

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ
УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

ISSN 0135-1710

АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ И ПРИБОРЫ АВТОМАТИКИ

**Всеукраинский межведомственный
научно-технический сборник**

Основан в 1965 г.

Выпуск 167

Харьков
2014

СОДЕРЖАНИЕ

ХАХАНОВ В.И., ХАХАНОВА И.В., TAMER BANI AMER, FARID DAHIRI (ФАРИД ДАХИРИ) ИМПЛЕМЕНТАЦИЯ КУБИТНЫХ СТРУКТУР ДАННЫХ В ПАРАЛЛЕЛЬНЫЙ ВЫЧИСЛИТЕЛЬ.....	4
СЕРДЮК Н.Н. АРХИТЕКТУРА ИНФОРМАЦИОННО-АНАЛИТИЧЕСКОЙ СИСТЕМЫ УПРАВЛЕНИЯ БЕЗОПАСНОСТЬЮ ПРОИЗВОДСТВА.....	17
БАРАННИК В.В., ОТМАН ШАДИ О.Ю., ХАХАНОВА А.В. МЕТОД СНИЖЕНИЯ ИНТЕНСИВНОСТИ ВИДЕОПОТОКА В ИНФОКОММУНИКАЦИОННЫХ СИСТЕМАХ.....	23
КОНОХ И.С. ПРЕДСТАВЛЕНИЕ ОБРАЗОВ ДИНАМИЧЕСКИХ ПРОЦЕССОВ В СИСТЕМАХ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ С ПОМОЩЬЮ САМОНАСТРАИВАЮЩИХСЯ АГЕНТОВ.....	28
БАРАННИК В.В., ДВУХГЛАВОВ Д.Э., ТВЕРДОХЛЕБ В.В. МЕТОД ДИНАМИЧЕСКОГО УПРАВЛЕНИЯ БИТОВОЙ СКОРОСТЬЮ ВИДЕОПОТОКА С ИСПОЛЬЗОВАНИЕМ ТРЕХМЕРНОГО ПРЕДСТАВЛЕНИЯ ТРАНСФОРМАНТ.....	37
НОВОЖИЛОВА М.В., ШТАНЬ И.В. РЕШЕНИЕ ДЕТЕРМИНИРОВАННОЙ ЗАДАЧИ ОПТИМИЗАЦИИ ТРЕХУРОВНЕВОЙ СЕТИ ПОСТАВОК ОДНОГО ТОВАРА.....	44
ЮДИН О.К., КУРИНЬ К.О., ЗЮБИНА Р.В. МЕТОДИКА ОЦІНКИ ЕФЕКТИВНОСТІ ТЕХНОЛОГІЇ СТИСНЕННЯ ЗОБРАЖЕНЬ НА БАЗІ МЕТОДУ ІНВАНІАНТНО-ПРОСТОРОВОГО КОДУВАННЯ ЗА ШВИДКОДІЄЮ.....	51
РЯБУХА Ю.Н. МЕТОД ОБРАБОТКИ ВИДЕОРЕСУРСОВ С СОХРАНЕНИЕМ ЦЕЛОСТНОСТИ В ИНФОРМАЦИОННЫХ СИСТЕМАХ.....	59
РЕФЕРАТИ.....	65
ПРАВИЛА ОФОРМЛЕНИЯ РУКОПИСЕЙ ДЛЯ АВТОРОВ НАУЧНО-ТЕХНИЧЕСКОГО СБОРНИКА.....	69

ИМПЛЕМЕНТАЦИЯ КУБИТНЫХ СТРУКТУР ДАННЫХ В ПАРАЛЛЕЛЬНЫЙ ВЫЧИСЛИТЕЛЬ

Предлагается практическая реализация генератора HDL-кода «квантовых» процессоров, использующих диаграммы Хассе для параллельных векторно-логических (теоретико-множественных) вычислений булеанов, применяемых для ускорения решения задач, моделирования, верификации, диагностирования. Программно-аппаратная реализация процессора основывается на использовании языков программирования: C++, Verilog, Python 2.7 и платформ: Microsoft Windows, X Window (в Unix и Linux) и Macintosh OS X. Генератор HDL-кода дает возможность автоматически синтезировать HDL-коды процессорной структуры от 1 до 16 двоичных разрядов для параллельной обработки соответствующего количества входных векторов или слов. Верификация HDL-кода процессора выполняется на тестовых примерах задачи покрытия, использующих две стратегии оптимизации: реверсивный алгоритм для устранения избыточности и разбиение матрицы покрытий на части в целях их последующей параллельной обработки процессорами Хассе.

1. Описание Verilog-модели устройства

Интерфейс. Модель процессора реализована с помощью одного из основных языков разработки ASIC-проектов – языка описания аппаратуры Verilog.

Интерфейс устройства в двух версиях представлен на рис. 1. Кроме входов синхронизации (clk) и сброса (rst), он содержит входные векторы данных i_i ; m – количество входов данных, n – разрядность векторов. Выход out – результат объединения всех векторов входных данных. Его разрядность равняется разрядности входных векторов и равна n . Выходы o_i содержат значения признаков соответствующего уровня, бит вектора равен 1, если вектор в соответствующей точке содержит все единицы, и 0 – в противном случае; i – обозначает уровень обработки векторов. Например, 8-входовой процессор с 8 разрядными входами данных будет иметь следующие выходы:

- out – 8-битовый выход;
- o7 – 8-битовый выход признака 7-го уровня;
- o2, o6 – выходы признаков 2-го и 6-го уровней;
- o3, o5 – выходы признаков 3-го и 5-го уровней;
- o4 – выходы признаков 4-го уровня;
- o8 – выход признака 8-го уровня.

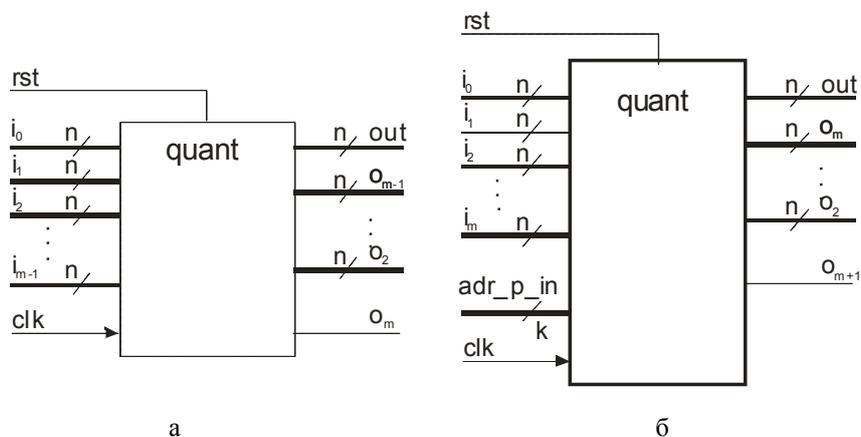


Рис. 1. Интерфейс квантового процессора

Разрядность выходов o_i зависит от числа входов i_m и вычисляется по формуле биномиальных коэффициентов

$$\binom{m}{k} = C_m^k = \frac{m!}{k!(m-k)!}, \quad (1)$$

где m – количество входов данных, а k – уровень их обработки в процессоре. Для $m=8-16$ параметры выходов приведены в табл. 1.

Таблица 1
Параметры выходов процессора

Уровень	Выход	Разрядность				
		m=8	m=10	m=12	m=14	m=16
2	o2	28	45	66	91	120
3	o3	56	120	220	364	560
4	o4	70	210	495	1001	1820
5	o5	56	252	792	2002	4368
6	o6	28	210	924	3003	8008
7	o7	8	120	792	3432	11440
8	o8	1	45	495	3003	12870
9	o9		10	220	2002	11440
10	o10		1	66	1001	8008
11	o11			12	364	4368
12	o12			1	91	1820
13	o13				14	560
14	o14				1	120
15	o15					16

Как видно из табл. 1, количество выходов значительно увеличивается с ростом числа входных векторов, поэтому в данном случае реализуется интерфейс (см. рис. 1, б) с мультиплексированием выходов o_i . В нем вход adr_p_in используется как адрес для выбора необходимого выхода признаков или выводится только часть регистров признаков.

Интерфейс устройства на языке Verilog для 16-входового процессора представлен в листинге 1. По умолчанию для входов и выходов используется тип данных `wire`.

Листинг 1
Интерфейс устройства на языке Verilog

```
// Вариант 1
module device
  (input clk, rst,
   input [7:0] i0, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, i11, i12, i13, i14, i15,
   output [7:0] out, o15,
   output o16);
// Вариант 2
module device
  (input clk, rst,
   input [7:0] i0, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, i11, i12, i13, i14, i15,
   input [24:0] adr_p,
   output [7:0] out, p_o16,
   output o2, o3, o4, o5, o6, o7, o8, o9, o10, o11, o12, o13, o14, o15, o16,
   output [7:0] o_pr);
```

При увеличении числа входных векторов размер кода существенно увеличивается, что требует больше времени на моделирование и синтез схемы. Производительность систем, работающих с моделями цифровых устройств, существенно повышается, если модель имеет иерархическую структуру, поэтому выгоднее выделить элементарную ячейку процессора в виде отдельного Verilog-модуля (листинги 2, 3) и на основе его строить схему

процессора (рис.2). Выход регистра – вектора пересечений представлен сигналом O, сигнал p_in соответствует регистру признака для элементарной ячейки, вход en_p позволяет отключать или подключать выход признака p. Параметр BITS определяет разрядность входных векторов.

Листинг 2

Элементарная ячейка

```

module element
  #(parameter BITS = 8)
  (input rst, clk,
   input [BITS-1:0] I0, I1,
   input en_p,
   output reg [BITS-1:0] O,
   output p);
  reg p_in;
  always @(posedge clk, posedge rst)
    if (rst) O <= 'b0;
    else O <= I0 | I1;
  always @(posedge clk, posedge rst)
    if (rst) p_in <= 1'b0;
    else p_in <= &O;
  assign p = (en_p)? p_in: 1'bZ;
endmodule

```

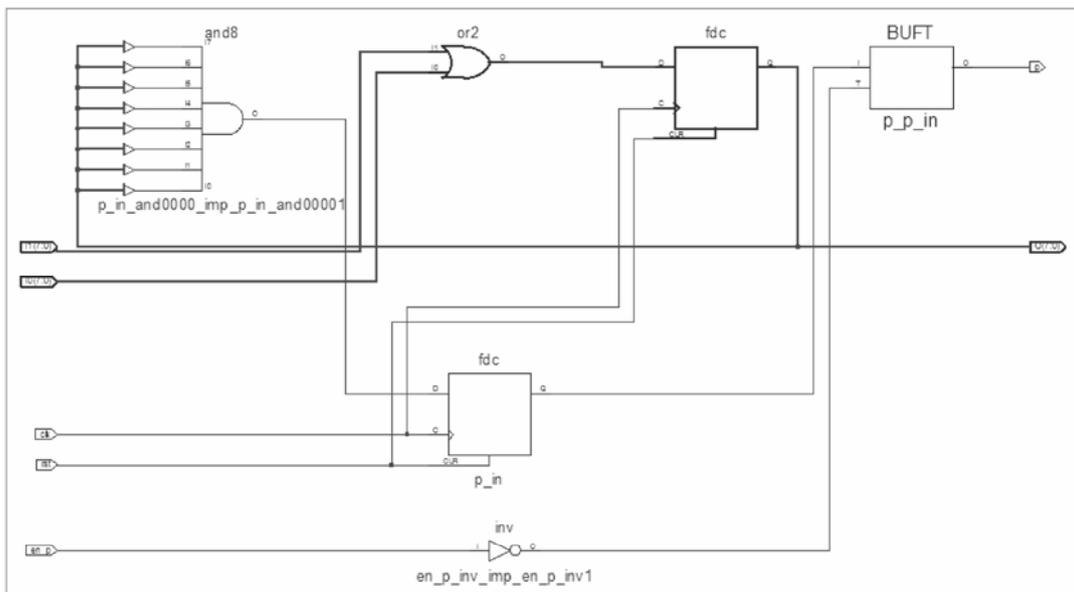


Рис. 2. Синтез схемы элементарной ячейки в Xilinx ISE Design Suite 11.1

Листинг 3

Фрагмент отчета синтеза элементарной ячейки процессора

* HDL Synthesis *

Performing bidirectional port resolution...
 Synthesizing Unit <element>.
 Related source file is "component.v".
 Found 8-bit register for signal <O>.
 Found 1-bit tristate buffer for signal <p>.
 Found 1-bit register for signal <p_in>.

Summary:
inferred 9 D-type flip-flop(s).
inferred 1 Tristate(s).
Unit <element> synthesized.

HDL Synthesis Report
Macro Statistics
Registers : 2
1-bit register : 1
8-bit register : 1
Tristates : 1
1-bit tristate buffer : 1

При использовании элементарной ячейки процессора ее подключение выполняется так, как показано в листинге 4. Каждая ячейка соединяется с декодером адреса `adr_p_in`, определяющим, будет ли данный признак в текущий момент подан на выход `o_pr` или нет. Декодер основан на тристабильном буфере. В данном фрагменте ячейки принадлежат второму уровню процессора и обрабатывают информацию с входов `i0` и `i1-16`. Ячейке `comp_0_1` соответствует адрес 'd16, это означает, что если на вход `adr_p_in` подать значение 16, то значение признака поступит на внешний выход.

Листинг 4
Фрагмент структурной модели процессора

```
// COMPONENT LEVEL: 2
assign o_pr = (adr_p_in == 'd16) ? p_0_1 : 'bZ;
element #(8) comp_0_1 (rst, clk, r_0, r_1, r_0_1, p_0_1);

assign o_pr = (adr_p_in == 'd32) ? p_0_2 : 'bZ;
element #(8) comp_0_2 (rst, clk, r_0, r_2, r_0_2, p_0_2);

assign o_pr = (adr_p_in == 'd48) ? p_0_3 : 'bZ;
element #(8) comp_0_3 (rst, clk, r_0, r_3, r_0_3, p_0_3);

assign o_pr = (adr_p_in == 'd64) ? p_0_4 : 'bZ;
element #(8) comp_0_4 (rst, clk, r_0, r_4, r_0_4, p_0_4);

assign o_pr = (adr_p_in == 'd80) ? p_0_5 : 'bZ;
element #(8) comp_0_5 (rst, clk, r_0, r_5, r_0_5, p_0_5);

assign o_pr = (adr_p_in == 'd96) ? p_0_6 : 'bZ;
element #(8) comp_0_6 (rst, clk, r_0, r_6, r_0_6, p_0_6);
```

Присваиваемый ячейке адрес строится на основе ее индекса. Под каждый индекс отводится необходимое для его представления количество разрядов. Например, для 16 входных переменных на каждый индекс записывается 4 бита, индексы кодируются справа налево. Так, для ячейки с индексом `0_1` адрес в двоичном виде будет выглядеть как `0001_0000`, что соответствует 16 в десятичной форме.

2. Генератор

Для генерации кода моделей устройств различной конфигурации разработан программный генератор. Создание моделей с большим числом входных векторов возможно только с помощью генератора.

Исходными данными являются: количество входных векторов и их разрядность, на их основе выполняется генерация Verilog-кода. Программа написана с использованием языка Python 2.7. Разработаны две версии генератора, которые реализуют поведенческую и структурные модели процессора.

2.1. Генератор поведенческой модели процессора

Структура поведенческой Verilog-модели представлена на рис. 3. Она включает декларацию имени модуля (Заголовок), декларативные части для объявления входных и выходных линий, переменных, соответствующих регистрам каждого уровня; код, описывающий регистры и значения признаков; завершающую часть, закрывающую модуль.

Заголовок
Декларация входов
Декларация выходов
Декларация регистров
Реализация регистров
Присвоение значений выходов
Завершение

Рис. 3. Организация Verilog-модели

Для генерирования кода разработан основной класс Generator, а также вспомогательный класс genIndex, генерирующий индексы внутренних регистров модели. Структура классов, описанная с помощью инструмента ArgoUML, представлена на рис. 4. Переменные класса genIndex: levels – количество уровней модели (соответствует числу входов), level – текущий уровень для генерирования индексов регистров, ind_list – список строковых значений индексов. Два метода класса: genLevel() и getIndList() генерируют и возвращают список индексов. Объекты класса genIndex используются в методах создания описания и реализации регистров класса Generator.

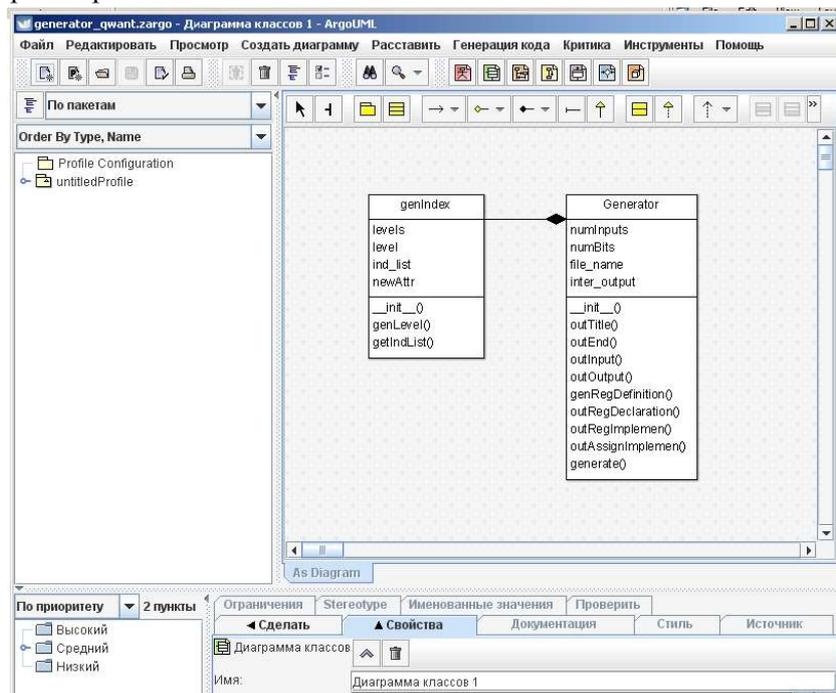


Рис. 4. Структура классов программы (ArgoUML)

Для реализации каждой части модели создан отдельный метод класса Generator, которые описаны в табл. 2. Класс Generator содержит поля: numInputs – число входов модели, numBits – разрядность входных/выходных векторов, file_name – имя файла модели, inter_output – позволяет при необходимости отключать генерирование выходов признаков для регистров внутренних уровней. Метод инициализации переменных задает значения переменных по умолчанию (листинг 5): numInputs = 4, numBits = 8, file_name = “model.v”, inter_output = True. Любой из параметров может быть изменен при создании экземпляра класса Generator, например следующим образом:

```

numInputs = 16
numBits = 8
obj1 = Generator(numInputs, numBits, “model”+str(numInputs)+”.v”, False).

```

Таблица 2
Методы класса Generator поведенческой модели

Метод	Описание
outTitle()	Генерация заголовка
outInput()	Генерация деклараций входов
outOutput()	Генерация деклараций выходов
outRegDeclaration()	Генерация деклараций переменных для описания регистров
outRegImplemen()	Генерация операторов, реализующих регистры
outAssignImplemen()	Генерация операторов assign для формирования при знаков уровней и присвоения значений выходов
outEnd()	Генерация завершающей части кода модели. Завершение
genRegDefinition(name1, name2, vec = True)	Вызывается из outRegDeclaration()
generate()	Является основным методом класса, который, используя остальные методы класса, выполняет генерацию целой модели устройства и запись ее в файл.

Листинг 5

Фрагмент класса Generator с функцией инициализации полей class Generator

```

""" Model code generator """
def __init__(self, numInputs = 4, numBits = 8, file_name = "model.v",
             inter_output = True):
    self.numInputs = numInputs
    self.numBits = numBits
    self.file_name = file_name
    self.inter_output = inter_output
    if self.numInputs <= 4 :
        self.coef = 2
    elif self.numInputs <= 8 :
        self.coef = 3
    elif self.numInputs <= 16 :
        self.coef = 4

```

Класс Generator содержит также функции outTitle() и outEnd(), которые открывают и закрывают внешний файл (листинг 6).

Листинг 6

Функции outTitle() и outEnd()

```

def outTitle(self):
    self.f = open(self.file_name, 'w')
    self.f.write("module device\n (input clk, rst,\n")
def outEnd(self):
    self.f.write("\nendmodule")
self.f.close()

```

Рис. 5 представляет полную структуру классов программы генератора поведенческой модели процессора, включающую реализацию двух классов Generator и genIndex, а также вспомогательные функции для подготовки и формирования данных, работы со списками индексов регистров каждого уровня и признаков полноты покрытия.

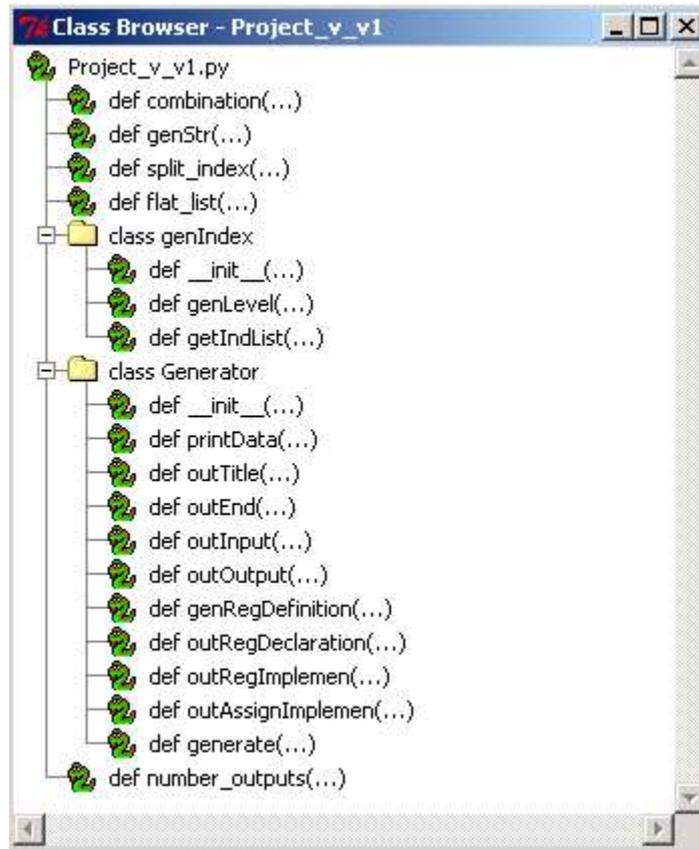


Рис. 5. Браузер классов из программы Python 2.7 Shell

Генерация индексов регистров и переменных является достаточно сложной задачей и для этой цели создан отдельный класс genIndex (листинг 7). Для генерации индексов используются списки, принадлежащие языку Python. Класс genIndex содержит поля: levels – число уровней модели, которое соответствует количеству векторов входных данных; level – уровень, для которого генерируются индексы, ind_list – список индексов. Метод genLevel() генерирует список индексов, а метод getIndList() – возвращает список индексов.

Листинг 7

Класс генератора индексов переменных

```
class genIndex:
    """ generate list of index for the level """
    def __init__(self, levels = 16, level = 1):
        self.levels = levels
        self.level = level
        self.ind_list = []
        self.genLevel()
    def genLevel(self):
        for i in xrange(self.levels):
            self.ind_list.append([str(i)])
        if self.level > 1:
            list_t = []
            ii=0
```

```

for i1 in range(self.levels-1):
    ii += 1
    t=[]
    for i2 in xrange(ii, self.levels):
        t.append(str(i1)+»_»+str(i2))
    #print t
    list_t.append(t)
#print «list_level 1»,list_t
self.ind_list = list_t
if self.level > 2:
    iLevel = 3
    while iLevel <= self.level:
        list_t2 = []
        list_t.pop(0)
        i=0
        while (len(list_t) > 0):
            t = []
            for i1 in xrange(len(list_t)):
                list_i1 = list_t[i1]
                for i2 in xrange(len(list_i1)):
                    s = str(i) + «_» + list_i1[i2]
                    t.append(s)
                i += 1
            list_t.pop(0)
            list_t2.append(t)
        #print «list_level «,iLevel, «:», list_t2
        list_t = list_t2
        iLevel += 1
    self.ind_list = list_t2
def getIndList(self):
    return self.ind_list

```

В табл. 3 показано, как увеличивается число внешних линий устройства с увеличением числа входов (numInputs). При этом видно, что разрядность входных сигналов (numBits) несущественно влияет на общее число входов и выходов модели.

Таблица 3

Зависимость количества внешних линий модели от числа входов (numInputs) и их разрядности (numBits)

numInputs	numBits							
	4	8	12	16	20	24	28	32
4	31	55	79	103	127	151	175	199
6	83	115	147	179	211	243	275	307
8	279	319	359	399	439	479	519	559
10	1 051	1 099	1 147	1 195	1 243	1 291	1 339	1 387
12	4 127	4 183	4 239	4 295	4 351	4 407	4 463	4 519
14	16 419	16 483	16 547	16 611	16 675	16 739	16 803	16 867
16	65 575	65 647	65 719	65 791	65 863	65 935	66 007	66 079
18	262 187	262 267	262 347	262 427	262 507	262 587	262 667	262 747

С увеличением числа входных векторов значительно увеличивается размер модели, а следовательно, и количество кода для ее реализации, в результате обработка таких моделей системами синтеза и имплементации занимает очень много времени. Подобные программы лучше работают с кодом, состоящим из нескольких модулей. В целях повышения скорости выполнения операций синтеза и имплементации разработана структурная модель процессора, в котором отдельный узел реализован в виде отдельного модуля – элементар-

ной ячейки. Кроме того, в эту модель добавлен вход адреса, позволяющий выборочно считывать информацию с узлов-признаков, что дало возможность существенно сократить число внешних выходов устройства.

2.2. Генератор структурной модели процессора

Структурная Verilog-модель состоит из областей, которые представлены на рис. 6. Вместо области реализации регистров и присвоения значений выходов, как было в поведенческой модели, эта модель содержит описание реализации регистров 1-го уровня с помощью конструкции always и реализацию регистров остальных уровней процессора, включая оператор реализации копии модуля элементарной ячейки, оператор assign, описывающий тристабильный буфер для подключения выхода признака и условный оператора реализации мультиплексора выбора ячейки. Программа генератора усложняется за счет добавления нескольких методов (структура программы из отладочного средства для Python 2.7 представлена рис. 7). Описание методов класса генератора включены в табл. 4. Программа содержит 380 строк кода.

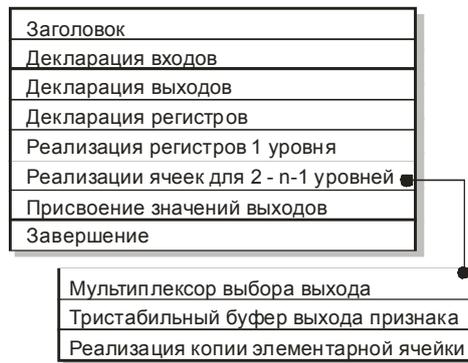


Рис. 6. Организация Verilog-модели

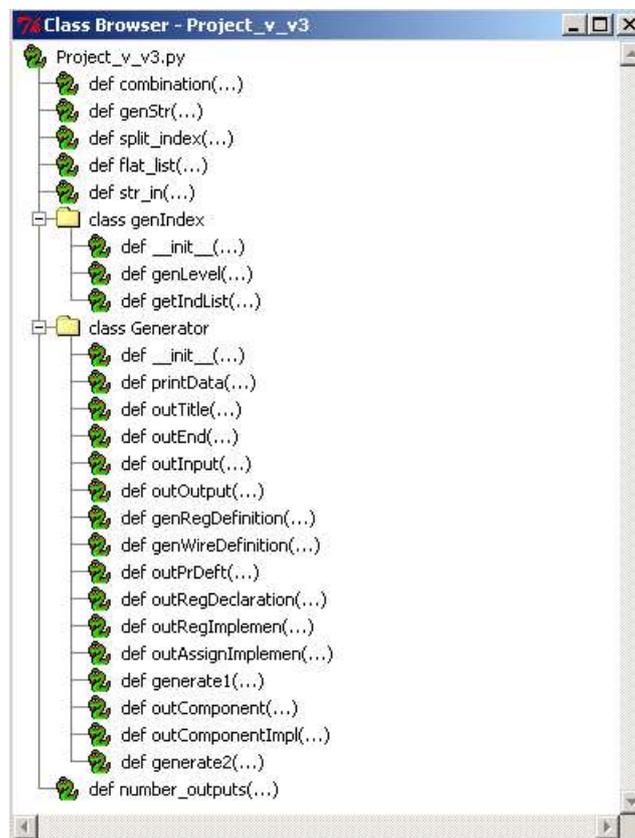


Рис. 7. Структура программы генератора в браузере классов из программы Python 2.7 Shell

Присваиваемый каждой ячейке адрес генерируется на основе строкового представления ее индекса с помощью функции `str_in` (листинг 8).

Листинг 8

Функция генерации адреса ячейки

```
def str_in(l0, step):
    """ Form integer address value based on element index """
    i = 0
    l = l0.split("_")
    l.reverse()
    for j in xrange(len(l)):
        i *= 2**step
        i += int(l[j])
        #print 'i', i
    return i
```

Таблица 4

Методы класса `Generator` поведенческой модели

Метод	Описание
<code>outTitle()</code>	Генерация заголовка
<code>outInput()</code>	Генерация деклараций входов
<code>outOutput()</code>	Генерация деклараций выходов
<code>outRegDeclaration()</code>	Генерация деклараций переменных для описания регистров
<code>outRegImplemen()</code>	Генерация операторов, реализующих регистры
<code>outAssignImplemen()</code>	Генерация операторов <code>assign</code> для формирования признаков уровней и присвоения значений выходов
<code>outEnd()</code>	Генерация завершающей части кода модели. Завершение
<code>genRegDefinition(name1, name2, vec = True)</code>	Декларация регистровых переменных, вызывается из функции <code>outRegDeclaration()</code>
<code>genWireDefinition()</code>	Генерация переменных типа данных <code>wire</code> класса цепи, моделирующих тристабильные буферы
<code>OutComponent()</code>	Генерирует файл с моделью элементарной ячейки процессора
<code>OutComponentImpl()</code>	Генерирует оператор реализации копии модуля элементарной ячейки процессора
<code>printData()</code>	Вывод числа входов процессора и их разрядности
<code>generate2()</code>	Является основным методом класса, который, используя остальные методы класса, выполняет генерацию целой модели устройства и запись ее в файл.

Несмотря на то, что `python` – это интерпретируемый язык программирования, для программ, созданных с помощью этого языка, в случае необходимости можно легко получить исполняемый файл, используя средства `PyInstaller` (<http://www.pyinstaller.org/>) или `py2exe` (<http://www.py2exe.org/>), или подобные им. Конвертор `PyInstaller` создает независимые исполняемые программы для операционных систем `Windows`, `Linux`, `Mac OS X`, `Solaris` и `AIX`, в то время как `py2exe` реализует только `Windows` приложения.

2.3. Графический интерфейс пользователя генератора

Для оформления графического интерфейса пользователя была применена открытая библиотека `tkinter` языка `python`, которая является де-факто стандартом для решения подобных задач. Доступность, переносимость, простота получения, документированность и наличие расширений определяют популярность `tkinter` для формирования GUI в Python-приложениях в течение многих лет: В отличие от более сложных систем, библиотека `tkinter` позволяет сразу же приступить к работе с ней, без необходимости предварительно осваивать крупные модели взаимодействия классов. Несмотря на простоту прикладного интерфейса библиотеки `tkinter`, она дает возможность добавлять новые виджеты, написанные на языке Python, или подключать дополнительные расширения, такие как `Pmw`, `Tix` и `ttk`. Важным преимуществом является переносимость. Сценарий на языке Python, в котором графический интерфейс строится с помощью библиотеки `tkinter`, будет работать без изменений на всех основных современных оконных платформах: Microsoft Windows, X Window (в Unix и Linux) и Macintosh OS X. Более того, внешний вид созданного интерфейса будет привычен для пользователей каждой из этих платформ. Эта особенность развивалась по мере того как библиотека Tk (на которой основана `tkinter`) становилась все более зрелой. Графический интерфейс, реализованный сценарием Python/`tkinter`, в Windows выглядит так, как должен выглядеть интерфейс программы для Windows; в Unix и Linux обеспечивает такое же взаимодействие и на Mac он выглядит так, как должна выглядеть программа Mac.

Библиотека `tkinter` является модулем стандартной библиотеки Python, поставляемой вместе с интерпретатором. Более того, в большинство пакетов установки Python (включая стандартный пакет установки Python для Windows, Mac и большинство дистрибутивов Linux) уже включена поддержка `tkinter`. Благодаря этому сценарию, написанные с использованием модуля `tkinter`, сразу могут работать с большинством интерпретаторов Python, не требуя дополнительных действий по установке. Поскольку задействованная в ней библиотека Tk используется также языками программирования Tcl и Perl (и многими другими), ей уделяется больше внимания и усилий разработчиков, чем другим имеющимся инструментариям.

Окно графического интерфейса генератора представлено на рис. 8. В нем перед генерацией Verilog-кода можно задавать количество входов Number of inputs (число входных векторов), разрядность входных векторов Number of bits, имя файла и путь для сохранения создаваемой модели File name.

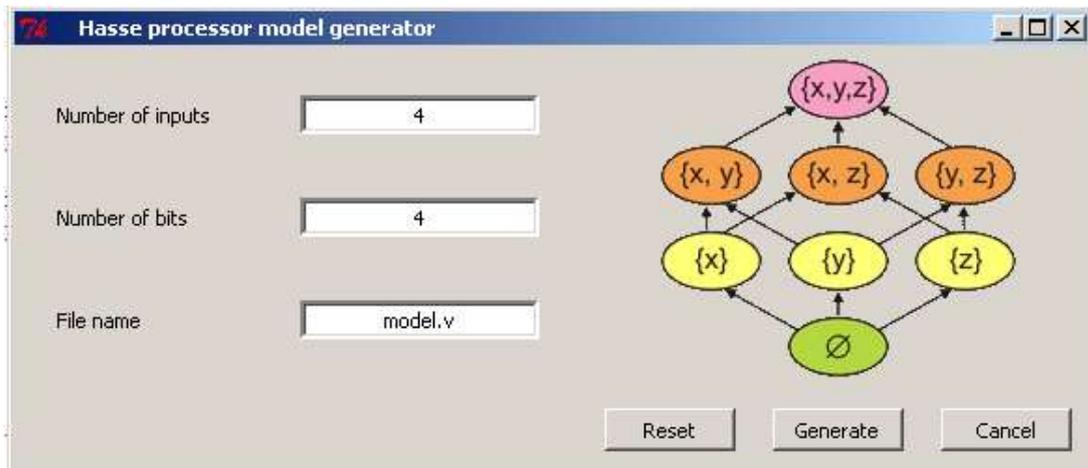


Рис. 8. Окно графического интерфейса для Windows

По умолчанию число входов и их разрядность равна 4, генерируемая модель сохраняется в файл `model.v`. Кнопка `Reset` сбрасывает введенные значения параметров к заданным по умолчанию, `Generate` – генерирует Verilog-модель процессора, `Cancel` – закрывает окно генератора. В программе использовались экземпляры классов метки `Label`, поля ввода `Entry`, кнопки `Button`, `PhotoImage` для вывода изображения. Все классы принадлежат библиотеке `Tkinter`, которая в свою очередь основана на библиотеке `Tk()`:

```
import sys
from Tkinter import *
```

```
root = Tk()
root.title(" Hasse processor model generator")
```

Для упорядочивания графического отображения объектов классов в окне генератора используется метод сетки – grid. Следующий фрагмент кода представляет создание метки и поля ввода для ввода значения параметра Number of inputs и размещение его в нулевой строке и колонке сетки:

```
l1 = Label(root, text = 'Number of inputs')
l1.grid(row = 0, column = 0, padx = 20, pady = 20, sticky=W)
num_levels = IntVar(value = 4)
i1 = Entry(root, justify = "center", width = 20, textvariable = num_levels)
i1.config(bd = 3, relief = SUNKEN)
i1.grid(row = 0, column = 1)
```

3. Синтез и имплементация

Реализация устройства на микросхемах FPGA фирмы Xilinx Spartan 3E (xc3s1600e-4-fg320) и Virtex 5 xc5vlx50. Для синтеза и имплементации использовался программный пакет фирмы Xilinx ISE Design Suite 11.1. Модель устройства, с конфигурацией 10 входов, разрядность входов 8 бит, реализована с использованием микросхемы фирмы Xilinx серии Spartan-3E: xc3s1600e. При этом полученная схема содержит 9 247 триггера FF и 18 062 4-х входовые таблицы преобразования LUT, что составляет 31 и 61% доступных ресурсов микросхемы:

Logic Utilization:

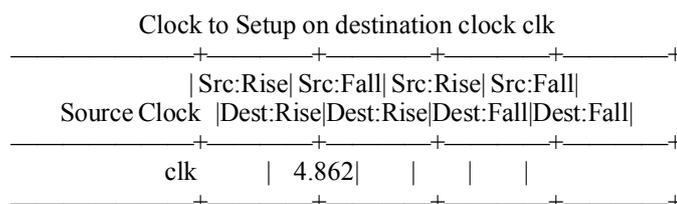
Number of Slice Flip Flops: 9,247 out of 29,504 31%

Number of 4 input LUTs: 18,062 out of 29,504 61%.

Исходя из результатов статического временного анализа (табл. 5), минимальный рабочий период синхросигнала равен 41 ns, что соответствует частоте 24 МГц.

Таблица 5
Временные параметры устройства

Clock clk to Pad		
Destination	clk (edge) to PAD	Clock Internal Clock(s) Phase
o2	13.343(R)clk_BUF	0.000
o3	15.381(R)clk_BUF	0.000
o4	16.611(R)clk_BUF	0.000
o5	19.541(R)clk_BUF	0.000
o6	20.813(R)clk_BUF	0.000
o7	15.613(R)clk_BUF	0.000
o8	12.385(R)clk_BUF	0.000
o9	13.186(R)clk_BUF	0.000
o_pr<0>	40.912(R)clk_BUF	0.000
o_pr<1>	37.572(R)clk_BUF	0.000
o_pr<2>	37.575(R)clk_BUF	0.000
o_pr<3>	37.859(R)clk_BUF	0.000
o_pr<4>	38.169(R)clk_BUF	0.000
o_pr<5>	37.284(R)clk_BUF	0.000
o_pr<6>	38.165(R)clk_BUF	0.000
o_pr<7>	37.858(R)clk_BUF	0.000
out<0>	7.854(R)clk_BUF	0.000
out<1>	8.086(R)clk_BUF	0.000
out<2>	8.088(R)clk_BUF	0.000
out<3>	8.057(R)clk_BUF	0.000
out<4>	7.534(R)clk_BUF	0.000
out<5>	7.465(R)clk_BUF	0.000
out<6>	8.073(R)clk_BUF	0.000
out<7>	8.032(R)clk_BUF	0.000
p_o10<0>	7.716(R)clk_BUF	0.000



Для вычисления временных параметров используются результаты статического анализа, выполняемые программным обеспечением на этапе Place&Route. Система вычисляет задержки путей между регистрами (Tclk_to_clk) и от синхровхода до выхода (Tclk_to_pad). Максимальный из этих параметров определяет минимальный рабочий период синхросигнала, который может быть подан на вход устройства.

$Period = \max\{Tclk_to_clk, Tclk_to_pad_max\};$

Например, для временных параметров:

Tclk_to_clk = 4.862 ns

Tclk_to_pad_max = 40.912 ns

Минимальный рабочий период синхросигнала будет равен $Period = 40.912$ ns, что соответствует частоте Fclk = 24 МГц.

Выводы

Представлена практическая реализация генератора HDL-кода специализированных процессоров, использующих диаграммы Хассе для параллельных векторно-логических вычислений булеанов, применяемых для ускорения моделирования, верификации и диагностирования. Программно-аппаратная реализация процессора основана на использовании языков программирования: C++, Verilog, Python 2.7 и платформ: Microsoft Windows, X Window (в Unix и Linux) и Macintosh OS X. Генератор дает возможность автоматически синтезировать HDL-коды процессорной структуры от 1 до 16 двоичных разрядов для параллельной обработки соответствующего количества входных векторов или слов. Теоретические и практические результаты дают возможность за счет аппаратной и структурной избыточности на 50% – быстродействие интерпретативного моделирования, на 5% повысить выход годной продукции, на 12% – глубину диагностирования неисправных функциональных блоков и на 15% уменьшить время отладки HDL-кода в процессе проектирования цифровых систем на кристаллах.

Список литературы: 1. *Michael A. Nielsen & Isaac L. Chuang.* Quantum Computation and Quantum Information. Cambridge University Press. 2010. 676p. 2. *Stig Stenholm, Kalle-Antti Suominen.* Quantum approach to informatics. John Wiley & Sons, Inc., 2005. 249p. 3. *Mark G. Whitney.* Practical Fault Tolerance for Quantum Circuits. PhD dissertation. University of California, Berkeley. 2009. 229p. 4. *Mikio Nfirhara.* Quantum Computing. An Overview. Higashi-Osaka. Kinki University, 2010. 53p. 5. *Горбатов В.А.* Основы дискретной математики. М.: Высш. шк. 1986. 311 с. 6. *Hahanov V.I., Litvinova E.I., Chumachenko S.V., Baghdadi Ammar Awni Abbas, Eshetie Abebech, Mandefro.* Qubit Model for solving the coverage problem // Proc. of IEEE East-West Design and Test Symposium. IEEE. USA. Kharkov. 14-17 September 2012. P.142 - 144.

Поступила в редколлегию 11.06.2014

Хаханов Владимир Иванович, декан факультета КИУ ХНУРЭ, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. IEEE Senior Member. IEEE Computer Society Golden Core Member. Научные интересы: проектирование и тестирование вычислительных систем, сетей и программных продуктов. Увлечения: баскетбол, футбол, теннис, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +380 57 70-21-326. E-mail: hahanov@kture.kharkov.ua.

Хаханова Ирина Витальевна, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: проектирование и тестирование цифровых систем и сетей на кристаллах. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +380 57 70-21-326. E-mail: hahanova@mail.ru.

Tamer Bani Amer, аспирант кафедры АПВТ ХНУРЭ. Научные интересы: проектирование и тестирование вычислительных систем. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +380 57 70-21-326. E-mail: hahanov@kture.kharkov.ua.

Dahiri Farid (Дахири Фарид), магистрант кафедры АПВТ ХНУРЭ. Научные интересы: компьютерная инженерия, методы оптимизации, программирование. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +380 57 70-21-326. E-mail: dahiri.farid@gmail.com.

АРХИТЕКТУРА ИНФОРМАЦИОННО-АНАЛИТИЧЕСКОЙ СИСТЕМЫ УПРАВЛЕНИЯ БЕЗОПАСНОСТЬЮ ПРОИЗВОДСТВА

Рассматривается задача предприятия, заключающаяся в создании безопасных условий труда в пределах существующих возможностей техники в целях снижения риска аварийных ситуаций, травм или гибели людей. Возникает необходимость создания полномасштабных информационных систем, которые ориентированы на мониторинг, анализ и управление безопасностью производства как состоянием процессов, обусловленных действием различных групп вредных факторов.

Введение

Безопасность производственного процесса – его свойство сохранять соответствие требованиям безопасности труда в условиях, установленных нормативно-технической документацией. Несмотря на развитие науки и техники, несчастные случаи на производстве и профессиональные заболевания были и остаются причиной многих человеческих трагедий и самых серьезных экономических потерь. Поэтому и сегодня задачи создания средств и систем производственной безопасности остаются актуальными. Сложность и увеличение масштабов технологических систем, рост автоматизации производства заостряют проблему надежности человеческого звена. Задачи снижения производственного травматизма, обеспечения безопасных условий труда требуют контроля факторов производственной среды, оказывающих вредное влияние на здоровье работников, которое имеет принципиальное значение в опасной производственной ситуации. *Актуальной проблемой* предприятия в этом случае является создание безопасных условий труда в пределах существующих возможностей техники в целях снижения риска аварийных ситуаций, травм или гибели людей.

1. Анализ существующих информационных систем и технологий управления безопасностью производства

Исследования показали, что рынок информационных систем (ИС) управления безопасностью производства ориентирован на разработку уникальных систем, отвечающих на конкретные потребности заказчика, или же систем документооборота отдела промышленной безопасности. Типовые ИС управления безопасностью производства практически полностью отсутствуют. Наиболее близкие к современному представлению ИС управления объектами или процессами позволяют автоматизировать основные производственные функции специалиста по безопасности производства: аттестация рабочих мест, формирование отчетности по несчастным случаям, профзаболеваниям, инструктажам и др., ведение документации в соответствии с текущим законодательством. Однако главным недостатком данных систем является практически полное отсутствие функций анализа ситуации, сложившейся в области безопасности производства, а также прогноза развития непредвиденных и чрезвычайных ситуаций и аварий, которые могут возникнуть.

Существующие ИС и технологии ориентированы, прежде всего, на формирование и ведение различных видов отчетной документации. Большинство из них представляют собой разновидности специализированных систем электронного документооборота, при этом стандартные функции управления такого документооборота в рассмотренных системах представлены минимально.

В связи с этим возникает необходимость создания полномасштабных ИС, ориентированных на мониторинг, анализ и управление безопасностью производства как состоянием этих процессов, обусловленных действием различных групп вредных факторов. Под процессом следует понимать совокупность взаимосвязанных и взаимодействующих видов деятельности, преобразующих входы (материалы, информационные потоки) в выходы, которые представляют ценность для предприятия (согласно стандарту ISO 9000:2005).

Такая ориентация ИС требует перехода от традиционного представления процессов предприятия как человеко-машинных систем к процессным моделям, позволяющим описать (в том числе на формальном уровне) взаимосвязь конкретного процесса предприятия, механизмов и условий его выполнения.

По отношению к ИС управления безопасностью производства предлагаемое представление позволит формально описать взаимосвязи между производственными процессами как разновидность бизнес-процессов предприятия, исполнителями этих процессов и требованиями к безопасности труда как к ограничению воздействия вредных и опасных производственных факторов на исполнителей этого процесса. Однако к настоящему времени такие модели для ИС управления безопасностью производства практически не разработаны.

2. Выделение нерешенной части проблемы и постановка задачи исследования

Как показано выше, существующие ИС и технологии управления безопасностью производства ориентированы, прежде всего, на формирование и ведение различных видов отчетной документации. *Главной проблемой*, затрудняющей разработку процессных моделей управления безопасностью процессов предприятия, является отсутствие системного подхода к процессному описанию задач управления безопасностью деятельности сотрудников предприятия. Суть этой *проблемы* заключается в отсутствии базовых концепций построения и формальных описаний общесистемных решений, предназначенных для интеграции отдельных известных задач мониторинга, анализа и управления безопасностью предприятия и его процессов. В основе подобных формализованных представлений, согласно современным стандартам ISO 15288 и ISO 12207, находится формализованное описание такого понятия как «архитектура системы» [1, 2].

Под термином «архитектура системы» в настоящее время подразумеваются фундаментальные понятия и свойства системы в окружающей ее среде, воплощенные в ее элементах, отношениях, а также в принципах ее проектирования и развития [3]. Для практического применения в ходе создания ИС и технологий используется понятие «описание архитектуры» – артефакт, который выражает архитектуры для понимания, анализа и сравнения возможных для системы архитектур. Данный артефакт зачастую используется как своего рода «чертеж» для проектирования и конструирования ИС и технологий (рис. 1) [3, 4].

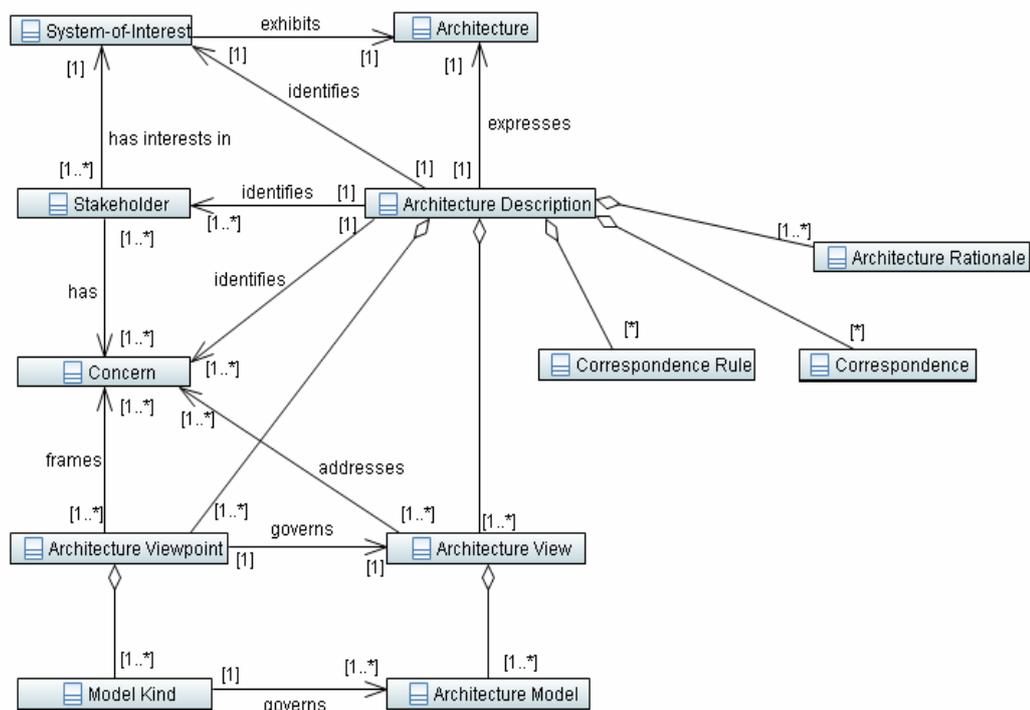


Рис. 1. Диаграмма классов, описывающая ядро понятия «описание архитектуры»

Из рис. 1 следует, что описание архитектуры как основной артефакт создаваемой информационно-аналитической системы (ИАС) управления безопасностью производства определяется, главным образом, следующим:

а) перечнем физических и юридических лиц (Stakeholder), заинтересованных в создании ИАС управления безопасностью производства, а также их видением данной системы, отражающим интересы этих лиц;

б) перечнем проблем заинтересованных лиц, для разрешения которых создается ИАС управления безопасностью производства;

в) перечнем архитектурных точек зрения как наборов конвенций конструирования, интерпретации, использования и анализа конкретного типа взгляда на архитектуру создаваемой ИАС управления безопасностью производства;

г) перечнем взглядов на архитектуру создаваемой ИАС управления безопасностью производства, воплощенных в виде наборов архитектурных моделей.

Рассмотренный подход к описанию архитектуры позволяет сформулировать цель данного исследования: разработка основных архитектурных решений ИАС управления безопасностью производства в соответствии с процессными представлениями о воздействии вредных производственных факторов на участников производства.

Такая точка зрения на описание архитектуры как основной артефакт создаваемой ИАС управления безопасностью производства позволяет сформулировать основную задачу исследования: определение той архитектурной точки зрения, на основе которой данный артефакт может быть создан.

3. Основные материалы исследования

Главной аксиомой описания архитектуры ИАС управления безопасностью производства следует признать данное в [4] определение информационной системы, состоящей из персонала и комплекса средств автоматизации и направленной на достижение главной цели деятельности ИС – формирование и отображение единого целостного информационного представления объекта или процесса в соответствии с поставленными перед ней задачами. Под комплексом средств автоматизации следует понимать вычислительное и коммуникационное оборудование, программное обеспечение, лингвистические средства и информационные ресурсы, нормативные и распорядительные документы и прочие средства, обеспечивающие главную цель деятельности ИС.

Данное определение выделяет как основную архитектурную модель создаваемой ИАС управления безопасностью производства модель, описывающую хранимые и обрабатываемые структуры данных. Поэтому решение задачи исследования определяется тем представлением данных, которое формируется, исходя из формального описания задач обеспечения безопасности производства.

Технологические процессы с точки зрения условий труда в настоящее время характеризуются такими показателями как уровень его безопасности, уровень охраны, уровень условий труда, функциональное состояние оператора и т.п.

Учитывая воздействие на организм многих факторов разной природы, для совершенствования физиолого-гигиенической регламентации трудовой деятельности необходима суммарная оценка нагрузки с точки зрения условий труда.

Естественно считать, что если при одинаковых условиях труда одно изделие требует для своего изготовления больше времени (больше персонала), чем другое, то суммарная нагрузка с точки зрения условий труда, **insalubrity**, т.е. вредность для здоровья от процесса изготовления такого изделия соответственно выше. Однако интегральные показатели для конкретных реализаций одного и того же процесса являются случайными величинами. Поэтому процесс в целом будет характеризоваться средними значениями этих величин. В качестве такого показателя количественных характеристик вредных условий труда предлагается математическое ожидание (среднее значение) суммарной нагрузки за все время осуществления процесса. Такая **insalubrity (IAB)** должна суммироваться по всему персоналу, занятому в процессе [5].

Рассмотрим совокупность значений вредных производственных факторов (ВПФ), действующих на одного работника в некий момент времени как вектор следующего вида:

$$[\phi^{1k}(t_j), \phi^{2k}(t_j), \dots, \phi^{ik}(t_j), \dots, \phi^{nk}(t_j)],$$

ϕ – вектор значений ВПФ, действующих на k -го работника в момент времени t_j (когда проводится замер).

Пусть за время с момента начала участия k -го работника и до момента окончания его участия в процессе t_m проводится m замеров. Тогда среднее значение одного ВПФ ϕ^{ik} за промежуток времени $[t_1, t_m]$ вычисляется по формуле [5]:

$$\mu_{\phi^{ik}(T)} = \frac{\sum_{j=1}^m \phi^{ik}(t_j)}{m}, \quad (1)$$

где $\mu_{\phi^{ik}(T)}$ – среднее арифметическое для m значений i -го ВПФ, действующего на k -го работника в промежуток времени $T=[t_1, t_m]$; m – количество замеров значения i -го ВПФ, действующего на k -го работника в промежуток времени $T=[t_1, t_m]$; t_j – момент времени, в который осуществляется j -й замер, $t_1 \leq t_j \leq t_m$.

Тогда вектор усредненных значений ВПФ, действующих на k -го работника в промежуток времени $T=[t_1, t_m]$, будет иметь вид

$$[\mu_{\phi^{1k}(T)}, \mu_{\phi^{2k}(T)}, \dots, \mu_{\phi^{ik}(T)}, \dots, \mu_{\phi^{nk}(T)}]. \quad (2)$$

Для всего персонала суммарная **insalubrity** участия в процессе за промежуток времени $T=[t_1, t_m]$ будет рассматриваться как величина, имеющая следующий вид [5]:

$$IAB = \sum_{k=1}^S IAB_k = \sum_{k=1}^S [\mu_{\phi^{1k}(T)}, \mu_{\phi^{2k}(T)}, \dots, \mu_{\phi^{ik}(T)}, \dots, \mu_{\phi^{nk}(T)}]. \quad (3)$$

Предлагаемое представление суммарной вредной нагрузки IAB позволяет описать воздействие комплекса ВПФ на состояние организма человека, оценить нагрузку с точки зрения условий труда и безопасности производства.

Выражение (3) позволяет сформулировать базовую концепцию построения ИАС управления безопасностью производства как концепцию формирования единого целостного и непротиворечивого представления о воздействии вредных производственных факторов на организм персонала предприятия, участвующего в выполнении процессов этого предприятия. Поскольку каждый процесс может быть описан выражением (3), то базовая концепция ИАС управления безопасностью производства может быть представлена как результат объединения формальных описаний воздействия комплекса вредных производственных факторов по каждому из процессов предприятия. Такое объединение может производиться на основе многомерной модели данных, которая позволяет организовать сбор, обработку и отображение больших массивов данных о комплексном воздействии разнородных вредных производственных факторов. Пусть на исследуемом предприятии будет n процессов, каждый из которых рассматривается с точки зрения безопасности жизнедеятельности выражением (3). Тогда формально концепцию построения ИАС управления безопасностью производством можно описать следующим образом:

$$M_{IAS} = \bigcup IAB_i, \quad i = 1, \dots, n, \quad (4)$$

где IAB_i – суммарная **insalubrity**, описываемая выражением (3).

Данная модель позволяет рассматривать ИАС управления безопасностью производства как результат объединения описаний векторов усредненных значений ВПФ, в котором описания показателей ВПФ, общие для ряда процессов предприятия, не будут дублироваться в схеме хранилища данных. Такая концепция позволяет сделать вывод о том, что главной архитектурной точкой зрения на ИАС управления безопасностью производства является точка зрения, представляющая эту систему как разновидность систем поддержки принятия решений (СППР), основанных на хранилище данных.

Поэтому следующим шагом в формировании архитектурной точки зрения на создаваемую ИАС является выбор концепции построения СППР, наиболее соответствующего специфике ИАС управления безопасностью производства.

В настоящее время выделяют такие концепции построения СППР [6]:

- а) с физическим хранилищем данных;
- б) с виртуальным хранилищем данных;
- в) на основе витрин данных;
- г) СППР, использующая физическое хранилище данных как их централизованный источник для отдельных витрин данных.

Анализ достоинств и недостатков рассмотренных концепций построения СППР позволяет выделить в качестве основной архитектурной точки зрения на создаваемую ИАС управления безопасностью производства концепцию построения СППР с физическим хранилищем данных. Схема структуры такой СППР показана на рис. 2 [6].

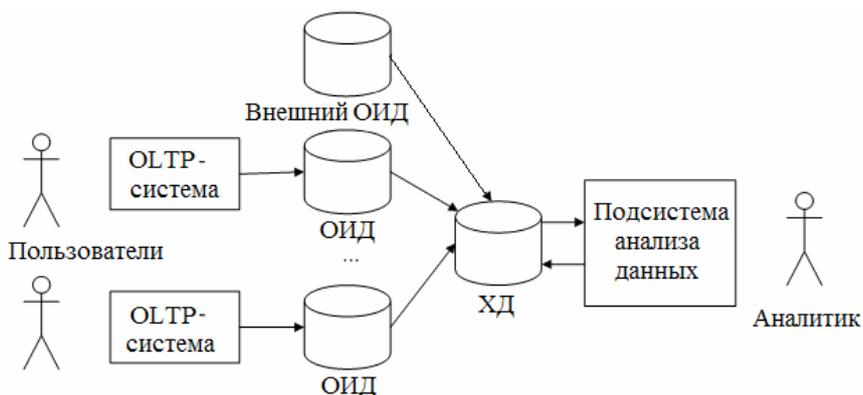


Рис. 2. Схема структуры системы поддержки принятия решений с физическим хранилищем данных

Данная архитектурная точка зрения позволяет рассматривать описание архитектуры ИАС управления безопасностью производства как артефакт, обеспечивающий реализацию системы со следующей структурой (рис. 3).

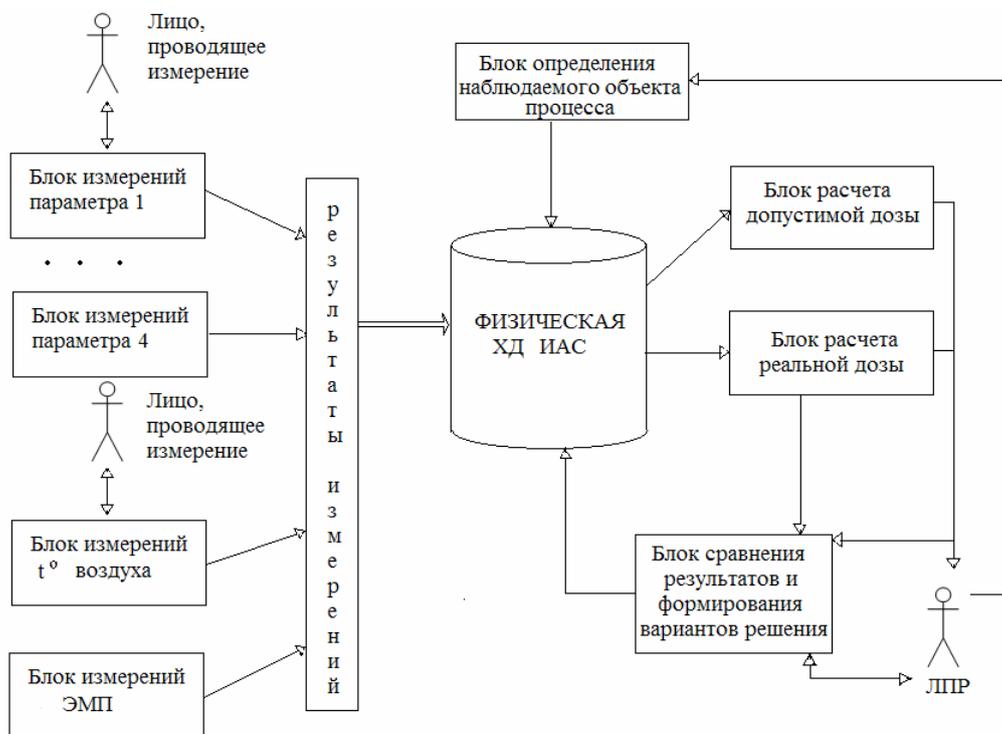


Рис. 3. Схема структуры информационно-аналитической системы управления безопасностью производства

На основе результатов математического моделирования такого типа производств рабочая производственная среда характеризуется следующими вредными факторами: повышенные уровни электромагнитных полей и излучений, параметры производственного микроклимата (повышенная температура и влажность воздуха), производственный шум. Кроме того, возможно случайное эпизодическое воздействие высоких температур, паров свинца и олова (при пайке), электрического тока. Концептуальная схема физического хранилища данных ИАС управления безопасностью производства имеет вид, показанный на рис. 4.



Рис. 4. Обобщенная концептуальная схема хранилища данных информационно-аналитической системы управления безопасностью производства

4. Выводы и перспективы дальнейших исследований

Полученная структурная схема ИАС и обобщенная концептуальная схема хранилища данных позволяют сформировать набор спецификаций на разработку обеспечивающей части ИАС управления безопасностью производства. В ходе ее разработки основное внимание следует уделить информационному, математическому и программному обеспечению ИАС.

Заслуживающей внимание *перспективой* дальнейшего исследования является изучение возможности автоматизированной настройки схемы хранилища данных ИАС на особенности предметной области, определяемые конкретным видом математических моделей задач управления безопасностью производства. Проведение данных исследований позволит сформировать типовую ИАС управления безопасностью производства, настройка которой на конкретный тип производства требует минимальных затрат.

Список литературы: 1. *ГОСТ ИСО/МЭК 15288–2005*. Системная инженерия. Процессы жизненного цикла систем [Текст]. Введ. 01–01–2007. М.: Стандартинформ, 2006. 57 с. 2. *ГОСТ ИСО/МЭК 12207–2010*. Системная и программная инженерия. Процессы жизненного цикла программных средств [Текст]. Введ. 01–03–2012. М.: Стандартинформ, 2011. 106 с. 3. *Сайт ISO/IEC/IEEE 42010 Website* [Электронный ресурс]. Режим доступа: <http://www.iso-architecture.org/ieee-1471/index.html>. Заголовок с экрана. 4. *Левыкин В.М.* Паттерны проектирования требований к информационным системам: моделирование и применение / В.М. Левыкин, М.В. Евланов, М.А. Керносов. Харьков: ООО «Компанія «Сміт», 2014. 320 с. 5. *Сердюк Н.Н.* Функциональная задача оценки влияния вредных производственных факторов на человека // *Східно-Європейський журнал передових технологій*. 2013. № 4/4 (64). 6. *Барсегян А.А.* Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP / А.А. Барсегян, М.С. Куприянов, В.В. Степаненко, И.И. Холод. СПб.: БХВ-Санкт-Петербург, 2008. 384 с.

Поступила в редколлегию 20.05.2014

Сердюк Наталья Николаевна, ассистент кафедры охраны труда ХНУРЭ. Научные интересы: управление условиями труда на рабочем месте. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 7021-360.

МЕТОД СНИЖЕНИЯ ИНТЕНСИВНОСТИ ВИДЕОПОТОКА В ИНФОКОММУНИКАЦИОННЫХ СИСТЕМАХ

Показывается актуальность вопросов, связанных с повышением качества предоставления видеoinформационных услуг с использованием беспроводных телекоммуникационных технологий. Обосновывается важность уменьшения нагрузки на сеть на основе снижения интенсивности сжатого видеопотока. Выявляются уязвимые стороны функционирования стандартизированных технологий обработки базовых кадров. Рассматриваются отличительные этапы технологии кодирования базовых кадров для снижения интенсивности их кодового представления. Излагаются основные этапы оценки интенсивности потока, приходящегося на базовый кадр, с учетом формирования кодовых конструкций сжатого представления трансформант на основе диагонально-неравномерного позиционного кодирования. Разрабатывается метод для оценки интенсивности на группу кадров и всего видеопотока с учетом: дифференцированного вклада типов кадров в интенсивность и качество визуального восприятия реконструированного видеопотока; компрессии базовых кадров на основе их трансформирования и последующего диагонально-неравномерного позиционного кодирования.

1. Введение

Совершенствование сектора предоставления видеoinформационных услуг с использованием беспроводных телекоммуникационных технологий связан с необходимостью снижения нагрузки на сеть [1, 2]. Это неизбежно повысит качество получаемой видеoinформации. Поэтому уменьшение интенсивности потока кадров в процессе их компрессии является актуальным направлением научно-прикладных исследований [3-5]. Здесь ключевая составляющая интенсивности потока определяется интенсивностью кодового представления базового кадра. Это обусловлено тем, что базовый кадр в группе формирует кадры Р-типа и В-типа [4-6]. Для обработки базового кадра используется принципиально новый метод снижения интенсивности его кодового описания. Он базируется на диагонально-неравномерном позиционном кодировании трансформант в неравновесном базисе оснований. В то же время для такого метода отсутствует соответствующий метод оценки интенсивности. Отсюда *цель исследования* состоит в разработке метода для оценки интенсивности видеопотока с учетом использования нового кодека для кодирования базовых кадров.

2. Обоснование необходимости обработки базового кадра

Базовой структурной единицей MPEG-потока является групповой кадр (ГК, Group of Pictures – GOP). Он состоит из нескольких кадров разного типа. Каждый тип кадров обрабатывается с использованием методов на JPEG совместимой платформе. При этом учитывается иерархия относительно зависимости порядка реконструкции кадров в группе. В результате устанавливается неравнозначное влияние кадров различных типов на качество реконструкции всех кадров в GOP. Для JPEG технологий такая неравнозначность учитывается путем выбора соответствующей стратегии квантизации. Здесь вклад интенсивности кодового представления базового кадра в суммарную интенсивность видеопотока в зависимости от качества визуальной реконструкции кадров изменяется от 50 до 75 % и возрастает с увеличением качества реконструкции видеопотока. Это указывает на значимое влияние интенсивности базового кадра на суммарную интенсивность видеопотока.

3. Оценка уязвимостей существующих технологий обработки базового кадра

Стандартизированные технологии обработки базовых кадров MPEG-технологий характеризуются наличием уязвимостей, а именно:

- 1) степень сжатия достигает наибольших значений для низких величин ПОСШ;

2) в случае необходимости обеспечить ПОСШ на уровне 45 дБ интенсивность кодового представления необходимо дополнительно снизить в 1,5 раза;

3) для базовых кадров с высоким пространственным разрешением, чтобы обеспечить своевременную доставку с использованием инфокоммуникационных систем с пропускной способностью $U_n \geq 100$ Мбит/с, при наличии искажений на уровне ПОСШ 25 – 30 дБ интенсивность сжатого видеопотока необходимо уменьшить как минимум в 1,3 раза;

4) для протокола TCP при увеличении числа узлов от 1 до 20 задержка увеличится в 8 раз, а при появлении потерь пакетов до 20% от их общего числа - в 18 раз;

5) в случае использования протокола UDP потерянные или поврежденные пакеты в ходе передачи не передаются вновь передающей стороной и при этом считается, что все сообщение является доставленным до конечного оборудования. Анализ результатов обработки базовых кадров в зависимости от количества потерянных пакетов позволяет заключить, что для насыщенных базовых кадров значение ПОСШ снижается в среднем на 86% при 1% потерь пакетов и на 92% соответственно при 3% потерь пакетов.

Поэтому необходимо создавать новые кодеки базовых кадров.

4. Разработка технологии кодирования базового кадра

Принципиально новый кодек базового кадра основан на том, что компрессия трансформанты обеспечивается в результате диагонально-неравномерного позиционного кодирования. Особенность метода заключается в том, что:

1) позиционные числа строятся на базе неравномерных диагоналей трансформанты;

2) основания элементов диагоналей определяются как значения динамических диапазонов на основе неравномерно-диагонального метода выявления;

3) первая и последняя диагонали исключаются из базиса оснований НДП чисел и обрабатываются отдельно. Это обусловлено тем, что: первая диагональ содержит низкочастотную компоненту трансформанты, которая имеет резко отличительные характеристики относительно других компонент трансформанты; последняя диагональ для достаточно широкого класса изображений будет содержать компоненту с нулевым значением.

5. Оценка интенсивности кодового представления базового кадра

Проведем сначала оценку интенсивности I_1 для компактно-представленного кадра I-типа с использованием разработанного метода.

Здесь длина $\bar{Q}_1^{(n \times n)}$ кодовой конструкции сжатого представления фрагмента определяется по формуле

$$\bar{Q}_1^{(n \times n)} = \bar{Q}_n + \bar{Q}_Z + \bar{Q}_R, \quad (1)$$

где \bar{Q}_n – количество разрядов на представление кодограммы компрессионного описания трансформанты; \bar{Q}_Z – компактное представление двоичной матрицы знаков; \bar{Q}_R – количество бит на представление шага R квантизации.

Соответственно величина \bar{Q}_n определяется как суммарное количество разрядов, задаваемое следующей формулой:

$$\bar{Q}_n = \bar{Q}_{DC} + \bar{Q}_{(n \times n)-2} + \bar{Q}_{v_d} \quad (\text{бит}), \quad (2)$$

здесь \bar{Q}_{DC} – длина статистического кода низкочастотной DC-компоненты; $\bar{Q}_{(n \times n)-2}$ – количество разрядов на представление трансформанты путем диагонально-неравномерного позиционного кодирования без учета первой и последней диагоналей; \bar{Q}_{v_d} – количество разрядов на представление последней диагонали трансформанты.

Соответственно суммарное количество $\bar{Q}'_{(n \times n)-2}$ разрядов на представление всех кодовых значений диагональных неравномерных позиционных чисел будет определяться на основе соотношения

$$\bar{Q}_{(n \times n)-2} = \sum_{\xi=2}^{2n-2} \bar{Q}_{\xi, n_{\xi}} = \sum_{\xi=2}^{2n-2} ([n_{\xi} \ell \log_2 d'_{\xi}] + 1) \text{ (бит)}. \quad (3)$$

Свернув выражения (1)–(3) в одно, получим соотношение для величины $I_1^{(n \times n)}$, а именно:

$$I_1^{(n \times n)} = \bar{Q}_{DC} + \sum_{\xi=2}^{2n-2} ([n_{\xi} \ell \log_2 d'_{\xi}] + 1) + \bar{Q}_{v_d} + \bar{Q}_Z + \bar{Q}_R. \quad (4)$$

На основе данного выражения интенсивность I_1 потока, приходящегося на один базовый кадр, будет формироваться как сумма интенсивностей его фрагментов, т.е.

$$I_1 = \sum_{i=1}^{MN/nn} I_{1,i}^{(n \times n)}. \quad (5)$$

Здесь $I_{1,i}^{(n \times n)}$ – интенсивность i -го фрагмента базового кадра; MN/nn – количество фрагментов размером $(n \times n)$ в базовом кадре; MN – размер базового кадра.

6. Оценка интенсивности видеопотока

Рассмотрим теперь оценку интенсивности $I(\text{GOP})'_k$ компрессированного видеопотока, приходящуюся на группу кадров, с использованием разработанной технологии диагонально-неравномерного позиционного кодирования. Здесь принимается в расчет, что группа состоит из 8 кадров и включает в себя один кадр I-типа, два кадра P-типа и пять кадров B-типа. Обработка кадров проводится в соответствии с режимом реального времени. В этом случае не учитывается компенсация движения между кадрами в потоке и допускается, что степень сжатия для кадров конкретного типа будет отличаться не менее, чем на 5%. Тогда с учетом выражения для интенсивности базового кадра получим следующее соотношение для оценки интенсивности $I(\text{GOP})'_k$:

$$\begin{aligned} I(\text{GOP})_k &= I_I + 2I_P + 5I_B = \sum_{i=1}^{MN/nn} I_{1,i}^{(n \times n)} + 2I_P + 5I_B = \\ &= \sum_{i=1}^{MN/nn} (\bar{Q}_{DC} + \sum_{\xi=2}^{2n-2} ([n_{\xi} \ell \log_2 d'_{\xi}] + 1) + \bar{Q}_{v_d} + \bar{Q}_Z + \bar{Q}_R)_i + 2I_P + 5I_B, \end{aligned} \quad (6)$$

где I_I , I_P , I_B – значения интенсивностей для компактно-представленных кадров соответственно I-типа, P-типа и B-типа.

Здесь величина $I(\text{GOP})'_k$ оценивается как количество бит, передаваемое за время t_{GOP} , равное $t_{\text{GOP}} = 8/v_t$ (с), где v_t – количество кадров, которое передается за 1 с.

Каждый тип кадров обрабатывается с использованием методов на JPEG совместимой платформе. При этом учитывается иерархия относительно зависимости порядка реконструкции кадров в группе. В результате устанавливается неравнозначное влияние кадров различных типов на качество реконструкции всех кадров в GOP.

Оценка интенсивностей $I(\text{GOP})'_k$ и $I_k^{(24)}$, приходящихся соответственно на группу кадров за время t_{GOP} и на v_t кадров за 1 с, рассматривается в таблице и на диаграммах рисунка. Здесь $v_t = 24$ кадрам. Расчеты проводятся по формуле (6). В последней строке таблицы приводятся расчеты величины $\Delta I(\text{GOP})'_k$, оцениваемой в процентах как уровень снижения интенсивности потока группы кадров за счет уменьшения интенсивности базового кадра. Получение диаграмм на рисунке осуществлялось для трех режимов обработки кадров с учетом обеспечения следующих уровней ПОСШ:

– режим 1, высокое качество базового кадра: I-кадр – 42 дБ, P-кадр – 28 дБ, B-кадр – 22 дБ;

– режим 2, хорошее качество базового кадра: I-кадр – 40 дБ, P-кадр – 28 дБ, В-кадр – 22 дБ;

– режим 2, достаточное качество базового кадра: I-кадр – 38 дБ, P-кадр – 28 дБ, В-кадр – 22 дБ.

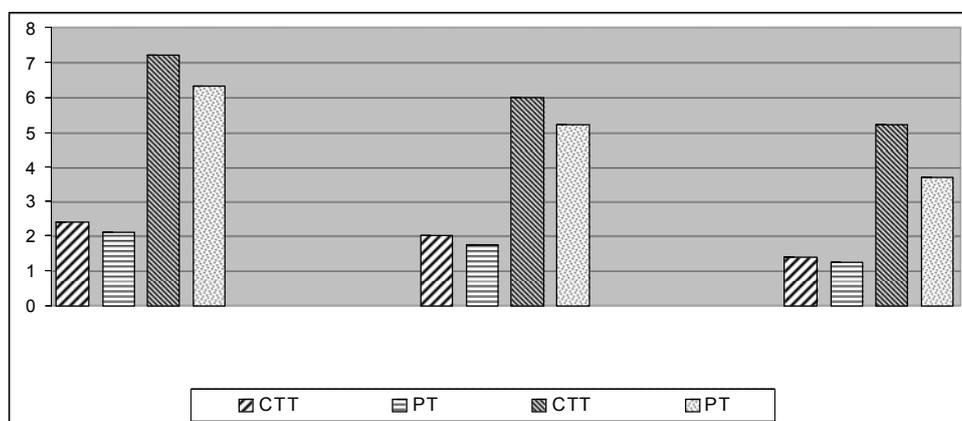
По результатам исследований данных в таблице и диаграмм на рисунке можно сделать такие заключения:

1. Интенсивность потока кадров с учетом использования для сжатия базового кадра созданной технологии изменяется в пределах от 3,7 Мбит/с для режима достаточного визуального качества до 6,3 Мбит/с для режима хорошего качества.

2. За счет обработки базовых кадров на основе диагонально-неравномерного позиционного кодирования трансформант достигается снижение суммарной интенсивности видеопотока от 10 до 13 %. Наибольшее снижение интенсивности происходит в случае наиболее весомого вклада интенсивности базовых кадров в суммарную интенсивность. Такое наблюдается для режима высокого визуального качества реконструкции видеопотока.

Зависимость интенсивностей $I(\text{GOP})'_k$ и $I_k^{(24)}$ от ПОСШ, М бит/с

Тип кадра	h дБ					
	42	40	40	40	38	38
I-кадр	42	40	40	40	38	38
P-кадр	28	28	30	32	28	30
В-кадр	22	22	24	24	22	24
$I(\text{GOP})'_k$	2,13	1,76	2,01	2,17	1,24	1,49
$I_k^{(24)}$	6,3	5,2	6	6,4	3,7	4,3
$\Delta I(\text{GOP})'_k, \%$	-11,25	-12	-10,7	-9,6	-11,43	-9,7



Значения интенсивностей $I(\text{GOP})'_k$ и $I_k^{(24)}$ для стандартизированной (СТТ) и разработанной (РТ) технологии обработки базовых кадров в зависимости от режимов ПОСШ видеопотока

3. Снижение интенсивности видеопотока обеспечивает:

– с одной стороны, использование для его передачи беспроводных технологий с более низкой пропускной способностью, а именно на уровне 4 - 10 Мбит/с;

– с другой стороны, повышение качества визуальной оценки видеопотока за счет использования соответствующих матриц коррекции компонент трансформант. В первую очередь появляется возможность снизить искажения без повышения интенсивности видео-

потока одновременно для Р-кадров с уровня 28 до уровня 32 дБ (таблица, второй и четвертый столбцы) и для кадров В-типа с 22 дБ до 24 дБ.

По материалам исследований можно сделать следующие выводы:

1) разработан метод оценки интенсивности на группу кадров и всего видеопотока с учетом:

- дифференцированного вклада типов кадров в интенсивность и качество визуального восприятия реконструированного видеопотока;
- компрессии базовых кадров на основе их трансформирования и последующего диагонально-неравномерного позиционного кодирования;

2) достигнуто снижение дисбаланса между интенсивностью I_k и скоростью передачи по сети U_n в среднем на 12 %. Это позволяет:

- использовать для передачи видеопотока ИКС технологий с более низкой пропускной способностью, а именно на уровне 4 - 10 Мбит/с;
- повысить качество визуальной оценки видеопотока за счет снижения искажений без повышения интенсивности видеопотока одновременно для Р-кадров с уровня 28 до уровня 32 дБ (таблица, второй и четвертый столбцы) и для кадров В-типа с 22 до 24 дБ.

Список литературы: 1. *Олифер В.Г.* Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 3-е изд. / *В.Г. Олифер, Н.А. Олифер.* СПб.: Питер, 2006. 958 с. 2. *Gonzales R.C.* Digital image processing / *R.C. Gonzales, R.E. Woods.* - Prentice Inc. Upper Saddle River, New Jersey 2002. 779 p. 3. *Lee S.Y.* Temporally coherent video matting / *S.Y. Lee, J.C. Yoon, I.K. Lee* // Graphical Models 72. 2010. P. 25 – 33. 4. *Красильников Н.Н.* Цифровая обработка изображений. М.: Вузовская книга, 2011. 320 с. 5. *Баранник В.В.* Кодирование трансформированных изображений в инфокоммуникационных системах / *В.В. Баранник, В.П. Поляков.* Х.: ХУПС, 2010. 234 с. 6. *Barannik V.* Method Of Encoding Transformant Uolsha Is In Systems Air Monitoring Of Earth / *V. Barannik, A. Yakovenko, A. Krasnorutkiy* // Lviv-Slavsko, Ukraine, Lviv Polytechnic National University, International Conference TCSET'2009, Modern problems of radio engineering, telecommunications and computer science, February 19 – 23, 2009. P. 381-383.

Поступила в редколлегию 23.05.2014

Баранник Владимир Викторович, д-р техн. наук, профессор, начальник кафедры боевого применения и эксплуатации АСУ Харьковского университета Воздушных Сил. Научные интересы: обработка и передача информации. Адрес: Украина, 61023, Харьков, ул. Сумская, 77/79.

Отман Шади О.Ю., аспирант ХНУРЭ. Научные интересы: обработка и передача информации. Адрес: Украина, 61166, Харьков, ул. Ленина, 14.

Хаханова Анна Владимировна, канд. техн. наук, доцент кафедры АПВТ ХНУРЭ. Научные интересы: обработка и передача информации. Адрес: Украина, 61166, Харьков, ул. Ленина, 14.

ПРЕДСТАВЛЕНИЕ ОБРАЗОВ ДИНАМИЧЕСКИХ ПРОЦЕССОВ В СИСТЕМАХ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ С ПОМОЩЬЮ САМОНАСТРАИВАЮЩИХСЯ АГЕНТОВ

Исследуется возможность использования универсальной структуры программного агента для распознавания образов входных сигналов и задания функциональных зависимостей. Предлагается модель и UML-диаграммы самонастраивающегося агента с учетом использования принципов объектно-ориентированного программирования и контейнерных классов работы со списками. Описывается метод организации работы программных агентов для автоматической аппроксимации переходных характеристик динамических объектов. Исследуется качество воспроизведения переходных процессов на основе метода идентификации запомненных образов и показывается принципиальная возможность вносить в структуру агентной системы априорные знания о динамическом объекте.

1. Введение

Перспективные направления развития информационно-управляющих систем характеризуются стремлением разработчиков передать рутинные, но слабоавтоматизируемые функции инженеров-разработчиков и операторов технической системе. Это требует развития информационных и программных технологий поддержки извлечения знаний, самообучения, принятия решений в условиях неопределенностей. Также для разработки современных информационных систем актуальна оптимизация поведения в условиях изменяющейся внешней среды, анализ эмпирического опыта, что, в определенной степени, входит в конфликт с жесткими ограничениями на время реакции. Кроме того, для подобных систем особенно остро стоит вопрос корректности их поведения. Частично проблему помогает решить тщательное и обширное тестирование, но гарантированно охватить все возможные ситуации за разумное время невозможно. Таким образом, целесообразно использовать интеллектуальные свойства автоматизированных систем для анализа правильности собственных действий и определения риска или факта возникновения нештатных ситуаций [1].

Такая ситуация характерна как для систем поддержки принятия решений и технологической диспетчеризации, так и для систем автоматического управления и регулирования технологических параметров. Методы разработки одноконтурных линейных и экстремальных систем автоматического регулирования хорошо исследованы и внедрены в инженерную практику, а автоматизация сложных объектов по-прежнему характеризуется большой науко- и трудоемкостью. Под термином “сложные объекты” понимаются технологические объекты, которые характеризуются количеством переменных состояния больше двух, перекрещивающимися обратными связями в своей структуре, нелинейными или нестационарными свойствами, наличием стохастических или неформализуемых составляющих в реакциях на входные воздействия, расширенным набором управляющих и осведомительных сигналов, для которого применение формальных методов синтеза затруднено.

Разработка подходов для придания техническим системам универсальных интеллектуальных свойств, упрощающих синтез таких систем и улучшающих качество работы, может базироваться на применении самонастраивающихся агентов [2], которые в совокупности смогут аккумулировать знания, принимать решение на изменение своей структуры, оценивать качество управления и прогнозировать протекание управляемых процессов на основе доступной информации.

Начальным шагом такой технологии будет исследование принципов автоматического разбиения мгновенных сигналов динамических объектов во временной области на узнаваемые образы и прогнозирование их будущей реакции группой самонастраивающихся агентов.

2. Структура агента

Для понимания способа организации интеллектуальных программных агентов, функционирующих по детерминированным и простым алгоритмам, необходимо перечислить желаемые свойства и варианты настройки, отталкиваясь от которых можно перейти к программной реализации и исследованию эффективности предложенного подхода.

К основным задачам агентов можно отнести:

- распознавание векторов входных сигналов и определение степени принадлежности текущего вектора к запомненному;
- вычисление функциональных зависимостей выхода агента от собственного состояния и входных сигналов;
- хранение логических правил (продукций) и вычисление продукционных заключений;
- формирование тактовых импульсов и тактированное восприятие входных сигналов для придания динамических свойств отдельному агенту или агентной группе;
- интерпретация вектора управляющих сигналов для включения-выключения агента и модификации алгоритма его работы.

Входные сигналы агента можно разделить на четыре группы: вектор текущих сигналов; вектор запомненных сигналов; вектор управляющих флагов; вектор весовых коэффициентов и констант. К выходным сигналам относятся – степень принадлежности текущего вектора входных сигналов запомненному (α), выходная константа (ζ), их произведение ($\alpha \cdot \zeta$). Выход произведения функции активации на выходную константу проходит через элемент настраиваемой временной задержки (Delay). Тип данных всех сигналов – вещественный.

На рис. 1 показана структура одиночного агента, реализующая все варианты его функционирования.

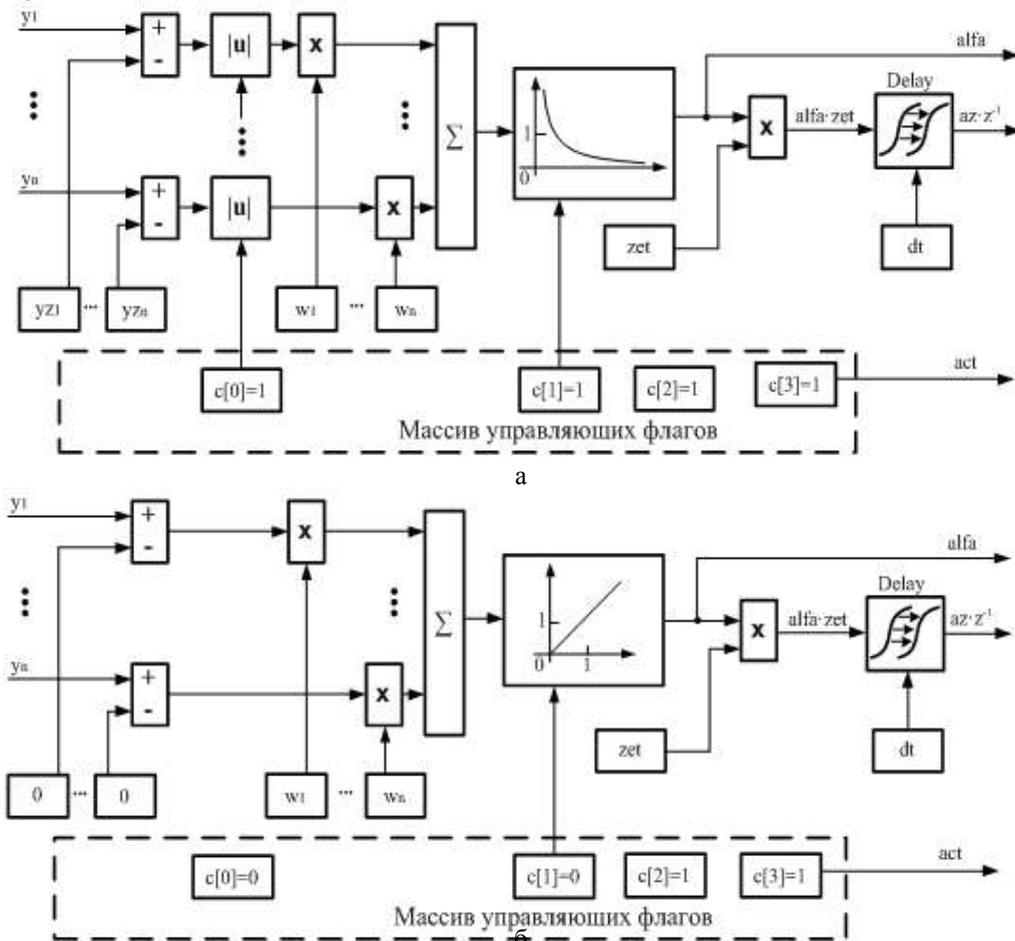


Рис. 1. Структурная схема агента: а – вариант распознавания вектора (флаги $c[0]=1$ и $c[1]=1$); б – вариант для вычисления функциональной зависимости (флаги $c[0]=0$ и $c[1]=0$)

Вариант, при котором управляющие флаги $c[0]=1$ и $c[1]=1$ (рис. 1,а), определяет, что рассчитывается абсолютная величина разности между элементами входного и запомненного векторов. От суммы разностей элементов рассчитывается функция $y=(1/x)$. Это соответствует варианту распознавания, при котором выходной сигнал α тем больше, чем меньше расстояние Хэмминга между векторами.

Второй вариант (рис. 1,б) целесообразен при вычислении некоторых функциональных зависимостей набором агентов, как в нейронной сети с одним скрытым слоем. Известна теорема [3,4], показывающая, что искусственная нейронная сеть может выступать универсальным аппроксиматором и такой вариант агента может соответствовать модели искусственного нейрона.

Элемент программируемой задержки Delay позволяет за счет внедрения обратных связей получать свойства генераторов сигналов, дискретных интеграторов и фильтров.

Приведенной структуры достаточно для программной реализации одиночного агента с учетом принципов объектно-ориентированного программирования. Можно отметить, что в вариантах распознавания векторов и вычисления функциональной зависимости структура агента в большой степени соответствует частям схемы персептронного нейрона. Однако технологии классических нейронных сетей не обладают свойствами самонастройки и самоорганизации, требуют соблюдения условий обучения с учителем, не дают качественную оценку степени распознавания ситуации и адекватности своего поведения в процессе работы. Как будет показано дальше, именно такая минимальная структура агента подходит для решения описанных проблем в составе мультиагентной системы.

3. Функционирование ансамбля агентов в единой группе

Необходимо отметить, что интеллектуальные свойства в задачах классификации и распознавания векторов сигналов начнут проявляться при объединении нескольких агентов в единую структуру. Для ее функционирования необходим групповой программный супервизор, обеспечивающий вычисление четкого итогового значения прогнозируемого сигнала, и добавление нового экземпляра агента в случае низкой степени распознавания текущего вектора (значение степеней принадлежности по всему ансамблю агентов ниже заданного порога). Также супервизор должен обладать возможностью разрешать или запрещать работу всех агентов, устанавливая и сбрасывая специальный контрольный сигнал. Выходы всех агентов (сигналы α и $\alpha\text{-zet}$), работающих в конфигурации распознавания векторов, обрабатываются супервизором по приближенному методу центра тяжести, аналогично методу дефаззификации в алгоритме нечеткого вывода Сугэно [4,5].

Структура элементарной мультиагентной системы показана на рис. 2.

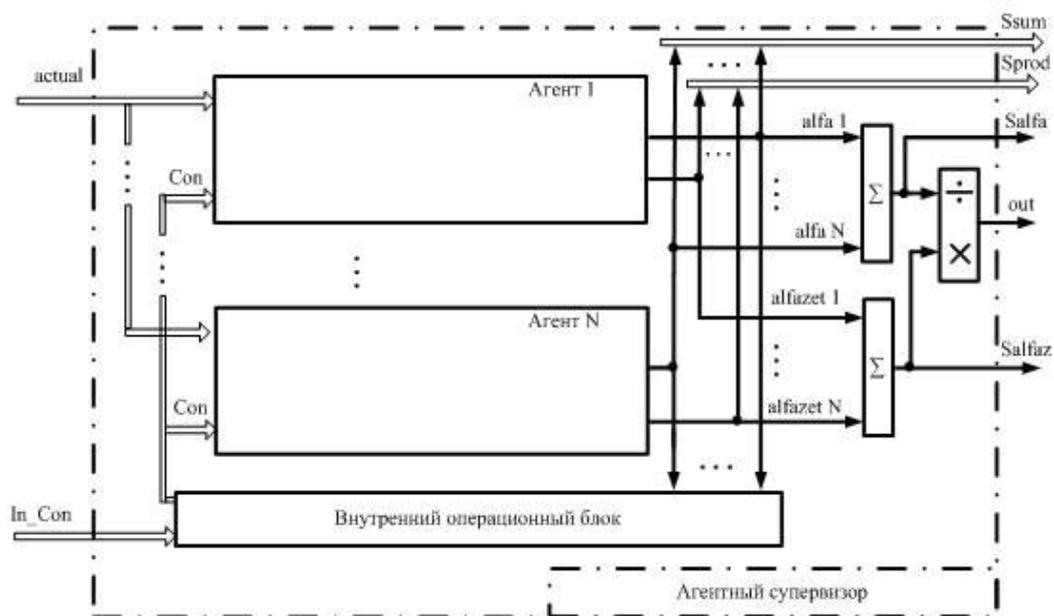


Рис. 2. Структура агентной группы с супервизором

При использовании ее в качестве системы, рассчитывающей будущее значение выходного сигнала динамического объекта (сигнал *out*), вектор *actual* содержит предыдущие временные отчеты выходного сигнала объекта и управляющего воздействия:

$$actual = (y_{i-k \cdot step}, \dots, y_i; u_{i-k \cdot step}, \dots, u_i), \quad (1)$$

где *k* – количество точек временного сигнала, подаваемого на вход агентов; *step* – шаг просмотра входных последовательностей в точках входного сигнала.

Сигналы, входящие в набор *In_Con*, служат для управления логикой работы супервизора. В процессе работы мультиагентной системы в дискретные моменты времени на входы *actual* подается очередной образ управляющих и выходных сигналов объекта управления. В случае, когда сигнал “узнавания” *alfa* всех агентов ниже заданного порога, то текущий входной образ считается новым и для его запоминания генерируется новый агент. В качестве выходной константы *zet* берется выход объекта на следующем шаге работы.

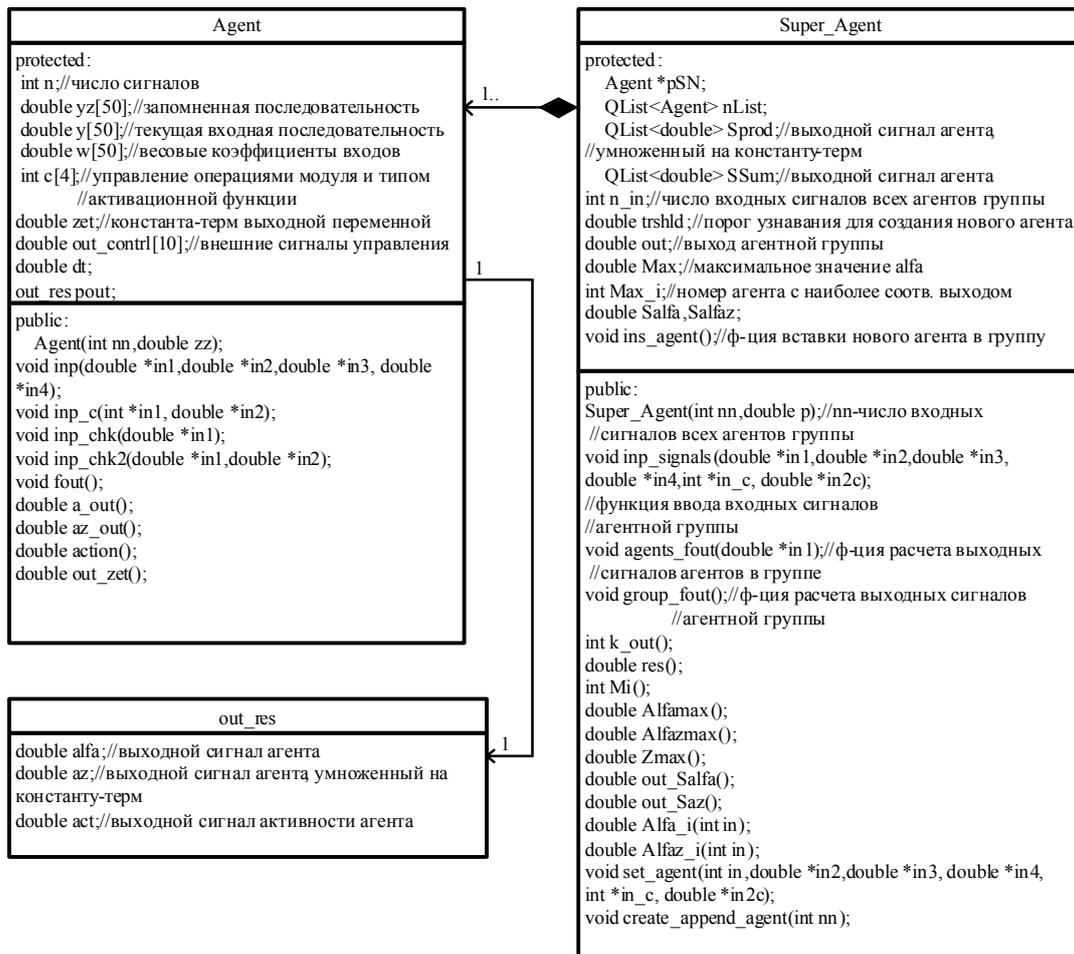


Рис. 3. UML-диаграмма классов тестовой мультиагентной системы

UML-диаграмма классов приложения (рис. 3), позволяющая исследовать эффективность применения агентного подхода для автоматизированного построения модели динамического объекта, содержит классы *Agent* и *Super_Agent*, которые реализуют сущности агентов и их супервизора. Классы связаны отношением композиции и один супервизор (один экземпляр класса *Super_Agent*) может содержать в контейнерном классе *QList* список из любого количества агентов. Структура *out_res* используется для хранения последних рассчитанных значений основных выходных сигналов агента.

Параметрический конструктор класса *Agent* позволяет задать количество входных сигналов и константу-терм *zet*. Остальные методы класса обеспечивают задание информационных и управляющих сигналов, а также расчет и вывод результатов работы агента.

Класс *Super_Agent* содержит:

- в переменной n_in количество входных сигналов группы (оно одинаковое для всех агентов);
- в переменной $trshld$ численное значение порога для сигналов $alfa$, которое служит критерием для создания нового агента в группе (если сигналы $alfa$ всех агентов меньше порогового значения, то образ считается нераспознанным и добавляется новый агент);
- в переменных $Ssum$ и $Sprod$ список основных выходных сигналов по всему ансамблю агентов;
- в переменную out помещается текущий итоговый результат группы, рассчитанный на основании выходов всех агентов;
- процедуру $create_append_agent(...)$ для добавления нового агента в список;
- процедуру $set_agent(...)$ для задания всех констант и коэффициентов нового агента;
- процедуру $inp_signals(...)$ для ввода нового значения входных сигналов во все агенты группы.

Остальные процедуры класса *Super_Agent* выполняют интерфейсные функции.

Предложенная структура классов ориентирована, главным образом, на реализацию задач распознавания образов и идентификации, но может быть легко расширена для вычисления сложных функциональных зависимостей и реализации оптимизационных задач.

4. Описание тестовой задачи

Эффективность предложенного метода идентификации объектов управления и моделей организации мультиагентной системы следует проверить на полностью детерминированных математических моделях объектов. Для данной цели в качестве тестового объекта было выбрано линейное динамическое звено второго порядка, параметры которого определяются пользователем тестовой программы. Работа динамического звена моделируется для серии ступенчатых воздействий, абсолютное значение которых зависит от базового уровня.

На входы мультиагентной системы подаются мгновенные отчеты входного и выходного сигналов динамического звена с заданным шагом, как показано на рис. 4.

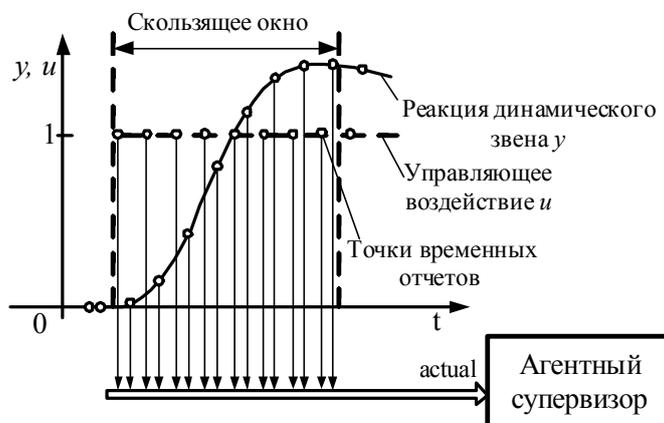


Рис. 4. Схема формирования входных сигналов мультиагентной системы

Первый просмотр серии переходных процессов эквивалентен обучению системы, в процессе которого происходит генерация новых агентов, каждый из которых настроен на свой уникальный образ входного вектора *actual*. Повторный просмотр процессов и расчет выхода мультиагентной системы позволяет оценить качество распознавания и идентификации динамических процессов.

5. Исследование эффективности применения агентной системы

В среде разработки IDE Qt 5.3 было создано тестовое приложение, позволяющее исследовать как работу одиночного агента, так и группы агентов под управлением супервизорного класса. На рис. 5 приведена UML-диаграмма активности приложения, в котором реализовано формирование тестовых примеров, работа мультиагентной системы и визуализация результатов.

Цикл исследования начинается с задания параметров динамического звена второго порядка, его начальных состояний, промежутка модельного времени и постоянного шага интегрирования (дискретности изменения модельного времени). Кроме того, необходимо задать базовое значение входного сигнала x , от которого будут вычислены мгновенные значения входного сигнала динамического звена по формуле:

$$u_i = \begin{cases} 0 \cdot x, \text{if } (i < n); \\ 1 \cdot x, \text{if } ((i \geq n) \& \& (i \leq n \cdot 2)); \\ 1.25 \cdot x, \text{if } ((i \geq n \cdot 2) \& \& (i \leq n \cdot 3)); \\ 0.5 \cdot x, \text{if } ((i \geq n \cdot 3) \& \& (i \leq n \cdot 4)); \\ 0.7 \cdot x, \text{if } ((i \geq n \cdot 4) \& \& (i \leq n \cdot 5)); \\ 0 \cdot x, \text{if } ((i \geq n \cdot 5) \& \& (i \leq n \cdot 6)). \end{cases} \quad (2)$$

Здесь (2): $i = (0, 1, \dots, \frac{t_2 - t_1}{dt})$ – номер точки; $n = \frac{1}{6} \left(\frac{t_2 - t_1}{dt} \right)$ – интервал, на протяжении которого входное воздействие постоянно; t_1, t_2, dt – соответственно, границы модельного времени и шаг его изменения.

Пользователь может создать экземпляр класса, выполняющего функцию численного расчета выхода звена, дать команду на вычисление и вывести в графической форме результаты. На рис. 6 приведено окно графического интерфейса пользователя, открытого на вкладке генерации переходных процессов. Базовое значение входного сигнала равно 0,9 отн. ед., соответственно, получены переходные процессы для последовательно изменяющихся ступенчатых воздействий [0; 0,9; 1,125; 0,45; 0,63; 0], равномерно распределенных на доступном временном интервале. Рис. 6 также демонстрирует результаты моделирования динамического звена при заданных параметрах. Эти данные используются дальше для обучающего цикла работы мультиагентной системы, которая на момент создания экземпляра класса содержит только один агент, который соответствует правилу, что если входной вектор содержит нулевые значения, то и выход агента равен нулю.

Опциональное изменение параметров генерации агентов для тестового примера доступно на вкладке “Настройка” (рис. 7). Можно установить шаг просмотра точек переходных процессов, количество осведомительных входов мультиагентной системы (длину вектора *actual*), пороговое значение критерия генерации нового агента, запуск обучения. Поле “Количество объектов прямой модели” показывает текущее количество сгенерированных агентов, строка прогресса информирует пользователя о количестве просмотренных точек обучающей последовательности.

На вкладке “Тест прямой идентификации”, фрагмент которой показан на рис. 8, в графической форме отображены результаты расчета выхода динамического звена и мультиагентной системы в одной системе координат. Можно сделать вывод о хорошей точности воспроизведения поведения динамического объекта агентной системой при условии совпадения контрольных и обучающих последовательностей. Уменьшая шаг просмотра, увеличивая порог и количество входов агентов, (см.рис. 7) возможно добиться полного совпадения результатов для выбранной дискретности расчета ценой значительного увеличения количества агентов после цикла обучения. По сути, для представленного сценария по окончании цикла обучения формируется продукционная база знаний, в которой каждый агент соответствует простому правилу, а выход всех агентов агрегируется по упрощенному методу центра тяжести, как на этапе дефаззификации в алгоритме Сугэно.

Для дальнейшего развития информационной технологии поддержки интеллектуального управления сложными технологическими процессами на основе самоорганизующихся агентных систем представляет интерес исследование точности воспроизведения реакции динамического объекта для случаев, не входящих в обучающую выборку. Были рассчитаны переходные процессы по приведенному сценарию для базового значения входного воздействия 1,2 отн. ед. Последовательно изменяющиеся ступенчатые воздействия принимали значения [0; 1,2; 1,5; 0,6; 0,84; 0]. Результаты моделирования для контрольного примера показаны на рис. 9.

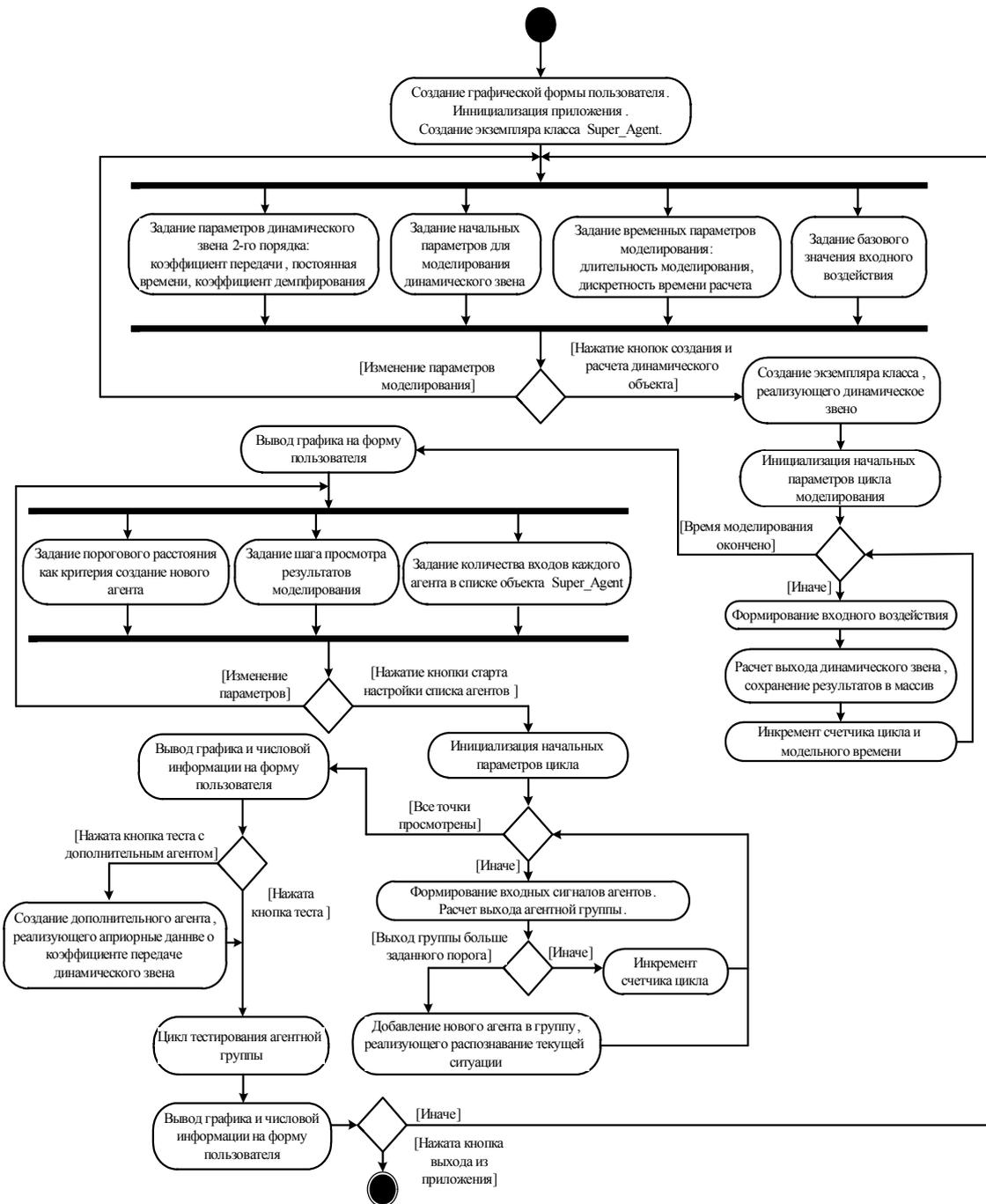


Рис. 5. UML-диаграмма активности работы тестового приложения, использующего агентный супер-визор

Можно отметить, что агентная система работает с абсолютными значениями сигналов и в случае, когда входные сигналы находятся внутри пределов обучающей выборки, наблюдается сохранение подобия кривых и малые (в пределах 10-15%) рассогласования абсолютных значений. Они зависят от заданных шага просмотра и порога узнавания. Если входные сигналы находятся за границами обучающей выборки, то рассогласование выхода агентной системы по сравнению с истинным значением достигает неприемлемо большой величины (кривые 5 и 4 рис. 9).

В ходе исследования была исследована возможность улучшения работы агентной системы путем введения дополнительных агентов, воплощающих априорное знание об

объекте управления. Был директивно создан один экземпляр класса супервизора, который содержал агент, настроенный на детектирование установившегося режима (когда разница между элементами входного вектора нулевая). В этом случае супервизор выдавал сигнал, пропорциональный входному воздействию в соответствии со значением коэффициента передачи объекта.

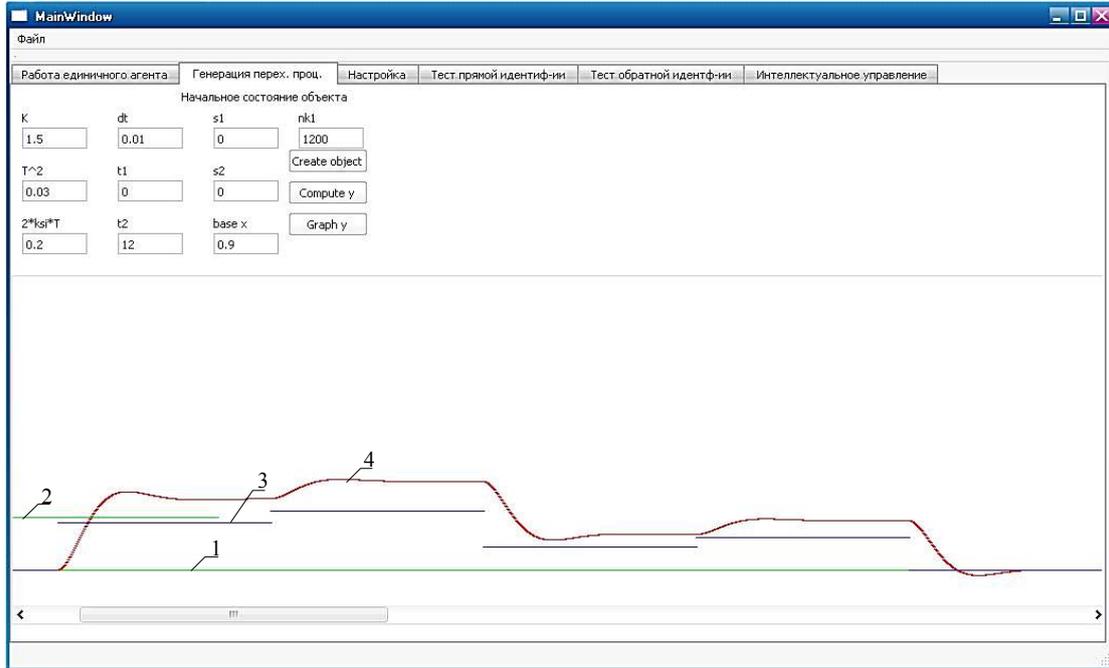


Рис. 6. Создание динамического звена второго порядка и моделирование серии переходных процессов (1 – ось модельного времени; 2 – отметка значения в одну относительную единицу; 3 – входные воздействия; 4 – реакция динамического звена)

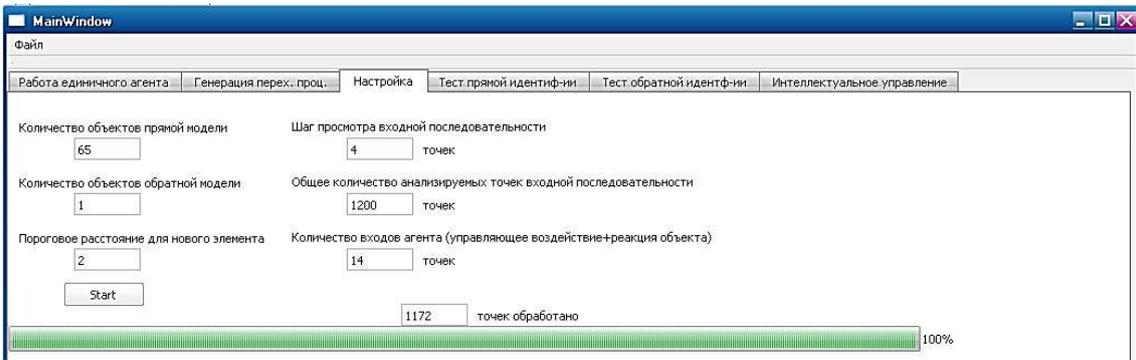


Рис. 7. Однократный просмотр точек переходного процесса с заданной дискретностью, настройка мультиагентной системы

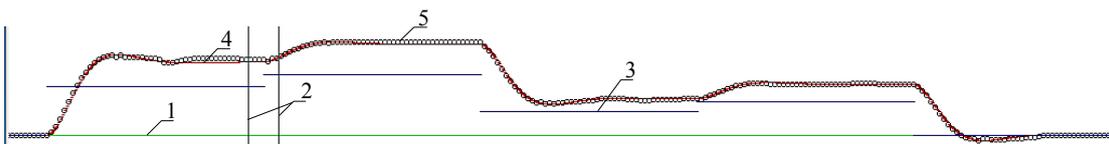


Рис. 8. Графическое отображение выхода мультиагентной системы, наложенного на обучающие переходные процессы (1 – ось модельного времени; 2 – границы окна; с выводом результатов в текстовой форме; 3 – входные воздействия; 4 – реакция динамического звена; 5 – реакция мультиагентной системы)

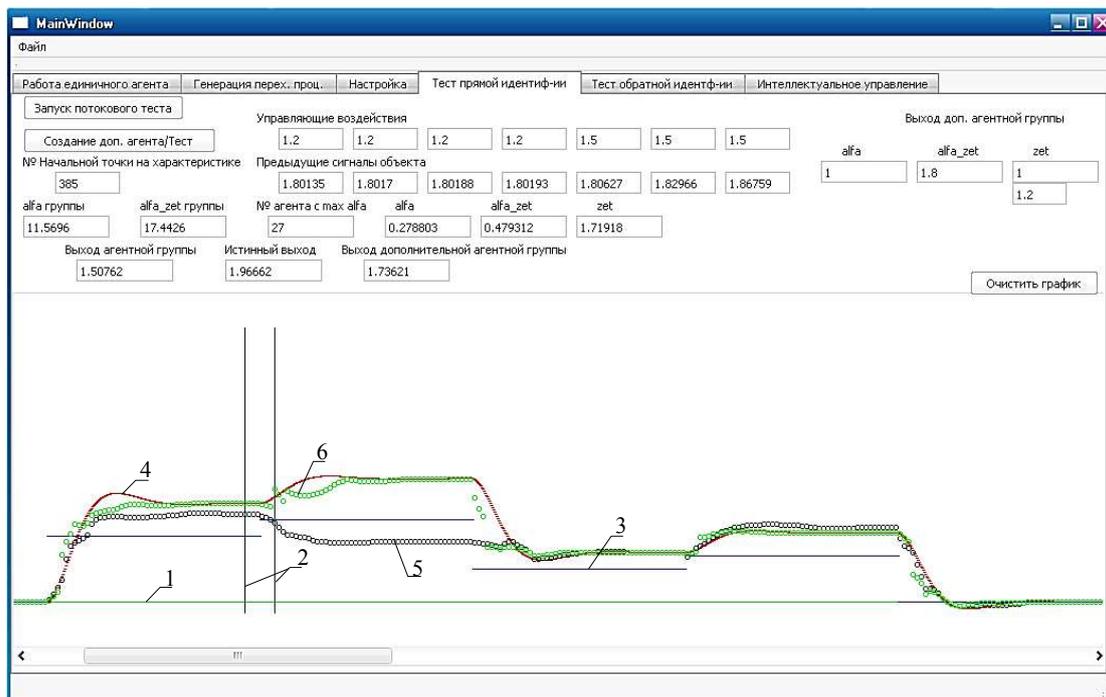


Рис. 9. Результаты тестирования работы мультиагентной системы на переходных процессах, не входящих в обучающую выборку (1 – ось модельного времени; 2 – границы окна; с выводом результатов в текстовой форме; 3 – входные воздействия; 4 – реакция динамического звена; 5 – реакция мультиагентной системы; 6 – реакция мультиагентной системы с дополнительным агентом, содержащим знание о коэффициенте передачи звена)

Агрегация выходов обученной агентной группы и дополнительной, с априорными знаниями о коэффициенте передачи звена, позволила сформировать сигнал (кривая 6 рис. 9), который более адекватно аппроксимировал выход тестового объекта, сводя ошибку в установленном режиме к нулю.

6. Выводы

По результатам проведенных исследований можно утверждать:

- разработанная модель универсального агента позволяет настроиться и идентифицировать определенный образ входного сигнального вектора, определить степень соответствия текущего вектора запомненному и вывести предустановленную константу;
- набор агентов эквивалентен нечеткой экспертной системе или искусственной нейронной сети и, таким образом, может выступать в качестве универсального аппроксиматора;
- введение обратных связей и использование звеньев чистого запаздывания позволит придать агентной системе динамические свойства и быть включенной в контур автоматического управления в качестве регулятора или корректирующего устройства;
- структура супервизорного класса и реализованные методы обеспечивают автоматическую классификацию, запоминание и воспроизведение сигнальных образов, в том числе и реакцию динамических объектов управления, имеющих высокий порядок или нелинейности в своей структуре;
- с ростом сложности процессов и требований к точности воспроизведения количество агентов в группе повышается, но отсутствует принципиальная невозможность идентификации сложных объектов;
- при вычислении группового выходного сигнала целесообразно установить адаптивное пороговое значение степени соответствия для агентов таким образом, чтобы при расчете большая группа слабоответствующих агентов не влияла на точность результата;
- настроенная агентная группа под управлением класса-супервизора может быть использована для организации адаптивных систем и систем с предсказывающей моделью для повышения качества автоматического управления сложными технологическими объектами;

– разработанный метод настройки мультиагентной системы реализует автоматическое обучение, если встречается незнакомая ситуация в рабочем цикле;

– показана принципиальная возможность введения в самонастраивающуюся систему априорной информации о динамическом объекте, что значительно повышает качество моделирования объекта и работу методов оптимизации управляющих алгоритмов, основанных на мультиагентных системах.

Список литературы: 1. *Васильев С. Н., Жерлов А. К., Федосов Е. А., Федунев Б. Е.* Интеллектуальное управление динамическими системами. М.: Физико-математическая литература, 2000. 352 с. 2. *Швецов А. Н.* Агентно-ориентированные системы: от формальных моделей к промышленным приложениям // Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению “Информационно-телекоммуникационные системы”, 2008. 101 с. [http://www.ict.edu.ru/lib/index.php?id_res=5656]. 3. *Горбань А. Н.* Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей // Сибирский журнал вычислительной математики. 1998. Т. 1, № 1. С. 12-24. 4. *Дьяконов В., Круглов В.* Математические пакеты расширения MATLAB. Специальный справочник. СПб.: Питер, 2001. 480 с. 5. *Леоненков А. В.* Нечеткое моделирование в среде MATLAB и fuzzyTECH. СПб.: Бхв-Петербург, 2003. 736 с.

Поступила в редколлегию 11.05.2014

Конох Игорь Сергеевич, канд. техн. наук, доцент кафедры информационно-управляющих систем КрНУ им. М. Остроградского. Научные интересы: автоматизация, интеллектуальные системы управления. Адрес: Украина, 39600, Кременчуг, ул. Первомайская, 20, тел.: (05366) 30157. E-mail: kis_saue@mail.ru.

УДК 629.391

В.В. БАРАННИК, Д.Э. ДВУХГЛАВОВ, В.В. ТВЕРДОХЛЕБ

МЕТОД ДИНАМИЧЕСКОГО УПРАВЛЕНИЯ БИТОВОЙ СКОРОСТЬЮ ВИДЕОПОТОКА С ИСПОЛЬЗОВАНИЕМ ТРЕХМЕРНОГО ПРЕДСТАВЛЕНИЯ ТРАНСФОРМАНТ

Рассматривается актуальность исследований, направленных на уменьшение битовой скорости обработки кадров видеопотока для повышения качества представления услуг видеосервиса для энергоэффективных инфокоммуникационных систем. Предлагается представление блоков видеокadra в виде битового куба и принципы организации управления передачей кадра для такого представления, которые изложены в виде готовых для реализации алгоритмов обработки кадра на передающей и принимающей стороне. Также рассматривается подход для выбора оптимальных параметров передачи блоков на основе принципов динамического программирования.

Введение

В современных условиях развития общества широкий сегмент потребителей услуг видеосервиса составляют пользователи мобильных устройств. Основной задачей при предоставлении услуг данной категории пользователей является управление интенсивностью видеотрафика в телекоммуникационных системах для повышения качества таких услуг. При этом необходимо учитывать затраты на обработку видеопотока, особенно для мобильных беспроводных телекоммуникационных систем.

Для решения указанной задачи необходимо снижение времени передачи данных путем использования методов разгрузки сетевого трафика на уровне конечных узлов и узлов сети, к которым относят маршрутизацию данных и использование резервных каналов. Однако в ряде работ показано, что более актуальным является управление при контроле уровня кодирования источника. На сегодняшний день предложены некоторые подходы к управлению параметрами обработки для различных вариантов представления кадров видеопотока [1-3]. При этом задача управления предполагает выбор параметров компрессии, обеспечивающих требуемое соотношение двух характеристик видеопотока: битовую скорость и качество визуального восприятия. Битовая скорость r характеризует интенсивность видеопотока сжатых кадров $I_{сж}$ и представляет собой объем сжатых видеоданных,

– разработанный метод настройки мультиагентной системы реализует автоматическое обучение, если встречается незнакомая ситуация в рабочем цикле;

– показана принципиальная возможность введения в самонастраивающуюся систему априорной информации о динамическом объекте, что значительно повышает качество моделирования объекта и работу методов оптимизации управляющих алгоритмов, основанных на мультиагентных системах.

Список литературы: 1. *Васильев С. Н., Жерлов А. К., Федосов Е. А., Федунев Б. Е.* Интеллектуальное управление динамическими системами. М.: Физико-математическая литература, 2000. 352 с. 2. *Швецов А. Н.* Агентно-ориентированные системы: от формальных моделей к промышленным приложениям // Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению “Информационно-телекоммуникационные системы”, 2008. 101 с. [http://www.ict.edu.ru/lib/index.php?id_res=5656]. 3. *Горбань А. Н.* Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей // Сибирский журнал вычислительной математики. 1998. Т. 1, № 1. С. 12-24. 4. *Дьяконов В., Круглов В.* Математические пакеты расширения MATLAB. Специальный справочник. СПб.: Питер, 2001. 480 с. 5. *Леоненков А. В.* Нечеткое моделирование в среде MATLAB и fuzzyTECH. СПб.: Бхв-Петербург, 2003. 736 с.

Поступила в редколлегию 11.05.2014

Конох Игорь Сергеевич, канд. техн. наук, доцент кафедры информационно-управляющих систем КрНУ им. М. Остроградского. Научные интересы: автоматизация, интеллектуальные системы управления. Адрес: Украина, 39600, Кременчуг, ул. Первомайская, 20, тел.: (05366) 30157. E-mail: kis_saue@mail.ru.

УДК 629.391

В.В. БАРАННИК, Д.Э. ДВУХГЛАВОВ, В.В. ТВЕРДОХЛЕБ

МЕТОД ДИНАМИЧЕСКОГО УПРАВЛЕНИЯ БИТОВОЙ СКОРОСТЬЮ ВИДЕОПОТОКА С ИСПОЛЬЗОВАНИЕМ ТРЕХМЕРНОГО ПРЕДСТАВЛЕНИЯ ТРАНСФОРМАНТ

Рассматривается актуальность исследований, направленных на уменьшение битовой скорости обработки кадров видеопотока для повышения качества представления услуг видеосервиса для энергоэффективных инфокоммуникационных систем. Предлагается представление блоков видеокadra в виде битового куба и принципы организации управления передачей кадра для такого представления, которые изложены в виде готовых для реализации алгоритмов обработки кадра на передающей и принимающей стороне. Также рассматривается подход для выбора оптимальных параметров передачи блоков на основе принципов динамического программирования.

Введение

В современных условиях развития общества широкий сегмент потребителей услуг видеосервиса составляют пользователи мобильных устройств. Основной задачей при предоставлении услуг данной категории пользователей является управление интенсивностью видеотрафика в телекоммуникационных системах для повышения качества таких услуг. При этом необходимо учитывать затраты на обработку видеопотока, особенно для мобильных беспроводных телекоммуникационных систем.

Для решения указанной задачи необходимо снижение времени передачи данных путем использования методов разгрузки сетевого трафика на уровне конечных узлов и узлов сети, к которым относят маршрутизацию данных и использование резервных каналов. Однако в ряде работ показано, что более актуальным является управление при контроле уровня кодирования источника. На сегодняшний день предложены некоторые подходы к управлению параметрами обработки для различных вариантов представления кадров видеопотока [1-3]. При этом задача управления предполагает выбор параметров компрессии, обеспечивающих требуемое соотношение двух характеристик видеопотока: битовую скорость и качество визуального восприятия. Битовая скорость r характеризует интенсивность видеопотока сжатых кадров $I_{сж}$ и представляет собой объем сжатых видеоданных,

которые требуется передать в единицу времени. Качество визуального восприятия данных видеокadra (блока), относительно исходного, выражается среднеквадратическим отклонением (СКО), которое обозначают d .

Рассмотрим суть данных величин для кадра видеопотока. При обработке изображений кадр является совокупностью из K_W блоков по ширине и K_H блоков по высоте. Общее число блоков обозначим K . Очевидно, что $K = K_W \cdot K_H$. При этом для кадра, состоящего из блоков с битовыми скоростями r_1, r_2, \dots, r_K , справедливо Вклад, вносимый сжатым кадром в общую интенсивность видеопотока, равен:

$$I_{сж} = \sum_{i=1}^K r_i. \quad (1)$$

Для вычисления СКО используется следующее выражение:

$$d = \frac{1}{H \cdot W} \sum_{x=1}^H \sum_{y=1}^W (S(x,y) - S^*(x,y))^2, \quad (2)$$

где H и W – высота и ширина в пикселях одного кадра соответственно; $S(x,y)$ и $S^*(x,y)$ – значения яркости исходного и восстановленного пикселей.

При управлении компрессией следует учитывать, что битовая скорость ограничена пропускной способностью канала телекоммуникационной сети B_W :

$$r_1, r_2, \dots, r_K \leq B_W. \quad (3)$$

Кроме того, управление битовой скоростью предполагает, что блоки, образующие кадр, кодируются с переменной битовой скоростью [2]:

$$r_1 \neq r_2 \neq \dots \neq r_K. \quad (4)$$

Также уровень СКО передаваемого потока является величиной, обратно пропорциональной интенсивности потока сжатых видеокadров, т.е: $d \sim 1/r$.

Данные принципы лежат в основе способов обработки видеопотоков, позволяющих осуществлять управление параметрами обработки видеокadров. Однако существующие алгоритмы управления не обеспечивают необходимую эффективность управления, в частности, не учитывают затраты на обработку, не всегда адекватно оценивается зависимость между битовой скоростью и СКО, а также используются ограниченные механизмы и технологии компрессии для управления битовой скоростью

Целью исследования является разработка метода динамического управления битовой скоростью видеокadров, позволяющего учитывать ограниченность ресурса на обработку видеопотока.

Основная часть

Важным моментом при разработке метода управления является представление видеокadров и анализ особенностей обработки изображения. При этом предлагается использовать представление блоков видеокadra в трехмерном пространстве.

Получение такого представления начинается с преобразования блоков видеокadra из модели RGB в модель $YCbCr$, после чего к ним применяется DC-преобразование. В результате для отдельного блока будет получена матрица размером 8×8 , которую будем обозначать Y , $Y = \| Y_{ij} \|$, где Y_{ij} - значение элемента (i, j) трансформанты, полученной после DC-преобразования. Значение каждого элемента лежит в диапазоне $[0; 255]$.

Затем каждый элемент трансформанты представляется в виде столбца битов, отвечающих представлению соответствующего элемента в двоичном виде (рис.1). Фактически выполняется следующее преобразование:

$$\| Y_{ij} \| \rightarrow \| \langle Y_{ij7}, Y_{ij6}, Y_{ij5}, Y_{ij4}, Y_{ij3}, Y_{ij2}, Y_{ij1}, Y_{ij0} \rangle^T \|, \quad (5)$$

$$y_{ijm} \in \{0, 1\}, \quad i = \overline{0, 7}; \quad j = \overline{0, 7}; \quad m = \overline{7, 0},$$

где Y_{ijm} – m -й бит двоичного разложения элемента (i, j) или элемент (i, j) m -го слоя битового куба.

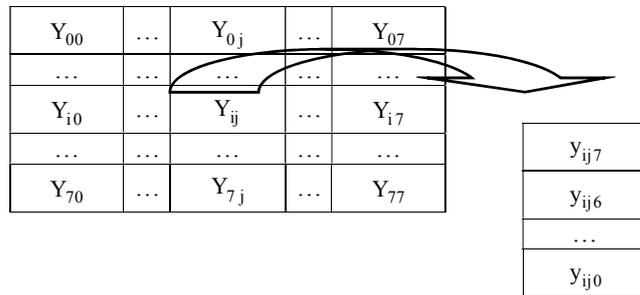


Рис. 1. Преобразование элементов матрицы, полученной после DC-преобразования, к двоичному виду

Совокупность двоичных представлений всех элементов матрицы Y составит битовый куб, пример которого представлен на рис.2. При этом верхний слой данного куба образуют старшие биты двоичного представления. Принятое расположение слоев битового куба предложено с учетом способа организации передачи блоков видеокадра, рассматриваемого ниже.

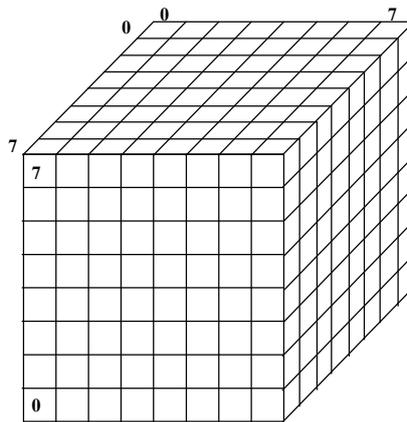


Рис. 2. Битовый куб

Представление трансформанты в трехмерном пространстве позволяет организовать послынную передачу данных. Чтобы определить слои, которые необходимо передать, предлагается задать для каждой k -й трансформанты вектор передачи V_k , который соответствует двоичному представлению чисел от 0 до $p-1$:

$$V_k = \langle v_{k7}, v_{k6}, v_{k5}, v_{k4}, v_{k3}, v_{k2}, v_{k1}, v_{k0} \rangle. \quad (6)$$

Например, если для передачи k -й трансформанты используется вариант передачи 237, то вектор передачи будет следующим: $V_k = \langle 1, 1, 1, 0, 1, 1, 0, 0 \rangle$.

Количество возможных вариантов передачи слоев битового куба P трансформанты будет равно 256 (2 в степени 8). Каждому варианту передаваемых слоев будет соответствовать изображение, отвечающее исходному, но различающееся степенью искажения относительно исходного изображения.

Таким образом, после того, как вектор передачи задан, можно сразу из k -й трансформанты формировать передаваемую последовательность S_{kp} при заданном векторе передачи p :

$$S_{kp} = v_{k7} \& \dots \& v_{km} \& \dots \& v_{k0} \& (y_{ijm} | v_{km} = 1). \quad (7)$$

Каждый элемент представленной последовательности занимает 1 бит. В начало последовательности размещаются биты вектора передачи V_k , совокупность которых определяет номера передаваемых слоев битового куба. Если m -й бит вектора передачи равен 1 (элемент v_{km}), то все 64 элемента y_{ijm} слоя m . Если соответствующий бит вектора передачи равен 0, то значения бит данного слоя битового куба не передаются. Алгоритм формирования последовательности для передачи k -й трансформанты при заданном векторе передачи p представлен ниже.

Алгоритм формирования сообщения из элементов битового куба

Исходные данные: битовый куб Y , вектор передачи V_k .

Цель: сформировать последовательность для передачи трансформанты S_{kp} .

Шаги:

1. Обнулить строку S_{kp}

2. Добавить байт передачи:

```
for (int m=7;m<0;m--)
```

$$S_{kp} = S_{kp} \& v_{km}$$

3. Добавить слои битового куба в соответствии с байтом передачи V_k :

```
for (int i=0;i<8;i++)
```

```
  for (int j=0;j<8;j++)
```

```
    for (int m=7;m<0;m--)
```

```
      if (vkm = 1)
```

$$S_{kp} = S_{kp} \& y_{ijm}$$

Определим величину $K1(V_k)$, которая будет показывать количество единиц в векторе передачи. Она позволяет установить размер передаваемой последовательности и соответственно битовую скорость сжатия k -й трансформанты:

$$r_k(V_k) = 8 + 64 \cdot K1(V_k) \text{ [бит]} \text{ или } r_k(V_k) = 1 + 8 \cdot K1(V_k) \text{ [байт]}.$$

Таким образом, длина кодовой комбинации будет составлять от 72 до 520 бит (от 9 до 65 байт).

На приемной стороне выполняется операция сборки битового куба из последовательности S_{kp} . Суть данной операции заключается в том, что вначале извлекается вектор передачи (его образуют первые 8 бит принятой последовательности), который затем анализируется следующим образом: если бит вектора передачи равен 1, то значениями следующих 64 бит заполняются ячейки соответствующего слоя битового куба; если элемент равен 0, то в куб добавляется слой из 0 элементов.

Алгоритм сборки битового куба на приемной стороне из полученной последовательности

Исходные данные: S_{kp} .

Цель: восстановить битовый куб $Y^{(np)}$ трансформанты из принятой последовательности.

Шаги:

1. Обнулить битовый куб:

```
for (int i=0;i<8;i++)
```

```
  for (int j=0;j<8;j++)
```

```
    for (int m=7;m<0;m--)
```

$$y_{ijm}^{(np)} = 0$$

2. Выделить из S_{kp} вектор передачи $V_k = \langle v_{km} \rangle, m = \overline{7,0}$:

```
for (int p=0;p<8;p++)
{
    m = 7 - p
    vkm = Skp [p]
}
```

3. Заполнить элементы битового куба для слоев, которые соответствуют ненулевым битам в векторе передачи V_k :

```
for (int m=0;m<8;m++)
{
    if vkm = 1
    {
        for (int i=0;i<8;i++)
            for (int j=0;j<8;j++)
            {
                yijm(np) = Skp [p]
                p=p+1
            }
    }
    else
        for (int i=0;i<8;i++)
            for (int j=0;j<8;j++)
                yijm(np) = 0
}
```

Данный подход к передаче трансформанты кадра изображения позволяет реализовать поиск оптимальной стратегии управления для кадра в целом. В качестве основы такого алгоритма предлагается использовать динамическое программирование основу которого составляют уравнения Беллмана.

Для применения принципа Беллмана необходимо соблюдение двух условий [4]:

1. «Отсутствие последствия», которое определяет, что состояние системы не зависит от принятых решений на предыдущих шагах.

2. Целевая функция должна быть аддитивной.

В рассматриваемом случае есть K точек принятия решения по количеству трансформант кадра. Для каждой трансформанты должен быть выбран свой вектор передачи V_{kp} , который и соответствует принятию решения на данном этапе. Формально сформулировать задачу можно так: найти комбинацию векторов передачи для трансформант видеокadra в трехмерном представлении, которая обеспечит минимальную битовую скорость

$$\sum_{k=1}^K r_k \longrightarrow \min \quad (8)$$

при заданных пропускной способности канала B_w

$$\sum_{k=1}^K r_k \leq B_w \quad (9)$$

и качестве восприятия принятого изображения

$$\sum_{k=1}^K d_k \leq d_{\max} . \quad (10)$$

Каждое решение представляет собой стратегию передачи кадра U , которую образует множество частных стратегий, определенных для каждой трансформанты, т.е.

$U = \{U_k\}, k = \overline{1, K}$. Каждой частной стратегии U_k соответствует битовая скорость $r_k = r(U_k)$, которая определяет количество бит в передаваемой последовательности, сформированной для трансформанты k . Наилучшей будет стратегия $U^* = \{U_1^*, U_2^*, \dots, U_k^*, \dots, U_K^*\}$, которой соответствует минимальная битовая скорость,

Также оценивается качество восприятия переданной трансформанты $d_k = d(U_k)$. Данный показатель может быть вычислен до отправки последовательности в канал и может использоваться для оценки выбранной стратегии с точки зрения качества сжатия. Для расчетов достаточно оценки в предположении, что канал передачи “идеальный”, т.е. передает информацию без искажений. Ограничение по пропускной способности канала V_w задается техническими характеристиками каналов связи. Ограничение на суммарную ошибку СКО задается исходя из необходимого уровня качества визуального восприятия видеопотока.

В рассматриваемой задаче следует принять во внимание, что стратегия выбирается как значение вектора передачи p , которое лежит в пределах от 0 до 255. Кроме того, как таковой функции переходов из состояния в состояние нет, но есть взаимосвязанность выбранных частных стратегий на уровне ограничений. Таким образом, стратегия, обеспечивающая наименьшую битовую скорость, может быть определена так:

$$U^* = \{V_1^*, V_2^*, \dots, V_k^*, \dots, V_K^*\}. \quad (11)$$

Согласно принципу оптимальности Беллмана, для каждой k -й трансформанты вектор передачи выбирается таким образом, чтобы вместе с векторами передачи рассмотренных трансформант (т.е. трансформант от $k+1$ до K) формировалась бы последовательность для передачи минимальной длины при выполнении заданных ограничений относительно r и d в соответствии с условиями (9) и (10).

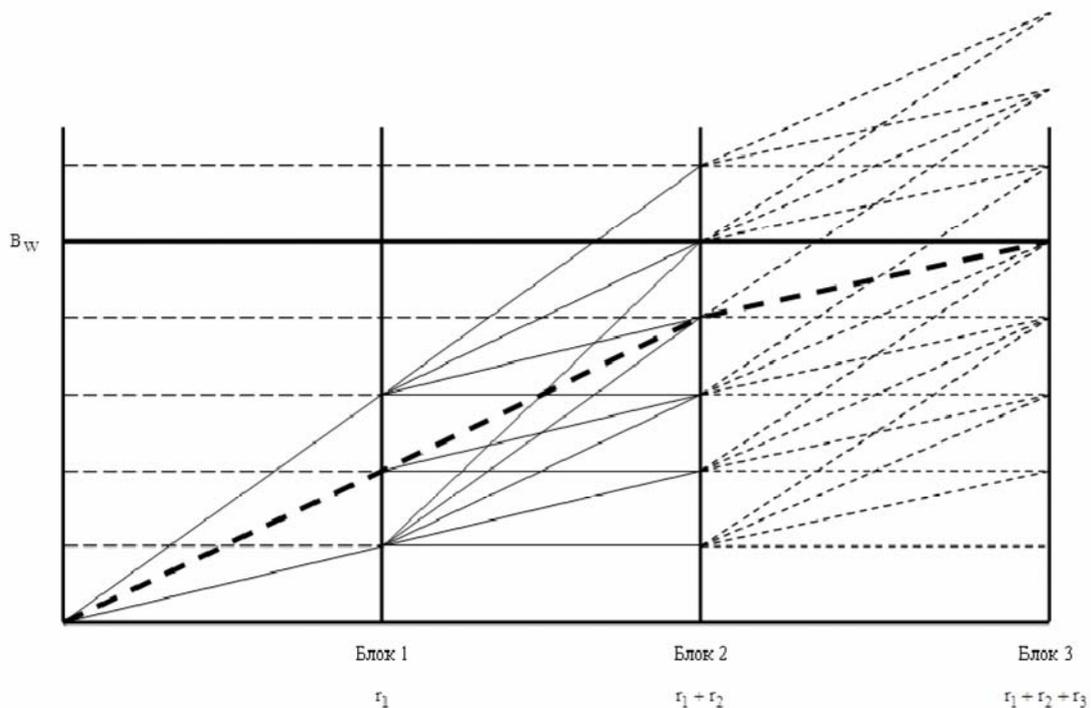


Рис. 3. Схема динамического программирования для 3 блоков

Применение схемы Беллмана предполагает решение задачи в два прохода. При первом проходе от шага K до первого определяется значение оптимальной битовой скорости.

Второй проход позволяет определить оптимальную стратегию управления в виде значений векторов передачи для каждой трансформанты.

Множество допустимых стратегий для каждого шага формируется с учетом следующих требований:

1. Исключение из рассмотрения стратегий, приводящих к переполнению буфера (нарушение условия (9)).

2. Исключение из рассмотрения стратегий, для которых суммарная ошибка превышает заданную (нарушение условия (10)).

3. Стратегии с одинаковым значением d_{\min} исключаются из рассмотрения, за исключением таких, для которых $g = g_{\min}$.

4. Стратегии с одинаковым значением g исключаются из рассмотрения, за исключением таких, для которых $d = d_{\min}$.

На рис. 3 представлена схема работы алгоритма при определении параметров компрессии для кадра, состоящая из 3 трансформант, каждая из которых представлена 3 битовыми плоскостями. Как видно из рис.3, метод позволяет уменьшать пространство поиска за счет учета объективно обусловленных ограничений (они находятся ниже уровня V_W).

Выводы

Разработанный метод динамического управления битовой скоростью видеопотока базируется на основе предложенного трехмерного представления трансформант видеокadra. Научная новизна метода заключается в обеспечении возможности снижения времени на обработку видеокadra за счет управления передачей слоев битового куба без дополнительных преобразований, а также в использовании принципов динамического программирования для определения параметров обработки трансформант, обеспечивающих минимальную битовую скорость для видеокadra в целом.

Практическая значимость метода заключается в возможности его использования для разработки программного обеспечения процессов обработки изображений в энергоэффективных инфокоммуникационных системах.

Список литературы: 1. *Беляев Е.А., Тюрликов А.М.* Управление скоростью и ошибкой кодирования в системе сжатия и передачи видеoinформации с ограничениями на память передающего и принимающего устройств // Компьютерная оптика. 2007. Т.31, №2. С.69-76. 2. *Двухглавов Д.Э., Оксюк А.Г., Твердохлеб В.В.* Анализ подходов к управлению скоростью передачи видеопотока // Сучасна спеціальна техніка: науково-практичний журнал. 2014. №2. С.29-35. 3. *Двухглавов Д.Э., Харченко Н.А.* Методы контроля битовой скорости при компрессии видеоданных // Системи обробки інформації. Х.: ХУПС. 2014. Вип. 9 (125). С. 140 – 145. 4. *Зайченко Ю.И.* Исследование операций. К.: Слово, 2003.

Поступила в редколлегию 13.06.2014

Баранник Владимир Викторович, д-р техн. наук, профессор, начальник кафедры боевого применения и эксплуатации АСУ Харьковского университета Воздушных Сил. Научные интересы: обработка и передача информации. Адрес: Украина, 61023, Харьков, ул. Сумская, 77/79.

Двухглавов Дмитрий Эдуардович, канд. техн. наук, доцент кафедры автоматизированных систем управления Национального технического университета “Харьковский политехнический институт”. Научные интересы: обработка и передача информации. Адрес: Украина, 61002, Харьков, ул. Фрунзе, 21.

Твердохлеб Виталий Викторович, соискатель ХНУРЭ. Научные интересы: обработка и передача информации. Адрес: Украина, 61166, Харьков, ул. Ленина, 14.

РЕШЕНИЕ ДЕТЕРМИНИРОВАННОЙ ЗАДАЧИ ОПТИМИЗАЦИИ ТРЕХУРОВНЕВОЙ СЕТИ ПОСТАВОК ОДНОГО ТОВАРА

Предлагается и исследуется метод оптимизации трехуровневой децентрализованной сети поставок товара в предположении, что функции, описывающие детерминированное поведение основных субъектов рынка, являются полиномами второго порядка. При таком достаточно слабом предположении процесс оптимизации сводится к определению седловой точки функции Лагранжа рассматриваемой задачи. Решаются тестовые задачи, известные из научной литературы.

1. Введение

Рассмотрим задачу оптимизации сети поставок некоторого товара или услуги на конкурентном рынке [1-3] одного товара. Сеть поставок – это модель поведения и взаимодействия субъектов рынка в общем случае трех типов – производителей, крупных дистрибуторов и розничных торговцев. Поведение субъектов рынка описывается такими основными функциями: стоимость производства товара, транзакционные издержки, спрос. В предположении, что сеть поставок является децентрализованной [3], каждый из субъектов рынка стремится к максимизации своей прибыли, конкурируя или вступая в кооперацию с другими субъектами рынка. Достижение этих целей подразумевает определение состояния равновесия данного рынка товаров.

Указанная задача представляет интерес как с теоретической, так и с практической точки зрения в качестве основы осуществления прогноза реакции рынка на увеличение объемов поставок продукции, коррекцию ценовой политики, появление новых субъектов рынка и, следовательно, принятия инвестиционных решений. Кроме того, необходимость в решении подобной задачи возникает на этапе не только стратегического, но и оперативно-го планирования.

Поэтому разработка эффективных методов решения задачи оптимизации сети поставок для основных типов функций является актуальной задачей для поддержки принятия инвестиционного решения на рынке товаров, находящихся в различных фазах своего жизненного цикла.

2. Анализ предыдущих исследований

В настоящий момент существует обширная научная литература по различным аспектам моделирования и решения задачи оптимизации сети поставок. С 1965 года под эгидой ISM (Institute for Supply Management) выпускается ведущий профессиональный журнал в этой области – International Journal of Supply Chain Management (<http://www.emeraldinsight.com/toc/scm/20/1>). Среди множества работ, опубликованных в этом и других научных изданиях, можно выделить статьи, посвященные представлению и решению задачи имитационного моделирования [1,4].

Однако основное внимание исследователей сосредоточено на подходах, развивающих оптимизационные методы решения задач о сетях поставок. Прежде всего, выделим множество работ школы А. Nagurney [3,5,6], в которых предложены как детерминированные, так и вероятностные оптимизационные модели 2-х и 3-уровневых сетей поставок с различными характеристиками спроса.

В этих и других [7,8] работах необходимые условия оптимальности решения сформулированы в виде конечномерной задачи решения вариационного неравенства [6, 9].

Отметим, что решение вариационного неравенства – задача, требующая декомпозиции и применения ряда численных методов оптимизации и удовлетворения строгим условиям для обеспечения сходимости процесса решения.

Целью данной работы является построение и реализация эффективного метода решения задачи оптимизации децентрализованной сети поставок в предположениях о квадратичном (линейном) характере функций себестоимости, транзакционных издержек и спроса.

3. Изложение основного материала

Математическая модель задачи. Следуя [7], составим математическую модель задачи оптимизации децентрализованной трехуровневой сети поставок (рис. 1). Вершины сети представляют три множества агентов рынка: производителей (множество S_1), каждый из которых выпускает товар в количестве q_i , дистрибьюторов (множество S_2), ритейлеров – розничных торговцев (множество S_3). Дугами показано движение товара по сети в количествах q_{ij} , q_{jk} , $i \in S_1$, $|S_1|=I$, $j \in S_2$, $S_2=\{I+1, I+2, \dots, I+J\}$, $|S_2|=J$, $k \in S_3$, $S_3=\{I+J+1, \dots, I+J+K\}$, $|S_3|=K$.

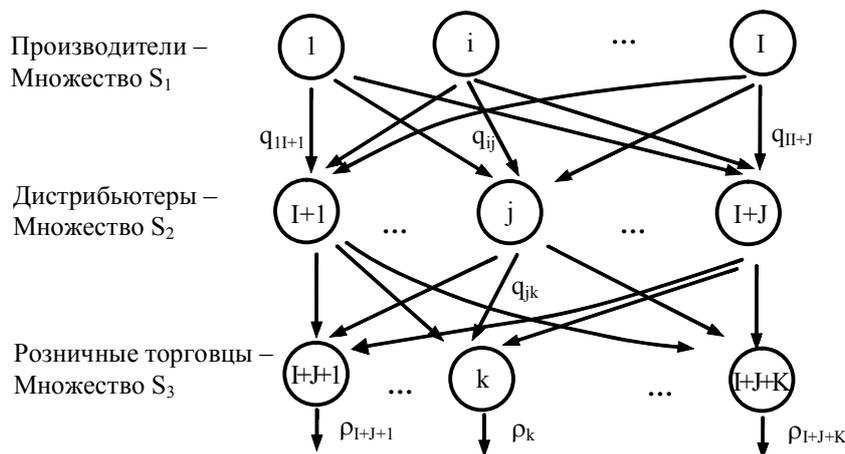


Рис. 1. Графическая модель трехуровневой сети поставок одного товара

В рассмотрение также введены векторы

$$Q_1 = (q_{1(I+1)}, \dots, q_{ij}, \dots, q_{I(I+J)}), Q_2 = (q_{(I+1)(I+J+1)}, \dots, q_{jk}, \dots, q_{(I+J)(I+J+K)}).$$

Положим, что затраты на производство составляют $f_i(Q_1)$, $i \in S_1$, транзакционные затраты $c_{ij}(q_{ij})$, $c_{jk}(q_{jk})$ есть функции от q_{ij} , q_{jk} соответственно, функции операционных затрат имеют вид $c_j(Q_1)$, $c_k(Q_2)$, $j \in S_2$, $k \in S_3$.

Свойство. Все субъекты децентрализованной сети поставок конкурируют таким образом, что каждый из них, не договариваясь с остальными, пытается максимизировать свою прибыль.

Тогда задача максимизации прибыли i -го производителя имеет вид:

$$\sum_{j \in S_2} \rho_{ij}^* q_{ij} - [f_i(Q_1) + \sum_{j \in S_2} c_{ij}(q_{ij})] \rightarrow \max, \quad (1)$$

$$\text{при условии } q_{ij} \geq 0, \quad (2)$$

причем цена ρ_{ij}^* единицы товара, передаваемой i -м производителем j -му дистрибьютору, является равновесной, т.е. соответствует состоянию равновесия всей сети поставок.

Следовательно, цена ρ_{ij}^* – экзогенная величина для производителя i .

Далее, введем в рассмотрение задачу максимизации прибыли j -го дистрибьютора:

$$\sum_{k \in S_3} \rho_{jk}^* q_{jk} - [c_j(Q_1) + \sum_{i \in S_1} \rho_{ij}^* q_{ij}] \rightarrow \max, \quad (3)$$

при условии
$$\sum_{i \in S_1} q_{ij} = \sum_{k \in S_3} q_{jk}, \quad (4)$$

где цена ρ_{jk}^* единицы товара, передаваемой j -м дистрибьютором k -му ритейлеру, является равновесной, т.е. соответствует состоянию равновесия всей сети поставок. Таким образом, цены $\{\rho_{ij}^*, \rho_{jk}^*\}$ – экзогенные величины для дистрибьютора j .

В свою очередь, задача максимизации прибыли k -го ритейлера имеет вид:

$$\rho_k \sum_{j \in S_2} q_{jk} - \left[\sum_{j \in S_2} c_{jk}(q_{jk}) + c_k(Q_2) + \sum_{j \in S_2} \rho_{jk}^* q_{jk} \right] \rightarrow \max, \quad (5)$$

при условии
$$\sum_{j \in S_2} q_{jk} = d_k(\rho_k), \quad (6)$$

где $d_k(\rho_k)$ – функция спроса k -го ритейлера.

Обычно полагают, что спрос $d_k(\rho_k)$ ограничен сверху некоторой заданной величиной $\bar{d}_k(\rho_k)$.

Запишем необходимые условия оптимальности первого порядка для задач (1)-(2), (3)-(4), (5)-(6). В задаче (1)-(2) ограничение (2) тривиально, поэтому условие максимума функции прибыли i -го производителя имеет вид системы уравнений

$$\frac{\partial}{\partial q_{ij}} \left\{ \sum_{j \in S_2} \rho_{ij}^* q_{ij} - [f_i(Q_1) + \sum_{j \in S_2} c_{ij}(q_{ij})] \right\} = 0, \quad j = I+1, I+2, \dots, I+J, \quad (7)$$

откуда непосредственно следует:

$$\rho_{ij}^* - \left[\frac{\partial f_i(Q_1)}{\partial q_{ij}} + \frac{\partial c_{ij}(q_{ij})}{\partial q_{ij}} \right] = 0 \quad \text{или} \quad \left[\frac{\partial f_i(Q_1)}{\partial q_{ij}} + \frac{\partial c_{ij}(q_{ij})}{\partial q_{ij}} \right] = \rho_{ij}^* \quad \text{в случае, если } q_{ij} > 0. \quad (8)$$

Задача (3),(4) есть задача нелинейной оптимизации с одним ограничением-равенством. Следовательно, необходимые условия оптимальности первого порядка в данном случае подразумевают построение соответствующей функции Лагранжа $L(\gamma_j, q_{ij}, q_{jk})$ вида

$$L(\gamma_j, q_{ij}, q_{jk}) = \rho_k \sum_{j \in S_2} q_{jk} - \left[\sum_{j \in S_2} c_{jk}(q_{jk}) + c_j(Q_2) + \sum_{j \in S_2} \rho_{jk}^* q_{jk} \right] + \gamma_j \left(\sum_{i \in S_1} q_{ij} - \sum_{k \in S_3} q_{jk} \right), \quad (9)$$

где g_j – множитель Лагранжа задачи максимизации прибыли j -го дистрибьютора.

Таким образом, необходимыми условиями существования условного экстремума функции прибыли (3) j -го дистрибьютора являются следующие $(I+K+1)$ уравнения:

$$\frac{\partial L(\gamma_j, q_{ij}, q_{jk})}{\partial q_{ij}} = -\rho_{ij}^* - \frac{\partial c_j}{\partial q_{ij}} + \gamma_j = 0; \quad (10)$$

$$\frac{\partial L(\gamma_j, q_{ij}, q_{jk})}{\partial q_{jk}} = \rho_{jk}^* - \gamma_j = 0; \quad (11)$$

$$\frac{\partial L(\gamma_j, q_{ij}, q_{jk})}{\partial \gamma_j} = \sum_{i \in S_1} q_{ij} - \sum_{k \in S_3} q_{jk} = 0, \quad i=1,2,\dots,I, \quad k=(I+J+1), \dots, (I+J+K). \quad (12)$$

Задача (5,6) максимизации прибыли k -го ритейлера также представляет собой задачу нелинейной оптимизации с одним ограничением-равенством. Соответствующая ей функция Лагранжа $L(\delta_k, q_{jk}, \rho_k)$ с множителем Лагранжа δ_k есть функция вида

$$L(\delta_k, q_{jk}, \rho_k) = \rho_k \sum_{j \in S_2} q_{jk} - \left[\sum_{j \in S_2} c_{jk}(q_{jk}) + c_k(Q_2) + \sum_{j \in S_2} \rho_{jk}^* q_{jk} \right] + \delta_k \left(\sum_{j \in S_2} q_{jk} - d_k(\rho_k) \right) \quad (13)$$

Необходимые условия максимума функции прибыли k -го ритейлера первого порядка выражаются системой уравнений

$$\frac{\partial L(\delta_k, q_{jk}, \rho_k)}{\partial q_{jk}} = \rho_k - \left(\frac{\partial c_{jk}}{\partial q_{jk}} + \frac{\partial c_k}{\partial q_{jk}} + \rho_{jk}^* \right) - \delta_k = 0, \quad (14)$$

$$\frac{\partial L(\delta_k, q_{jk}, \rho_k)}{\partial \rho_k} = \sum_{j \in S_2} q_{jk} + \delta_k \frac{\partial d_k(\rho_k)}{\partial \rho_k} = 0, \quad (15)$$

$$\frac{\partial L(\delta_k, q_{jk}, \rho_k)}{\partial \delta_k} = \sum_{j \in S_2} q_{jk} - d_k(\rho_k) = 0, \quad j = I+1, I+2, \dots, I+J. \quad (16)$$

Обычным подходом к решению задачи (1)-(6) оптимизации децентрализованной сети поставок является определение условий равновесного состояния (эквilibрума) сети, получаемых из (8), (10),(11), (14)-(16), вида

$$\frac{\partial f_i}{\partial q_{ij}} + \frac{\partial c_{ij}}{\partial q_{ij}} \begin{cases} = \rho_{ij}^*, \text{ if } q_{ij} > 0, \\ > \rho_{ij}^*, \text{ if } q_{ij} = 0; \end{cases} \quad (17)$$

$$\frac{\partial c_j}{\partial q_{ij}} + \rho_{ij}^* \begin{cases} = \gamma_j, \text{ if } q_{ij} > 0, \\ \geq \gamma_j, \text{ if } q_{ij} = 0; \end{cases} \quad \rho_{jk}^* \begin{cases} = \gamma_j, \text{ if } q_{jk} > 0, \\ \leq \gamma_j, \text{ if } q_{jk} = 0; \end{cases} \quad (18)$$

$$\left(\frac{\partial c_{jk}}{\partial q_{jk}} + \frac{\partial c_j}{\partial q_{jk}} + \rho_{jk}^* \right) \begin{cases} = \rho_k - \gamma_k, \text{ if } q_{jk} > 0, \\ \geq \rho_k - \gamma_k, \text{ if } q_{jk} = 0; \end{cases} \quad \sum_{j \in S_2} q_{jk} \begin{cases} = -\gamma_k \frac{\partial d_k(\rho_k)}{\partial \rho_k}, \text{ if } \rho_k > 0, \\ \leq \gamma_k \frac{\partial d_k(\rho_k)}{\partial \rho_k}, \text{ if } \rho_k = 0; \end{cases} \quad (19)$$

построение обобщенных условий равновесия

$$\left(\frac{\partial f_i}{\partial q_{ij}} + \frac{\partial c_{ij}}{\partial q_{ij}} + \frac{\partial c_j}{\partial q_{ij}} + \frac{\partial c_{jk}}{\partial q_{jk}} + \frac{\partial c_j}{\partial q_{jk}} \right) \begin{cases} = \rho_k - \gamma_k, \text{ if } q_p^k > 0, \\ \geq \rho_k - \gamma_k, \text{ if } q_p^k = 0, \end{cases} \quad \forall k \in S_3, p \quad (20)$$

и эквивалентная замена условий (20) вариационным неравенством вида:

$$\sum_{i \in S_1} \sum_{j \in S_2} \left[\frac{\partial (f_i(Q_1^*) + c_{ij}(q_{ij}^*) + c_j(Q_1^*))}{\partial q_{ij}} \right] (q_{ij} - q_{ij}^*) + \sum_{j \in S_2} \sum_{k \in S_3} \left[\frac{\partial (c_{jk}(q_{jk}^*) + c_k(Q_2^*))}{\partial q_{jk}} \right] (q_{jk} - q_{jk}^*) + \sum_{i \in S_1} (\lambda_k^* + \rho_k^*) (d_k(\rho_k) - d_k(\rho_k^*)) - \sum_{k \in S_3} \lambda_k^* \left[\frac{\partial d_k(\rho_k^*)}{\partial \rho_k} + d_k(\rho_k^*) \right] (\rho_k - \rho_k^*) \geq 0, \forall \{q_{ij}, q_{jk}, \rho_k\} \in \Omega. \quad (21)$$

Для решения вариационного неравенства вида (21) предложено множество численных методов: метод проекции, метод диагонализации, их многочисленные модификации и другие методы [6-10].

В данной статье предлагается другой, более простой подход, основанный на следующем предположении о виде функций $f_i(Q_1)$, $c_{ij}(q_{ij})$, $c_{jk}(q_{jk})$, $c_j(Q_1)$, $c_k(Q_2)$, $d_k(\rho_k)$, $i \in S_1, j \in S_2, k \in S_3$.

Предположение. Функции производственных затрат $f_i(Q_1)$, транзакционных затрат $c_{ij}(q_{ij})$, $c_{jk}(q_{jk})$, операционных издержек $c_j(Q_1)$, $c_k(Q_2)$ и спроса $d_k(\rho_k)$, $i \in S_1, j \in S_2, k \in S_3$ являются полиномами не выше второго порядка.

На основе анализа специфики структурной идентификации функций как трендовых моделей по статистическим рядам данных, а также опираясь на концептуальное положение о соответствии уровня точности аналитической модели имеющихся статистических данных, можно сделать обоснованный вывод о достаточности полиномов первого или второго порядка как моделей рассматриваемых характеристик реальной сети поставок.

Принимая во внимание **свойство**, отметим, что условия оптимальности первого порядка представляют собой систему $AQ = 0$ уравнений (8), (10)-(12), (14)-(16), Q – вектор переменных вида

$$Q = (q_{1(I+1)}, \dots, q_{ij}, \dots, q_{|S_1||S_2|}, q_{(I+1)(I+J+1)}, \dots, q_{jk}, \dots, q_{|S_2||S_3|}, \rho_{1(I+1)}^*, \dots, \rho_{|S_1||S_2|}^*, \rho_{(I+1)(I+J+1)}^*, \rho_{|S_1||S_2|}^*, \gamma_{I+1}, \dots, \gamma_{I+J}, \delta_{I+J+1}, \dots, \delta_{I+J+K}).$$

Размерность матрицы коэффициентов A условий оптимальности первого порядка для всей сети поставок определяется величиной

$$N = IJ + J(I + K + 1) + JK + 2K = 2J(I + K) + J + 2K. \quad (22)$$

С другой стороны, общее количество переменных задачи оптимизации децентрализованной сети поставок: $I^*J + J^*K + K$. Общее количество множителей Лагранжа: $J + K$. Соответственно, размерность вектора равновесных цен составляет: $IJ + JK$.

При реализации предлагаемого метода решения задачи оптимизации сети поставок рассмотрим 2 случая.

Случай 1. Функции спроса $d_k(\rho_k)$ линейны. Тогда $A Q = 0$ представляет собой систему линейных уравнений, матрица коэффициентов которой является редкозаполненной. Так, в первой строке матрицы коэффициентов не более $(J+1)$ ненулевых коэффициентов, что значительно меньше N из (22).

Очевидно, систему уравнений $A Q = 0$ можно решить в 3 этапа.

1. Классифицировать равновесные цены ρ_{ij}^* , ρ_{jk}^* как параметры, исключить эти параметры из рассмотрения на 1-м этапе и построить редуцированную систему уравнений $A_r Q_r = 0$, размерность которой $N_r = J(I+K) + J + 2K$.

Это можно сделать, выполнив элементарные арифметические операции сложения над одноименными строками матрицы A вида (8) и (10), а также (11) и (15).

2. Однократным решением редуцированной системы $A_r Q_r = 0$ найти вектор

$$Q_r = (q_{1(I+1)}, \dots, q_{ij}, \dots, q_{|S_1||S_2|}, q_{(I+1)(I+J+1)}, \dots, q_{jk}, \dots, q_{|S_2||S_3|}, \gamma_{I+1}, \dots, \gamma_{I+J}, \delta_{I+J+1}, \dots, \delta_{I+J+K}).$$

3. Определить равновесные цены ρ_{ij}^* , ρ_{jk}^* из ограничений (8), (11).

Случай 2. Монотонно убывающие функции спроса $d_k(\rho_k)$ – полиномы второго порядка. Это означает, что функции $2K$ уравнений, порождаемых условиями (15), (16), редуцированной системы $A_r Q_r = 0$ содержат нелинейные слагаемые вида $\delta_k \frac{\partial d_k(\rho_k)}{\partial \rho_k}$, $d_k(\rho_k)$.

Однако, структура функций ограничений (15), (16) близка к сепарабельной. Используя идею методики линейаризации, предложенной в [11], процесс решения полученной нелинейной системы $A_r Q_r = 0$ можно представить как решение последовательности линейных систем уравнений $A_r^n Q_r^n = 0$, $n=0, 1, \dots$. Для этого на каждом шаге приближения функция спроса $d_k(\rho_k)$ представляется линейной аппроксимацией $\bar{d}_k^n(\rho_k)$, которая строится с использованием заданного ограничения на величину спроса $\bar{d}_k(\rho_k)$.

Рассмотрим суть процесса аппроксимации. Пусть $\varepsilon > 0$ – заданная точность аппроксимации. Начальное приближение ($n=0$) строится следующим образом. Сформулируем естественное ограничение на максимальную величину цены ρ_{k0} при некотором ненулевом спросе d_{k0} и выделим диапазон Δ_0 (рис. 2) возможных значений $\rho_k \in \Delta_0 = [\bar{\rho}_k, \rho_{k0}]$.

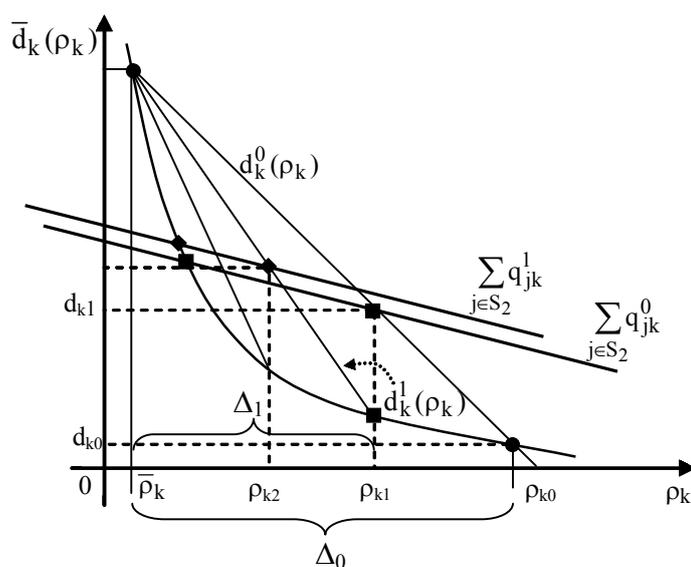


Рис. 2. Иллюстрация процесса приближения к нулю функции условия (16)

Построим уравнение секущей вида $d_k^0(\rho_k) = (\rho_k - \bar{\rho}_k)(d_{k0} - \bar{d}_k) + \bar{d}_k(\rho_{k0} - \bar{\rho}_k)$.

Тогда в качестве аппроксимаций функций ограничений вида (16) принимаются линейные функции

$$\sum_{j \in S_2} q_{jk} - (\rho_k - \bar{\rho}_k)(d_{k0} - \bar{d}_k) + \bar{d}_k(\rho_{k0} - \bar{\rho}_k), \quad k \in S_3, \quad (23)$$

формирующие правые части соответствующих уравнений системы $A_r^0 Q_r^0 = 0$.

Отметим, что функции уравнений, соответствующих условиям (15), также становятся линейными, так как в случае использования приближения (23) слагаемое $\delta_k \frac{\partial d_k(\rho_k)}{\partial \rho_k}$ в

$$(15) \approx \delta_k (d_{k0} - \bar{d}_k).$$

Решение построенной системы уравнений $A_r^0 Q_r^0 = 0$ и его анализ определяют диапазон $D_1 < D_0$ возможных значений, на котором осуществляются построения следующей итерации ($n=1, 2, \dots$).

Процесс продолжается, пока не выполнится условие сходимости $|d_{kn} - d_{k(n-1)}| \leq \varepsilon$, $k \in S_3$.

Пример [7]. Рассмотрим задачу оптимизации трехзвенной сети поставок со следующими параметрами: $S_1 = \{1, 2, 3\}$, $S_2 = \{4, 5\}$, $S_3 = \{5, 6, 7\}$.

Функции $f_i(Q_1)$ стоимости производства имеют вид:

$$f_1(q) = (q_{14} + q_{15})^2 + 3(q_{14} + q_{15}) + 0,5(q_{24} + q_{25})(q_{34} + q_{35}) + 10;$$

$$f_2(q) = (q_{24} + q_{25})^2 + 3(q_{24} + q_{25}) + (q_{14} + q_{15})(q_{34} + q_{35}) + 10;$$

$$f_3(q) = (q_{34} + q_{35})^2 + 3(q_{34} + q_{35}) + (q_{14} + q_{15})(q_{24} + q_{25}) + 10.$$

Функции транзакционных и операционных издержек $c_{ij}(q_{ij})$, $c_{jk}(q_{jk})$, $c_j(q_j)$, $c_k(q_k)$ таковы: $c_{ij}(q_{ij}) = q_{ij}^2 + 2q_{ij}$, $c_{jk}(q_{jk}) = q_{jk}^2 + 0,5q_{jk}$, $c_j(q_j) = 0,5q_j^2 + q_j$, $c_k(q_k) = 0,5$, где $q_j = q_{1j} + q_{2j} + q_{3j}$, $j = \{4, 5\}$, $q_k = q_{4k} + q_{5k}$, $k = \{6, 7, 8\}$.

Функции спроса ритейлеров являются линейными и имеют вид:

$$d_6(p_6) = -3p_6 + 900;$$

$$d_7(p_7) = -3p_7 + 1200;$$

$$d_8(p_8) = -2p_8 + 1000.$$

Очевидно, в нашей терминологии это **случай 1**. Другими словами, для решения задачи достаточно однократного решения системы уравнений $A Q = 0$.

Построим систему уравнений $A Q = 0$. Первые 6 ограничений системы $A Q = 0$, получаемые из (7), (8), имеют вид:

$$4q_{ij} + 2q_{i(S_2/j)} + 5 - \rho_{ij}^* = 0,$$

где выражение S_2/j означает элемент индексного множества S_2 , не равный j .

Далее, условия (10)-(12) принимают вид

$$\sum_{i \in S_1} q_{ij} - (\gamma_j - \rho_{ij}^* - 1) = 0, \quad \rho_{jk}^* - \gamma_j = 0, \quad \sum_{i \in S_1} q_{ij} - \sum_{k \in S_3} q_{jk} = 0.$$

Условие (14) генерирует ограничение $2q_{jk} - \rho_k + \delta_k + \rho_{jk}^* + 1 = 0$.

Условия (15) первого порядка: $q_{46} + q_{56} - 3\delta_6 = 0$, $q_{47} + q_{57} - 3\delta_7 = 0$, $q_{48} + q_{58} - 2\delta_8 = 0$.

Условия (16) выражаются уравнениями:

$$q_{46} + q_{56} + 3p_6 - 900 = 0,$$

$$q_{47} + q_{57} + 3p_7 - 1200 = 0,$$

$$q_{48} + q_{58} + 2p_8 - 1000 = 0.$$

Построенная таким образом матрица A системы уравнений $A Q = 0$ имеет ранг $r_A = 32$, ранг матрицы коэффициентов редуцированной системы $A_r Q_r = 0$: $r_{A_r} = 20$.

В результате решения $A_T Q_T = 0$ и вычисления равновесных цен получен вектор Q с компонентами:

– количество передаваемого товара: $q_{ij}=30,90$; $q_{i6}=4,48$; $q_{i7}=34,48$; $q_{i8}=53,73$;

– конечные цены: $\rho_6 = 297,01$; $\rho_7 = 377,01$; $\rho_8 = 446,27$;

– множители Лагранжа: $\gamma_j = 284,07$; $\delta_6 = 2,99$; $\delta_7 = 22,99$; $\delta_8 = 53,73$;

– равновесные цены: $\rho_{ij}^* = 190,38$, $\rho_{jk}^* = 284,07$, $i = \{1, 2, 3\}$, $j = \{4, 5\}$, $k = \{6, 7, 8\}$,

что полностью совпадает с результатами, полученными в [8] с применением алгоритма, представляющего собой комбинацию метода диагонализации и технологии назначения равновесного трафика пользователя (user-equilibrium traffic assignment technique).

4. Выводы и направления дальнейших исследований.

Предложен эффективный математический аппарат решения задачи оптимизации децентрализованной сети поставок. Показано, что при весьма слабых ограничениях построенная методика обеспечивает оптимальное решение широкого класса практических приложений и может быть использована на этапе как стратегического, так и оперативного (экспресс-анализ) планирования. В дальнейшем предусматривается рассмотрение задачи оптимизации сети поставок при условии выпуска производителем конечного множества различных видов товаров.

Список литературы: 1. *Chatfield D.C.* SCML: An information framework to support supply chain modeling / D.C. Chatfield, T.P. Harrison, J.C. Nayya // *European Journal of Operational Research*, 2009. Vol. 196. P. 651–660. 2. *Черешнев В. В.* Имитационное моделирование конкурентного поведения производителя на потребительском рынке / В. В. Черешнев, Д. Н. Верзилин, Е. С. Зайчик // *Имитационное моделирование. Теория и практика: Международная научно-практическая конференция, 17-19 октября 2007, Спб.* 2007. С. 245-248. 3. *Nagurney A.* Supply Chain Network Equilibrium Model with Random Demands / A. Nagurney, J.Dong, D.Zhang // *European Journal of Operational Research*, 2004. № 156. P. 194-212. 4. *Bousqueta F.* Multi-agent simulations and ecosystem management: a review / F.Bousqueta, C. Le Page // *Ecological Modelling*, 2004. Vol. 176. № 3-4. P. 313-332. 5. *Nagurney A.* Supply chain supernetworks and environmental criteria/ A. Nagurney, F. Toyasaki // *Transportation Research*, 2003. N 8. P. 185–213. 6. *Nagurney A.* Network Economics: A Variational Inequality Approach / A. Nagurney. Dordrecht: Kluwer Academic Publishers, 1999. 316 p. 7. *Hsueh Che-Fu.* Equilibrium analysis and corporate social responsibility for supply chain integration / Che-Fu Hsueh, Mei-Shiang Chang // *European Journal of Operational Research*, 2008. N 190. P. 116–129. 8. *Dong J.* A supply chain network equilibrium model / J. Dong, D. Zhang // *Transportation Research*, 2002. N 38. P. 281–303. 9. *Marcotte P.* Application of Khobotovs algorithm to variational inequalities and network equilibrium problems/ P. Marcotte // *INFOR*, 1991. Vol. 29, No. 4. P. 258-270. 10. *Sheffi Y.* Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods/ Y. Sheffi. – NJ: Prentice-Hall, Englewood Cliffs, 1985. 410 p. 11. *Новожилова М.В.* Оптимизационная задача распределения ограниченных ресурсов проекта с сепарабельными ограничениями / М.В. Новожилова, М.Н. Мурун, И.А. Чуб // *Кибернетика и системный анализ*, 2013. № 4. С. 74-88.

Поступила в редколлегию 23.05.2014

Новожилова Марина Владимировна, д-р физ.-мат. наук, профессор, заведующая кафедрой Харьковского национального университета строительства и архитектуры. Научные интересы: системный анализ, математическое моделирование сложных динамических систем. Адрес: Украина, 61000, Харьков, ул. Сумская, 40, тел.: (057) 706-20-49.

Штань Игорь Владимирович, студент гр. ЭК-41 ХНУСА. Научные интересы: нелинейные оптимизационные задачи математического программирования. Адрес: Украина, 61000, Харьков, ул. Сумская, 40, тел.: (057) 706-20-49.

МЕТОДИКА ОЦІНКИ ЕФЕКТИВНОСТІ ТЕХНОЛОГІЇ СТИСНЕННЯ ЗОБРАЖЕНЬ НА БАЗІ МЕТОДУ ІНВАРІАНТНО-ПРОСТОРОВОГО КОДУВАННЯ ЗА ШВИДКОДІЄЮ

Аналізуються технології обробки та передачі даних в інформаційно-телекомунікаційних системах (ІТКС). Визначаються основні показники ефективності функціонування ІТКС, а також роль і місце алгоритмів стиснення зображень для скорочення часу доставки даних в ІТКС. Пропонується методика оцінки ефективності технології стиснення на базі методу інваріантно-просторового кодування за швидкодією.

1. Актуальність дослідження

Сучасне суспільство не здатне існувати без розгалужених інформаційно-телекомунікаційних систем (ІТКС). Без них важко уявити функціонування всіх найважливіших галузей людського існування – промисловості, господарства, медицини, освіти, транспортної системи.

Інформаційно-телекомунікаційна система – це сукупність автоматизованих систем та мереж і каналів передачі інформації [1-3].

Канал передачі даних (КПД) – організаційно-технічна система, яка складається з комплексів телекомунікаційного обладнання (вузлів комутації) та реалізує технологію інформаційного обміну з використанням первинної мережі зв'язку [1-3]. КПД є сукупністю технічних засобів між джерелом повідомлення та споживачем. Технічні засоби у складі КПД вирішують задачу надійної та швидкісної доставки повідомлення до споживача. З цією метою вони обробляють сигнали за певними алгоритмами.

Найважливішим показником ефективності функціонування КПД є час доставки інформації t_{dost} .

Значення цього показника залежить від багатьох факторів: структури каналу зв'язку, його пропускної здатності, способів з'єднання каналів зв'язку між взаємодіючими абонентами, протоколів інформаційного обміну, методів доступу абонентів до середовища передачі, обсягів даних, що передаються. У загальному випадку час доставки t_{dost} у t_{dost} залежить від часових складових: часу обробки t_{obr} та часу передачі t_{per} :

$$t_{dost} = t_{obr} + t_{per} \quad (1)$$

Однією з найактуальніших є задача зменшення часу доставки даних. Для цього розробляються спеціальні стандарти, протоколи й методи обробки даних, які є складовими інформаційного і програмно-математичного забезпечення ІТКС.

Час обробки даних t_{obr} в ІТКС:

$$t_{obr} = t_{comp} + t_{dec} = \frac{n_{comp} + n_{dec}}{S_{op}}, \quad (2)$$

де n_{com} – число машинних операцій, відведених на процедуру стиснення зображень; n_{dec} – число машинних операцій, відведених на процедуру відновлення зображень; t_{comp} – час, необхідний для стиснення даних; t_{dec} – час, необхідний для відновлення стиснених даних; S_{op} – швидкість виконання машинних операцій, тактова частота мікропроцесора обчислювальної системи.

Як відомо, найбільша частка інформації міститься у відео- та графічних даних. Частка інформаційного потоку на базі графіки займає до 90% від загального потоку інформації, що обробляється в ІТКС. Разом з тим суттєвим недоліком графічних даних є їх значний об'єм, який може змінюватися в широкому діапазоні.

Таким чином, важливою та актуальною науковою задачею є розробка і реалізація нових технологій стиснення, що забезпечують збільшення ступеня компресії при заданому рівні спотворень даних у відновленій інформаційній структурі та часі, необхідному для обробки зображення.

В ході попередніх наукових досліджень [4-8] була запропонована технологія стиснення зображень на базі методу інваріантно-просторового кодування (ІПК). Структурна схема технології наведена на рис. 1.



Рис. 1. Структурна схема технології стиснення на базі методу ІПК

2. Мета дослідження

Метою даної роботи є оцінка ефективності технології стиснення зображень на базі методу інваріантно-просторового кодування з точки зору скорочення часу доставки зображень каналами передачі даних.

Задачі наукового дослідження:

1. Створити методику оцінки ефективності технології стиснення на базі методу ІПК за швидкодією.
2. Створити аналітичну модель розрахунку часу обробки зображень при використанні технології стиснення на базі методу ІПК.
3. Розрахувати часові затрати на обробку тестових зображень кодеком на базі ІПК.
4. Провести порівняльну характеристику запропонованої технології стиснення зображень за швидкодією з кодеками на базі алгоритму JPEG.

Згідно з рис. 1 процедура стиснення зображень складається з таких етапів:

- Етап 1. Представлення матриці зображення у форматі YUV.
- Етап 2. Дискретне косинусне перетворення.
- Етап 3. Квантування.
- Етап 4. Формування матриці знаків.
- Етап 5. Площинне розбиття матриці зображення.
- Етап 6. Розрахунок структурної ознаки – сумарної бітової кореляції.
- Етап 7. Розрахунок інваріантного коду двійкової послідовності.
- Етап 8. Кодування матриць знаків методом RLE.

У табл. 1 наведена кількість цілочисельних та речовинних операцій, необхідних для виконання кожного з наведених етапів для зображення розмірністю L рядків на C стовпців.

Експериментальні розрахунки проводилися на ноутбучі компанії DELL, модель – Inspiron 5748; ОЗП – 8 Гб; ОС – Windows-7, 64-bit; мікропроцесор – Intel Core i7-4510 (архітектура – Nehalem), тактова частота – 3,1 ГГц.

Таблиця 1

Кількість арифметичних операцій, необхідних для реалізації методу стиснення зображень на базі ІПК

	Етап 1	Етап 2	Етап 3	Етап 4	Етап 5
Цілочисельне додавання	-	-	-	-	-
Цілочисельне віднімання	-	-	-	-	-
Цілочисельне ділення	-	-	$3 \times L \times C$	-	$3 \times L \times C$
Цілочисельне порівняння	-	-	-	$3 \times L \times C$	-
Цілочисельне взяття по модулю	-	-	-	-	-
Речовинне додавання	$7 \times L \times C$	$3 \times L \times C$	-	-	-
Речовинне множення	$6 \times L \times C$	$6 \times L \times C$	-	-	-
Операція знаходження косинуса	-	32	-	-	-
	Етап 6	Етап 8	Етап 9	Всього	
Цілочисельне додавання	-	-	$24 \times L \times C$	$80 \times L \times C$	
Цілочисельне віднімання	$24 \times L \times C$	$\frac{33}{8} \times L \times C$	-	$\frac{225}{8} \times L \times C$	
Цілочисельне ділення	-	$\frac{21}{4} \times L \times C$	-	$\frac{45}{4} \times L \times C$	
Цілочисельне порівняння	-	$\frac{15}{8} \times L \times C$	$24 \times L \times C$	$\frac{231}{8} \times L \times C$	
Цілочисельне взяття по модулю	$24 \times L \times C$	$\frac{3}{4} \times L \times C$	-	$\frac{99}{4} \times L \times C$	
Речовинне додавання	-	-	-	$10 \times L \times C$	
Речовинне множення	-	-	-	$12 \times L \times C$	
Операція знаходження косинуса	-	-	-	32	

В табл. 2 наведені значення кількості машинних операцій (тактів), необхідних згідно з специфікацією [9] для реалізації арифметичних процедур, перелічених у табл. 2, цим мікропроцесором.

Таблиця 2

Кількість машинних операцій, необхідних для реалізації арифметичних дій мікропроцесором Core i7-4510

Операція	Кількість тактів	Операція	Кількість тактів
Цілочисельні операції		Речовинні операції	
Додавання	1	Додавання	3
Віднімання	1	Множення	5
Ділення	20	Ділення	25
Порівняння	1	Знаходження значення косинуса	80
Взяття по модулю	1		

Тоді згідно з даними, зведеними у табл. 1 та 2, загальна кількість машинних операцій, необхідних для стиснення зображення розмірністю L рядків на C стовпців, складе:

$$\begin{aligned}
 n_{\text{com}} = & 124 \times L \times C + \frac{225}{8} \times L \times C + 180 \times L \times C + \\
 & + \frac{231}{8} \times L \times C + \frac{99}{4} \times L \times C + 30 \times L \times C + \\
 & + 60 \times L \times C + 2560 = \frac{3806}{8} \times L \times C + 2560.
 \end{aligned} \tag{3}$$

З урахуванням порядку значення тактової частоти $S_{op} = 3,1$ ГГц ($2,4 \cdot 10^9$ оп/с) доданок 2560 можна не брати до уваги, наприклад, на стиснення зображення розмірністю 1280·1280 буде витрачено 28,63 мс.

Для проведення подальшого порівняння запропонованого методу стиснення за швидкістю із існуючими методами, оцінимо час, необхідний для стиснення тестових зображень методом на базі ІПК. В табл. 3 наведений перелік тестових зображень із зазначенням їх розміру та часу стиснення методом на базі ІПК з урахуванням формул (2) та (3).

Таблиця 3

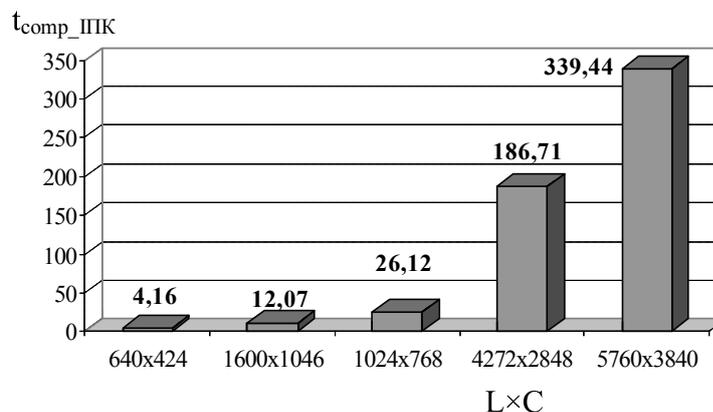
Час стиснення тестових зображень методом на базі ІПК

Тестове зображення	Розмірність зображення, $L \times C$, пікселі	Час стиснення методом на базі ІПК, t_{comp_ISC} , мс
test1_1.bmp	640×424	4,16
test2_1.bmp	1024×768	12,07
test3_1.bmp	1600×1064	26,12
test4_1.bmp	4272×2848	186,71
test5_1.bmp	5760×3840	339,44

На рис. 2 наведені діаграми залежності часу стиснення зображень методом на базі ІПК від розміру матриці зображення.

Рис. 2. Діаграми залежності величини часу стиснення стиснення t_{comp_ISC} від розмірності зображень для розробленого методу на базі ІПК

Щоб порівняти отримані результати для розробленого методу з швидкістю існуючих алгоритмів стиснення зображень, розглянемо бенчмарки швидкості стиснення за алгоритмом JPEG. Для отримання результатів оцінки часу стиснення використаємо один з найшвидших на сьогоднішній день комерційних кодеків для багатоядерних процесорів: кодек JPEG із IPP-7.1 компанії Intel, `uic_transcoder_con.exe`, version 7.1 build 205.85, [7.7.1058.205], name `ippj8-7.1.dll+`, 64-біт.



Для порівняння з розробленим методом на базі ІПК був використаний той самий набір тестових зображень (табл. 3), рівень якості (параметр $-q$) обраний відповідно до усередненого значення пікового співвідношення сигнал/шум, розрахованого при виконанні тестів. Командний рядок мав такий вигляд: `uic_transcoder_con.exe -otestX_2.jpg -itestX_1.bmp -t1 -q65 -n1`, що означає стиснення зображення `testX_1.bmp` (X – визначає тестове зображення, від 1-го до 5-го), створення нового стисненого зображення `testX_2.jpg`, використаний алгоритм – Baseline JPEG ($-jb$), вимірювання часу стиснення програмою проводилося з підвищеною точністю ($-t1$), якість стиснення 65 Дб ($-q65$).

Приклад результату виконання процедури стиснення тестових зображень за допомогою кодека IPP-7.1 наведений на рис. 3.

```

Командная строка Visual Studio (2010)
D:\w_ipp-samples-binary_p_7.1.1.013\ipp-samples.7.1.1.013\uiс\uiс.intel64.vс2010
.d.st\_bin\release>uiс_transcoder_con.exe -itest1_1.bmp -otest1_2.jpg -jb -q65
-nl -t1
Intel(R) Integrated Performance Primitives
version: 7.1.1 (x37466), [7.1.37466.1]
name: ippJP AUX (e9)
date: Sep 25 2012
WARNING: initialization of CTimer may take long!
image: test1_1.bmp, 640x424x3, 8-bits unsigned, color: RGB, sampling: 444
decode time: 0.45 msec
encode time: 4.83 msec
D:\w_ipp-samples-binary_p_7.1.1.013\ipp-samples.7.1.1.013\uiс\uiс.intel64.vс2010
.d.st\_bin\release>

```

Рис. 3. Результат стиснення тестового зображення test1_1.bmp з використанням алгоритму JPEG засобами кодека IPP-7.1

В табл. 4 зведені значення часу стиснення тестових зображень t_{comp_JPEG} за алгоритмом JPEG за допомогою кодека IPP-7.1.

Таблиця 4

Час стиснення тестових зображень за алгоритмом JPEG

Тестове зображення	Розмірність зображення, $L \times C$, пікселі	Час стиснення методом на базі ІПК, t_{comp_JPEG} , мс
test1_1.bmp	640×424	4,83
test2_1.bmp	1024×768	13,96
test3_1.bmp	1600×1064	25,57
test4_1.bmp	4272×2848	185,06
test5_1.bmp	5760×3840	315,43

На рис. 4 наведені діаграми залежності часу стиснення зображень від розміру матриці зображення для розробленого методу на базі ІПК та алгоритму JPEG. Потенційний вигравш за часом доставки, що забезпечується розробленим методом на базі ІПК в порівнянні з JPEG: $\Delta_{dost} = t_{dost_JPEG} - t_{dos_ISC}$.

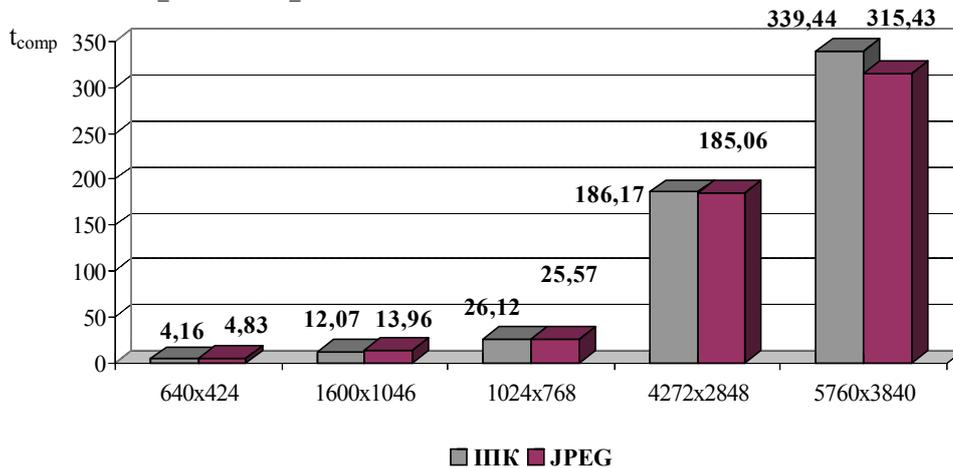


Рис. 4. Діаграми залежності часу стиснення t_{comp} від розмірності зображень

Тут t_{dost_JPEG} – час доставки зображень при використанні кодеків на базі алгоритму JPEG; t_{dost_ISC} – час доставки даних при використанні кодеків на базі розробленого методу стиснення зображень.

Час доставки для кожного з порівнюваних випадків з урахуванням формули (1) складе:

$$t_{dost_JPEG} = t_{obr_JPEG} + t_{per_JPEG}; \quad t_{dost_ISC} = t_{obr_ISC} + t_{per_ISC}$$

де t_{obr_ISC} – час обробки даних кодеком на базі розробленого методу ІПК. Оскільки розроблений метод стиснення зображень є симетричним, то час відновлення даних вважаємо рівним часу стиснення, тоді:

t_{per_ISC} – час передачі даних по КПД після стиснення кодеком на базі розробленого методу ІПК. Для зображення з кількістю рядків L довжиною кожного рядка C (кількістю стовпців), а також глибиною q оцифровки кожного пікселя при пропускну здатності КПД S_{ch} ця величина складатиме:

$$t_{per_ISC} = \frac{L \times C \times q}{k_{ISC} \times S_{ch}}.$$

Тут t_{obr_JPEG} – час обробки даних кодеком на базі JPEG. Оскільки алгоритм JPEG є симетричним, то час відновлення даних вважаємо рівним часу стиснення, тоді:

$$t_{obr_JPEG} = 2 \times t_{comp_JPEG}.$$

Де t_{per_JPEG} – час передачі даних по КПД після стиснення кодеком на базі розробленого методу ІПК. Для зображення з кількістю рядків L даного масиву довжиною кожного рядка C (кількістю стовпців), а також глибиною q оцифровки кожного пікселя при пропускну здатності КПД S_{ch} ця величина складатиме:

$$t_{per_JPEG} = \frac{L \times C \times q}{k_{JPEG} \times S_{ch}}.$$

Тоді абсолютний вииграш в часі доставки зображення, що забезпечується розробленим методом, в порівнянні з алгоритмом JPEG складе:

$$\Delta_{dost} = \frac{L \times C \times q}{S_{ch}} \times \frac{k_{ISC} - k_{JPEG}}{k_{ISC} \times k_{JPEG}} + 2 \times (t_{comp_JPEG} - t_{comp_ISC}).$$

В реальних обставинах дані передаються від одного абонента до іншого по гібридних мережах, в які можуть бути залучені декілька технологій.

Наприклад, якщо дані передаються з персонального комп'ютера у локальній мережі, яка має вихід до мережі Інтернет на певний віддалений сервер, який, наприклад, знаходиться в межах того самого міста, то скоріше за все вони будуть передаватися по каналах передачі на основі витвої пари в межах локальної мережі та по оптоволоконних каналах в межах міської мережі. Очевидно, що це вплине на результуючу пропускну здатність каналу передачі. Для того щоб оцінити реальні пропускну здатності каналів передачі, були зібрані дані за допомогою онлайн-сервісу, який вираховує пропускну здатність каналу передачі між персональним комп'ютером та сервером в певних містах будь-якої країни світу.

В табл. 5 наведені значення реальних пропускну здатностей для каналів передачі даних для різних міст-отримувачів з урахуванням того, що місто-відправник – Київ.

Проведемо розрахунок значення Δ_{dost} , використовуючи такі умови:

– значення t_{comp_ISC} та t_{comp_JPEG} – розраховані під час тестових досліджень швидкодії розробленого методу стиснення зображень (табл. 5 та 6);

– згідно з дослідженнями ступеня стиснення, описаними в попередніх роботах, розроблений метод забезпечує вииграш в стисненні в порівнянні з алгоритмом JPEG в діапазоні від 1,07 до 1,11 разів залежно від ступеня кореляції пікселів зображення. Вважатимемо, що на практиці досягатиметься найнижчий з вииграшів у стисненні, тобто $k_{ISC} = 1.07 \times k_{JPEG}$, значення k_{ISC} для цього випадку складає 5,95.

Тоді абсолютний виграш у часі доставки з урахуванням всіх наведених умов складе:

$$\Delta_{\text{dost}} = \frac{L \times C \times q}{10^8} \times \frac{0,07}{5,95} + 2 \times (t_{\text{comp_JPEG}} - t_{\text{comp_ISC}}). \quad (4)$$

Відносний виграш у часі доставки:

$$\delta_{\text{dost}} = \frac{\Delta_{\text{dost}}}{t_{\text{dost_JPEG}}} \times 100\%. \quad (5)$$

В табл. 6 зведені всі вихідні дані для розрахунку виграшу у часі доставки для тестових зображень.

Таблиця 5
Пропускні спроможності S_{ch} сучасних каналів зв'язку

Місто-отримувач	Пропускна здатність, S_{ch}
Київ (Україна)	92 Мбіт/с
Одеса (Україна)	83 Мбіт/с
Запоріжжя (Україна)	52 Мбіт/с
Санкт-Петербург (Росія)	90 Мбіт/с
Москва (Росія)	56 Мбіт/с
Варшава (Польща)	83 Мбіт/с
Брюссель (Бельгія)	67 Мбіт/с
Лондон (Великобританія)	36 Мбіт/с
Нью-Йорк (США)	18 Мбіт/с
Гонконг (Гонконг)	3 Мбіт/с
Токіо (Японія)	2,5 Мбіт/с
Сідней (Австралія)	6 Мбіт/с

Діаграми на рис. 5 дозволяють порівняти час доставки, що забезпечується при використанні розробленого методу стиснення для обробки зображень при передачі їх по КПД, та час доставки при використанні алгоритму JPEG для зображень різного розміру.

Наведені результати дають можливість зробити такі висновки: використання розробленого методу на базі ІПК для обробки зображень дозволяє скоротити час доставки в порівнянні з випадком використання алгоритму JPEG на значення, що змінюється в діапазоні від 2 до 32 мс залежно від розміру зображення й забезпечує виграш в діапазоні від 1 до 10%, що доводить доцільність використання розробленого методу стиснення зображень в каналах передачі даних.

Таблиця 6
Виграш у часі доставки тестових зображень, що забезпечується методом на базі ІПК в порівнянні з алгоритмом JPEG

Тестове зображення	Розмірність зображення, $L \times C$, пікселі	Час стиснення ІПК, $t_{\text{comp_JPEG}}$, мс	Час стиснення ІПК, $t_{\text{comp_ISC}}$, мс	Час доставки JPEG, $t_{\text{dost_JPEG}}$, мс	Час доставки ІПК, $t_{\text{dost_ISC}}$, мс	Абсолютний виграш у часі доставки, Δ_{dost} , мс	Відносний виграш у часі доставки, δ_{dost} , %
test1_1.bmp	640×424	4,83	4,16	21,4	19,26	2,14	10
test2_1.bmp	1024×768	13,96	12,07	61,92	55,86	6,07	9,8
test3_1.bmp	1600×1064	25,57	26,12	124,75	120,91	3,84	3,1
test4_1.bmp	4272×2848	185,06	186,71	896,24	864,17	32,06	3,5
test5_1.bmp	5760×3840	315,43	339,44	1587,33	1571,05	16,43	1

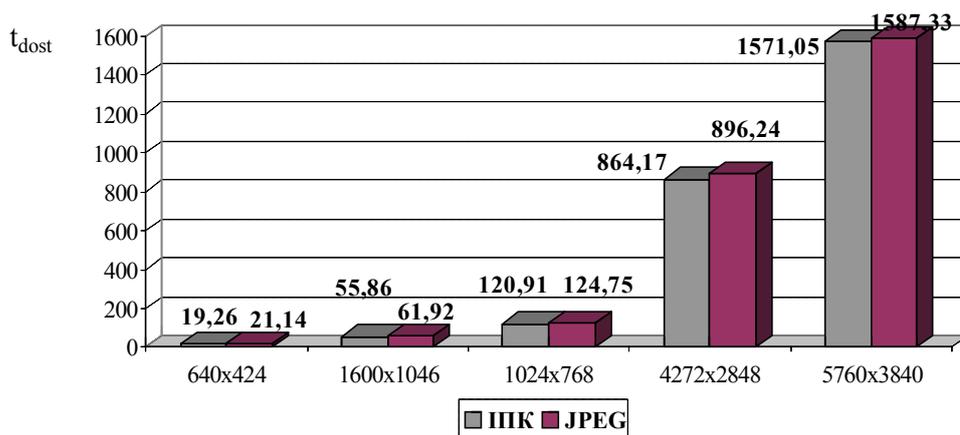


Рис. 5. Діаграми залежності часу доставки t_{dost} від розмірності зображень

3. Висновки

В ході наукових досліджень, описаних у статті, були отримані такі наукові результати:

1. Створено методику оцінки ефективності технології стиснення на базі методу ШК за швидкодією.
2. Створено аналітичну модель розрахунку часу обробки зображень при використанні технології стиснення на базі методу ШК.
3. Проведено розрахунок часових затрат на обробку тестових зображень кодеком на базі ШК.
4. Проведено порівняльну характеристику запропонованої технології стиснення зображень за швидкодією з кодеками на базі алгоритму JPEG.

Отримані результати дозволяють зробити такі висновки: використання розробленого методу на базі ШК для обробки зображень дозволяє скоротити час доставки в порівнянні з використанням алгоритму JPEG на значення, що змінюється в діапазоні від 2 до 32 мс залежно від розміру зображення й забезпечує вигоду в діапазоні від 1 до 10%, що доводить доцільність використання розробленого методу стиснення зображень в каналах передачі даних.

Список літератури: 1. *Методы передачи изображений. Сокращение избыточности* / Под. ред. У.К. Прэтта. М.: Радио и связь, 1983. 263 с. 2. *Трофимов Б.Е. Передача изображений в цифровой форме* / Б.Е. Трофимов, О.В. Куликовский. М.: Связь, 1980. 130 с. 3. *Розенфельд А. Распознавание и обработка изображений* / А. Розенфельд. М.: Мир, 1972. 324 с. 4. *Юдін О.К. Метод кодування двійкових послідовностей за кількістю бітових переходів* / К.О. Курінь, М.Г. Луцький, О.К. Юдін // Наукоємні технології. К.: Вид-во Нац. авіац. ун-ту «НАУ-друк», 2012. № 4 (16). С. 98-102. 5. *Гумен М.Б. Технологія стиснення на базі методу кодування двійкових послідовностей за кількістю бітових переходів* / М.Б. Гумен, О.К. Юдін, К.О. Курінь // Захист інформації: науково-технічний журнал. К.: Вид-во Нац. авіац. ун-ту «НАУ-друк», 2012. №4. С. 12-18. 6. *Юдін О.К. Decompression of images on base of method of decoding according to the amount of bit changes* / К.О. Курінь, О.К. Юдін // Наукоємні технології. К.: Вид-во Нац. авіац. ун-ту «НАУ-друк», 2013. № 2 (18). С. 197-201. 7. *Юдін О.К. Method of structural coding of binary sequences* / К.О. Курінь, О.К. Юдін // Nauka i Studia – Przemysl: “Nauka i studia”, 2012. №20(65). С.36-41. 8. *Юдін О.К. Method of decoding of binary sequences, invariant to statistical* / К.О. Курінь, О.К. Юдін // Наукоємні технології. К.: Вид-во Нац. авіац. ун-ту «НАУ-друк», 2013. № 4 (20). С. 430-435. 9. *Fog A. Lists of instruction latencies, throughputs and micro-operation breakdowns for Intel, AMD and VIA CPUs. Part 4.* / A. Fog – Technical University of Denmark, 2014. 249 p.

Поступила в редколлегию 19.06.2014

Юдін Олександр Костянтинович, д-р техн. наук, професор, начальник кафедри комп'ютеризованих систем захисту інформації КНАУ. Наукові інтереси: обробка та передача інформації. Адреса: Україна, Київ, вул. Космонавта Комарова, 1.

Курінь Ксенія Олександрівна, здобувач кафедри комп'ютеризованих систем захисту інформації КНАУ. Наукові інтереси: обробка та передача інформації. Адреса: Україна, Київ, вул. Космонавта Комарова, 1.

Зюбіна Руслана Віталіївна, асистент кафедри комп'ютеризованих систем захисту інформації КНАУ. Наукові інтереси: обробка та передача інформації. Адреса: Україна, Київ, вул. Космонавта Комарова, 1.

МЕТОД ОБРАБОТКИ ВИДЕОРЕСУРСОВ С СОХРАНЕНИЕМ ЦЕЛОСТНОСТИ В ИНФОРМАЦИОННЫХ СИСТЕМАХ

Показывается, что развитие мультимедийных технологий способствует появлению новых видеоинформационных сервисов, связанных с предоставлением услуги цифрового видео высокого качества в трехмерном разрешении. Излагаются этапы разработки трехмерного кодирования данных, начиная с младших элементов по схеме, когда код формируется в результате поэтапного обобщения отдельных укрупненных элементов. Выявляются недостатки поэтапного трехмерного полиадического кодирования данных, начиная с младших элементов, для универсальных вычислительных архитектур. Разрабатывается поэтапная *рекуррентная* схема вычислений, начиная с младших элементов.

Ключевые слова: трехмерные структуры видеоданных, полиадическое число.

Введение

Значимое развитие мультимедийных технологий и развитие новых видеоинформационных сервисов связано с необходимостью обработки многомерных динамических видеосцен [1 – 3]. Происходит существенное увеличение интенсивности видеопотоков. Несмотря на рост производительности инфокоммуникационных систем, проявляется дисбаланс относительно требований видеоприложений, в том числе с использованием мобильного беспроводного доступа [1]. Такая ситуация приводит к угрозам безопасности видеоинформационного ресурса. Возникают проблемы относительно обеспечения своевременной доступности видеоинформации и неконтролируемых потерь целостности [5].

Технологии обработки видеоресурса базируются на подходах, использующих технологии обработки видеосцен преимущественно как двумерных объектов [2 – 5]. Поэтому требуется разработка новых технологий, которые будут учитывать многомерность структуры видеоресурсов. Значит, актуальной проблематикой исследований является совершенствование теоретической и технологической базы для создания новых методов обработки видеоресурсов как трехмерных объектов. В работах [6 – 10] показано, что ключевым направлением является построение теоретических основ и методов обработки трехмерных структур данных. Необходимо создать метод обработки трехмерно-структурированных видеоресурсов (ТСВ) без потери целостности на основе устранения структурной избыточности в трехмерном пространстве. В работе [10] предложен подход для создания кодирования, которое потенциально обеспечивает сокращение структурной избыточности в трехмерном пространстве. Основным недостатком такого подхода является то, что кодирование допускается проводить, начиная со старших элементов, и весовой коэффициент текущего элемента зависит от оснований всех последующих, не обработанных элементов. Это приводит к усложнению процесса обработки. Отсюда цель исследований заключается в создании метода обработки трехмерных структур видеоресурсов без потери их целостности в условиях снижения времени обработки как составляющей обеспечения доступности видеоинформационного ресурса.

1. Создание трехмерного рекуррентного кодирования трехмерно-структурированных видеосцен

Предлагается разработать кодирование трехмерных неравновесных позиционных чисел, начиная с младших элементов. Суть такого кодирования состоит в следующем. Процесс кодирования организуется на основе следующих этапов [11]:

На первом этапе формируется код для отдельных вертикалей трехмерной структуры данных (ТСД). Организация первого этапа для (j_i) -й вертикали ТСВ задается выражениями:

$$N_1^{(ji)} = a_{ji1}; \quad N_z^{(ji)} = N_{z-1}^{(ji)} + a_{jiz} \prod_{\gamma=1}^{z-1} \psi_{ji\gamma}, \quad (1)$$

где $N_z^{(ji)}$ – значение кода для z элементов (ji) -й вертикали, т.е. значение кода $N_{n_c}^{(ji)}$ для полной (ji) -й вертикали ТСВ (для всех $z = \overline{1, n_c}$ элементов (ji) -й вертикали) находится по формуле

Перед формированием кода с учетом добавления очередного элемента проверяются неравенства $V_z^{(ji)} = \prod_{\gamma=1}^z \psi_{ji\gamma} \leq 2^M - 1$; $V_{n_c}^{(ji)} = \prod_{\gamma=1}^{n_c} \psi_{ji\gamma} \leq 2^M - 1$. Здесь где $V_z^{(ji)}$, $V_{n_c}^{(ji)}$ – значения накопленных произведений оснований элементов ТСВ для проверки на допустимость соответственно элементов a_{jiz} и a_{jin_c} .

Второй этап рекуррентного кодирования состоит в обобщении кодов отдельных вертикалей по строкам. Тогда для столбца с индексом j используются следующие формулы:

$$N_{i, n_c}^{(j)} = N_{n_c}^{(ji)}; \quad N_{i, n_c}^{(j)} = N_{i-1, n_c}^{(j)} + N_{n_c}^{(j,i)} \prod_{\gamma=1}^{n_c-1} \psi_{ji\gamma} \prod_{\beta=1}^{i-1} V_{n_c}^{(j\beta)}, \quad (2)$$

где $N_{i, n_c}^{(j)}$ – значение кода, сформированного для i величин $N_{n_c}^{(j,\beta)}$, $\beta = \overline{1, i}$, принадлежащих j -му столбцу.

Соответственно проверка на допустимость добавления элемента $N_{n_c}^{(j,i)}$ проводится по правилу, заданному неравенством $V_{i, n_c}^{(j)} = \prod_{\beta=1}^i V_{n_c}^{(j\beta)} = \prod_{\beta=1}^i \prod_{\gamma=1}^{n_c} \psi_{j\beta\gamma} \leq 2^M - 1$, где $V_{i, n_c}^{(j)}$ – накопленное произведение оснований элементов ТСД для i вертикалей по n_c элементов, принадлежащих j -у столбцу.

Третий этап используется для обобщения кодов $N_{n_{стр}, n_c}^{(j)}$, вычисленных для отдельных столбцов в один код $N_{n_{стр\delta}, n_{стр}, n_c}$ всей трехмерной структуры видеоданных. Процесс формирования величины $N_{n_{стр\delta}, n_{стр}, n_c}$ для j -го количества столбцов (вертикальных сечений) ТСВ задается следующими соотношениями:

$$V_{j, n_{стр}, n_c} = \prod_{\eta=1}^j V_{n_{стр}, n_c}^{(\eta)} = \prod_{\eta=1}^j \prod_{\beta=1}^{n_{стр}} V_{n_c}^{(\eta\beta)} = \prod_{\eta=1}^j \prod_{\beta=1}^{n_{стр}} \prod_{\gamma=1}^{n_c} \psi_{\eta\beta\gamma} \leq 2^M - 1, \quad (3)$$

$$N_{j, n_{стр}, n_c} = N_{j-1, n_{стр}, n_c} + N_{n_{стр}, n_c}^{(j)} \prod_{\eta=1}^j \prod_{\beta=1}^{n_{стр}} \prod_{\gamma=1}^{n_c-1} \psi_{\eta\beta\gamma} \prod_{\eta=1}^{j-1} V_{n_{стр}, n_c}^{(\eta)}, \quad (4)$$

где $\prod_{\eta=1}^j \prod_{\beta=1}^{n_{стр}} \prod_{\gamma=1}^{n_c-1} \psi_{\eta\beta\gamma} \prod_{\eta=1}^{j-1} V_{n_{стр}, n_c}^{(\eta)}$ – весовой коэффициент элемента $N_{n_{стр}, n_c}^{(j)}$.

По аналогии с выражениями (3) и (4) значение кода с учетом добавления элемента $N_{n_{стр\delta}, n_c}^{(n_{стр\delta})}$ определяется формулой

$$N_{n_{стр\delta}, n_{стр}, n_c} = N_{n_{стр\delta}-1, n_{стр}, n_c} + N_{n_{стр}, n_c}^{(n_{стр\delta})} \prod_{\eta=1}^{n_{стр\delta}} \prod_{\beta=1}^{n_{стр}} \prod_{\gamma=1}^{n_c-1} \psi_{\eta\beta\gamma} \prod_{\eta=1}^{n_{стр\delta}-1} V_{n_{стр}, n_c}^{(\eta)}, \quad (5)$$

где $\prod_{\eta=1}^{n_{\text{стб}}} \prod_{\beta=1}^{n_{\text{стр}}} \prod_{\gamma=1}^{n_c-1} \Psi_{\eta\beta\gamma} \prod_{\eta=1}^{n_{\text{стб}}-1} V_{n_{\text{стр}}, n_c}^{(\eta)}$ – весовой коэффициент элемента $N_{n_{\text{стр}}, n_c}^{(n_{\text{стб}})}$;

$N_{n_{\text{стб}}-1, n_{\text{стр}}, n_c}$ – значение кода, сформированного для $(n_{\text{стб}} - 1)$ -го количества вертикальных сечений.

Таким образом, создана трехмерная обработка видеоресурсов, обеспечивающая формирование кода-номера по схеме поэтапного обобщения отдельных укрупненных элементов. Такая схема удобна в случае конвейерной реализации на спецустройствах. В то же время для реализации разработанного кодирования на универсальных вычислительных системах требуется разработать поэлементную рекуррентную схему вычислений. В этом случае нужно получить выражение для формирования кода-номера с учетом добавления элемента, расположенного в произвольном месте ТСД.

2. Разработка метода кодирования элементов трехмерных видеосцен на основе поэлементной рекуррентной схемы

Рассмотрим процесс вычисления кода $N^{(jiz)}$ с учетом добавления элемента a_{jiz} . Поскольку элемент a_{jiz} находится в произвольном месте ТСВ, то отсекаемая часть ТСВ будет в общем случае состоять из трех частей: неполная вертикаль с координатами (ji) ; неполное j -е вертикальное сечение; $(j-1)$ полных вертикальных сечений. Поэтому предлагается вычислять код $N^{(jiz)}$ как сумму трех составляющих, образованных для каждой структурной части ТСВ. С учетом выражения (1) составляющая для первой части равна

$$(N_{z-1}^{(ji)} + a_{jiz} \Psi_{j,i,z-1}) \prod_{\phi=1}^{z-2} \Psi_{ji\phi} \prod_{\beta=1}^{i-1} \prod_{\phi=1}^{n_c} \Psi_{j\beta\phi} \prod_{\alpha=1}^{j-1} \prod_{\beta=1}^{n_{\text{стр}}} \prod_{\phi=1}^{n_c} \Psi_{\alpha\beta\phi}. \quad (6)$$

В соответствии с формулами (1), (2) составляющая для второй структурной части равна

$$(N_{i-2, n_c}^{(j)} + N_{n_c}^{(j,i-1)} \prod_{\gamma=1}^{n_c-1} \Psi_{j,i-1,\gamma} \prod_{\beta=1}^{i-2} V_{n_c}^{(j\beta)}) \prod_{\alpha=1}^{j-1} \prod_{\beta=1}^{n_{\text{стр}}} \prod_{\phi=1}^{n_c} \Psi_{\alpha\beta\phi}. \quad (7)$$

Для третьей структурной части согласно соотношению (4) значение составляющей кода $N^{(jiz)}$ определяется как

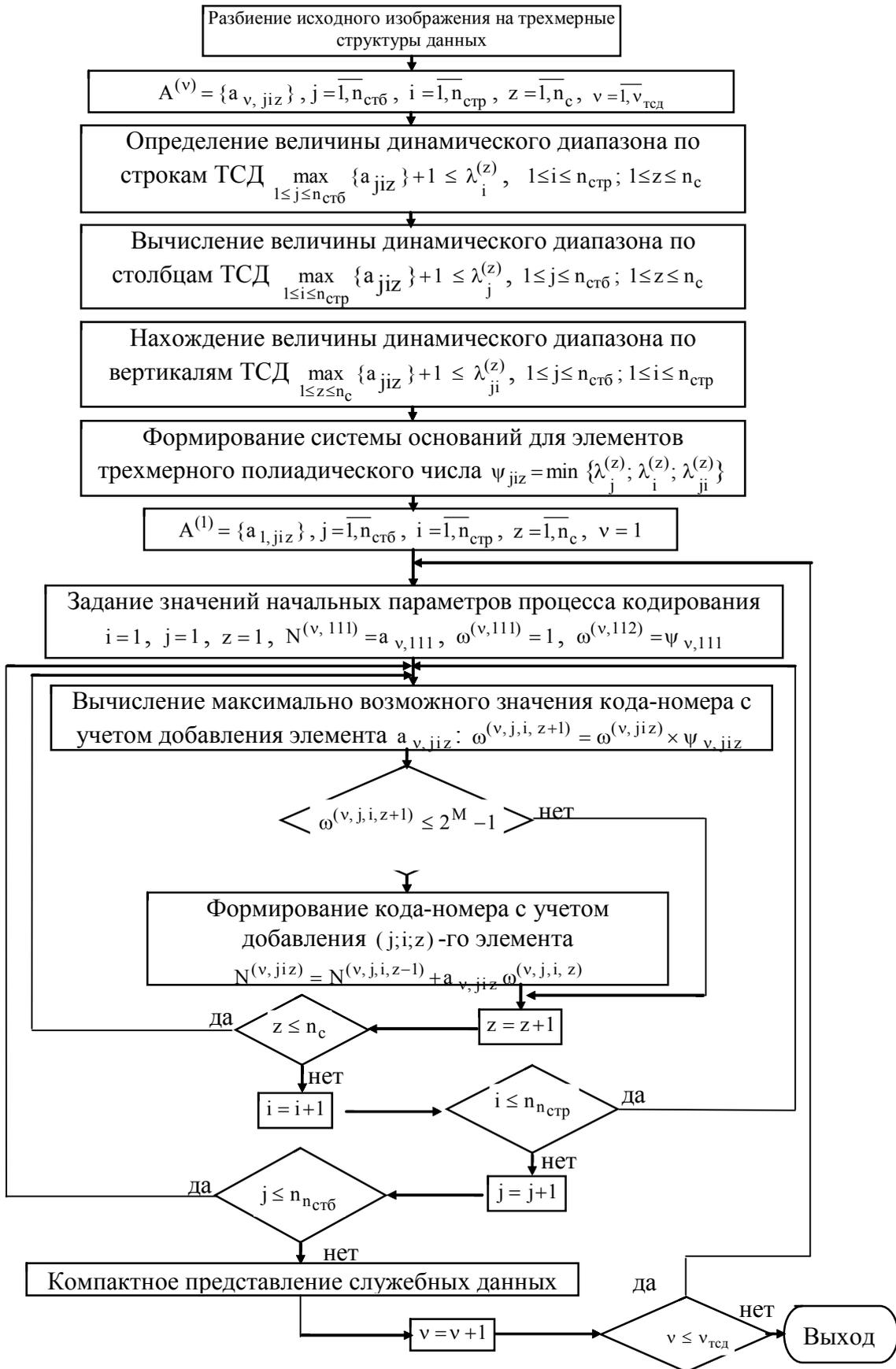
$$N_{j-2, n_{\text{стр}}, n_c} + N_{n_{\text{стр}}, n_c}^{(j-1)} \prod_{\eta=1}^{j-1} \prod_{\beta=1}^{n_c-1} \prod_{\gamma=1}^{n_{\text{стр}}} \Psi_{\eta\beta\gamma} \prod_{\eta=1}^{j-2} V_{n_{\text{стр}}, n_c}^{(\eta)}. \quad (8)$$

Расписав в выражениях (6) – (8) величины $N_{z-1}^{(ji)}$, $N_{n_c}^{(j,i-1)}$, $N_{i-2, n_c}^{(j)}$, $N_{n_{\text{стр}}, n_c}^{(j-1)}$ и $N_{j-2, n_{\text{стр}}, n_c}$, получим соотношение для вычисления кода $N^{(jiz)}$ без потери целостности видеoinформационного ресурса, начиная с младших элементов, с учетом добавления элемента a_{jiz} , расположенного в произвольном месте ТСД, а именно:

$$N^{(jiz)} = \sum_{\eta=1}^j \sum_{k=1}^i \sum_{\gamma=1}^z a_{\eta k \gamma} \prod_{\phi=1}^{\gamma-1} \Psi_{\eta k \phi} \prod_{\beta=1}^{k-1} \prod_{\phi=1}^{n_c} \Psi_{\eta\beta\phi} \prod_{\alpha=1}^{\eta-1} \prod_{\beta=1}^{n_{\text{стр}}} \prod_{\phi=1}^{n_c} \Psi_{\alpha\beta\phi}, \quad (9)$$

где $\prod_{\phi=1}^{\gamma-1} \Psi_{\eta k \phi} \prod_{\beta=1}^{k-1} \prod_{\phi=1}^{n_c} \Psi_{\eta\beta\phi} \prod_{\alpha=1}^{\eta-1} \prod_{\beta=1}^{n_{\text{стр}}} \prod_{\phi=1}^{n_c} \Psi_{\alpha\beta\phi}$ – весовой коэффициент (η, k, γ) -го элемента трехмерной структуры данных.

Для вычисления кода $N^{(jiz)}$ с учетом известного кода $N^{(j,i,z-1)}$, полученного до добавления элемента a_{jiz} , необходимо использовать рекуррентную формулу. Граф-схема компактного представления данных на основе рекуррентного трехмерного кодирования, начиная с младших элементов, приведена на рисунке.



Граф-схема трехмерного кодирования

Для ее получения разобьем соотношение (9) на два слагаемых: одно из них включает только элемент a_{jiz} , а другое – все предыдущие элементы ТСД:

$$\begin{aligned}
 N^{(jiz)} &= \sum_{\eta=1}^j \sum_{k=1}^i \sum_{\gamma=1}^z a_{\eta k \gamma} \prod_{\phi=1}^{\gamma-1} \Psi_{\eta k \phi} \prod_{\beta=1}^{k-1} \prod_{\phi=1}^{n_c} \Psi_{\eta \beta \phi} \prod_{\alpha=1}^{\eta-1} \prod_{\beta=1}^{n_{стр}} \prod_{\phi=1}^{n_c} \Psi_{\alpha \beta \phi} = \sum_{\eta=1}^j \sum_{k=1}^i \sum_{\gamma=1}^z a_{\eta k \gamma} \times \\
 &\times \prod_{\phi=1}^{\gamma-1} \Psi_{\eta k \phi} \prod_{\beta=1}^{k-1} \prod_{\phi=1}^{n_c} \Psi_{\eta \beta \phi} \prod_{\alpha=1}^{\eta-1} \prod_{\beta=1}^{n_{стр}} \prod_{\phi=1}^{n_c} \Psi_{\alpha \beta \phi} + a_{jiz} \prod_{\phi=1}^{z-1} \Psi_{ji\phi} \prod_{\beta=1}^{i-1} \prod_{\phi=1}^{n_c} \Psi_{j\beta\phi} \times \\
 &\times \prod_{\alpha=1}^{j-1} \prod_{\beta=1}^{n_{стр}} \prod_{\phi=1}^{n_c} \Psi_{\alpha\beta\phi} = N^{(j,i,z-1)} + a_{jiz} \prod_{\phi=1}^{z-1} \Psi_{ji\phi} \prod_{\beta=1}^{i-1} \prod_{\phi=1}^{n_c} \Psi_{j\beta\phi} \prod_{\alpha=1}^{j-1} \prod_{\beta=1}^{n_{стр}} \prod_{\phi=1}^{n_c} \Psi_{\alpha\beta\phi}, \quad (10)
 \end{aligned}$$

где $\prod_{\phi=1}^{z-1} \Psi_{ji\phi} \prod_{\beta=1}^{i-1} \prod_{\phi=1}^{n_c} \Psi_{j\beta\phi} \prod_{\alpha=1}^{j-1} \prod_{\beta=1}^{n_{стр}} \prod_{\phi=1}^{n_c} \Psi_{\alpha\beta\phi}$ – весовой коэффициент (j, i, z) -го элемента трехмерной структуры данных.

Пересчет значения кода $N^{(jiz)}$ по выражению (10) проводится в том случае, когда выполняется неравенство

$$\omega^{(jiz)} \Psi_{ji\phi} \leq 2^M - 1. \quad (11)$$

Соотношение (3) согласно выражению (4) примет вид (см. рисунок):

$$N^{(jiz)} = N^{(j,i,z-1)} + a_{jiz} \omega^{(jiz)}. \quad (12)$$

Формулы (9) – (12) позволяют вычислить значение кода в направлении младших элементов как для произвольного количества элементов ТСД, так и для всей ТСВ. Отличительной возможностью разработанного кодирования является то, что значение весового коэффициента зависит только от оснований предыдущих элементов ТСВ. Эта особенность обеспечивает возможность организации процесса восстановления за один проход.

Выводы

1. Обосновано, что основные недостатки трехмерного кодирования в направлении со старших элементов состоят в том, что весовой коэффициент текущего элемента зависит от оснований всех последующих, не обработанных элементов. Это приводит к усложнению процесса декодирования. Поэтому необходимо разработать кодирование трехмерных неравновесных позиционных чисел, начиная с младших элементов. В этом случае весовой коэффициент будет зависеть только от оснований предыдущих, ранее обработанных элементов ТСВ.

2. Разработано трехмерное кодирование видеоресурсов, начиная с младших элементов. Отличительной особенностью разработанного представления является то, что наибольший весовой коэффициент соответствует последнему элементу трехмерного неравновесного позиционного числа. Предложенное кодирование может быть реализовано двумя схемами: формирование кода по схеме поэтапного обобщения отдельных укрупненных элементов и по схеме поэлементных рекуррентных вычислений. Отличительной особенностью разработанного кодирования является то, что значение весового коэффициента зависит только от оснований предыдущих элементов ТСД. Эта особенность обеспечивает возможность организации процесса восстановления за один проход.

3. Для созданного подхода сжатие обеспечиваются за счет исключения структурной избыточности, обусловленной ограниченностью и неравномерностью динамических диапазонов элементов видеоданных одновременно по трем координатам трехмерных структур данных.

Список литературы: 1. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов / В.Г. Олифер, Н.А. Олифер. СПб.: Питер, 2006. 958 с. 2. *Advanced video coding for generic audiovisual services* –ITU-T Recommendation H.264 and ISO/IEC 14496-10 (AVC), 2009. 3. ISO/IEC 13818 (MPEG-2): Generic coding of moving pictures and associated audio information, 1994. 4. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. М.: Техносфера, 2005. 1072 с. 5. Баранник В.В.

Структурно-комбинаторное представление данных в АСУ / В.В. Баранник, Ю.В. Стасев, Н.А. Королева Х.: ХУПС, 2009. 252 с. **6.** *Беляев Е.А.* Сжатие видеоинформации на основе трехмерного дискретного псевдо-косинусного преобразования для энергоэффективных систем видеонаблюдения / Е.А. Беляев, Т.М. Сухов, Н.Н. Шостацкий // Компьютерная оптика. 2010. Т. 34. С. 260 – 272. **7.** *Barannik V.V.* Method of the 3-D Image Processing / V.V. Barannik, S.V. Karpenko // Modern problems of Radio Engineering, Telecommunications and Computer Science. Proceedings of the International Conference TCSET'2008, Lviv-Slavsko, Ukraine, February 20 – 24, 2008. P. 115 – 117. **8.** *B. Furht, Ken Gustafson, Hesong Huang and Oge Marques,* An Adaptive Three-Dimensional DCT Compression Based on Motion Analysis // Proceedings of the 2003 ACM symposium on Applied computing, 2003. **9.** *T. Mekky,* On the computation of the 3-D DCT // IEEE International Conference on Electronics, Circuits and Systems. 2003. Vol. 3. P.1141 – 1143. **10.** *Barannik V.V.* Method of the 3-D Image Processing / V.V. Barannik, S.V. Karpenko // Modern problems of Radio Engineering, Telecommunications and Computer Science. Proceedings of the International Conference TCSET'2008, Lviv-Slavsko, Ukraine, February 20 – 24, 2008. P. 115 – 117. **11.** *Баранник В.В.* Трехмерное полиадическое кодирование в направлении, начиная с младших элементов / В.В.Баранник, Ю.Н. Рябуха // Сучасна спеціальна техніка. 2013. №3. С. 15 – 20.

Поступила в редколлегию 14.06.2014

Рябуха Юрий Николаевич, канд. техн. наук, соискатель Харьковского университета Воздушных Сил. Научные интересы: информационно-телекоммуникационные технологии, кодирование, защита и передача информации. Адрес: Украина, 61023, Харьков, ул. Сумская, 77/79.