможности, связанные с использованием текстов в САПР Active HDL 5.1 [5].

Данные возможности можно отнести к достоинствам САПР:

- 1) В просмотрщике структуры проекта:
  - модификация (создание) текстового файла конфигурации VHDL;
- 2) В графических процессах:
  - текстовое описание процессов;
- 3) В редакторе блок-схем:
  - механизм синтеза кода на VHDL по построенной блок-схеме для любого элемента блок-диаграммы;
  - предусмотрены комментарии ко всем элементам блок-схемы (отдельная вкладка), вводятся пользователем; комментарии автоматически входят в генерируемый код VHDL;
  - генерация по блок-диаграмме архитектуры и сохранение ее в проекте и т.д.

Вместе с тем, к типичным недостаткам САПР Active HDL 5.1 можно отнести:

- отсутствие явного языка формирования технического задания;
- имеющиеся комментарии могут рассматриваться лишь как фрагменты ТЗ.

## **2. Постановка задачи**

Следует отметить, что из названных средств мастера являются самым развитым методом синтеза решений в САПР на базе VHDL. Однако, разработка каждого нового мастера, при условии его полноценного построения, есть задача разработки маленького специализированного САПР, ориентированного на некоторый класс



решений (устройств). Разработка каждого нового мастера предполагает творческий процесс программирования, включая:

- разработку диалоговых шагов, включая определение параметров проектирования и их возможных значений;
- определение фрагментов текстов, соответствующих тому или иному набору значений параметров требуемого решения, определяемому в диалоге с пользователем;
- разработку алгоритма компоновки определенных фрагментов текстов по заданному набору значений параметров.

Перед разработчиком встает задача разработки языка ТЗ, определение состава возможных решений и построение алгоритма выбора нужного решения по заданному набору значений параметров. Т.о., разработка такого мастера есть форма передачи разработчиком своего опыта по проектированию некоторого класса устройств. Недостатки такого пути:

- сложность процесса передачи знаний;
- необходимость отладки и верификации мастеров.

Т.о., такой процесс передачи знаний имеет высокую трудоемкость. С целью повышения эффективности процессов создания мастеров можно применить технологию создания мастера как экспертной системы, способной обучаться, т.е. перенимать опыт проектировщика на ряде примеров или приобретать знания напрямую от эксперта.

### ***3. Предлагаемый путь создания интеллектуальной надстройки***

#### ***3.1. Общая модель интеллектуальной надстройки***

Для решения задачи предлагается использовать путь создания интеллектуальной надстройки над VHDL, описанный в [6–9]. Основанием такого выбора является возможность учесть как достоинства, так и недостатки САПР на базе VHDL.

По классификации, приведенной в [7], любой САПР на базе VHDL, как конкретный проблемно-ориентированный САПР имеет следующие характерные черты:

- САПР может обеспечить для обучения базы знаний совокупность апробированных решений, представленных в виде текстов,
- САПР не имеет явно выраженного языка задания ТЗ.

Соответственно, возможны следующие пути построения интеллектуальной надстройки, являющиеся вариантом модели САПР решения типичных задач проектирования:

*1) Создание целевого пространства систем (ЦПС), т.е. пространства возможных решений-прототипов.*

Имеется два пути:

А) САПР может быть источником решений-прототипов; далее ЦПС в нужной форме представления создается инструментальной оболочкой автоматически;

В) Эксперт в данной предметной области может самостоятельно ввести ЦПС в нужной форме представления;

II) Создание пространства обликос систем (ПОС), т.е. пространства технических заданий.

Имеется три пути:

С) ПОС как совокупность слов-ТЗ может быть сформирован оболочкой автоматически на основе имеющегося ЦПС, заданного в требуемой форме представления.

Д) Источником множества слов-ТЗ для создания ПОС может быть только пользователь — эксперт в предметной области, в силу отсутствия в САПР языка описания ТЗ. В этом случае ПОС в нужной форме представления формируется автоматически.

Е) Эксперт в данной предметной области может самостоятельно ввести ПОС в требуемой форме представления.

Главная цель функционирования *К0*, составляющая главный признак множества решений в ПОС, соответствует классу проектируемых устройств и задается в этом случае неявно.

Построенная таким образом экспертная система может выступать в качестве надстройки над САПР VHDL. При этом САПР может выступать в двух качествах:

1) Как источник семантически верных решений, предназначенных для обучения базы знаний;

2) Как приемник синтезированных в базе знаний решений для последующей их интерпретации и документирования.

### ***3.2. Модули знаний как отражение структурно-функциональной модели аппаратуры в VHDL***

В соответствии с [8] для организации базы знаний используется модульный принцип. Модуль знаний в инструментальной оболочке для построения интеллектуальных САПР формируются в форме порождающей контекстно-свободной грамматики, эквивалентной И-ИЛИ-дереву [5], с определенным над ней множеством продукций, явных или неявных. Грамматика описывает множество синтаксически верных (возможных) решений, а продукции, как явные так и неявные, определяют множество правил вывода, описывающих вычленение семантически верного подмножества решений, соответствующего некоторому набору требований, т.е. продукции описывают логику проектирования, исходя из требований к решению.

Специфика формирования модулей знаний в данном случае определяется спецификой языка VHDL. Модель устройства на языке VHDL имеет две явно определенные стороны описания: структурную и функциональную.

Структурная часть — это:

1. Совокупность блоков — процессов, включая их внешние полюса (входы-выходы),
2. Список связей между блоками.

Функциональная часть — это:

1. Тексты описания процессов, составляющих базовые блоки структуры;
2. Задание значений переменных — полюсов, а так же описание моделей сигналов.

Рассмотрим возможные способы формирования модулей знаний, отличающиеся уровнем знания о грамматике языка VHDL в среде инструментальной оболочки.

Минимальный уровень знаний о языке — это способность оболочки выделить значимую часть текста и — незначимую. К незначимой части можно отнести комментарии и лишние пробелы.

Глубина знаний о процедурном делении языка VHDL предполагает, что отдельный модуль знаний может задавать множество решений, относящихся к таким классам: 1) проекты одного типа в целом; 2) архитектуры одного типа; 3) процессы одного типа.

В каждом случае пара ЦПС — ПОС строится для различного числа модулей знаний. Наиболее сложный случай предполагает построение модуля знаний для отдельного типа блоков из блок-схемы, рассматриваемого на любом структурном уровне. В текстовой модели на языке VHDL в этом случае должно быть явно выделено:

- идентификатор модели, т.е. блока;
- модель внешней границы как совокупности входов-выходов, т.е. свойств внешней границы блока;
- модель внутренней границы блока, как идентифицируемая совокупность внутренних переменных (инкапсулированные свойства метода);
- внутренняя модель блока, функциональная или структурная.

Все четыре компонента нуждаются в собственной нумерации прототипов. Это позволит рассматривать подмножества:

- моделей блоков  $B_i$ ;
- моделей внешней границы  $G_{ij}$ , соответствующих типу блока  $B_i$ ;

- моделей внутренней границы блока  $Vkji$ , соответствующих внешней границе  $Gij$ ;
- моделей функции или структуры блока, соответствующих внутренней границе  $Vkji$ .

Данный подход был ранее описан в [9].

Наличие таких текстовых фрагментов позволит создать модуль знаний, стыкуемый с другими модулями знаний на уровне списка внешних свойств блока.

### **Выводы**

В работе изложены следующие результаты:

- 1) Анализ средства автоматизации проектирования аппаратуры в среде языка VHDL;
- 2) Определение возможных путей создания интеллектуальной надстройки над САПР VHDL в среде специализированной инструментальной оболочки;
- 3) Анализ специфики языка VHDL с точки зрения формирования модулей знаний.

### **Литература**

1. The VHDL Cookbook First Edition. July, 1990 Peter J. Ashenden Dept. Computer Science University of Adelaide South Australia © 1990, Peter J. Ashenden.
2. Бибило П.Н. Основы языка VHDL. — Минск: Ин-т техн. кибернетики НАН Беларуси, 1999. — 202 с., ил.
3. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики — СПб.: БХВ–Петербург, 2002. — 608с.: ил.
4. Е.З. Перельройзен. Проектируем на VHDL — М.: СОЛОН–Пресс, 2004 — 448с.: ил.—(Серия «Библиотека профессионала»).
5. Продукты фирмы Aldec, Inc., [www.aldec.com.ua](http://www.aldec.com.ua).
6. Григорьев А.В. Семиотическая модель базы знаний САПР. Научные труды Донецкого государственного технического университета Серия: Проблемы моделирования и автоматизации проектирования динамических систем, выпуск 10: — Донецк, ДонГТУ, 1999. — С. 30–37.
7. Григорьев А.В. Уровни квалификации экспертов и пути создания интеллектуальных САПР. В этом же сборнике.
8. Григорьев А.В. Принципы организации вывода решений в базе знаний инструментальной оболочки для создания интеллектуальных САПР. // Практика і перспективи розвитку інституційного партнерства». Вісник ДонГТУ – ТРТУ. Донецьк: РВА ДонНТУ, 2003 — С.96–106.
9. А.В. Григорьев. Содержание некоторых категорий абстракций в теории интеллектуальных САПР. Материалы XVIII–й международной конференции "Интеллектуальные САПР — 2003" CAD–2003. Дивноморское, 3–10 сентября 2003, в 2–х томах. — Таганрог: Из-во ТРТУ. — Т.2., С. 11–18.

УДК 004.2 (045)

## ПРЕДСТАВЛЕННЯ ЦИФРОВОЇ ІНФОРМАЦІЇ У ПАРАЛЕЛЬНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМАХ НЕПОДІЛЬНИМИ КОДАМИ «М з N»

І.А.Жуков, В.О.Гуменюк, Н.В.Пашенко  
Національний авіаційний університет Інститут  
комп'ютерних технологій

### *Анотація*

*Розглядається можливість практичного вирішення проблеми автоматичного контролю в ЕОМ загалом і у високопродуктивних паралельних обчислювальних системах зокрема. Пропонується реалізація обчислювальних пристроїв, що працюють при нетрадиційному представленні цифрової інформації, на основі програмувальних логічних інтегральних схем — ПЛІС.*

При постійному підвищенні вимог до продуктивності обчислювальних засобів і достовірності результатів рішення фундаментальних і прикладних завдань, об'єднаних поняттям «*Grand challenges*», продовжує залишатися актуальною проблема реалізації масових паралельних обчислювальних процесів. Внаслідок цього виникають обмеження точності моделювання складних систем і процесів.

Практичне рішення проблеми підвищення продуктивності обчислювальних систем звичайно зв'язують, у першу чергу, зі збільшенням тактової частоти роботи елементів і кількості цих елементів, що дозволяє вводити паралелізм обробки й програмованість структури. Однак удосконалювання обчислювальних систем завжди супроводжував розрив між швидкодією логічних елементів і елементів пам'яті. Цей розрив при зростанні ступеня інтеграції й швидкодії великих інтегральних схем (ВІС) має тенденцію до збільшення.

Порівнювальний аналіз двох основних типів структур високопродуктивних обчислювальних систем: багатомашинних комплексів на базі мікропроцесорів з фіксованим по процесорах розподілом пам'яті й багатопроцесорних комплексів з пам'яттю, що розподіляється в процесі рахунку [1] надає можливість зробити наступні висновки.

Головним недоліком системи з архітектурою *SMP*, яка застосовує загальну пам'ять, поділовану всіма процесорами є наявність системної шини, що приводить до поганої масштабованості. Цей важливий недолік *SMP*-системи не дозволяє вважати їх по-справжньому перспективними. Причини поганої масштабованості полягають у тому, що в конкретний момент часу шина здатна обробляти тільки одну транзакцію, внаслідок чого виникають проблеми рішення конфліктів, при одночасному зверненні

декількох процесорів до одних й тих самих областей загальної фізичної пам'яті. Наявність такого конфлікту, залежить від продуктивності каналу зв'язку й від кількості обчислювальних елементів. Тепер конфлікти можуть відбуватися при включенні 24-х процесорів. Крім того, системна шина має обмежену пропускну здатність й обмежене число слотів. Все це, значною мірою, перешкоджає збільшенню продуктивності при збільшенні числа процесорів і числа користувачів, які підключаються. У реальних системах можна використати не більше 32 процесорів.

Головними недоліками масивної-паралельної архітектури, яка використовує фізично розділену пам'ять (система будується з окремих модулів, які містять процесор, локальний банк операційної пам'яті, два комунікаційних процесори (рутери) або мережний адаптер, іноді — жорсткі диски й/або інші пристрої уведення/виводу) є:

- помітне зниження швидкості міжпроцесорного обміну, внаслідок відсутності загальної пам'яті — відсутнє загальне середовище для збереження даних, призначених для обміну між процесорами;
- кожен процесор може використати тільки обмежений обсяг локального банку пам'яті.

Внаслідок зазначених архітектурних недоліків потрібні значні зусилля для того, щоб максимально використати системні ресурси. Саме цим визначається висока ціна програмного забезпечення для масивних-паралельних систем з розділеною пам'яттю.

Головна особливість гібридної архітектури *NUMA* — неоднорідний доступ до пам'яті. Суть цієї архітектури — в особливій організації пам'яті, а саме: пам'ять є фізично розподіленою по різних частинах системи, але логічно поділюваною, так що користувач бачить єдиний адресний простір. Система складається з однорідних базових модулів (плат), які складаються з невеликого числа процесорів і блоку пам'яті.

Всі центральні процесори системи одержують однакові значення змінних у будь-який момент часу. Оскільки кеш-пам'ять належить окремому комп'ютеру, а не всій багатопроцесорній системі в цілому, дані, які попадають у кеш одного комп'ютера, можуть бути недоступні іншому. Щоб уникнути цього, інформація, що зберігається в кеш-пам'яті процесорів, синхронізується.

Для забезпечення подібної когерентності кешів існують кілька можливостей:

- використання механізму відстеження шинних запитів (*snoopy bus protocol*), у якому кеші відслідковують змінні, передані до кожного із центральних процесорів й, при необхідності, модифікують власні копії таких змінних;

- виділення спеціальної частини пам'яті, відповідальної за відстеження вірогідності всіх використовуваних копій змінних.

Однак відомі в цей час методи забезпечення когерентності кеш-пам'яті при збільшенні кількості процесорів, що працюють на загальну пам'ять, істотно (у десятки й сотні разів) збільшують час доступу процесорів до даних оперативної пам'яті.

Отже необхідний пошук та дослідження додаткових засобів підвищення продуктивності паралельних обчислювальних систем.

Огляд засобів забезпечення достовірності результатів обчислень показує, що в більшості сучасних високопродуктивних мікропроцесорах широкого призначення (*UltraSparc III2*, *Itanium 2*, *Power4*, *PA-8700*) арифметико-логічний пристрій та реєстри не контролюються і лише в *SPARC64 V1* застосовується контроль парності, який вносить значні часові затримки в загальний процес обчислень. Крім того, цей метод контролю не забезпечує високого рівня достовірності результатів обчислень [1].

У [1,2] в якості альтернативного до традиційного двійкового представлення даних (у кожному розряді цифри представлені взаємозалежно) пропонується використання нероздільних кодів « $M$  з  $N$ » (де  $N$  — кількість позицій, що представляють один розряд числа;  $M$  — фіксована кількість «одиниць» у цих розрядах).

Співвідношення «нулів» й «одиниць» у рамках даного коду фіксовано. Величину основи  $p$  числення вибирають із умови:

$$C_N^M \geq p.$$

Такі коди є нероздільними, у яких розряди кодового слова неможливо розділити на інформаційні й контролюючі. Помилки перших двох ( $R_1$  й  $R_2$ ) категорій порушують співвідношення «нулів» й «одиниць» у даному коді й виявляються з методичною ймовірністю рівною 1 у термін  $t = a \cdot \Delta t_k$ , де  $\Delta t_k$  — розв'язна здатність автоматичного контролю за часом;  $0 < a < 1$  [1].

Внаслідок надзвичайно малого ступеня ймовірності двох і більше помилкових переходів  $0 \rightarrow 1$  і такої ж кількості помилкових переходів  $1 \rightarrow 0$  в одній кодовій комбінації за час  $t = a \cdot \Delta t_k$  на практиці ймовірність  $R_3$  помилки третьої категорії, що не виявляється може бути визначена наближеним вираженням:

$$R_3 \approx 0,9M(N - M) a^2 K_s^2 \cdot 10^{-27}.$$

Тобто, подання  $p$ -ічних цифр кодом « $M$  з  $N$ » дозволяє практично вирішити проблему автоматичного контролю обчислювальної системи.

Відмінною рисою функціонування таких пристроїв контролю є включення їхніх елементів тільки при виникненні помилок. Відпадає

необхідність у нагромадженні операндів у вхідних регістрах операційного пристрою й наступному їхньому контролю, що загалом приводить до значних непродуктивних часових витрат.

Аналіз властивостей таких кодів і структур відповідних обчислювальних пристроїв показує, що виконання в них функції контролю не пов'язане з додатковими часовими затримками й не знижує загальну продуктивність обчислювальної системи [1]. До переваг такого подання інформації можна віднести те, що кожний розряд числа може контролюватися окремо. Тому, наприклад, операції типу «додавання/вирахування» можна виконувати поетапно, із запам'ятовуванням переносів. Крім того, такий підхід має й наступні переваги:

- відповідним чином може бути скорочена розрядність суматора;
- скорочена й спрощена схема переносів має більшу швидкодію.

Тому доцільно дослідити можливості застосування кодів « $M$  з  $N$ » у високопродуктивних обчислювальних системах, для підвищення продуктивності її складових: процесорів, які входять у її состав, а також для підвищення достовірності результатів при рішенні критичних завдань. Крім того, у випадку фіксованого рівня загальної продуктивності обчислювальної системи, підвищення продуктивності окремих процесорів надає можливість для скорочення їхньої кількості, що сприяє подальшому поліпшенню багатьох параметрів обчислювальної системи.

Для реалізації схем контролю в обчислювальних пристроях передбачається використати програмувальні логічні інтегральні схеми — ПЛІС.

Крім збільшення логічних ресурсів і підвищення швидкодії за рахунок удосконалювання архітектурних і технологічних рішень, загальну тенденцію розвитку ПЛІС визначають два фактори, що забезпечують як підвищення продуктивності, так і розширення можливостей застосування. До цих факторів відносяться: вбудовування в кристал швидкодіючих модулів загального призначення й елементів, що забезпечують реалізацію стандартів вводу-виводу для здійснення взаємодії з різними зовнішніми пристроями. Аналіз параметрів різних типів ПЛІС показує, що на цей час одними з найбільш ефективними є пристрої компанії *Star Bridge Systems*, яка виробляє сімейство програмно реконфігуємих обчислювачів *Hypercomputer System HC-X* і пропонує комплексне рішення для організації реконфігуємих обчислень (рис. 1).

Старша модель сімейства, суперкомп'ютер *HC-98m*, складається з керуючого комп'ютера й двоплатного програмно реконфігуємого обчислювача, що включає 14 ПЛІС *Virtex-II* серії 6000 й 4 *Virtex-II* серії 4000. У сукупності це становить 98 млн. вентилів. Після включення



живлення перша ПЛІС програмується з постійного запам'ятовувального пристрою на платі.

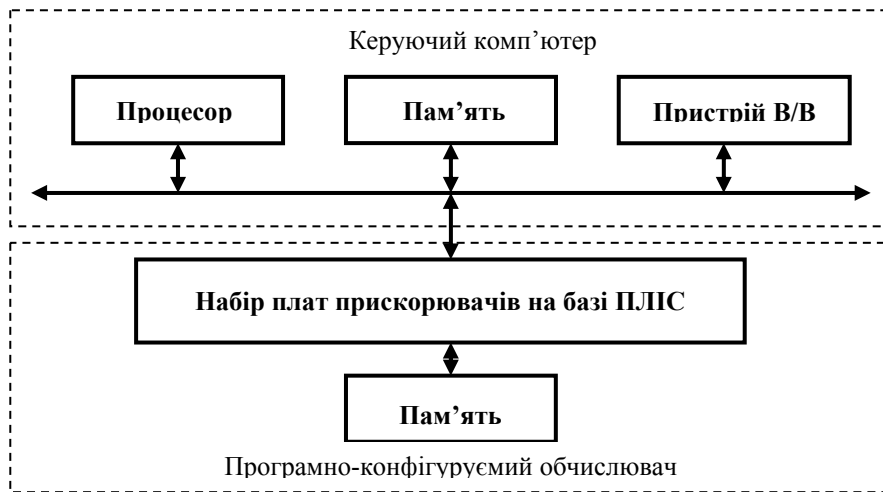


Рисунок 1 — Структура вузла універсальної обчислювальної системи, побудованого на ПЛІС

У цієї ПЛІС формується порт шини *PCI-X*, через який керуючий комп'ютер буде, використовуючи середовище розробки програм *Viva*, програмувати інші ПЛІС.

**Висновки.** Проблема автоматичного контролю в ЕОМ загалом і у паралельних обчислювальних системах зокрема може бути практично вирішена при представленні цифрової інформації неподільними кодами «М з N». При реалізації обчислювальних пристроїв, що працюють при такому нетрадиційному представленні цифрової інформації доцільно використати ПЛІС типу *Star Bridge Systems*.

### Література

1. Гуменюк В.А., Жуков И.А., Гуменюк А.В. Применение неразделимых кодов «М из N» в высокопроизводительных параллельных вычислительных системах// Проблемы інформатизації та управління. Зб. наук. праць. — 2004. — №11. С. 256–263.
2. Брюхович Е.И. О проблеме автоматического контроля в ЭВМ и контролеспособности позиционных счислений// УСиМ. — 1977. — №4. — С. 71–75.
3. Палагин А.В., Опанасенко В.Н., Сахарин В.Г. Вычислительные системы с реконфигурируемой (программируемой) архитектурой.// Проблемы інформатизації та управління. Зб. наук. праць. — 2004. — №10. — С. 5–14.

## **МОДЕЛИРОВАНИЕ МУЛЬТИАГЕНТНЫХ АЛГОРИТМОВ МАРШРУТИЗАЦИИ СООБЩЕНИЙ В КОМПЬЮТЕРНЫХ СЕТЯХ**

Ю.В. Ладыженский, А.В. Мирецкий, В.А. Мирецкая  
Кафедра ПМиИ, ДонНТУ

### ***Анотація***

*Представлений дискретно-подійний симулятор, що дозволяє досліджувати властивості алгоритмів маршрутизації. Описано основні об'єкти моделі комп'ютерної мережі. Розглянуто динамічний алгоритм маршрутизації AntNet.*

### ***Введение***

В настоящее время компьютерные сети продолжают расти и изменяться. Для обеспечения качественного обслуживания в сетях необходим механизм, позволяющий быстро и без потерь доставлять данные от одного пользователя к другому. Центральное место в этом механизме отводится маршрутизации. Правильно организованная маршрутизация в совокупности с компонентами управления доступом, потоками и перегрузкой обеспечивает качественное обслуживание в сети.

Известны исследования, показавшие, что в некоторых случаях в условиях постоянного роста трафика имеет место снижение производительности сети [1]. В большей части таких случаев, производительность падает не из-за поступающего трафика, а из-за маршрутизаторов, не справляющихся с поддержанием актуальности таблиц маршрутизации и выбором оптимальных маршрутов для данного класса трафика. Эти проблемы характерны при использовании сетей Ethernet и глобальных сетей с коммутацией пакетов для передачи аудио- и видеоданных [1]. Для решения возникшей проблемы следует или модернизировать уже существующие алгоритмы маршрутизации, или найти новые подходы к задаче маршрутизации. Для определения эффективности алгоритма маршрутизации, необходимо построить модель сети с исследуемым алгоритмом маршрутизации, в которой происходят процессы, приближенные к реальности. Наиболее важными показателями эффективности работы сети с исследуемым алгоритмом маршрутизации являются пропускная способность и задержка пакетов.

Одним из подходов к решению проблемы маршрутизации является применение муравьиных алгоритмов [2].

**Модель компьютерной сети. Дискретно-событийный симулятор.** Модель компьютерной сети представлена как направленный взвешенный граф с  $N$  узлами, которые обрабатывают и перенаправляют данные (см. рис. 1). Все линии связи рассматриваются как каналы передачи битов, характеризующиеся пропускной способностью (бит/сек) и задержкой передачи (сек), доступ к ним осуществляется по мультиплексной схеме. Узлы в модели действуют одновременно и как хосты (конечная точка сессии), и как шлюзы (маршрутизаторы, элементы перенаправления данных).

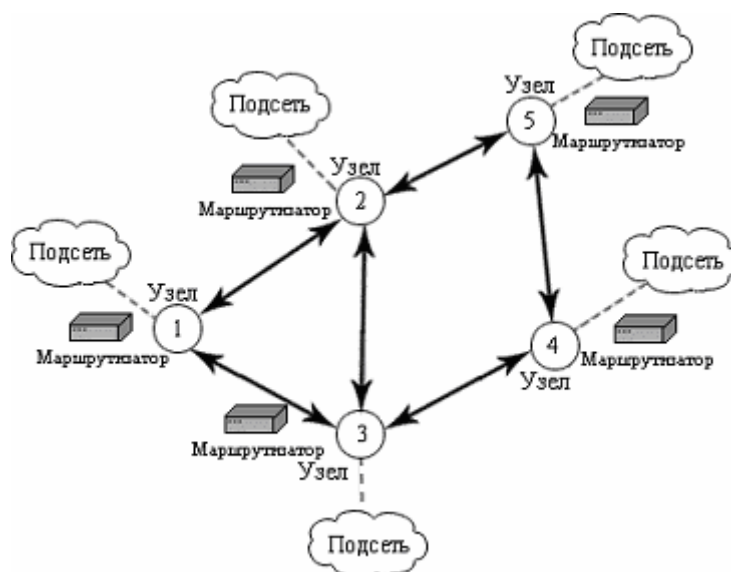


Рисунок 1 — Модель компьютерной сети

Узел компьютерной сети состоит из двух процессоров. Первый процессор является процессором маршрутизации. Второй — коммуникационный процессор, который отвечает за прием и отправку пакетов по каналам передачи данных. Он представляет собой совокупность из  $N_i$  одинаковых наборов компонент (устройств RX/TX для приема/передачи данных по каналу), где  $N_i$  — количество каналов передачи, связанных с  $i$ -м узлом (см. рис. 2).

Для организации очередей пакетов, а также для хранения обрабатываемых пакетов, в узле имеется специальный буфер памяти. Если этот буфер в процессе работы заполнится (не будет достаточно свободной памяти для хранения вновь прибывших пакетов), то будут иметь место отказы в обслуживании (другими словами, потери пакетов).

Приемник RX получает пакет из канала передачи данных. Если в буфере имеется достаточно места для хранения пакета, то пакет

ставится в очередь IN, иначе имеет место отказ в обслуживании. Как только маршрутизатор освобождается от обработки предыдущего пакета, ему на обработку передается пакет из очереди IN. После обработки, пакет попадает в очередь OUT или в приоритетную очередь OUT, в зависимости от своего приоритета. Как только передатчик TX освобождается от отправки предыдущего пакета, мультиплексор передает ему новый пакет, взятый из какой-либо очереди OUT. Выбор очереди осуществляется в зависимости от алгоритма и от наличия пакетов в очередях. Передатчик TX передает пакет по каналу следующему узлу.

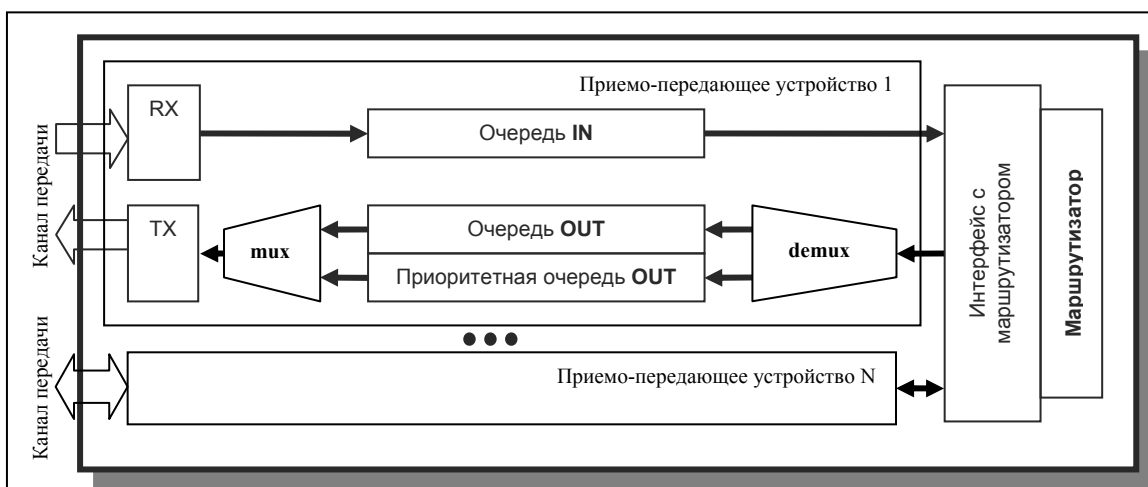


Рисунок 2 — Модель узла компьютерной сети

Пакет при попадании в маршрутизатор ставится в массив обрабатываемых пакетов на время своей обработки. При помощи массива маршрутизатор моделируется как многоканальное устройство. В маршрутизаторе реализованы 2 алгоритма обработки пакетов: алгоритм маршрутизации пакетов данных и алгоритм обработки пакетов маршрутизации. Выбор того или иного алгоритма зависит только от типа поступившего пакета.

Модель рабочей нагрузки объединяет простые механизмы управления потоками, реализованные с использованием фиксированного окна производительности для сгенерированных пакетов в сессии. Окно определяет максимальное количество пакетов данных, ожидающих отправки. Однажды посланный, пакет рассматривается как подтвержденный. Это означает, что на транспортном уровне нет контроля ошибок и подтверждений получения.

Для выполнения моделирования описанной системы построен дискретно-событийный симулятор. Главная структура данных в этом

симуляторе — это очередь записей о событиях, отсортированных по времени наступления. Специальный объект Планировщик управляет этой очередью. Текущее время моделирования изменяется в соответствии со временем текущего выполненного события. Каждое событие при выполнении передается на обработку соответствующему обработчику. Так, например, маршрутизатор при получении пакета планирует событие «Окончание обработки пакета». При выполнении этого события, маршрутизатор освобождается от обработки пакета.

*Мультиагентный адаптивный алгоритм маршрутизации AntNet.* Построенный симулятор использовался для моделирования коммуникационной сети с алгоритмом маршрутизации пакетов данных AntNet.

AntNet — это адаптивный, распределенный, базирующийся на агентах алгоритм маршрутизации [2].

Данный алгоритм основывается на поведении муравьев в природе. Поведение этих насекомых основано на роевом интеллекте (Swarm Intelligence). Различают два вида взаимодействия между особями [3]:

- прямое взаимодействие (обмен пищей, визуальный контакт, химический контакт и др.);

- косвенное взаимодействие или стигмержи (stigmergy) — две особи взаимодействуют косвенно, когда одна из особей модифицирует окружающую среду, а другая, со временем, реагирует на это изменение. В природе косвенное взаимодействие осуществляется через феромон (pheromone) — специальный, довольно стойкий секрет, оставляемый как след при перемещении насекомого. Чем больше концентрация феромона на тропе, тем больше муравьев будет по ней двигаться. Со временем феромон испаряется, что позволяет муравьям адаптировать свое поведение под изменения внешней среды.

Процедура поиска кратчайших путей с помощью алгоритма маршрутизации данных AntNet состоит в следующем. Представим компьютерную сеть как окрестность, в которой взаимодействуют муравьи, при этом узлы этой сети являются источниками пищи для муравьев. Узлы соединены линиями связи, на которых муравьи могут оставлять след феромона. Каждый узел в сети характеризуется двумя структурами данных: локальной моделью трафика и вероятностной таблицей маршрутизации. Муравьев в такой сети будем называть мобильными агентами. Каждый мобильный агент движется от одного узла сети к другому (от муравейника к источнику пищи), при этом собирая информацию о

линиях связи, соединяющих эти узлы. Чтобы распространить собранную информацию по сети, агент должен вернуться в узел, отправивший его. Для этого у каждого агента существует память, организованная в виде стека, в которую записан весь маршрут движения агента от узла-источника в узел-приемник. При выборе пути движения от узла-источника в узел-приемник через промежуточные узлы агенты используют вероятностное правило выбора пути. Вероятность выбора пути прямо пропорциональна количеству феромонов на этом пути с учетом загруженности этого пути. То есть, линии связи, содержащие большее количество феромонов, среди малонагруженных линий, являются наиболее удобными маршрутами для пересылки пакетов данных в сети.

В алгоритме AntNet мобильные агенты постоянно исследуют состояние сети и производят поиск оптимальных маршрутов для передачи пакетов данных. Агенты отмечают оптимальные пути при помощи феромонного следа, т.е. по их данным вносятся определенные изменения в таблицы маршрутизации. Агенты поддерживают таблицы маршрутизации узлов сети в актуальном состоянии.

### ***Заключение***

Разработан и реализован дискретно-событийный симулятор, позволяющий произвести моделирование работы компьютерной сети и алгоритма маршрутизации, который используется в маршрутизаторах сети. Симулятор выполнен таким образом, что к нему можно подключать внешние моделирующие модули с другими алгоритмами маршрутизации без перекомпиляции и без внесения изменений в исходный код симулятора.

В реальности в компьютерной сети могут иметь место обрывы линий связи, подключение новых узлов к сети и другие исключительные ситуации. В дальнейших работах планируется реализация в симуляторе и этих ситуаций с целью приблизить модель к реальной компьютерной сети.

### **Литература**

1. Столлингс В. Современные компьютерные сети. 2-е изд. — СПб: Питер, 2003. — 783 с.
2. Di Caro G., Dorigo M. AntNet: Distributed Stigmergetic Control for Communications Networks: Journal of Artificial Intelligence Research. — 1998. — №9 — pp. 317–365.
3. С.Д. Штовба Муравьиные алгоритмы: Мастерская решений: Exponenta Pro. — 2003. -№4(4) — сс. 70–75.

## МОДЕЛИРОВАНИЕ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ ПОИСКА НЕЗАВИСИМЫХ МНОЖЕСТВ НА ГРАФАХ

Ю.В. Ладыженский, В.А. Куркчи  
Кафедра ПМИИ, ДонНТУ

### **Анотація**

*Представлено експериментальні результати порівняння властивостей алгоритмів Люби, Голдберга-Спенсера і поліноміального алгоритму. Показано, що недетермінований алгоритм Люби є найбільш ефективним.*

### **Введение**

Задача поиска наибольшего независимого множества вершин графа возникает при распараллеливании вычислительных процессов и производственных работ [1,2]. Именно поэтому исследователи уделяют ей много внимания.

Разработан ряд параллельных эвристических и метаэвристических алгоритмов для решения этой задачи. Авторы анализировали потенциальные достоинства и недостатки алгоритмов. А вопросы точности и быстродействия алгоритмов рассматривались теоретически либо не рассматривались вообще [3,4].

Целью данной работы является проведение экспериментального сравнения параллельных алгоритмов: недетерминированного алгоритма Любы [5], алгоритма Голдберга-Спенсера [6], полиномиального алгоритма [7].

### **Постановка задачи**

Граф  $G(V, E)$  состоит из непустого множества вершин  $V$  ( $V = \{v_1, v_2, \dots, v_n\}$ ) и множества ребер  $E$  ( $E = \{(v_i, v_j)\}, i \neq j, |E| = m$ ), состоящего из неупорядоченных пар различных элементов множества  $V$ .

Функция соответствия графа  $\Gamma: V \rightarrow 2^V$ , определяет множество вершин, смежных с заданной:

$$\Gamma(v_i) = \{v_k \mid (v_i, v_k) \in E\}.$$

Подмножество вершин графа  $S$  называют независимым, если любые две вершины, принадлежащие  $S$ , несмежны:

$$S = \{v_i \in V \mid \Gamma(v_i) \cap S = \emptyset\}.$$

Независимое подмножество вершин  $I$  максимально, если добавление в него любой вершины нарушит условие независимости. Каждая вершина графа  $G$ , не принадлежащая  $I$ , смежна, по меньшей мере, с одной из вершин, принадлежащих  $I$ :

$$I = \{v_i \mid \Gamma(v_i) \cap S = \emptyset \ \& \ \forall (v_j \in V \setminus I) \exists v_k \in I : (v_j, v_k) \in E\}$$

Пусть множество  $W$  есть множество всех максимальных независимых множеств графа.

Задача о наибольшем независимом множестве вершин графа заключается в нахождении множества  $S \in W$ , для которого

$$|S| = \max_{I \in W} |I|.$$

Допустимым приближенным решением задачи считается любое максимальное независимое множество.

### **Описание алгоритмов**

Алгоритм Любы [5] начинает с пустого множества  $S$  и итерационно расширяет его с сохранением независимости до получения максимального независимого множества. Для этого на каждой итерации строится немаксимальное независимое множество  $I$  следующим образом. В начале итерации множество  $I$  пусто. Каждая вершина из  $V$  добавляется в это множество с вероятностью  $(2d)^{-1}$ , где  $d$  — степень вершины в исходном графе. Если полученное множество не является независимым, его независимость восстанавливается: для каждой пары смежных вершин, принадлежащих множеству  $I$ , из него удаляется вершина с меньшей степенью. Когда немаксимальное независимое множество получено, оно добавляется к множеству  $S$  и удаляется из графа со всеми смежными с ним вершинами. Итерации продолжаются пока множество вершин графа не пусто.

Алгоритм Любы рассчитан на параллельную вычислительную модель с  $\max(n, m)$  процессорами и общей памятью. Это позволяет выполнять любые операции над множествами за одну операцию: каждый процессор обрабатывает один элемент множества.

Алгоритм Голдберга-Спенсера [6] также начинает с пустого множества  $S$  и итерационно расширяет его с сохранением независимости до получения максимального независимого множества. Но немаксимальное множество  $I$  строится по-другому. Для построения немаксимального независимого множества алгоритм Голдберга-Спенсера использует раскраски графа. В раскрашенном графе вершины одного цвета несмежны [8], поэтому цветовой класс  $K \subseteq V$  — множество вершин одного цвета — является независимым множеством.

Изначально каждая вершина получает свой цвет. Затем цвета попарно объединяются до получения единственного цветового класса или класса, удовлетворяющего условию



$$\left| \bigcup_{v_i \in K} \Gamma(v_i) \right| \geq (n + m) / \log_2 n.$$

Пусть  $C = \{c_1, c_2, \dots, c_r\}$  — множество цветов, использованных в текущей раскраске графа. Соответственно  $K_i$  — множество вершин, окрашенных в цвет  $c_i$ . Существует

$$\frac{r!}{2^{r/2} (r/2)!} \text{ (для четного } r \text{)}$$

разбиений цветов на пары. Поэтому среди множества всех возможных разбиений выбирается только  $r$  из них. Рассматриваются разбиения цветов на пары  $\pi_k = \{(c_i, c_j) \mid \text{index}(i, j, r) = k\}$ , где

$$\text{index}(i, j, r) = \begin{cases} (i + j) \bmod \chi(r), & 0 \leq i, j \leq \chi(r) - 1 \\ 2i \bmod \chi(r), & j = \chi(r) \\ 2j \bmod \chi(r), & i = \chi(r) \end{cases}$$

$$\chi(r) = \begin{cases} r, & r \bmod 2 = 1 \\ r - 1, & \text{иначе} \end{cases}.$$

Для каждого  $\pi_k$  определяется величина

$$f(\pi_k) = \sum_{(c_i, c_j) \in \pi_k} |W_{ij}|,$$

где  $W_{ij} = \{(v_a, v_b) \in E \mid v_a \in K_i \ \& \ v_b \in K_j\}$  — множество ребер между цветовыми классами  $K_i$  и  $K_j$ .  $f(\pi_k)$  определяет количество вершин, которые нарушат условие независимости цветовых классов, если объединить цветовые классы согласно разбиению  $\pi_k$ .

Затем выбирается разбиение с наименьшим значением  $f$ . Для выбранного разбиения производится попарное объединение цветовых классов, при этом вершины  $v_l$ , удовлетворяющие условию

$$\exists (c_i, c_j) \in \pi_k \mid (v_l, v_a) \in W_{ij}$$

теряют свой цвет для сохранения независимости цветовых классов.

Когда большой цветовой класс найден, он рассматривается как немаксимальное независимое множество  $I$ . Также как и алгоритм Любы, алгоритм Голдберга-Спенсера добавляет множество  $I$  к множеству  $S$  и удаляет его из графа со всеми смежными с ним вершинами. Итерации продолжаются пока множество вершин графа не пусто. На всех итерациях алгоритма, кроме первой, для поиска множества  $I$  используется раскраска, полученная на предыдущей итерации.

Алгоритм Голдберга-Спенсера рассчитан на параллельную вычислительную модель с  $27(n + 2m)/7$  процессорами и общей памятью. Алгоритм использует список вершин и ребер и выполняет все указанные выше операции с помощью сортировок этого списка.

Полиномиальный алгоритм [7] использует вектор переменных  $\bar{x} = \{x_i\}$ , каждая из переменных которого соответствует одной вершине графа ( $x_i \Leftrightarrow v_i$ ), и вычисляет:

$$\max_{x \in [0,1]^n} F(\bar{x}) = \sum_{i=1}^n \left( (1 - x_i) \prod_{j \in \Gamma(i)} x_j \right),$$

Целевая функция представляется в виде

$$F(\bar{x}) = (1 - x_i)A_i(\bar{x}) + x_i B_i(\bar{x}) + C_i(\bar{x}),$$

где

$$A_i(\bar{x}) = \prod_{j \in \Gamma(i)} x_j, \quad B_i(\bar{x}) = \sum_{k \in \Gamma(i)} \left( (1 - x_k) \prod_{j \in \Gamma(k), j \neq i} x_j \right), \quad C_i(\bar{x}) = \sum_{k \notin \Gamma(i)} \left( (1 - x_k) \prod_{j \in \Gamma(k)} x_j \right).$$

Начиная с некоторого начального решения  $\bar{x}^{(0)}$ , алгоритм последовательно вычисляет

$$x_i^{(i)} = \begin{cases} 1, & A_i(\bar{x}^{(i-1)}) \leq B_i(\bar{x}^{(i-1)}), \\ 0, & \text{иначе} \end{cases}, \quad \forall i = \overline{1, n}.$$

Максимальное независимое множество составляет из вершин  $v_j$ , для которых  $x_j^{(n)} = 0$ .

Полиномиальный алгоритм рассчитан на параллельную вычислительную модель с  $n(n + 1)$  процессорами и общей памятью.

### **Методика эксперимента**

Для получения оценки временной сложности алгоритмов, каждый алгоритм разбивался на шаги. Экспериментальная оценка временной сложности  $T$  алгоритмов определялась как  $T = T_a * T_s$ , где  $T_a$  — количество выполненных алгоритмом шагов, а  $T_s$  — количество операций, выполняемых алгоритмом за один шаг.

Для алгоритма Любы шагом является итерация. Временная сложность параллельного выполнения итерации составляет  $\lceil \log_2 n \rceil + 8$  операции [5]. Для алгоритма Голдберга-Спенсера шагом является сортировка, которая параллельно выполняется за время, равное  $426 + 2970 \lceil \log_2(n + 2m) \rceil$  [8,9].

Начальное значение вектора переменных полиномиального алгоритма —  $1 - 1/d_i$ , где  $d_i$  — степень вершины  $v_i$ . Можно показать, что полиномиальный алгоритм выполняет количество операций, равное:

$$\max_i d_i + \sum_{i=1}^n \lceil \log_2 d_i \rceil + 5n.$$

Для сравнения точности алгоритмов сравнивались отношения мощности найденных ими независимых множеств к мощности множества, найденного алгоритмом Любы.

Чтобы оценить зависимость исследуемых величин от плотности графа, для каждого значения плотности ( $D = 2 \lfloor E \rfloor / (n(n-1))$ ) генерировалось 10 случайных графов. Данные по этим 10 графам усреднялись. Рассматривались графы на 300 вершинах.

Для генерации случайных графов и чисел использовался генератор псевдослучайных последовательностей с равномерным распределением Льюиса-Гудмана-Миллера [10].

Таблица 1 — Фрагмент результатов работы алгоритма Любы

D	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>
0,1	37	32	34	1105	619	902
0,2	18	18	20	1141	904	1622
0,3	12	13	14	665	709	2779
0,4	11	9	11	2545	739	3919
0,5	8	9	9	2790	4822	4683
0,6	6	6	8	586	507	3263
0,7	5	6	6	1929	1136	941
0,8	4	4	4	853	785	1002
0,9	4	3	3	376	721	438

### *Результаты экспериментов*

Фрагмент результатов работы алгоритма Любы представлен в таблице 1, где  $I_i$  — размер найденного множества для графа  $i$  с заданной плотностью,  $T_i$  — количество выполненных операций при обработке того же графа. В таблице 2 представлены фрагменты результатов работы алгоритма Голдберга-Спенсера и полиномиального алгоритма, где  $I'_i$  — отношение размера найденного независимого множества к размеру множества, найденному алгоритмом Любы. Средние значения точности и производительности исследуемых алгоритмов приведены в таблице 3.

На рис. 1 представлен сравнительный график зависимостей точности алгоритмов от плотности графа, где G-S — означает алгоритм Голдберга-Спенсера, а Poly — полиномиальный алгоритм. Из рисунка видно, что полиномиальный алгоритм имеет в среднем такую же точность, как и алгоритм Любы, а алгоритм Голдберга-Спенсера находит решения худшего качества. Алгоритм Голдберга-Спенсера работает медленнее других алгоритмов, однако на графах свыше 450000 вер-

шин имеет теоретическую временную сложность меньшую, чем полиномиальный алгоритм.

Таблица 2 — Результаты работы алгоритма Голдберга-Спенсера и полиномиального алгоритма

Голдберг-Спенсер						Полиномиальный					
$\Gamma_1$	$\Gamma_2$	$\Gamma_3$	$T_1$	$T_2$	$T_3$	$\Gamma_1$	$\Gamma_2$	$\Gamma_3$	$T_1$	$T_2$	$T_3$
0,89	1,00	1,03	3,E+06	3,E+06	3,E+06	0,92	1,00	1,00	3148	3157	3168
1,11	0,89	0,80	3,E+06	2,E+06	2,E+06	1,11	1,11	1,00	3471	3453	3474
1,08	1,00	0,93	3,E+06	3,E+06	3,E+06	1,17	1,08	0,86	3709	3710	3710
0,91	1,11	0,91	2,E+06	3,E+06	2,E+06	0,91	1,11	0,91	3797	3807	3802
1,00	0,67	0,89	2,E+06	2,E+06	2,E+06	1,00	0,78	0,78	4075	4075	4069
0,83	1,00	0,88	1,E+06	1,E+06	2,E+06	0,83	1,00	0,88	4105	4100	4099
1,00	0,83	0,83	1,E+06	2,E+06	2,E+06	1,00	0,83	0,83	4135	4140	4134
1,00	1,00	1,00	1,E+06	1,E+06	1,E+06	1,25	1,25	0,75	4158	4161	4158
0,50	1,00	1,00	1,E+06	1,E+06	1,E+06	0,75	0,67	1,00	4485	4482	4482

Таблица 3 — Средние значения точности и производительности

Любы	Голдберг-Спенсер		Полиномиальный	
	$T_{cp}$	$\Gamma_{cp}$	$T_{cp}$	$\Gamma_{cp}$
1044,3	3255837,9	0,95	3153,3	0,97
1224,6	2659937,8	0,94	3467,8	1,06
1287,4	2592009,2	0,98	3710,7	1,01
1978,9	2345106,2	1,00	3798,4	1,01
3106,7	2071431,6	0,91	4073,1	0,99
1375,6	1635339,3	0,91	4102	0,92
1482,2	1587775,1	0,93	4132,9	1,04
971,9	1376218,2	0,94	4159,4	0,99
581,8	1299555,7	0,86	4481,4	0,93

### Выводы

Алгоритм Любы и полиномиальный алгоритм имеют приблизительно одинаковую точность, но алгоритм Любы работает быстрее, что делает его привлекательным для практического использования.

С другой стороны, полиномиальный алгоритм зависит от начального приближения, поэтому вероятно, что его точность можно повысить за счет изменения правила построения начальных приближений. Кроме того, вычислительный процесс полиномиального алгоритма основан на формулах, допускающих структурную реализацию на программируемых СБИС.

Целью дальнейших исследований является построение правил выбора начальных приближений и методов структурной реализации алгоритмов решения задачи о наибольшем независимом множестве.

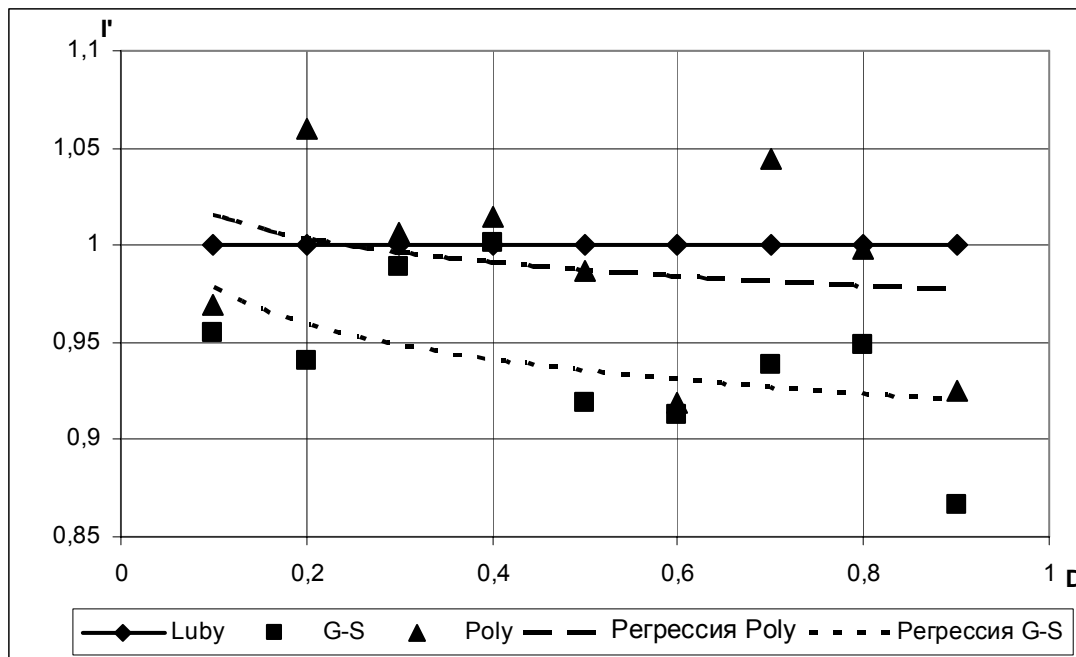


Рисунок 1 — Сравнительный график зависимостей точности от плотности для различных алгоритмов

### *Литература*

1. Deo N. Graph theory with applications to engineering and computer science. Prentice-Hall, 1974.
2. Berge C. The theory of graphs and its applications. Methuen, 1962.
3. Bomze I. M., Budinich M., Pardalos P.M., Pelillo M. The maximum clique problem. In D.-Z. Du and Pardalos P.M., editors, Handbook of Combinatorial Optimization, pp 1–74. Kluwer Acad. Publishers, 1999.
4. Blum C., Roli A. Metaheuristics in combinatorial optimization: overview and conceptual comparison. In ACM Computing Surveys, Vol. 35, No. 3, September 2003, pp. 268–308.
5. Luby. M. A simple parallel algorithm for the maximal independent set problem. SIAM J. Comput., 4:1036–1053, 1986.
6. Goldberg M., Spenser T., A new parallel Algorithm for the maximal independent set problem, SIAM J. on Comp. 18, pp 419–427, 1989.
7. Abello J., Butenko S., Pardalos P.M. Finding independent set in a graph using continuous multivariable polynomial formulation. AT&T Labs technical report, 2000.
8. Cole R., Parallel merge sort.//Proc. 27<sup>th</sup> Annual Symp. On Foundations of Computer Science. — 1986.— pp. 511–516.
9. Natvig L., Logarithmic time cost optimal parallel is not yet fast in practice., Supercomputing'90, NewYork, 1990.
10. Кнут Д.Э. Искусство программирования, том 2. Получисленные алгоритмы, 3–е изд.: Пер. с англ. – М.: Издательский дом "Вильямс", 2003. — 832 с.: ил. — Парал. тит. англ.

# КЛАССИФИКАЦИЯ МЕТОДОВ БАЛАНСИРОВКИ НАГРУЗКИ ПРОЦЕССОРОВ ПРИ РАСПРЕДЕЛЕННОМ МОДЕЛИРОВАНИИ ЦИФРОВЫХ УСТРОЙСТВ

Д.В. Бельков  
Кафедра ВМиП, ДонНТУ

## *Abstract*

*This paper reviews the major load balancing algorithms and discusses their relative merits on distributed simulation of digital devices. The classification of load balancing methods is proposed.*

## *Введение*

В условиях ограниченности вычислительных ресурсов существенно снизить затраты на моделирование цифровых устройств (ЦУ) за счет параллельных вычислений позволяет использование компьютерных сетей. Важной задачей распределенного моделирования ЦУ является разбиение схемы на подсхемы и распределение подсхем на моделирующие процессоры. Исходную схему можно представить в виде графа, в котором вершины соответствуют элементам, а ребра — электрическим цепям исходной схемы. В таком случае задача разбиения схемы на подсхемы формулируется как задача разбиения графа на подграфы. Алгоритмы разбиения графов должны обеспечивать балансировку нагрузки процессоров и минимальное число межпроцессорных обменов [1].

Таким образом, при распределенном моделировании ЦУ на этапе разбиения графа схемы на подграфы для повышения эффективности моделирования необходимо решить задачу балансировки нагрузки процессоров.

Применение статических методов балансировки нагрузки может привести к блокировке процессов, заранее распределенных по процессорам. Этому недостатка лишены динамические методы балансировки, которые допускают миграцию процессов по процессорам в зависимости от нагрузки.

Целью данной статьи является классификация методов динамической балансировки, которые можно применять при распределенном моделировании цифровых устройств.

## *Классификация методов балансировки нагрузки*

Простой моделью параллельных вычислений является модель Master/Slave. Имеется главный (мастер) процессор и подчиненные (slave) процессоры. Подчиненные процессоры выполняют вычисления и обращаются к мастеру для дальнейшей обработки. Существует много различных вариантов этой модели, например, может быть

несколько типов главных процессоров или иерархия подчиненных процессоров. Однако подход Master/Slave целесообразно использовать только, если вычислительные задания выполняются независимо и асинхронно каждым процессором. Количество сообщений между мастером и подчиненными процессорами должно быть небольшим [2].

В работе [3] предложена система параллельного моделирования цифровых устройств dlbSIM. Она содержит главный (master) компонент и множество подчиненных (slave) компонент. Блок управления загрузкой системы dlbSIM показан на рисунке 1.

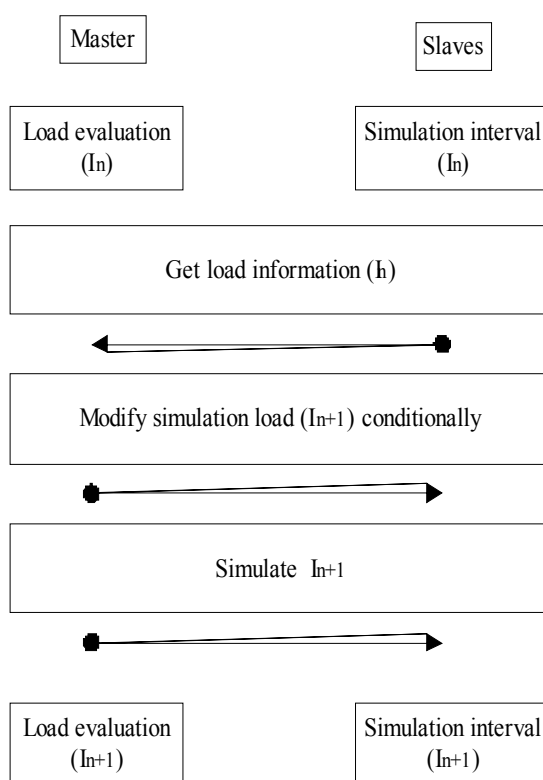


Рисунок 1 — Блок управления загрузкой системы dlbSIM

Каждый подчиненный компонент запускается на отдельном процессоре и выполняет процесс моделирования одной или нескольких подсхем, состоящий из 4 шагов:

1. CLOCK — выполнение моделирования всех подсхем, назначенных на процессор, в течение заданного интервала моделирования.
2. GET — Запись управляющих сигналов в выходные порты процессора.
3. TRANSFER — синхронизация работы подчиненных компонент.
4. PUT — Чтение данных, необходимых для продолжения моделирования из входных портов процессора.

Балансировка загрузки процессоров выполняется главным компонентом на шаге TRANSFER. Целью является сокращение времени моделирования всей схемы за счет выравнивания интервалов моделирования подсхем на каждом процессоре.

Блок управления загрузкой, используя информацию о загрузке процессоров, оценивает и модифицирует уровень их загрузки. В течение интервала моделирования  $I_n$  (шаг CLOCK), подчиненные компоненты работают независимо от главного компонента. После завершения интервала  $I_n$  информация о загрузке процессоров передается главному компоненту, который сравнивает ее с информацией о загрузке процессоров на интервале  $I_{n-1}$  и принимает решение о необходимости модификации загрузки. Для эффективной работы системы моделирование подсхем должно происходить независимо и асинхронно на каждом из подчиненных компонентов.

Другим примером модели Master/Slave является алгоритм инициативного сервера [4]. Владеющий маркером узел (мастер), если его нагрузка меньше пороговой величины, ищет наиболее нагруженный узел сети и забирает у него такое количество запросов, которое необходимо для достижения пороговой величины. Поиск наиболее нагруженного узла выполняется путем послыки сообщений ко всем другим узлам и получения от них соответствующих ответов о нагрузке. После завершения процесса балансировки нагрузки маркер передается узлу с минимальной нагрузкой.

Геометрический подход разбиения графов требует, чтобы вершины графа имели геометрические координаты. Это возможно не для всех графов. Геометрические алгоритмы идеальны для параллельного выполнения, но не учитывают структуру графа [2].

Алгоритм RCB (Recursive Coordinate Bisection) был впервые предложен Бергером и Бохари как статический алгоритм. В нем используются разрезающие плоскости, которые ортогональны одной из координатных осей и выбирается плоскость, разрезающая наименьшее число ребер графа. Алгоритм RCB разделяет граф на части без учета их свойств. Этого недостатка нет у метода URВ (Unbalanced Recursive Coordinate Bisection), который разделяет граф на  $q$  частей при соотношении их размеров:  $1:(q-1)$  или  $2:(q-2)$  и т.д. Простота и быстродействие RCB и URВ привлекательны для динамической балансировки, однако недостатком этих алгоритмов является плохое качество разбиения графа.

Алгоритм RIB (Recursive Inertial Bisection) использует понятия механики. Целью является определение центра масс и направления инерционного движения, вдоль которого вершины графа имеют минимальный вращающий момент. Разрезающая плоскость выбирается так, чтобы она была ортогональна к этому направлению [2].



Известен метод, который взамен разрезающей плоскости использует сферу. Ее внутренняя часть соответствует одному разделу графа, а внешняя часть — другому. Этот алгоритм является более сложным и затратным, чем алгоритмы, использующие разрезающие плоскости, но может быть легче распараллелен [2].

При использовании глобальных методов балансировки нагрузки процессоры выполняют вычисления синхронно, а затем следует фаза балансировки. При этом недогруженные процессоры вынуждены простаивать, ожидая пока сильно загруженные процессоры, завершат вычисления. Только после этого выполняется процесс балансировки.

Локальные методы балансировки нагрузки бывают как синхронными, так и асинхронными. Соседние процессоры могут выполнять балансировку, в то время как другие процессоры заняты вычислениями. Процессом балансировки можно управлять с помощью запросов, поступающих от процессоров, когда они простаивают [4].

Многие локальные методы имеют 2 шага. На первом шаге определяется, сколько работ будет перемещаться от одного процессора к соседнему для балансировки нагрузки. На втором шаге используются эвристики для выбора объектов, которые будут мигрировать. Эти эвристики минимизируют коммуникационные затраты. Большинство работ основано на алгоритме Кернигана-Лина (KL) или алгоритме KL/FM [1].

При определении потока работ может быть использован алгоритм Кубенко [2]. Считается, что рабочая нагрузка удовлетворяет уравнению  $\frac{\partial u}{\partial t} = \alpha \nabla^2 u$ , где  $u$  — рабочая нагрузка,  $\alpha$  — коэффициент диффузии. Для решения уравнения применяется метод конечных разностей. Рабочая нагрузка на итерации  $(t+1)$  вычисляется по формуле  $u_i^{t+1} = u_i^t + \sum_j \alpha_{ij} (u_j^t - u_i^t)$ , где  $u_i^t$  — рабочая нагрузка на итерации  $t$ ,

$j$  — номер процессора. Если  $i$ -й и  $j$ -й процессоры не соединены, то  $\alpha_{ij} = 0$ , иначе  $\alpha_{ij} > 0$  и  $1 - \sum_j \alpha_{ij} \geq 0$ .

Метод Ху и Блейка [2] минимизирует поток работ через ребра графа процессоров методом сопряженных градиентов. Решается система уравнений  $LX = B$ , где  $B_i$  — разность между загрузкой процессора  $i$  и средней загрузкой процессоров. Элементы вектора  $X$  должны принимать значения 1 или  $-1$ :  $X_i = 1$ , если вершина  $i$  должна быть в первом подграфе и  $X_i = -1$  если вершина  $i$  должна быть во втором подграфе,  $\sum_i X_i = 0$ . Значение  $L_{ij}$  равно весу вершины  $i$ , если

$i=j$ . Если  $i \neq j$  и вершины  $i$  и  $j$  соединены, то  $L_{ij} = -1$ . Если  $i \neq j$  и вершины  $i$  и  $j$  не соединены, то  $L_{ij} = 0$ .

Кроме геометрического подхода при разбиении графов применяется структурный подход. Структурные методы можно разделить на методы обхода графа и спектральные методы.

Алгоритм рекурсивной структурно-уровневой бисекции [5] определяет максимально близкие вершины, вычисляя расстояние между ними. Затем от этих вершин выполняется поиск в ширину. Половина вершин графа помещается в первый подграф, вторая половина — во второй подграф. Алгоритм повторяется для каждого подграфа. Этот алгоритм обладает высоким быстродействием, но получаемое разбиение имеет плохое качество. Поиск в ширину применяется также жадным алгоритмом Фархата [5], но подграфы конструируются, начиная с вершины, выбранной произвольно. Вес каждого подграфа равен  $|V|/p$ , где  $|V|$  — число вершин графа,  $p$  — число процессоров. Похожим алгоритмом является жадный алгоритм растущего графа, предложенный Кариписом и Кумаром. Он конструирует подграфы для бисекции, начиная с произвольного корня. Вершины добавляются в подграф в таком порядке, чтобы минимизировалось число обрезанных ребер [5].

Спектральный метод RSB решает задачу, которую можно сформулировать следующим образом. Необходимо найти вектор  $X$ , где  $X_i = 1$ , если вершина  $i$  должна быть в первом подграфе и  $X_i = -1$  если вершина  $i$  должна быть во втором подграфе,  $\sum_i X_i = 0$ ,  $X^T L X \rightarrow \min$ , где значение  $L_{ij}$  равно весу вершины  $i$ , если  $i=j$ . Если  $i \neq j$  и вершины  $i$  и  $j$  соединены, то  $L_{ij} = -1$ . Если  $i \neq j$  и вершины  $i$  и  $j$  не соединены, то  $L_{ij} = 0$ .

Алгоритм RSB использует собственный вектор матрицы, ассоциированный с вершинами графа. Хотя этот метод имеет высокое качество результата, определение собственного вектора матрицы требует значительных вычислительных затрат.

Для повышения качества разбиения графа могут быть использованы уточняющие методы, например, алгоритм KL/FM или имитации отжига [2].

В работе [6] приведен итеративный динамический метод разбиения графов с учетом балансировки нагрузки при минимизации межпроцессорных связей. Алгоритм KL используется совместно с алгоритмом Ху и Блейка. Предложенный метод применяется системой JOSTLE.

В многоуровневых методах балансировки оригинальный граф подвергается серии преобразований, которые сводят его к более

простому графу. Полученный после преобразования граф, разбивается на подграфы с помощью одного из известных алгоритмов, разбиение этого графа может быть выполнено быстрее, чем оригинального. Затем происходит постепенный возврат к оригинальному графу. При этом может быть использован уточняющий алгоритм, например, алгоритм KL/FM.

Многоуровневый метод RSB (MRSB) преобразует исходный граф, используя понятие максимального независимого множества графа и алгоритм RSB для разбиения графа. Известно несколько похожих методов, которые используют понятие максимального паросочетания графа на этапе его преобразования. Максимальное паросочетание может быть найдено с помощью простого жадного алгоритма.

При многофазной стратегии балансировки нагрузки на каждой фазе применяется метод многоуровневой балансировки на основе алгоритма [6]. Результаты предыдущей фазы учитываются при переходе на новую фазу. С каждой новой фазой качество разбиения графа и балансировки нагрузки улучшается.

В работе [7] предложен генетический алгоритм многоуровневого разбиения графа. Хотя данный алгоритм формирует высококачественное решение задачи, чрезвычайно большое время его работы не дает возможности применить этот алгоритм в типичных задачах. Авторы работы [7] предлагают использовать генетический алгоритм при формировании тестов (benchmarks) для алгоритмов разбиения графов.

Стандартный подход к задаче разбиения графов имеет серьезные недостатки. Задача разбиения графов не вполне адекватна задаче балансировки нагрузки. Величина потока сообщений, передаваемых между процессорами, в действительности не пропорциональна числу разрезанных ребер графа и слабо предсказуема коммуникационными затратами [8].

Основы классификации методов балансировки нагрузки, которые можно применять при распределенном моделировании цифровых устройств, показаны в таблице 1.

### ***Выводы***

Большинство пакетов балансировки нагрузки предназначено для решения статической задачи балансировки. Среди алгоритмов динамической балансировки нагрузки большинство является последовательными. По скорости и качеству, лучшим среди последовательных алгоритмов балансировки является многоуровневый метод Кариписа и Кумера, который использует быстрый жадный алгоритм на этапе разбиения графа и алгоритм KL/FM на этапе уточнения.

Перспективным направлением исследований в области распределенного моделирования цифровых устройств является разработка

эффективных параллельных динамических алгоритмов балансировки нагрузки процессоров.

Таблица 1 — Классификация методов балансировки

<b>Геометрические</b>	<b>Структурные</b>
Алгоритм RCB	Рекурсивная структурно-уровневая бисекция
Алгоритм RIB	Жадные алгоритмы
Алгоритм URB	Алгоритм RSB
<b>Локальные</b>	<b>Уточняющие</b>
Алгоритм Кубенко	Алгоритм KL
Алгоритм Ху и Блейка	Алгоритм KL/FM
Алгоритмы, управляемые запросами	Алгоритм имитации отжига

Таблица 1 — Классификация методов балансировки (окончание)

<b>Многоуровневые</b>	<b>Master/Slave</b>
Алгоритм MRSB	Алгоритм Фрамонто
Алгоритм Кариписа и Кумера	Алгоритм Херинга
Алгоритм Вальшава	Алгоритм инициативного сервера

### *Литература*

1. Отчет по научно-исследовательской работе Г23–2000 “Развитие теории распределенных вычислений в специализированных сетях для автоматизированного проектирования ЦУ в базе ПЛИС”. — Донецк: ДонНТУ, 2001. — 280 с.
2. Hendricson B., Devine K. Dynamic load balancing in computational mechanics. // [www.cs.sandia.gov/~bahendr/load\\_balancing](http://www.cs.sandia.gov/~bahendr/load_balancing).
3. Hering K., Loser J., Markwardt J. dlbSIM — a parallel functional logic simulator allowing dynamic load balancing. // Proc. of DATE '01. IEEE Press. 2001. — P. 471–478.
4. Асад Ахмад. Динамическое выравнивание нагрузки в распределенных вычислительных системах. Автореферат диссертации. — К.: 1996. — 22 с.
5. Chamberlain B.L. Graph partitioning algorithms for distributing workloads of parallel computations. // [www.cs.sandia.gov/~cham/balance](http://www.cs.sandia.gov/~cham/balance).
6. Walshaw C., Cross M. Mesh partitioning: a multilevel balancing and refinement algorithm. // SIAM journal science computing. — 2000. — № 1. — P. 63–80.
7. Soper A.J., Walshaw C., Cross M. A combined evolutionary search and multilevel optimization approach to graph-partitioning. // Journal of global optimization. — 2004. — № 1. — P. 225–241.
8. Hendrickson B. Load balancing fictions, falsehoods and fallacies. // [www.cs.sandia.gov/~bahendr/load\\_balancing/fictions](http://www.cs.sandia.gov/~bahendr/load_balancing/fictions).

## ОЦЕНКА ЭФФЕКТИВНОСТИ КЛАСТЕРНЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ ВЕРОЯТНОСТНЫХ МОДЕЛЕЙ

Т.В. Михайлова

Донецкий национальный технический университет

### Анотація

Пропонуються модифіковані методи аналізу і синтезу багатопроцесорних обчислювальних ресурсів різної топології за допомогою імовірнісних моделей, що дозволяють аналізувати і проектувати більш широкий клас паралельних обчислювальних середовищ

Одной из проблем многопроцессорных вычислительных систем является оценка ее эффективности. В настоящее время широкое распространение получили кластеры различной архитектуры. По совместному использованию дискового пространства их можно классифицировать следующим образом: с совместным использованием дискового пространства и без предоставления доступа к ресурсам [1].

На основании методик [2, 3] можно построить модели кластеров с совместным использованием дискового пространства (рис.1, рис.3) и без предоставления доступа к ресурсам (рис.5).

В каждом из кластеров  $M$  рабочих станций пользователей,  $N_1$  серверов,  $N_2$  дисковых массивов.

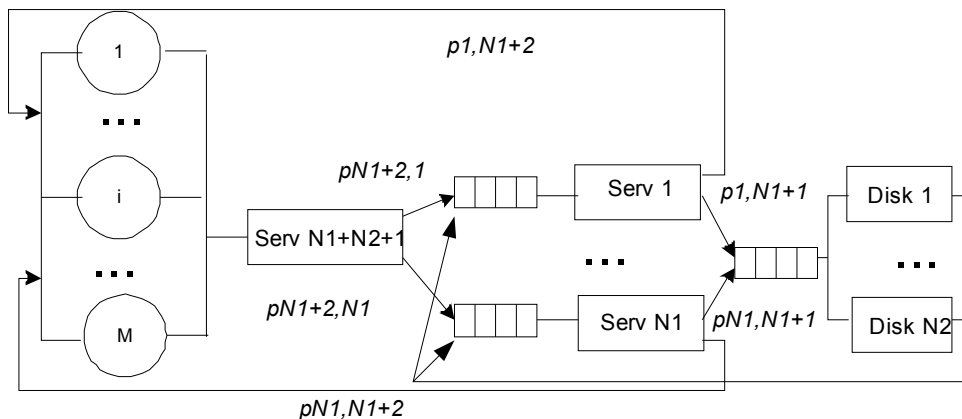


Рисунок 1

Функционирование рассматриваемых систем можно представить замкнутой стохастической сетью, содержащей соответствующие системы массового обслуживания (СМО), в которой циркулирует  $M$  заявок.

Графы передач этих сетей для каждой из моделей приведены на рис.2, рис.4, рис.6, соответственно.

Так как в ВС находится постоянное число задач, то предполагается, что после завершения очередной задачи пользователь приступает к следующей. По графу передач можно определить коэффициенты посещения каждой СМО [3].

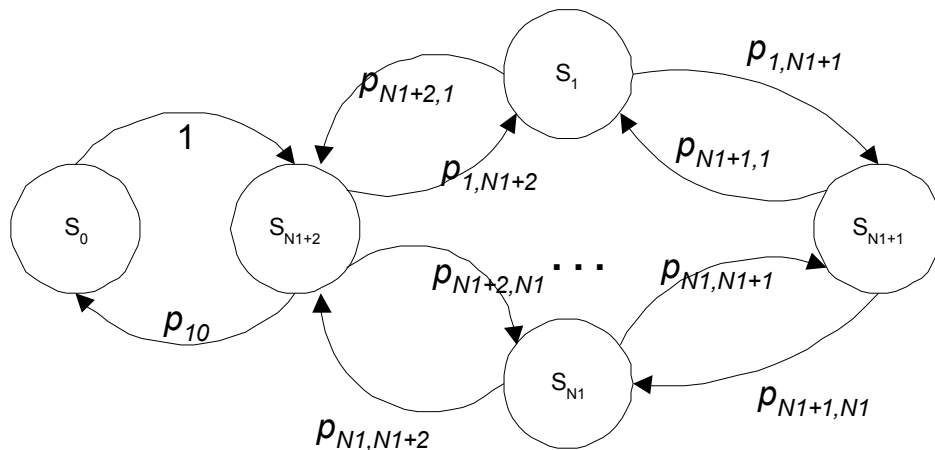


Рисунок 2

За состояние системы принимают распределение заявок по СМО, а по теореме Джексона получают соответствующие состояниям стационарные вероятности.

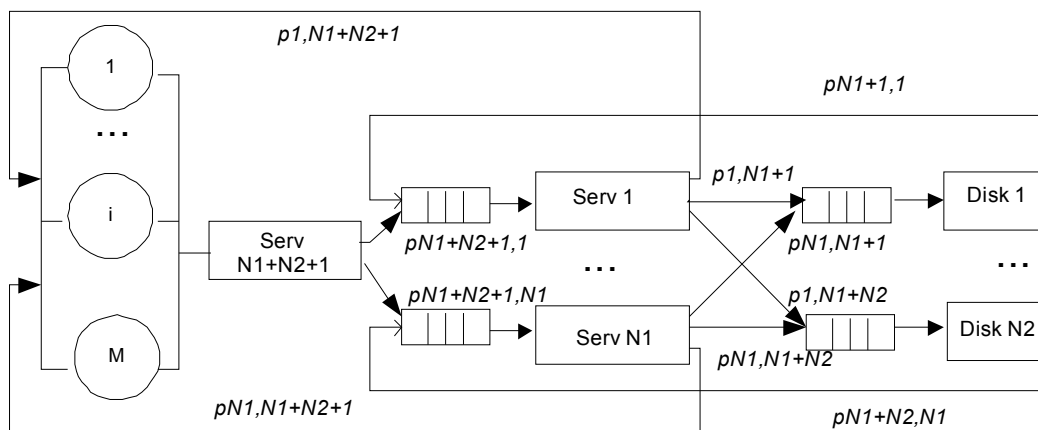


Рисунок 3

Используя стационарные вероятности можно вычислить основные характеристики вычислительной среды: загрузки устройств; среднее число занятых устройств в  $s$ -м узле; среднее число задач, находящихся в  $s$ -м узле; среднее число задач, находящихся в очереди к  $s$ -му узлу; средние времена пребывания и ожидания в  $s$ -м узле; средние времена пребывания и ожидания в системе [2].

Если анализируется конкретный вид кластера для решения определенного класса задач, то с помощью этих характеристик можно определить эффективность вычислительной среды (в зависимости от критерия: равномерная загрузка всех узлов, минимальное время отклика и т.д.) [2].

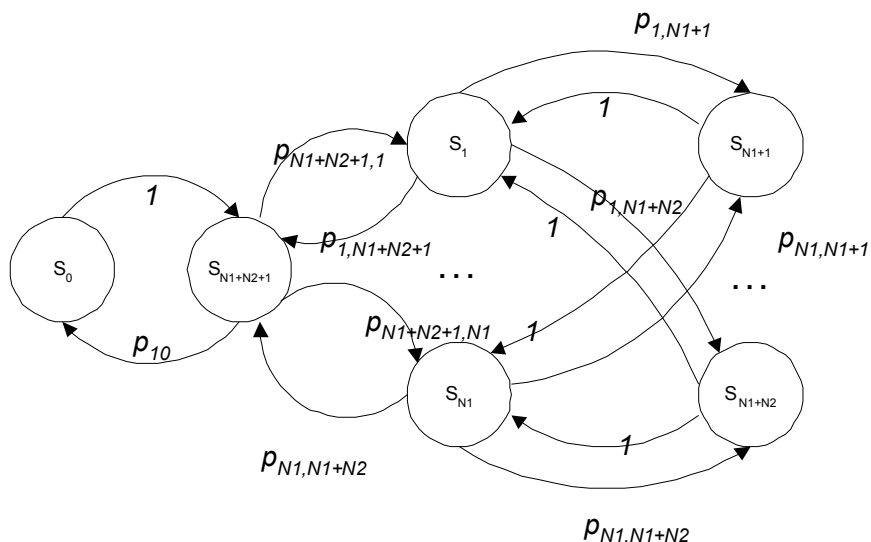


Рисунок 4

Для подбора оптимального коэффициента мультипрограммирования можно использовать методику [4], в которой предлагается критерий сбалансированности, составляющие которого цена простоя оборудования и штраф за задержку выполнения запроса.

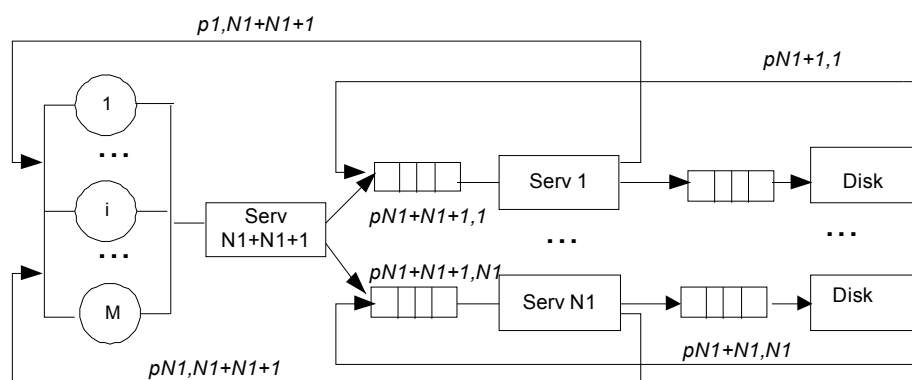


Рисунок 5

Для выбора вида кластера для определенного класса решаемых задач можно сравнить получаемые при моделировании характеристики и выбрать подходящую структуру вычислительной среды. Для иллюстрации вышесказанного пронумеруем рассматриваемые модели: 1, 2, 3, соответствующие структурам, изображенным на рис.1, рис.3, рис.5.

Например, при решении класса задач с равномерным обращением к серверам и равномерном обращении серверов к дискам при сравнении моделей 1 и 2 меньшее время отклика для модели 1 для любой средней длительности обслуживания заявки (рис.7).

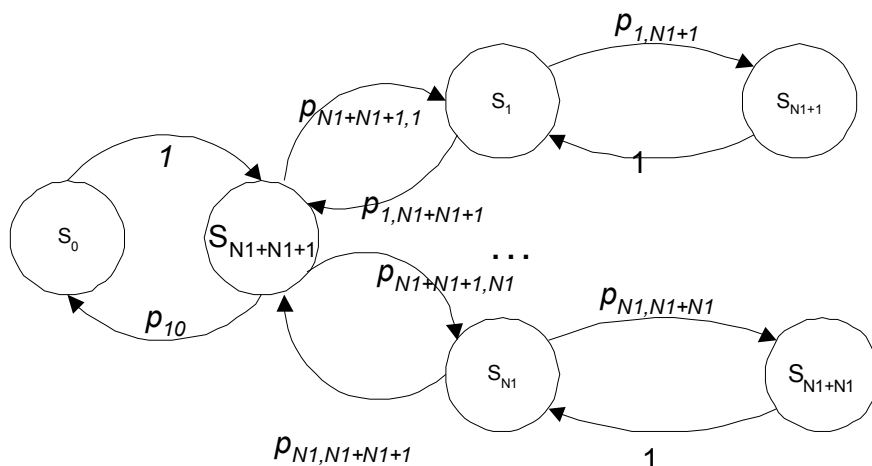


Рисунок 6

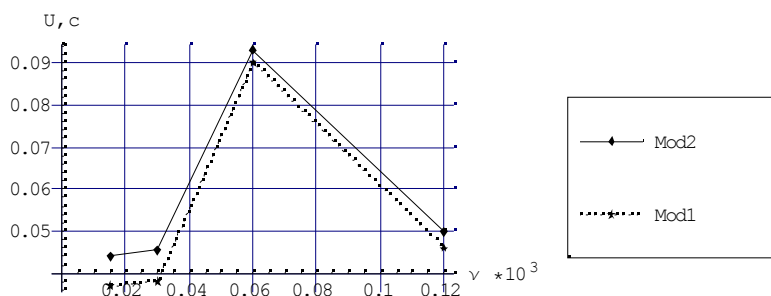


Рисунок 7 — Зависимость времени отклика от средней длительности обслуживания заявки на сервере

А при решении задач с неравномерным обращением к серверам при исследовании моделей 2 и 3 время отклика для модели 3 меньше при средней длительности обслуживания заявки меньшей 0.04, а в противном случае, время отклика меньше для модели 2 (рис.8).

Вероятностные модели можно использовать для анализа эффективности решаемых классов задач [3].

Для оптимизации состава и структуры вычислительных систем можно использовать методы [5], позволяющие определить структуру вычислительной среды минимальной стоимости при заданном времени отклика или, наоборот, с минимальным временем отклика заданной стоимости. Алгоритм с использованием теоремы Джексона имеет комбинаторный порядок, а алгоритмы с использованием



теоремы о среднем [5] — полиномиальный, что позволяет решать задачи, которые вообще не решаются аналитическим методом на современных ЭВМ в течение реального времени.

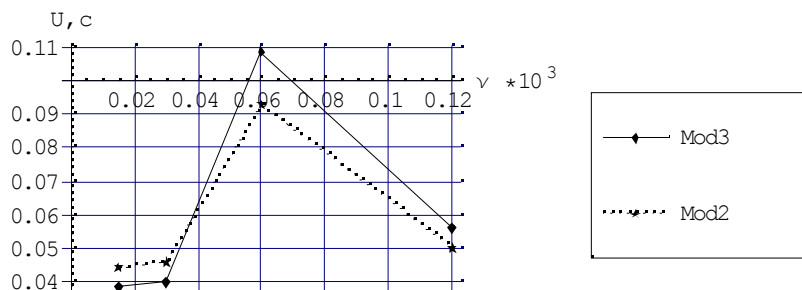


Рисунок 8 — Зависимость времени отклика от средней длительности обслуживания заявки на сервере

Рассмотренные в [5] способы оптимизации состава и структур высокопроизводительных ВС можно использовать для проектирования ВС, причем при большом коэффициенте мультипрограммирования — только численным (градиентным) или с использованием метода средних.

Таким образом, использование вероятностных моделей при проектировании, эксплуатации и оптимизации вычислительных систем позволяет вырабатывать рекомендации по рациональному использованию ресурсов этой вычислительной среды.

### Литература

1. Спортак М., Франк Ч., Паппас Ч. и др. Высокопроизводительные сети. Энциклопедия пользователя. — К.: "ДиаСофт", 1998. — 432 с.
2. Основы теории вычислительных систем/ С.А. Майоров, Г.И. Новиков, Т.И. Алиев и др. — М.: Высшая школа, 1978. — 408 с.
3. Михайлова Т.В. Анализ оценки эффективности кластерных систем с использованием вероятностных моделей //Системний аналіз та інформаційні технології. Тези доповідей учасників Міжнародної науково-практичної конференції студентів, аспірантів та молодих вчених. 1–3 липня 2003р., м. Київ, 83–85 с.
4. Фельдман Л.П., Михайлова Т.В. Оценка эффективности кластерных систем с использованием моделей Маркова. //Известия ТРТУ. Тематический выпуск: Материалы Всероссийской научно-технической конференции с международным участием «Компьютерные технологии в инженерной и управленческой деятельности». — Таганрог: ТРТУ, 2002. — №2 (25). — С. 50–53.
5. Фельдман Л.П., Михайлова Т.В. Способы оптимизации состава и структуры высокопроизводительных вычислительных систем //Научные труды Донецкого государственного технического университета. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ–2001). — Донецк: ДонГТУ. — 2000. — С. 80–85.

## АНАЛІЗ ЕПІДЕМІОЛОГІЧНОГО ПОРОГУ ПОШИРЕННЯ КОМП'ЮТЕРНОГО ВІРУСУ В ІНТЕРМЕРЕЖАХ

Н.О. Возняк

Львівський національний університет ім. І.Франка

### **Анотація**

*Здійснено аналіз моделі поширення комп'ютерного вірусу в інтермережах в околі стаціонарної точки, в якій відсутні інфіковані хости. Одержано умови початку поширення вірусу в зв'язаних підмережах в залежності від коефіцієнтів моделі, а також знайдено аналітичний вираз для базового репродуктивного коефіцієнта.*

### **Вступ**

Довгий час математична епідеміологія була засобом для моделювання і дослідження поширення біологічних вірусів. Навіть сьогодні побудова і аналіз моделей поширення СНІДу чи тяжкого гострого респіраторного синдрому дозволяють виробляти стратегії захисту від цих небезпечних недуг. Тим часом людство вже кілька років атакує інший вид інфекції — комп'ютерний вірус. Витрати чималих коштів на ліквідацію наслідків комп'ютерних інфекцій дають привід для більш детального вивчення цього епідеміологічного феномену. Власне тому останнім часом з'явилося декілька фахових публікацій (переважно закордонних авторів) на дану ематіку [5,6,7].

Комп'ютерна епідеміологія враховує такі основні особливості поширення комп'ютерних вірусів:

- незалежність від географічного розміщення хостів
- швидке інфікування мережі

Будемо розглядати складну комп'ютерну мережу (інтермережу) як сукупність деякої кількості пов'язаних мереж. Мережі, що входять у складену мережу, будемо називати підмережами. Нами було запропоновано [1, 2] модель, яка дозволяє описати поширення комп'ютерного вірусу в поєднаних між собою підмережах. Така модель дає змогу дослідити умови поширення вірусу між підмережами і, що найголовніше, одержати важливі епідеміологічні величини, які дозволяють будувати стратегії запобігання проникнення вірусу в підмережі.

### **Епідеміологічна модель**

Для моделювання процесу поширення комп'ютерного вірусу в інтермережі будемо використовувати наступну нелінійну систему звичайних диференціальних рівнянь [1,2]:

$$\begin{cases} \dot{S}_i(t) = -S_i(t) \sum_{j=1}^K \delta_{ji} I_j(t) \\ \dot{I}_i(t) = S_i(t) \sum_{j=1}^K \delta_{ji} I_j(t) - \mu_i I_i(t) \quad , j = 1, 2, \dots, K \\ \dot{R}_i(t) = \mu_i I_i(t) \\ \delta_{ii} = \beta_i, N_i = S_i(t) + I_i(t) + R_i(t) \end{cases} \quad (1)$$

де  $S_i$  — кількість незахищених хостів в  $i$ -ій підмережі,  $I_i$  — кількість інфікованих хостів в  $i$ -ій підмережі,  $R_i$  — кількість захищених хостів в  $i$ -ій підмережі,  $\mu_i$  — швидкість імунізації хостів в  $i$ -ій підмережі,  $\delta_{ii} = \beta_i$  — швидкість поширення вірусу в  $i$ -ій підмережі, а кожен коефіцієнт  $\delta_{ij}, i \neq j$  задає ймовірність попадання вірусу з підмережі  $i$  в підмережу  $j$ .

Подальші обчислення ми будемо проводити для випадку  $i=2$  (хоча всі результати можуть бути одержані аналогічним чином для будь-якої скінченої кількості підмереж).

$$\begin{cases} \dot{S}_1(t) = -\beta_1 S_1(t) I_1(t) - \delta_{BA} S_1(t) I_2(t), \\ \dot{I}_1(t) = \beta_1 S_1(t) I_1(t) + \delta_{BA} S_1(t) I_2(t) - \mu_1 I_1(t) = 0 \\ \dot{R}_1(t) = \mu_1 I_1(t), \\ \dot{S}_2(t) = -\beta_2 S_2(t) I_2(t) - \delta_{AB} S_2(t) I_1(t), \\ \dot{I}_2(t) = \beta_2 S_2(t) I_2(t) + \delta_{AB} S_2(t) I_1(t) - \mu_2 I_2(t) = 0, \\ \dot{R}_2(t) = \mu_2 I_2(t) \end{cases} \quad (2)$$

### **Аналіз поширення комп'ютерного вірусу в інтермережі**

#### **Епідеміологічний поріг**

При математичному дослідженні кожної епідеміологічної моделі важливо визначити її епідеміологічний поріг. Ця величина вказує на те, за якої умови можливе поширення хвороби при невеликій кількості початково інфікованих хостів.

Для обчислення значення епідеміологічного порогу потрібно дослідити стійкість системи звичайних диференціальних рівнянь (2) в її стаціонарній точці, де відсутні інфіковані хости:

$$(S_i = S_i^0, I_i = 0, R_i = 0) \text{ для } i=1,2 \quad (3)$$

Дослідити стійкість системи (2) в стаціонарній точці (3) можна або записавши відповідну функцію Ляпунова, або визначивши власні значення матриці Якобі, обчисленої в стаціонарній точці (3).

Матриця Якобі для моделі (2) має вигляд:

$$J = \begin{pmatrix} -\beta_1 I_1 - \delta_{BA} I_2 & -\beta_1 S_1 & 0 & 0 & -\delta_{BA} S_1 & 0 \\ \beta_1 I_1 + \delta_{BA} I_2 & \beta_1 S_1 - \mu_1 & 0 & 0 & \delta_{BA} S_1 & 0 \\ 0 & \mu_1 & 0 & 0 & 0 & 0 \\ 0 & -\delta_{AB} S_2 & 0 & -\beta_2 I_2 - \delta_{AB} I_1 & -\beta_2 S_2 & 0 \\ 0 & \delta_{AB} S_2 & 0 & \beta_2 I_2 + \delta_{AB} I_1 & \beta_2 S_2 - \mu_2 & 0 \\ 0 & 0 & 0 & 0 & \mu_2 & 0 \end{pmatrix} \quad (4)$$

В околі стаціонарної точки  $(S_i = S_i^0, I_i = 0, R_i = 0), i = 1,2$  матриця Якобі має вигляд:

$$J_b = \begin{pmatrix} 0 & -\beta_1 S_1^0 & 0 & 0 & -\delta_{BA} S_1^0 & 0 \\ 0 & \beta_1 S_1^0 - \mu_1 & 0 & 0 & \delta_{BA} S_1^0 & 0 \\ 0 & \mu_1 & 0 & 0 & 0 & 0 \\ 0 & -\delta_{AB} S_2^0 & 0 & 0 & -\beta_2 S_2^0 & 0 \\ 0 & \delta_{AB} S_2^0 & 0 & 0 & \beta_2 S_2^0 - \mu_2 & 0 \\ 0 & 0 & 0 & 0 & \mu_2 & 0 \end{pmatrix} \quad (5)$$

Обчисливши власне значення матриці (5) з найбільшою дійсною частиною

$$\text{Re } \lambda_{\max} = \frac{1}{2} \left( a_1 + a_2 + \sqrt{(a_1 - a_2)^2 + 4a_3 a_4} \right), \quad (6)$$

де

$$\begin{aligned} a_1 &= \beta_1 S_1^0 - \mu_1, \\ a_2 &= \beta_2 S_2^0 - \mu_2, \\ a_3 &= \delta_{BA} S_1^0, \\ a_4 &= \delta_{AB} S_2^0 \end{aligned} \quad (7)$$

ми визначимо умову епідеміологічного порогу, оскільки стійкість матриці (5) визначається умовою  $\text{Re } \lambda_{\max} < 0$ :

$$a_1 + a_2 + \sqrt{(a_1 - a_2)^2 + 4a_3 a_4} < 0 \quad (8)$$

Остання умова і визначає епідеміологічний поріг.

### *Базовий репродуктивний коефіцієнт*

Епідеміологічний поріг дозволяє визначити так званий базовий репродуктивний коефіцієнт  $R_0$  (кількість захворювань до яких призведе один інфікований хост в цілком незахищеній мережі). Ми можемо визначити його з умови (8) [3]:

$$R_0 = \frac{\beta_1 S_1^0 + \beta_2 S_2^0 + \sqrt{(\beta_1 S_1^0 - \mu_1 - \beta_2 S_2^0 + \mu_2)^2 + 4\delta_{AB}\delta_{BA}S_1^0 S_2^0}}{\mu_1 + \mu_2} \quad (9)$$

Якщо базовий репродуктивний коефіцієнт  $R_0 < 1$ , то епідемія не розпочнеться. Інакше, при наборі параметрів моделі, що задовольняють умову  $R_0 > 1$ , існує тенденція до збільшення кількості інфікованих хостів.

### *Аналіз епідеміологічного порогу*

Аналізуючи нерівність (8) бачимо, якщо одна з підмереж є „непропускною” для вірусів із другої підмережі  $\delta_{AB} = 0 \vee \delta_{BA} = 0$ , то зміна коефіцієнта проникності віруса для іншої підмережі ніяк не впливає на величину епідеміологічного порогу. Цей результат є дещо несподіваним, оскільки ми очікували, що коефіцієнти проникнення вірусу між підмережами повинні мати незалежний вплив на порогове значення. Ми пояснюємо одержаний результат тим, що в околі стаціонарної точки, в якій відсутні інфіковані хости, малі збурення хвороби не мають впливу на поширення віруса в інші підмережі.

Інший важливий результат одержимо поклавши

$$\beta = \beta_1 = \beta_2, \mu = \mu_1 = \mu_2, \delta_{AB} = \delta_{BA}, S_0 = S_1^0 = S_2^0$$

(тобто підмережі зв'язані між собою і містять однакову кількість хостів, а коефіцієнти поширення вірусу, імунізації хостів і проникнення вірусу між підмережами співпадають), тоді умову епідеміологічного порогу (8) можна переписати:

$$\frac{\beta_0 S_0 + \delta S_0}{\mu} < 1 \Rightarrow S_0 < \frac{\mu}{\beta + \delta} \quad (10)$$

Зауважимо, що для  $\delta = 0$  одержимо  $S_0 < \frac{\mu}{\beta}$  — відому [4] умову епідеміологічного порогу для класичної SIR моделі.

Нерівність (10) показує, чим більшим є коефіцієнт проникності вірусу між підмережами, тим меншим є епідеміологічний поріг системи загалом.

### **Висновки**

В даній роботі для запропонованої раніше моделі поширення комп'ютерного вірусу в інтермережі у випадку двох пов'язаних підмереж визначено епідеміологічний поріг (8) та базовий репродуктивний коефіцієнт (9) як функції від швидкості поширення вірусу, швидкості імунізації та коефіцієнтів ймовірності проникнення вірусу в підмережу.

Величину епідеміологічного порогу необхідно враховувати при побудові сучасних комп'ютерних мереж, оскільки таким чином можна суттєво знизити ймовірність масово враження хостів в підмережі. Розробляючи стратегію захисту інтермережі слід брати до уваги не лише захищеність окремо взятих підмереж, а й вплив взаємозв'язків між ними на стійкість до вірусу системи загалом.

### **Література**

1. Возняк Н.О., Щербатий М.В., Поширення вірусів в інформаційних системах.// Сучасні проблеми прикладної математики та інформатики: Тези доп. Десятої Всеукраїнської наукової конференції (Львів, 23–25 вересня 2003 р.). — Львів, 2003.— С. 37.
2. Возняк Н.О., Щербатий М.В., Врахування зв'язності мережі при моделюванні поширення вірусів в інформаційних системах, Механіка середовища, методи комп'ютерних наук та моделювання, 2004, т.2, с.97–107.
3. Castillo–Chavez C., Z. Feng and W. Huang. On the computation  $R_0$  and its role on global stability, In: *Mathematical Approaches for Emerging and Reemerging Infectious Diseases: An Introduction*, IMA Volume 125, 229–250, Springer–Verlag, Berlin–Heidelberg–New York. Edited by Carlos Castillo–Chavez with Pauline van den Driessche, Denise Kirschner and Abdul–Aziz Yakubu.
4. Kermack W.O., McKendrick A.G., A contribution to the mathematical theory of epidemics // *Proceedings of the Royal Society of London Series A.*, 1927, pp.55–83.
5. Leveille J., *Epidemic Spreading in Technological Networks*, 2002, HP, Technical Report: HPL–2002–287.
6. Moore D., Shannon C., Voelker G. M., Savage S., *Internet Quarantine: Requirements for Containing Self–Propagating Code*, San Diego Supercomputer Center, University of California, San Diego, INFOCOM–2003
7. Weissman G., *Beubonica: Mitigating Duration and Peak Intensity with a New Model of Computer Virus Epidemics*, Dartmouth Undergraduate Journal of *Science*, Spring 2001 Volume III, No. 2.

## МОДЕЛИРОВАНИЕ РАСПАРАЛЛЕЛИВАНИЯ ЗАДАЧИ С ПОМОЩЬЮ ГРАФА ПОТОКОВ ДАННЫХ

В.И. Костин, М.В. Краснокутская  
Каф. ПМиИ, ДонНТУ

### **Анотація**

*Описується представлення задачі, що розпаралелюється, за допомогою графа потоків даних. Врівноваження обчислювального завантаження між процесорами можна відобразити на задачі розбиття графа. Представлено два алгоритми рішення цієї задачі і описується деякі особливості їх застосування до графів з великою кількістю вершин.*

### **Введение**

Параллельные вычисления позволяют значительно повысить эффективность и скорость обработки информации при решении современных задач. Такие задачи возникают в электромеханике, при оптимизации сложных систем, при прогнозировании погоды, моделировании разнообразных технических и природных процессов.

Одна из основных проблем, которая возникает в каждом параллельном вычислении, это распределение обработки данных между процессорами. Решением может быть использование математической модели, в основе которой лежит граф потоков данных (ГПД). Программа представляется набором вычислительных узлов-подзадач, которые имеют фиксированное количество информационных входов и выходов. Каждая подзадача выполняется на отдельном процессоре. Узлы-подзадачи — вершины графа потоков данных, а информационные потоки между ними — ребра графа. Оптимальное распределение обработки данных между процессорами минимизирует время выполнения всех вычислений. Задача распределения обработки данных на процессоры, сводится к задаче разбиения графа. Необходимо разбить граф потоков данных так, чтобы количество связей между подграфами было минимальным.

В настоящее время известно большое количество алгоритмов, решающих проблему разбиения графа. Но так как разбиение графа является NP-трудной задачей, то все известные алгоритмы дают приближение к оптимальному решению. Среди наиболее известных можно выделить алгоритм Kernighan-Lin/Fiduccia-Mattheyses (KL/FM), алгоритм спектральной бисекции.

Оба алгоритма улучшают начальное (возможно случайное разбиение графа), поэтому при нахождении наилучшего разбиения эффективным является последовательное применения обоих алгоритмов. Рассмотрим эти алгоритмы.

### *Алгоритмы разбиения графа*

Алгоритм Kernighan-Lin (KL) [1] используется для улучшения начального разбиения графа, путем обмена вершинами из начальных наборов с целью уменьшения грань-соединения. KL алгоритм использует понятие веса. Это величина, показывающая на сколько изменится грань-соединение при перемещении вершины из одного подмножества в другое. Вес рассчитывается для каждой вершины как количество соединений вершины с другим подмножеством, минус количество соединений с подмножеством, в котором вершина находится. Пока есть вершины с положительным весом, алгоритм меняет их местами с вершинами из другого подмножества.

Недостатком данного подхода является то, что перемещения вершин могут привести к локальному минимуму. Для преодоления локальных минимумов, алгоритм перемещает и вершины с отрицательным весом. Причем, если вершина уже была перемещена, то она не участвует в дальнейшем просмотре.

Суть алгоритма заключается в том, что необходимо найти два множества вершин одинаковой мощности  $X$  и  $Y$ , из  $A$  и  $B$  соответственно.  $X$  и  $Y$  выбираются таким образом, что перемещение  $X$  в  $B$  и  $Y$  в  $A$  даст максимально возможное уменьшение грань-соединения. Такой обмен выполняется до тех пор, пока улучшается необходимый критерий.

KL алгоритм состоит из просмотров вершин. При каждой итерации алгоритм отыскивает пару вершин по одной из каждого набора и производит обмен. Пара вершин выбирается таким образом, чтобы максимально улучшить качество разбиения. Если пара была перемещена, то она уже не рассматривается в этой итерации. Когда все вершины были перемещены, итерация заканчивается.

Fiduccia and Mattheyses [2] предложили модификацию KL алгоритма. FM алгоритм использует KL алгоритм в качестве базового, но за один его шаг можно переместить только одну вершин. На каждом шаге алгоритма для каждой вершины пересчитывается вес. После того как вес вершины пересчитан, для каждого подмножества строится очередь вершин. Вершины



помещаются в очередь в порядке убывания веса. Просмотр вершин происходит в том порядке, в котором вершины расположены в очереди. Вершина перемещается только в том случае, если условия балансировки не нарушаются. При перемещении вершины в другой набор она удаляется из очереди. При пересчете весов всех вершин очереди строятся заново.

KL/FM алгоритм позволяет избежать некоторых типов локальных минимумов, так как позволяет временно увеличивать грань-соединение.

Хотя алгоритм KL/FM позволяет избежать локальных минимумов, возможности его ограничены. Качество разбиения сильно зависит от начального разбиения.

В алгоритме спектральной бисекции [3] в соответствии каждой вершине графа ставится переменная  $x$ , равная  $+1$  или  $-1$ , таким образом, что сумма всех  $x$ -ов равна  $0$ . Первое условие подразумевает разбиение на два различных набора. Второе требует чтобы наборы были равного размера, учитывая четность исходного количества. Назовем вектор  $x$  вектор-индикатор, т.к. он показывает принадлежность каждой вершины к набору.

Затем определяется функция от  $x$ , определяющая число граней, пересекающихся между наборами. Теперь, когда имеется функция для минимизации, преобразуем ее к матричной форме, используя матрицу Лапласа. Так как разбиение графа — NP-трудная задача, необходимо ослабить ограничения дискретности на  $x$  и сформулировать новую непрерывную задачу:

$$\text{Минимизировать : } f(x) = \frac{1}{4} x^t L x$$

$$\text{Ограничения : } x^t 1 = 0, \quad x^t x = n$$

где  $1$  — это  $n$  мерный вектор  $(1, 1, 1, 1, \dots)^t$ ,

$n$  — число вершин графа,

$L$  — матрица Лапласа

$$L_{ij} = \begin{cases} -1, & \text{если вершины } i \text{ и } j \text{ соединены ребром} \\ d_i, & \text{если } i = j \\ 0, & \text{иначе} \end{cases}$$

$d_i$  — это число граней, смежных  $i$ -ой вершине.

Эта непрерывная задача - только приближение к дискретной, и значения, определяющие ее решение, должны быть отображены назад к  $\pm 1$  в соответствии с некоторой соответствующей схемой. Идеально, когда решение близко к  $\pm 1$ .

Если  $U_1, U_2 \dots$  — нормализованные собственные векторы  $L$  с соответствующими собственными значениями  $\lambda_1 \leq \lambda_2 \leq \lambda_3 \dots$ , то матрица  $L$  имеет следующие свойства:

(I)  $L$  — симметрична.

(II)  $U_i$  — попарно ортогональны.

(III)  $U^T U = I$ ,  $\lambda_1 = 0$

(IV) Если граф замкнутый, то только  $\lambda_1$  принимает нулевое значение.

Затем выразим  $X$  в терминах собственных векторов  $L$ :  $x = \sum \alpha_i U_i$ , где  $\alpha_i$  — вещественные константы, такие, что  $\sum (\alpha_i)^2 = n$ . Свойство (II) гарантирует, что это всегда возможно. Заменой для  $x$  мы получаем функцию, для минимизации, зависящую от собственного значения матрицы Лапласа  $\lambda_2$ .

$$f(x) = 0.25(\alpha_2^2 \lambda_2 + \alpha_3^2 \lambda_3 + \dots + \alpha_n^2 \lambda_n)$$

начиная с  $\lambda_1 = 0$ .

Очевидно

$$(\alpha_2^2 + \alpha_3^2 + \dots + \alpha_n^2) \lambda_2 \leq (\alpha_2^2 \lambda_2 + \alpha_3^2 \lambda_3 + \dots + \alpha_n^2 \lambda_n)$$

учитывая упорядоченность собственных величин  $f(x) \geq n \lambda_2 / 4$ .

Мы можем минимизировать  $f(x) = n \lambda_2 / 4$ , выбирая  $x = \sqrt{n} U_2$ .

Полученный вектор  $x$  — решение непрерывной задачи. Остается решить задачу отображения вектора  $x$  к дискретному разделению. Для этого находится медиана значений  $x_i$ , и затем отображаются вершины выше значения медианы в один набор, ниже в другой. Если несколько вершин имеют значение медианы, то они распределяются, не нарушая равновесия. Это решение — самая близкая дискретная точка к непрерывному оптимуму.

### ***Особенности применения алгоритмов к графам большой размерности***

При программной реализации любого из этих алгоритмов встает задача выбора типа данных для представления информации о графе.

Задание графов с помощью матриц удобно для алгоритмов, использующие матричные вычисления (например, алгоритм спектральной бисекции). Однако, при обработке графа большой размерности ( $n=1000, 10000$ ), матрицы занимают слишком много памяти.

Следует учесть, что матрицы графов потоков данных довольно разрежены, т. е. матрицы содержат много нулей. Исходя из этого граф можно представлять списком ребер, структурой смежности.

При описании графа списком его ребер каждое ребро представляется парой инцидентных ему вершин. Это представление можно реализовать двумя массивами. Каждый элемент в массиве есть метка вершины, а  $i$ -е ребро графа выходит из вершины —  $i$ -ого элемента первого массива, и входит в вершину, задаваемую  $i$ -м элементом второго массива.

Структура смежности состоит из списков вершин графа, смежных с вершиной графа. Такие списки составляются для каждой вершины графа. Структуры смежности могут быть удобно реализованы массивом из  $n$  (число вершин в графе) линейно связанных списков или массивов динамической длины.

Такие способы представления графа позволяют значительно сократить объем памяти, необходимый для хранения графа. С другой стороны в этом случае усложняется доступ к вершинам графа, а, следовательно, и сам алгоритм.

### ***Выводы***

Использование списка ребер эффективно при разреженности матрицы смежности графа более 50%. Использование структуры смежности эффективно при любой разреженности матрицы смежности графа.

По этим алгоритмам разрабатывается программное приложение на Visual C++. Планируется тестирование алгоритмов на эффективность на графах различной размерности и с различной степенью разреженности матрицы смежности.

### ***Литература***

1. B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. The Bell System Technical Journal, 1970.
2. C. M. Fiduccia and R. M. Mattheyses. A linear time heuristic for improving network partitions. In In Proc. 19th IEEE Design Automation Conference, 1982.
3. B. Hendricson and R. Leland. Multidimensional spectral load balancing. Sandia National Laboratories, Albuquerque, NM, 1993.

## **СПОСОБЫ И СРЕДСТВА МОДЕЛИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ С ЦЕЛЬЮ ОБЕСПЕЧЕНИЯ ЭФФЕКТИВНОСТИ ФУНКЦИОНИРОВАНИЯ WEB-СЕРВИСОВ**

А.Я. Аноприенко, С.В. Рычка, Хасан Аль Абабнех  
Донецкий национальный технический университет

### ***Анотація***

*Розглядається комплексний підхід до моделювання складних обчислювальних мереж з використанням різних засобів моделювання.*

Современные вычислительные сети являются ярким примером сложных динамических систем, сложность которых определяется как сложностью их структуры, так и сложностью поведения. В связи с этим как сугубо теоретический их анализ, так и эмпирические наблюдения не дают полного представления о происходящих в них процессах при необходимости их исследования в целях повышения производительности и/или надежности. Наиболее эффективным инструментом в этом случае является имитационное моделирование.

В ДонНТУ исследования в области моделирования современной вычислительной инфраструктуры начались с конца 90-х годов, что нашло отражение в работах [1–3]. Объективно это было связано с нарастанием процессов усложнения корпоративной сети, когда количество взаимосвязанных через сеть компьютеров во второй половине 90-х годов стало исчисляться сотнями, а в начале текущего десятилетия преодолело тысячный рубеж (рис. 1).

При этом следует отметить, что динамика усложнения корпоративных сетей фактически явилась отражением процессов роста суммарного количества компьютеров, подключенных к Интернет в мире (рис. 2).

Непосредственных повода к началу углубленных исследований в области имитационного моделирования компьютерных сетей в ДонНТУ можно выделить два:

- необходимость повышения эффективности эксплуатации корпоративной сети [2];
- необходимость разработки высокопроизводительных информационно-моделирующих сред для исследования, разработки и сопровождения сложных динамических систем [4, 5].

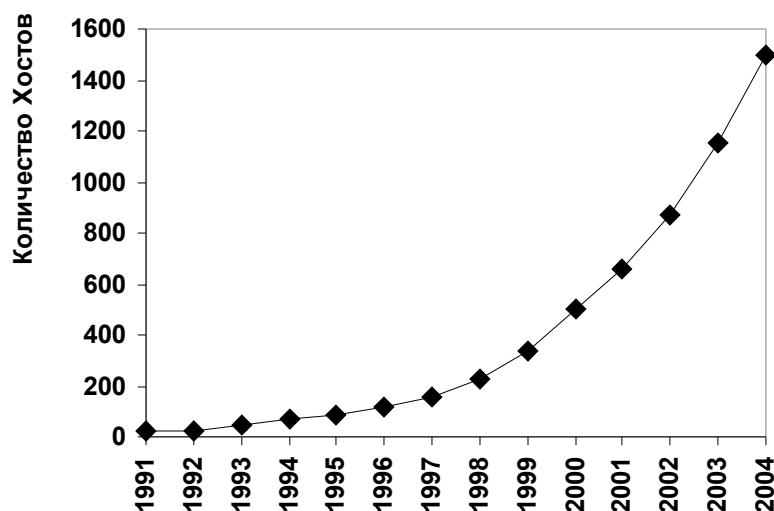


Рисунок 1 — Рост количества компьютеров в корпоративной сети ДонНТУ

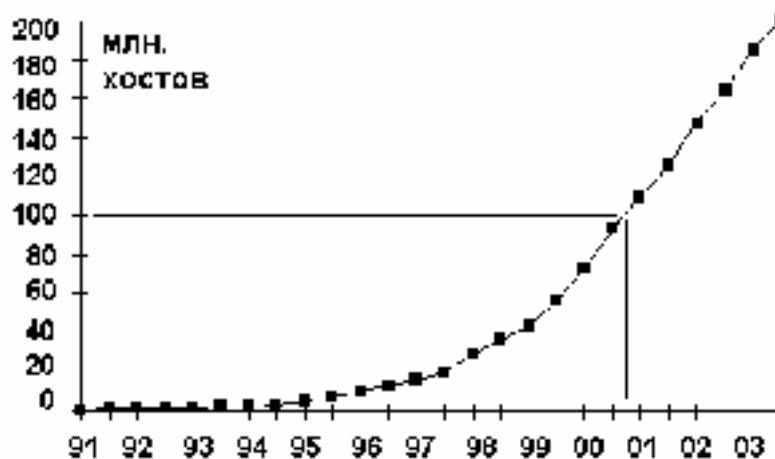


Рисунок 2 — Рост суммарного количества компьютеров, подключенных к Интернет (хосты — компьютеры, имеющие собственный IP-адрес)

Опыт проведения работ в данном направлении показал, что наиболее продуктивным является многоуровневый подход к моделированию с комплексным использованием различного инструментария [6].

В частности, для укрупненного моделирования корпоративной сети в целом наиболее целесообразным представляется использование таких специализированных систем как NetCracker, имеющих, как правило, обширные библиотеки с описанием конкретных сетевых и вычислительных устройств [7]. В ДонНТУ к настоящему времени разработана многоуровневая модель такого рода, охватывающая

практически все основные сегменты корпоративной сети. Такая модель позволяет не только детально документировать текущее состояние сети и различные варианты ее модификации, но и выявлять критичные по производительности и нагрузке фрагменты [2]. Однако для детального исследования такого рода инструментарий в большинстве случаев непригоден.

Одним из возможных вариантов в этом случае является использование таких универсальных средств моделирования как Matlab/Simulink [8]. Преимущество такого подхода заключается в возможности создания моделей с самой различной специализацией и степенью детализации. Главным недостатком при этом заключается в высокой трудоемкости создания таких моделей и необходимости весьма существенных вычислительных ресурсов при работе с детальными моделями сетей уже средней степени сложности.

Третьим подходом, наиболее целесообразным при моделировании вычислительных сетей с целью обеспечения эффективности функционирования в первую очередь web-сервисов является использование комплекса взаимосвязанных расчетных таблиц в среде Excel. Такой подход, детально рассмотренный в работе [9], позволяет получить целый комплекс необходимых оценок при использовании относительно небольших вычислительных ресурсов. В частности пример, представленный в таблице 1, показывает насколько существенно на пропускную способность протокола TCP, а, следовательно, и на эффективность функционирования работающего через него web-сервиса влияют такие параметры как значения RTT и максимального размера окна получателя.

Таблица 1 — Модель пропускной способности протокола TCP  
Входные параметры

Предельная (номинальная) пропускная способность сети (Кбит/с)	Вариант 1	Вариант 2
	100 000	100 000
Период передачи-подтверждения приема — Round Trip Time (RTT), с	0,05	0,01
Предельная (номинальная) пропускная способность сети (Кбайт/с)	12 500	12 500
Вероятность потери пакетов p	0,001	0,001
TCP таймаут, с	0,1	0,1
Максимальный размер сегмента MSS, байт	1 460	1 460
Максимальный размер окна получателя $W_m$ , сегментов	20	100
Расчетные значения:		
<b>Максимальный размер окна получателя <math>W_m</math>, байт</b>	<b>29 200</b>	<b>146 000</b>
<b>Реальная пропускная способность TCP (Кбайт/с)</b>	<b>515</b>	<b>3 808</b>
<b>Реальная пропускная способность TCP (Кбит/с)</b>	<b>4 117</b>	<b>30 463</b>

### *Литература*

1. Аноприенко А.Я., Джон С.Н. Задачи, методы и средства моделирования сетевой инфраструктуры // Научные труды Донецкого государственного технического университета. Выпуск 29. Серия "Проблемы моделирования и автоматизации проектирования динамических систем" — Севастополь: «Вебер». — 2001. — С. 312–319.
2. Аноприенко А.Я., Джон С.Н., Рычка С.В. Моделирование университетской сетевой инфраструктуры / Вісник Кременчуцького державного політехнічного університету. Наукові праці КДПУ. Випуск 2 (11). — Кременчук, КДПУ, 2001. — С. 306–308.
3. Аноприенко А.Я., Джон С.Н., Рычка С.В. Особенности моделирования и оценки эффективности работы сетевой инфраструктуры / Наукові праці Донецького державного технічного університету. Серія “Обчислювальна техніка та автоматизація”. Випуск 38. — Донецьк, ДонДТУ, 2002. — С. 205–210.
4. Аноприенко А.Я., Святный В.А. Высокопроизводительные информационно-моделирующие среды для исследования, разработки и сопровождения сложных динамических систем // Научные труды Донецкого государственного технического университета. Выпуск 29. Серия "Проблемы моделирования и автоматизации проектирования динамических систем" — Севастополь: «Вебер». — 2001. — С. 346–367.
5. Аноприенко А. Я., Потапенко В.А. WEB-ориентированная среда для интеграции моделирующих, вычислительных и информационных сервисов // Научные труды Донецкого национального технического университета. Выпуск 70. Серия: «Информатика, кибернетика и вычислительная техника» (ИКВТ–2002): — Донецк: ДонНТУ, 2003. — С. 61–70.
6. Джон С. Н. Повышение эффективности обмена данными в информационно-вычислительных сетях на базе стека протоколов TCP/IP: Автореф. дис. к-та техн. наук: 05.13.13 / Донецкий национальный технический университет. — Донецк, 2005. — 19 с.
7. Пономаренко Л.А., Щелкунов В.И., Скляр А.Я. Инструментальные средства проектирования, имитационного моделирования и анализа компьютерных сетей. — Киев: Наукова думка, 2002. — 508 с.
8. Аноприенко А.Я., Джон С.Н., Нури А. Ж. Многоуровневое моделирование сетей на базе стека протоколов TCP/IP в среде Matlab/Simulink // Научные труды Донецкого национального технического университета. Выпуск 39. Серия: Информатика, кибернетика и вычислительная техника (ИКВТ–2002): — Донецк: ДонНТУ, 2002. — С. 271–279.
9. Менаске Д., Алмейда В. Производительность Web-служб. Анализ, оценка и планирование: Пер. с англ. — СПб: ООО «ДиаСофтЮП», 2003. — 480 с.

## **МОДЕЛЬ ОЦЕНКИ ЭФФЕКТИВНОСТИ РИСК-МЕНЕДЖМЕНТА В INTERNET-КОМПАНИИ**

Т.А. Мамедова, В.И. Костин  
Каф. ПМиИ, ДонНТУ

### ***Анотація***

*Погроза активам компанії змушує керівництво звертатися до методів керування ризиками з метою зниження втрат компанії внаслідок їх настання. В електронному бізнесі рівень ризикових ситуацій набагато перевищує відсоток ризику в традиційному. Розроблені показники результативності ризик-менеджменту дозволяють оцінити ефективність керування ризиками.*

### ***Введение***

Практическая угроза различных рисков для компании реализуется посредством возможных потерь или убытков. Именно угроза материальным или нематериальным активам компании вынуждает руководство обращаться к методам управления рисками для снижения потерь компании вследствие их наступления. Если говорить об уровне рискованных ситуаций в электронном бизнесе, то он намного превышает процент риска в традиционном, так как является высоко динамичной и довольно нестабильной средой, что связано, прежде всего, с интенсивностью Internet-пространства.

Однако сам по себе риск-менеджмент существовать не может, необходимо иметь четкую систему его оценки. В условиях рыночных отношений проблема оценки эффективности управления рисками финансово-хозяйственной деятельности предприятий приобретает самостоятельное теоретическое и прикладное значение как важная составная часть теории и практики риск-менеджмента.

Для оценки эффективности использования отдельных инструментов риск-менеджмента в Internet-компаниях и эффективности затрат на управление рисками в целом автором разработаны 4 показателя результативности КРІ. Ключевые показатели результативности (Key Performance Indicators — КРІ) [1] являются, по сути, измерителями достижимости целей риск-менеджмента, а также показателями его эффективности.

В процессе принятия решения по осуществлению тех или иных мероприятий по выявлению и управлению рисками необходимо, в первую очередь, обеспечить выполнение такого условия: чтобы как можно большее число рисков было выявлено и изучено перед их непосредственным наступлением.



$$KPI_1 = \frac{n_2(\text{inf}_1, \text{inf}_2, \dots, \text{inf}_n)}{n_1(\text{inf}_1, \text{inf}_2, \dots, \text{inf}_n) + n_2(\text{inf}_1, \text{inf}_2, \dots, \text{inf}_n)} \rightarrow \min, \quad (1)$$

где  $n_1(\text{inf}_1, \text{inf}_2, \dots, \text{inf}_n)$  — функция, показывающая число известных рисков, за счет которых произошло снижение экономической эффективности;

$n_2(\text{inf}_1, \text{inf}_2, \dots, \text{inf}_n)$  — функция, показывающая число неизвестных рисков, за счет которых произошло снижение экономической эффективности;

Аргументы функций показывают насколько достоверно, объективно, полно и т. д. выполнены расчеты и оценены факторы, которые привели к снижению экономической эффективности Internet-компаний.

Достоверная оценка означает меру приближенности информации к первоисточнику или точность передачи информации.

Объективность информации — это мера отражения информацией реальности.

Полнота проведенного анализа означает отражение исчерпывающего характера соответствия полученных сведений целям исследования.

Кроме того, используемыми аргументами могут быть:

- порядок информации — количество передаточных звеньев между первоисточником и конечным пользователем;
- релевантность — степень приближения информации к существу вопроса или степень соответствия информации поставленной задаче;
- актуальность информации (значимость) — важность информации для оценки риска [2].

Для анализа рассчитанного показателя  $KPI_1$  на базе имеющейся эмпирической шкалы риска [2] разработана новая эмпирическая шкала, позволяющая сделать выводы о том, как сработал риск-менеджер относительно выявления рисков. Разработанная эмпирическая шкала приведена в таблице 1.

Далее перейдем к оценке доходности Internet-компаний.

$$KPI_2 = R^i - ERR^{i-1} \geq 0, \quad i = 1 \dots n, \quad (2)$$

где  $R^i$  — общая рентабельность компании в текущем периоде, которая рассчитывается таким образом:

$$R = \frac{\Pi}{K},$$

где  $\Pi$  — прибыль компании до налогообложения;

$K$  — капитал компании;

$ERR^{i-1}$  — ожидаемая доходность, рассчитанная в предшествующем периоде следующим образом:

$$ERR = \sum_{j=1}^n IRR_j \cdot p_j,$$

где  $IRR_j$  — внутренняя ставка доходности  $j$ -го события;  
 $p_j$  — вероятность  $j$ -го события.

Таблица 1 — Эмпирическая шкала показателя  $KPI_1$

№ п/п	Величина показателя $KPI_1$	Наименование градаций показателя $KPI_1$	Характеристика
1	0,0 – 0,15	Минимальный	Информация очень высокого качества, ситуация оценена адекватно. Группа управления рисками сработала отлично. Число неизвестных наступивших рисков — минимально.
2	0,15 – 0,3	Малый	Информация высокого качества, ситуация оценена достаточно адекватно. Группа управления рисками сработала хорошо. Число неизвестных наступивших рисков — допустимо.
3	0,3 – 0,5	Средний	Информация среднего качества, ситуация оценена недостаточно адекватно. Группа управления рисками сработала удовлетворительно. Число неиз. наступ. рисков — выше допустимого.
4	0,5 – 0,8	Высокий	Информация низкого качества, ситуация оценена неадекватно. Группа управления рисками сработала неудовлетворительно. Число неизвестных наступивших рисков — выше допустимого.
5	0,8 – 1,0	Максимальный	Информация отсутствует, ситуация оценена крайне неадекватно. Группа управления рисками не справилась с работой. Число неизвестных наступивших рисков — максимально.

То есть, смысл показателя  $KPI_2$  заключается в том, что разница между текущей доходностью и ожидаемой (прогнозируемой в предшествующем временном периоде) не должна быть отрицательной.

Теперь, необходимо оценить, обоснованно ли применена методика управления каждым риском. Может, стоило вообще принять риск и, повинувшись судьбе, ждать его наступления.

$$KPI_3 = C_{\text{упр.р.}i} - V_i < 0, \quad i = 1 \dots n, \quad (3)$$

где  $C_{\text{упр.р.}i}$  — стоимость управления  $i$ -м риском;

$V_i$  — величина возможного ущерба от  $i$ -го риска.

Рассчитывая стоимость управления риском, необходимо учитывать не только стоимость конкретного мероприятия в денежной и иной форме, но и проиндексировать привлекаемые ресурсы для управления риском на стоимость альтернативного размещения этих ресурсов:

$$C_{\text{упр.р.}i} = \sum_{j=1}^k C_j \cdot A_j, \quad i = 1 \dots n,$$

где  $C_j$  — денежная оценка  $j$ -го ресурса, привлекаемого для управления риском;

$A_j$  — стоимость альтернативного размещения  $j$ -го ресурса.

Таким образом, средства, потраченные на внедрение и реализацию стратегии управления риском, не должны превышать величину возможного убытка. Иначе, такой риск лучше принять (в этом случае убытки будут незначительными), а не уклоняться от него.

Далее стоит выяснить долю стоимости управления всеми рисками по отношению к общему чистому денежному потоку Internet-компании.

$$KPI_4 = \frac{\sum_{i=1}^n C_{\text{упр.р.}i}}{\text{ЧДП}} \rightarrow \min, \quad (4)$$

где ЧДП — чистый денежный поток компании, который находится следующим образом:

$$\text{ЧДП} = \text{ЧП} + A,$$

где ЧП — чистая прибыль (прибыль за вычетом выплат по кредиту после налогообложения);

$A$  — амортизация.

Исходя из выражения (4), можно сделать вывод о нецелесообразности и неэффективности внедрения выбранной методики управления рисками, если показатель  $KPI_4$  будет ниже критического значения (например,  $KPI_4^{\text{кр.}} = 0,5$ ), определяемого отдельно для каждой компании. Это будет означать, что компания не в состоянии финансировать реализацию выбранной методики. В этом случае целесообразнее пересмотреть выбранную политику управления рисками и остановиться на менее дорогом варианте влияния на риск.

### **Выводы**

В результате работы получены показатели, характеризующие итог деятельности в области риск-менеджмента в рамках определенной Internet-компании. В зависимости от величины показателей можно сделать выводы о:

- целесообразности корректировки методики анализа и оценки степени того или иного риска в случае, если  $KPI_1 \geq 0,3$  и/или хотя бы один из остальных показателей результативности не будет удовлетворять выдвинутым выше условиям;
- эффективности использования отдельных инструментов риск-менеджмента и эффективности затрат на управление рисками в целом;
- возможности использования конкретной разработанной методики управления риском при реализации последующих мероприятий, минимизирующих как сам риск, так и его возможные последствия;
- работе группы по управлению рисками и использовании соответствующих выводов в системе мотивации и оплаты труда риск-менеджеров.

Разработанная модель оценки эффективности управления рисками может использоваться консалтинговыми компаниями, оказывающими услуги в области финансового прогнозирования, оценки связанных с бизнесом рисков и поддержки принятия оптимальных, с точки зрения доходности и риска, управленческих решений.

Несмотря на всю сложность риск-менеджмента данная модель позволит внедрить процесс управления рисками в Internet-компаниях, и, пройдя весь этот путь, последняя сможет обеспечить себе условия для построения эффективной системы управления.

### **Литература**

1. Суворов А.И. Система сбалансированных показателей деятельности предприятия как основа реализации его стратегических целей. Источник: [http://срб.ру/analytic1\\_4.html](http://срб.ру/analytic1_4.html).
2. Кинев Ю.Ю. Оценка рисков финансово-хозяйственной деятельности предприятий на этапе принятия управленческого решения // Менеджмент в России и за рубежом, 2000 — №5. Источник: <http://www.hedging.ru/publications/384>.

# ИССЛЕДОВАНИЕ КОМБИНИРОВАННОГО ПРОТОКОЛА СИНХРОНИЗАЦИИ ЛОГИЧЕСКИХ ПРОЦЕССОВ ПРИ РАСПРЕДЕЛЕННОМ СОБЫТИЙНОМ МОДЕЛИРОВАНИИ ЦИФРОВЫХ УСТРОЙСТВ

Ю.В. Ладыженский, Ю.В. Попов  
Кафедра ПМИИ, ДонНТУ

## ***Анотація***

*У статті розглянуті основні властивості різних алгоритмів розподіленого подійного моделювання. Проведено експериментальне дослідження комбінованого протоколу синхронізації логічних процесів.*

## ***Введение***

Проверка работоспособности проекта является важным этапом создания любого электронного устройства. Выполнение процесса моделирования является эффективным средством проверки работоспособности проекта. Моделирование больших и сверхбольших проектов требует много вычислительных ресурсов. Одним из возможных решений проблемы ускорения моделирования является использование нескольких компьютеров, работающих параллельно.

Разные алгоритмы моделирования и некоторые их модификации рассмотрены в [1,2]. Эффективным средством проектирования систем распределенного моделирования является объектно-ориентированный подход [3]. Вопросы параллельного и распределенного VHDL-моделирования рассматривались в [4].

Цели и задачи исследований включают в себя анализ алгоритмов распределенного моделирования, экспериментальное исследование и выявление особенностей комбинированных протоколов синхронизации логических процессов.

## ***1. Методы моделирования цифровых устройств***

Функционирование цифровых устройств можно представить наборами состояний и событий (рис. 1). Текущие значения сигналов в узлах дискретной схемы представляют собой состояние схемы, а изменения сигналов во времени — события.  $T(e_i)$  — момент виртуального времени, на который запланировано событие  $e_i$ . Выполнение процесса моделирования означает обработку событий, которые появляются в определенном порядке во времени и изменение состояний, соответствующее этим событиям. Для изменения состояний в будущем планируются новые события [5].

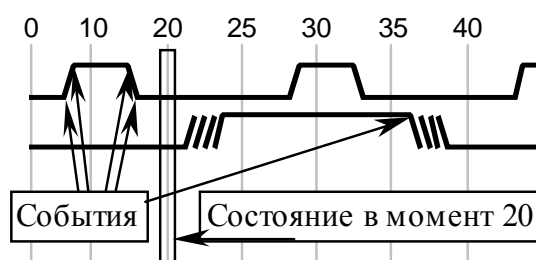


Рисунок 1 — События и состояния

Алгоритмы моделирования отличаются порядком обработки и планирования новых событий. При последовательном моделировании события обрабатываются по одному, в порядке их наступления в виртуальном времени (VT). Если  $T(e_i) = T(e_j)$ , то события  $e_i$  и  $e_j$  обрабатываются последовательно.

При параллельном моделировании ускорение достигается за счет распараллеливания алгоритма моделирования.

При распределенном моделировании учитывается параллелизм моделируемой схемы. При этом схема разрезается на части и каждая часть моделируется на отдельном моделирующем процессоре (МодПр).  $P(e_i)$  — процессор, на котором обрабатывается событие  $e_i$ .

После разрезания схемы возникают события, которые генерируются в одних частях схемы, а воздействуют на другие части схемы. Для обеспечения правильной последовательности обработки необходима синхронизация копий МодПр.

Один из способов синхронизации — следить за хронологическим порядком обработки событий глобально по всей системе. Достичь этого можно, если на каждом МодПр локальное виртуальное время (LVT) совпадает с глобальным виртуальным временем (GVT). Такой алгоритм распределенного моделирования называется синхронным.

В асинхронных протоколах синхронизации локальные виртуальные часы показывают разное LVT, отличное от GVT. Это позволяет проводить одновременную обработку событий, запланированных на разные моменты VT.

При асинхронном моделировании возможна ситуация, когда, например, процессор P1 продвинул свое время далеко вперед, и в этот момент к процессору P1 пришло сообщение о событии от процессора P2, LVT которого меньше. Такое событие называется событием в локальном виртуальном прошлом, или запоздавшим событием.

В консервативных протоколах синхронизации МодПр перестают увеличивать свое LVT и переходят в режим ожидания, если

есть возможность того, что в момент времени  $t_{late} \geq LVT$  придет запоздавшее событие  $e_{late}$ ,  $T(e_{late}) = t_{late}$ .

В консервативных протоколах синхронизации возможна ситуация, когда МодПр ожидают событий друг от друга по циклу. В результате ожидания ни один из МодПр системы не может увеличить LVT. Такое состояние системы называется состоянием тупика. Алгоритмы вывода системы из тупика допускают ввод системы в состояние тупика, но при этом предоставляют механизм обнаружения и вывода системы из этого состояния. Алгоритмы предупреждения тупиков не допускают возникновения тупиков.

Скорость консервативного моделирования зависит от того, насколько сильно МодПр может увеличить LVT до перехода в состояние тупика. Момент VT, в котором ожидается состояние тупика, называется горизонтом расширения LVT (LVTH). Алгоритмы определения LVTH называются алгоритмами предсказания.

В оптимистических протоколах синхронизации МодПр не ожидают возможности обработки событий и допускают появление запоздавших событий. При появлении запоздавшего события в момент  $t_{late}$  производится откат в одно из прошлых состояний.

Сохранение состояний системы требует много памяти. Экономия памяти достигается за счет сохранения только изменившихся переменных состояния. Однако при таком способе экономии памяти для выполнения отката требуется перебрать все изменения в моменты времени от LVT до  $t_{late}$  (из настоящего в прошлое).

В комбинированных протоколах синхронизации МодПр обрабатывают события, запланированные на моменты времени  $LVT \leq T(e_i) \leq LVTH + \Delta t$ , где  $0 \leq \Delta t \leq +\infty$  — степень оптимистичности протокола синхронизации.

## ***2. Экспериментальное исследование комбинированного протокола синхронизации логических процессов***

Эксперименты проводились на трех схемах из iscas89 [5] s27, s400 и s1196. Схемы были разрезаны на три части. Основным критерием оптимальности разрезания было равномерное распределение нагрузки. На вход этих схем подавалось одно и то же входное воздействие, сгенерированное случайным образом. Моделирование велось до момента  $GVT = 10000$ . Использовались компьютеры P4 2.4 ГГц. Сетевое оборудование — 100 Мбит. На машинах установлена Windows-98 SE, для передачи сообщений по

сети использовался DirectX 8.0. Одному компьютеру соответствует один моделирующий процессор.

Зависимость физического времени моделирования и количества обработанных событий в системе от степени оптимизма (в единицах VT) при использовании оптимистического протокола синхронизации приведена на рис. 2. На рисунках видно, что процесс моделирования очень неустойчив при большой степени оптимизма. Чем больше степень оптимизма, тем больше значения имеют случайные задержки и разность в скорости обработки событий.

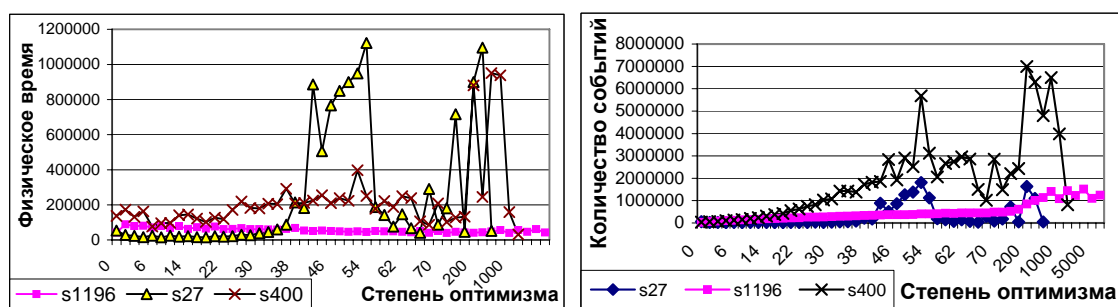


Рисунок 2 — Зависимость физического времени моделирования и количества обработанных событий в системе от степени оптимизма

На рис. 3 показана зависимость физического времени моделирования (RealTime) и суммарного физического времени простоя (WaitTime) от степени оптимизма. Величина  $K_{WR} = \text{WaitTime} / \text{RealTime}$ . Из графика видно, что хотя с увеличением степени оптимизма время простоя увеличивается не так быстро, как время моделирования, все же простой занимает основную часть (в среднем больше половины) времени моделирования. Следует ожидать, что при дальнейшем увеличении степени оптимизма время простоя будет уменьшаться до нуля.

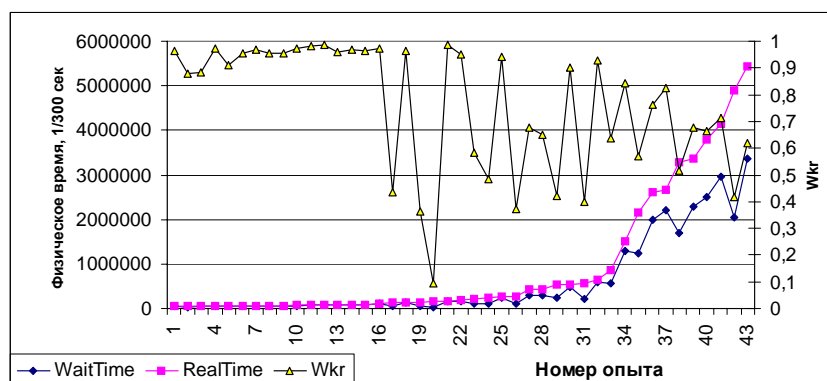


Рисунок 3 — Зависимость физического времени моделирования (RealTime) и времени простоя (WaitTime) от степени оптимизма



### ***Выводы***

Приведена классификация и рассмотрены свойства основных алгоритмов моделирования дискретных цифровых устройств.

Проведено экспериментальное исследование комбинированного протокола синхронизации. Исследована зависимость физического времени моделирования, суммарного количества обработанных событий и суммарного времени простоя от степени оптимизма.

Показано, что при моделировании небольших схем увеличение степени оптимизма может значительно замедлить процесс моделирования. При моделировании схем средних размеров увеличение степени оптимизма приводит к увеличению числа обрабатываемых событий, однако общее время моделирования уменьшается.

При увеличении степени оптимизма процессоры значительную часть времени проводят в ожидании возможности увеличения LVT. Это связано с тем, что после откатов, число которых увеличивается при увеличении степени оптимизма, требуется переычисление некоторых данных, а значит требуется повторное ожидание возможности увеличения LVT.

### ***Литература***

1. Ferscha Alois. Parallel and Distributed Simulation of Discrete Event Systems. In Hardbound of Parallel and Distributed Computing. McGraw-Hill, 1995.
2. Chandy K.M., Misra J. Asynchronous Distributed Simulation via a Sequence of Parallel Computations. Communications of the ACM, 24(11): 198–206, November, 1981.
3. Ладыженский Ю.В., Попов Ю.В. Объектно-ориентированная модель протоколов синхронизации при распределенном логическом моделировании цифровых устройств // Наукові праці Донецького національного технічного університету. Серія: Обчислювальна техніка та автоматизація. Випуск 64. — Донецьк: Вид-во ДонНТУ, 2003. — 280с. — С. 212–221.
4. Lungeanu D., Shi C. Parallel and Distributed VHDL Simulation. — ACM SIGDA Publications on CD-ROM: Date 2000. — Department of Computer Science, University of Iowa. — p. 658.
5. Brglez F., Bryan D., Kozminski K. Combinational Profiles of Sequential Benchmark Circuits. ISCAS'89 Benchmark Circuits. — Proc. IEEE Int. Symposium on Circuits and Systems. — May 1989. — pp. 1929–1934.

## О МЕТОДОЛОГО-ТЕХНОЛОГИЧЕСКИХ АСПЕКТАХ ПОСЛЕДОВАТЕЛЬНОГО ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

В.Б. Бигдан, В.А. Пепеляев, М.А. Сахнюк  
Институт кибернетики имени В.М. Глушкова НАН Украины

### *Анотація*

*В доповіді розглядаються особливості розробки та реалізації проблемно-орієнтованих застосувань, що базуються на методології та технологіях послідовного імітаційного моделювання.*

Характерной особенностью современного этапа развития имитационного моделирования, как эффективной методологии исследования и проектирования сложных систем, является "сосуществование" трёх различных направлений, ориентированных на процессы последовательного моделирования, распределённого моделирования и моделирования на основе концепции виртуальной реальности (VR — Virtual Reality) соответственно.

В представленной ниже таблице приведен перечень наиболее известных систем имитационного моделирования, базированных на соответствующих подходах.

Вид имитационного моделирования	Имитационные системы или программные продукты
Последовательное моделирование	GPSS, ARENA, EM-Plant, QUEST, AutoMod, WITNESS, ProModel, SIMUL8, ISS 2000, Crystal Ball, AnyLogic
Распределенное моделирование	SIMNET, SPEEDES, ParaSol, HLA, НЕДИС-Р
Моделирование на основе VR	продукт фирмы Tecnomatix, продукт фирмы DELMIA

Заметим, что термин «последовательное моделирование» (синонимы — сосредоточенное или нераспределенное) здесь и далее используется для представления подходов, реализуемых на однопроцессорных платформах. Последовательное моделирование связано с созданием и внедрением на современных вычислительных платформах языков и систем в традиционном для имитационного моделирования стиле. Коммерческие системы семейства GPSS, ISS 2000, ARENA, SIMUL8, QUEST, ProModel, AutoMod, WITNESS, Crystal Ball, базируются на технологических стандартах последовательного модели-

рования с использованием возможностей визуализации и анимации. Система AnyLogic разработана российской фирмой XJ Technology, Санкт-Петербург. Интерактивная система ISS 2000 разработана в Киевском национальном техническом университете «КПИ» [1,2].

Указанные системы моделирования ориентированы на решение задач бизнеса, образования, управления финансами, социально-экономическими и производственными процессами, исследование транспортных и логистических систем. За сравнительно короткий период пользователями указанных систем стали очень многие солидные фирмы: IBM, Bell Laboratories, Motorola, Ford Motor Company, Boeing Aircraft, British Airways, Virgin Atlantic, Hewlett Packard Corporation, USA Air Force, British Steel, Nissan Motors, ИМПЭКС БАНК, Русский Алюминий.

Второе направление в области современного имитационного моделирования связано с методологией и технологическими аспектами распределенного, реализуемого на сетевой архитектуре моделирования. Наиболее известными здесь являются такие системы как SIMNET, SPEEDES, ParaSol. Система SIMNET в свое время интенсивно использовалась для обучения воинского персонала в чрезвычайных ситуациях, SPEEDES специализировалась главным образом, на выполнении заказов NASA, ParaSol — для реализации различных научно-исследовательских проектов. Следует отметить, что разработанная по инициативе Министерства обороны США технология HLA (High Level Architecture) принята в качестве стандарта создания и реализации распределенных имитационных приложений. В 1998 году HLA номинирована в НАТО [3,4].

Система НЕДИС-Р разработана в Институте кибернетики НАН Украины под руководством Гусева В.В. Система поддерживает последовательное и распределенное моделирование дискретно-событийных процессов. При этом обеспечивается автоматическое формирование распределенных имитационных моделей на основе их сосредоточенных аналогов для консервативной и оптимистической схем синхронизации. Система не имеет аналогов в отечественной практике имитационного моделирования [5,6].

Моделирование на основе схем VR ведет свое начало от видеоигровой индустрии. В конце 90-х годов успехи в автомобилестроительной промышленности Германии (создание условий для автоматизации процессов на всех этапах жизненного цикла изделий, начиная от их эскизного проектирования вплоть до этапа утилизации) способствовали развитию концепции e – Manufacturing. Основная суть e - Manufacturing определяется непрерывным использованием имитационных моделей в процессе проектирования и эксплуатации производст-

венных систем. Базированные на концепции e – Manufacturing схемы VR нашли широкое применение в практике исследования и проектирования различного рода производственных и логистических систем. Лидерство здесь принадлежит Германским фирмам Tecnomatix и DELMIA [7].

### ***Методологические аспекты последовательного имитационного моделирования***

Методы имитационного моделирования в отличие от классических подходов не предоставляют в распоряжение исследователя или разработчика сложных систем строгого математического аппарата для формализации соответствующих процессов, но предлагают соответствующие средства в виде множества методологических схем и технологических стандартов, предусмотренных общей методологией имитационного моделирования и поддерживаемых специально разработанными языками и системами моделирования. Наиболее известные методологические схемы: модели теории массового обслуживания и теории автоматов, модели сетей Петри, модели динамических систем, агрегативные модели, модели системной динамики, модели событийного и процессионного подхода, принятые в высокоразвитых языках моделирования.

Любая система моделирования обязательным образом предоставляет в распоряжение разработчика соответствующую методологическую схему и обобщенную модель эволюции реальных систем. Определяемая в понятийных терминах (world of view) концептуальной базы метода имитационная модель формируется как иерархически структурированное множество объектов различных классов. При этом эволюция реальных систем трактуется как выполнение имитационной модели в виде цепочки происходящих во времени непрерывных или дискретных изменений состояний системы.

К настоящему времени в зарубежной и отечественной практике последовательного имитационного моделирования накоплен значительный опыт разработки и применения методолого-технологических стандартов, в первую очередь касающихся методологии создания имитационных моделей, проектирования сценариев и программ имитационных экспериментов, организации и структурирования данных, решения проблем оценки достоверности имитационных моделей [8,9].

На рисунке 1 представлена типовая схема Балчи процесса поэтапной разработки, реализации (акроним M&S — Modeling and Simulation) и оценки достоверности (акроним VV&T — Verification Validation and Tetsting) нераспределенных имитационных приложений. Международным сообществом имитаторов приняты стандартные подходы к решению

проблемы оценки достоверности, включающие 15 принципов и порядка 70 эвристических приемов. Сплошные стрелки на рисунке указывают на переход от одного этапа процесса разработки приложения к другому. Прерывистые стрелки — используются для представления процедур, оценивающих достоверность таких переходов.

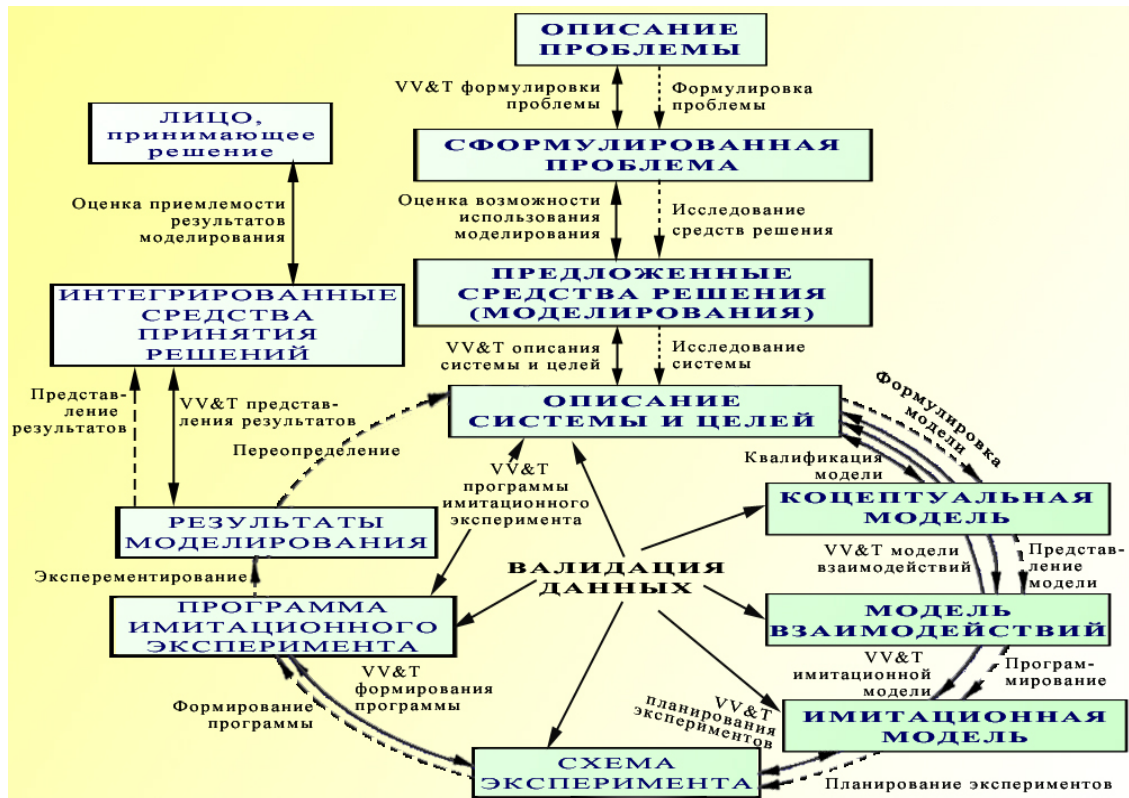


Рисунок 1 — Схема Балчи разработки и реализации имитационных приложений

### *Особенности реализации имитационных экспериментов в последовательном моделировании*

Основными составляющими любого последовательного имитационного эксперимента являются: симулятор, диспетчер, время, календарь, множество наблюдаемых переменных, имитационная модель и сценарий эксперимента. На рисунке 2 представлена общая структура и функциональные связи указанных компонент.

В определение имитационной модели наряду с описанием пассивных и активных объектов (PASOBJ и AKTOBJ) включается описание факторов, откликов и переменных модели.

Сценарий эксперимента включает определение наблюдаемых переменных, функций цели и стандартного набора модулей, ориентированных на поддержку соответствующих этапов эксперимента.

Набор таких модулей является инвариантным по отношению к приложениям, но их функциональность определяется спецификой самих приложений. Основное функциональное назначение модулей сценария отражается в их названии. Наблюдаемые переменные являются множеством регистрируемых в процессе эксперимента откликов модели. Имитационная модель передает значения своих откликов процедурам-измерителям, которые управляют регистрацией последних.

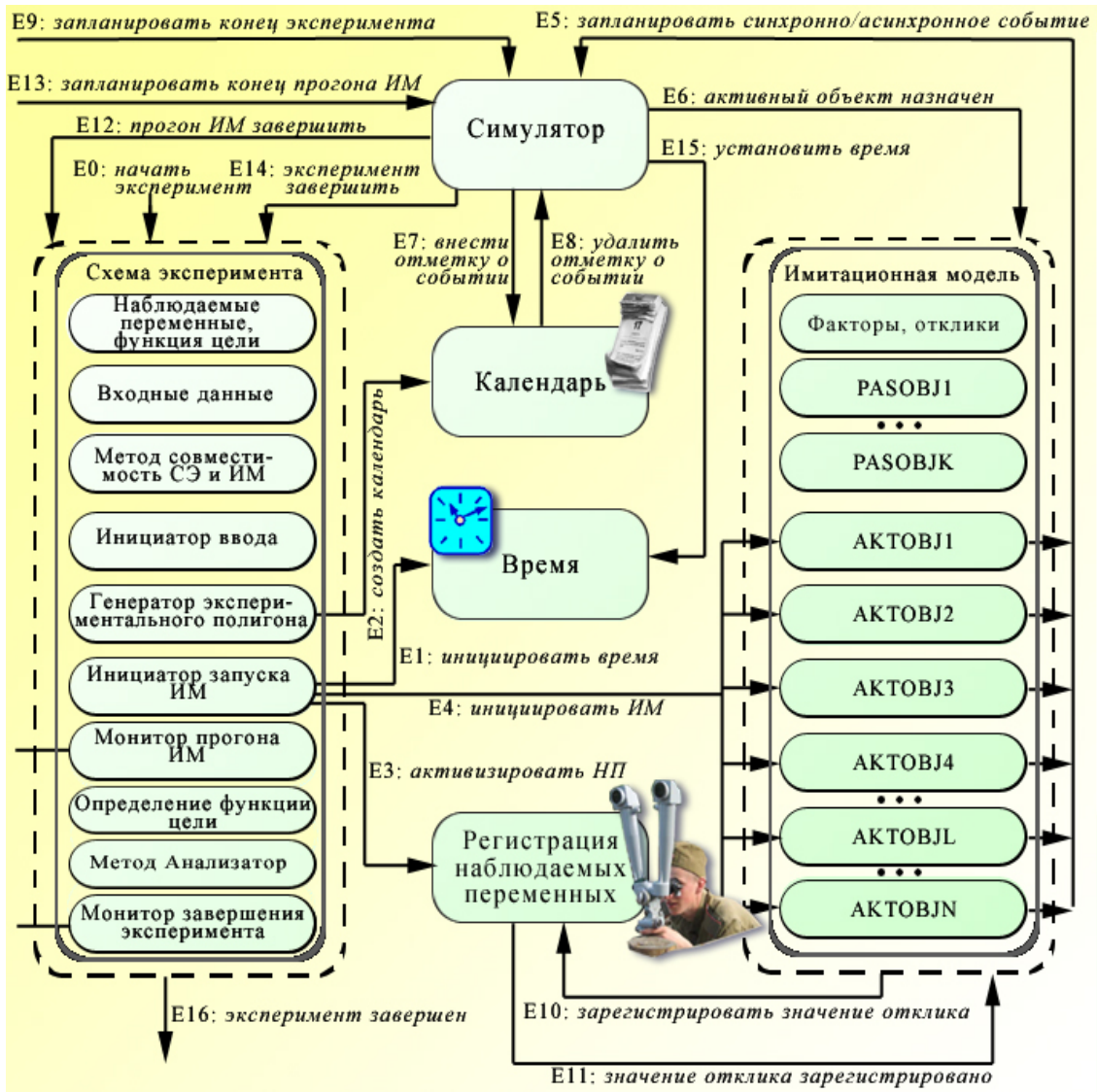


Рисунок 2 — Стандартная схема реализации типовых сценариев последовательных имитационных экспериментов

### *Современные подходы к организации программной среды поддержки процессов имитационного моделирования*

На рисунке 3 представлена интегрированная программно-информационная среда поддержки процессов проектирования и реализации



нераспределенных имитационных приложений на основе принятых стандартов, касающихся имитационных моделей, сценариев и программ имитационных экспериментов.

В процессе реализации компьютерных экспериментов имитационная модель (ИМ) и соответствующая схема эксперимента (СЭ) интегрируются в рамках программы имитационных экспериментов (ПИЭ) с такими общесистемными компонентами как симулятор, время моделирования, календарь.

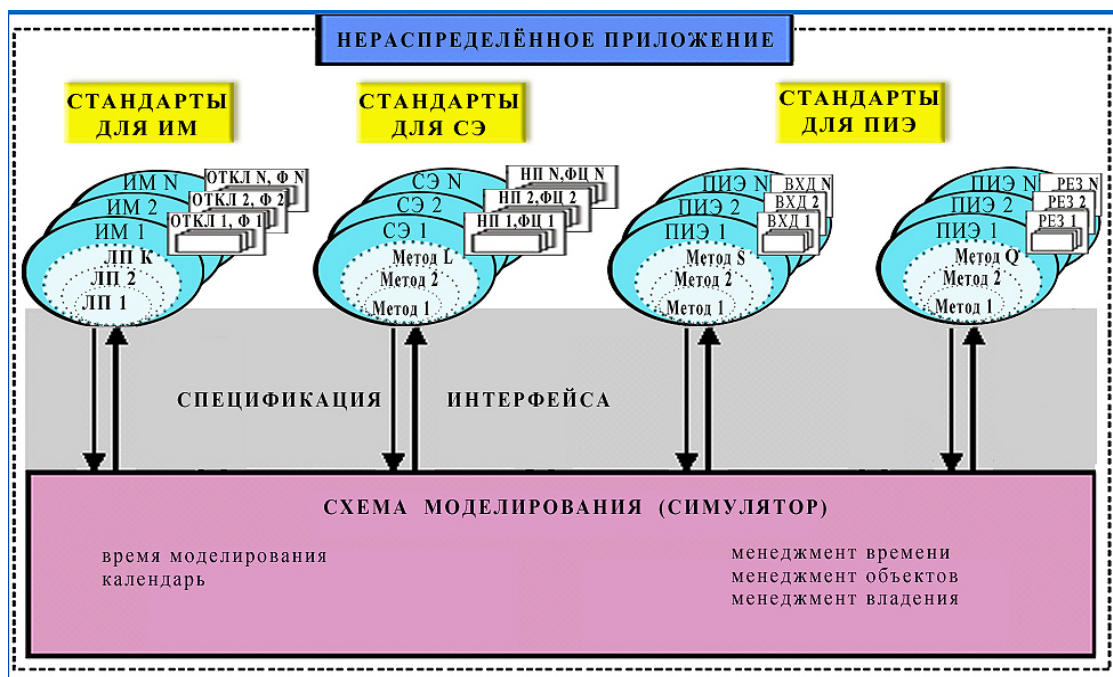


Рисунок 3 — Структура и организация программно-информационной среды поддержки имитационного моделирования

Представленная на рисунке 3 среда включает:

ИМ 1, ИМ 2, ... ИМ N — версии имитационных моделей, отличающиеся множествами логических процессов (ЛП 1, ЛП 2, ... ЛП K), факторов (Ф 1, Ф 2, ... Ф N) и оцениваемых выходных переменных (ОТКЛ 1, ОТКЛ 2, ... ОТКЛ N);

СЭ 1, СЭ 2, ... СЭ N — версии схем имитационных экспериментов, содержащие наборы методов для поддержки различных этапов экспериментов (метод 1, метод 2, ... метод L), множества входных переменных (ВХД 1, ВХД 2, ... ВХД N), методов их подготовки и ввода (метод 1, метод 2, ... метод S), множества наблюдаемых в эксперименте параметров (НП 1, НП 2, ... НП N), модели определения функции цели (ФЦ 1, ФЦ 2, ... ФЦ N);

ПИЭ 1, ПИЭ 2,... ПИЭ N — версии программ имитационных экспериментов, ориентированных на реализацию конкретных сценариев, скомпонованных на основе соответствующих версий ИМ и СЭ;

РЕЗ 1, РЕЗ 2,... РЕЗ N — накапливаемые результаты, отображаемые с помощью специальных методов (метод 1, метод 2,... метод Q).

### *Литература*

1. Борщёв А.В., Карпов Ю.Г. Профессиональный инструмент имитационного моделирования AnyLogic // В трудах первой всероссийской научно-практической конференции ИММОД–2003, ФГУП ЦНИИ технологии судостроения, Санкт-Петербург, 2003. — Том 1, С.64–69.
2. Томашевский В.Н., Богушевская Н.В. Интерактивная система имитационного моделирования ISS-2000. // В трудах первой всероссийской научно-практической конференции ИММОД–2003, ФГУП ЦНИИ технологии судостроения, Санкт-Петербург, 2003. — Том 1. — С.190–194.
3. Fujimoto R.M. Parallel and Distributed Simulation // in Proceedings of the Winter Simulation Conference. — 1999. — P.122–131.
4. High Level Architecture <http://www.dmsomil/public/transition/hla>.
5. Галаган Т.Н., Гусев В.В., Марьянович Т.П., Яценко Н.М. Один подход к автоматизации построения распределенной модели из ее сосредоточенного аналога // Проблемы программирования, 2002. — №1–2. — С.182–197.
6. Бигдан В.Б., Марьянович Т.П., Сахнюк М.А. От последовательных к распределенным технологиям в имитационном моделировании // В трудах первой всероссийской научно-практической конференции ИММОД–2003, ФГУП ЦНИИ технологии судостроения, Санкт-Петербург, 2003. — Том 1. — С.59–63.
7. Талуев Ю.И., Рихтер К. Комплексное применение имитационного моделирования при моделировании при реализации концепции e-Manufacturing // В трудах первой всероссийской научно-практической конференции ИММОД–2003, ФГУП ЦНИИ технологии судостроения, Санкт-Петербург, 2003. — Том 1. — С.23–27.
8. Бигдан В.Б. О Применении методолого-технологических стандартов в процессах разработки распределенных приложений // Компьютерная математика. — 2002. — №1. — С.42–53.
9. Пепеляев В.А., Черный Ю.М. О современных подходах к оценке достоверности имитационных моделей // В трудах первой всероссийской научно-практической конференции ИММОД–2003, ФГУП ЦНИИ технологии судостроения, Санкт-Петербург, 2003. — Том 1. — С.142–147.



**ОЦЕНКИ И КОНТРОЛЬ ПОГРЕШНОСТИ  
РЕШЕНИЯ ЗАДАЧИ КОШИ ДЛЯ ОБЫКНОВЕННЫХ  
ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ  
ПАРАЛЛЕЛЬНЫМИ РАЗНОСТНЫМИ МЕТОДАМИ**

Л.П. Фельдман

Кафедра ПМиИ, ДонНТУ

E-mail: feldman@r5.dgtu.donetsk.ua

***Анотація***

*Виведення формул для блокових методів чисельного рішення задачі Коші для систем ЗДР, що дозволяють виконати рішення на паралельних обчислювальних системах. Дано оцінки точності отриманих методів. Визначено умови стійкості чисельного рішення.*

***Введение***

Одним из главных вопросов, возникающих при численном решении систем обыкновенных дифференциальных уравнений, является проблема оценки погрешности приближенного решения. Априорная оценка глобальной погрешности разностного метода позволяет судить о сходимости приближенного решения задачи к точному и, следовательно, о его применимости. Оценка локальной погрешности, получаемая на каждом шаге вычислений, позволяет автоматически выбирать шаг интегрирования, обеспечивающего заданную точность приближенного решения.

Статья содержит обобщение результатов исследований [1,2,3,5], посвященных параллельным методам численного решения задачи Коши для системы обыкновенных дифференциальных уравнений и является продолжением ранее опубликованных работ [6,7,8,9,10]. В ней приводится доказательство сходимости приближенного решения для  $m$ -шаговых  $k$ -точечных блочных методов, что представляет обобщение ранее опубликованных результатов. В [7] приведено доказательство сходимости для одношаговых  $k$ -точечных, а в [9] — для многошаговых многоточечных методов при  $k=m$ . Рассмотрены также и методы оценки локальной погрешности решения блочными разностными схемами и даны практические рекомендации их использования для более широкого набора параллельных разностных схем.

***1. Параллельные вычислительные схемы блочных методов.***

Для упрощения изложения рассмотрим вначале решение задачи Коши для одного обыкновенного дифференциального уравнения первого порядка

$$x' = f(t, x), x(t_0) = x_0 \quad (1)$$

блочными методами. По аналогии с последовательными методами численного решения задачи Коши для обыкновенных дифференциальных уравнений различают два типа параллельных методов: одношаговый блочный метод и многошаговый блочный метод. Также как и в последовательных методах, одношаговые блочные методы позволяют получить решение, используя начальные условия задачи. Для начала решения многошаговым блочным методом необходимо найти, каким-либо одношаговым методом значения в соответствующем числе точек начального отрезка. При использовании вычислительной схемы параллельного одношагового блочного метода множество точек равномерной сетки с шагом  $\tau$ , т.е.  $\Omega_\tau = \{t_l = l\tau, l = 0, 1, 2, \dots\}$ , разобьем на  $N$  блоков, содержащих  $k$  точек каждый (рис 1). В каждом блоке введем номер точки  $i = \overline{1, k}$  и обозначим через  $t_{n,i} = t_{0,1} + [(n-1)k + i]\tau$  точку  $n$  блока с номером  $i$ , а множество  $k$  точек  $t_{n,i}$  принадлежащих блоку, через  $T_n^{(k)}$ , т.е.  $t_{n,i} \in T_n^{(k)}, i = \overline{1, k}, n = 1, 2, \dots$ . Начальную точку  $t_{0,1}$  включим в Блок 0. При численном решении задачи Коши одношаговым блочным методом для каждого следующего блока новые  $k$  значений приближенного решения вычисляются одновременно с использованием значения только в последней точке предшествующего блока.

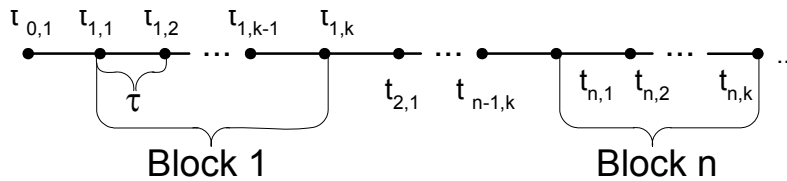


Рисунок 1 — Схема разбиения на блоки для одношагового  $k$ -точечного метода

Обозначим через  $u_{n,i}$  — приближенное значение решения задачи Коши (1) в точке  $t_{n,i}$  — обрабатываемого блока. Тогда для одношаговых блочных методов разностные уравнения имеют вид:

$$u_{n,i} = u_{n,0} + i\tau \left[ b_1 f_{n,0} + \sum_{j=1}^k a_{i,j} f_{n,j} \right], i = \overline{1, k}, n = 1, 2, \dots, \quad (2)$$

где  $f_{n,j} = f(t_n + j\tau, u_{n,j})$ , шаблон которых приведен на рис. 2.

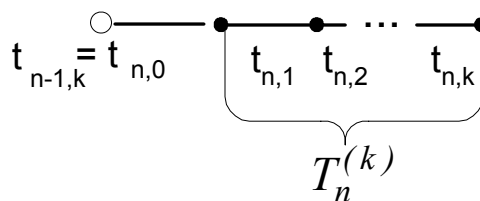


Рисунок 2 — Шаблон одношаговой  $k$ -точечной разностной схемы

В случае многошагового блочного метода начальный блок будет содержать точки сетки, в которых заданы начальные значения приближенного решения необходимые для продолжения расчета (рис 3).

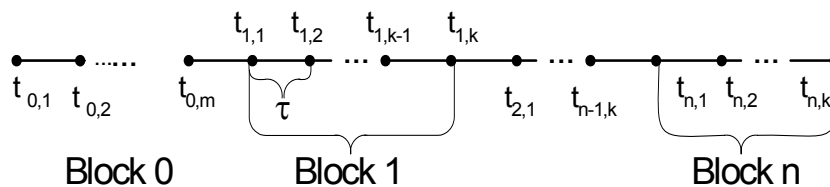


Рисунок 3 — Схема разбиения на блоки для  $m$ -шагового  $k$ -точечного метода

В общем случае уравнения многошаговых разностных методов для блока из  $k$  точек при использовании вычисленных значений приближенного решения в  $m$  предшествующих блоку узлах, с учетом введенных выше обозначений можно записать в виде:

$$u_{n,i} = u_{n,0} + i\tau \left[ \sum_{j=1}^m b_{i,j} f_{n,j-m} + \sum_{j=1}^k a_{i,j} f_{n,j} \right], i = \overline{1, k}, n = 1, 2, \dots \quad (3)$$

Формулы (3) определяют  $m$ -шаговый  $k$ -точечный разностный метод, шаблон которого приведен на рис. 4. В нем множество точек

$$T_n^{(k)} = \{t_{n,1}, t_{n,2}, \dots, t_{n,k}\},$$

в которых по формулам (3) определяются приближенные значения решения. Множество  $T_{n-1}^{(m)} = \{t_{n,1-m}, t_{n,2-m}, \dots, t_{n,0}\}$ , содержит точки, приближенное значение решения в которых, было вычислено на предыдущем этапе.

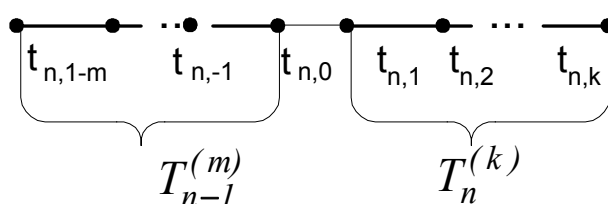


Рисунок 4 — Шаблон  $m$ -шаговой  $k$ -точечной разностной схемы

Определить коэффициенты  $a_{ij}$  и  $b_{ij}$  формул (2) и (3) можно интегро-интерполяционным методом. Построим интерполяционный многочлен  $L_{m+k-1}(t)$  с узлами интерполяции  $t_{n,j-m}$  и соответствующим им значениям правой части уравнения (1)  $f_{n,j-m} = f(t_{n,j-m}, u_{n,j-m})$ ,  $j = \overline{1, m+k}$ . Проинтегрировав его в пределах  $(t_{n,0}, t_{n,i})$ ,  $i = \overline{1, k}$

$$u_{n,i} = u_{n,0} + \int_{t_{n,0}}^{t_{n,i}} L_{m+k-1}(t) dt,$$

получим уравнения (3) для выбранных  $m$  и  $k$ .

Например, для получения формул трехшагового двухточечного метода, необходимо построить интерполяционный многочлен  $L_5(t)$  с узлами интерполяции  $t_{n,j-m}$  и соответствующих им  $f_{n,j-m}$ ,  $m=3, k=2, j=\overline{1-m, k}$  и

проинтегрировать в соответствующих пределах. В результате получим следующие формулы

$$u_{n,1} = \frac{1}{720} \tau (11 f_{n,-2} - 74 f_{n,-1} + 456 f_{n,0} + 346 f_{n,1} - 19 f_{n,2}) + u_{n,0}$$

$$u_{n,2} = -\frac{1}{90} \tau (f_{n,-2} - 4 f_{n,-1} - 24 f_{n,0} - 124 f_{n,1} - 29 f_{n,2}) + u_{n,0}$$

Формулы для трехшагового двухточечного метода определяют значения приближенного решения неявно, Решение приведенной выше нелинейной системы (37) начинается со значения  $n=1$ , при этом необходимо заранее каким-либо иным одношаговым методом вычислить значения  $u_{01}$  и  $u_{02}$  (см. рис. 3)

## 2. Погрешность аппроксимации блочных методов.

Выражения для невязок  $m$ -шагового  $k$ -точечного разностного метода (2) на решении  $x(t)$  исходного дифференциального уравнения имеют вид

$$r_{n,i} = -\frac{x_{n,i} - x_{n,0}}{i\tau} + \sum_{j=1}^m b_{i,j} x'_{n,j-m} + \sum_{j=1}^k a_{i,j} x'_{n,j}, \quad i = \overline{1, k}, \quad (4)$$

где  $x_{n,i} = x(t_n + i\tau)$ ,  $x_{n-l,m} = x_{n,0}$ ,  $x'_{n,i} = x'(t_n + i\tau) = f(t_n + i\tau, x_{n,i})$ ,  $x'_{n-l,m} = x'_{n,0}$ .

Разлагая  $x(t_n + i\tau)$  и  $x'(t_n \pm j\tau)$  в ряды Тейлора в окрестности точки  $t_n$ , получим

$$\frac{x_{n,i} - x_{n,0}}{i\tau} = \sum_{l=1}^p \frac{(j\tau)^{l-1}}{l!} x^{(l)}_{n,0} + O(\tau^p), \quad i = \overline{1, k},$$

$$x'_{n-l,j} = \sum_{l=1}^p \frac{((j-m)\tau)^{l-1}}{(l-1)!} x^{(l)}_{n,0} + O(\tau^p), \quad j = \overline{1, m},$$

$$x'_{n,j} = \sum_{l=1}^p \frac{(j\tau)^{l-1}}{(l-1)!} x^{(l)}_{n,0} + O(\tau^p), \quad j = \overline{1, k},$$

Подставляя эти разложения в выражение (4) для невязки, будем иметь

$$r_{n,i} = -\sum_{l=1}^p \frac{(j\tau)^{l-1}}{l!} x^{(l)}_{n,0} + \sum_{j=1}^m b_{i,j} \sum_{l=1}^p \frac{((j-m)\tau)^{l-1}}{(l-1)!} x^{(l)}_{n,0} + \sum_{j=1}^k a_{i,j} x'_{n,j} \sum_{l=1}^p \frac{(j\tau)^{l-1}}{(l-1)!} x^{(l)}_{n,0} + O(\tau^p), i = \overline{1, k}. \quad (5)$$

Сгруппируем члены с одинаковыми производными и изменим порядок суммирования в последнем выражении, тогда получим

$$r_{n,i} = \left( \sum_{j=1}^m b_{i,j} + \sum_{j=1}^k a_{i,j} - 1 \right) x'_{n,0} + \sum_{l=2}^p \frac{\tau^{l-1}}{(l-1)!} x^{(l)}_{n,0} \times \left( \sum_{j=1}^m b_{i,j} (j-m)^{l-1} + \sum_{j=1}^k j^{l-1} a_{i,j} - \frac{i^{l-1}}{l} \right) + O(\tau^p), i = \overline{1, k}. \quad (6)$$

Отсюда следует, что погрешность аппроксимации имеет порядок  $p$ , если выполнены условия

$$\sum_{j=1}^m b_{i,j} + \sum_{j=1}^k a_{i,j} = 1, \quad (7)$$

$$\sum_{j=1}^{m-1} b_{i,j} (j-m)^{l-1} + \sum_{j=1}^k j^{l-1} a_{i,j} = \frac{i^{l-1}}{l}, i = \overline{1, k}, l = \overline{2, k+m}.$$

Система уравнений (7) для каждого фиксированного  $i$  содержит  $p$  уравнений и  $k+m$  неизвестных  $a_{i,j}$ ,  $j = \overline{1, k}$  и  $b_{i,j}$ ,  $i, j = \overline{1, m}$ . Положим, чтобы  $p = k + m$ , тогда из системы (7) при фиксированном  $i$  можно будет определить неизвестные коэффициенты  $b_{i,j}$ ,  $j = \overline{1, m}$  и  $a_{i,j}$ ,  $j = 1, 2, \dots, k$ . Поступая аналогично для каждого  $i = \overline{1, k}$ , определим все неизвестные коэффициенты  $b_{i,j}$ ,  $j = \overline{1, m}$  и  $a_{i,j}$ ,  $i, j = 1, 2, \dots, k$ , которые в дальнейшем будем считать элементами матриц  $\mathbf{A}$  и  $\mathbf{B}$  соответственно. Отсюда следует, что наивысший порядок аппроксимации  $m$ -шагового  $k$ -точечного блочного метода равен  $m + k$ . Его погрешность в соответствии с (6) определяется формулой

$$r_{n,i} = \left( \frac{\tau^{k+m}}{(k+1)!} x^{(m+k+1)}_{n,0} \left[ \sum_{j=1}^k j^{k+m} a_{i,j} + \sum_{j=1}^{k-1} b_{i,j} (j-m)^{k+m} - \frac{i^{k+m}}{2k+1} \right] \right) + O(\tau^{m+k+1}), i = \overline{1, k}. \quad (8)$$

Элементы  $b_{i,j}, a_{i,j}$ , матриц  $\mathbf{B}$  и  $\mathbf{A}$  можно найти, решая систему (7) для любых заданных  $k$  и  $m$ .

### 3. Сходимость и оценка погрешности многошаговых блочных методов.

Обозначим через:  $U_n^{(k)} = \{u_{n,i}, i = 1, 2, \dots, k\}$  — вектор значений приближенного решения в точках блока  $T_n^{(k)}$ ;

$F_n^{(k)} = \{f(t_{n,i}, u_{n,i}), i = \overline{1, k}\}$  — вектор, компонента которого равна значению правой части уравнения (1) в точке блока  $T_n^{(k)}$  для соответствующего значения приближенного решения;

$U_{n-1}^{(m)} = \{u_{n,j-m}, j = 1, 2, \dots, m\}$  — вектор значений приближенного решения в точках блока  $T_{n-1}^{(m)}$ ;

$F_{n-1}^{(m)} = \{f(t_{n,j-m}, u_{n,j-m}), j = \overline{1, m}\}$  — вектор, компонента  $j = \overline{1, m}$  которого равна значению правой части уравнения (1) в соответствующей точке блока

$T_{n-1}^{(m)}$  для соответствующего ей значению приближенного решения (см. рис. 4).

Запишем систему (2) в векторной форме

$$\mathbf{D}^{-1}(U_n^{(k)} - u_{n,0}\mathbf{e}) / \tau = \mathbf{B}F_{n-1}^{(m)} + \mathbf{A}F_n^{(k)}, \quad (9)$$

где  $\mathbf{D} = (d_{ii})$  — диагональная матрица с элементами  $d_{ii} = i, i = 1, 2, \dots, k$ ,  $\mathbf{B}$  — матрица с элементами  $b_{i,j}, i = \overline{1, k}, j = \overline{1, m}$ ,  $\mathbf{A}$  — матрица с элементами  $a_{i,j}, i, j = \overline{1, k}$ ,  $\mathbf{e}$  — единичный вектор-столбец.

Обозначим через  $X_n^{(k)} = \{x_{n,i} | i = \overline{1, k}\}$  — вектор значений точного решения задачи (1) в точках  $t_{n,i}$  блока  $T_n^{(k)}$ , через  $\Phi_n^{(k)} = \{f(t_{n,i}, x_{n,i}) | i = \overline{1, k}\}$  и  $\Phi_{n-1}^{(m)} = \{f(t_{n,j-m}, x_{j-m}) | j = \overline{1, m}\}$  — вектора значений правой части уравнения (1) в точках  $t_{n,i}$  и  $t_{n,j-m}$  для соответствующих им значениям  $x_{n,i}$  и  $x_{n,j-m}$  точного решения в этих точках. Получим уравнение, для определения вектора

$$Z_n^{(k)} = U_n^{(k)} - X_n^{(k)} \quad (10)$$

погрешностей в блоке  $T_n^{(k)}$ . Подставим (10) в левую часть уравнения (9), добавим к правой части и вычтем из нее выражение  $\mathbf{B}\Phi_{n-1}^{(m)} + \mathbf{A}\Phi_n^{(k)}$ . Тогда уравнение для погрешности примет вид

$$\mathbf{D}^{-1}(Z_n^{(k)} - z_{n,0}\mathbf{e}) / \tau = -\mathbf{D}^{-1}(X_n^{(k)} - x_{n,0}\mathbf{e}) / \tau + \mathbf{B}\Phi_{n-1}^{(m)} + \mathbf{A}\Phi_n^{(k)} + \mathbf{B}(F_{n-1}^{(m)} - \Phi_{n-1}^{(m)}) + \mathbf{A}(F_n^{(k)} - \Phi_n^{(k)}).$$

Входящее в правую часть выражение

$$\mathbf{r}_n = -\mathbf{D}^{-1}(X_n^{(k)} - x_{n,0}\mathbf{e}) / \tau + \mathbf{B}\Phi_{n-1}^{(m)} + \mathbf{A}\Phi_n^{(k)} \quad (11)$$

представляет собой вектор невязок разностных уравнений (9) на точном решении уравнения (1). Поскольку разностные уравнения (9) аппроксимируют исходное уравнение (1) в точках блока с порядком  $O(\tau^{k+m})$ , то имеет место оценка

$$\|\mathbf{r}_n\| = O(\tau^{k+m}). \quad (12)$$

Оставшиеся члены правой части уравнения для погрешности обозначим через

$$\rho_n = \mathbf{B}(F_{n-1}^{(m)} - \Phi_{n-1}^{(m)}) + \mathbf{A}(F_n^{(k)} - \Phi_n^{(k)}) \quad (13)$$

Тогда уравнение для погрешности запишется короче

$$Z_n^{(k)} = z_{n,0} \mathbf{e} + \tau \mathbf{D}(\mathbf{r}_n + \rho_n). \quad (14)$$

Вектор-функция  $\rho_n$  зависит нелинейно от погрешности  $Z_n^{(k)}$ . Вид этой зависимости определяется функцией  $f(t, x)$ . В дальнейшем будем предполагать, что  $f(t, x)$  удовлетворяет условию Липшица по второму аргументу, т.е.

$$|f(t, x_1) - f(t, x_2)| \leq L|x_1 - x_2| \quad (15)$$

для всех  $t, x_1, x_2$  из рассматриваемой области. Согласно формуле конечных приращений Лагранжа для соответствующих компонент  $F_n^{(k)}$  и  $\Phi_n^{(k)}$  имеем

$$f(t_{n,i}, u_{n,i}) - f(t_{n,i}, x_{n,i}) = l_{n,i} z_{n,i}, \quad i = \overline{1, k},$$

аналогично для соответствующих компонент  $F_{n-1}^{(m)}$  и  $\Phi_{n-1}^{(m)}$ , где

$$f(t_{n,i-m}, u_{n,i-m}) - f(t_{n,i-m}, x_{n,i-m}) = l_{n,i-m} z_{n,i-m}, \quad i = \overline{1, m}$$

$$l_{n,i} = f_x(t_{n,i}, x_{n,i} + \theta z_{n,i}), \quad 0 < \theta < 1, \quad i = \overline{1, k},$$

$$l_{n,i-m} = f_x(t_{n,i-m}, x_{n,i-m} + \eta z_{n,i-m}), \quad 0 < \eta < 1, \quad i = \overline{1, m}.$$

Подставляя последние в (13), получим

$$\rho_n = \mathbf{B}\mathbf{D}^{(m)} Z_{n-1}^{(m)} + \mathbf{A}\mathbf{D}^{(k)} Z_n^{(k)},$$

где  $\mathbf{D}^{(m)}$  — диагональная матрица с элементами  $d_{i,i}^{(m)} = l_{n,i-m}$ ,  $i = \overline{1, m}$  и  $\mathbf{D}^{(k)}$  — диагональная матрица с элементами  $d_{i,i}^{(k)} = l_{n,i}$ ,  $i = \overline{1, k}$

Для норм матриц в силу (15) справедливы следующие оценки

$$\|\mathbf{D}^{(m)}\| \leq L, \quad \|\mathbf{D}^{(k)}\| \leq L. \quad (16)$$

Заменим  $\rho_n$  в уравнении (14) полученным для него выражением, запишем его в виде

$$Z_n^{(k)} = z_{n,0} \mathbf{e} + \tau \mathbf{D} \mathbf{r}_n + \tau \mathbf{D} \mathbf{B} \mathbf{D}^{(m)} Z_{n-1}^{(m)} + \tau \mathbf{D} \mathbf{A} \mathbf{D}^{(k)} Z_n^{(k)}.$$

Введем нормы  $\|Z_n^{(k)}\| = \max_{1 \leq i \leq k} |z_{n,i}|$  и  $\|Z_{n-1}^{(m)}\| = \max_{1 \leq i \leq m} |z_{n,i-m}|$ . Далее,

учитывая (12) и (16) получим неравенство

$$\|Z_n^{(k)}\| \leq C\tau^{k+m+1} + (1 + k\tau L \|\mathbf{B}\|) \|Z_{n-1}^{(m)}\| + \tau k L \|\mathbf{A}\| \|Z_n^{(k)}\|,$$

которое преобразуем к виду

$$(1 - \tau k L \|\mathbf{A}\|) \|Z_n^{(k)}\| \leq C\tau^{k+m+1} + (1 + k\tau L \|\mathbf{B}\|) \|Z_{n-1}^{(m)}\|$$

Если на  $\tau$  наложить ограничение

$$\tau < \tau_0 = \frac{1}{kL \|\mathbf{A}\|}, \quad (17)$$

то оценка, связывающая нормы погрешностей в точках соседних блоков  $T_n^{(k)}$  и  $T_{n-1}^{(m)}$ , примет вид

$$\|Z_n^{(k)}\| \leq \{C\tau^{k+m+1} + (1 + kL\tau \|\mathbf{B}\|) \|Z_{n-1}^{(m)}\|\} / (1 - \tau k L \|\mathbf{A}\|), \quad (18)$$

т.к. в силу условия (17) имеет место  $(1 - \tau k L \|\mathbf{A}\|) > 0$ .

Рассмотрим следующие два случая:

1. Пусть  $m \leq k$ . В этом случае  $T_{n-1}^{(m)} \subseteq T_{n-1}^{(k)}$  (см. рис. 5), следовательно

$\|Z_{n-1}^{(m)}\| \leq \|Z_{n-1}^{(k)}\|$ , и неравенство (18), учитывая последнюю оценку, примет вид

$$\|Z_n^{(k)}\| \leq \{C\tau^{k+m+1} + (1 + kL\tau\|B\|)\|Z_{n-1}^{(k)}\|\} / (1 - \tau kL\|A\|)$$

Подставляя последовательно в последнее неравенство выражения погрешностей для блоков  $n-1, n-2, \dots, 1$ , получим

$$\|Z_n^{(k)}\| \leq \left\{ C\tau^{k+m+1} \left[ 1 + \frac{1 + k\tau L\|B\|}{1 - \tau kL\|A\|} + \left( \frac{1 + k\tau L\|B\|}{1 - \tau kL\|A\|} \right)^2 + \dots + \left( \frac{1 + k\tau L\|B\|}{1 - \tau kL\|A\|} \right)^{n-1} \right] + \left( \frac{1 + k\tau L\|B\|}{1 - \tau kL\|A\|} \right)^n \|Z_0^{(m)}\| \right\} / (1 - \tau kL\|A\|),$$

где  $\|Z_0^{(m)}\| = \text{Max}_{1 \leq i \leq m} |z_{0,i}|$  — норма погрешности приближенного решения, полученного каким-либо способом на начальном участке интегрирования для первых  $m$  узлов. Упростим последнее выражение

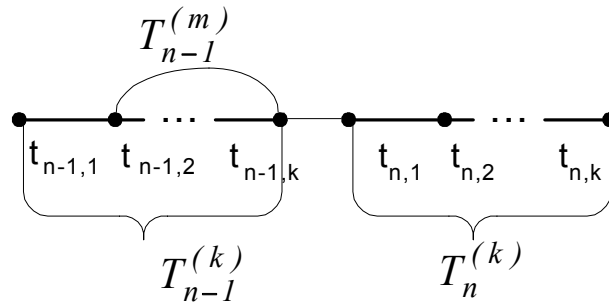


Рисунок 5 — Схема блоков  $m$ -шагового  $k$ -точечного метода для  $m < k$

$$\begin{aligned} \|Z_n(k)\| &\leq C\tau^{k+m} \frac{\left( \frac{1 + k\tau L\|B\|}{1 - \tau kL\|A\|} \right)^n - 1}{kL(\|B\| + \|A\|)} + \left( \frac{1 + k\tau L\|B\|}{1 - \tau kL\|A\|} \right)^n \|Z_0^{(m)}\| \\ &\leq C\tau^{k+m} \frac{e^{\tau knL(\|B\| + \|A\|)} - 1}{kL(\|B\| + \|A\|)} + \frac{e^{\tau knL(\|B\| + \|A\|)}}{1 - \tau kL\|A\|} \|Z_0^{(m)}\| \end{aligned}$$

Последнее неравенство справедливо для любого  $0 < n \leq N$ ; из этого следует:

$$\|Z_n(k)\| = C\tau^{k+m} \frac{e^{kTL(\|B\| + \|A\|)}}{kL(\|B\| + \|A\|)} + \frac{e^{kTL(\|B\| + \|A\|)}}{1 - \tau kL\|A\|} \|Z_0^{(m)}\| \quad (19)$$



2. Пусть  $m > k$ . В этом случае  $T_{n-1}^{(m)} \subset \{T_{n-1}^{(k)} \cup T_{n-2}^{(m)}\}$  (см. рис. 6), следовательно

$$\|Z_{n-1}^{(m)}\| \leq \text{Max} \{ \|Z_{n-1}^{(k)}\|, \|Z_{n-2}^{(m)}\| \}. \quad (20)$$

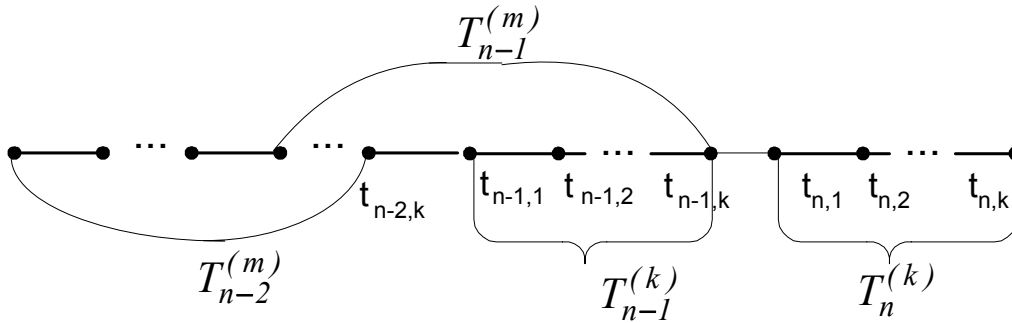


Рисунок 6 — Схема блоков  $m$ -шагового  $k$ -точечного метода для  $m > k$

Запишем неравенство (18) для  $\|Z_{n-1}^{(k)}\|$

$$\|Z_{n-1}^{(k)}\| \leq \{ C\tau^{k+m+1} + (1 + kL\tau\|B\|) \|Z_{n-2}^{(m)}\| \} / (1 - \tau kL\|A\|),$$

и подставим в (20)

$$\|Z_{n-1}^{(m)}\| \leq \text{Max} \{ [C\tau^{k+m+1} + (1 + kL\tau\|B\|) \|Z_{n-2}^{(m)}\|] / (1 - \tau kL\|A\|), \|Z_{n-2}^{(m)}\| \}$$

Поскольку  $0 < 1 - \tau kL\|A\| < 1$ , то

$$\|Z_{n-1}^{(m)}\| \leq [C\tau^{k+m+1} + (1 + kL\tau\|B\|) \|Z_{n-2}^{(m)}\|] / (1 - \tau kL\|A\|). \quad (21)$$

Подставляя последовательно в (18) выражения (21) погрешностей  $\|Z_{n-1}^{(m)}\|$  для блоков  $n-1, n-2, \dots, 1$ , получим так же, как и для случая 1, оценку (19)

Таким образом, если правая часть уравнения (1)  $f(t,x)$  удовлетворяет условию Липшица по второму аргументу с константой  $L$  и  $r_n$  — невязка  $m$ -шагового  $k$ -точечного блочного метода (2), определенная согласно (11) с оценкой (12), то при

$$\tau < \tau_0 = \frac{1}{kL\|A\|} \text{ и } \tau kn \leq T$$

для погрешности метода имеет место оценка (19).

### **Следствие.**

Если разностное уравнение (2) аппроксимирует исходное уравнение (1) и  $\|Z_0\| = \theta$ , то решение разностной задачи (2) сходится при  $\tau \rightarrow 0$  к решению исходной задачи (1), причем порядок точности совпадает с порядком аппроксимации (12).

#### 4. Алгоритм решения разностных уравнений многошаговым блочным методом.

Итерационные формулы параллельного решения системы разностных уравнений многошагового многоточечного метода (2) для произвольного блока  $n$  получим, используя (9):

$$u_n^{s+1} = u_n e + \tau DB v'_{n-1} + \tau DA (u'_n)^s, \quad s = \overline{1, k}, \quad (22)$$

где  $s$  — номер итерации. По формуле (22) пересчитываются компоненты вектора приближенного решения  $u_n^{s+1}$ , принадлежащие блоку  $n$ . При этом компоненты вектора  $v'_{n-1, i} = F_{n-1, i} = f(t_{n-1, i}, u_{n-1, i}), i = \overline{1, m}$ , определенные в узлах предшествующего блока, сохраняют свои значения при этих вычислениях, а компоненты вектора  $(u'_n)^s = (F_{n, i})^s = f(t_{n, i}, (u_{n, i})^s), i = \overline{1, k}$  вычисляются вновь на каждой итерации. Чтобы начать решение  $m$ -шаговым разностным методом, необходимо определить каким-либо одношаговым разностным методом значения приближенного решения  $u_{0, i}$  в первых  $m-1$  точках, примыкающих к начальной точке  $t_0$  отрезка интегрирования. Таким образом будут определены значения компонент  $v'_{0, i} = F_{0, i} = f(t, +i\tau, u_{0, i}), i = \overline{1, m}$  вектора  $v'_0$ . Теперь следует задать исходное приближение для компонент  $u_{i, i}, i = \overline{1, k}$  вектора  $u^0$ , значения которых могут быть, например, найдены по  $m$ -шаговым формулам Адамса-Башфорта. В общем случае  $m$ -шаговые формулы Адамса-Башфорта для значений приближенного решения в узлах  $k$ -точечного блока на первой итерации в векторной форме могут быть записаны в виде

$$u_n^1 = u_n e + \tau B_1 v'_{n-1}. \quad (23)$$

Можно показать, что выполнение  $k$  шагов вычислений по формуле (22) обеспечит получение приближенного решения с локальной ошибкой порядка  $O(\tau^{m+k+1})$ .

Приведем результаты решения следующей задачи Коши

$$x' = -10(t-1)x, \quad x(0) = 1, \quad (24)$$

трехшаговым четырехточечным методом с шагом  $\tau = 0.017$  (см. рис.7).

Поскольку точное решение задачи Коши известно, то сможем найти глобальную погрешность в каждой точке.

#### 5. Методы с контролем на шаге.

Для оценки локальной погрешности при решении одношаговым многоточечным методом используем следующий подход. Решаются две задачи на одной сетке с одним и тем же шагом  $\tau$ .

первая — одношаговым  $k$ -точечным методом;

вторая — одношаговым  $(k+1)$ -точечным методом.

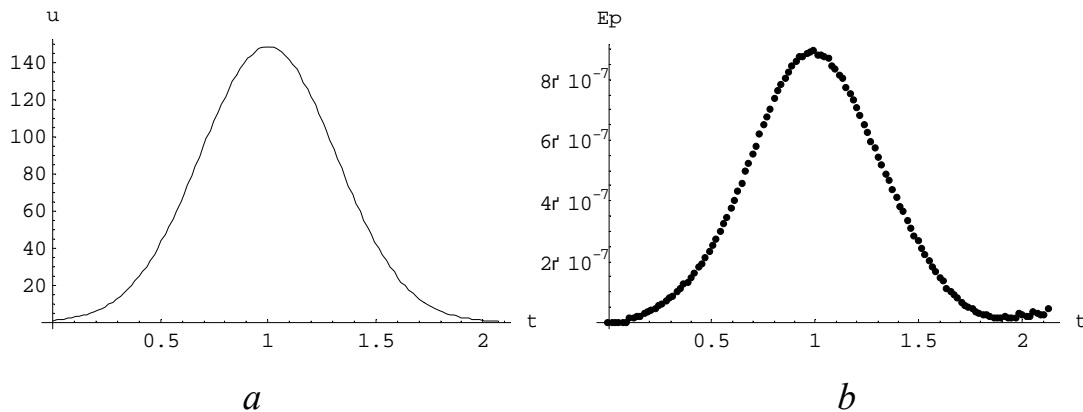


Рисунок 7 — Графики приближенного решения уравнения (24) трехшаговым четырехточечным методом *a* и глобальной погрешности *b*.

Второе решение необходимо для оценки локальной погрешности, поэтому основным является  $k$ -точечный метод. Используются значения приближенных решений в совпадающих  $k$  узлах основного блока. Лишняя  $k+1$  точка является начальным приближением в расчетах решения в следующем блоке. Локальная погрешность приближенного решения одношаговым  $k$ -точечным методом в  $i$ -ом узле блока определяется формулой

$$u_{n,i}^{(k)} - x(t_{n,i}) \approx \varphi^{(k+2)}(t_{n,0}, x_{n,0}) \tau^{k+2}, \quad i = \overline{1, k}$$

и для  $(k+1)$ -точечного метода локальная погрешность в том же узле определяется формулой

$$u_{n,i}^{(k+1)} - x(t_{n,i}) \approx \varphi^{(k+3)}(t_{n,0}, x_{n,0}) \tau^{k+3}, \quad i = \overline{1, k}.$$

Вычитая из верхнего соотношения нижнее, получим представление главного члена погрешности  $k$ -точечного метода на шаге в виде

$$\gamma_{n,i}^{(k)} = \varphi_i^{(k+1)}(t_{n,0}, x_{n,0}) \tau^{k+1} \approx u_{n,i}^{(k)} - u_{n,i}^{(k+1)}, \quad i = \overline{1, k}, \quad (25)$$

который может быть использован для оценки локальной погрешности.

Для иллюстрации приведем оценки локальной ошибки, полученные по результатам решения задачи (24) с шагом  $\tau = 0.017$ , одношаговыми трехточечным (основное решение) и четырехшаговыми методами (корректирующее решение). Оценка модуля пошаговой погрешности, полученная как разность решений одношаговыми трех- и четырехточечными методами, и действительная локальная погрешности отличаются на величину порядка  $\tau^5$ , что обеспечивает надежность используемой оценки.

Из сравнения графиков, представленных на рис. 8, следует, что оценка локальной погрешности приближенного решения уравнения (24) трехшаговым четырехточечным методом меньше (в 15 раз) соот-

ветствующей накопленной погрешности. Это полностью согласуется с теоретическими оценками.

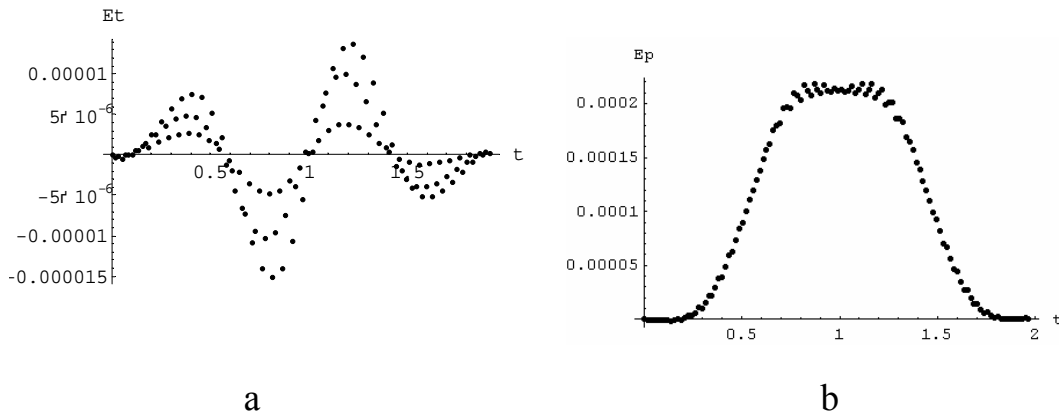


Рисунок 8 — Графики а — оценки локальной погрешности  $Et$  и б — глобальной погрешности  $Ep$  при решении задачи (24) одношаговым трехточечным методом

Для оценки локальной погрешности при решении многошаговым многоточечным методом используется аналогичный подход. Решаются две задачи на одной сетке с одним и тем же шагом  $\tau$ . первая —  $m$ -шаговым  $k$ -точечным методом; вторая —  $(m+1)$ -шаговым  $k$ -точечным методом. Второе решение используется для оценки локальной погрешности, поэтому основным является  $m$ -шаговый  $k$ -точечный метод. Оцениваются значения приближенных решений в совпадающих  $k$  узлах основного блока. Локальная погрешность приближенного решения  $m$ -шаговым  $k$ -точечным методом в  $i$ -ом узле блока определяется формулой

$$\gamma_{n,i} = u_{n,i}^{(k,m)} - x(t_{n,i}) \approx \varphi_i^{(k+m+1)}(t_{n,0}, x_{n,0}) \tau^{k+m+1}, i = \overline{1, k}, \quad (26)$$

а для  $(m+1)$ -шагового  $k$ -точечного метода локальная погрешность в том же узле определяется формулой

$$u_{n,i}^{(k,m+1)} - x(t_{n,i}) \approx \varphi_i^{(k+m+2)}(t_{n,0}, x_{n,0}) \tau^{k+m+2}, i = \overline{1, k}. \quad (27)$$

Вычитая из (26) соотношение (27), получим представление главного члена погрешности одношагового  $k$ -точечного метода на шаге в виде

$$u_{n,i}^{(k,m)} - u_{n,i}^{(k,m+1)} \approx \varphi_i^{(k+m+1)}(t_{n,0}, x_{n,0}) \tau^{k+m+1}, i = \overline{1, k}.$$

Таким образом, для главного члена погрешности получаем оценку

$$\gamma_{n,i} \approx u_{n,i}^{(k,m)} - u_{n,i}^{(k,m+1)}, i = \overline{1, k}. \quad (28)$$

Приведем оценку локальной погрешности  $\gamma$  приближенного решения двухшаговым четырехточечным методом как разность соответствующих значений решений, полученных двухшаговым и трехшаговым четырехточечными методами.

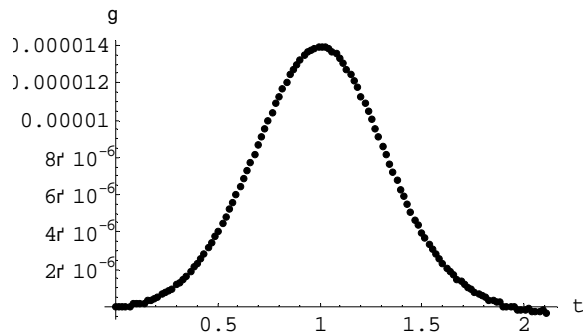


Рисунок 9 — График оценки локальной погрешности при решении уравнения (24) двухшаговым четырехточечным методом.

В рассматриваемом случае оценка локальной погрешности приближенного решения двухшаговым четырехточечным методом практически совпадает со значениями накопленной погрешности. Таким образом, получается завышенная оценка локальной погрешности. Это можно объяснить тем, что решение трехшаговым четырехточечным методом с машинной точностью совпадает с точным решением задачи (24).

### 6. Заключение. Оценка эффективности блочных алгоритмов.

При численном решении задачи Коши для сравнительной характеристики методов можно рассматривать различные показатели. В случае произвольной правой части уравнения о трудоемкости метода естественно судить по числу обращений для вычисления значений правой части уравнения на каждый узел сетки. Для оценки эффективности одношаговых блочных методов найдем отношение времени выполнения алгоритма Рунге-Кутты на однопроцессорной ЭВМ ко времени выполнения одношагового блочного алгоритма соответствующего порядка на параллельной ВС. Определим время выполнения алгоритма Рунге-Кутты  $k+1$  порядка точности на одном процессоре. Обозначим через  $t_f$  — время вычисления значения функции  $f(t,x)$ ,  $t_{ad}$ ,  $t_{mul}$  — время выполнения операции сложения и умножения соответственно. Время последовательного вычисления приближенных значений решения с точностью  $O(\tau^{k+1})$  во всех  $k$  узлах блока составит

$$T_s = (k + 1)^2 t_f + k^2 (t_{ad} + t_{mul}).$$

Для параллельного выполнения вычислений по формулам (3) закрепим за каждым узлом блока процессор. При его реализации на  $k$  процессорах можно одновременно вычислять значения  $F_{n,i,s}$ , а

затем также одновременно получить по формулам (3) значения  $u_{n,i,s}$  для каждого фиксированного  $s$ . Объединим процессоры в кольцо, чтобы иметь возможность одновременной передачи данных соседним процессорам. Обозначим через  $t_{ia}$  — время передачи числа соседнему процессору. Время параллельного вычисления приближенных значений решения с той же точностью для всех узлов блока составит

$$T_p = kt_f + k2(t_{ad} + t_{mu}) + k(k-1)t_{ia}.$$

Ускорение параллельного одношагового  $k$ -точечного алгоритма можно будет теперь вычислить по формуле

$$W(k) = T_s / T_p.$$

Если учитывать только время вычислений правой части уравнения, т.к. времена выполнения арифметических операций и обмена значительно меньше времени вычисления правой части, то ускорение  $k$ -точечного параллельного алгоритма можно считать приближенно равным

$$W(k)_{pot} = (k+1)^2 / k.$$

Для оценки ускорения  $m$ -шагового  $k$ -точечного блочного метода сравним время его выполнения на мультипроцессорной системе со временем выполнения алгоритма  $m$ -шагового метода Адамса-Башфорта на однопроцессорной ЭВМ. Метод Адамса-Башфорта можно рассматривать как многошаговый одноточечный блочный метод. Последовательное  $k$ -кратное применение формул Адамса-Башфорта позволяет вычислить приближенное решение в тех же  $k$  узлах блока, в которых параллельно за  $k$  итераций может быть вычислено решение  $m$ -шаговым  $k$ -точечным блочным методом. В этом случае время вычисления будет приблизительно одинаково. Точность приближенного решения, полученного  $m$ -шаговым  $k$ -точечным блочным методом, имеет порядок  $O(\tau^{m+k})$ , а точность приближенного решения, полученного по  $m$  шаговой формуле Адамса-Башфорта, имеет порядок  $O(\tau^{m+1})$ . Поэтому для получения решения с одинаковой точностью для метода Адамса-Башфорта надо выбрать шаг сетки мельче в  $M^{\frac{k-1}{m+k}}$  раз, чем шаг для  $m$ -шагового  $k$ -точечного метода. Здесь  $M$  — число узлов сетки на отрезке решения задачи методом Адамса-Башфорта.

Таким образом, ускорение параллельного  $m$ -шагового  $k$ -точечного алгоритма равно

$$W(m,k) \approx M^{\frac{k-1}{m+k}}.$$

Аналогично могут быть получены оценки эффективности решения задачи Коши для системы обыкновенных дифференциальных урав-

нений параллельными блочными методами. Так, например, для наиболее употребительной формулы Адамса-Башфорта с  $m = 4$  получим

$$W(4,4) \approx M^{\frac{3}{8}}.$$

Если на отрезке интегрирования взято сто узлов  $M = 100$ , то  $W(4,4) = 5.62$ .

### *Литература*

1. Worland P.V. Parallel method for the numerical solution of ordinary differential equations. IEEE Trans. Comp. C-2,10 (1976) 1045–8.
2. Дж. Холл, Дж. Уатт. Современные численные методы решения обыкновенных дифференциальных уравнений. М.: Мир, 1979. — 312с.
3. Системы параллельной обработки. Под ред. Ивенса Д. — М.: Мир, 1985. — 416с
4. Самарский А.А. Гулин А.В. Численные методы. — М.: Наука. 1989
5. Молчанов И.Н. Введение в алгоритмы параллельных вычислений. АН УССР, Инст. Кибернетики им.В.М. Глушкова. — Киев: Наукова думка. 1990. — 128 с.
6. Фельдман Л.П. Параллельные интерполяционные алгоритмы численного решения задачи Коши для обыкновенных дифференциальных уравнений на SIMD компьютере. Научн. Тр. ДонГТУ. Серия: Проблемы моделирования и автоматизации проектирования динамических систем, выпуск 10: — Донецк: ДонГТУ, 1999, с. 20–25.
7. Фельдман Л.П. Сходимость и оценка погрешности параллельных одношаговых блочных методов моделирования динамических систем с сосредоточенными параметрами. Научн. Тр. ДонГТУ. Серия: Информатика, Кибернетика та обчислювальна техніка, выпуск 15: — Донецк: ДонГТУ, 2000, с. 34–39.
8. L.P. Feldmann Implementierung und Effizienzanalyse von parallelen blockartigen Simulationsalgorithmen für dynamische Systeme mit konzentrierten Parametern. In: Möller, D.P.F. (Hrsg.): Tagungsband 14. ASIM-Symposium Simulationstechnik in Hamburg, September 2000, SCS-Europe BVBA, Ghent/Belgium 2000, S. 241–246.
9. Фельдман Л.П., Дмитриева О.А. Разработка и обоснование параллельных блочных методов решения обыкновенных дифференциальных уравнений на SIMD-структурах. Научн. Тр. ДонГТУ. Серия: Проблемы моделирования и автоматизации проектирования динамических систем, выпуск 29: — Донецк: ДонГТУ, 2001, с. 70–79.
10. L.P. Feldman, O.A. Dmitrieva, S. Gerber. Abbildung der blockartigen Algorithmen auf Parallelrechnerarchitekture. In: Tavangarian, D., Grützner, R. (Hrsg.): Tagungsband 15. ASIM-Symposium Simulationstechnik in Rostock, September 2002, SCS-Europe BVBA, Ghent/Belgium 2002, S.359—364.

**ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ ЧИСЛЕННОГО  
МОДЕЛИРОВАНИЯ СЛОЖНЫХ ДИНАМИЧЕСКИХ  
СИСТЕМ С СОСРЕДОТОЧЕННЫМИ ПАРАМЕТРАМИ  
НА ОСНОВЕ НЕЯВНЫХ МЕТОДОВ РУНГЕ-КУТТЫ**

И.А. Назарова

Донецкий национальный технический университет

**Анотація**

*У роботі запропоновані паралельні алгоритми чисельного рішення СЗДР для неявних методів Рунге-Кутта, досліджена ефективність їх відображення на структури паралельних ОЗ різної топології.*

Моделирование сложных динамических систем с сосредоточенными параметрами требует решения систем обыкновенных дифференциальных уравнений большой размерности (СОДУ). Эффективное решение таких систем возможно лишь с использованием высокопроизводительных параллельных вычислительных средств, как нового направления в области компьютерного моделирования [1–3]. В данной статье предлагается решение поставленной задачи в применении к одному конкретному классу методов интегрирования СОДУ, а именно неявным методам Рунге-Кутты, предназначенным для решения жестких задач.

Методы Рунге-Кутты с числом стадий равным  $s$  в общем виде определяются формулами (1) и (2):

$$k_i = f(x_n + c_i h, y_n + h \sum_{j=1}^s a_{ij} k_j), \quad i = \overline{1, s} \quad (1)$$

$$y_{n+1} = y_n + h \cdot \sum_{i=1}^s b_i \cdot k_i. \quad (2)$$

Коэффициенты  $a_{ij}, c_i, b_i$  определяют уникальный вариант метода Рунге-Кутты и выбираются из соображений точности. Если  $a_{ij} = 0$  при  $i \leq j$ , то метод Рунге-Кутты является явным. В том случае, если  $a_{ij} = 0$  при  $i < j$  (нижняя треугольная матрица), и хотя бы одно значение  $a_{ii} \neq 0$ , то  $k_i$  определяются неявно из уравнения:

$$k_i = f(x_n + c_i h, y_n + h \sum_{j=1}^i a_{ij} \cdot k_j), \quad i = \overline{1, s}, \quad (3)$$

такой метод называют диагонально неявным (ДНРК-метод).



Если в диагонально- неявном методе Рунге-Кутты все диагональные элементы одинаковы:  $a_{ii} = \gamma$ , то такой метод называют однократно диагонально неявным (ОДНРК-метод). Во всех остальных случаях имеем полностью неявный метод Рунге-Кутты.

Для неявных методов шаговые коэффициенты  $k_i, i = 1, \dots, s$  нельзя вычислить последовательно, так как (2) представляют собой систему уравнений, определяющих величины  $k_i$  неявным образом. Для полностью неявных методов РК все  $m \cdot s$  неизвестных должны определяться одновременно, что существенно усложняет задачу (всего шаговых коэффициентов  $s$  и размерность каждого вектора  $\bar{k}_i$  равна  $m$ ). В случае ДНРК имеем последовательность систем размерности  $m$ : для  $k_1$ , затем для  $k_2$  и так далее.

Для системы из  $m$  ОДУ:

$$\begin{cases} \bar{y}' = F(x, \bar{y}); \\ \bar{y}(x_0) = \bar{y}_0. \end{cases} \quad (4)$$

полностью неявный метод Рунге-Кутты (ПНРК) имеет вид:

$$\bar{y}_{n+1} = \bar{y}_n + h \cdot \sum_{i=1}^s b_i \cdot \bar{k}_i, \quad \bar{k}_i = F(x_n + c_i h, \bar{y}_n + h \sum_{j=1}^s a_{ij} \bar{k}_j), \quad i = \overline{1, s}.$$

Рассмотрим параллельный алгоритм численного решения СОДУ полностью неявным методом Рунге-Кутты на основе квадратурных формул Радо и Лобатто[4]. Именно эти неявные методы имеют оптимальное сочетание характеристик устойчивости и точности. Так например,  $s$ -стадийный метод РадоIA имеет порядок практически в 2 раза больше, чем число стадий и обладает А-устойчивостью.

Применение полностью неявного метода Рунге-Кутты требует определения шаговых  $\bar{k}_i, i = 1, \dots, s$  коэффициентов, которые связаны системой нелинейных уравнений. Для решения такой системы используем метод последовательных итераций:

$$\begin{cases} \bar{k}_1^{(0)} = \bar{f}(x_n + c_1 h, \bar{y}_n); \\ \bar{k}_2^{(0)} = \bar{f}(x_n + c_2 h, \bar{y}_n); \\ \dots \\ \bar{k}_s^{(0)} = \bar{f}(x_n + c_s h, \bar{y}_n). \end{cases} \quad \dots \quad \begin{cases} \bar{k}_1^{(N+1)} = \bar{f}[x_n + c_1 h, \bar{y}_n + h \cdot \bar{k}_1^{(N)}]; \\ \bar{k}_2^{(N+1)} = \bar{f}[x_n + c_2 h, \bar{y}_n + h \cdot \bar{k}_2^{(N)}]; \\ \dots \\ \bar{k}_s^{(N+1)} = \bar{f}[x_n + c_s h, \bar{y}_n + h \cdot \bar{k}_s^{(N)}]. \end{cases}$$

Анализ эффективности полученных параллельных алгоритмов производился на основе следующих показателей:

- время решения при помощи последовательного алгоритма;
- время решения при помощи параллельного алгоритма без учета и с учетом обменных операций (потенциальный и реальный параллелизм);
- анализ коммуникационной сложности алгоритма в зависимости от выбранной топологии соединения процессоров и модели передачи данных;
- коэффициенты ускорения,  $K_{уск}$  и эффективности,  $E$  параллельного алгоритма;
- масштабируемость алгоритма на основе функции изоэффективности.

Вычислительные схемы параллельных алгоритмов получены с помощью аппарата графов влияния путем отыскания максимально-независимого множества вершин на минимальном остове[5].

Для неявных методов Рунге-Кутты время интегрирования в основном определяется числом обращений к функции правой части уравнения (1) и соотношением между временем вычисления функции  $f$  и флопом. Проведенные эксперименты показали, что для сложных функций правой части (1) (время вычисления функции  $T_f$  значительно превышает время одного флопа) коэффициент ускорения практически равен числу процессоров, а коэффициент эффективности — единице:

$$K_{уск}^{nom} \approx \frac{s \cdot T_f + N \cdot s \cdot T_f}{T_f + N \cdot T_f} = s = p$$

Для оценки времени выполнения операции передачи одного сообщения объемом  $n$  байт между двумя задачами локализованными на различных процессорах (“точка-точка”) при распределенной памяти использовалась следующая модель[6–7]:

$$T_{m-m} = t_n + t_k \cdot n \cdot l, \quad t_k = \frac{y}{B},$$

где  $t_n$  — латентность, длительность подготовки сообщения для передачи;  $l$  — длина маршрута;  $t_k$  — время передачи одного байта;  $y$  — число байт в слове;  $B$  — пропускная способность канала передачи данных (байт/секунда). Эта модель подразумевает использование способа передачи неделимых блоков информации, т.е.

сообщений, поскольку объемы пересылаемых данных невелики (слово или 4 байта).

При реализации параллельных алгоритмов исследовались топологии: линейка/кольцо, решетка/тор и гиперкуб и оптимальные алгоритмы покоординатной маршрутизации. Выбор оптимальной топологии производился при анализе множества характеристик  $T_{обм}, T_{выч}, Z, K_{уск}, E$ , как некоторых функций  $\varphi(p, t_n, t_k, t_{on}, T_f)$  на основе так называемой функции изоэффективности [8], позволяющей исследовать масштабируемость параллельного алгоритма. Определение характеристик параллелизма осуществлялось с помощью пакета *Mathematica*® (Wolfram Research Inc.).

Численный эксперимент и проведенный сравнительный анализ динамических характеристик параллельных алгоритмов на основе неявных методов Рунге-Кутты показал, что достаточно большая по сравнению с явными методами вычислительная сложность этих методов не является препятствием для использования их в высокопроизводительных мультипроцессорных системах. Однако требуется тщательный учет всех составляющих параллельной системы для того, чтобы такое решение было масштабируемым и эффективным. В частности практически линейное ускорение и близкая к единичной эффективность достигаются при использовании синхронной передачи данных в топологии гиперкуб.

### *Литература*

1. Houwen P.J., Sommeijer B.P. Parallel ODE solver. // Proceedings of the International Conference on Supercomputing. — ACM Press, 2001, p.71–81.
2. Jackson K.R., Norsett S.P. The potential for parallelism in Runge-Kutta methods. Part 1: R–K formulas in standard form, SIAM J.Numer. Anal. 32,2001, p.49–82.
3. Houwen P.J., Sommeijer B.P. CWI Contribution to the development of parallel Runge-Kutta methods, Preprint NM– R9506, CWI, Amsterdam, 2003.
4. Хайпер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи: Пер. с англ. — М.: Мир, 1999. — 685с.
5. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. — СПб.: БХВ–Петербург, 2002. — 608с.
6. Foster I. Designing and Building Parallel Programs. — Addison–Wesley, 1999. — 302 с.
7. Немнюгин С.А., Стесик О.Л. Параллельное программирование для многопроцессорных ВС. — СПб.: БХВ–Петербург, 2002. — 400с.
8. Kumar V., Gupta A. Analyzing scalability of parallel algorithms and architectures.// Journal of Parallel and Distributed Computing.,22(3), 1994, p.379–391.

## ПАРАЛЛЕЛЬНЫЕ МЕТОДЫ РЕШЕНИЯ СИСТЕМ С РАЗРЕЖЕННЫМИ МАТРИЦАМИ КОЭФФИЦИЕНТОВ

О.А. Дмитриева

Донецкий национальный технический университет

### *Анотація*

*В роботі розглядається проблема паралельної організації процесів розв'язання систем лінійних алгебраїчних рівнянь, від уміння ефективної реалізації якої часто залежить сама можливість математичного моделювання найрізноманітніших процесів із застосуванням паралельних ЕОМ. Особлива увага приділена способам компактного розміщення елементів матриць в пам'яті, а також розподіленню ресурсів багатопроцесорних систем при роботі з розрідженими матрицями.*

При работе с математическими моделями многих практических задач возникает проблема решения систем линейных алгебраических уравнений с большими и разреженными матрицами коэффициентов. Примером могут служить решения уравнений в частных производных методами конечных разностей или конечных элементов, неявные разностные схемы, описывающие решение краевых задач или задач Коши [1]. Среди множества определений разреженной матрицы в данной работе принято то, которое утверждает, что приписывание матрице свойства разреженности эквивалентно утверждению о существовании алгоритма, использующего ее разреженность и делающего вычисления с ней дешевле по сравнению со стандартными алгоритмами [2].

Основное внимание в работе уделяется созданию эффективных алгоритмов решения систем линейных уравнений, матрицы которых представляются в упакованном виде, а также распределению ресурсов многопроцессорных систем при работе с разреженными матрицами. При этом возникает серьезная проблема представления в вычислительных системах с параллельной архитектурой разреженных матриц коэффициентов в специальных упакованных форматах  $RR(C)O$  — *row — wise representation complete and ordered* (строчное представление, полное и упорядоченное) и  $RR(C)U$  — *row — wise representation complete and unordered* (строчное представление, полное, неупорядоченное). В работе используется второй способ представления разреженных матриц, так как в результате матричного умножения значения, получающиеся в сетке процессорных элементов, расположены не в том порядке, как при изначальной засылке, и для того, чтобы выполнить следующий расчет, нужно потратить какое-то время на переупорядочивание элементов, для того, чтобы привести их к исходному виду, что не является обязательным для  $RR(C)U$  формата.

Имеющиеся работы, связанные с параллельной реализацией решения систем линейных уравнений, как правило, имеют направленность на эффективное распараллеливание и практически не учитывают разреженность [3–7]. Если применять для решения таких систем, например, параллельный метод *LU-декомпозиции*, то разреженность будет утрачена, и практически все нулевые элементы станут ненулевыми [6].

Для устранения этого недостатка в работе предлагаются два класса методов. В качестве точных методов решения используются алгоритмы упорядочения системы уравнений, которые позволяют сохранить или свести к минимуму искажение ненулевой структуры матрицы коэффициентов, например, алгоритмы Марковица. В соответствии с этими алгоритмами при упорядочении уравнений новый главный элемент выбирается на пересечении тех строк и столбцов, которым отвечает минимальный вес и среди которых главный элемент еще не выбирался. При этом понятно, что приведенное упорядочение системы уравнений с точки зрения сохранения ее ненулевой структуры не совпадает с классическим упорядочением [5], когда основной целью является выбор главных элементов с наибольшими значениями. Поэтому для согласования этих подходов в работе преимущество отдается алгоритму Марковица, но требуется, чтобы значение элемента, который претендует на место главного, было не меньше некоторого порогового значения, которое выбирается из условий обеспечения точности расчетов. Если это условие не выполняется, то рассматривается в качестве претендента другой элемент, пусть даже с большим коэффициентом веса.

Следует отметить, что при этом сложность алгоритмов, которые обрабатывают только ненулевые элементы матрицы уравнений, значительно уменьшается и оценивается не как раньше  $n^3/3$ , где  $n$  — размерность решаемой системы уравнений, а линейной зависимостью от общего количества ненулевых элементов матрицы.

Увеличение размерности систем линейных (или линеаризованных) уравнений снижает эффективность прямых методов, так как время решения существенно возрастает, даже для случая кодированных разреженных систем. Поскольку в современных инженерных задачах приходится иметь дело с системами линейных уравнений, размерность которых составляет сотни и тысячи, то выход из положения состоит в применении подходов диакоптики, то есть в решении больших систем уравнений по частям. Если система линейных уравнений

$$Ax = b \quad (1)$$

очень большой размерности  $n \times n$  имеет разреженную матрицу коэффициентов, то ее можно привести к блочно-диагональному виду с обрамлением и сформировать вспомогательную систему уравнений

значительно меньшей размерности  $m \times m$  для вычисления вектора  $X_2$  так называемых *определяющих* величин или переменных связи (рис. 1).

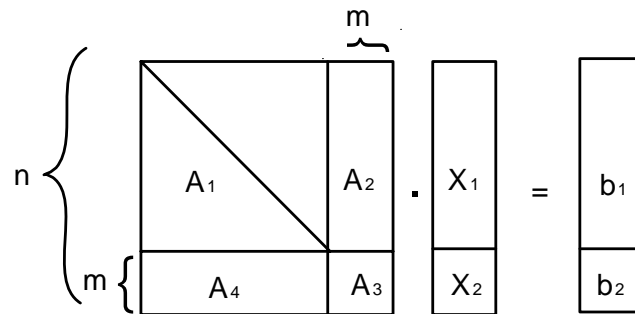


Рисунок 1 — Система уравнений с блочно-диагональной матрицей с обрамлением

И решение исходной системы (1) сводится к решению системы

$$A^* X_2 = b^* \quad (2)$$

При этом определяющие величины — элементы вектора  $X_2$  находятся с помощью следующего алгоритма логической сортировки

1. Определяется строка исходной матрицы  $A$  с минимальным количеством ненулевых элементов  $k$ .

2. Если таких строк несколько, то выбирается строка, число ненулевых элементов которой в столбцах, которые определяются  $k$  переменными, максимальное. При этом  $(k-1)$  переменная будет принадлежать к определяющим величинам, а  $k$ -ая переменная находится из выбранного уравнения.

3. Выбранные  $k$ -ые переменные исключаются из дальнейшего рассмотрения и осуществляется переход к следующей строке.

Для формирования вспомогательной системы уравнений (2), и определения вектора неизвестных  $X_2$  используется следующее матричное уравнение

$$A = \hat{A} + CD + D' D, \quad (3)$$

которое можно получить, выделив в структуре полученной блочно-диагональной матрицы с обрамлением составляющие, показанные на рис.2.

Теперь, используя треугольную матрицу  $\hat{A}$ ,  $m+1$  раз ( $m$  — размерность вектора  $X_2$ ) решаем систему

$$\hat{A} X^j = b^j, \quad j = \overline{0, m}, \quad (4)$$

где  $(b^j)_{j=0} = b$ ,

$(b^j)_{j=1, m} = C^j$  ( $j$ -й столбец матрицы  $C$ ).

Матрица  $A^*$  и вектор  $b^*$  правой части уравнения (2) набираются по столбцам из полученных решений уравнения (4), используя матрицу-маску  $D$ :

$$A^* = [DX^{(1)}, DX^{(2)}, \dots, DX^{(m)}] = [X_2^{(1)}, \dots, X_2^{(m)}], \quad (5)$$

$$b^* = DX^{(0)} = X_2^{(0)}$$

После решения уравнения (2) и нахождения вектора определяющих величин  $X_2$  переменные вектора  $X_1$  определяются из уравнения, матрица которого имеет блочно-диагональную форму с обрамлением:

$$X_{1i} = b_i - \sum_{j=1}^{i-1} A_{1,ij} X_j - \sum_{r=1}^m A_{2,ir} X_{2r}, \quad (6)$$

где  $A_{1,ij}$ ,  $A_{2,ir}$  элементы матриц  $A_1$  и  $A_2$  соответственно.

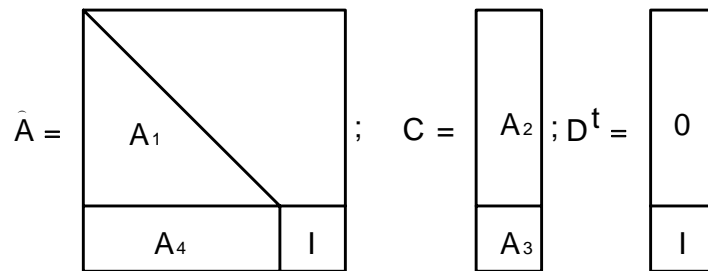


Рисунок 2 — Составляющие блочно-диагональной матрицы с обрамлением

Такой подход к решению систем линейных уравнений с разреженными матрицами коэффициентов позволяет значительно сократить объем матричных вычислений, переходя от исходной размерности системы к размерности системы определяющих величин, которая в реальных задачах имеет порядок в несколько десятков, а то и сотен раз меньший.

### Литература

1. Писсанецки С. Технология разреженных матриц. — М.: Мир, 1988. — 410 с.
2. Эстербю О., Златев З. Прямые методы для разреженных матриц. — М.: Мир, 1987. — 120 с.
3. J. Choi. A New Parallel Matrix Multiplication Algorithm on Distributed-Memory Concurrent Computers// Practice and Experience, July 1997, Vol. 6, №7, P. 543–570.
4. G. Fox, S. Otto, and A. Hey. Matrix Algorithms on a Hypercube I: Matrix Multiplication// Parallel Computing, 1997, vol. 3, P. 17–31.
5. Abdel Krim Amoura, Evripidis Bampis, and Jean-Claude Konig. Scheduling Algorithms for Parallel Gaussian Elimination With Communication Costs// IEEE Trans. on Parallel and Distributed Systems, July 1998, Vol. 9, №7, P. 687–699.
6. Antoine P. Petitet and Jack Dongarra. Algorithmic Redistribution Methods for Block-Cyclic Decompositions// IEEE Trans. on Parallel and Distributed Systems, December 1999, Vol. 10, №12, P. 1201–1216.
7. Alverson G.A., Griswold W.G., Lin C., Notkin D. and Snyder L. Abstractions for Portable, Scalable Parallel Programming // IEEE Trans. on Parallel and Distributed Systems, January 1998, Vol. 9, №1, P. 71–86.

УДК 621.372.061

## **БЛОЧНЫЕ ЧИСЛЕННЫЕ ОПЕРАТОРНЫЕ МЕТОДЫ АЛГЕБРАИЗАЦИИ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ**

С.Е.Саух

Институт проблем моделирования в энергетике  
им. Г.Е.Пухова НАНУ

### ***Анотація***

*Аналізуються властивості числових методів створених на основі різних функціональних перетворень. Сформульовано загальні вимоги до числових операторних методів алгебраїзації диференційних рівнянь. Представлено сучасні блокові числові операторні методи розроблені на основі одно- та багатовимірних рядів Ньютона.*

Работы академика Г.Е.Пухова в области создания новых операторных методов посвящены решению проблем, связанных с ограниченным применением классического операционного исчисления. Им предложено множество новых, более практичных и универсальных инструментов анализа систем и, самое главное, он привлек внимание своих учеников и последователей к фундаментальным проблемам, лежащим в основе разработки таких операционных методов анализа.

Так, в теории метода точек для представления непрерывных функций в области изображений Г.Е.Пухов предложил использовать изображающие векторы из конечного набора значений этих функций – «точек», а инфинитезимальным операциям над функциями-оригиналами были поставлены в соответствие матричные операторы интегрирования и дифференцирования, действующие на изображающие векторы. Поскольку матричные операторы строились на основе различных форм полиномиальных представлений непрерывных функций, то размерность и числовые значения элементов матриц определялись видом полиномиального представления и не зависели от значений функций в «точках». Такой подход позволил распространить на нелинейные системы методы исследования, характерные для традиционного операционного исчисления, что впервые было продемонстрировано в работах [1, 2] на примерах решения задач нелинейной электротехники.

В работах [1, 2] не рассматривались вопросы о полноте соответствия основных соотношений теории метода точек основным соотношениям традиционного операционного исчисления. В частности, в теории метода точек в ряде случаев не соблюдалась взаимная обратимость матричных операторов дифференцирования и интегрирования, что было показано в работе [3]. Матричные операторы оказывались взаимнообратными только в случае



использования для их построения тригонометрических функций. Однако такие операторы не сопрягались с начальными условиями и поэтому использовались ограниченно, например, для решения уравнений установившихся периодических режимов в нелинейных электрических цепях.

Проблема неполного соответствия в значительной степени была решена в процессе разработки теории локально-интегральных преобразований [4]. Принципиальное отличие этой теории от теории метода точек заключается в том, что при определении изображающих векторов использовались средние значения функций на некотором интервале, а не значения функций в «точках», что позволяло расширить множество преобразуемых функций и естественным образом включить в него класс обобщенных функций. В теории локально-интегральных преобразований было введено понятие определяющей матрицы линейной системы – аналога коэффициента передачи в операционном исчислении, построены взаимнообратные матричные операторы интегрирования и дифференцирования, корректно учитывающие начальные условия, разработаны основы время-частотного анализа систем.

В первоначальном виде теория локально-интегральных преобразований была основана исключительно на кусочно-постоянном представлении непрерывных функций. Поэтому для ее дальнейшего развития и создания вычислительных методов решения, например, задачи Коши для обыкновенных дифференциальных уравнений, потребовалось ввести специальные матричные операторы осреднения. Эти операторы строились независимо от значений осредняемой функции только на основе ее представления «скользящими», сдвинутыми один относительно другого, интерполяционными полиномами Ньютона. При этом на каждом последующем шаге осреднения функция была представлена новым полиномом, проходящим через узлы интерполяции, сдвинутые на один шаг вправо. Введение матричных операторов осреднения позволило построить новый класс параметрических многошаговых численных схем [5]. Однако сам способ построения матричных операторов осреднения породил проблему учета начальных условий, аналогичную проблеме «начала таблицы» многошаговых методов.

Анализируя связи между выбираемыми формами преобразования функций в область их изображений, с одной стороны, и возможностями достичь максимального упрощения трудно реализуемых операций над функциями-оригиналами при переходе в область изображений, с другой стороны, в работах [6, 7] Г.Е.Пухов

предложил использовать операционное исчисление на основе дифференциальных преобразований. Взяв за основу ряд Тейлора, как форму представления обратного преобразования изображения в функцию-оригинал, и построив формальные правила действий над коэффициентами ряда Тейлора для разных форм преобразуемых функций, он получил теоретическую основу для разработки новых аналитических, численно-аналитических и численных методов решения дифференциальных уравнений. Предложенное операционное исчисление получило широкое распространение и оказалось пригодным для анализа как линейных, так и нелинейных систем, в том числе и для некоторых одномерных по пространству систем с распределенными параметрами.

Однако, основанные на дифференциальных преобразованиях методические подходы к исследованию систем оказалось невозможно использовать в рамках традиционного операционного исчисления, поскольку относительно дифференциального спектра в целом неприменимыми какие-либо привычные инженеру-исследователю интерпретации. Вместо частотного спектра Фурье первые три составляющие дифференциального спектра интерпретировались как расстояние, скорость и ускорение. Интерпретировать другие компоненты, равно как и дифференциальный спектр в целом, не удалось.

Методы, основанные на дифференциальных преобразованиях, используются только потому, что они позволяют эффективно решать интегро-дифференциальные уравнения. Благодаря гибкости при выборе функций, аппроксимирующих решение, построенные на основе дифференциальных преобразований вычислительные алгоритмы имеют высокую аппроксимационную точность. Вместе с тем следует отметить, что такие алгоритмы строятся на основе предварительного выполнения аналитических преобразований решаемых уравнений в форму дифференциальных изображений. Поскольку преобразованные уравнения могут коренным образом отличаться от формы исходного уравнения, то создаваемые алгоритмы не могут быть воплощены в универсальные программные коды-решатели уравнений, т.е. в коды, не требующие доопределения при изменении вида уравнений. Именно поэтому исследователи, использующие дифференциальное исчисление, разрабатывают специализированные программные средства для решения уравнений определенного вида.

Работы в области теории метода точек, локально-интегральных преобразований и преобразований Тейлора послужили базой для создания численных операторных методов на основе преобразований

Ньютона [8–11]. Необходимость такой разработки вызвана нерешенностью упомянутых выше проблем объединения возможностей операторных и численных методов анализа систем, а именно:

- учета начальных условий в методе точек и «начала таблицы» в методах на основе локально-интегральных преобразований;
- взаимной обратимости операторов дифференцирования и интегрирования функций аппроксимированных полиномами;
- полиномиальной аппроксимации обобщенных функций, особенно  $\delta$ -функции, имеющей большое значение в исследовании динамических систем;
- применимости операционного исчисления к многомерным функциям, определенным в областях сложной геометрической формы.

Использование численных операторных методов на основе преобразований Ньютона позволило комплексно решить указанные проблемы. Кроме того, эти методы в отличие от других, обладают необходимым для компьютерной реализации свойством алгоритмичности. В результате их применения для решения задачи Коши были получены новые блочные численные схемы решения систем интегро-дифференциальных уравнений с высокой аппроксимационной точностью и абсолютной устойчивостью, независимой от порядка используемых аппроксимирующих функций [8, 10, 11].

В случае целых одномерных функций преобразования Ньютона схожи с дифференциальными преобразованиями. Их принципиальное отличие состоит в использовании для представления функций ряда Ньютона, а не ряда Тейлора. Поскольку коэффициенты ряда Ньютона связаны не с производными разлагаемой в ряд функции, а с ее равноотстоящими значениями, входящими в определение разделенных разностей соответствующего порядка, изображение функции представляется вектором ее значений. Требование алгоритмичности преобразований удовлетворяет переход от ряда Ньютона к конечной сумме — интерполяционному полиному Ньютона. Естественно, при этом возникают погрешности аппроксимации преобразуемых функций, которые могут быть уменьшены до приемлемого уровня как повышением порядка интерполяционного полинома, так и дроблением интервала определения функции на локальные подынтервалы (не обязательно равной длины). Такая приближенная замена преобразуемой функции совокупностью интерполяционных полиномов Ньютона позволяет представлять функцию изображающим вектором ее значений, подобно тому, как это делается в методе точек.

Операторы интегрирования и дифференцирования строятся на локальных подынтервалах. При построении операторов последовательно соблюдается правило соответствия порядков интерполяционных полиномов, представляющих соответствующие функции. В этом правиле учтено, что при дифференцировании полиномов порядка  $k$  получают полиномы порядка  $k-1$ . Следовательно, для своего однозначного представления изображающий вектор производной должен включать  $k$  ее значений, а изображающий вектор дифференцируемой функции должен состоять из  $k+1$  значения. Соответственно, изображающий вектор интегральной функции должен быть образован из  $k+1$  значений, а изображающий вектор интегрируемой функции — из  $k$  значений. Тогда связь между изображающими полиномами порядка  $k$  и  $k+1$  может быть легко установлена путем введения матричных операторов дифференцирования и интегрирования, которые оказываются взаимнообратными и позволяют корректно учитывать начальные условия с помощью векторов, сопряженных с матричными операторами [10, 11].

Введение сопряженных векторов позволило решить проблему полиномиальной аппроксимации обобщенной  $\delta$ -функции. Поскольку соотношения, реализующие операции интегрирования и дифференцирования функций в области изображений Ньютона, совпадают по форме с аналогичными соотношениями в операционном исчислении Лапласа, выяснилось, что вектор, сопряженный с матричным оператором дифференцирования, представляет собой изображение  $\delta$ -функции для случая ее кусочно-полиномиальной аппроксимации порядка  $k$ . Найденное аппроксимационное изображение  $\delta$ -функции позволяло завершить разработку численных операторных методов. Так, были получены матричные аналоги коэффициентов передачи линейных систем и созданы численные алгоритмы работы с ними, что позволило предложить оригинальный численный метод приближенного обращения лапласовых изображений в область оригиналов.

Таким образом впервые удалось преодолеть трудности, ранее препятствовавшие объединению операторных и численных методов анализа систем в рамках единого подхода. Арсенал исследователя был существенно расширен не только за счет более совершенных блочных численных схем решения интегро-дифференциальных уравнений, но и за счет корректно построенных численных операторных методов анализа систем. Новые методы были исследованы на множестве тестовых задач с начальными и краевыми условиями, а также на примере решения

одномерных уравнений газовой динамики, определенных на графе произвольной топологии [11]. Поскольку по своим точностным характеристикам и вычислительной устойчивости, а также благодаря возможности операторного представления моделей элементов топологического графа исследуемой системы предложенные методы превзошли известные численные методы, то они были положены в основу программного комплекса моделирования нестационарных неизотермических режимов в газопроводах.

Дальнейшая разработка численных операторных методов с использованием преобразований Ньютона была ориентирована на решение многомерных задач. Именно в этой области возможности традиционных численных методов таких, как метод сеток, проекционные методы и метод конечных элементов, оказывались значительно шире возможностей ранее разработанных операторных методов. Операторные методы использовались только для случаев простых прямоугольных форм областей определения многомерных функций.

Сложность разработки многомерного варианта преобразований Ньютона прежде всего состояла в том, что автору не удавалось найти в научной литературе готового варианта многомерного ряда Ньютона. С этой целью в работах [9, 10] были исследованы особенности построения многомерного ряда Тейлора и по аналогии с ним построен многомерный ряд Ньютона, коэффициенты которого, также как и коэффициенты одномерного ряда Ньютона, зависят от значений функции в точках ее многомерной области определения. С учетом требования алгоритмичности преобразований был осуществлен переход от ряда Ньютона к конечной его сумме — интерполяционному полиному Ньютона. В результате было обнаружено, что узлы интерполяции многомерного полинома равномерно заполняют симплексную подобласть. Это свойство многомерных интерполяционных полиномов Ньютона позволило создать многомерный вариант численных операторных методов не уступающих по своим возможностям, а в ряде случаев превосходящих традиционно применяемые численные методы.

В работах [9, 10] впервые описаны алгоритмы построения матричных операторов дифференцирования  $k$ -го порядка точности, действующие в нормированных симплексных областях произвольной мерности локальной системы координат. Эти специально разработанные алгоритмы базируются на множестве целых чисел, обладают свойством вложенности и обеспечивают получение операторов дифференцирования  $(N + 1)$ -мерных функций из заданных операторов дифференцирования  $N$ -мерных функций. Преобразование

локальной системы координат в глобальную позволяет применять построенные операторы в симплексной области произвольных размеров при любых вариантах задания условий на ее границах.

Многомерные операторные методы на основе преобразований Ньютона имеют высокую аппроксимационную точность и просты в применении. Их использование значительно расширяет инструментальные возможности программных средств решения краевых задач для уравнений в частных производных.

### *Литература*

1. Пухов Г.Е. Введение в теорию метода точек // Труды Таганрогского радиотехнического института. — 1954. — Вып. 1. — С. 47–77.
2. Пухов Г.Е., Борковский Б.А. К расчету электрической цепи, содержащей выпрямитель // Труды Таганрогского радиотехнического института. — 1954. — Вып. 1. — С. 78–82.
3. Пухов Г.Е. Методы анализа и синтеза квазианалоговых электронных цепей. — Киев: Наукова думка, 1967. — 568 с.
4. Береговенко Г.Я., Пухов Г.Е. Ступенчатые изображения и их применение. — Киев: Наукова думка, 1983. — 216 с.
5. Береговенко Г.Я., Саух С.Е. Параметрическая схема преобразования ступенчатых изображений и многошаговые методы численного решения дифференциальных уравнений // Электронное моделирование. — 1987. — N 5. — С. 7–11.
6. Пухов Г.Е. Преобразования Тейлора и их применение в электротехнике и электронике. — Киев: Наукова думка, 1978. — 260 с.
7. Пухов Г.Е. Дифференциальные преобразования функций и уравнений. — Киев: Наукова думка, 1980. — 420 с.
8. Саух С.Е. Локальные интегро-дифференциальные преобразования одномерных функций и уравнений. — Киев, 1988. — 53 с. (Препринт / АН УССР. Ин-т проблем моделирования в энергетике; N 142 ).
9. Саух С.Е. Многомерные локальные интегро-дифференциальные преобразования функций и дифференциальных уравнений. — Киев, 1990. — 53 с. (Препринт / АН УССР. Ин-т проблем моделирования в энергетике; 90–34).
10. Береговенко Г.Я., Пухов Г.Е., Саух С.Е. Численные операторные методы решения дифференциальных уравнений и анализа динамических систем. — Киев: Наукова думка, 1992. — 262 с.
11. Береговенко Г.Я., Гершгорин А.Е., Саух С.Е. Моделирование газотранспортных систем численными операторными методами (прямые и обратные задачи). — Киев, 1992. — 44 с. (Препринт / АН УССР. Ин-т проблем моделирования в энергетике; 92–53 ).
12. Саух С.Е. Численные операторные методы, основанные на рядах Ньютона // Электронное моделирование. — 1996. — N 4. — С. 63–69.

## ОСНОВЫ ПОСТРОЕНИЯ ПАРАМЕТРИЧЕСКИХ МОДЕЛЕЙ ПЕТРИ КОММУТИРУЕМЫХ СЕТЕЙ

Д.А. Зайцев

Одесская национальная академия связи

Т.Р. Шмелёва

Одесская дирекция одесской железной дороги

### **Abstract**

*The technique of construction of parametric models of switched telecommunication networks in the form of colored Petri nets was presented. The standard model is invariant with respect to network's topology and consists of a fixed number of elements. The components of the modeled network are workstations, servers, switches and hubs.*

### **Введение**

Раскрашенные сети Петри [1] представляют собой универсальный графический язык для спецификации и анализа асинхронных параллельных систем и процессов. А промышленные моделирующие системы, такие как CPN Tools, Design/CPN, Miss-RdP [2], создают удобную среду для реализации технологий управляемого моделью (model driven) проектирования сложных систем.

При проектировании телекоммуникационных сетей распределенных систем технологического управления возникает задача оперативной оценки временных характеристик. Для этих целей предложено использовать модели коммутируемых сетей [3], собранные из компонентов, представляющих рабочие станции, серверы, коммутаторы и концентраторы [4,5]. Для измерения характеристик моделей разработаны специальные измерительные фрагменты сетей Петри [6,7]. При этом каждая новая структурная схема моделируемой сети требует повторного ввода главной страницы модели, что существенно снижает оперативность выполнения оценок. В [8] представлена параметрическая модель одноуровневой коммутируемой сети топологии "звезда". Однако крупные телекоммуникационные сети, как правило, имеют более сложные топологии, в большинстве случаев древовидной структуры.

Целью настоящей работы является построение типовой параметрической модели коммутируемой сети с произвольным количеством уровней, а также формулирование основных принципов организации параметрических моделей.

### **Коммутируемые сети**

Построение типовой модели выполним для коммутируемой Ethernet [3]. Основным элементом сети является коммутатор, направляющий прибывший фрейм в порт, к которому подключено устройство назначения. Для адресации устройств используются 6-байтные MAC-адреса (Media Access Control).

Информация о подключении устройств к портам задаётся таблицей коммутации, которая может быть либо статической, либо динамической. Влияние процессов динамического ведения таблиц коммутации на пропускную способность сети изучено в [5], поэтому в настоящей работе рассмотрим статические таблицы.

Среди терминальных устройств по особенностям формирования трафика будем различать рабочие станции и серверы. В качестве протокола взаимодействия рассмотрим двухфазный процесс "запрос-ответ". Интервалы между запросами, времена их исполнения, а также объёмы передаваемой информации будем оценивать случайными равномерно распределёнными функциями. Методика расчёта временных и емкостных параметров модели для заданной спецификации программного обеспечения и оборудования описана в [6,7].

Пример двухуровневой коммутируемой сети представлен на Рис. 1; сеть содержит три коммутатора (SWI), шесть рабочих станций (WS) и два сервера (S).

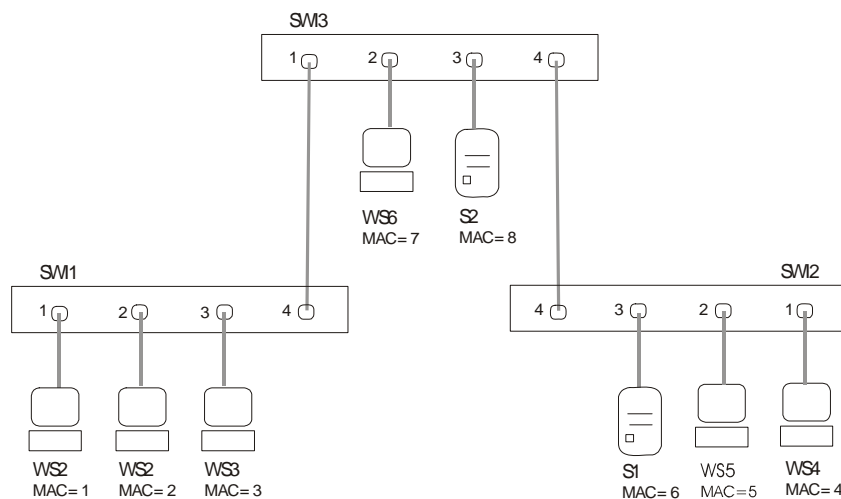


Рисунок 1 — Коммутируемая сеть диспетчерского центра железной дороги

### **Типовая параметрическая модель**

Типовая параметрическая модель коммутируемой Ethernet представлена на Рис. 2. Описания типов данных (цветов), переменных и функций приведены на Рис. 3.



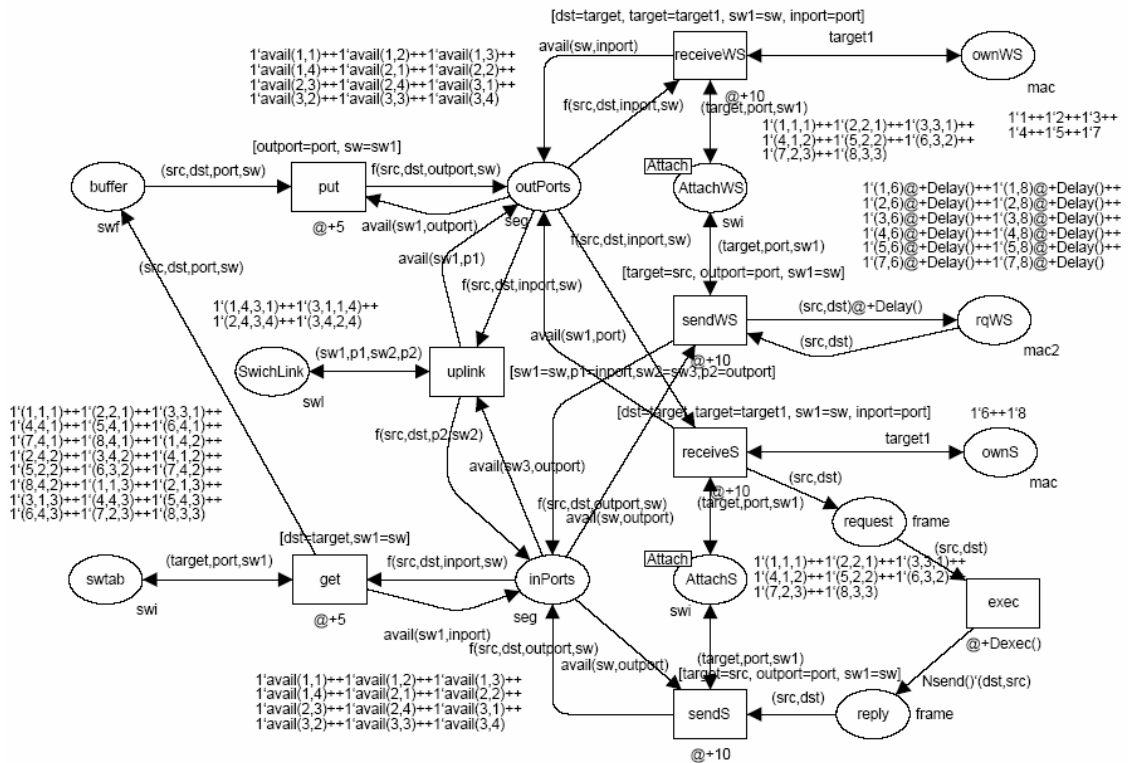


Рисунок 2 — Параметрическая модель коммутируемой сети

```

color mac=INT;
color portnum=INT;
color swch=INT;
color mac2=product mac*mac timed;
color nfrm=INT;
color sfrm=product nfrm*INT timed;
color frm=product mac*mac*portnum*swch timed;
color nseg=product swch*portnum;
color seg=union f:frm+avail:nseg timed;
color swi=product mac*portnum*swch;
color swf=product mac*mac*portnum*swch timed;
color frame=product mac*nfrm timed;
color swl=product swch*portnum*swch*portnum;
var src,dst,target,target1: mac;
var sw,sw1,sw2,sw3:swch;
var port, inport, output: portnum;
var p1, p2: portnum;
color dex= int with 100..200;
fun Dexec()=dex.ran();
color nse = int with 10..20;
fun Nsend()=nse.ran();
color Delta= int with 1000..2000;
fun Delay()=Delta.ran();

```

Рисунок 3 — Описания типов, переменных и функций

Элементы серверов содержат суффикс *S*, элементы рабочих станций — суффикс *WS*, элементы коммутаторов не имеют суффикса. Все сегменты локальной сети представлены парой позиций *inPorts*, *outPorts*. Для моделирования передачи фреймов от терминальных устройств к коммутатору используется позиция *inPorts*; для моделирования передачи фреймов от коммутаторов к терминальным устройствам — позиция *outPorts*. Исключения составляют связи между коммутаторами. Для представления таких связей использован переход *uplink*.

При моделировании трафика мы абстрагируемся от содержимого фреймов, рассматривая только адреса отправителя и получателя. Ключевым средством параметризации является абстрактный тэг, ассоциированный с объектами модели. Тэг содержит номер коммутатора и номер порта, идентифицируя, таким образом, сегмент сети. Каждый из сегментов может быть либо свободен, что моделируется специальной меткой *avail*, либо занят передачей фрейма *f*. Перед передачей фрейма каждое из устройств ожидает освобождения необходимого сегмента, проверяя наличие соответствующей метки *avail*. При передаче метка *avail* извлекается и замещается фреймом. Приём фрейма представляет собой обратный процесс. Как правило, принимающее устройство проверяет адрес назначения *dst*, извлекает фрейм и замещает его соответствующей меткой *avail*.

Топология сети представлена маркировкой позиций *Attach* и *SwitchLink*. Совмещённая позиция *Attach* (*AttachWS*, *AttachS*) содержит информацию о подключении терминальных устройств, указывая для каждого MAC-адреса номер коммутатора и номер порта. Заметим, что такой способ представления позволяет моделировать несколько терминальных устройств, подключённых к одному порту коммутатора, например, посредством концентратора. Позиция *SwitchLink* содержит описания связей между коммутаторами, указывая номер коммутатора и номер порта для каждого из двух концов линии связи.

Таблицы коммутации всех коммутаторов представлены маркировкой позиции *swtab*. Отметим, что дублирование информации связано с тем, что в общем случае каждому коммутатору должен быть известен MAC-адрес каждого из устройств сети. Позиция *buffer* моделирует внутренний буфер коммутатора.

Позиция *rqWS* содержит таблицу запросов рабочих станций к серверам, заданную указанием пар MAC-адресов. После выполнения запроса пара адресов блокируется на интервал времени, заданный функцией *Delay()*. Позиция *ownWS* перечисляет MAC-адреса рабочих станций. Сервер размещает полученный запрос в позиции *request*, выполняет его в течение времени *Dexec()* и формирует *Nsend()* фреймов ответа в позиции *reply*, которые затем передаются в обратном направ-

лении к запросившей рабочей станции, что моделируется изменением порядка MAC-адресов отправителя и получателя в заголовке фреймов.

Для измерения характеристик функционирования локальной сети построенная модель может быть дополнена измерительными фрагментами, описанными в [7,8].

### Имитация динамики модели

Для отладки модели система CPN Tools предоставляет режимы имитации динамики сети с визуальным отображением всех событий, происходящих в текущий момент модельного времени. Обеспечивается пошаговая имитация либо имитация заданного количества шагов. Под шагом подразумевается момент модельного времени, для которого имеется непустой список событий.

Рассмотрим процесс продвижения фрейма запроса от одной из рабочих станций к серверу, а также доставки одного из фреймов ответа сервера в обратном направлении. На Рис. 4 представлено начальное состояние модели. Отметим, что в соответствии с маркировкой позиции *rqWS*, сгенерированной с помощью случайной функции *Delay()*, начальный момент модельного времени равен 1063.

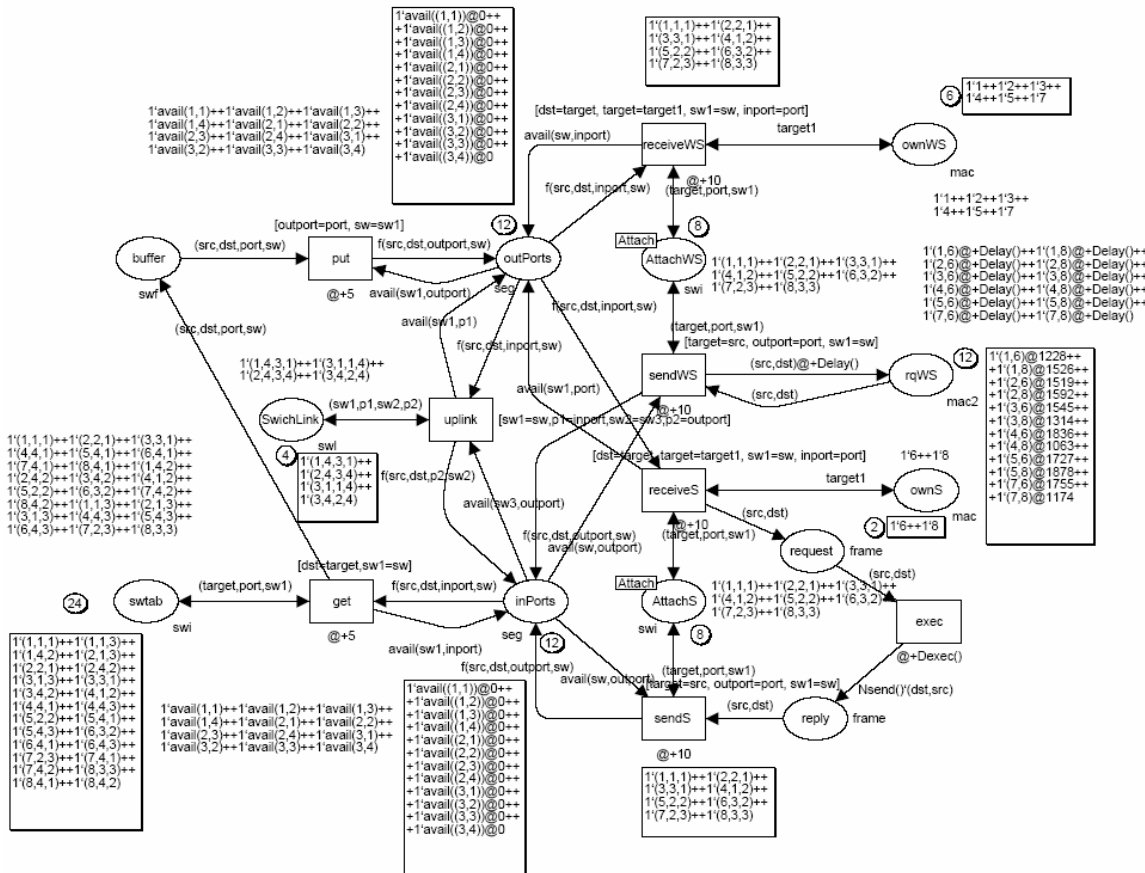


Рисунок 4 — Начальное состояние модели

Для компактного представления процесса имитации используем трассировочную таблицу (Табл. 1), в которой отображены состояния модели на каждом шаге. Заметим, что кроме указанных в таблице позиций изменяется также маркировка позиции  $rqWS$ ; так как эти изменения связаны только с временными параметрами, рассмотрим состояние указанной позиции после выполнения 32 шагов (Рис. 5).

Таблица 1 — Трассировка динамики модели

Шаг	Время	Маркировка позиций				
		InPorts	OutPorts	Buffer	request	reply
0	1063	–	–			
1	1073	l`f((4,8,1,2))@1073	–			
2	1078	l`avail((2,1))@1078	–	l`(4,8,4,2))@1078		
3	1083	–	l`f((4,8,4,2))@1083			
4	1083	l`f((4,8,4,3))@1083	l`avail((2,4))@1083			
5	1088	l`avail((3,4))@1088	–	l`(4,8,3,3))@1088		
6	1093	–	l`f((4,8,3,3))@1093			
7	1103	–	l`avail((3,3))@1103		l`(4,8))@1103	
8	1174	–	–			20`(8,4))@1244
9	1184	l`f((7,8,2,3))@1184	–			–
10	1189	l`avail((3,2))@1189	–	l`(7,8,3,3))@1189		–
11	1194	–	l`f((7,8,3,3))@1194			–
12	1204	–	l`avail((3,3))@1204		l`(7,8))@1204	–
13	1228	–	–			20`(8,4))@1244++ 16`(8,7))@1362
14	1238	l`f((1,6,1,1))@1238	–			–
15	1243	l`avail((1,1))@1243	–	l`(1,6,4,1))@1243		–
16	1244	–	l`f((1,6,4,1))@1248			–
17	1248	l`f((8,4,3,3))@1254	l`f((1,6,4,1))@1248			19`(8,4))@1244++ 16`(8,7))@1362
18	1248	l`f((1,6,1,3))@1248++ l`f((8,4,3,3))@1254	l`f((1,6,4,1))@1248			–
19	1253	l`f((8,4,3,3))@1254++ l`avail((3,1))@1253	l`avail((1,4))@1258	l`(1,6,4,3))@1253		–
20	1254	l`f((8,4,3,3))@1254	l`f((1,6,4,3))@1258			–
21	1258	l`avail((3,3))@1259	l`f((1,6,4,3))@1258	l`(8,4,4,3))@1259		–
22	1258	l`f((1,6,4,2))@1258	l`avail((3,4))@1258	l`(8,4,4,3))@1259		–
23	1259	l`avail((2,4))@1263	–	l`(1,6,3,2))@1263++ l`(8,4,4,3))@1259		–
24	1259	l`f((8,4,3,3))@1269	–	l`(1,6,3,2))@1263++ l`(8,4,4,3))@1259		18`(8,4))@1244++ 16`(8,7))@1362
25	1263	l`f((8,4,3,3))@1269	l`f((8,4,4,3))@1264	l`(1,6,3,2))@1263		–
26	1264	l`f((8,4,3,3))@1269	l`f((1,6,3,2))@1268++ l`f((8,4,4,3))@1264			–
27	1264	l`f((8,4,3,3))@1269++ l`f((8,4,4,2))@1264	l`f((1,6,3,2))@1268++ l`avail((3,4))@1264			–
28	1268	l`f((8,4,3,3))@1269++ l`f((8,4,4,2))@1264	l`f((1,6,3,2))@1268	l`(8,4,1,2))@1269		–
29	1269	l`f((8,4,3,3))@1269++ l`f((8,4,4,2))@1264	l`avail((2,3))@1278	l`(8,4,1,2))@1269	l`(1,6))@1278	–
30	1269	l`f((8,4,3,3))@1269++ l`avail((2,4))@1269	l`f((8,4,1,2))@1274		l`(1,6))@1278	–
31	1274	l`avail((3,3))@1274	l`f((8,4,1,2))@1274	l`(8,4,4,3))@1274	l`(1,6))@1278	–
32	1274	–	l`avail((2,1))@1284	l`(8,4,4,3))@1274	l`(1,6))@1278	–

Запрос рабочей станции WS4, отправленный на шаге 1 серверу S2, доставлен серверу на 7-м шаге и выполнен сервером на 8-м шаге, в результате чего сгенерированы 20 фреймов ответа. Начиная с 17-го шага выполняется доставка фреймов ответа в обратном направлении. На 32-м шаге первый из фреймов ответа возвращается к рабочей станции WS4 и поглощается переходом *receiveWS*.

Следует отметить, что Табл. 1 отображает также другие происходящие в модели события, например, выполнение запроса рабочей станции WS6, отправленного на 9-м шаге.

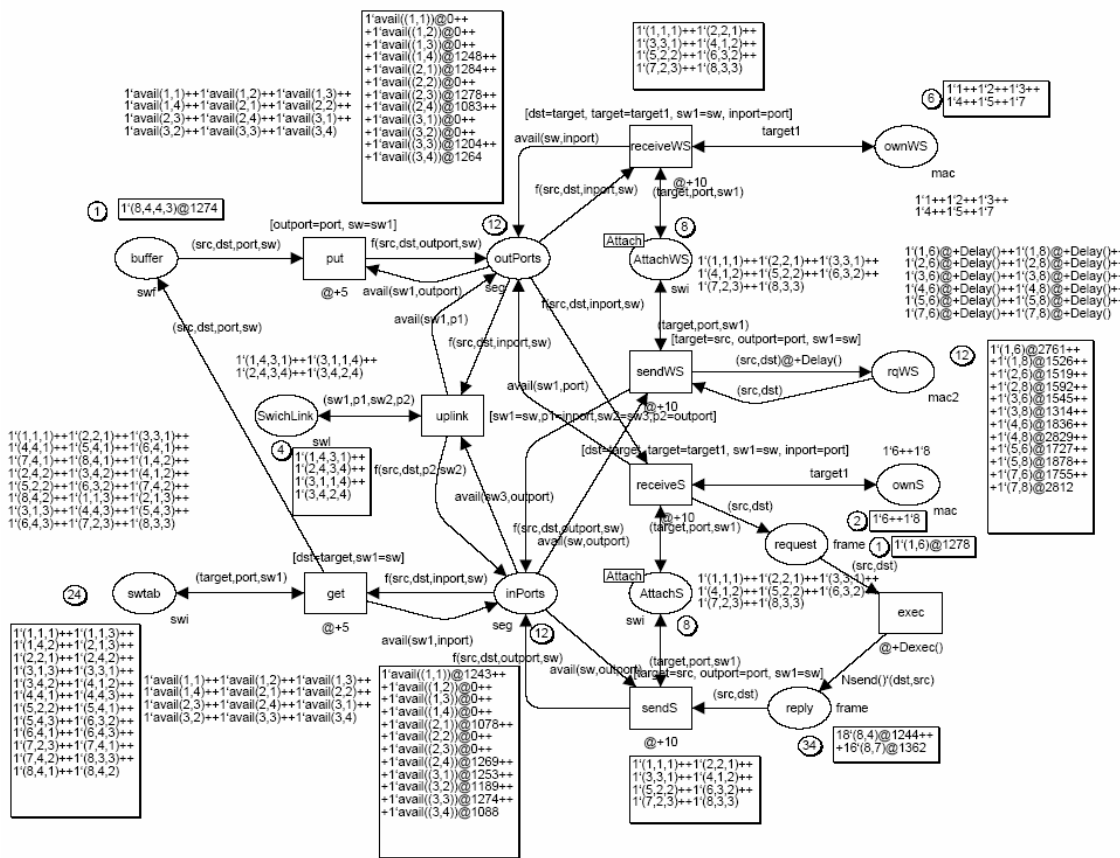


Рисунок 5 — Состояние модели на шаге 32

Для измерения характеристик функционирования коммутируемых сетей целесообразно применять режим скоростного моделирования, обеспечивающий выполнение заданного числа шагов без визуализации динамики модели. При этом отображается лишь конечное состояние модели. Скоростная имитация позволяет накапливать статистическую информацию на длительных интервалах реального времени моделируемых сетей, обеспечивающих наблюдение стационарного режима функционирования модели. Вопросы измерения характеристик моделей изучены в [7].

### ***Принципы параметризации моделей***

Построение параметрических моделей возможно для сетей, собранных из фиксированного множества типовых элементов. Каждый типовой элемент представляется в единственном экземпляре. Вводятся специальные позиции, маркировка которых описывает топологию сети. Динамические элементы, представленные фишками, дополняются специальными тэгами, описывающими местонахождение фишки по отношению к топологической информации.

### ***Заключение***

Таким образом, в настоящей работе представлена методика параметризации моделей телекоммуникационных сетей, построенных на основе фиксированного множества типовых элементов. Методика изучена на примере построения параметрической модели Петри коммутируемой Ethernet. Преимуществом параметрических моделей является фиксированная структура для произвольной заданной конфигурации телекоммуникационной сети.

### ***Литература***

1. Jensen K. Colored Petri Nets — Basic Concepts, Analysis Methods and Practical Use. — Vol. 1–3, Springer-Verlag, 1997.
2. Beaudouin-Lafon M., Mackay W.E., Jensen M. et al. CPN Tools: A Tool for Editing and Simulating Coloured Petri Nets. LNCS 2031: Tools and Algorithms for the Construction and Analysis of Systems, 2001, 574–580. (<http://www.daimi.au.dk/CPNTools>).
3. Breyer R., Riley S. Switched, fast, and gigabit Ethernet, MacMillan Technical Pub., 1999, 618 p.
4. Zaitsev D.A. Switched LAN simulation by colored Petri nets. Mathematics and Computers in Simulation, vol. 65, no. 3, 2004, 245–249.
5. Зайцев Д.А., Шмельёва Т.Р. Моделирование коммутируемой локальной сети раскрашенными сетями Петри // Зв'язок, № 2(46), 2004, с. 56–60.
6. Zaitsev D.A. An Evaluation of Network Response Time using a Coloured Petri Net Model of Switched LAN // Proceedings of Fifth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, October 8–11, 2005, Aarhus, Denmark, p. 157–167.
7. Зайцев Д.А. Измерительные фрагменты в моделях Петри телекоммуникационных сетей // Зв'язок, № 2(54), 2005, с. 65–71.
8. Зайцев Д.А., Шмельёва Т.Р. Параметрическая модель Петри одноуровневой коммутируемой сети // Труды Одесской национальной академии связи им. А.С.Попова, № 1, 2005.

УДК 381.3

## СЕМІОТИЧНА МОДЕЛЬ УЗАГАЛЬНЕНОГО ГЕНЕТИЧНОГО АЛГОРИТМУ ДЛЯ ПРОЕКТУВАННЯ ДИНАМІЧНИХ ОБ'ЄКТІВ

Г.О.Селевко

Донецький національний технічний університет

### **Анотація**

*У роботі розкрито семіотичну модель узагальненого генетичного алгоритму для проектування динамічних об'єктів (СМ УГА ПДО). Зроблено можливість вирішити проблему попередньої збіжності і розв'язати задачу узагальнення цільової функції ГА.*

В даний час генетичні алгоритми (ГА) є добре відомою і ефективною методологією з оптимізації, яку використовують для розв'язання різних задач. Ця методологія основана на аналогії процесів натуральної селекції і генетичних перетворень у біології. ГА широко використовуються як ефективний пошук у задачах штучного інтелекту, оптимізації структур, вирішення нелінійних рівнянь, оптимізації перевезень, вирішення різних проблем розкладів і т.і. В останній час почались розробки і дослідження можливостей застосування ГА для ефективного розв'язання задач проектування інтелектуальних систем [1–6]. Необхідно підкреслити, що задачі генетичного алгоритму з точки зору структури є відкритими і динамічними. Наряду з ефективним механізмом оптимізації у ГА дослідники виділяють наступні головні складнощі у реалізації ГА [2–5]: попередня збіжність рішення ГА у локальному оптимумі (попадання у локальну “яму”); відсутність узагальненої схеми побудови цільової функції ГА.

Автор у даній роботі ставить мету розробити механізм ГА, який по-перше, узагальнює процес побудови цільової функції генетичного алгоритму, і по-друге, допомагає вирішити проблему попередньої збіжності рішення генетичного алгоритму у локальному оптимумі.

Простий генетичний алгоритм (ПГА) має наступну ідею: пошук оптимального рішення задачі за допомогою постійного еволюціонування рішення. Еволюція рішення задачі відбувається за допомогою взаємодії таких елементів ГА як генетичні оператори (ГО) і цільова функція (ЦФ). Отже, для ефективного застосування ПГА для задач проектування необхідні механізми, які вирішують проблему попередньої збіжності ГА у локальному оптимумі і розв'язують задачі побудови ЦФ. При розв'язанні задач проектування, задач, що складно формалізуються і для вирішення перелічених вище проблем

ГА пропонується застосовувати узагальнений ГА для проектування динамічних об'єктів (УГА ПДО). Головні моменти УГА ПДО, які відрізняють його від ПГА, наступні:

1) першим етапом, який передує генерації початкової популяції є формування структур умов задачі (побудова структур даних, визначення генів ГА, значень генів, системи кодування, структури взаємодії генів і їх значень, структури обмежень);

2) два методи генерації початкової популяції;

3) формування архіву оптимальних (квазіоптимальних) рішень після кожної генерації;

4) використання блоку запобіжності попередньої збіжності (БЗПЗ), який виконує функцію аналізу оптимального рішення і прийняття заходів для виходу з локальних „ям”;

5) схема узагальненого розрахунку ЦФ на базі обмежень і показників взаємодії генів ГА і їх значень.

На рисунку 1 наведено базисну структуру УГА ПДО.

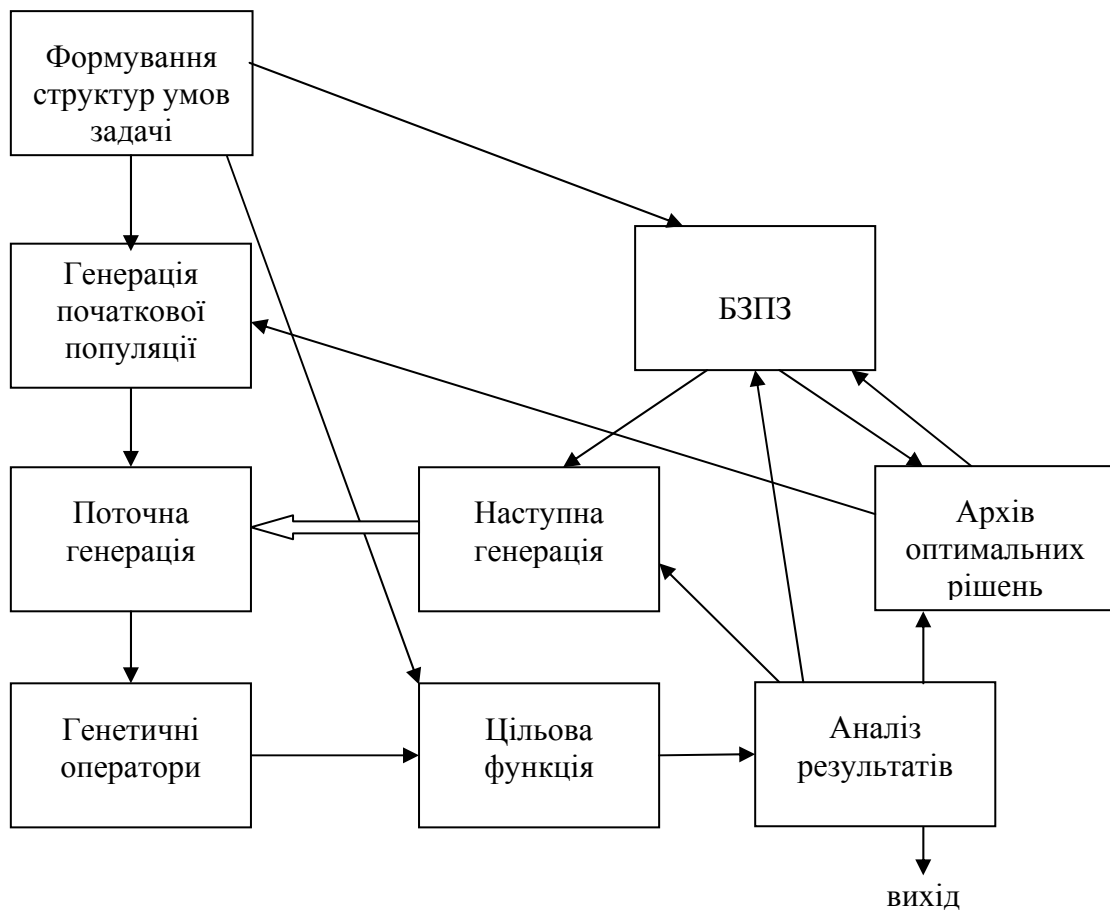


Рисунок 1 — Базисна структура УГА ПДО



УГА ПДО допомагає вирішити поточні труднощі при реалізації ГА, розширює можливості ГА, але стає проблема формалізації і автоматизації такого відкритого і динамічного засобу оптимізації як УГА ПДО.

Одною з сучасних і універсальних моделей формалізації є семіотична модель (СМ), а завдяки своїм механізмам зміни структур і форм елементів дає змогу формалізувати динамічні задачі, які важко формалізувати [7–12]. Семіотична модель має наступний вигляд:

$$СМ = (Т, Р, А, П, \chi_T, \chi_P, \chi_A, \chi_P) \quad (1)$$

Розглянемо детальніше кожен елемент семіотичної моделі. Сигнатура (Т) семіотичної моделі включає безліч базових елементів моделі. Синтаксичні правила Р застосовуються для того, щоб з базових елементів побудувати такі сукупності, які у рамках даної системи є синтаксично вірними сукупностями. Аксиоми А містять безліч тверджень і положень системи, які є семантично вірними. Семантичні правила П (правила виводу) дають можливість довести, що деякі синтаксично вірні сукупності Р є і семантично вірними сукупностями. Правила виводу П розширюють, якщо це можливо, безліч аксіом А.

Пропонується застосовувати семіотичну модель для формалізації узагальненого генетичного алгоритму для проектування динамічних об'єктів. СМ УГА ПДО дає можливість розглядати генетичні алгоритми не просто як метод оптимізації, а як комплексний механізм оптимізації для розв'язання будь-якого класу задач. СМ УГА ПДО має вигляд, який наведено у формулі (1). У рамках СМ УГА ПДО сигнатура — є елементом, який „зберігає” усі умови задачі. У сигнатурі визначається ген ГА, механізм кодування. Елементами сигнатури СМ УГА ПДО є:

- Структура  $S_j$ , яка має поля  $V_k$  та записи  $Z_l$ ;
- Сукупність генів  $g_i$ ;
- Сукупність значень генів  $Zn_q$ ;
- Структура взаємодії генів і їх значень  $Sxg$ .

Схематично елементи сигнатури СМ УГА ПДО і їх взаємодії наведено на рисунку 2.

Правила зміни сигнатури  $\chi_T$  СМ ГА:

- 1) Побудова нової структури  $S_j$  ;
- 2) Побудова нового поля  $V_k$  структури  $S_j$  ;
- 3) Побудова нового запису  $Z_l$  структури  $S_j$  ;
- 4) Визначення сукупності генів  $g_i$ ;
- 5) Визначення сукупності значень генів  $Zn_q$ ;
- 6) Побудова структури взаємодії генів і їх значень  $Sxg$ .

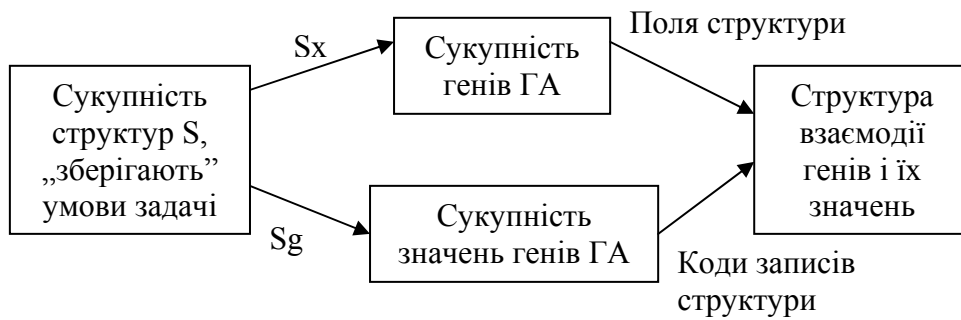


Рисунок 2 — Елементи сигнатури СМ УГА ПДО і їх взаємодія

Відповідно правил 4) и 5) зміни сигнатур необхідно зауважити, що сукупності генів  $g_i$  і їх значень  $Zn_q$  визначаються з сукупності структур, які є елементами сигнатури. Тобто відповідно задачі оптимізації визначається структура  $Sx$ , яка формує сукупність генів хромосоми  $g_i$ , і структура  $Sg$ , яка формує сукупність значень генів  $Zn_q$ :

$$\begin{aligned} \{g_i\} &\rightarrow Sx \\ \{Zn_q\} &\rightarrow Sg \end{aligned}$$

Організація сигнатури таким чином дає можливість умовам задачі ГА впливати не тільки на формування початкової популяції, а і на розрахунок цільової функції ГА та на роботу блоку запобігання попередньої збіжності. Також розкривається можливість для однієї моделі розв'язувати різні види задач оптимізації, змінюючи елементи пошуку — гени та аллелі — значення гену.

Граматиною  $P$  СМ УГА ПДО є популяція рішень  $R$  моделі, яка формується з сукупності хромосом  $X_m$ :  $R = \{X_m\}$ . Хромосома є текстом граматики семіотичної моделі, а з точки зору теорії генетичних алгоритмів є закодованим рішенням моделі — об'єктом оптимізації. Отже, сукупність генів моделі складають хромосому:  $X = \{g_i\}$ . Правила зміни граматики  $\chi_p$  є правилами побудови нової популяції. Відповідно теорії ГА нова популяція будується за допомогою генетичних операторів (ГО), але початкова популяція є умовою задачі, тобто формується відповідно даних сигнатури СМ УГА ПДО. Відповідно структурі УГА ПДО початкова популяція може формуватися не тільки за умовами задачі, а також під впливом БЗПЗ з аксіом моделі. Отже, правилами зміни граматики  $P$  є:

- 1) Алгоритм побудови нової популяції за допомогою ГО;
- 2) Формування випадкової початкової популяції;
- 3) Формування початкової популяції на базі існуючих рішень задачі (з сигнатури, аксіом).

Аксиомами СМ УГА ПДО є архів рішень моделі (хромосом), які на даному етапі були оптимальним (квазіоптимальним) рішенням поставленої задачі. Правила зміни аксіом  $\chi_A$  відкривають можливість додавання нових рішень і відкидання тих, що стали не актуальними.

Для реалізації усіх можливостей УГА ПДО в СМ УГА ПДО необхідна чітка і налагоджена структура правил виводу. На правила виводу покладається один з тих елементів ГА, що складно формалізувати і узагальнити — розрахунок ЦФ. Механізм СМ дає інструменти для динамічної формалізації ЦФ. Крім формування ЦФ правила виводу СМ УГА ПДО оцінюють відповідність нових рішень (хромосом) обмеженням моделі. Обмеження є умовою задачі, тобто є складовою сигнатури. Правила виводу реалізують алгоритм перевірки відповідності нових рішень обмеженням задачі і розрахунку коефіцієнту обмеження  $k$  для ГА. Цільова функція популяції  $F$  це сукупність цільових функцій кожної хромосоми популяції:  $F = \{f_y\}$ . Цільова функція розраховується як ефективність взаємодії генів і їх значень з урахуванням коефіцієнту обмеження. В напрямок дії правил виводу СМ УГА ПДО входить і організація роботи БЗПЗ. Правила виводу СМ УГА ПДО можна сформулювати наступним чином:

1) розрахунок пристосованості хромосом популяції (відповідність рішення задачі обмеженням моделі та цільова функцію  $F$  популяції ГА);

2) організація роботи БЗПЗ (керування формуванням наступної генерації, аналіз аксіом і результатів оптимізації).

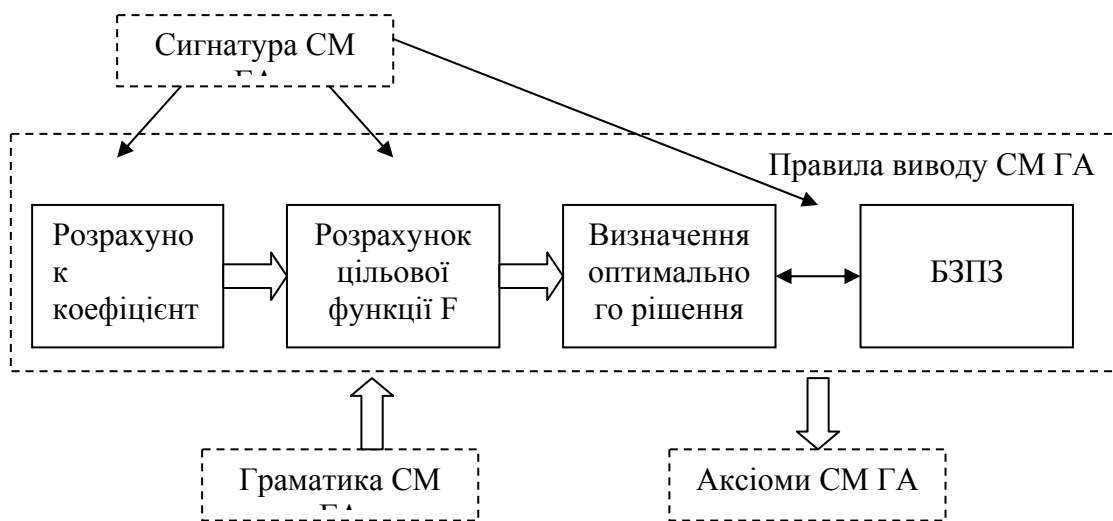


Рисунок 3 — Схема взаємодії складових Правил виводу і інших елементів СМ УГА ПДО

Правила зміни правил виводу  $\chi_{\Pi}$  дають можливість змінити параметри розрахунку цільової функції, доповнити роботу БЗПЗ.

Отже, було запропоновано комплексний механізм СМ УГА ПДО, який розкриває нові можливості генетичному алгоритму для розв'язання задач оптимізації проектування. Відмітні характеристики запропонованого підходу від ПГА наступні:

1) вплив умов задачі на реалізацію ГА (побудова і зміна структур даних, визначення генів ГА і значень генів перед початком виконання РГА, автоматизація системи кодування, формування структури взаємодії генів і їх значень, визначення структури обмежень);

2) можливість моделювання УГА ПДО для різних початкових популяцій;

3) застосування архіву оптимальних (квазіоптимальних) рішень для керування розвитком ГА;

4) використання БЗПЗ, який виконує функцію аналізу оптимального рішення і прийняття заходів для виходу з локальних „ям”;

5) застосування механізму узагальнення розрахунку ЦФ на базі обмежень і показників взаємодії генів ГА і їх значень.

Запропонований підхід СМ УГА ПДО дає можливість адаптувати механізм ГА для задач, що складно формалізувати. СМ УГА ПДО підвищує ефективність ПГА завдяки своїй структурі, що забезпечує гнучкість застосування алгоритму для різного класу задач. У свою чергу гнучкість СМ УГА ПДО і узагальнення ЦФ обумовлюють успішну адаптацію Га до задач проектування, для яких характерна динамічність моделі, змінв обмежень, напрямків оптимізації. Підвищується ефективність рішення завдяки механізму СМ, тобто реалізована постійна взаємодія всіх складових УГА ПДО і вплив на швидкість і ефективність пошуку та вирішення проблеми виходу з локальних “ям”.

### *Література*

1. Вороновский Г.К Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. — Х.: ОСНОВА, 1997.
2. Holland J. H. Adaptation in natural and artificial systems. An introductory analysis with application to biology, control and artificial intelligence. — London: Bradford book edition, 1994.
3. Wong F., Yong Tan P. Neural Networks And Genetic Algorithm For Economic Forecasting //AI in economics and business administration. Institute of systems Science, National University of Singapore 1999.

4. Люгер, Джорж, Ф. Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание.: Пер с англ. — М.: Издательский дом "Вильямс", 2003. — 864с.
5. Курейчик В.М. Генетические алгоритмы и их применение. Таганрог: Изд-во ТРТУ, издание второе, дополненное. — 2002. — 242с.
6. Григорьев А.В., Селевко А.А. Использование генетических алгоритмов в задачах кадрового менеджмента //Труды Международной научно-технической конференции «Интеллектуальные системы» (IEEE AIS'03) и «Интеллектуальные САПР» (CAD-2003). Научное издание в 3-х томах. Г.: Издательство Физико-математической литературы, — 2003, Т.1. — 612 с.
7. Поспелов Д.А. Ситуационное управление: теория и практика. — М.: Наука, 1986. — 288с.
8. Головина Е.Ю., Модель представления знаний в семиотической системе, <http://www.inftech.webservis.ru/it/conference/scm/2000/session10/golovina.htm>
9. Григорьев А.В. Семиотическая модель базы знаний САПР // Научные труды Донецкого государственного технического университета Серия: Проблемы моделирования и автоматизации проектирования динамических систем. — Донецк: ДонГТУ, 1999. — Выпуск 10. — С. 30–37.
10. Kecheng Liu, Lily Sun, Joseph Barjis, Jan L.G. Dietz Modelling dynamic behaviour of business organisations-extension of DEMO from a semiotic perspective //Knowledge-Based Systems № 16, 2003. — С. 101–111.
11. Francois Cooren, Robert E. Sanders Implicatures: a schematic approach //Journal of Pragmatics №34, 2002. — С. 1045–1067.
12. А.В. Григорьев, А.А. Селевко Семиотическая модель производства как основа для решения задачи оптимального распределения трудовых ресурсов между операциями технологического процесса //Серия «Проблемы моделирования и автоматизации проектирования динамических систем» (МАП -2004). Выпуск 71: — Донецк: ДонНТУ, 2004. — С. 80–85.

## АЛГОРИТМ АВТОМАТИЗИРОВАННОГО ПОСТРОЕНИЯ БАЗЫ ЗНАНИЙ НА ОСНОВЕ АНАЛИЗА ЕСТЕСТВЕННО-ЯЗЫКОВЫХ МЕДИЦИНСКИХ ТЕКСТОВ

И.А. Коломойцева  
Кафедра ПМиИ, ДонНТУ

### *Анотація*

*У статті розглядаються алгоритми синтаксичного і морфологічного аналізу природно-мовних медичних текстів, які застосовуються а пошуково-аналітичній системі «Допоможи собі сам». Система працює з лексично необробленими текстовими базами даних.*

В любой экспертной системе можно выделить четыре основных функциональных блока:

- базу знаний;
- блок вывода заключений;
- блок "объяснения" вывода заключений;
- блок наполнения и редактирования базы знаний.

База знаний является важной частью экспертной системы, качество наполнения которой серьезно влияет на всю работу системы. Практические методы получения знаний можно разделить на два больших вида: коммуникативные и текстологические. Основной принцип деления связан с источником знаний. В первом случае таким источником является эксперт, а во втором — литература, документы, учебники [1]. Именно текстологический способ извлечения знаний может быть использован для создания автоматизированной системы построения базы знаний.

Процесс автоматизации принятия решения в экспертных системах невозможен без привлечения информации, которая не может быть выражена количественно. Это семантическая (смысловая) информация. Такую информацию возможно извлечь из естественно-языковых текстов. В случае медицинских экспертных систем знания извлекаются из специальных медицинских текстов, зафиксированных рассказов врачей-экспертов о различных проявлениях заболеваний, методов их лечения и т. п. [2, 3]

На протяжении последних двух-трех десятилетий многие исследователи, занимающиеся проблемами автоматической обработки текста и «понимания» естественного языка, получили ряд интересных результатов [4, 5, 6].

Анализ текста предполагает наличие нескольких фаз:

- 1) морфологический анализ;

- 2) синтаксический анализ;
- 3) семантический анализ;
- 4) прагматический анализ.

В данной статье представляется алгоритм обработки текстов с целью получения тематических знаний из естественно-языковых медицинских текстов.

Входными данными для анализа служат тексты, полученные в результате поиска в сети Интернет. Эти тексты сформированы системой поиска и имеют следующую структуру:

- название болезни;
- текст с описанием болезни.

Исходными данными для анализа могут также быть и тексты с описанием болезней, предоставленные самим пользователем. На файл пользователя накладываются следующие ограничения:

- соблюдение структуры: название болезни — текст с описанием болезни;
- в одном файле описание одной болезни.

Анализ естественно-языкового медицинского текста предполагает разбор на симптомы, характерные для выбранной болезни, и методы возможного лечения.

Все естественно-языковые медицинские тексты, в которых содержится описание болезней, имеют общую логическую структуру, приведенную на рисунке 1.

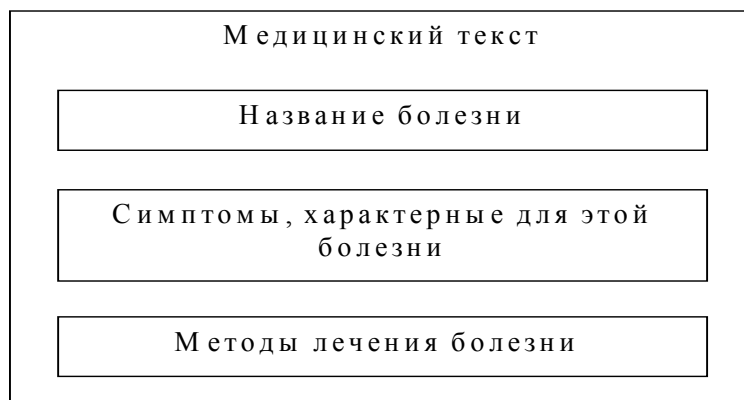


Рисунок 1 — Общая схема логической структуры естественно-языкового медицинского текста, содержащего описание болезни

Текст такой структуры возможно обработать по следующему алгоритму.

Алгоритм анализа естественно-языкового медицинского текста.

1. Получить входной текст.

2. Выделяем из текста заголовок — название болезни.
3. Выбор предложения из текста.
4. Если в предложении описываются симптомы болезни, то найти в предложении описание симптомов и занести их в базу знаний.
5. Если в предложении описывается возможное лечение болезни, то найти в предложении описание возможного лечения и занести его в базу знаний.

Блок-схема алгоритма анализа естественно-языкового медицинского текста приведена на рисунке 2.

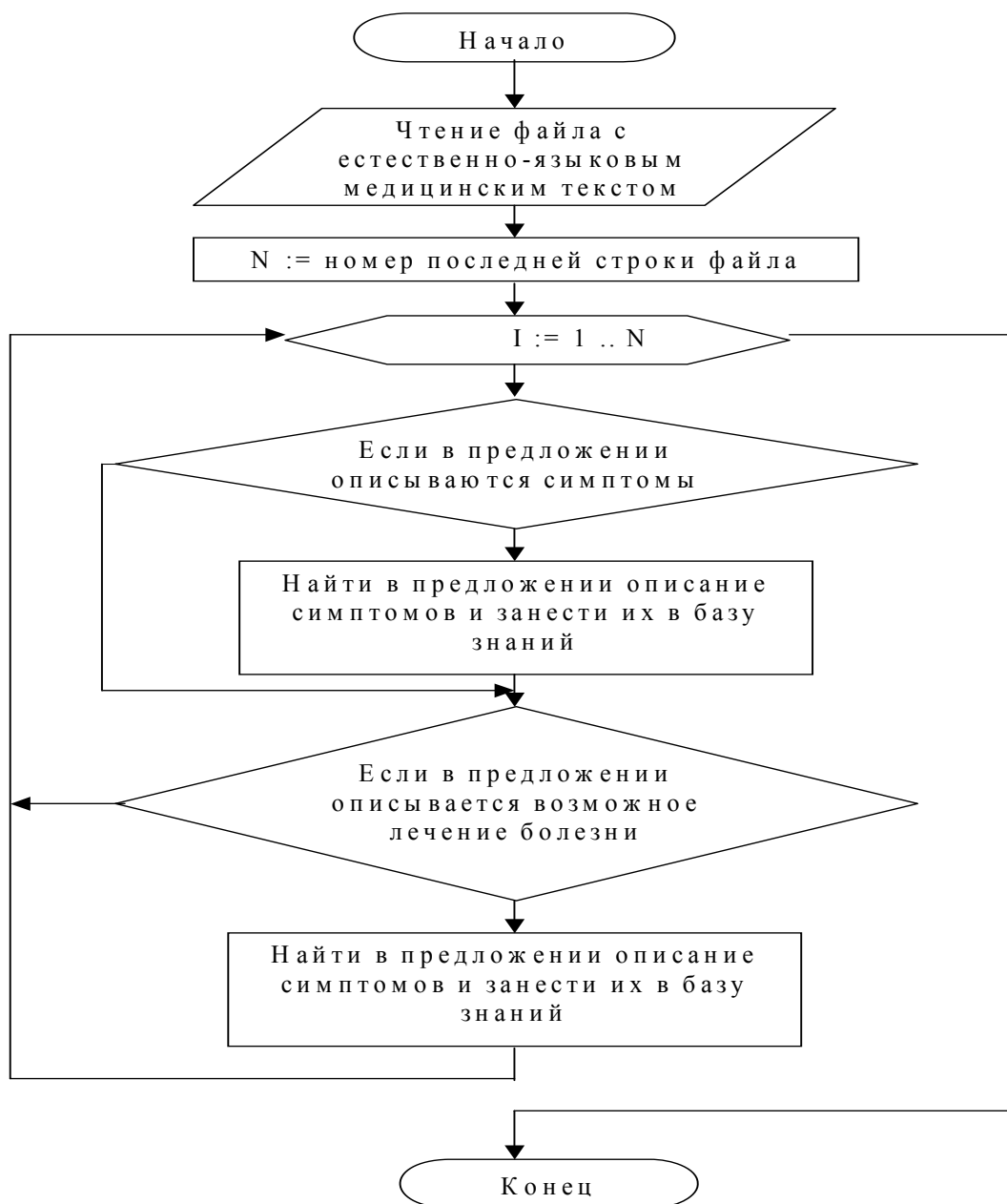


Рисунок 2 — Блок-схема алгоритма анализа естественно-языкового медицинского текста.



Выделение симптомов болезни осуществляется методом поиска в предложениях характерных конструкций для симптомов.

Таковыми характерными конструкциями могут выступать следующие слова:

- симптомы;
- характерно;
- диагностика;
- наблюдается;
- заметно;
- появляются;
- другие.

Алгоритм выделения симптомов представлен на рисунке 3.

ПОЛУЧИТЬ текст с описанием болезни  
ДЛЯ всех предложений в тексте ДЕЛАТЬ  
ДЛЯ всех характерных слов из словаря для симптомов ДЕЛАТЬ  
ЕСЛИ в предложении есть характерное слово ТО  
    Выделить из предложения перечисляемые элементы  
    КОНЕЦ\_ЕСЛИ  
КОНЕЦ\_ДЛЯ  
КОНЕЦ\_ДЛЯ

Рисунок 3 — Алгоритм выделения симптомов

Выделение описания лечения болезни осуществляется методом поиска в предложениях текста характерных конструкций для возможного лечения.

Таковыми характерными конструкциями могут выступать следующие слова:

- лечение;
- применяются;
- назначается;
- необходимо;
- другие.

Алгоритм выделения возможного лечения приведен на рисунке 4.

Результатом анализа естественно-языкового медицинского текста являются следующие полученные знания:

- 1) название болезни;
- 2) список симптомов;
- 3) список возможного лечения.

Полученные знания сохраняются в базу знаний. Логическая структура базы знаний представлена в таблице 1.

ПОЛУЧИТЬ текст с описанием болезни  
 ДЛЯ всех предложений в тексте ДЕЛАТЬ  
 ДЛЯ всех характерных слов из словаря для  
 возможного лечения ДЕЛАТЬ  
 ЕСЛИ в предложении есть характерное слово ТО  
 Выделить из предложения перечисляемые элементы  
 КОНЕЦ\_ЕСЛИ  
 КОНЕЦ\_ДЛЯ  
 КОНЕЦ\_ДЛЯ

Рисунок 4 — Алгоритм выделения возможного лечения

Таблица 1 — Логическая структура БЗ

Название болезни <sub>1</sub>	Симптом 1	Лечение 1
	...	...
	Симптом М	Лечение L
...		
Название болезни <sub>N</sub>	Симптом 1	Лечение 1
	...	...
	Симптом М	Лечение L

Описанный алгоритм реализован программно в среде визуального проектирования Delphi.

### *Литература*

1. Базы знаний интеллектуальных систем / Гаврилова Т.А., Хорошевский В.Ф. — СПб: Питер, 2000. — 384 с.
2. Логический подход к искусственному интеллекту: От модальной логики к логике баз данных: Пер. с франц./Тейз А., Грибомон П., Юлен Г. и др. — М.:Мир, 1998.— 494 с.
3. Осипов Г. С. Приобретение знаний интеллектуальными системами: Основы теории и технологии. — М.: Наука. Физматлит, 1997. — 112 с.
4. Поспелов Д. А. Логико-лингвистические модели в системах управления. — М.: Энергоиздат, 1981. — 232 с.
5. Коломойцева И.А. Формирование базы знаний и постановка диагноза на основе естественно-языковых психологических текстов /Известия ТРТУ. Тематический выпуск: Материалы Всероссийской научно-технической конференции с международным участием «Компьютерные и информационные технологии в науке, инженерии и управлении». — Таганрог: Издательство ТРТУ, 2005. №1 (45). — С. 142–145.
6. Морозов В.Н. Коломойцева И.А. Программная система для формирования тематических знаний на основе естественно-языковых текстов сети Интернет / Матеріали студентської науково-технічної конференції «Комп'ютерний моніторинг і інформаційні технології». — Донецьк, 2005 (электронное издание). — 2 с.

## МОДЕЛИРОВАНИЕ И ВИЗУАЛИЗАЦИЯ ПОТОКОВ КРОВИ В СОСУДАХ С РАЗЛИЧНОЙ ПАТОЛОГИЕЙ

Ю.А. Скобцов, Ю.В. Родин, В.С. Оверко  
ДонНТУ, ИНВХ АМН, ДонНУ

### *Анотація*

*На основі системи рівнянь Нав'є-Стокса і рівняння нерозривності розроблено моделі поведінки потоків крові у судинах людини і проведено комп'ютерні експерименти. Розраховано розподіл полів швидкості, завихореності і тисків для різних патологій — наявності атеросклеротичних бляшок, звитості, шунтування судин. Отримані результати дозволяють більш якісно діагностувати судинні захворювання, своєчасно вибрати адекватні заходи терапевтичного лікування та розширити показання до превентивного хірургічного лікування.*

Давно признано, что течение жидкостей и газов играет жизненно важную роль в биологических и физиологических процессах, происходящих в человеческом организме. Кровь представляет собой суспензию и при детальном рассмотрении является существенно неоднородной жидкой средой, поведение которой описывается сложной системой уравнений в частных производных.

Как правило, пульсационный поток крови в сосудистой системе здорового человека является почти ламинарным, хотя максимальное число Рейнольдса в нем достигает 10000 [1]. Ламинарность в здоровой сосудистой системе нарушается практически только в аорте в течение малых долей каждого цикла сердечного сокращения, где наблюдаются всплески турбулентности. Однако в результате заболеваний наблюдается закупорка или стеноз (сокращения русла) сосудов, что ведет к нарушению ламинарности и возникновению турбулентности и иногда явления кавитации. В частности стеноз имеет место при атеросклерозе и других сосудистых заболеваниях, профилактика и лечение которых имеют колоссальное практическое значение.

Данная работа посвящена методам математического моделирования движения жидкости в кровеносных сосудах, в основе которых лежит численное решение системы уравнений Навье-Стокса и уравнения неразрывности при следующих допущениях: жидкость несжимаемая; массовые силы отсутствуют. Система замыкалась следующими граничными условиями: для компонент скоростей на стенке задавались условия прилипания и неперетекания, на входе — профиль Пуазейля, на выходе — «мягкие» граничные условия. Для давления на всех границах

задавались градиентные условия: на стенке — полученное из соответствующей проекции уравнений Навье-Стокса.

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial y} + v \frac{\partial u}{\partial z} + w \frac{\partial u}{\partial z} &= -\frac{\partial P}{\partial x} + \frac{1}{\text{Re}} \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right), \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial z} + w \frac{\partial v}{\partial z} &= -\frac{\partial P}{\partial y} + \frac{1}{\text{Re}} \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right), \\ \frac{\partial w}{\partial t} + u \frac{\partial w}{\partial y} + v \frac{\partial w}{\partial z} + w \frac{\partial w}{\partial z} &= -\frac{\partial P}{\partial z} + \frac{1}{\text{Re}} \left( \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} &= 0 \end{aligned}$$

Система содержит следующий безразмерный параметр:

$$\text{Re} = \frac{U \cdot D}{\nu} \text{ — число Рейнольдса.}$$

В качестве начальных данных задается исходное поле скоростей и давления. В качестве начального поля скоростей выбран профиль Пуазейля. На продольный градиент давления налагалась синусоидальная волна возмущений. Проведены исследования поведения потоков крови в сосудах с различными патологиями: стеноз вследствие наличия атеросклеротических бляшек, извитость, шунтирование сосудов.

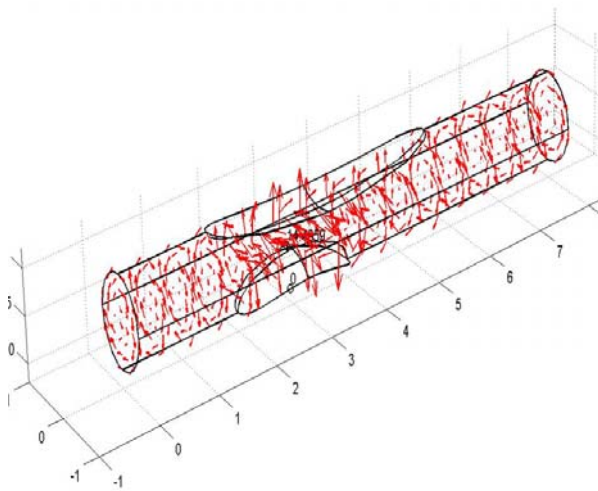


Рисунок 1 — Поле завихренности

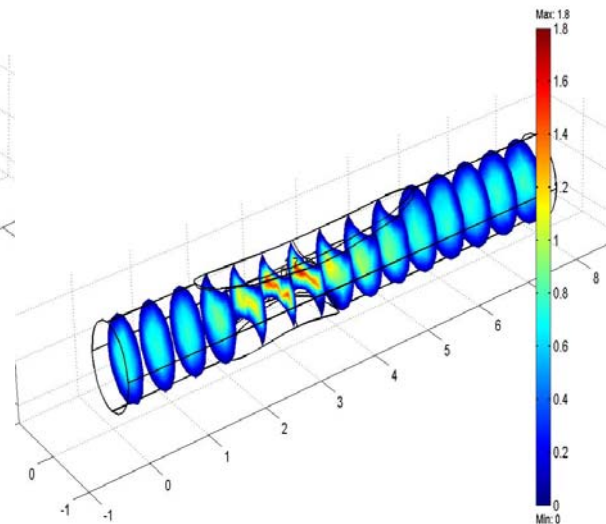


Рисунок 2 — Поле скорости

## ***1. Моделирование поведения потока крови при стенозе сосуда в пространственной области***

Расчеты проведены для области, имитирующей сосуд с анастомозом 30%. Расчетная область приведена на рис.1. Для численного решения сформулированной задачи в качестве основы был взят описанный в [2–4] метод. В настоящей работе для аппроксимации конвективных членов использованы разности против потока, что приводит к повышению устойчивости расчетного алгоритма для более высоких чисел Рейнольдса и Пекле.

1. Исследование модели показало, что наблюдается значительное увеличение скорости в области анастомоза и, как следствие, образование каверн.
2. Увеличение скорости в области сужения приводит к возрастанию местного числа Рейнольдса и возникновению турбулентного следа за бляшкой. Возникающие при этом пульсации давления способствуют разрушению поверхности бляшки и могут служить причиной возникновения кавитационных явлений. Увеличение касательных напряжений при турбулизации потока приводит к разрушению макромолекул, обычно стабилизирующих поток, и способствует возникновению дополнительной неустойчивости.
3. Полученные распределения гидродинамических характеристик потока в пространственной области позволяют точнее оценить механизмы деструктивного воздействия на бляшку:
4. а) в 3-мерной области с пониженным давлением может возникать явление кавитации, при котором паровые пузырьки переносятся в область с большим давлением вниз по потоку (на задней стенке) и схлопываются, что приводит к изъязвлению бляшки.  
б) возникновение турбулентного пятна на границе 3-мерной вихревой области приводит к тому, что касательные напряжения могут увеличиваться на два порядка.
5. Риск образования кавитационных каверн также возрастает при гипертермии потока в пораженном сосуде, что имеет большое прикладное значение (например, это следует учитывать при повышении температуры после операции).
6. Таким образом, впервые исследованы условия возникновения кавитации и турбулентности, которые разрушают бляшку.

Исследованные патофизиологические механизмы гидродинамики кровообращения позволяют более качественно диагностировать заболевание, своевременно выбрать адекватные меры

терапевтического лечения, профилактики ишемических нарушений и расширить показания превентивному хирургическому лечению. На рис.1–2. приведены для примера поля завихренности и скоростей в стенозированных сосудах. Рассчитано также распределение давления при различных параметрах стеноза.

## ***2. Исследование поведения потоков крови в извитых сосудах***

Кроме наличия атеросклеротических бляшек нарушению кровоснабжения мозга (и следовательно значительному увеличению риска возникновения инсульта) способствует извитость кровеносных сосудов (в данном случае сонных артерий). Исследовано влияние степени искривления кровеносного сосуда на изменение расхода жидкости в пораженном сосуде и получены следующие результаты.

1. Исследовано поведение потоков крови в извитых сосудах при различных радиусах кривизны.
2. Смыкание сосудов приводит к увеличению областей отрывных течений, и, как следствие, потере потоком устойчивости.
3. Показано, что дальнейшее развитие вихревых структур приводит к резкому снижению расхода, а значительная деформация поля давления приводит к возникновению обратных течений, что блокирует нормальный кровоток в пораженном сосуде.

На рис.3 приведены распределение давления и линий тока для извитого сосуда.

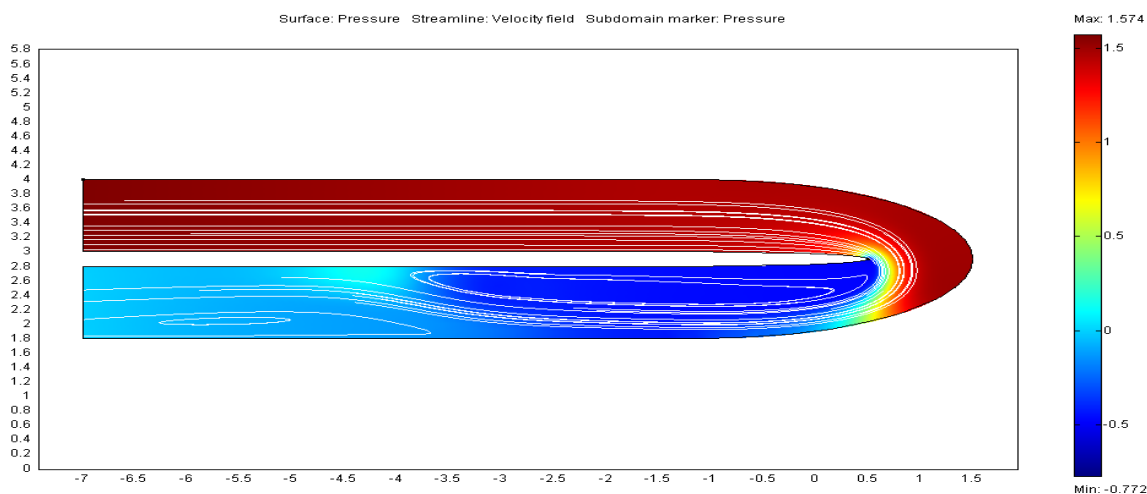


Рисунок 3 — Распределение давления и линий тока для извитого сосуда

### **3. Расчет поведения потока крови при шунтировании пораженного сосуда с целью выбора оптимальной геометрии (угла вшивания, радиуса и т.п.)**

Проводимые операции по шунтированию пораженных кровеносных сосудов требуют аналитического исследования биогидромеханики крови. В частности, для увеличения эффективности работы шунта требуется определить факторы, которые влияют на кровоток в шунтированном сосуде.

Основной вклад в увеличение сопротивления составляют кривизна входных и выходных зон шунта, угол входа и пульсационный характер течения крови. На рис.4 представлены распределение полей скоростей и давления.

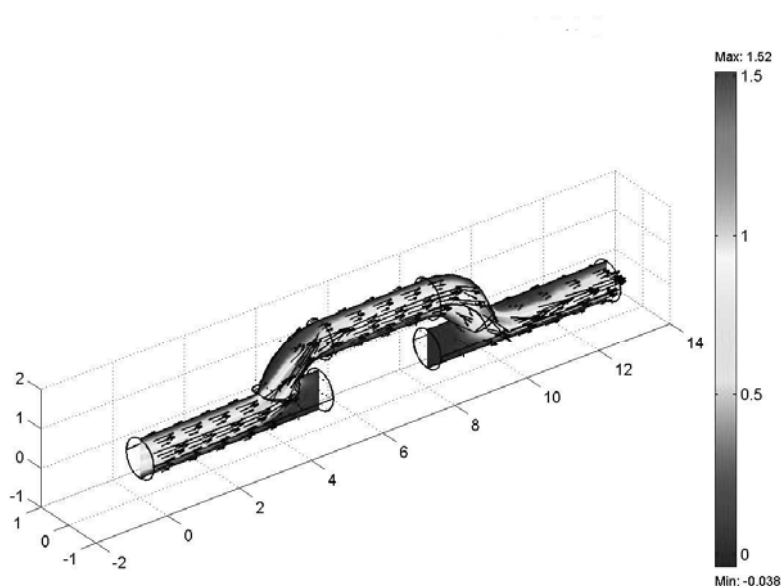


Рисунок 4 — Распределение давлений и линии тока в шунтирующем сосуде

Получены следующие результаты [5].

1. Исследовано поведения потоков крови в шунтированных кровеносных сосудах.
2. Рассчитаны оптимальные геометрические характеристики сосудов – протезов, которые шунтируют стенозированный сосуд: угол, под которым он вшивается; радиус кривизны; соотношение диаметров шунтируемого и шунтирующего сосудов и т.п.
3. Сравнение расчетных данных при углах 45 и 70 градусов показывает, что в первом случае потери давления по сравнению с

прямой трубой гораздо меньше, чем во втором. Поэтому желательно удерживать шунт к сосуду в пределах 45 градусов.

4. Выбор оптимальных характеристик шунтирующего сосуда позволяет существенно уменьшить риск послеоперационных осложнений.

### *Литература*

1. Каро, Педли и др. Механика кровообращения. М, Мир. — 1981. — 624 с.
2. 2.В.К.Гусак., В.С.Оверко, Ю.А.Скобцов., Ю.В.Родин и др. Математическое моделирование энергетических воздействий на атеросклеротическую бляшку в зависимости от угла атаки // Вестник неотложной и восстановительной медицины. — том 2, Донецк 2001. — №2-3. — С. 133-139.
3. 3.Ю.А.Скобцов, Н.В.Финошин, В.С.Оверко, Ю.В.Родин. Исследование течения потока крови в атеросклеротически стенозированном кровеносном сосуде // Труды института прикладной математики и механики НАН Украины. — Донецк:2003, том.8. — С.135-141.
4. Ю.В.Родин, Штутин А.А., Скобцов Ю.А., Оверко В.С. Математическое моделирование гидродинамических и кавитационных воздействий на атеросклеротическую бляшку, как обоснование к оперативному лечению при малых степенях стеноза// Актуальные вопросы диагностики и лечения сосудисто-мозговой недостаточности и распространенного атеросклероза. — Киев: Национальный медицинский институт имени А.А.Богомольца. — 2003. — С.18-21.
5. Скобцов Ю.А., Финошин Н.В., Оверко В.С., Родин Ю.В. Исследование способов шунтирования кровеносных сосудов// Вестник Донецкого национального университета. — Донецк: 2004.



## ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ФУНКЦИОНИРОВАНИЯ КАНАЛА ИЗМЕРЕНИЯ ТЕМПЕРАТУРЫ

А.А. Зори, А.В. Вовна, А.А. Григор

Донецкий национальный технический университет,  
кафедра электронной техники

E-mail: Zori@kita.dgtu.donetsk.ua; Vovna\_Alex@ukr.net

### *Анотація*

*Розроблені імітаційні математичні моделі електронних схем, дозволяють за допомогою відповідних пакетів прикладних програм конструювати і проектувати інформаційно вимірвальні прилади і системи.*

**Введение.** Температура не является непосредственно измеряемой физической величиной, ее значение определяют косвенно с помощью чувствительных к температуре физических параметров, которые могут быть непосредственно измерены. Таким образом, устройства для измерения температуры состоят из чувствительного элемента, находящегося в тепловом равновесии со средой, и измерительного прибора, регистрирующего свойства термометрического вещества чувствительного элемента.

Методы измерения температуры подразделяются на контактные и бесконтактные. В настоящее время чаще применяются и наиболее полно развиты контактные (зондовые) методы, а по видам их термометрических свойств их можно разделить на механические; электрические; термодинамические; оптические [1]. В данной работе для измерения температуры в диапазоне от 0 до 600°C используются терморезисторные и термоэлектрические термометры.

**Постановка задачи проектирования.** Целью работы является разработка математических имитационных моделей первичных измерительных преобразователей и электронных схем, удобных для дальнейшей алгоритмизации и реализации их в соответствующих программах прикладных программ.

Задачей данной работы является получение математических моделей для имитации функционирования каналов информационно измерительных систем для выбора альтернативных вариантов схемных решений.

**Решение задач и результаты исследования.** Структурная схема исследуемого канала измерения температуры приведена на рис. 1. На схеме обозначены ПИП — первичный измерительный преобразователь (преобразовывает измеряемую температуру в напряжение);

ДУ — дифференциальный усилитель, НП — нормирующий преобразователь, УВХ — устройство выборки хранения, АЦП — аналого-цифровой преобразователь; МП — микропроцессор; ЦИ — цифровая индикация; МПС — микропроцессорная система.

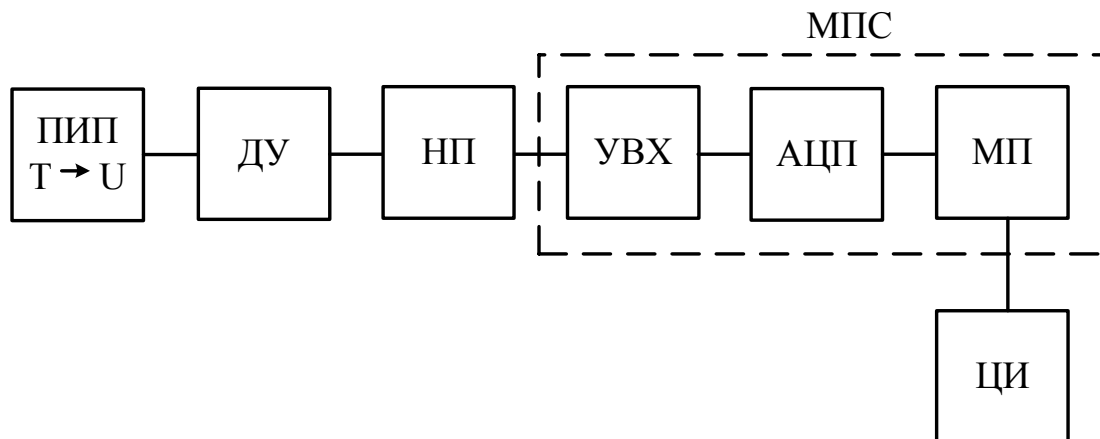


Рисунок 1 — Структурная схема исследуемого канала измерения температуры

В данном канале производилось измерения средних температур (от 0 до плюс 600°C), для чего в первом варианте используется термопара хромель-алюмель (ТХА), которая обладает наиболее близкой к прямой термоэлектрической характеристикой [2], которая в заданном диапазоне температур описывается уравнением типа:

$$E_{tp}(T) = \alpha_{tp} \cdot T,$$

где  $E_{tp}(T)$  — напряжение на выходе ПИП (термопары) как функция температуры,  $\alpha_{tp} = 4,069 \cdot 10^{-5}$  В/°С — температурный коэффициент для ТХА<sub>68</sub>. В результате предложен элемент, моделирующий изменение напряжения на выходе термопары от измеряемой температуры в ППП EWB 5.12 Pro [3] (см. рис. 2).

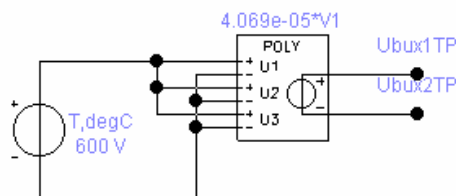


Рисунок 2 — Имитационная модель ПИП — термопары

В качестве второго способа измерения температуры используем ПИП платиновый терморезистор, включенный в одно из плеч измерительного моста [1] (см. рис. 3).

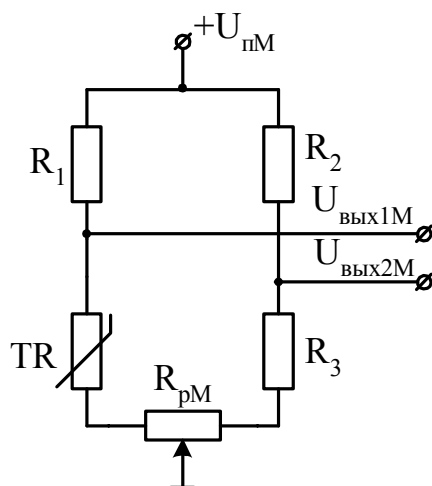


Рисунок 3 — Измерительный мост с терморезистором

Аппроксимируем номинальную характеристику преобразования терморезистора 100П [2] в заданном диапазоне температур:

$$R(T)=R_0 \cdot (1+\alpha \cdot T),$$

где  $R_0=100$  Ом — сопротивление терморезистора при температуре  $T=^{\circ}\text{C}$ ;  $\alpha_{\text{тс}}=3,9113 \cdot 10^{-3}$  В/ $^{\circ}\text{C}$  — температурный коэффициент платинового терморезистора.

Составим модель терморезистора для моделирования его работы в ППП EWB5.12 Pro. С помощью пакета прикладных программ MathCad 2000 Pro аппроксимируем степенным полиномом 3-ей степени зависимость напряжения  $U_{\text{зи}}$  полевого транзистора J2N4091 от измеряемой температуры. Полученные значения коэффициентов полинома используются в элементе Polynomial Source ППП EWB 5.12 Pro. В результате получен элемент, моделирующий изменение сопротивления платинового терморезистора, от измеряемой температуры [3] (см. рис. 4).

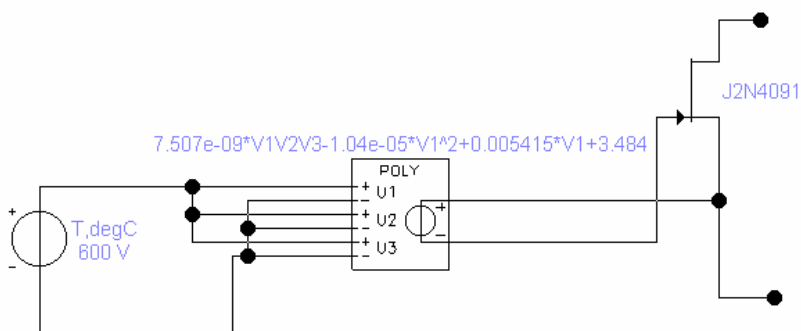


Рисунок 4 — Модель терморезистора в ППП EWB5.12 Pro

Дальнейшее масштабирование выходного сигнала до входного уровня аналоговых каналов микропроцессорной системы осуществляется дифференциальным измерительным услителем и нормирующим преобразователем. Принципиальная схема, имитирующая работу системы нормирующих преобразователей в ППП EWB5.12 Pro приведена на рис.5.

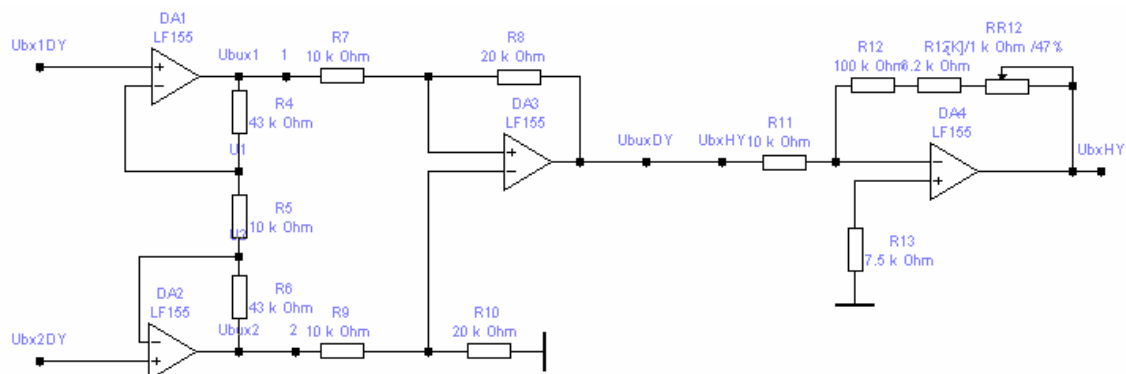


Рисунок 5 — Принципиальная схема нормирующих преобразователей канала измерения температуры

Дифференциальный усилитель выполнен по схеме прецизионного измерительного усилителя, при этом схему включения выбирают такой, чтобы изменение выходных сигналов плеч взаимно вычитались. Такой подход достаточно эффективен, так как изменения выходных сигналов, вызванные нагревом близких по параметрам компонентов, в первом приближении одинаковы. Зависимость напряжения на выходе нормирующего преобразователя как функция изменения температуры приведена на рисунке 6.

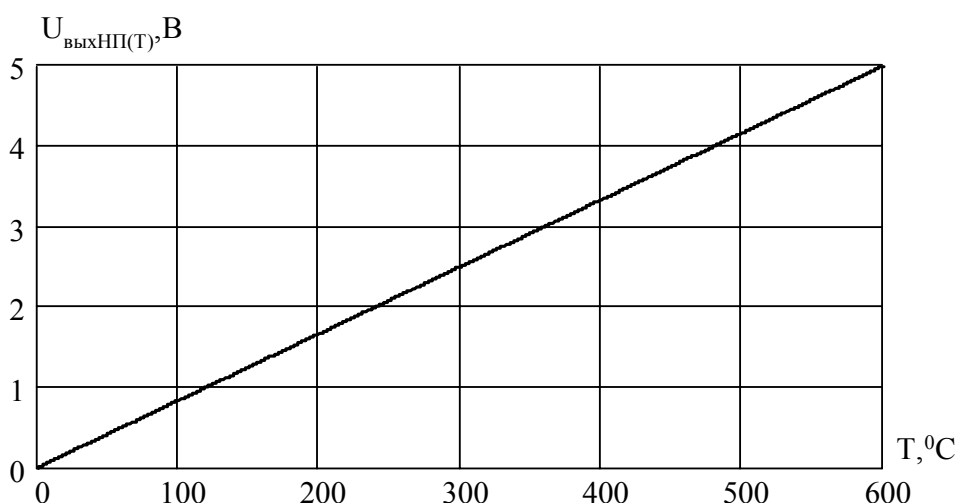


Рисунок 6 — Зависимость напряжения на выходе нормирующего преобразователя как функция изменения температуры

В процессе имитационного функционирования измерительной системы установлены значения чувствительностей промоделированных каналов измерения температуры, произведен анализ переходных процессов, спектральный анализ, исследовано влияние температуры на аналоговые полупроводниковые элементы.

Экспериментально установлено, что показатели точности при измерении температуры в полученных имитационных моделях, по сравнению с опытными образцами приборов не выходят за допустимые пределы. Так, например, относительная погрешность измерения температуры при имитационном моделировании и согласно экспериментальных данных лежит в диапазоне  $(0,5 \dots 1)\%$ .

### ***Выводы***

1. Разработаны математические имитационные модели первичных измерительных преобразователей и электронных схем, удобных для дальнейшей алгоритмизации и реализации их в соответствующих программах прикладных программ.

2. Полученная модель имитации функционирования канала измерения температуры позволяет производить альтернативный выбор вариантов схемных решений и возможность подбора наиболее приемлемой элементной базы.

3. Необходимым этапом при проектировании и конструировании многоканальных информационно измерительных систем является этап математического имитационного моделирования.

### ***Литература***

1. Зорі А.А., Коренєв В.Д., Хламова М.Г. Електронні системи вимірювання і контролю параметрів водних середовищ: Навчальний посібник. — Донецьк: РВА ДонНТУ, 2002. — 352 с.
2. Геращенко О.А., Гордов А.Н., Лах В.И., Стадник Б.И., Ярышев Н.А. Справочник: Температурные измерения. — К.: Наукова думка, 1984. — 495 с.
3. Методические указания к выполнению лабораторных работ по дисциплине «Теория измерительных приборов» для студентов специальности 7.090902 НАП (Научные, аналитические и экологические приборы и системы), Часть 2 / А.А. Зори, А.В. Вовна., Е.А. Григор — Донецк: ДонНТУ, 2004. — 41 с.

## МОДЕЛИРОВАНИЕ ГРАДУИРОВОЧНЫХ ХАРАКТЕРИСТИК ИНТЕЛЛЕКТУАЛЬНЫХ ДАТЧИКОВ ПРОСТРАНСТВЕННЫМИ ЭЛЕМЕНТАМИ

С.И. Клевцов

Таганрогский государственный радиотехнический университет

### *Анотація*

*У статті розглядаються перспективні моделі апроксимації градуївальної характеристики інтелектуального датчика, що дозволяють істотно підвищити точність обчислення вимірюваних фізичних величин в інтелектуальному датчику.*

Повышение точности и достоверности определения значений физических величин, таких как давление, температура и других, представляет собой одну из важных задач совершенствования математических и алгоритмических методов обработки результатов измерений в интеллектуальных датчиках. Ее решение во многом определяет степень надежности и безопасности функционирования сложных технических объектов и промышленных производств.

В последние годы интеллектуальные датчики физических величин получили широкое распространение в системах мониторинга и диагностики технических объектов. Для обработки данных, поступающих с чувствительного элемента, в интеллектуальном датчике используются различные модели его градуировочной характеристики, которые, в основном, базируются на полиномиальном виде функции преобразования чувствительного элемента [1, 2, 3, 4, 5]. Характерным для этих моделей является использование единого представления градуировочной характеристики для всей области изменения полезного сигнала и внешних факторов (области определения градуировочной характеристики).

Такой подход имеет несомненные преимущества, например такие, как простота и скорость обработки, однако при существенной нелинейности и зависимости от внешних факторов, не обеспечивает низкую и сопоставимую по величине погрешность во всех точках области определения.

Дальнейшее снижение погрешности обработки данных в интеллектуальных датчиках при измерении физических величин связано с использованием специализированных пространственных моделей их градуировочных характеристик. Для этого в качестве модели градуировочной характеристики предлагается использовать систему локальных линейных или нелинейных про-

пространственных элементов. При таком подходе в зависимости от заданных требований по точности вычислений в рамках аппроксимации одной градуировочной характеристики имеется возможность применения различных базовых функций.

Исследование возможностей пространственной модели градуировочной характеристики проводилось с использованием данных испытаний перспективного датчика давления, имеющего в своем составе первичный преобразователь давления (канал давления) и первичный преобразователь температуры (канал температуры). Эталонные давление и температура, подаваемые на датчик при его испытаниях, находились в пределах  $0-80 \text{ кгс/см}^2$  и  $-4^\circ\text{C} - +35^\circ\text{C}$  ( $269 \text{ K} - 308 \text{ K}$ ) соответственно.

Исходные данные представляют собой часть рабочего диапазона датчика. Для этой части рабочего диапазона датчика характерна незначительная нелинейность, как в системе «эталонное давление — выходной сигнал с канала давления датчика» при фиксированной температуре, так и в большей степени в системе «выходной сигнал с канала давления датчика — температура» при фиксированном давлении. Поэтому в качестве локальных пространственных элементов градуировочной поверхности были использованы плоскостные элементы.

Характер функциональных связей эталонного давления и сигналов, снимаемых с канала давления датчика при различных температурах показывает, что на доминирующую линейную зависимость в функции  $P=P(U_p)$  при  $T=\text{const}$  накладывается незначительная нелинейность, которую можно оценить в  $2,1\% - 3,7\%$  во всем диапазоне изменения  $P$ . Таким образом, график  $P=P(U_p)$  имеет небольшую нелинейность (выпуклость). Таким же образом можно охарактеризовать канал температуры датчика.

Относительная погрешность исходных данных составляет для канала давления погрешность не выше  $0,05\%$ , для канала температуры — не выше  $0,07\%$ .

Градуировочная характеристика датчика в виде пространственной системы плоскостей может быть построена несколькими способами. Исходной информацией являются массивы градуировочных данных канала давления и канала внешнего фактора (температура), полученные при испытаниях датчика в требуемых диапазонах изменения давления и температуры.

Исходные данные представляются в следующем виде:

1. Имеется область определения градуировочной характеристики:

$$Q = Q_P \cup Q_T, \quad Q_P = [P_{\min}, P_{\max}], \quad Q_T = [T_{\min}, T_{\max}],$$

где  $P_{min}, P_{max}$  — минимальное и максимальное эталонные давления, установленные при испытаниях датчика;

$T_{min}, T_{max}$  — минимальное и максимальное эталонные температуры, установленные при испытаниях датчика.

2. Имеются множество фиксированных значений эталонных давления и температуры, при которых проводятся тестовые замеры сигналов с каналов давления и температуры датчика давления:

$$P = \{P_i\}_{i=1}^I, P_i \in Q_P, \forall i = 1, I, T = \{T_k\}_{k=1}^K, T_k \in Q_T, \forall k = 1, K.$$

3. Каждой паре значений  $(P_i, T_k)$  ставятся в соответствие множество значений сигналов с каналов давления и температуры датчика давления:

$$U^{ik} = \{U_{P_j}^{ik}, U_{T_j}^{ik}\}_{j=1}^m,$$

где  $U_{P_j}^{ik}, U_{T_j}^{ik}$  —  $j$ -е значения сигналов, полученных соответственно с каналов давления и температуры при установлении эталонных значений  $P_i$  и температуры  $T_k$  на испытательном стенде;

$m$  — количество испытаний (величина выборки при заданных  $P_i$  и  $T_k$ , как правило, не меняется при  $\forall i, k, i \in 1, I; k \in 1, K$ ).

Таким образом, имеется множество троек экспериментальных значений, на основе которых строится аппроксимация градуировочной характеристики

$$\{(P_j, U_{P_j}, U_{T_j})\}_{j=1}^n,$$

где  $n$  — количество экспериментальных точек.

Расчет коэффициентов аппроксимации с использованием метода наименьших квадратов базируется на модели элементарной плоскости в виде

$$P = A + B_1 U_P + B_2 U_T$$

в системе плоскостей пространственной градуировочной характеристики датчика давления.

Построенная пространственная градуировочная характеристика датчика давления представляет собой систему плоскостей в виде прямоугольников. Каждая такая плоскость имеет собственную, отличную от других плоскостей, ориентацию в пространстве. Это очевидно при сравнении коэффициентов аппроксимации, полученных для различных плоскостей градуировочной характеристики.

Абсолютная погрешность определения давления при использовании пространственной характеристики в виде системы плоскостей не превышает  $0,014 \text{ кгс/см}^2$  при диапазоне значений входного сигнала давления от  $0 \text{ кгс/см}^2$  до  $80 \text{ кгс/см}^2$ , что приводит к максимальной от-



носительной погрешности  $\sim 0,076\%$ . Характер изменения относительной погрешности представлен на рисунке.

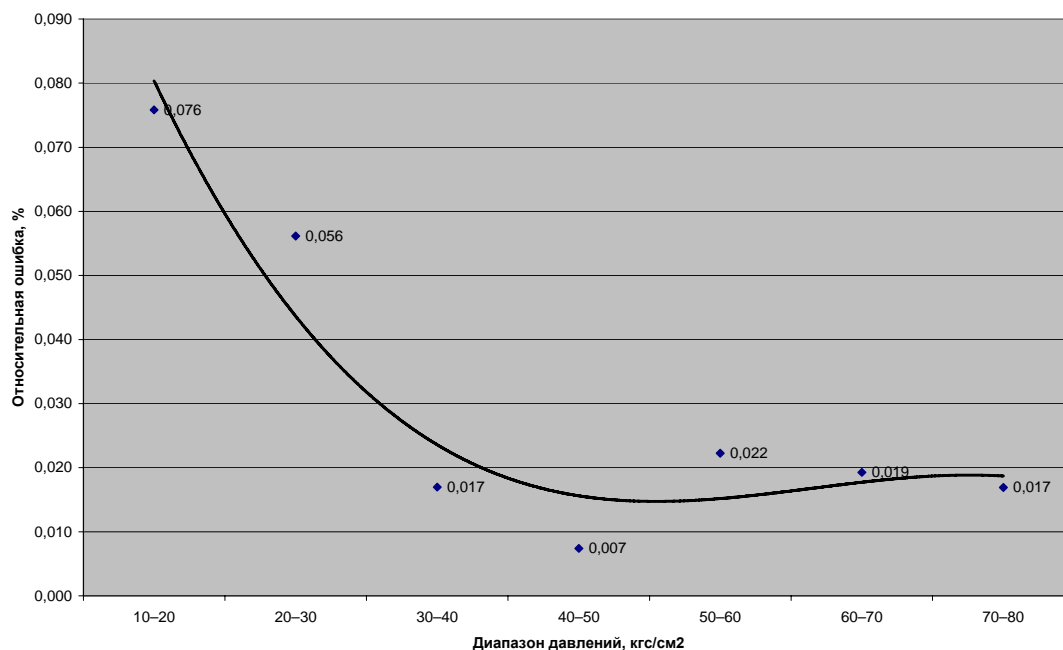


Рисунок — Относительная погрешность определения давления для различных диапазонов давлений (средняя для всех диапазонов температур)

Учитывая разброс исходных данных в  $\sim 0,05\%$  можно говорить об оценке погрешности определения давления за счет неточности построения градуировочной характеристики в размере  $< 0,03\%$ . Уменьшение погрешности определения давления за счет неточности построения градуировочной характеристики связано с повышением точности аппроксимации градуировочной характеристики, что в рамках рассматриваемой модели требует большей детализации градуировочной характеристики.

При аппроксимации указанным способом формируется система плоскостей, покрывающая всю поверхность градуировочной характеристики датчика. Однако на границах областей области определения  $U \times T$  построенные плоскости практически не соприкасаются. Это приводит к дополнительной ошибке аппроксимации, которую необходимо учитывать при использовании данного метода вычисления давления.

Ошибку аппроксимации градуировочной характеристики можно оценить как разность между точками "идеальной" градуировочной по-

верхности и соответствующего локального элемента построенной системы плоскостей.

Анализ результатов показывает, что представленная модель градуировочной характеристики обеспечивает высокую точность вычисления значений физической переменной во всем диапазоне воздействующих факторов при небольшой нелинейности истинной градуировочной характеристики. В случае существенной нелинейности реальной градуировочной характеристики эффективная аппроксимация возможна при использовании нелинейных локальных элементов.

Выбор линейного или нелинейного типа локального элемента градуировочной поверхности, определение вида базовой функциональной зависимости и схемы аппроксимации зависят от характеристик реальной функции преобразования чувствительного элемента интеллектуального датчика и представляют собой перспективную область исследований. Их результаты могут помочь в создании интеллектуальных микропроцессорных систем градуировки чувствительных элементов интеллектуальных датчиков в процессе проведения их испытаний.

### *Литература*

1. Семенов Л.А., Сирая Т.Н. Методы построения градуировочных характеристик средств измерений. — М.: Изд-во стандартов, 1986.
2. Шапонич Д., Жигич А. Коррекция пьезорезистивного датчика давления с использованием микроконтроллера.//Приборы и техника эксперимента. 2001. №1.
3. Бобровников Н.Р., Яркин С.В., Гридин Ю.Н., Стрыгин В.Д., Чертов Е.Д. Математическое обеспечение микропроцессорных преобразователей аналоговых пневматических сигналов.//Приборы и системы. Управление, контроль, диагностика. — 2002. — №2 — С.36–39.
4. Гутников В.С., Клементьев А.В., Лопатин В.В., Соловьев А.Л., Кривченко Т.И. Микропроцессорный измеритель давления и температуры.// Приборы и системы. Управление, контроль, диагностика. — 1995. — №8. — С.28–30.
5. Клевцов С.И., Линьков В.С., Веретельников Ю.А., Кузьминов В.Г. Погрешности вычисления давления в интеллектуальном датчике при матрично-полиномиальной аппроксимации его градуировочной характеристики// Известия ТРТУ. 2004. №2.

**ОБОСНОВАНИЕ АЛГОРИТМА ОЦЕНИВАНИЯ  
ЧАСТОТНЫХ ПАРАМЕТРОВ РАДИОИМПУЛЬСОВ  
БОЛЬШОГО УРОВНЯ ПО ИНФОРМАТИВНЫМ  
СИГНАЛАМ АКУСТООПТИЧЕСКОГО  
СПЕКТРОАНАЛИЗАТОРА**

Ю.В. Рябкин  
ГАХК “Топаз”

***Анотація***

*Запропоновано алгоритм розпізнавання радіочастотних сигналів високого рівня енергії на виході акустооптичного спектроаналізатору. Даний алгоритм може бути використаний в інформаційно-вимірвальних системах контролю параметрів пасивної радіолокації.*

Использование акустооптических спектроанализаторов в радиотехнических системах при оценивании параметров радиосигналов большой энергии затруднительно в связи с тем, что на выходе спектроанализатора обнаруживаются компоненты, отсутствовавшие во входном сигнале, что приводит к неопределенности частотных оценок. В связи с этим практический интерес вызывает следующая постановка задачи. Пусть на вход радиотехнической системы поступают периодические радиосигналы с монохроматическим заполнением. С помощью радиоканала измеряется длительность импульса, период повторения импульсов и другие параметры, кроме частоты. Полагаем, что на вход акустооптического спектроанализатора поступает сигнал постоянной частоты, и в случае большой энергии сигнала на выходе может иметь совокупность сигналов разных частот. Ставится задача определения частоты входного радиосигнала по совокупности выходных сигналов спектроанализатора. Эта задача относится к классу задач многоальтернативного распознавания. [1].

Известно [2], что для преодоления априорной неопределенности в отношении параметров распределения выходных сигналов спектроанализатора целесообразно использовать этап обучения, либо моделирования в реальном времени. Для этого необходимо выполнить описание процедур построения эталонных описаний классов сигналов и принятия решения о частоте входного сигнала по совокупности выходных сигналов, количество которых изменяется в зависимости от интенсивности входного сигнала, а также рассмотреть особенность

построения эталонных описаний совокупностей выходных сигналов и принятия решения о частоте входного сигнала при переменном количестве выходных сигналов.

Особенности процесса обучения при изменяющемся количестве выходных сигналов спектроанализатора и информация о распределении распознаваемого вектора содержится в результатах независимых наблюдений, составляющих обучающую выборку.

Представим возможное многообразие входных сигналов в виде таблицы 1, в которой количество строк равно количеству разрешаемых интервалов по частоте, где,  $l=1, \dots, N$ , соответствует середине интервала, а количество столбцов соответствует количеству уровней квантования интенсивности сигнала  $I_s$ ,  $S=1, \dots, M$ , равное интенсивности в середине интервала квантования. Образующее параметрическое пространство векторов входных сигналов  $V=(I_s, \omega_l)^T$  имеет размерность  $K=MN$ , что соответствует  $K$  классам входных воздействий  $(b_1, \dots, b_k)$ .

Таблица 1 — Возможное многообразие входных сигналов

$\omega$	$I$	1	2	3	...	$M$
1		$\omega_1, I_1$	$\omega_1, I_2$	$\omega_1, I_3$	...	$\omega_1, I_M$
2		$\omega_2, I_1$	$\omega_2, I_2$	$\omega_2, I_3$	...	$\omega_2, I_M$
3		$\omega_3, I_1$	$\omega_3, I_2$	$\omega_3, I_3$	...	$\omega_3, I_M$
$\vdots$		$\vdots$	$\vdots$	$\vdots$	...	$\vdots$
$N$		$\omega_N, I_1$	$\omega_N, I_2$	$\omega_N, I_3$	...	$\omega_N, I_M$

Примем, что распределение выходного вектора  $x=(x_1, \dots, x_p)$  нормально и имеет вид:

$$p(x, p | B_j) = (2\pi)^{-\frac{1}{2}} R^{-\frac{1}{2}} \exp\left[-\frac{1}{2} (x - a_j)^T R^{-1} (x - a_j)\right], \quad (1)$$

где  $a_j = M\{x\} = (a_{j1}, a_{j2}, \dots, a_{jp})$  — вектор математических ожиданий компонент  $x_i$ ,  $i=1, \dots, p$ , а  $R^{-1}$  — квадратная матрица из  $p \times p$  строк и столбцов, обратная матрице  $R$  с элементами  $R_{ij} = M[(x_i - a_{ij})(x_j - a_{ij})^T]$ .

Особенностью рассматриваемой задачи является тот факт, что при  $I > I_{пор}$  количество выходных сигналов может быть больше одного, и в общем случае достигает величины  $N$ . Поэтому результаты обучения будем помещать в базу данных, имеющих многослойную структуру, в которой номер  $p$ -го слоя соответствует количеству выходных сигналов на выходе спектроанализатора. Количество слоев структурной базы данных

равно  $N$ . В каждом слое отведем  $K$  ячеек для хранения данных о выборочном среднем и выборочной корреляции матрицы (см. табл.2).

Таблица 2 — Данные о выборочном среднем и выборочной корреляции матрицы на каждом слое

N слоя	N класса					к+1
	1	2	3		к	
1	$a_{11}, R_{11}$				$a_{1k}, R_{1k}$	$a_1, W_1$
2						
⋮	⋮	⋮	⋮		⋮	⋮
N	$a_{N1}, R_{N1}$				$a_{Nk}, R_{Nk}$	$a_N, W_N$

Для каждого из классов путем  $m$ -кратной подачи на вход спектроанализатора сигнала с параметрами  $b_j$  по количеству выходных сигналов  $p$  и их параметрам вычисляются выборочные оценки:

вектор средних

$$a_{jp} = \left( \frac{1}{m} \right) \sum_{i=1}^m x_{ip}^{(i)} \quad ; \quad x_{ip}^{(i)} = (x_{1i}, \dots, x_{pi}), \quad (2)$$

и корреляционная матрица

$$R_{ip} = \frac{1}{m-1} \sum_{i=1}^m (x_{ip}^j - a_{jp})(x_{ip}^j - a_{jp})^T, \quad (3)$$

которые заносятся в  $p$ -й слой банка данных на место с номером  $j$ .

Поскольку в каждом из слоев средние и корреляционные матрицы для каждого из классов различны, в качестве общей корреляционной матрицы для  $p$ -го слоя примем смешанную корреляционную матрицу

$$W_p = \frac{1}{k} \sum_{j=1}^k R_{jp}, \quad (4)$$

а в качестве среднего

$$a_p = \frac{1}{k} \sum_{j=1}^k a_{jp}, \quad (5)$$

которые разместим в столбцы с номером  $k+1$ .

При необходимости для каждого из слоев можно уменьшить количество признаков (выходных сигналов) с помощью известных декоррелирующих преобразований. Эта операция не является принципиальной, поэтому будем далее считать, что признаки некоррелированы. Более существенным является то, что, несмотря на различимость входных (эталонных) сигналов, выходные в результате

нелинейных преобразований могут стать неразличимыми. В связи с этим необходимо проверить различимость эталонных классов с помощью какого-либо критерия.

С практической точки зрения представляет интерес критерий выделения признаков, которые связаны с верхней границей ошибок классификации. Такие критерии разработаны в основном для случая двух классов, тогда верхняя граница Чернова [1] для вероятности ошибок классификации двух нормальных совокупностей имеет вид:

$$a \leq \frac{1}{2} \exp \left[ -\frac{1}{8} (a_{p1} - a_{p2})^T W_p^{-1} (a_{p1} - a_{p2}) \right] \quad (6)$$

Из (6) следует, чтобы ошибка классификации не превышала заданного значения  $a_0$  необходимо выполнение неравенства:

$$(a_{p1} - a_{p2})^T W_p^{-1} (a_{p1} - a_{p2}) \geq -8 \ln(2a_0) = C_0 \quad (7)$$

или

$$d \geq C_0, \quad (8)$$

где  $d = (a_{p1} - a_{p2})^T W_p^{-1} (a_{p1} - a_{p2})$  — расстояние между классами.

Распространяя эту методику на случай многих классов, вычислим попарные расстояния между классами одного слоя, которые можно записать в виде матрицы:

$$M_p = \begin{pmatrix} d_{11}^{(p)} & \dots & d_{1k}^{(p)} \\ d_{21}^{(p)} & \dots & d_{2k}^{(p)} \\ \vdots & \dots & \vdots \\ d_{k1}^{(p)} & \dots & d_{kk}^{(p)} \end{pmatrix}. \quad (9)$$

Диагональные элементы этой матрицы характеризуют различимость каждого класса с самим собой, и равны нулю. Элементы, расположенные симметрично относительно диагонали матрицы равны между собой,  $d_{ij} = d_{ji}$ .

Таким образом, анализу можно подвергать только треугольную матрицу, за исключением диагональных элементов.

Зададим граничное значение вероятности ошибочной классификации для пары классов  $a_0$  и для него вычислим по известной методике пороговое расстояние между классами  $C_0$ . Для этого в матрице (9) выделим те элементы, которые не удовлетворяют неравенству (8). Это означает, что соответствующие этим элементам пары классов не могут быть различимы с заданной достоверностью. И если  $d_{ij} < C_0$ , и входные сигналы классов  $i$  и  $j$  различимы между собой, то выходные сигналы этих классов с заданной достоверностью не могут быть

отнесены к одному из названных классов. Данное обстоятельство необходимо учитывать далее при формулировке решающего правила. Поэтому для удобства перейдем от матрицы (9) к матрице связности, в которой количество строк и столбцов равно количеству входных классов  $K$ , а вместо  $d_{ij}$  запишем 0, если  $d \geq C_0$  и 1, если  $d_{ij} < C_0$ .

Таким образом, на этапе обучения формируется таблица с числовыми характеристиками законов распределения классов и для каждого из слоев матрица связности, показывающая возможности попарного различения классов с заданной достоверностью.

Пусть дана контрольная выборка  $x_p = (x_1, \dots, x_p)^T$ , принадлежащая одному из классов. Требуется определить, к какому классу она принадлежит. Для принятия решения необходимо использовать результаты обучения, записанные в слое с номером  $p$  таблицы 2. Далее, в соответствии с общими правилами построения решающего правила в многоальтернативных задачах [1], вычисляются оценочные функции

$$Q_{pj} = (x_p - x_p^{(j)})^T W_p^{-1} (x_p - x_p^{(j)}), \quad j = 1, 2, \dots, K \quad (10)$$

Решающее правило заключается в том, чтобы относить наблюдения  $x_p$  к тому классу, для которого оценочная функция максимальна. Допустим, что этому условию удовлетворяет класс с номером  $S$ . Затем необходимо проанализировать строку матрицы связности с номером  $S$ . Если все элементы этой строки равны нулю, то принимается решение об отнесении выборки  $x_p$  к классу  $S$ . Если же в строке имеются элементы, равные единице, то однозначное решение с заданной достоверностью не может быть принято. Тогда в этом случае принимается решение о возможности принадлежности выборки к одному из классов: классу  $S$  и классам, под номерами которых в строке матрицы связности стоят единицы.

### ***Литература***

1. Г. Ван Трис. Теория обнаружения, оценок и модуляции. — М.: Советское радио, 1972. — 744 с.
2. В.И.Тихонов, В.Н. Хорисов. Статистический анализ и синтез радиотехнических устройств и систем. — М.: Радио и связь, 1991. — 608 с.

## МОДЕЛИРОВАНИЕ ПАРАМЕТРОВ ТОНКОЙ СТРУКТУРЫ ГИДРОФИЗИЧЕСКИХ ПОЛЕЙ ПРИ ЗОНДИРОВАНИИ ВОДНОЙ СРЕДЫ

А.А. Зори, Е.О. Савкова

Донецкий национальный технический университет, факультет КИТА

### *Анотація*

*В статті розглянуто питання використання імітаційного моделювання характеристик дрібномасштабної турбулентності для визначення параметрів зондування. Запропонований алгоритм дозволить підбирати швидкість занурення зонда або частоту опитування датчиків для підвищення точності експериментальних даних, які описують турбулізовані шари.*

При комплексных исследованиях океанической среды используемые информационные измерительные системы должны включать специальные средства и методические разработки, позволяющие обеспечить достаточный объем необходимой информации требуемой точности. Поскольку выход научно-исследовательского судна в море для выполнения комплекса экспериментов требует существенных материальных затрат (при этом оплачивается и время нахождения судна в море), то требуется серьезная подготовка к такому мероприятию. Одним из этапов этой подготовки является планирование всех экспериментов.

Основные информационно-измерительные системы для исследований океана используют принцип зондирования водной среды, который позволяет получить распределение гидрофизических параметров по глубине. Возникает необходимость в определении оптимальных параметров зондирования для обеспечения исследователей избыточной информацией заданной точности.

Решение поставленной задачи может быть решено путем использования имитационного моделирования основных параметров тонкой структуры прослоек водной среды, которые затем используются для определения стратегии вертикального зондирования.

Мелкомасштабная турбулентность при вертикальном зондировании характеризуется следующими параметрами:

- количество турбулентных возмущений в заданном диапазоне глубин;
- вертикальные размеры турбулентных возмущений;
- расположение турбулентных возмущений относительно друг друга по глубине.



Под толщиной турбулизированного слоя (вертикальный размер возмущения) понимается разность глубин, в пределах которых амплитуда пульсаций превышает нулевой уровень, что можно определить при вертикальном зондировании. Толщина турбулентного слоя, как и величина пульсаций в нем, может служить параметром для сравнения турбулентных возмущений. Такой слой называют  $\tau$ -слой.

Как показано в [1], статистические совокупности геометрических размеров турбулентных возмущений и вертикальных измерений скорости течения хорошо описываются одно и двух параметрическими экспоненциальными распределениями. Это обусловлено удачно выбранными масштабами пространственного осреднения результатов измерений  $\tau$ -слоев.

Выбор масштабов определяется целями исследования и техническими возможностями информационных измерительных систем.

Автоматическое определение особых точек и слоев гидрофизических полей — одна из сложнейших задач при проведении океанографических исследований. Ее решение требует разработки специальных приемов и методик. Для определения стратегии вертикального зондирования при проведении гидрофизического эксперимента необходимо получить данные о распределении прослоек с мелкомасштабной турбулентностью по глубине, т.е. требуется знать глубину залегания прослойки и ее ширину. По этим данным система может просчитать параметры зондирования — скорость погружения зонда (для систем с регулированием скорости) или частоту опроса первичных преобразователей (для систем, позволяющих управлять частотой опроса) в зависимости от глубины. Организованная таким образом система получения и обработки океанографической информации будет оптимальной.

В статье [2] приведена методика и алгоритм, использующий эту методику для определения глубины залегания прослоек и их толщины.

Анализ полученных результатов моделирования с данными гидрофизических исследований позволяют сделать вывод о возможности использования такой модели распределения. Но, поскольку параметры распределения характеристик тонкой структуры — величины случайные, то в представленной методике их моделирование требует накопления информации о всех проводимых исследованиях данного вида по акваториям океана. Эти данные учитывают количество проведенных измерений и суммарную толщину  $\tau$ -слоев либо расстояний между ними.

Предлагается добавить в методику моделирование количества турбулентных возмущений в заданном диапазоне глубин и использовать этот дополнительный параметр для контроля моделирования параметров распределения прослоек.

Имеется две разновидности  $\tau$ -слоев:

1. расположение  $\tau$ -слоев носит характер случайной последовательности и может быть описан с помощью распределения Пуассона.

$$p_1(n) = \frac{(\mu H)^n}{n!} \exp(-\mu H), \quad (1)$$

$p_1(n)$  — вероятность того, что в слое толщиной  $H$  находится  $n$   $\tau$ -слоев.  $\mu$  — параметр распределения. Его оценка — среднее количество  $\tau$ -слоев на 1 м.

2. расположение  $\tau$ -слоев также случайно, но характеризуется кластерной структурой, то есть группировкой  $\tau$ -слоев по глубине.

В этом случае описание также возможно с помощью распределения Пуассона: сначала с параметром  $\mu_1$  вычисляется распределение вероятностей возможного числа кластеров в диапазоне глубины  $H$

$P_2(n) = \frac{(\mu_1 H)^n}{n!}$ , а затем с параметром  $\mu_2$  вычисляется распределение вероятностей числа  $\tau$ -слоев в кластере  $P_3(n) = \frac{(\mu_2 H)^n}{n!}$ .

В литературе [3] приводится анализ измерений, который показывает, что число  $\tau$ -слоев в кластере не превышает 4–5. Следовательно принять среднее число  $\tau$ -слоев в кластере за 3 не будет большой ошибкой. Тогда, используя условное распределение Неймана типа А, можно вычислить распределение вероятностей числа  $\tau$ -слоев, которые образуют заданное число кластеров.

$$p_4(n/w) = \frac{(\bar{n} w)^n}{n!} \exp(-\bar{n} w), \quad (2)$$

где  $p_4(n/w)$  — условная вероятность того, что в  $w$  кластерах находится  $n$   $\tau$ -слоев;

$\bar{n}$  — среднее число  $\tau$ -слоев в кластере.

$\mu_1$  и  $\mu_2$  — являются случайными величинами, изменяющимися по глубине и горизонтали, таким образом в определенных гидрологических условиях может формироваться вполне определенная совокупность случайных значений  $\mu$ , которая может быть описана распределением вероятностей полученным с помощью Байесовских методов.

$$g_1(\mu/x) = \frac{g(\mu) * l(\mu/x)}{\int g(\mu) * l(\mu/\sigma) * d\mu} \quad (3)$$

$g(\mu)$  — априорная плотность вероятности  $\mu$

$l(\mu/x)$  — функция правдоподобия [4]

$x$  — выборочные данные наблюдаемой случайной величины (в рассматриваемых задачах  $x$  означает количество измерений и сумму случайных величин);

$g_1(\mu/x)$  — апостериорная плотность вероятности  $\mu$

Определение  $g(\mu)$  является наиболее сложной задачей. Это объясняется практически полным отсутствием сведений о значениях  $\mu$  в океане.

Исходя из предположения, что в исследуемой акватории в сезонный период исследований расположение  $\tau$ -слоев носит характер случайной последовательности, используем формулу (1) для моделирования количества  $\tau$ -слоев в диапазоне глубины  $H$ . В качестве параметра  $\mu$  принимаем среднее число  $\tau$ -слоев для данной акватории океана. Далее выполняется алгоритм, реализующий методику, описанную в [2]. Наряду с распределением прослоек производится подсчет числа  $\tau$ -слоев. Полученный результат сравнивается с промоделированной величиной  $n$ . Если расхождение велико, то процесс моделирования параметров распределения прослоек повторяется.

**Выводы.** Апробирование данного алгоритма позволило:

- снизить погрешность параметров распределения турбулизированных прослоек до 3,9 %;
- с большей точностью определять стратегию проведения вертикальных зондирований океанических водных масс;
- повысить точность измерений тонкой структуры.

Следовательно, предлагаемая методика может быть использована при гидрофизических исследованиях.

### **Литература**

1. Парамонов А. Н. Автоматизация гидрофизического эксперимента. — Л.: Гидрометеиздат. — 1982. — 224 с.
2. Зори А. А., Савкова Е. О. Использование имитационного моделирования для определения характеристик мелкомасштабной турбулентности. Наукові праці ДонНТУ. Серія: “Обчислювальна техніка та автоматизація”. — 2004. — С. 326–331.
3. Поздынин В. Д. Элементы вероятностного описания мелкомасштабной турбулентности в океане. — Л.: Гидрометеиздат. — 1989. — 52 с.
4. Поздынин В. Д. Мелкомасштабная турбулентность в океане. — М.: Наука. — 2002. — 202 с.

## АППАРАТНАЯ И ПРОГРАММНАЯ ОРГАНИЗАЦИЯ СИСТЕМЫ МОДЕЛИРОВАНИЯ ЭЛЕКТРОХИМИЧЕСКОГО ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА

Н.А. Киктев, Н.И. Чичикало

Донецкий национальный технический университет, Украина

### *Анотація*

*В статті розглядаються питання програмної та технічної реалізації комп'ютерного моделювання технологічного процесу електрохімічного отримання карбонатів для створення АСУ ТП. Всі функції АСУ ТП електрохімічного розчинення реалізуються програмно. Програмне забезпечення створено на об'єктно-орієнтованій алгоритмічній мові Visual Basic 6.0 з використанням керуючих процедур на мові символічного кодування Асемблер.*

На основании анализа и синтеза объекта управления — технологического процесса получения углекислых солей — определены следующие функции программного обеспечения АСУ ТП: классификация ситуаций в автоматизированном технологическом комплексе с учетом хода реального времени, команд оператора и событий на объекте управления; выдача советов оператору по управлению в соответствии с правилами ведения технологического процесса; управление автоматически выполняемыми функциями АСУ ТП ЭХР в зависимости от текущей ситуации.

В АСУ ТП ЭХР возможны нормальные и аварийные ситуации. Признаками аварийной ситуации являются превышение предельных значений параметров процесса — температуры, плотности тока, кислотности (рН), уровня электролита. При одновременной фиксации обеих аварийных ситуаций более приоритетным является нарушение по току. Технологический процесс может быть остановлен оператором только по завершении полного технологического цикла.

Все функции АСУ ТП ЭХР реализуются программно. Программное обеспечение написано на языке Visual Basic 6.0 с использованием управляющих процедур на языке символічного кодирования Асемблере. В состав специального программного обеспечения входят алгоритмы первичной обработки информации, расчета неизменяемых величин, непосредственного цифрового управления по стандартному ПИД-закону регулирования, программно-логического управления, определения времени окончания рабочей стадии.

Предусмотрено два режима функционирования системы — информационно-советующий, при котором часть функций управления

выполняет оператор, воздействующий на основные устройства с помощью команд дистанционного управления, и автоматический режим непосредственного цифрового управления. Система управления электрохимическим получением карбонатов может входить в состав распределенных иерархических систем как подсистемы нижнего уровня. Для этого предусмотрена возможность обмена информацией между системой и ЭВМ вышестоящего уровня, выполняющего функции координатора. Общая структура ПО (рис. 1) включает общесистемное программное обеспечение (ОСПО: операционная система (ОС) Windows'98, сервисные программы обслуживания аппаратных средств), предназначенное для обеспечения функционирования технических и программных средств АСУ ТП ЭХР; прикладное программное обеспечение общего назначения (ППО ОН: система программирования Visual Basic 6.0, СУБД Visual FoxPro 6.0, графическая среда Corel Draw 7.0), предназначенное для обеспечения функционирования и корректировки специального ПО и специальное прикладное программное обеспечение (СП ПО).

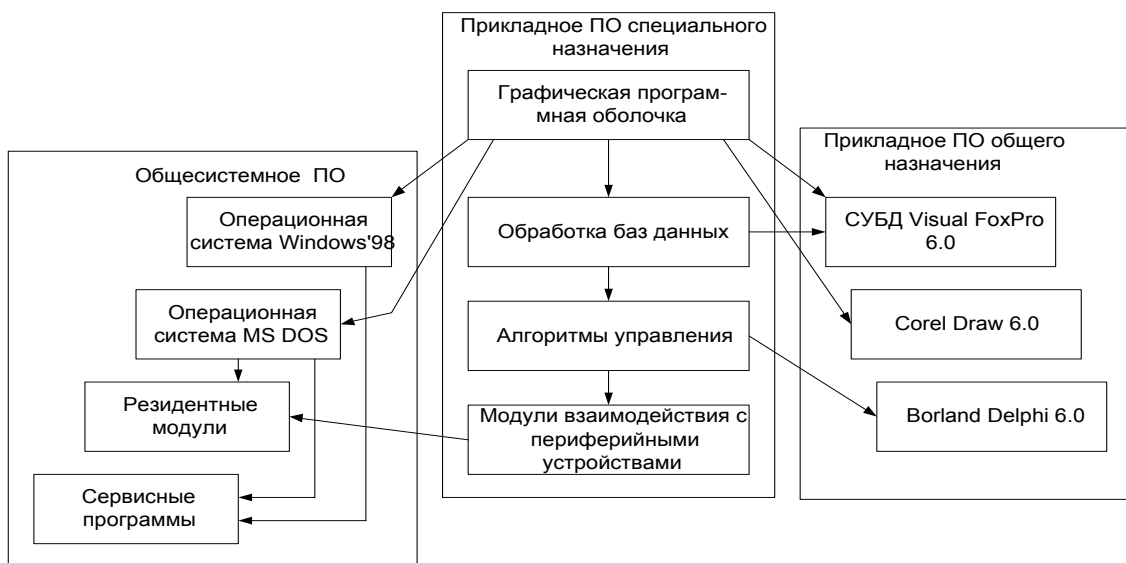


Рисунок 1 — Укрупненная структура ПО АСУ ТП ЭХР

СП ПО (рис.2) включает программную реализацию алгоритмов управления технологическим процессом получения карбонатов, графическую программную оболочку, обеспечивающую интерактивный режим работы оператора-технолога, программные модули взаимодействия ЭВМ с периферийными устройствами.

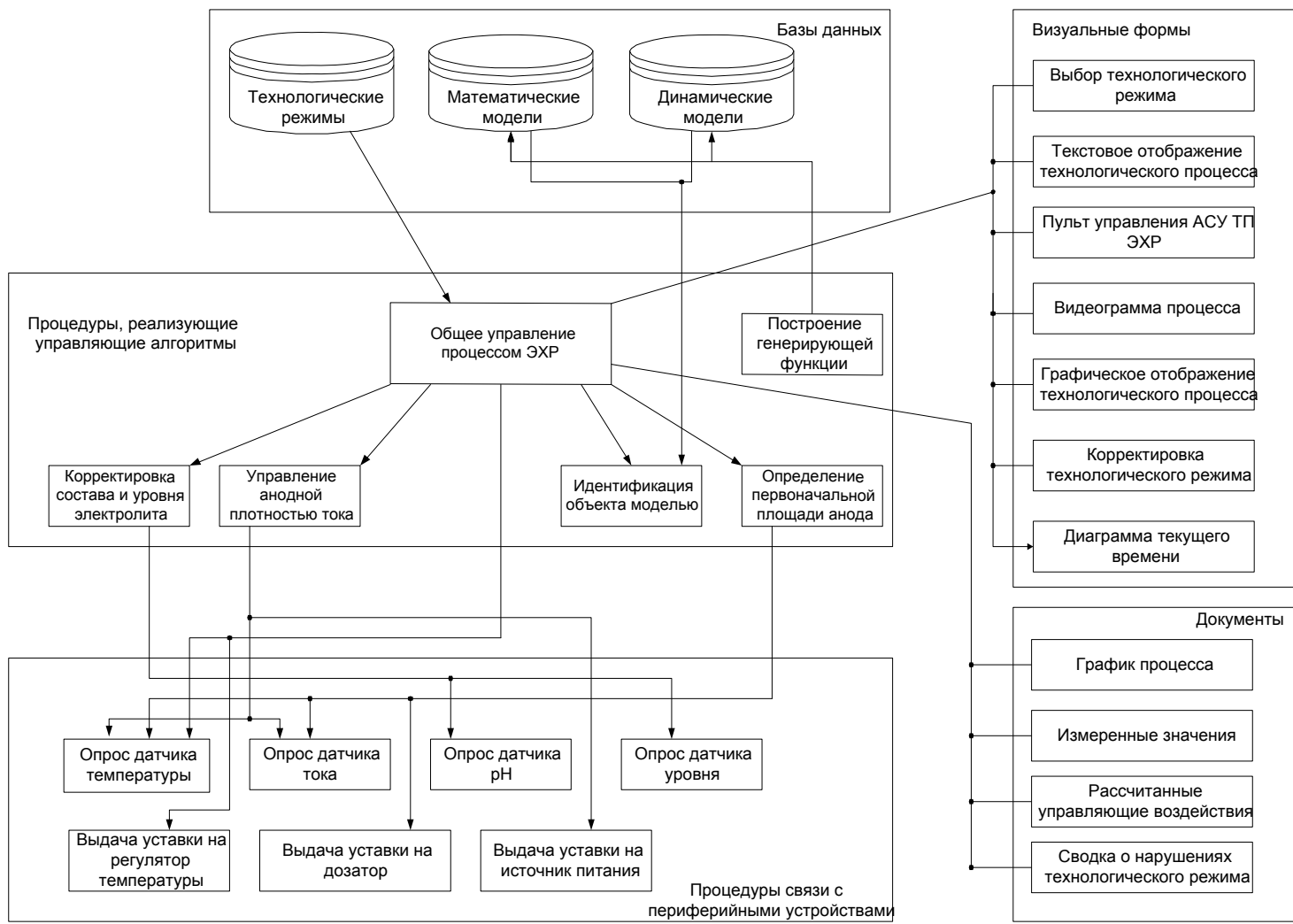


Рисунок 2 — Структура специального прикладного программного обеспечения АСУ ТП ЭХР

Общий алгоритм управления включает следующие модули-процедуры: алгоритмы контроля температуры, кислотности, тока, определения площади загрузки, управления током, алгоритм формирования сводки об отклонениях технологических параметров за смену, алгоритм графического состояния ТП, алгоритм выборки генерирующей функции из БД. С целью визуального контроля технолога за ходом технологического процесса разработана графическая программная оболочка, обеспечивающая интерактивный режим задания начальных условий процесса, использование моделей и технологических параметров, предварительно занесенных в банк данных. Предусмотрено текстовое и графическое отображение состояния процесса. Программная оболочка, созданная при помощи средств visual-программирования, представляет собой windows-приложение, позволяющее одновременно наблюдать за ходом множества параметров технологического процесса. Подготовка данных о технологическом режиме, считываемых из технологических баз данных, осуществляется в статическом режиме до начала технологического процесса с последующим переносом в текстовый файл данных. Данные отображаются на экран и сохраняются в выходном текстовом файле для последующего анализа.

Непосредственно перед запуском системы управления процессом анодного растворения на экране появляются данные о фактическом состоянии параметров, которые обновляются по мере их опроса. Кроме численного отображения состояния процесса программная оболочка предусматривает графический режим. В этом случае пользователь выбирает параметры и отображает в графическом режиме (один или несколько). Под графиком отображается диаграмма, которая показывает нахождение текущего параметра процесса в области допустимых значений и вне ее. Для удобства управления процессом предусмотрен виртуальный пульт. Алгоритмы управления в реальном времени разработаны в соответствии с методами и включают последовательный циклический опрос средств измерений, считывание и обработку исходной информации, запуск ассемблерных процедур и функций, вычисляющих значения зависимых переменных. Информационный обмен данными и результатами программ, работающих в реальном времени с СУБД, осуществляется через файлы данных. Предусмотрено текстовое и графическое отображение состояния процесса. Логические алгоритмы созданы в среде Visual Basic 6.0.

Схема управления технологическим процессом получения карбонатов обеспечивает: приготовление раствора электролита, подготовку лома металлов для анода, электрохимическое растворение металла в растворе солей натрия, фильтрацию и сушку осадка. Комплекс технических средств АСУ ТП электрохимического растворения (КТС АСУ ТП ЭХР) включает в себя персональную электронно-вычислительную ма-

шину, устройства ввода-вывода информации, устройства связи с объектом (УСО). ОС АСУ ТП ЭХР должна обеспечивать решение нескольких задач в реальном масштабе времени, организацию приоритетного обслуживания заявок, возможность включения новых задач в систему. Комплекс периферийных устройств должен обеспечивать реализацию следующих функций: ввод информации от ОУ, распечатку документов и технологической информации, осуществление аварийной сигнализации, выдачу управляющих воздействий на ОУ, оперативный обмен информацией оператора с ЭВМ. Существующие автоматические гальванические линии не приспособлены к управлению от ПЭВМ. Для автоматизированного управления электрохимическим производством от ПЭВМ необходимо решить ряд задач, важнейшей из которых является обеспечение максимальной скорости процесса. Для ее решения необходимо строгое соблюдение технологического режима — поддержание на заданном уровне анодной плотности тока, температуры, кислотности и уровня. Заданная плотность тока обеспечивается подачей на объект управляющего воздействия, технически реализуемого реверсивным тиристорным преобразователем, включающим: ДТ — датчик тока, НУ — нормирующий усилитель, АК — аналоговый коммутатор, АЦП — аналого-цифровой преобразователь, БГР — блок гальванической развязки, ЦАП — цифро-аналоговый преобразователь, ЗДВ — задатчик времени, ФВК и ФВА — фазовращатели катода и анода, ФИК и ФИА — фазовращатели импульсов, КФК и КФА — ключи фазовращателей, СБ — силовой блок, А и К — анод и катод. При рассогласовании текущего значения плотности тока  $D_a = I_{изм} / S_a$  ( $I_{изм}$  — измеренное значение тока,  $S_a$  — площадь анода) с заданным по технологическому режиму, определяется требуемое значения тока  $I_3$ , которое устанавливается тиристорным преобразователем в цепи анода.

Определение площади анодной массы в мерной ванне предполагает управление токовым режимом ванны непосредственно от ЭВМ. Для управления выпрямительным агрегатом типа ВАКР используется выход на дистанционное управление. При этом вместо переменного резистора задатчика тока подключается ступенчато изменяющий величину сопротивления резистор с управлением величиной сопротивления от ЭВМ

Для обеспечения температуры электролита, соответствующей технологическому регламенту, и уменьшения времени переходного процесса при установлении требуемого значения температуры принята двухконтурная схема регулирования температурой электролита: УУ — устройство управления, УСогл — устройство согласования, МПС — микропроцессорная система, УСО — устройство связи с объектом, ЛР — локальный регулятор, К — кондиционер.



Внешний контур обеспечивает в помещении, где расположена гальваническая ванна, температуру согласно технологическому регламенту. Внутренний контур обеспечивает требуемую температуру электролита непосредственно в ванне. Температура воздуха в помещении обеспечивается установкой двух устройств кондиционирования типа SHARP AE-A099E, рассчитанных на площадь помещения  $30\text{ м}^2$  каждый, что удовлетворяет площади помещения цеха  $50\text{ м}^2$ . Управление работой кондиционера осуществляется от ЭВМ путем выдачи уставки на требуемую температуру, выбираемую из базы данных для производимого в этот момент продукта. Выключение устройства осуществляется автоматически при достижении требуемой температуры воздуха. Для измерения температуры электролита (до  $100^\circ\text{C}$ ) лучше всего использовать термометры сопротивления типов ЭТМ — X или ЭТМ — XIV, имеющие защитные кожуха из нержавеющей стали, которые включены в мостовую схему.

При нарушении уровня электролита ЭВМ рассчитывает необходимое количество электролита. Опрос датчика уровня осуществляется при наступлении момента опроса уровня или при необходимости корректировки состава электролита (нарушение режима кислотности). В первом случае осуществляется добавление электролита стандартного состава, а во втором — в составе одного корректирующего компонента: ДУ — датчик уровня, УН — усилитель напряжения, ПФ — полосовой фильтр, НУ — нормирующий усилитель, МУ — масштабирующий усилитель, МПС — микропроцессорная система, СК — сетевая карта, ДЖК — дозатор жидких компонентов

Управление кислотностью электролита осуществляется согласно экспериментально установленной зависимости  $\text{pH}=\text{f}(\text{C}_1, \text{C}_2)$ , где  $\text{C}_1$  и  $\text{C}_2$  — концентрации составляющих компонентов электролита —  $\text{NaHCO}_3$  и  $\text{Na}_2\text{CO}_3$ . При нарушении кислотности ЭВМ получает сигнал от датчика и формирует управляющее воздействие по массе недостающего компонента, которое передается на станцию дозирования: ДрН — датчик кислотности, НУ — нормирующий усилитель, ВП — вторичный преобразователь, УСО — устройство связи с объектом, ДЖК1 и ДЖК2 — дозатор жидких компонентов, СК — смеситель компонентов. Для контроля pH раствора от ЭВМ в ваннах установлены датчики pH типа ДПГ-3М. Сигналы от датчиков через модули УСО вводятся в ЭВМ.

Учитывая поставленные требования к КТС АСУ ТП ЭХР, а также современный уровень развития вычислительной техники предлагается использовать ПЭВМ типа Pentium с тактовой частотой 266 МГц, оперативной памятью 32 Мбайт, емкостью жесткого диска 4 Гбайта, монитором SVGA 15'' с разрешением 1320 dpi. Операционная система — Windows'95, обеспечивающая многооконный режим работы. На вход контроллера

ADAM-4017 поступает аналоговый сигнал 0...2.5 В, выход контроллера ADAM-4024 содержит 14-разрядные аналоговые сигналы 0...2.5 В, 0.1 А. Преобразователь ADAM-4520 позволяет подключать до 256 модулей удаленного ввода-вывода. 485-й интерфейс позволяет связывать объекты на расстоянии до 1,2 км. Скорость передачи указанных средств — до 115 Кбайт/с, частота обновления буфера — 10 раз/с. Структурная схема комплекса технических средств АСУ ТП ЭХР приведена на рис. 3.

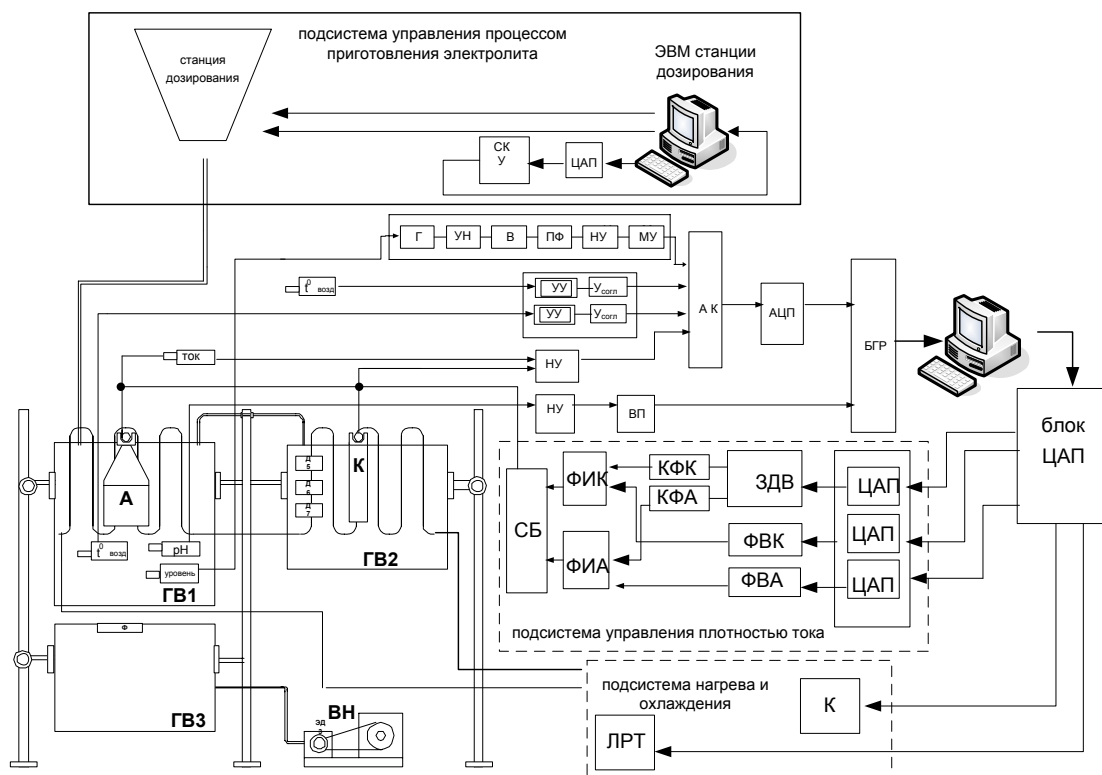


Рисунок 3 — Аппаратная реализация системы управления процессом ЭХР

Таким образом, представленная структура АСУ ТП ЭХР обеспечивает получение углекислых солей, соответствующих стандартам, благодаря точному соблюдению технологического регламента на протяжении всего процесса электрохимического растворения.

### Литература

1. Петров Ю.П. Новые главы теории управления и компьютерных вычислений. — СПб, БХВ-Петербург, 2004. — 240 с.
2. Смоленцев Г.П., Смоленцев М.Г. Автоматизация процессов электрохимического маркирования изделий. Сборн. научн. трудов Самарского аэрокосмич. универс. им. акад. С.П.Королева., 2000. — С. 64–66.
3. Hawhee, Howard, moore Thomas. Programming Language — Visual BASIC. Riders Publishing, 1999. — 1202p. ISBN: 0735700028.

## МОДЕЛИРОВАНИЕ И АППАРАТНАЯ ОРГАНИЗАЦИЯ СИСТЕМЫ КОНТРОЛЯ МАГНИТНОЙ НЕСИММЕТРИИ ЭЛЕКТРИЧЕСКОЙ МАШИНЫ

Е.Ю. Ларина, В.Ю. Ларин, Д.А. Данько

Донецкий национальный технический университет

### *Анотація*

*У статті запропонований метод контролю стану електричної машини, наприклад, асинхронного двигуна за фактором магнітної несиметрії, рівень якої залежить від наявності ушкоджень в обмотках ротора, реалізований шляхом моделювання поточного стану об'єкта на підставі інформації, що надходить від датчиків.*

Известно большое количество принципов построения систем управления асинхронным двигателем [1]. В современной теории векторного управления используется, например, принцип прямого управления моментом. Его суть заключается в использовании в системе управления передаточных функций, обратных по отношению к передаточным функциям векторной модели АД. Это позволяет получить в качестве независимых входных переменных системы величины, входящие в уравнение электромагнитного момента. Однако ни одна из известных моделей не учитывает влияние насыщения путей магнитных потоков на электромагнитные параметры двигателя.

Как известно, из-за электромагнитной несимметрии ротора наблюдается колебание режимных параметров во времени относительно средних значений. При этом пульсации тока создают пульсирующее падение напряжения в отдельных узлах системы. Это отрицательно сказывается на работе нагрузок. Контроль уровня несимметрии электрической машины позволит своевременно устранить его причину за счет поддержания оптимальных параметров в сложных режимах управления и уменьшить отрицательные последствия [2]. Эта проблема не решена до настоящего времени и является актуальной.

Если рассматривать магнитное поле идеальной асинхронной машины, то характеристика распределения магнитного поля во времени (магнитодвижущей силы МДС) имеет вид идеальной синусоидальной формы. Всякое отличие формы МДС АД от синусоидальной характеризует неравномерность магнитного поля двигателя. Факторы, вызывающие неравномерность магнитного поля, приведены ниже. Классификация дает возможность определить стадии учета факторов и выделить те, которые целесообразно контролировать (рис. 1). Конструктивные факторы, создающие высшие гармоники ЭДС, вызывают в электрических машинах ряд нежела-

тельных явлений и могут быть устранены лишь на стадии проектирования и скорректированы после ремонта, чтобы приблизить к идеальной синусоидальной форму распределения магнитного поля.

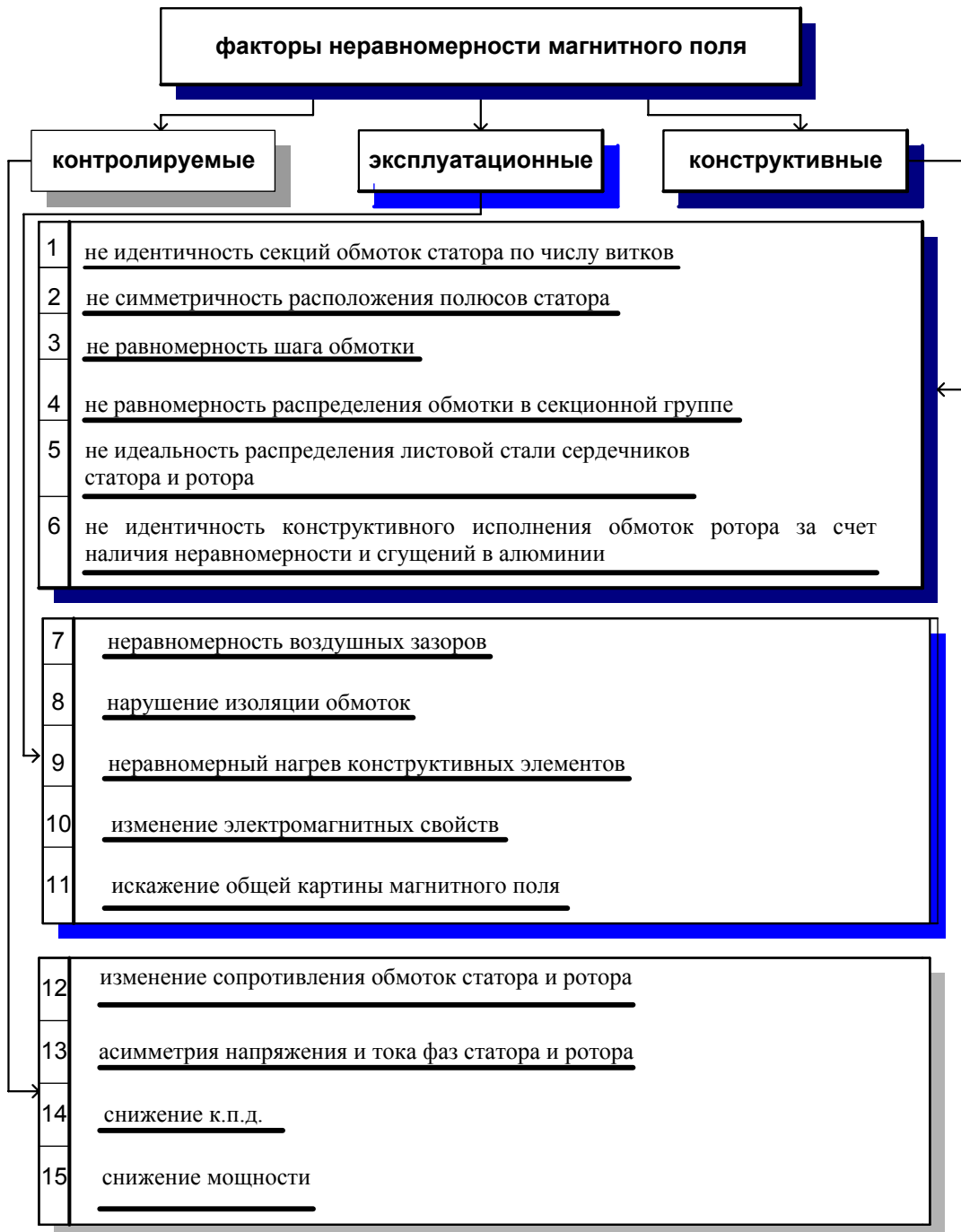


Рисунок 1 — Классификационная схема факторов, вызывающих неравномерность магнитного поля

Эксплуатационные — могут возникать из-за нарушений условий эксплуатации, в частности воздействий непредвиденного уровня вибра-

ций, наличия ударных нагрузок, повышенной влажности, воздействия высокочастотных помех, технологического неравномерного перегрева и т.п. Эксплуатационные факторы могут быть учтены при наличии своевременной информации об их появлении. Контролируемые — обусловлены отклонением режимов работы АД от номинальных. Для их устранения система управления должна получать соответствующую информацию от средств контроля в процессе работы АД.

Чтобы определить возможность увеличения информативности о параметрах короткозамкнутого АД, таких, как ток ротора или магнитный поток, рассмотрим процесс взаимодействия параметров электромагнитной системы АД.

В АД магнитное поле является переменным, через него происходит взаимодействие между отдельными частями машины и преобразование энергии. Поле статора при вращении пересекает проводники обмотки ротора и наводит в них ЭДС. Так как проводники обмотки ротора короткозамкнуты, то в них возникают токи. Ток в каждом из проводников обмотки ротора, взаимодействуя с полем статора, создаёт электромагнитную силу  $F_{эм}$ . Совокупность  $F_{эм}$  создают электромагнитный момент  $M$ , который приводит ротор во вращение с частотой  $n_2$  в направлении вращающегося поля [1, 3, 4].

Под воздействием подводимого к фазам обмоток статора напряжения  $U_1$  в них возникают токи  $I_1$ , создающие вращающееся магнитное поле (поток  $\Phi_1$ ), большая часть которого сцепляется с обмоткой ротора и с обмоткой статора и составляет основной поток обмотки статора. Часть магнитного потока, которая сцепляется только с обмотками статора, определяет поток рассеяния статора  $\Phi_{\sigma 1}$ . Токи  $I_2$  ротора, вызываемые основным потоком статора  $\Phi_1$ , проходя по проводникам обмотки ротора, создают своё магнитное поле. Часть этого магнитного поля — основной поток ротора  $\Phi_2$  сцепляется с обмотками ротора и обмотками статора. Другая часть — поток рассеяния ротора сцепляется только с витками обмотки ротора. Основные потоки ротора  $\Phi_2$  и статора  $\Phi_1$  вращаются в пространстве с одинаковой (синхронной) частотой  $n_1$ . Сумма двух полей с потоками  $\Phi_1$  и  $\Phi_2$  составляет основное поле  $\Phi$  асинхронного двигателя:  $\Phi = \Phi_1 + \Phi_2$ . Основной поток АД, вращаясь в пространстве, пересекает обмотку статора с синхронной частотой  $n_1$  и обмотку ротора с частотой скольжения  $n_s = n_1 - n_2$ . Ток статора измерить несложно, это решенная задача. Измерить ток ротора — сложная задача. При наличии современной элементной базы измерять его возможно при условии встраивания параллельно каждой обмотке ротора дополнительно один не замкнутый накоротко виток, подключенный к интеллектуальному датчику, настроенному на дискретное срабатывание в случае

снижения на определенную величину силы магнитного потока в обмотке ротора или настроенного на аналоговый контроль уровня снижения магнитного потока. На рис. 2 представлена структурная схема преобразования электрической мощности в механическую на валу двигателя и показан способ контроля магнитного потока обмотки ротора с помощью чувствительного элемента ЧЭ и измерительного передатчика ИП, образующих интеллектуальный датчик. Информация может быть передана беспроводным способом. Если присвоить каждому датчику номер контролируемой фазы, то можно в реальном времени контролировать и идентифицировать равномерность (или уровень неравномерности) магнитного поля ротора.

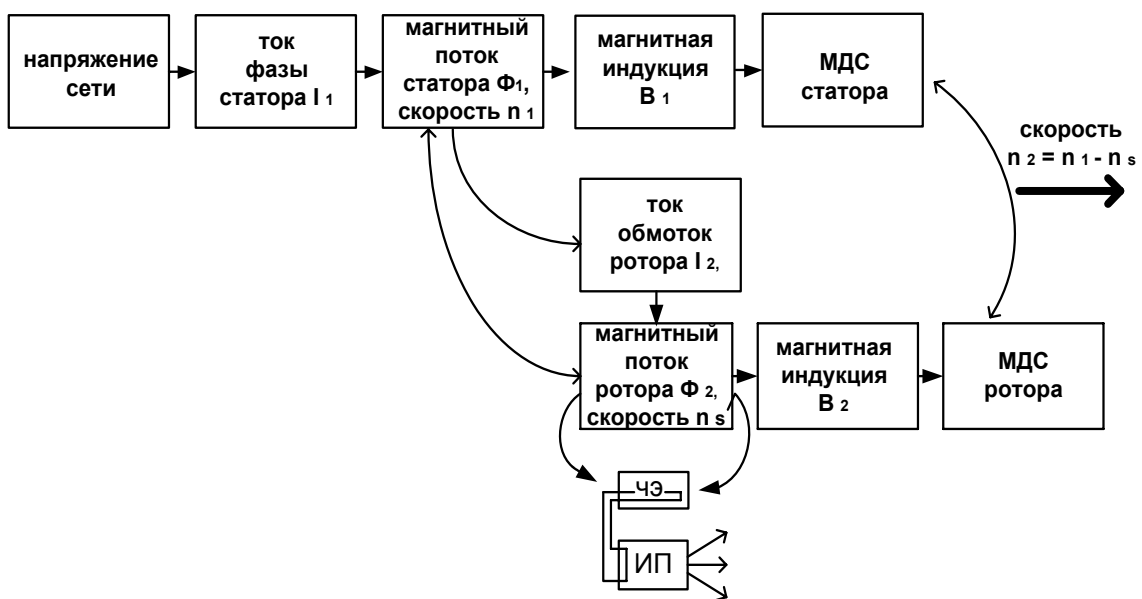


Рисунок 2 — Способ контроля магнитного потока обмотки ротора

Представленный анализ также делает очевидным возможность решения поставленной задачи путем определения скорости скольжения. Устройство для решения этой задачи представлено на рис. 3.

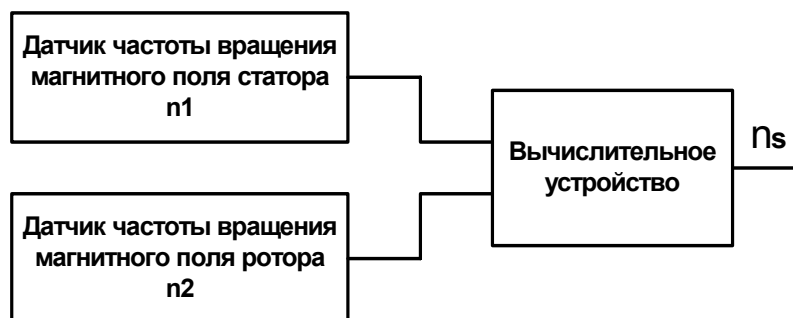


Рисунок 3 — Устройство контроля скорости скольжения АД

Рассмотрим влияние ЭДС первой гармоники поля (табл. 1).

Таблица 1 — Характеристики ЭДС первой гармоники

ЭДС в каждом проводнике витка обмотки ротора	$e_{\text{пр}} = E_{\text{мпр}} \sin \omega t$ , где $E_{\text{мпр}}$ — амплитуда ЭДС в проводнике, $\omega$ — циклическая частота первой гармоники тока. $E_{\text{мпр}} = 2f B_{\text{м1}} l \tau$ , где $f$ — частота тока, вызвавшего индукцию $B_{\text{м1}}$ , $B_{\text{м1}}$ — амплитуда индукции основной гармоники поля в зазоре, $l$ — расчётная активная длина машины, $\tau$ — полюсное деление.
Действующее значение ЭДС	$E_{\text{пр}} = \frac{E_{\text{мпр}}}{\sqrt{2}} = \sqrt{2} f B_{\text{м1}} l \tau$
ЭДС витка	$E_{\text{в}} = 2 E_{\text{пр}} \sin \frac{\beta \pi}{2}$ , где $\beta$ — относительный шаг витков катушки с укороченным шагом, $\beta = \frac{y}{\tau}$ , где $y$ — шаг обмотки, $\tau$ — полюсное деление
ЭДС катушки	$E_{\text{к}} = 4.44 f \omega_{\text{к}} k_{\text{у}} k_{\text{с}} \Phi$ , где $f$ — частота тока промышленной сети, $\omega_{\text{к}}$ — число витков катушки, $k_{\text{у}}$ — коэффициент укорочения, $k_{\text{с}}$ — коэффициент скоса пазов обмотки, $\Phi$ — поток одного полюса при синусоидальном распределении индукции
ЭДС катушечной группы	$E_{\text{г}} = 4.44 f q \omega_{\text{к}} k_{\text{об}} k_{\text{с}} \Phi$ , где $q$ — число катушек одной фазы, $k_{\text{об}}$ — обмоточный коэффициент учитывающий укорочение шага и распределение обмотки, $k_{\text{об}} = k_{\text{у}} k_{\text{р}}$ , где $k_{\text{р}}$ — коэффициент распределения обмотки
ЭДС фазы обмотки	$E = 4.44 f \omega_{\text{к}} k_{\text{об}} k_{\text{с}} \Phi$ .
Число витков фазы при последовательном соединении витков каждой параллельной ветви катушки	$\omega = n q \omega_{\text{к}}$ , где $n$ — количество катушечных групп
Число витков фазы, если m-фазная обмотка заполняет все z пазов и имеет a параллельных ветвей	$\omega = \frac{z s_{\text{п}}}{2 a m}$ , где $s_{\text{п}}$ — число эффективных проводников в фазе.
Число витков трёхфазной обмотки	$\omega = \frac{p q s_{\text{п}}}{a}$ , где $q$ — число пазов на полюс и фазу, $q = \frac{z}{2 p m}$ , $p$ — число пар полюсов.

Каждая гармоника создаёт свою ЭДС. Если эти ЭДС имеют различные частоты, то действующее значение суммарной ЭДС определяется как  $E_{\Sigma} = \sqrt{E_1^2 + E_3^2 + E_5^2 + \dots + E_v^2 + \dots}$ , где  $v$  — нечётное целое положительное число, номер гармоники. Высшие гармоники ЭДС могут вызывать в электрических машинах ряд нежелательных явлений, о которых указывалось выше. Для моделирования этого процесса возьмем реальные данные АД (табл. 2).

Таблица 2 — Данные для моделирования и анализа процесса МДС 1-, 3-, 5-й гармоник и основной МДС магнитного поля

Частота тока промышленной сети:	$f := 50, \omega := 2\pi \cdot f$
Номер гармоники: Число витков:	$w := 500, v := \begin{pmatrix} 1 \\ 3 \\ 5 \end{pmatrix}$
Обмоточный коэффициент: Число зубцов:	$коб := 1.2$ $z := 30$
Коэффициент скоса пазов обмотки: Число пар полюсов:	$кс := 1.4$ $p := 4$
Число зубцовых делений: Активная длина машины:	$y := 1$ $l\delta := 0.8$
Действующее значение тока одной фазы: Магнитный поток первой гармоники одного полюса статора при синусоидальном распределении индукции для симметричной нагрузки: Полюсное деление:	$I := 1000$ $\Phi 1 := \frac{2}{\pi} \cdot Bm1 \cdot \frac{\tau}{1} \cdot l\delta$ $\tau := \frac{z}{2p}, \tau = 3.75$
ЭДС фазы: Число фаз:	$E := 4.44f \cdot w \cdot коб \cdot кс \cdot \Phi 1$ $m := 3$
$E = \blacksquare$	Ток трёхфазной обмотки симметричный
Полюсное деление:	$q := \frac{z}{2p \cdot m}, q = 1.25$
Шаг катушки:	$\beta := \frac{y}{\tau}, \beta = 0.267$
Коэффициент распределения:	$kp1 := \frac{\sin\left(\frac{v_{0,0} \cdot \pi}{2m}\right)}{q \cdot \sin\left(\frac{v_{0,0} \cdot \pi}{2m \cdot q}\right)}, kp1 = 0.983$ $kp3 = 0.841, kp5 = 0.462$
Зададимся временем модуляции:	$t := 0, 10^{-4} .. 1$



Продолжение табл. 2

Коэффициент укорочения шагов обмотки:	$ky1 := \sin\left(\frac{\nu_{0,0} \cdot \beta \cdot \pi}{2}\right)$ , $ky1 = 0.407$ $ky3 = 0.951$ $ky5 = 0.866$
Обмоточный коэффициент:	$коб1 := kp1 \cdot ky1$ , $коб1 = 0.4$ $коб3 = 0.8$ , $коб5 = 0.4$
Мгновенное значение тока фазы а, b и с:	$ia(t) := \sqrt{2}I \cdot \sin(\omega \cdot t)$ $ib(t) := \sqrt{2}I \cdot \sin\left(\omega \cdot t - \frac{2\pi}{3}\right)$ $ic(t) := \sqrt{2}I \cdot \sin\left(\omega \cdot t - \frac{4\pi}{3}\right)$
Амплитуда МДС первой гармоники фазы:	$F\phi1 = 4.5 \times 10^4$ , $F\phi3 = 3 \times 10^4$ $F\phi5 = 9 \times 10^3$
МДС фазы (рис. 4):	
<p>Рисунок 4 — Вид гармоник и основной МДС магнитного поля</p>	
$F\phi(t) := F\phi1 \cdot \cos(\omega \cdot t) \cdot \cos\left(\nu_{0,0} \cdot \frac{\pi}{m}\right) + F\phi3 \cdot \cos(\omega \cdot t) \cdot \cos\left(\nu_{1,0} \cdot \frac{\pi}{m}\right) + F\phi5 \cdot \cos(\omega \cdot t) \cdot \cos\left(\nu_{2,0} \cdot \frac{\pi}{m}\right)$	
Расчётная активная длина машины:	$l := 0.68$
Амплитуда индукции первой гармоники магнитного поля в зазоре:	$Bm11 := 1$ , $Bm11\_5 := 1.5$ , $Bm12 := 2$
Амплитуда ЭДС в проводнике:	$Empp1 := 2f \cdot Bm11 \cdot l \cdot \tau$ , $Empp1\_5 := 2f \cdot Bm11\_5 \cdot l \cdot \tau$ $Empp2 := 2f \cdot Bm12 \cdot l \cdot \tau$
Время:	$t := 0, 10^{-4} .. 1$

Продолжение табл. 2

<p>Мгновенное значение ЭДС в проводнике:</p>	$e_{np1}(t) := E_{mnp1} \sin(\omega \cdot t)$ $e_{np1\_5}(t) := E_{mnp1\_5} \sin(\omega \cdot t)$ $e_{np2}(t) := E_{mnp2} \cdot \sin(\omega \cdot t)$
<p>sum(n) = ■ (Рис. 5)</p>	<pre> sum(n) ←   s ← 0              for x ∈ 1.. n              s ← s + 1              </pre>

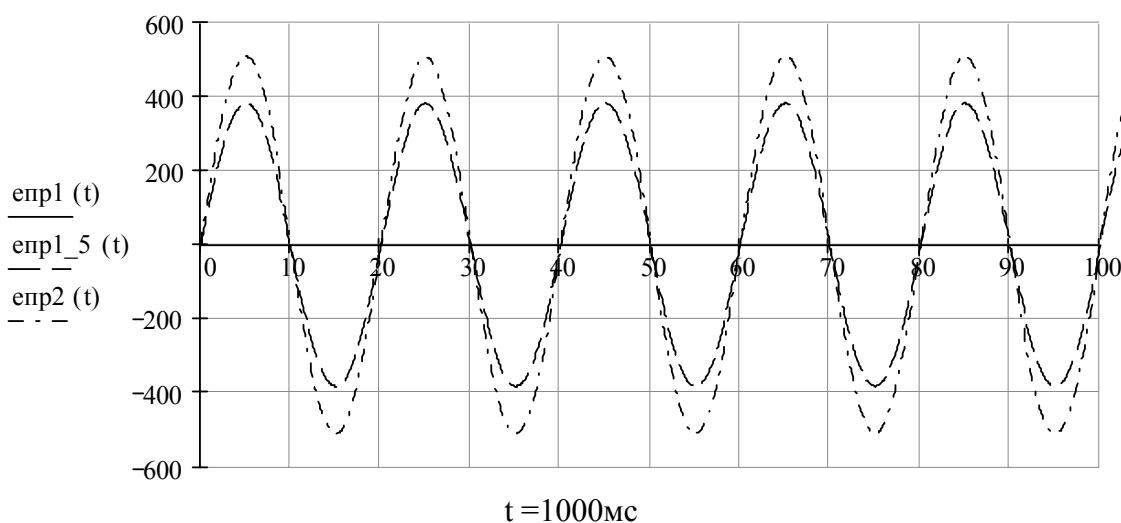


Рисунок 5 — ЭДС трёхфазной обмотки при различных значениях индукции

**Выводы.** Обеспечивается контроль несимметрии магнитного поля ротора с визуализацией на экране дисплея.

### Литература

1. Усольцев А.А. Векторное управление асинхронными двигателями, Учебное пособие по дисциплинам электромеханического цикла. — Санкт-Петербург, 2002. — 430 с.
2. Ларина Е.Ю. Способ и средства учета асимметрии питающего напряжения асинхронных двигателей// Вимірювальна та обчислювальна техніка в технологічних процесах. — 1998. Спец. вип. "Сучасні інформаційні та енергозберігаючі технології життєзабезпечення людини". — №3.— С. 94–97.
3. Алгоритмы технической диагностики регулируемых асинхронных электроприводов Крюков О.В., Марков В.В. Нижегородский государственный технический университет, №4, 2000, Электротехника — С. 36–39.
4. Захаров П.А. Нижегородский государственный технический университет, Захаров А.М. Ивановский государственный энергетический университет, Исследование переходных режимов работы АД, №3, 2000, Электротехника. — С. 58–60.

УДК 681.5

**ИСПОЛЬЗОВАНИЕ МОБИЛЬНЫХ ДИАГНОСТИЧЕСКИХ СИСТЕМ ДЛЯ ВИЗУАЛЬНОГО КОНТРОЛЯ И МОДЕЛИРОВАНИЯ СЦЕНАРИЕВ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ ГАЗОПЕРЕРАБАТЫВАЮЩИХ ПРЕДПРИЯТИЙ**

А.М. Солодкий, А.Н. Тамыев

Астраханский газоперерабатывающий завод

ООО «Астраханьгазпром»

***Анотація***

*У статті відбитий підхід до створення мобільних діагностичних систем стосовно до автоматизованих систем контролю технологічних процесів. Розглянуто проблеми технологічного характеру, що виникають на діючих технологічних процесах, запропоновані шляхи рішення на основі мобільних конструктивів і робітників станцій на основі персональних комп'ютерів. Запропоновано оптимальні форми відображення результатів моделювання на моніторах за допомогою колірних інформаційних матриць.*

Развитие информационных технологий, а также сильная конкурентная борьба в сфере создания аппаратных и программных продуктов заставляет производителя развивать приоритетные направления, в разработках обращенные к конечному пользователю.

В последнее для контроля и управления технологическими процессами газоперерабатывающих предприятий все чаще используются информационно-управляющие системы. Несмотря на неоспоримые преимущества при использовании таких систем их применение связано с особенностью их обслуживания, спецификой перерабатываемых продуктов производств, находящихся под контролем управляющих систем, а также особенностями используемых технологий.

Своеобразие функций управления связано с тем, что некоторые технологии, применяемые в газоперерабатывающей промышленности, используются с высокими температурами, давлениями, что при нарушениях технологических регламентов может привести к выбросам вредных веществ в атмосферу.

Ещё одной опасностью при выполнении технологических операций, например на газоперерабатывающем предприятии, является вынужденный технологический останов, происходящий иногда при аварийном снятии напряжения или остановке технологической установки производящей воздух. Это так называемый «ударный останов», который может привести не только к выбросам вредных веществ в атмосферу, но и к опасности взрыва на перерабатывающем

предприятию, что может сопровождаться человеческими жертвами.

В процессе контроля технологического процесса информационно-управляющей системой часто возникает необходимость в определении виновника нештатной ситуации. Проблемы можно разделить электрические, технологические, проблемы информационно-управляющей системы.

Как правило, производители автоматизированных систем, магистральных информационных шин, межмагистральных соединителей, датчиков полевых приборов разные фирмы, а пользователю приходится решать проблемы по постановке диагноза, определяя, кто является виновником сбоев, помех или динамических нарушений в цепочке передачи информации от измерения полевого прибора до устройства визуализации информационно-управляющей системы.

Эта проблема является особенно актуальной, так как эксплуатационному персоналу необходимо не только правильно поставить диагноз появившегося нарушения в системе, но сделать это необходимо в очень короткое время, оставив промежуток времени на устранение нарушения. Это может быть связано с довольно рискованным сценарием последствия, которого могут быть как минимум останов технологического оборудования или другой сценарий невозможности по необходимости останов технологического оборудования.

Наконец в процессе работы оборудования, например информационно-измерительно-управляющей системы возникает необходимость в подключении в режиме on-line стандартных или нестандартных полевых приборов с использованием режима отладки.

У производителей отсутствует информация, как работает, например полевой прибор, соединители и блоки системы в реальных условиях эксплуатации.

Перечисленные проблемы вызывают необходимость в проработке сценариев, чтобы избежать их в дальнейшем или в короткое время принять по ним правильное техническое решение.

Опыт, накопленный эксплуатационным персоналом Астраханского газоперерабатывающего завода определил приоритетные задачи, стоящие перед обслуживающим персоналом различных служб и производств в части требования к сервисному оборудованию цеха информационно-управляющих систем службы автоматизации, необходимому для обслуживания производств с непрерывным технологическим циклом под контролем современных информационно-управляющих систем. Результатом проделанной исследовательской работы и анализа нештатных технических сценариев при работе технологического оборудования было разработано

устройство для контроля электронных блоков, отвечающее современным требованиям, предъявляемым к оборудованию своего класса.

Разработанное «Устройство для контроля электронных блоков» относится к вычислительной, информационно-измерительной технике и может быть использовано в различных отраслях промышленности для размещения моделирования технических сценариев, настройки электронного и технологического оборудования в различных режимах.

К недостаткам сервисного оборудования аналогичного назначения относится невозможность функционального расширения работы без изменения конструкции, например, проверки технологического оборудования или имитации работы датчиков полевых приборов в составе электронных блоков, располагающихся в конструкции.

Задачи, которые выполнены при разработке «Устройства для контроля электронных блоков» — снижение аппаратных затрат при компактной конструкции, а также расширение функциональных возможностей и создания имитации работы технологического оборудования и проверка электронных модулей.

Решение задач достигается тем, что создается конструкция, в корпус которой вмонтирована корзина, крепящаяся спереди к корпусу болтами, а внутри корпуса опирается на направляющие. При этом передняя часть корпуса выше и ниже габаритов корзины закрыта фальшпанелями, выполняющими декоративные функции.

Сущность технических решений заключается в том, что предлагаемая компоновка конструкции позволяет настраивать электронные модули как автономно, так и в составе технологического оборудования, а также расширять функциональных возможностей путем подключения устройства непосредственно к действующему технологическому процессу.

Конструкция устройства представлена на рис.1.

Устройство содержит корпус 1, корзину для модулей 2 с блоком питания 3, верхнюю фальшпанель 4 с тумблерами 5, нижнюю фальшпанель 6, направляющую 7, клеммник 8, кабель 9, клеммник для подключения блока питания 10, электронные модули 11, нижнюю направляющую 12, кабель 13 и блок вентиляции 14.

Предлагаемое устройство для проверки модулей, включает корпус 1, блок вентиляции 14, расположенный внутри корпуса, блок питания 10 и отличается тем, что в корпус 1 вмонтирована корзина 2 для размещения проверяемых модулей. Ячейки корзины — одного типоразмера. Блок питания 10 располагается слева и занимает две ячейки корзины. Блок вентиляторов для обеспечения теплового режима расположен под корзиной и лицевой частью корпуса закрытой фальшпанелями 4 и 6 выше

и ниже габаритов корзины. В верхнюю фальшпанель 4 вмонтированы тумблеры 5 для имитации работы технологического оборудования и соединяют кабель 13 с клеммниками 8, установленными на направляющие, расположенными между нижними фальшпанелями. На клеммники подсоединяются датчики, например, термопары или термосопротивления, а с другой стороны клеммники подсоединяются витыми парами к проверяемым модулям соответствующих датчиков.

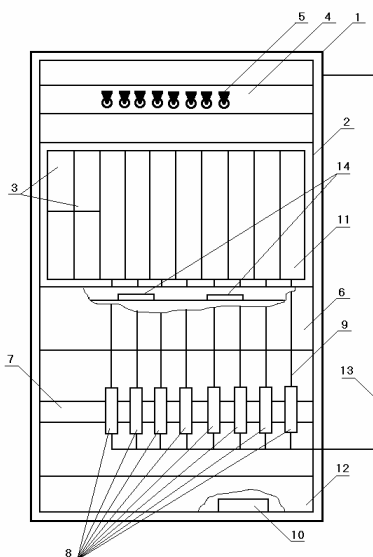


Рисунок 1 — Устройство контроля электронных блоков

Устройство работает следующим образом.

Для проверки электронного модуля, его вставляют в корзину 2, а устройство подсоединяют к персональному компьютеру (кабели подключения не показаны). Структурная схема для контроля электронных блоков в составе автономной рабочей станции показана на рис.2.

При необходимости с помощью тумблеров на верхней фальшпанели и электронного модуля вставленного в корзину 2 устройство позволяет производить автономную наладку электронного оборудования для различных технологических процессов. Подсоединение датчиков к клеммнику дает возможность на устройстве проверять параметры датчиков, а дополнительно вставленный в корзину 2 электронный модуль дает возможность быстро осуществить контроль датчика и диагностику проверяемого электронного модуля. Применение устройства для контроля электронных блоков, подробно описанного в примерах, подтверждают его универсальность, перестраиваемость, экономичность из-за ограниченного набора применяемых стандартных модулей, а также отсутствия специально разрабатываемых приборов и программного обеспечения.



Рисунок 2 — Структурная схема для устройства контроля электронных блоков

Конструкция устройства обладает следующими преимуществами:

- На устройстве можно проверять различные модули в одной и той же ячейке.
- Устройство может вмещать несколько модулей как одинакового, так и разного функционального назначения.
- Устройство, благодаря своей конструкции, имеет возможность для расширения функций контроля.
- К устройству подсоединяются различные датчики для проверки технологических параметров.
- Конструкция устройства позволяет делать имитацию технологического оборудования.
- Вентилизация устройства расширяет возможности проверки электронных модулей, например: проводить проверку технологического оборудования в оптимальных режимах.
- Устройство и легко монтируемая на его основе система моделирования может собираться в различных, что делает её востребованной для полевых информационно-управляющих систем.

Конструкция «Устройство для контроля электронных блоков» защищена патентом на полезную модель и явилась базой для подачи нескольких заявок на изобретения в Федеральный институт промышленной собственности, в частности на изобретение «Система моделирования технологических процессов». Пробная промышленная эксплуатация

«Системы моделирования технологических процессов» показала её эффективность в плане визуального контроля аналоговых и дискретных параметров с помощью цветowych информационных матриц (рис. 3).



Рисунок 3 — Визуальный контроль аналоговых и дискретных параметров с помощью цветowych информационных матриц

Цветовые информационные матрицы отражают массив аналоговых параметров и позволяют оператору технологического процесса и сервисному персоналу контролировать операции, нарушения параметров и сводить к минимуму время принятия решений в случае нештатных или предаварийных ситуаций. При решении научных и технических задач, а также проработке идеологий построения «Системы моделирования технологических процессов» использовался подход к диагностированию, предложенный Зинченко В.И., описанный в работах [1], технологический процесс, а также пользовательский подход к решению проблемных ситуаций, отраженный в источнике [2].

### Литература

1. Зинченко Ю.Е. Т-модель дискретного устройства и решение диагностических задач на ее основе // Вестник ТРТУ – ДонГТУ. Материалы II Международного семинара «Практика и перспективы институционального партнерства». — Донецк, ДонГТУ, 2001, N 1. — С 36–47.
2. Зинченко Ю.Е., Жилин К.Е., Алексеев П.С., Глазков Д.Е.. Диагностирование дискретных устройств на основе имитационного моделирования// Сборник научных трудов ХТУРЭ. — 1999г. — С. 251–2.



# ПОДСИСТЕМА ОПТИМИЗАЦИИ НА БАЗЕ ЭВОЛЮЦИОННЫХ ВЫЧИСЛЕНИЙ ДЛЯ ПАРАЛЛЕЛЬНОЙ МОДЕЛИРУЮЩЕЙ СРЕДЫ

К.С. Теплинский, В.А. Трубаров, В.А. Святный  
Донецкий национальный технический университет

## ***Аннотация***

*Шляхом дослідження, тестування і комбінації різних модифікацій ГА розроблений алгоритм, що може бути ефективно використаний для рішення задач глобальної оптимізації і підтримує як автоматичне, так і ручне налаштування для рішення специфічних задач. Також була впроваджена комбінація ГА і локального детерміністичного методу.*

## ***1 Введение***

В этом докладе рассматриваются проблемы оптимизации в моделировании сложных динамических систем. Основной целью является исследование и разработка различных методов оптимизации и их модификаций для решения задачи идентификации параметров для сложных моделей биологических и технологических процессов. Основными особенностями рассматриваемых моделей являются сложные, нелинейные, мультимодальные целевые функции с большим количеством параметров и ограничений. Такой характер целевых функций требует решения задачи глобальной оптимизации.

## ***2 Реализация и система тестирования***

Разработка высокопроизводительной подсистемы оптимизации предполагает не только реализацию непосредственно численных методов оптимизации, а также их тестирование, интеграцию в среду, подготовку к последующему распараллеливанию. В связи с этим была реализована объектно-ориентированная модель (ООМ) специального программного комплекса (рис. 1), которая включает в себя: систему описания задач, интерфейсы для подключения классов, выполняющих оптимизацию, систему формирования отчетов и сбора статистики экспериментов, тестирующую систему, интерфейс к моделирующей среде, интерфейс к системе распараллеливания.

Для оценки целесообразности применения разработанных алгоритмов оптимизации была разработана специальная система тестирования. Данная система позволяет выполнить predetermined тесты и сформировать развернутый отчет о результатах тестирования, на основе которых можно объективно осуществить выбор необходимого алгоритма и его параметров.

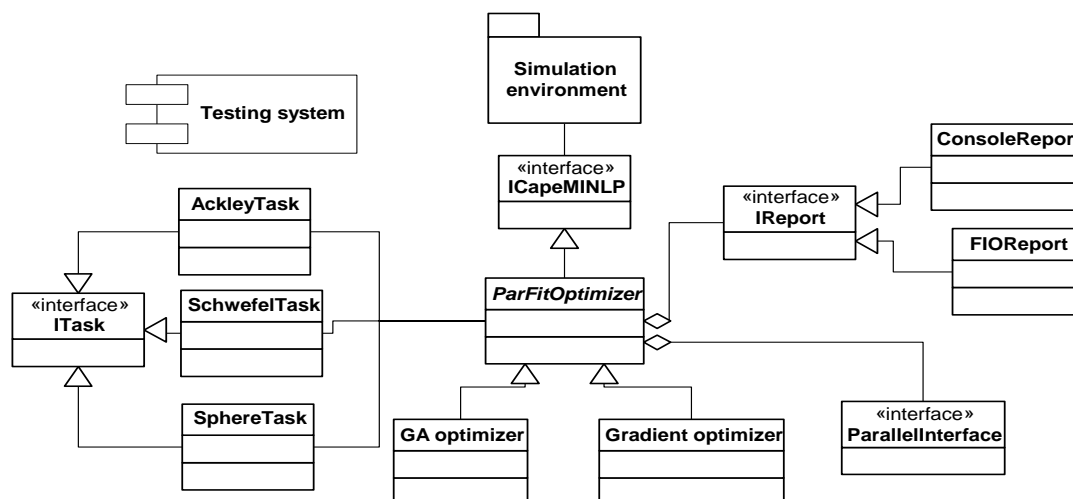


Рисунок 1 — ООМ разработанного программного комплекса

Основным оценочным критерием работы алгоритма является количество вычислений целевой функции, при условии, что решение алгоритма с заданной точностью успешно найдено. В случае с генетическими алгоритмами, содержащими элемент случайности, необходимо оценивать среднестатистические значения производительности и вероятность сходимости алгоритма. Для получения этих значений производится многократный запуск решения задачи с одними и теми же наборами параметров и сбор статистики экспериментов.

В качестве задач для тестирования полученных методов были выбраны искусственные целевые функции, общепринятые для тестирования задач глобальной оптимизации [4, 6]: сфера, степ функция, функция Швевеля и функция Экли.

### ***3 Генетический алгоритм и результаты его тестирования***

#### ***3.1 Биологические аспекты, теория, алгоритм работы***

Основу ГА [3] составляет поиск в многомерном пространстве, который имитирует эволюцию биологических индивидуумов (имеется ввиду научный подход, который базируется на эволюционной теории Дарвина).

Базовой теорией для ГА является теория схем [3]. Понятие схемы было введено для определения множества хромосом, обладающих некоторыми одинаковыми свойствами, т.е. имеющие одинаковые биты в определённых позициях. Эта теория описывает влияние операторов ГА на формирование результата.

Все шаги (операторы) генетического алгоритма — мутация, кроссовер, селекция — имеют множество модификаций, а также представление индивидуумов может быть различным [4–6].

### **3.2 Кроссовер**

Оператор кроссовера является важнейшим оператором генетического алгоритма [4]. Он реализует механизм генетического наследования важных свойств родителей потомками. Используется два основных типа кроссовера: многоточечный и однородный [4–6].

Для тестирования кроссовера были проведены тесты на упомянутых искусственных целевых функциях (более 500000 экспериментов). По результатам экспериментов была выведена зависимость настройки кроссовера от длины генотипа, обеспечивающая наиболее эффективный поиск глобального оптимума (рис. 2а). Длина генотипа (битовой строки представления параметров) определяется количеством параметров целевой функции и степенью их дискретизации. Представленные результаты показаны вместе с изменением размера популяции (рис. 2б), так как они связаны между собой. Таким образом, для эффективного поиска глобального оптимума необходимо с увеличением длины генотипа увеличивать количество точек и вероятность кроссовера (рис. 2а). Вместе с тем необходимо также увеличивать размер популяции, но при увеличении вероятности кроссовера необходимый размер популяции несколько снижается, а потом продолжает расти до следующего увеличения вероятности (рис. 2б).

### **3.3 Мутация**

Оператор мутации является вспомогательным оператором, который инвертирует один случайный бит индивидуума. Мутация может быть применена либо к одному биту индивидуума (одноточечная) либо ко всем битам индивидуума с вероятностью  $p_m$  на бит.

Оператор мутации способствует формированию новых признаков у членов популяции, что позволяет повысить вероятность поиска глобального оптимума. С другой стороны, высокая мутация приводит к дестабилизации процесса поиска и, как следствие, к расходимости алгоритма. Для решения этой проблемы необходимо оценить пороговое значение вероятности мутации. Так как вероятность мутации задается для каждого отдельно взятого гена, а расходимость зависит от вероятности мутации индивидуумов и мощности отбора, то пороговое значение вероятности мутации зависит от длины генотипа. Данная зависимость, установленная в результате тестирования (3000 экспериментов), приведена на рис. 3.

В ходе серии экспериментов, упомянутых в пункте 3.2, была также выведена оптимальная зависимость вероятности мутации от длины генотипа (рис. 3). Данные величины вероятности мутации

помогают в поиске глобального оптимума и вместе с тем не приводят к расходимости. Эти значения меньше граничных, так как принимается во внимание работа кроссовера.

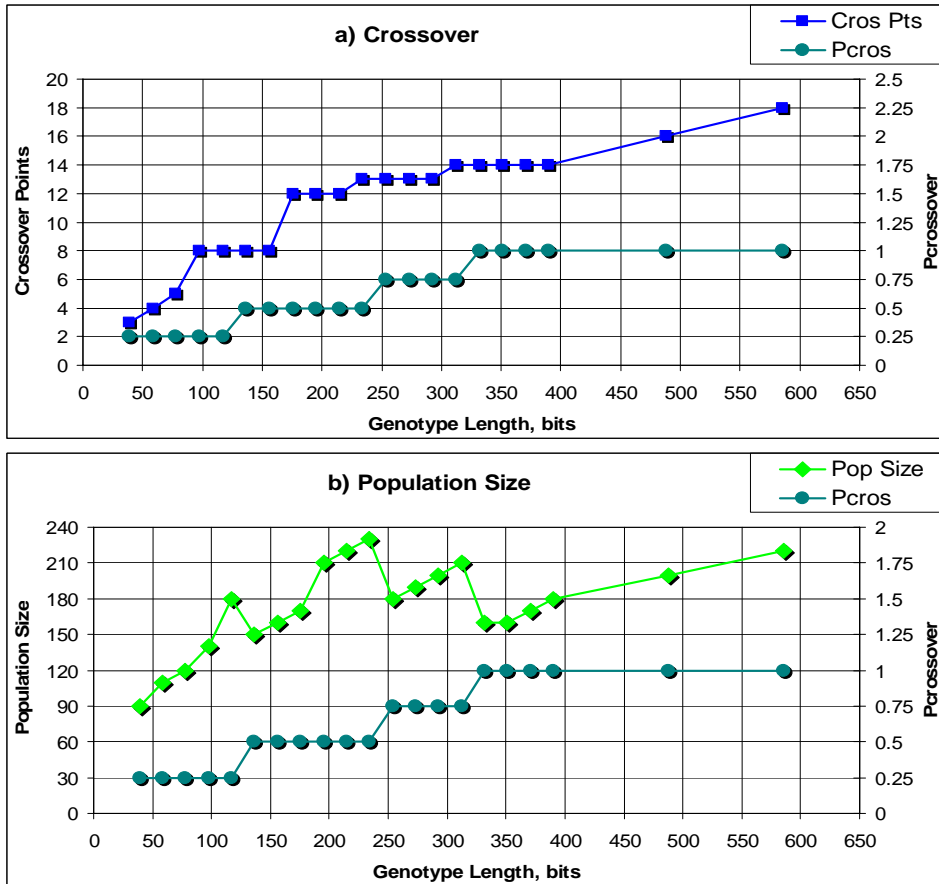


Рисунок 2 — Зависимость кроссовера (а) и размера популяции (б) от длины гено типа

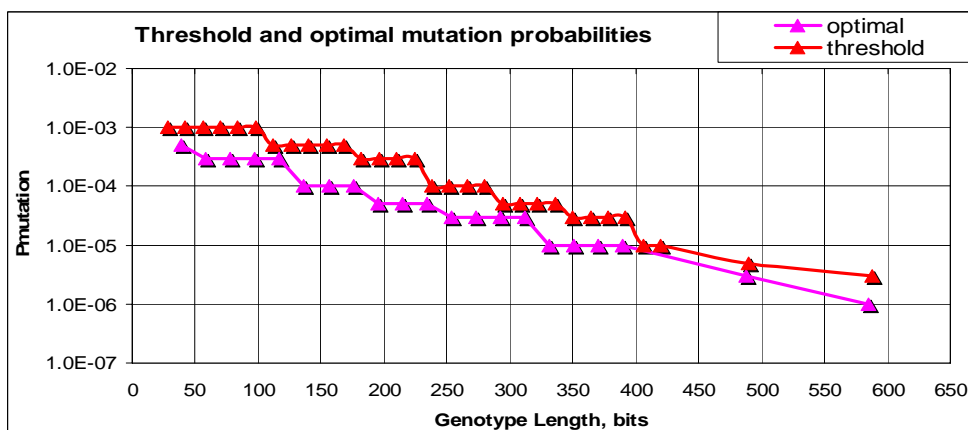


Рисунок 3 — Граничные и оптимальные вероятности мутации

### 3.4 Селекция (отбор)

Оператор селекции формирует следующее поколение из новых индивидуумов, образованных с помощью оператора мутации и кроссовера  $[\lambda, \mu]$ , или одновременно из новых индивидуумов и их родителей  $[\lambda + \mu]$ . Базовыми методами отбора являются селекция, основанная на принципе колеса рулетки и её модификация — универсальный стохастический отбор [4]. Согласно результатам экспериментов, универсальный стохастический отбор, где случайным образом выбирается только смещения относительно начала колеса рулетки, а остальные значения распределяются равномерно, обеспечивает большую вероятность успешного поиска глобального оптимума. Вероятность отбора в этих методах рассчитывается пропорционально значениям их функции приспособленности или исходя из ранга индивидуума, игнорируя абсолютное значение функции приспособленности (ранговая селекция). Ранговая селекция позволяет избежать масштабирования, необходимого для пропорционального отбора, при этом мощность отбора определяется напрямую, в отличие от выбора коэффициентов масштабирования. Альтернативным методом отбора является турнирная селекция [7], при которой случайным образом выбираются участники турнира, и лучший из них переходит в следующее поколение. Этот процесс повторяется, пока не сформируется новое поколение.

Подводя итоги тестирования различных модификаций оператора отбора, можно сделать вывод, что лучшие показатели производительности алгоритма наблюдаются при использовании турнирной или ранговой селекции при большом размере популяции. Турнирная селекция (количество участников турнира 2) показала наилучшие результаты при оптимизации функций низкой сложности (сфера, степ функция), а при тестировании сложных мультимодальных функций (функции Швевеля и Экли) — ранговая селекция со значением мощности отбора равном 1.7. Турнирная и ранговая селекция с высоким значением мощности отбора обеспечивают высокую интенсивность сходимости. Сравнивая значения времени поглощения [7] этих двух видов селекции можно сказать, что они довольно близки, а при увеличении значения мощности отбора ранговой селекции до 2.0 их значения были бы идентичны [7]. Поэтому, использование ранговой селекции для оптимизации простых функций не приводит к значительному ухудшению производительности системы и может быть выбрано в качестве метода селекции по умолчанию.

#### ***4 Механизм кэширования***

В систему заложен механизм сохранения значения целевой функции для всех индивидуумов, скопированных без изменения из предыдущих поколений. За счет этого достигается уменьшение числа ее вычислений в среднем на 61%. Также в системе имеется встроенный механизм кэширования вычисления целевой функции. Данный механизм позволяет избежать ее пересчета для повторно образовавшихся индивидуумов в процессе оптимизации. Особенно сокращается процесс вычисления на завершающем этапе работы алгоритма, когда значения параметров членов популяции колеблется вблизи точки глобального оптимума. По результатам тестирования установлено, что применение кэширования при использовании ГА позволяет уменьшить в среднем на 41% число вычислений целевой функции. В целом, за счет сугубо технических решений, количество вычислений сокращается в среднем на 76%.

#### ***5 Использование локальных детерминистических методов***

Эти методы используются в разработанной подсистеме для улучшения точности решения, найденного ГА. Локальные детерминистические методы (ЛДМ) начинают свою работу после того, как ГА приблизительно найдет глобальный оптимум, и просто сводят решение к нему (т.е. производят локальную оптимизацию, начиная от приблизительного расположения глобального оптимума). Таким образом, они не только увеличивают точность решения, но и уменьшают количество вычислений целевой функции, так как ГА не доходит до конца. Критерием останова ГА является формирование заданного процента схемы решения. Процент схемы определяется процентом зафиксированных позиций схемы текущей популяции (отношение порядка схемы [4] к длине генотипа). Было проведено исследование (около 100000 экспериментов) для различных сложных функций (Швефеля и Экли), ориентированное на определение процента схемы, при котором можно останавливать ГА (обнаружено расположение глобального оптимума). Результаты приведены на рис. 4 и показывают минимальный процент схемы, при котором уже находится глобальный оптимум, и максимальный процент схемы, при котором ещё есть выигрыш по вычислениям.

Из этих графиков следует, что для довольно гладких функций (таких как функция Швефеля) достаточно сформировать 10–20% схемы для эффективного поиска глобального оптимума. Это достигается уже при маленькой длине генотипа (маленьком количестве параметров). Тогда как для функций с сильными биениями

(функция Экли) эффект совместного использования ГА и ЛДМ достигается только при большой длине генотипа (большом количестве параметров), причём необходимо сформировать не менее 50% схемы.

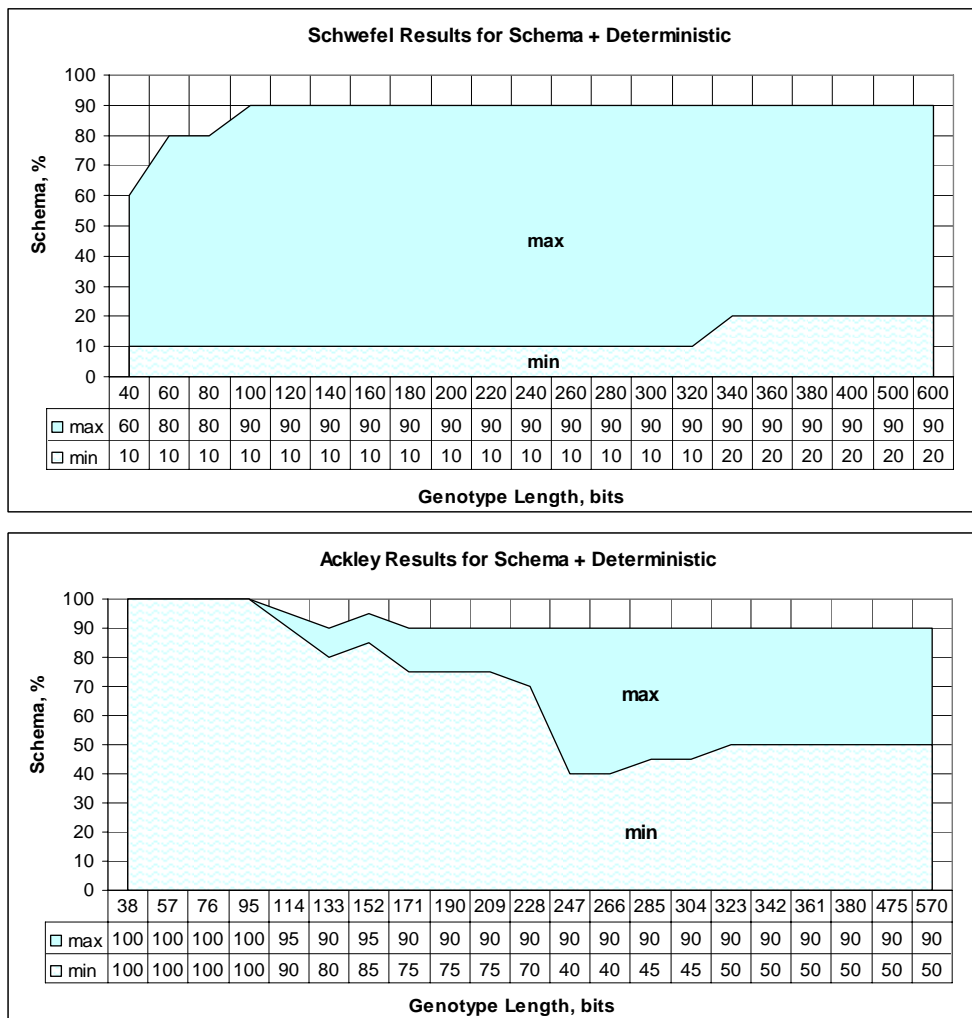


Рисунок 4 — Эффективные значения процентов сформированной схемы для начала работы ЛДМ

Проведена оценка эффективности использования комбинации ГА + ЛДМ относительно работы обычного ГА. Хотя при маленькой длине генотипа эта эффективность не всегда велика (особенно для функций с сильными биениями), зато при большой длине генотипа снижение количества вычислений целевой функции составляет для функции Швевеля 75%, а для функции Экли — 45%.

### **6 Выводы и планы на будущее**

Разработана подсистема оптимизации на базе генетического алгоритма, которая может быть сконфигурирована на реализацию

различных модификаций ГА. Было проведено интенсивное тестирование производительности алгоритма, аналитическая и экспериментальная оценка эффективности его модификаций на различных искусственных целевых функциях. Была разработана модификация ГА, в которой он комбинируется с ЛДМ, для увеличения точности найденного решения. Критерием останова ГА является формирование заданного процента схемы решения. В большинстве случаев, это даёт не только увеличение точности решения, а также уменьшает количество вычислений целевой функции. Разработанный алгоритм может быть эффективно использован для решения задач глобальной оптимизации и может быть настроен как автоматически, так и вручную для решения специфических задач (для этого используются выведенные зависимости параметров ГА от длины генотипа). Также была внедрена оптимизация с ограничениями. Для параллельного решения задачи оптимизации мы планируем портировать параллельный ГА с помощью MPI на суперкомпьютер с MIMD архитектурой. Планируется дальнейший сбор информации о характере работы подсистемы с целью поиска и внедрения новых идей.

### *Литература*

1. Назаров А.В., Лоскутов А.И. Нейросетевые алгоритмы прогнозирования и оптимизации систем. СПб.: Наука и Техника, 2003. — 384 с.
2. E. Polak. Optimization. Algorithms and Consistent Approximation. Springer-Verlag, New York, Inc., 1997.
3. J.H. Holland. Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor, MI, 1975.
4. Th. Bäck. Evolutionary Algorithms in Theory and Practice. Oxford University Press, Inc., New York, 1996.
5. L.J. Eshelman, R.A. Caruna and J.D. Schaffer. Biases in the Crossover landscapes. In J.D Schaffer, editor, Proceedings of the Third International Conference on Genetic Algorithms, pages 10–19. Morgan Kaufmann, 1989.
6. D.E. Goldberg. Genetic Algorithms in Search, Optimizations and Machine Learning. Addison Wesley, Reading, MA, 1989.
7. D.E. Goldberg and K. Deb. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. In G.J.E. Rawlins, editor, Foundation of Genetic Algorithms, pages 69–93. Morgan Kaufmann Publishers, San Mateo, CA, 1991.
8. A.D. Bethke. Genetic Algorithms as Function Optimizers. PhD thesis, University of Michigan, 1981. Diss. Abstr. Int. 41(9), 3503B, University Microfilms No. 8106101.



УДК 531.7

## МОДЕЛИРОВАНИЕ РАБОТЫ СХЕМЫ ИСТОЧНИКА СТАБИЛИЗИРОВАННОГО НАПРЯЖЕНИЯ ПИТАНИЯ МАГНИТОУПРУГИХ ДАТЧИКОВ НАГРУЗКИ

В.Ю. Ларин

Донецкий национальный технический университет

### *Анотація*

*В статті розглянуто поетапне моделювання ланок електронної схеми джерела стабілізованої напруги живлення магнітопружних датчиків навантаження, яке виконано за допомогою пакета електронного проектування Electronic Workbench 5.12*

Современные пакеты компьютерного проектирования электронных устройств (EDA) позволяют разработчикам электронной измерительной аппаратуры осуществить проверку расчетов разработанной схемы с большой достоверностью. Наличие интегрированных библиотек электронных компонентов позволяет использовать модели таких устройств при создании компьютерного варианта электронной схемы и затем в динамике выполнять её компьютерный анализ.

В данном случае пакет EDA — Electronic WorkBench 5.12 использовался для создания компьютерного варианта схемы источника стабилизированного напряжения питания магнитострикционных датчиков и проведения моделирования его работы с целью подтверждения адекватности расчетных значений разработанной схемы и её возможной корректировки. Функциональная схема моделируемого устройства представлена на рис.1.

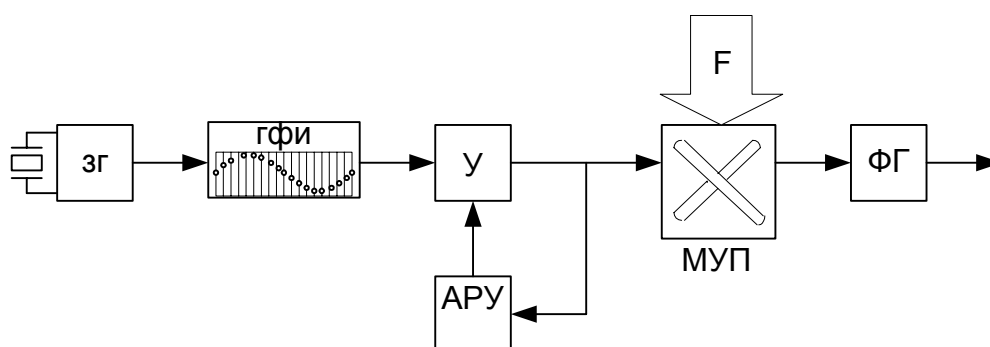


Рисунок 1 — Функциональная схема источника стабилизированного напряжения питания датчиков нагрузки

Согласно функциональной схемой в это устройство входят следующие блоки: задающий генератор (ЗГ); графический формирова-

тель импульсов (ГФИ), усилитель (У), охваченный цепью автоматической регулировки усиления (АРУ), магнитоупругий преобразователь (МУП), фоновый генератор (ФГ).

При создании электронной модели этой схемы был принят блочный принцип, предусматривающий последовательное моделирование работы функциональных звеньев.

**Моделирование задающего генератора.** Задающий генератор собран на микросхеме КР1533ЛН1 аналог SN74ALS04В или на двух элементах НЕ. Сопротивления, стоящие в схеме, задают генерируемую частоту, а кварцевый резонатор и конденсатор используется в качестве времязадающей цепи (рис.2).

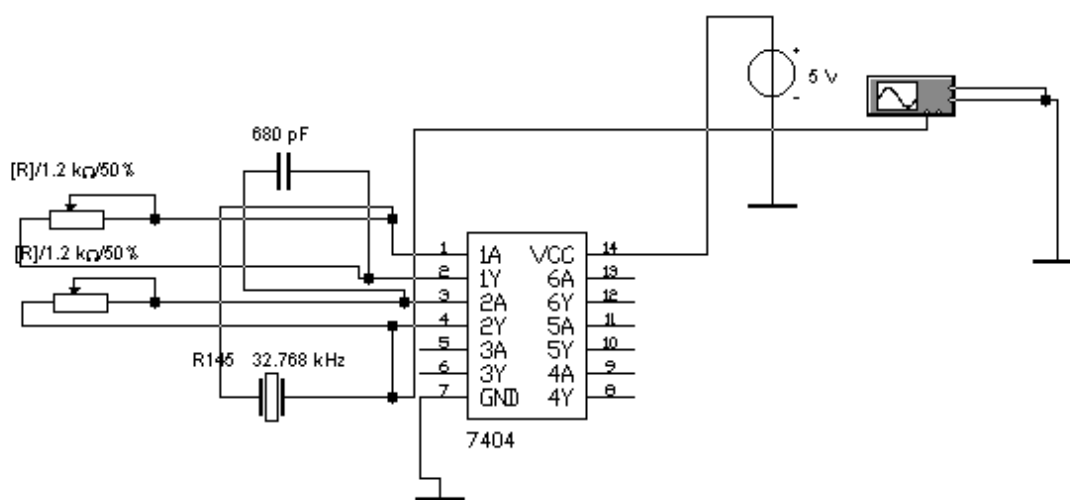


Рисунок 2 — Схема задающего генератора

На выходе генератора получается двухполярный сигнал прямоугольной формы.

**Моделирование графического формирователя импульсов.** Графический формирователь импульсов (рис.3) образован: счетчиком на микросхеме Кр1533ИЕ5, но в данной схеме применяем аналог SN74ALS93, дешифратором на микросхеме Кр1533ИД3, в данной схеме применяем аналог SN74ALS154 и диодно-резистивным мостом. В начале смоделируем работу счетчика и дешифратора, т.к. пакет прикладных программ Electronics Workbench 5.12 не поддерживает моделирование громоздких схем. Питание микросхем осуществляется источником постоянного напряжения на 5 вольт. Исходя из этого, для моделирования из пакета прикладных программ Electronics Workbench 5.12 выбираем напряжение питания микросхем от этого источника, т.е. на 5 вольт.

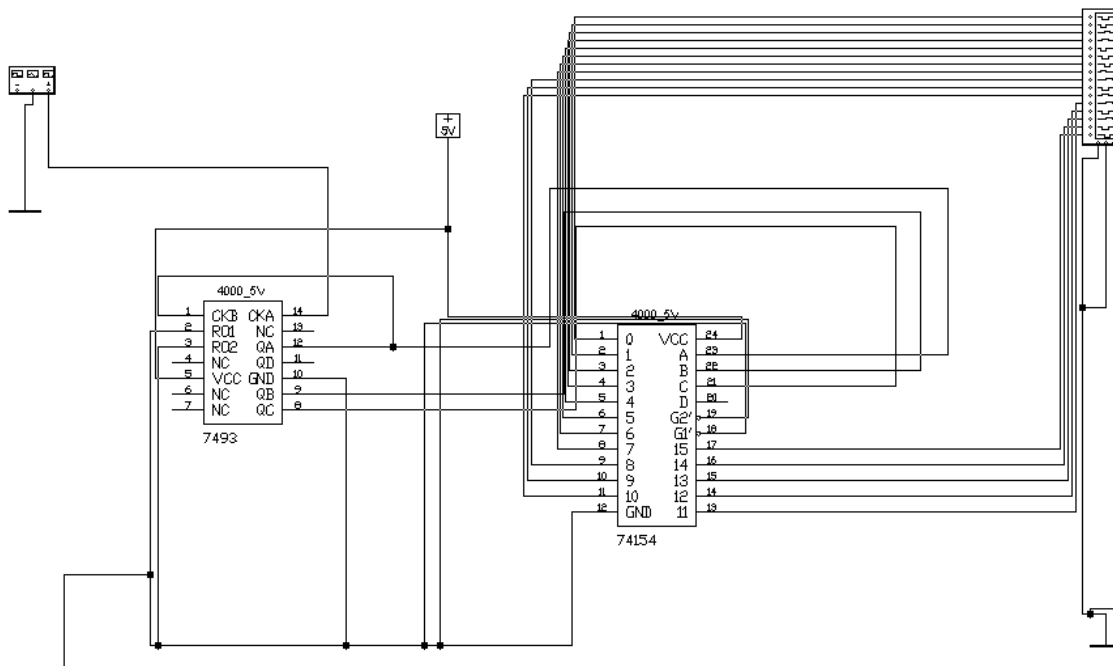


Рисунок 3 — Схема графического формирователя импульсов

Задающий генератор и источник питания для микросхем берем стандартные из набора библиотеки приборов пакета Electronics Workbench. При моделировании на выходе счетчика DD2 получаем счетные импульсы, кратные генерируемой частоте. При моделировании работы дешифратора DD3 применяем логический анализатор. С помощью логического анализатора получаем визуальное прослеживание логической единицы на выходах дешифратора с “0” по “15” (Рис.4).

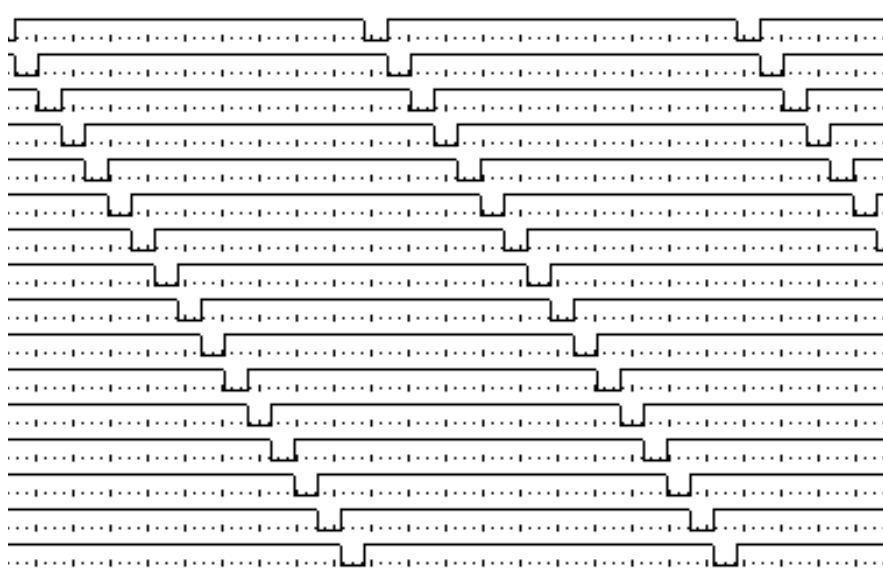


Рисунок 4 — Сигнал на выходе дешифратора

При моделировании диодно-резистивного моста с помощью логического генератора задаемся определенным алгоритмом, точно таким же, как и на выходе дешифратора. Получив требуемый сигнал подбираем сопротивления так, чтобы на выходе диодно-резистивного моста можно было получить ступенчатый сигнал. Диоды используются для защиты дешифратора от пробоя обратным током. На выходе диодно-резистивного делителя получаем ступенчатый сигнал:

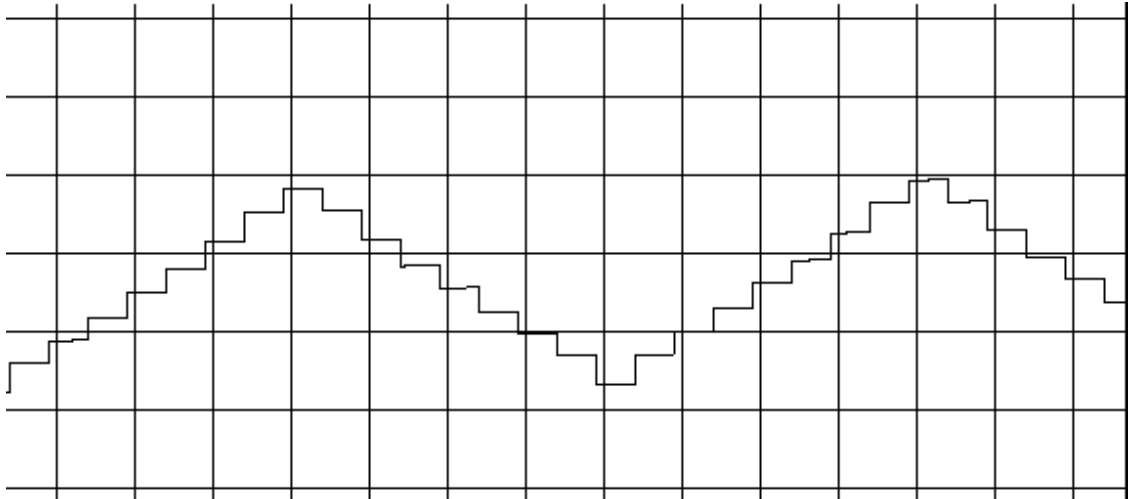


Рисунок 5 — Сигнал на выходе диодно-резистивного делителя

Для моделирования двухкаскадного усилителя требуется разбить его на две части:

**1. Моделирование первого каскада усилителя, собранного на операционном усилителе К140УД608 аналог MC1456C:**

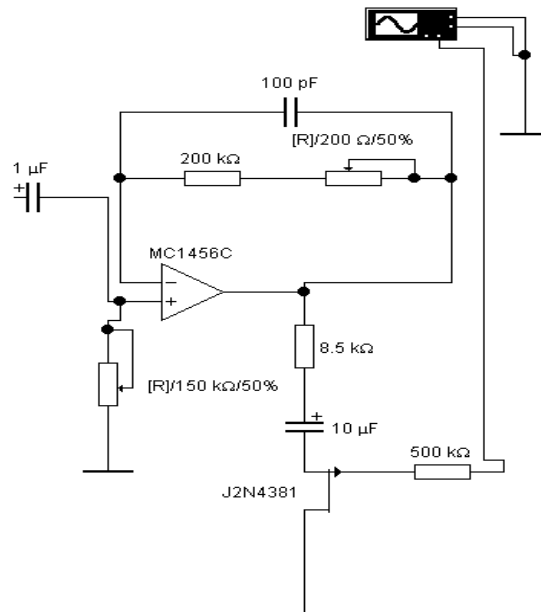


Рисунок 6 — Схема первого каскада усилителя

На выходе операционного усилителя охваченного местной ООС получаем усиленный ступенчатый сигнал:

**2. Моделирование оконечного каскада усилителя и фонового генератора.** Усиленный сигнал с операционного усилителя поступает на фильтр, образованный подстроечным сопротивлением и емкостью. С помощью этого фильтра получаем четкую по форме синусоиду. После этого синусоидальный сигнал попадает на двухтактный усилитель мощности, где сигнал усиливается и подается на магнитоупругий датчик. В качестве магнитоупругого преобразователя взят обычный трансформатор с сердечником с заданным коэффициентом трансформации.

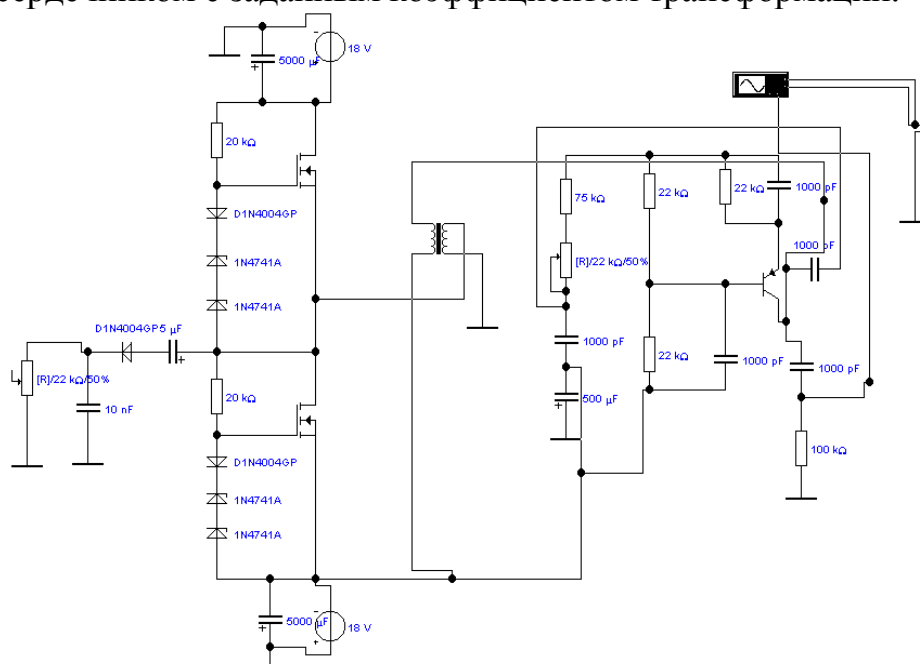


Рисунок 7 — Схема оконечного каскада усилителя и фонового генератора

На магнитоупругий первичный преобразователь поступает уже синусоидальный сигнал. С выхода магнитоупругого преобразователя сигнал поступает на фоновый генератор, после которого подключается схема ввода информации.

### **Литература.**

1. Карлашук Electronics Workbench 5.12. — М.: "Солон", — 2002. — 480 с.
2. Ларин В.Ю. Первичные преобразователи для информационно-измерительных систем. Наукові праці Донецького національного технічного університету. Серія: Обчислювальна техніка та автоматизація. Випск 64 / Редкол.: Башков Є.О. (голова) та ін. — Донецьк: Вид-во ДонНТУ, 2003. — С. 274–279.