

ISSN 1816-0301

# ИНФОРМАТИКА

**1 (41)**

**ЯНВАРЬ-МАРТ  
2014**

## **Редакционная коллегия:**

*Главный редактор*

**А.В. Тузиков**

*Заместитель главного редактора*

**М.Я. Ковалев**

*Члены редколлегии*

С.В. Абламейко, В.В. Анищенко, П.Н. Бибило, М.Н. Бобов,  
А.Н. Дудин, А.Д. Закревский, С.Я. Килин, В.В. Краснопрошин,  
С.П. Кундас, Н.А. Лиходед, П.П. Матус, С.В. Медведев, А.А. Петровский,  
Ю.Н. Сотсков, Ю.С. Харин, А.Ф. Чернявский, В.Н. Яролик  
Н.А. Рудая (*заведующая редакцией*)

---

---

*Адрес редакции:*

220012, Минск,  
ул. Сурганова, 6, к. 305  
тел. (017) 284-26-22  
e-mail: [rio@newman.bas-net.by](mailto:rio@newman.bas-net.by)  
<http://uiip.bas-net.by>

---

---

# ИНФОРМАТИКА

---

---

ЕЖЕКВАРТАЛЬНЫЙ НАУЧНЫЙ ЖУРНАЛ

*Издается с января 2004 г.*

---

---

№ 1(41) • январь-март 2014

## СОДЕРЖАНИЕ

### МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

<b>Науменко В.В., Матальцкий М.А.</b> Анализ марковской сети с доходами, положительными и отрицательными заявками .....	5
<b>Вернер Ф., Кравченко С.А., Хасани К.</b> Минимизация суммарного времени обслуживания для системы с двумя приборами и одним сервером .....	15
<b>Курочка К.С.</b> Конечноэлементное моделирование прогибов тонких круглых трехслойных пластин .....	25
<b>Кульбак Л.И.</b> Интервальная оценка расчетных показателей надежности объекта .....	35
<b>Ерофеев В.Т.</b> Электродинамическая модель расчета эффективных параметров композитов из сферических биизотропных частиц .....	45
<b>Стрижнев А.Г., Ледник Г.В., Русакович А.Н.</b> Синтез цифрового регулятора для объекта управления пятого порядка с применением оптимальных управляющих воздействий .....	59

### ОБРАБОТКА СИГНАЛОВ, ИЗОБРАЖЕНИЙ И РЕЧИ

<b>Гецэвіч Ю.С., Окрут Т.І., Лабанаў Б.М.</b> Алгарытмы ідэнтыфікацыі рэплік са словамі аўтара ў электронных тэкстах на беларускай мове .....	68
<b>Залесский Б.А., Середин Э.Н.</b> Реализация алгоритмов отслеживания объектов на основе технологии CUDA .....	77
<b>Журавлёв А.А., Цветков В.Ю.</b> Масочно-фазовый метод локализации прямых линий на изображении .....	85
<b>Поляков А.С., Самсонов В.Е.</b> Оценка характеристик аппаратной реализации некоторых алгоритмов фильтрации изображений .....	97

## ЗАЩИТА ИНФОРМАЦИИ

<b>Бережной И.Б., Харин Ю.С.</b> Вероятностная модель динамики памяти криптографических генераторов Макларена – Марсальи.....	105
<b>Сергейчик В.В., Иванюк А.А.</b> Особенности обфускации VHDL-описаний и методы оценки ее сложности.....	116

## КОНФЕРЕНЦИИ

<b>Шестой Белорусский космический конгресс</b> .....	126
<b>XIII Международная конференция</b> «Развитие информатизации и государственной системы научно-технической информации» (РИНТИ–2014) .....	129

---

---

Редактор Г.Б. Гончаренко  
Корректор А.А. Михайлова  
Компьютерная верстка Д.С. Гавинович

---

Сдано в набор 07.01.2014. Подписано в печать 28.02.2014.  
Формат 60×84 1/8. Бумага офсетная. Гарнитура Таймс.  
Усл. печ. л. 15,1. Уч.-изд. л. 14,8. Тираж 100 экз. Заказ 10.

---

Государственное научное учреждение «Объединенный институт проблем информатики Национальной академии наук Беларуси».  
ЛИ № 02330/0549421 от 08.04.2009.  
ЛП № 02330/444 от 18.12.13.  
Ул. Сурганова, 6, 220012, Минск.

Отпечатано с оригинала-макета на ризографе Объединенного института проблем информатики Национальной академии наук Беларуси.  
Ул. Сурганова, 6, 220012, Минск.

© Объединенный институт проблем информатики  
Национальной академии наук Беларуси, 2014

# INFORMATICS

---

---

PUBLISHED QUATERLY

*Issued since 2004*

---

---

№ 1(41) • January-March 2013

## CONTENTS

### MATHEMATICAL MODELING

<b>Naumenko V.V., Matalytski M.A.</b> Analysis of markov network with incomes, positive and negative messages .....	5
<b>Werner F., Kravchenko S.A., Hasani K.</b> Minimizing mean flow time for the two-machine scheduling problem with a single server .....	15
<b>Kurachka K.S.</b> Finite element modeling of thin circular sandwich plates deflection .....	25
<b>Kulbak L.I.</b> Interval estimation of object reliability indices .....	35
<b>Erofeenko V.T.</b> Electrodynamical model for calculating effective parameters of composites from spherical bi-isotropic particles .....	45
<b>Stryzhniou A.G., Lednik H.V., Rusakovich A.N.</b> Synthesis of optimal digital controller for the fifth order controlled object .....	59

### SIGNAL, IMAGE AND SPEECH PROCESSING

<b>Hetsevich Y.S., Okrut T.I., Lobanov B.M.</b> Algorithms for identification of cues with authors' text insertions in belarusian electronic books .....	68
<b>Zalesky B.A., Seregin E.N.</b> Implementation of object tracking algorithms on the basis of CUDA technology .....	77
<b>Zhuravlyov A.A., Tsviatkou V.Yu.</b> Mask-phase method of localization of straight lines in an image .....	85
<b>Poljakov A.S., Samsonov V.E.</b> Evaluation of characteristics of hardware implementation of some image filtering algorithms .....	97

## INFORMATION SECURITY

<b>Berezhnoy I.B., Kharin Yu.S.</b> Probabilistic model of memory dynamics of Maclaren – Marsaglia cryptographic generators.....	105
<b>Sergeichik V.V., Ivaniuk A.A.</b> Features of obfuscation of VHDL-designs and its complexity evaluation methods.....	116

## CONFERENCES

<b>The Sixth Belarussian space congress</b> .....	126
<b>XIII International conference</b> «Informatization development and system of scientific and technical information».....	129

## МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

УДК 519.872

В.В. Науменко, М.А. Маталыцкий

АНАЛИЗ МАРКОВСКОЙ СЕТИ С ДОХОДАМИ,  
ПОЛОЖИТЕЛЬНЫМИ И ОТРИЦАТЕЛЬНЫМИ ЗАЯВКАМИ

*Рассматривается марковская сеть массового обслуживания с доходами в переходном режиме, с положительными и отрицательными заявками, которая может использоваться при прогнозировании доходов в информационно-телекоммуникационных системах и сетях с учетом попадания в них вирусов. Исследования проводятся в случаях, когда доходы от переходов между состояниями сети являются детерминированными функциями, зависящими от ее состояний, и случайными величинами с заданными средними значениями. В последнем случае предполагается, что все системы сети функционируют в режиме высокой нагрузки. Приводится пример расчета.*

**Введение**

Современные информационно-телекоммуникационные системы и сети становятся все более сложными, что обусловлено необходимостью повышения надежности передачи и обработки информации. Построение и исследование математических моделей для оценки качества их функционирования является важной задачей. Применение для этой цели классических моделей теории массового обслуживания (МО) не всегда дает адекватные результаты, поскольку необходимо, чтобы модели учитывали как характерные особенности систем, так и возможное влияние различных дестабилизирующих факторов, например внезапных сбоев, попадания вирусов, потери передаваемых или обрабатываемых данных.

Для учета подобных факторов была предложена концепция отрицательных заявок и связанных с ними сетей и систем МО. Принципиально новый класс сетей МО – G-сетей, в которых помимо потоков обычных (положительных) заявок рассматриваются также дополнительные пуассоновские потоки отрицательных заявок, был введен Е. Геленбе в [1]. При поступлении в систему сети отрицательная заявка уничтожает одну положительную заявку, если таковая имеется в данной системе, тем самым уменьшая число положительных заявок в системе на единицу. Затем отрицательная заявка исчезает из сети, не получив никакого обслуживания. Например, в компьютерных сетях «положительными» заявками являются задания (программы), а «отрицательными» заявками – компьютерные вирусы. При поступлении в компьютерную сеть вирус уничтожает или наносит вред, заражает одну из исполняемых программ, уменьшая количество действующих программ или запросов в системе на единицу. Следует отметить, что исследование G-сетей в стационарном режиме проведено в работах [2, 3].

При попадании вируса в информационную систему из-за потери информации или ее искажения система и вся информационно-телекоммуникационная сеть несут некоторые расходы или убытки. Учет их можно осуществить, применив в качестве модели сеть МО с доходами (НМ-сеть) и с положительными и отрицательными заявками. После исчезновения отрицательной заявки система массового обслуживания (СМО) вновь после обслуживания в ней положительных заявок получает некоторый доход (прибыль). Во второй части данной статьи описана методика нахождения ожидаемых доходов в системах такой сети. Отметим, что НМ-сети уже применялись при прогнозировании доходов в логистических транспортных системах [4].

**1. Постановка задачи**

Рассмотрим открытую G-сеть МО с  $n$  однолинейными СМО. В СМО  $S_i$  извне (из системы  $S_0$ ) поступает пуассоновский поток положительных (обычных) заявок с интенсивностью  $\lambda_{0i}^+$  и пуассоновский поток отрицательных заявок с интенсивностью  $\lambda_{0i}^-$ ,  $i = \overline{1, n}$ . Все поступа-

ющие в сеть потоки заявок являются независимыми. Длительности обслуживания положительных заявок в СМО  $S_i$  распределены по экспоненциальному закону с параметром  $\mu_i$ ,  $i = \overline{1, n}$ . Отрицательная заявка, поступающая в некоторую систему сети, в которой имеется по крайней мере одна положительная заявка, мгновенно уничтожает одну из них и наносит убыток этой СМО. При предположении экспоненциального распределения времени обслуживания положительных заявок можно не заботиться о том, какая именно заявка уничтожается. После этого отрицательная заявка сразу же покидает сеть или уничтожается, если в данной СМО не было заявок. Таким образом, в каждой СМО сети могут обслуживаться только положительные заявки, поэтому в дальнейшем, говоря об обслуживании положительных заявок, обычно для краткости называют их просто заявками [1]. Положительная заявка при переходе из одной СМО в другую приносит последней системе некоторый доход, и соответственно доход первой системы уменьшается на эту величину.

Положительная заявка, обслуженная в СМО  $S_i$ , с вероятностью  $p_{ij}^+$  направляется в СМО  $S_j$  как положительная заявка, с вероятностью  $p_{ij}^-$  – как отрицательная заявка, и с вероятностью

$$p_{i0} = 1 - \sum_{j=1}^n (p_{ij}^+ + p_{ij}^-) \quad (1)$$

заявка уходит из сети во внешнюю среду (СМО  $S_0$ ),  $i, j = \overline{1, n}$ . Под состоянием сети будем понимать вектор  $k(t) = (k, t) = (k_1(t), k_2(t), \dots, k_n(t))$ , где  $k_i(t)$  – число заявок в момент времени  $t$  в системе  $S_i$ ,  $i = \overline{1, n}$ .

## 2. Анализ сети с доходами, когда доходы от переходов между состояниями сети являются детерминированными функциями

Будем рассматривать сеть с учетом доходов и расходов СМО сети при обслуживании положительных и отрицательных заявок. Проанализируем случай, когда доходы от переходов между состояниями сети являются детерминированными функциями, зависящими от состояний сети и времени. Пусть функция  $v_i(k, t)$  – полный ожидаемый доход, который получает система  $S_i$  за время  $t$ , если в начальный момент времени сеть находится в состоянии  $k$ . Предположим, что эта функция дифференцируема по  $t$ ;  $I_i$  – вектор с ненулевыми компонентами за исключением компоненты с номером  $i$ , которая равна 1,  $i = \overline{1, n}$ ;  $I_0$  – нулевой  $n$ -вектор;  $r_i(k)$  – доход системы  $S_i$  в единицу времени, когда сеть находится в состоянии  $k$ ;  $r_{0i}(k + I_i, t)$  – доход системы  $S_i$ , когда сеть совершает переход из состояния  $(k, t)$  в состояние  $(k + I_i, t + \Delta t)$  за время  $\Delta t$ ;  $-R_{i0}(k - I_i, t)$  – доход этой системы, если сеть совершает переход из состояния  $(k, t)$  в состояние  $(k - I_i, t + \Delta t)$ ;  $r_{ij}(k + I_i - I_j, t)$  – доход системы  $S_i$  (расход или убыток системы  $S_j$ ), когда сеть изменяет свое состояние из  $(k, t)$  на  $(k + I_i - I_j, t + \Delta t)$  за время  $\Delta t$ ;  $-r_{ij}(k - I_i - I_j, t)$  – доход системы  $S_i$ , когда сеть изменяет свое состояние из  $(k, t)$  на  $(k - I_i - I_j, t + \Delta t)$  за время  $\Delta t$ ,  $i, j = \overline{1, n}$ . Требуется найти ожидаемые (средние) доходы систем сети за время  $t$  при условии, что нам известно ее состояние в начальный момент времени.

Возможные переходы между состояниями сети, вероятности переходов и доходы системы  $S_i$  от этих переходов указаны в таблице, согласно которой, например, если сеть остается в состоянии  $k$  в момент времени  $t + \Delta t$ , то ожидаемый доход системы  $S_i$  составит  $r_i(k)\Delta t$  за время  $\Delta t$  плюс ожидаемый доход  $v_i(k, t)$ , который эта система получила за предыдущие  $t$  еди-

ниц времени,  $u(x) = \begin{cases} 1, x > 0 \\ 0, x \leq 0 \end{cases}$  – функция Хевисайда. Такие переходы находятся так же, как и в [5], но с учетом отрицательных заявок.

Возможные переходы между состояниями сети, их вероятности и доходы системы  $S_i$

Возможные переходы между состояниями сети	Вероятности переходов	Доходы системы $S_i$ от переходов между состояниями
$(k, t) \rightarrow (k, t + \Delta t)$	$1 - \sum_{j=1}^n [\lambda_{0j}^+ + (\lambda_{0j}^- + \mu_j)u(k_j)] \Delta t + o(\Delta t)$	$r_i(k)\Delta t + v_i(k, t)$
$(k, t) \rightarrow (k + I_j, t + \Delta t), j \neq i$	$\lambda_{0j}^+ \Delta t + o(\Delta t)$	$r_i(k)\Delta t + v_i(k + I_j, t)$
$(k, t) \rightarrow (k - I_j, t + \Delta t), j \neq i$	$\mu_j p_{j0} u(k_j) \Delta t + \lambda_{0j}^- u(k_j) \Delta t +$ $+ \mu_j \sum_{\substack{c=1 \\ c \neq i}}^n p_{jc}^- (1 - u(k_c)) \Delta t + o(\Delta t)$	$r_i(k)\Delta t + v_i(k - I_j, t)$
$(k, t) \rightarrow (k + I_i, t + \Delta t)$	$\lambda_{0i}^+ \Delta t + o(\Delta t)$	$r_{0i}(k + I_i, t) + v_i(k + I_i, t)$
$(k, t) \rightarrow (k - I_i, t + \Delta t)$	$\mu_i p_{i0} u(k_i) \Delta t + \lambda_{0i}^- u(k_i) \Delta t +$ $+ \mu_i \sum_{\substack{c=1 \\ c \neq i}}^n p_{ic}^- (1 - u(k_c)) \Delta t + o(\Delta t)$	$-R_{i0}(k - I_i, t) + v_i(k - I_i, t)$
$(k, t) \rightarrow (k + I_i - I_j, t + \Delta t),$ $j \neq i$	$\mu_j p_{ji}^+ u(k_j) \Delta t + o(\Delta t)$	$r_{ij}(k + I_i - I_j, t) +$ $+ v_i(k + I_i - I_j, t)$
$(k, t) \rightarrow (k - I_i + I_j, t + \Delta t),$ $j \neq i$	$\mu_i p_{ij}^+ u(k_i) \Delta t + o(\Delta t)$	$-r_{ji}(k - I_i + I_j, t) +$ $+ v_i(k - I_i + I_j, t)$
$(k, t) \rightarrow (k + I_c - I_s, t + \Delta t),$ $c, s \neq i$	$\mu_s p_{sc}^+ u(k_s) \Delta t + o(\Delta t)$	$r_i(k)\Delta t + v_i(k + I_c - I_s, t)$
$(k, t) \rightarrow (k - I_i - I_j, t + \Delta t),$ $j \neq i$	$\mu_i p_{ij}^- \Delta t + o(\Delta t)$	$-r_{ij}(k - I_i - I_j, t) +$ $+ v_i(k - I_i - I_j, t)$
$(k, t) \rightarrow (k - I_c - I_s, t + \Delta t),$ $c, s \neq i$	$\mu_c p_{cs}^- \Delta t + o(\Delta t)$	$r_i(k)\Delta t + v_i(k - I_c - I_s, t)$

Используя формулу полной вероятности для математического ожидания, получаем систему разностно-дифференциальных уравнений (РДУ) для дохода  $v_i(k, t)$ :

$$\begin{aligned} \frac{dv_i(k, t)}{dt} = & r_i(k) - \sum_{j=1}^n [\lambda_{0j}^+ + (\lambda_{0j}^- + \mu_j)u(k_j)] v_i(k, t) + \\ & + \sum_{j=1}^n \left( \lambda_{0j}^+ v_i(k + I_j, t) + \left[ \mu_j p_{j0} u(k_j) + \lambda_{0j}^- u(k_j) + \mu_j \sum_{\substack{c=1 \\ c \neq j}}^n p_{jc}^- (1 - u(k_c)) \right] v_i(k - I_j, t) \right) + \end{aligned}$$

$$\begin{aligned}
& + \lambda_{0i}^+ v_i(k + I_i, t) + \left[ \mu_i p_{i0} u(k_i) + \lambda_{0i}^- u(k_i) + \mu_i \sum_{\substack{c=1 \\ c \neq i}}^n p_{ic}^- (1 - u(k_c)) \right] v_i(k - I_i, t) + \\
& + \sum_{\substack{j=1 \\ j \neq i}}^n \left[ \mu_j p_{ji}^+ u(k_j) v_i(k + I_i - I_j, t) + \mu_i p_{ij}^+ u(k_i) v_i(k - I_i + I_j, t) + \mu_i p_{ij}^- v_i(k - I_i - I_j, t) \right] + \quad (2) \\
& + \sum_{\substack{c,s=1 \\ c,s \neq i, c \neq s}}^n \left[ \mu_s p_{sc}^+ u(k_s) v_i(k + I_c - I_s, t) + \mu_c p_{cs}^- v_i(k - I_c - I_s, t) \right] + \\
& + \lambda_{0i}^+ r_{0i}(k + I_i, t) - \left[ \mu_i p_{i0} u(k_i) + \lambda_{0i}^- u(k_i) + \mu_i \sum_{\substack{c=1 \\ c \neq i}}^n p_{ic}^- (1 - u(k_c)) \right] R_{i0}(k - I_i, t) + \\
& + \sum_{\substack{j=1 \\ j \neq i}}^n \left[ \mu_j p_{ji}^+ u(k_j) r_{ij}(k + I_i - I_j, t) - \mu_i p_{ij}^+ u(k_i) r_{ji}(k - I_i + I_j, t) - \mu_i p_{ij}^- r_{ij}(k - I_i - I_j, t) \right].
\end{aligned}$$

Число уравнений в этой системе равно числу состояний сети, для открытой сети оно является бесконечным. В случае когда доходы от переходов между состояниями сети не зависят от времени, для решения системы (2) могут быть использованы алгоритмы, аналогичные тем, которые были предложены для НМ-сетей с обычными заявками, и основанные на применении метода многомерных  $z$ -преобразований [5, 6] и метода последовательных приближений, совмещенного с методом рядов [5, 7].

### 3. Анализ сети, когда доходы от переходов между состояниями сети являются случайными величинами

Пусть случайная величина (СВ)  $\xi_i$  – время обслуживания заявки в системе  $S_i$ , распределенная по экспоненциальному закону с функцией распределения  $F_{\xi_i}(t) = 1 - e^{-\mu_i t}$ ,  $i = \overline{1, n}$ . Рассмотрим динамику изменения доходов системы  $S_i$  сети. Пусть в начальный момент времени доход этой системы был равен  $v_{i0}$ . Нас будет интересовать доход системы  $V_i(t)$  в момент времени  $t$ . Разобьем отрезок  $[0, t]$  на  $m$  равных частей длиной  $\Delta t = \frac{t}{m}$ , считая  $m$  достаточно большой. Для нахождения дохода системы  $S_i$  выпишем условные вероятности тех событий, которые могут произойти на  $l$ -м отрезке времени,  $l = \overline{1, m}$ . Возможны следующие ситуации:

1. С вероятностью  $\lambda_{0i}^+ \Delta t + o(\Delta t)$  в систему  $S_i$  поступит положительная заявка из внешней среды, которая принесет ей доход в размере  $r_{0i}$ , где  $r_{0i}$  – СВ с математическим ожиданием (м.о.)  $M\{r_{0i}\} = a_{0i}$ ,  $i = \overline{1, n}$ .

2. С вероятностью  $\lambda_{0i}^- \Delta t + o(\Delta t)$  в систему  $S_i$  поступит отрицательная заявка из внешней среды, которая принесет ей доход (убыток) в размере  $-\bar{r}_{0i}$ , где  $\bar{r}_{0i}$  – СВ с м.о.  $M\{\bar{r}_{0i}\} = \bar{a}_{0i}$ ,  $i = \overline{1, n}$ .

3. С вероятностью  $\mu_i p_{i0} u(k_i^{(l)}) \Delta t + o(\Delta t)$  заявка из системы  $S_i$  перейдет во внешнюю среду, при этом доход системы  $S_i$  уменьшится на величину  $R_{i0}$ , где  $R_{i0}$  – СВ с м.о.  $M\{R_{i0}\} = b_{i0}$ ,

$k_i^{(l)}$  – число заявок в системе  $S_i$  (в очереди и на обслуживании) на  $l$ -м отрезке времени,  $l = \overline{1, m}$ ,  $i = \overline{1, n}$ .

4. Положительная заявка из системы  $S_i$  перейдет в систему  $S_j$  с вероятностью  $\mu_i p_{ij}^+ u(k_i^{(l)}) \Delta t + o(\Delta t)$ ,  $i, j = \overline{1, n}$ ,  $i \neq j$ . При таком переходе доход системы  $S_i$  уменьшится на величину  $R_{ij}(\xi_i)$ , а доход системы  $S_j$  увеличится на эту величину:

$$M \{R_{ij}(\xi_i)\} = \int_0^{\infty} R_{ij}(t) dF_{\xi_i}(t) = \mu_i \int_0^{\infty} R_{ij}(t) e^{-\mu_i t} dt = a_{ij}, \quad i = \overline{1, n}, \quad j = \overline{1, n}, \quad i \neq j.$$

5. С вероятностью  $\mu_j p_{ji}^+ u(k_j) \Delta t + o(\Delta t)$  положительная заявка перейдет из системы  $S_j$  в систему  $S_i$ , при этом доход  $S_i$  возрастет на величину  $R_{ji}(\xi_j)$ , а доход системы  $S_j$  уменьшится на эту величину,  $M \{R_{ji}(\xi_j)\} = a_{ji}$ ,  $j = \overline{1, n}$ ,  $j \neq i$ .

6. Положительная заявка из системы  $S_i$  перейдет в систему  $S_j$  с вероятностью  $\mu_i p_{ij}^- \Delta t + o(\Delta t)$  как отрицательная заявка,  $i, j = \overline{1, n}$ ,  $i \neq j$ . При таком переходе доход системы  $S_i$  увеличится на величину  $\bar{R}_{ij}$ , где  $\bar{R}_{ij}$  – СВ с м.о.  $M \{\bar{R}_{ij}\} = c_{ij}$ ,  $i, j = \overline{1, n}$ ,  $i \neq j$ .

7. С вероятностью  $1 - \sum_{j=1}^n [\lambda_{0j}^+ + (\lambda_{0j}^- + \mu_j) u(k_j^{(l)})] \Delta t + o(\Delta t)$  на отрезке времени  $\Delta t$  состояние сети не изменится.

8. За каждый малый промежуток времени  $\Delta t$  система  $S_i$  из-за нахождения в ней заявок увеличивает свой доход на величину  $r_i \Delta t$ , где  $r_i$  – СВ с м.о.  $M \{r_i\} = d_i$ ,  $i = \overline{1, n}$ .

Пусть  $\Delta V_{il}(\Delta t)$  – изменение дохода системы  $S_i$  на  $l$ -м отрезке времени, связанное с переходами между СМО сети заявок. Из этого следует

$$\Delta V_{il}(\Delta t) = \begin{cases} r_{0i} + r_i \Delta t \text{ с вероятностью } \lambda_{0i}^+ \Delta t + o(\Delta t), \\ -\bar{r}_{0i} + r_i \Delta t \text{ с вероятностью } \lambda_{0i}^- \Delta t + o(\Delta t), \\ -R_{i0} + r_i \Delta t \text{ с вероятностью } \mu_i p_{i0} u(k_i^{(l)}) \Delta t + o(\Delta t), \\ -R_{ij}(\xi_i) + r_i \Delta t \text{ с вероятностью } \mu_i p_{ij}^+ u(k_i^{(l)}) \Delta t + o(\Delta t), \quad j = \overline{1, n}, \quad j \neq i, \\ R_{ji}(\xi_j) + r_i \Delta t \text{ с вероятностью } \mu_j p_{ji}^+ u(k_j^{(l)}) \Delta t + o(\Delta t), \quad j = \overline{1, n}, \quad j \neq i, \\ -\bar{R}_{ij} + r_i \Delta t \text{ с вероятностью } \mu_i p_{ij}^- \Delta t + o(\Delta t), \quad j = \overline{1, n}, \quad j \neq i, \\ r_i \Delta t \text{ с вероятностью } 1 - \sum_{j=1}^n [\lambda_{0j}^+ + (\lambda_{0j}^- + \mu_j) u(k_j^{(l)})] \Delta t + o(\Delta t). \end{cases} \quad (3)$$

Общий доход системы  $S_i$

$$V_i(t) = v_{i0} + \sum_{l=1}^m \Delta V_{il}(\Delta t) = v_{i0} + V_i(\Delta t),$$

где  $V_i(\Delta t) = \sum_{l=1}^m \Delta V_{il}(\Delta t)$ ,  $i = \overline{1, n}$ .

Найдем выражение для ожидаемого дохода системы  $S_i$  в момент времени  $t$ . Предположим, что все системы сети функционируют в режиме высокой нагрузки, т. е.  $k_i(t) > 0$ ,  $\forall t > 0$ ,  $i = \overline{1, n}$ . Тогда с учетом (3) для математического ожидания можно записать:

$$\begin{aligned} M\{\Delta V_{il}(\Delta t)\} &= (a_{0i} + d_i \Delta t) (\lambda_{0i}^+ \Delta t + o(\Delta t)) + (-\bar{a}_{0i} + d_i \Delta t) (\lambda_{0i}^- \Delta t + o(\Delta t)) + \\ &\quad + (-b_{i0} + d_i \Delta t) (\mu_i p_{i0} \Delta t + o(\Delta t)) + \\ &\quad + \sum_{j=1}^n [(-a_{ij} + d_i \Delta t) (\mu_i p_{ij}^+ \Delta t + o(\Delta t))] + \sum_{j=1}^n [(a_{ji} + d_i \Delta t) (\mu_j p_{ji}^+ \Delta t + o(\Delta t))] + \\ &\quad + \sum_{j=1}^n [(-c_{ij} + d_i \Delta t) (\mu_i p_{ij}^- \Delta t + o(\Delta t))] + d_i \Delta t \left( 1 - \sum_{j=1}^n [\lambda_{0j}^+ + (\lambda_{0j}^- + \mu_j)] \Delta t + o(\Delta t) \right) = \\ &= [a_{0i} \lambda_{0i}^+ - \bar{a}_{0i} \lambda_{0i}^- - b_{i0} \mu_i p_{i0}] \Delta t + \sum_{j=1}^n [-a_{ij} \mu_i p_{ij}^+ + a_{ji} \mu_j p_{ji}^+ - c_{ij} \mu_i p_{ij}^- + d_i] \Delta t + o(\Delta t), \quad i = \overline{1, n}. \end{aligned}$$

Поскольку  $m \Delta t = t$  и при  $m \rightarrow \infty$  величина  $\Delta t \rightarrow 0$ , получим

$$\begin{aligned} v_i(t) &= M\{V_i(t)\} = \sum_{l=1}^m M\{\Delta V_{il}(\Delta t)\} = v_{i0} + \\ &\quad + \left( a_{0i} \lambda_{0i}^+ - \bar{a}_{0i} \lambda_{0i}^- - \mu_i \sum_{j=1}^n c_{ij} p_{ij}^- + d_i - b_{i0} \mu_i p_{i0} - \sum_{j=1}^n a_{ij} \mu_i p_{ij}^+ + \sum_{j=1}^n a_{ji} \mu_j p_{ji}^+ \right) m \Delta t = \\ &= v_{i0} + \left[ \lambda_{0i}^+ a_{0i} - \lambda_{0i}^- \bar{a}_{0i} - \mu_i \left( b_{i0} p_{i0} + \sum_{j=1}^n (a_{ij} p_{ij}^+ + c_{ij} p_{ij}^-) \right) + \sum_{j=1}^n \mu_j a_{ji} p_{ji}^+ + d_i \right] t, \quad i = \overline{1, n}. \end{aligned}$$

Для сравнения соотношение для общего дохода системы  $S_i$  в случае, когда в сети нет отрицательных заявок и все системы сети функционируют в режиме высокой нагрузки, имеет вид [7]

$$\tilde{v}_i(t) = v_{i0} + \left[ \lambda a_{0i} \tilde{p}_{0i} - \mu_i b_{i0} \tilde{p}_{i0} + \sum_{j=1}^n \mu_j a_{ji} \tilde{p}_{ji} - \mu_i \sum_{j=1}^n a_{ij} \tilde{p}_{ij} + d_i \right] t, \quad i = \overline{1, n},$$

где  $\lambda$  – интенсивность поступающего в сеть простейшего потока заявок;  $\tilde{p}_{0i}$  – вероятность поступления заявки в систему  $S_i$ ,  $i = \overline{1, n}$ ,  $\sum_{i=1}^n \tilde{p}_{0i} = 1$ ;  $\tilde{p}_{ij}$  – вероятность того, что заявка, завер-

шившая обслуживание в  $i$ -й СМО, направляется в  $j$ -ю СМО,  $\sum_{j=0}^n \tilde{p}_{ij} = 1$ ,  $i = \overline{1, n}$ ;  $\tilde{p}_{i0}$  – вероят-

ность, с которой заявка покидает сеть,  $\tilde{p}_{i0} = 1 - \sum_{j=1}^n \tilde{p}_{ij}$ ,  $i = \overline{1, n}$ .

Найдем разность  $\tilde{v}_i(t) - v_i(t)$ . Заметим, что  $p_{ij}^+ = \tilde{p}_{ij}$  и  $\lambda_{0i}^+ = \lambda \tilde{p}_{0i}$ ,  $i, j = \overline{1, n}$ . С учетом (1) получим

$$\tilde{v}_i(t) - v_i(t) = v_{i0} + \left[ \lambda a_{0i} \tilde{p}_{0i} - \mu_i b_{i0} \tilde{p}_{i0} + \sum_{j=1}^n \mu_j a_{ji} \tilde{p}_{ji} - \mu_i \sum_{j=1}^n a_{ij} \tilde{p}_{ij} + d_i \right] t -$$

$$\begin{aligned}
 & -v_{i0} - \left[ \lambda_{0i}^+ a_{0i} - \lambda_{0i}^- \bar{a}_{0i} - \mu_i \left( b_{i0} p_{i0} + \sum_{j=1}^n (a_{ij} p_{ij}^+ + c_{ij} p_{ij}^-) \right) + \sum_{j=1}^n \mu_j a_{ji} p_{ji}^+ + d_i \right] t = \\
 & = \left[ \lambda_{0i}^- \bar{a}_{0i} + \mu_i \sum_{j=1}^n c_{ij} p_{ij}^- - \mu_i b_{i0} \tilde{p}_{i0} + \mu_i b_{i0} p_{i0} \right] t = \\
 & = \left[ \lambda_{0i}^- \bar{a}_{0i} + \mu_i \sum_{j=1}^n c_{ij} p_{ij}^- - \mu_i b_{i0} \left( 1 - \sum_{j=1}^n \tilde{p}_{ij} \right) + \mu_i b_{i0} \left( 1 - \sum_{j=1}^n (p_{ij}^+ + p_{ij}^-) \right) \right] t = \\
 & = \left[ \lambda_{0i}^- \bar{a}_{0i} + \mu_i \sum_{j=1}^n p_{ij}^- (c_{ij} - b_{i0}) \right] t \quad i = \overline{1, n}.
 \end{aligned}$$

**Пример.** Пусть количество СМО в сети  $n = 10$ . Интенсивности входного потока положительных и отрицательных заявок  $\lambda_{0i}^+$  и  $\lambda_{0i}^-$  равны соответственно  $\lambda_{01}^+ = 2$ ,  $\lambda_{04}^+ = 4$ ,  $\lambda_{07}^+ = 3$ ,  $\lambda_{01}^- = 1$ ,  $\lambda_{04}^- = 2$ ,  $\lambda_{05}^- = 3$ , остальные равны нулю. Интенсивности обслуживания заявок  $\mu_i$  равны  $\mu_1 = \mu_2 = \mu_3 = 2$ ,  $\mu_4 = 1$ ,  $\mu_5 = 3$ ,  $\mu_6 = 4$ ,  $\mu_7 = 3$ ,  $\mu_8 = 13$ ,  $\mu_9 = 7$ ,  $\mu_{10} = 8$ . Пусть вероятности  $p_{ij}^+$  равны соответственно  $p_{12}^+ = 1/8$ ,  $p_{13}^+ = 1/8$ ,  $p_{21}^+ = p_{23}^+ = p_{24}^+ = p_{25}^+ = 1/10$ ,  $p_{31}^+ = p_{32}^+ = p_{36}^+ = p_{37}^+ = 1/10$ ,  $p_{42}^+ = p_{45}^+ = p_{48}^+ = 1/8$ ,  $p_{52}^+ = p_{54}^+ = p_{56}^+ = p_{58}^+ = 1/8$ ,  $p_{63}^+ = p_{65}^+ = p_{67}^+ = p_{69}^+ = 1/8$ ,  $p_{73}^+ = p_{76}^+ = p_{79}^+ = 1/8$ ,  $p_{84}^+ = p_{85}^+ = p_{89}^+ = p_{8,10}^+ = 1/8$ ,  $p_{96}^+ = p_{97}^+ = p_{98}^+ = p_{9,10}^+ = 1/8$ ,  $p_{10,8}^+ = p_{10,9}^+ = 1/5$ , остальные равны нулю. Вероятности того, что положительные заявки, обслуженные в СМО  $S_i$ , направляются в СМО  $S_j$  как отрицательные заявки, равны  $p_{12}^- = 1/9$ ,  $p_{13}^- = 1/9$ ,  $p_{21}^- = 1/11$ ,  $p_{23}^- = 1/11$ ,  $p_{24}^- = 1/11$ ,  $p_{25}^- = 1/11$ ,  $p_{31}^- = 1/11$ ,  $p_{32}^- = 1/11$ ,  $p_{36}^- = 1/11$ ,  $p_{37}^- = 1/11$ ,  $p_{42}^- = p_{45}^- = p_{48}^- = 1/9$ ,  $p_{52}^- = p_{54}^- = p_{56}^- = p_{58}^- = 1/9$ ,  $p_{63}^- = 1/9$ ,  $p_{65}^- = 1/9$ ,  $p_{67}^- = 1/9$ ,  $p_{69}^- = 1/9$ ,  $p_{73}^- = p_{76}^- = p_{79}^- = 1/9$ ,  $p_{84}^- = p_{85}^- = p_{89}^- = p_{8,10}^- = 1/9$ ,  $p_{96}^- = p_{97}^- = p_{98}^- = p_{9,10}^- = 1/9$ ,  $p_{10,8}^- = p_{10,9}^- = 1/6$ , остальные равны нулю. Вероятности выхода заявок из сети во внешнюю среду  $p_{40} = p_{70} = 7/24$ ,  $p_{10,0} = 4/15$ . Математические ожидания  $a_{01} = 10\,000$ ,  $a_{02} = 20\,000$ ,  $a_{03} = 30\,000$ ,  $a_{04} = 50\,000$ ,  $a_{05} = a_{06} = 10\,000$ ,  $a_{07} = 40\,000$ ,  $a_{08} = 30\,000$ ,  $a_{09} = 50\,000$ ,  $a_{010} = 10\,000$ ;  $\bar{a}_{01} = 50\,000$ ,  $\bar{a}_{02} = 60\,000$ ,  $\bar{a}_{03} = 10\,000$ ,  $\bar{a}_{04} = 25\,000$ ,  $\bar{a}_{05} = 50\,000$ ,  $\bar{a}_{06} = 7\,000$ ,  $\bar{a}_{07} = 3\,000$ ,  $\bar{a}_{08} = 2\,500$ ,  $\bar{a}_{09} = 1\,200$ ,  $\bar{a}_{010} = 9\,000$ ;  $b_{10} = 1\,000$ ,  $b_{20} = 2\,000$ ,  $b_{30} = 3\,000$ ,  $b_{40} = 5\,000$ ,  $b_{50} = b_{60} = b_{70} = 1\,000$ ,  $b_{80} = 3\,000$ ,  $b_{90} = 5\,000$ ,  $b_{10,0} = 1\,000$ ;  $c_{12} = 1\,000$ ,  $c_{13} = 1\,500$ ,  $c_{21} = c_{23} = c_{24} = c_{25} = 2\,000$ ,  $c_{31} = c_{32} = c_{36} = c_{37} = 1\,000$ ,  $c_{42} = c_{45} = c_{48} = 2\,000$ ,  $c_{52} = c_{54} = c_{56} = c_{58} = 1\,500$ ,  $c_{63} = c_{65} = c_{67} = c_{69} = 1\,700$ ,  $c_{73} = c_{76} = c_{79} = 2\,300$ ,  $c_{84} = 2\,000$ ,  $c_{85} = 2\,000$ ,  $c_{89} = 2\,000$ ,  $c_{8,10} = 2\,000$ ,  $c_{96} = c_{97} = c_{98} = c_{9,10} = 3\,000$ ,  $c_{10,8} = c_{10,9} = 1\,300$ ,  $d_1 = 100$ ,  $d_2 = 200$ ,  $d_3 = 300$ ,  $d_4 = 120$ ,  $d_5 = 200$ ,  $d_6 = 100$ ,  $d_7 = 800$ ,  $d_8 = 100$ ,  $d_9 = 120$ ,  $d_{10} = 300$ , остальные равны нулю. Случайные доходы системы  $S_i$  имеют вид

$$\begin{aligned}
 & R_{12}(\xi_1) = 300\xi_1, \quad R_{13}(\xi_1) = 0.1 + 100\xi_1, \\
 & R_{21}(\xi_2) = 2500 + \xi_2, \quad R_{23}(\xi_2) = 100\xi_2, \quad R_{24}(\xi_2) = 200\xi_2, \quad R_{25}(\xi_2) = 2000\xi_2, \\
 & R_{31}(\xi_3) = 0.5\xi_3 + 1000, \quad R_{32}(\xi_3) = 0.1\xi_3 + 10000, \quad R_{36}(\xi_3) = 0.5\xi_3 + 450, \quad R_{37}(\xi_3) = 0.5\xi_3 + 5000, \\
 & R_{42}(\xi_4) = 10\xi_4 - 100, \quad R_{45}(\xi_4) = (\xi_4 + 100)\xi_4, \quad R_{48}(\xi_4) = 2000\xi_4,
 \end{aligned}$$

$$\begin{aligned}
R_{52}(\xi_5) &= 1000\xi_5 - 10, \quad R_{54}(\xi_5) = 1000\xi_5 + 100, \quad R_{56}(\xi_5) = 1000\xi_5 + 300, \quad R_{58}(\xi_5) = 1000\xi_5, \\
R_{63}(\xi_6) &= \xi_6 + 1000, \quad R_{65}(\xi_5) = 5\xi_5 + 1000, \quad R_{67}(\xi_6) = 3\xi_6 + 1000, \quad R_{69}(\xi_6) = 2\xi_6 + 1000, \\
R_{73}(\xi_7) &= 1000\xi_7 - 100, \quad R_{76}(\xi_7) = 1000\xi_7, \quad R_{79}(\xi_7) = 100\xi_7 + 300, \\
R_{84}(\xi_8) &= \xi_8(\xi_8 + 1000), \quad R_{85}(\xi_8) = \xi_8(\xi_8 + 100), \quad R_{89}(\xi_8) = \xi_8(\xi_8 - 10), \quad R_{8,10}(\xi_8) = \xi_8(\xi_8 + 200), \\
R_{96}(\xi_9) &= \xi_9 + 1000, \quad R_{97}(\xi_9) = 3\xi_9 + 1000, \quad R_{98}(\xi_9) = 10\xi_9 + 1000, \quad R_{9,10}(\xi_9) = \xi_9 - 100, \\
R_{10,8}(\xi_{10}) &= 3000\xi_{10}, \quad R_{10,9}(\xi_{10}) = 3000\xi_{10} - 100.
\end{aligned}$$

Математические ожидания этих случайных доходов были рассчитаны в пакете Mathematica и равны соответственно

$$\begin{aligned}
a_{12} &= 144000e^{-2t}, \quad a_{13} = 48e^{-2t}(0,1 + 1000t), \\
a_{21} &= 48e^{-2t}(2500 + t), \quad a_{23} = 4800e^{-2t}t, \quad a_{24} = 9600e^{-2t}t, \quad a_{245} = 96000e^{-2t}t, \\
a_{31} &= 48e^{-2t}(1000 + 0,5t), \quad a_{32} = 48e^{-2t}(10\,000 + 0,1t), \quad a_{36} = 48e^{-2t}(450 + 0,5t), \quad a_{37} = 48e^{-2t}(5000 + 0,5t), \\
a_{42} &= 24e^{-t}(10t - 100), \quad a_{45} = 24e^{-t}(t + 100), \quad a_{48} = 48\,000te^{-t}, \\
a_{52} &= 72e^{-3t}(1000t - 10), \quad a_{54} = 72e^{-3t}(1000t + 100), \quad a_{56} = 72e^{-3t}(1000t + 300), \quad a_{54} = 72\,000e^{-3t}t, \\
a_{63} &= 96e^{-4t}(t + 1000), \quad a_{65} = 96e^{-4t}(5t + 1000), \quad a_{67} = 96e^{-4t}(3t + 1000), \quad a_{69} = 96e^{-4t}(2t + 1000), \\
a_{73} &= 72e^{-3t}(-100 + 1000t), \quad a_{76} = 72\,000te^{-3t}, \quad a_{79} = 72e^{-3t}(300 + 100t), \\
a_{84} &= 312e^{-13t}(1000 + t), \quad a_{85} = 312e^{-13t}(100 + t), \quad a_{89} = 312e^{-13t}(-10 + t)t, \quad a_{8,10} = 312e^{-13t}(200 + t), \\
a_{96} &= 168e^{-7t}(1000 + t), \quad a_{97} = 168e^{-7t}(1000 + 3t), \quad a_{98} = 168e^{-7t}(1000 + 10t), \quad a_{9,10} = 168e^{-7t}(t - 100), \\
a_{10,8} &= 576\,000e^{-8t}t, \quad a_{10,9} = 192e^{-8t}(3000 - 100t).
\end{aligned}$$

Доход в начальный момент времени  $v_{i_0} = 0$ ,  $i = \overline{1, n}$ . Рассмотрим временной интервал длиной в 24 ч,  $t \in [0, T]$ ,  $T = 24$ .

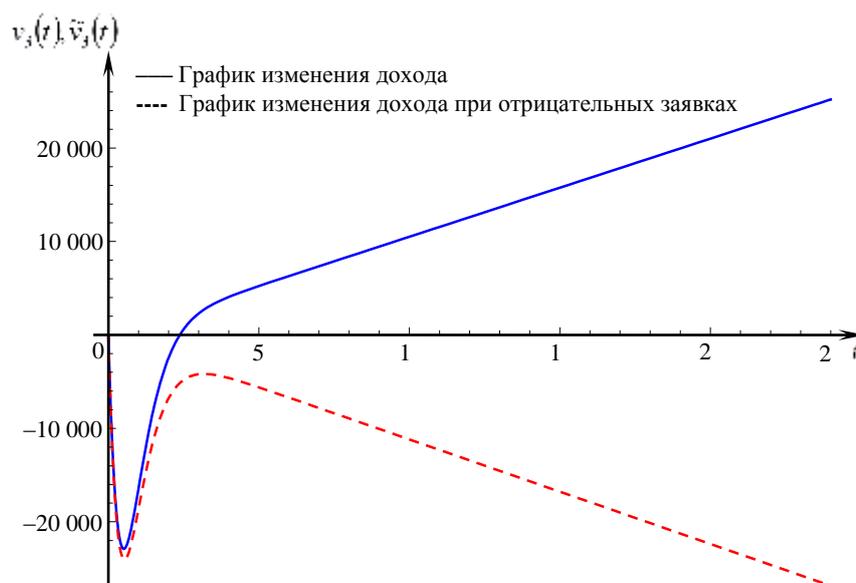
Нахождение ожидаемых доходов систем сети было реализовано в виде программы для пакета математических вычислений Mathematica. Были получены выражения для ожидаемых доходов. Например, выражение для ожидаемого дохода системы  $S_3$  имеет вид

$$\tilde{v}_3(t) = 0,2e^{-4t} \left[ e^t(135\,000t - 13\,500) + e^{2t}(64\,723,2t - 789\,594) + 5250e^{4t} + 240t + 240\,000 \right],$$

а в случае, когда в сети циркулируют отрицательные заявки, выражение для ожидаемого дохода системы  $S_3$  примет вид

$$v_3(t) = e^{-4t} \left[ 27e^t(1000t - 100) + e^{2t}(12\,975,4t - 157\,438,8) - \frac{61\,504}{55}e^{4t} + 48t + 48\,000 \right].$$

На рисунке показано изменение доходов системы  $S_3$  для НМ-сети с отрицательными заявками и без них. Видно, что отрицательные заявки не только уменьшают ожидаемый доход системы  $S_3$ , но при этом он может становиться вообще отрицательным.

Изменение доходов системы  $S_3$ 

### Заключение

В работе предложена методика нахождения ожидаемых доходов в системах НМ-сети МО с однолинейными СМО, положительными и отрицательными заявками в случаях, когда доходы от переходов между состояниями сети являются детерминированными функциями, зависящими от состояний сети и времени, либо являются СВ с известными средними. Дальнейшие исследования в этом направлении связаны с получением аналогичных результатов для сетей с многолинейными СМО.

### Список литературы

1. Gelenbe, E. Product form queueing networks with negative and positive customers / E. Gelenbe // J. of Appl. Prob. – 1991. – Vol. 28. – P. 656–663.
2. Бочаров, П.П. G-сети: развитие теории мультипликативных сетей / П.П. Бочаров, В.М. Вишневецкий // Автоматика и телемеханика. – 2003. – № 5. – С. 46–74.
3. Gelenbe, E. Stability of product-form G-networks / E. Gelenbe, R. Schassberger // Probab. Eng. and Inf. Sci. – 1992. – Vol. 6. – P. 271–276.
4. Матальцкий, М.А. О применении сетей обслуживания с доходами при решении задач транспортной логистики / М.А. Матальцкий, О.М. Китурко // Информатика. – 2010. – № 4. – С. 81–95.
5. Матальцкий, М.А. О некоторых результатах анализа и оптимизации марковских сетей с доходами и их применении / М.А. Матальцкий // Автоматика и телемеханика. – 2009. – № 10. – С. 97–113.
6. Колузаева, Е.В. Нахождение ожидаемых доходов в открытой двухузловой НМ-сети с помощью z-преобразований / Е.В. Колузаева // Вестник ГрГУ. Сер. 2. – 2009. – № 3. – С. 10–17.
7. Матальцкий, М.А. Системы и сети МО: анализ и применения / М.А. Матальцкий, О.М. Тихоненко, Е.В. Колузаева. – Гродно : ГрГУ, 2011. – 817 с.

Поступила 04.11.13

Гродненский государственный  
университет им. Я. Купалы,  
Гродно, ул. Ожешко, 22  
e-mail: m.matalytski@gmail.com  
victornn86@gmail.com

**V.V. Naumenko, M.A. Matalytski**

**ANALYSIS OF MARKOV NETWORK WITH INCOMES,  
POSITIVE AND NEGATIVE MESSAGES**

Markov queuing network with income in transient regime is considered. It has positive and negative messages, which can be used in forecasting income of information and telecommunication systems and networks affected by viruses. Investigations are carried out in the cases when incomes from transitions between network states are deterministic functions dependent on states, or they are random variables with given mean values. In the last case it is assumed that all network systems operate in a high load mode. An example is given.

УДК 519.95

Ф. Вернер<sup>1</sup>, С.А. Кравченко<sup>2</sup>, К. Хасани<sup>3</sup>МИНИМИЗАЦИЯ СУММАРНОГО ВРЕМЕНИ ОБСЛУЖИВАНИЯ  
ДЛЯ СИСТЕМЫ С ДВУМЯ ПРИБОРАМИ И ОДНИМ СЕРВЕРОМ

*Рассматривается задача минимизации суммарного времени обслуживания множества требований на множестве двух идентичных параллельных приборов. Перед обслуживанием требования необходима загрузка, которая осуществляется сервером. Известно, что задача NP-трудна в сильном смысле. В работе предлагаются две модели целочисленного линейного программирования и алгоритм имитации отжига (simulated annealing algorithm). Предложенные подходы тестируются на примерах задач, содержащих до 250 требований.*

## Введение

В работе рассматривается задача оптимального обслуживания  $n$  требований на двух параллельных идентичных приборах. Для каждого требования  $j$ ,  $j = 1, \dots, n$ , известно время обслуживания  $p_j$ . Обозначим через  $p_{\max}$  максимальное значение времени обслуживания, т. е.  $p_{\max} = \max\{p_j \mid j = 1, \dots, n\}$ . Перед началом обслуживания требование необходимо загрузить на прибор  $M_q$ ,  $q = 1, 2$ , при помощи сервера (робота). Для загрузки требуется время  $s_j$ , и загрузка может быть выполнена только в случае, если имеются свободный сервер и свободный прибор. Если же в текущий момент все приборы или сервер заняты, то загрузка нового требования не может быть начата. Предполагается, что время перевода сервера от одного прибора к другому пренебрежимо мало. Во время загрузки прибор  $M_q$  так же, как и сервер, задействован в течение времени  $s_j$ . Сервер может грузить не более одного требования одновременно. Необходимо построить расписание, минимизирующее суммарное время обслуживания всех требований. С использованием общепринятых обозначений задача может быть записана как  $P2, S1 \parallel \sum C_j$ . Данная задача является NP-трудной в сильном смысле, поскольку известно, что задача  $P2, S1 \mid s_j = s \mid \sum C_j$  тоже является NP-трудной в сильном смысле [1]. Известна вычислительная сложность для некоторых частных случаев рассматриваемой задачи. Задача  $P2, S1 \mid p_j = p \mid \sum C_j$  является NP-трудной в обычном смысле [2]. Известен полиномиальный алгоритм сложности  $O(n \log n)$  для задачи  $P2, S1 \mid s_j = 1 \mid \sum C_j$  [1]. Для задачи  $P3, S1 \mid s_j = 1 \mid \sum C_j$  в [2] разработан полиномиальный алгоритм сложности  $O(n^7)$ . Для задачи  $P, S1 \mid s_j = 1 \mid \sum C_j$  в [3] предложен алгоритм, который строит расписание с абсолютной погрешностью  $\sum_{i=1}^n \tilde{C}_i - \sum_{i=1}^n C_i^* \leq n'(m-2)$ , где  $m$  – число приборов,  $n' = |\{i \mid p_i < m-1\}|$ ,  $C_i^*$  обозначает время завершения обслуживания требования  $i$  в оптимальном расписании,  $\tilde{C}_i$  обозначает время завершения обслуживания требования  $i$  в построенном расписании. В работе [4] предложен приближенный алгоритм для задачи  $P, S1 \parallel \sum w_j C_j$  с отношением (оценкой)  $\Delta = \sum w_j \tilde{C}_j / \sum w_j C_j^*$ ,  $\Delta \leq (5 - \frac{1}{m})$  и там же показано, что правило SPT позволяет строить расписания для задачи  $P, S1 \mid s_j = s \mid \sum C_j$  с гарантированной оценкой  $\Delta \leq 3/2$ . Авторам не известны публикации с эвристическими алгоритмами для задачи  $P2, S1 \parallel \sum C_j$ , однако известны результаты разработки эвристик для близких моделей. В работе [5] рассматривается задача построения расписания для обслуживания множества требований на множестве приборов разной производительности. При этом время загрузки требований зависит от порядка обслуживания (sequence dependent setup) и критерием является минимизация взвешенной суммы моментов завершения обслужи-

вания требований. Несколько алгоритмов было предложено и протестировано. Максимальное количество требований и приборов, для которых применялись эвристики, равны соответственно 120 и 12. В работе [6] рассматривалась задача с идентичными приборами и требованиями, разделенными на семейства. Сервер использовался только в случае, если прибор, окончив обработку требования из одного семейства, должен обрабатывать требование из другого семейства. Критерием была минимизация взвешенной суммы моментов завершения обслуживания требований. Предложенные эвристики тестировались на примерах, в которых число семейств не превышало 8, количество требований не превышало 25 и количество приборов не превышало 5. Аналогичная предыдущей задаче рассматривалась в работе [7], где было предложено несколько алгоритмов типа ветвей и границ. Алгоритмы тестировались на примерах с максимальным количеством требований 25, максимальным количеством семейств 8 и с количеством приборов, не превышающим 5. В работе [8] рассматривалась задача с критерием  $\sum C_j$  и одним сервером, однако не допускалось одновременное использование сервера и прибора при загрузке. Таким образом, задача моделировалась как двухстадийная задача гибридного типа с одинаковыми маршрутами обслуживания требований. При этом ставилось условие отсутствия задержек между стадиями при обслуживании. Для задачи была предложена математическая модель и рассматривались некоторые полиномиально разрешимые частные случаи.

В данной работе исследуется задача  $P2, S1 \parallel \sum C_j$  и предлагаются две модели целочисленного линейного программирования и один эвристический алгоритм имитации отжига. Тесты выполняются для примеров с количеством требований, не превышающим 250.

### 1. Моделирование задачи $P2, S1 \parallel \sum C_j$

Поскольку для построения расписания достаточно знать порядок загрузки требований, оптимальное расписание для задачи  $P2, S1 \parallel \sum C_j$  можно искать в классе списочных расписаний, т. е. все требования выполняются в порядке очереди, заданной списком. При этом очередное требование обслуживается таким образом, чтобы время его завершения было минимальным.

**Пример.** Рассмотрим пример с пятью требованиями:  $s_1 = 2, p_1 = 4, s_2 = 2, p_2 = 3, s_3 = 1, p_3 = 5, s_4 = 2, p_4 = 4, s_5 = 1, p_5 = 2$ . Предположим, где требования обслуживаются в соответствии со списком  $\pi_0 = (3, 1, 4, 2, 5)$ .

Для данного случая требования 3–5 обслуживаются на приборе  $M_1$ , а требования 1, 2 – на приборе  $M_2$ . Суммарное время обслуживания равно 53.

Обозначим  $L_j = s_j + p_j, j = 1, \dots, n$ . Везде далее предполагаем, что требования пронумерованы таким образом, что справедливо  $L_1 \leq \dots \leq L_n$ .

Утверждение. Для расписания, построенного в соответствии со списком  $(1, \dots, n)$ , где выполняется  $L_1 \leq \dots \leq L_n$ , справедливо неравенство  $\frac{\sum \tilde{C}_j}{\sum C_j^*} \leq 2$ .

Здесь  $\sum \tilde{C}_j$  обозначает сумму моментов завершения обслуживания требований для расписания, соответствующего списку  $(1, \dots, n)$ .

Доказательство. Легко видеть, что справедливо

$$\tilde{C}_j \leq \sum_{k=1}^{j-1} s_k + \frac{1}{2} \sum_{k=1}^{j-1} p_k + L_j = \frac{1}{2} \sum_{k=1}^{j-1} s_k + \frac{1}{2} \sum_{k=1}^{j-1} L_k + L_j.$$

Поскольку  $\frac{1}{2} \sum_{k=1}^{j-1} s_k + \frac{1}{2} L_j \leq \frac{1}{2} \sum_{k=1}^j L_k \leq C_{[j]}^*$  и  $\frac{1}{2} \sum_{k=1}^{j-1} L_k + \frac{1}{2} L_j \leq C_{[j]}^*$ , получаем  $\tilde{C}_j \leq 2C_{[j]}^*$ . Здесь  $C_{[j]}^*$  обозначает время завершения обслуживания  $j$ -го (по порядку завершения обслуживания)

требования в оптимальном расписании. При этом используется неравенство  $\frac{1}{2} \sum_{k=1}^j L_k \leq C_{[j]}^*$ , которое следует непосредственно из  $L_1 \leq \dots \leq L_n$ . ■

В дальнейшем используются две нижние границы для оценки  $\sum C_j$ . Первая нижняя граница  $LB_1$  вычисляется по правилу  $\sum_j C_j^1 = LB_1$ , где  $C_j^1 = L_j + L_{j-2} + \dots + L_{j-2n}$ , и все  $L_k$  с  $k \leq 0$  полагаем равными нулю. Вторая нижняя граница  $LB_2$  вычисляется по правилу  $\sum_j C_j^2 = LB_2$ , где  $C_j^2 = L_j + ss_{j-1} + ss_{j-2} + \dots + ss_1$  и  $ss_1, \dots, ss_n$  – времена загрузки, перенумерованные в порядке убывания их значений, т. е.  $ss_1 \leq \dots \leq ss_n$ .

### 1.1. Модель M1

Модель M1 предлагается для нахождения оптимального расписания для задачи  $P2, S1 \parallel \sum C_j$ , причем оптимальное расписание отыскивается в классе списочных расписаний. Предположим, что имеется расписание  $s$ , соответствующее списку  $\pi$ . Введем переменную  $ft_j$  – время завершения обслуживания требования  $j$  и переменную  $st_j$  – момент времени, соответствующий началу загрузки требования  $j$ . Кроме того, пусть  $mt_j$  обозначает момент времени, соответствующий окончанию загрузки требования  $j$ . Обозначим  $O_{i,j}$  бинарную переменную, определяющую порядок требований  $i$  и  $j$  в списке  $\pi$ , т. е.  $O_{i,j} = 1$ , если  $st_i \leq st_j$ , а иначе  $O_{i,j} = 0$ . Для бинарной переменной  $p_{i,j}$  полагаем  $p_{i,j} = 1$  в случае  $mt_i \leq st_j \leq ft_i$ , т. е. если загрузка требования  $j$  начинается после загрузки требования  $i$ , но до завершения его обслуживания. Минимизация  $\sum C_j$  эквивалентна минимизации  $\sum ft_j$ . Поскольку  $O_{i,j}$  определяет порядок требований  $i$  и  $j$  в списке  $\pi$ , равенство  $O_{i,j} + O_{j,i} = 1$  должно выполняться. Неравенства  $ft_j - st_j \geq s_j + p_j$  и  $mt_j - st_j \geq s_j$  должны выполняться по определению. Поскольку имеется только два прибора, выполняется неравенство  $\sum_{i=1}^n p_{i,j} \leq 1$ .

В дальнейшем нам понадобится некоторое большое число  $P$ . Для ускорения работы с моделью удобно взять  $P = \frac{\sum_{i=1}^n p_i - p_{\max}}{2} + \sum_{i=1}^n s_i + p_{\max}$ . При увеличении значения  $P$  время поиска оптимального решения увеличивается. Если требование  $j$  предшествует требованию  $i$  в списке  $\pi$ , выполняется  $st_j - mt_i \geq -P$ . Однако если требование  $j$  следует после требования  $i$  в списке  $\pi$ , выполняется  $st_j - mt_i \geq (1 - p_{i,j})p_i$ ; таким образом, если требование  $j$  не обслуживается одновременно с требованием  $i$ , выполняется  $st_j - mt_i \geq p_i$ . В итоге получаем неравенство  $st_j - mt_i \geq (O_{i,j} - p_{i,j})p_i - O_{j,i}P$ .

Итак, получаем модель

$$\sum_{j=1}^n ft_j \rightarrow \min;$$

$$O_{i,j} + O_{j,i} = 1 \text{ для } i \neq j, i = 1, \dots, n, j = 1, \dots, n;$$

$$ft_j - st_j \geq s_j + p_j \text{ для } j = 1, \dots, n;$$

$$mt_j - st_j \geq s_j \text{ для } j=1, \dots, n;$$

$$\sum_{i=1}^n p_{i,j} \leq 1 \text{ для } j=1, \dots, n;$$

$$st_j - mt_i \geq (O_{i,j} - p_{i,j})p_i - O_{j,i}P \text{ для } i=1, \dots, n, j=1, \dots, n;$$

$$\sum_{j=1}^n ft_j \geq LB.$$

Переменные  $O_{ij}$  и  $p_{ij}$  являются бинарными; переменные  $ft_i$ ,  $st_i$  и  $mt_i$  являются положительными действительными числами; числа  $p_j$  и  $s_j$  известны заранее,  $i = 1, \dots, n$  и  $j = 1, \dots, n$ ,  $LB = \max\{LB_1, LB_2\}$ .

### 1.2. Модель M2

Модель M2 предлагается для поиска оптимального расписания в узком подмножестве, а именно когда требования из списка  $\pi$  должны обслуживаться с обязательной заменой прибора, т. е. рядом стоящие в списке  $\pi$  требования обслуживаются на разных приборах. Очевидно, что модель M2 не строит оптимальное расписание для задачи  $P2, S1 \parallel \sum C_j$ , но расписания, генерируемые такой моделью, оказываются близки к оптимальному расписанию. Опишем модель M2. Предположим, что расписание  $s$  задано списком  $\pi$ . Переменная  $N_{ij}$  определяет расположение требования  $j$  в списке  $\pi$ , т. е.  $N_{ij} = 1$ , если требование  $j$  находится в позиции  $i$ , а иначе  $N_{ij} = 0$ . Переменная  $ft_i$  обозначает время завершения обслуживания  $i$ -го требования из списка  $\pi$ , переменная  $st_i$  – время начала загрузки  $i$ -го требования из списка  $\pi$ , переменная  $mt_i$  – время завершения загрузки  $i$ -го требования из списка  $\pi$ . Минимизация  $\sum C_j$  соответствует минимизации  $\sum ft_j$ . Поскольку каждой позиции из списка  $\pi$  соответствует только одно требование, должно выполняться равенство  $\sum_j N_{i,j} = 1$ . Поскольку каждое требование занимает некую позицию в списке  $\pi$ , выполняется  $\sum_i N_{i,j} = 1$ . Неравенство  $ft_i - st_i \geq \sum_j N_{i,j}(s_j + p_j)$  выполняется, поскольку между моментом  $st_i$  и моментом  $ft_i$  должно обслужиться  $i$ -е требование из списка  $\pi$ . Аналогично, неравенство  $mt_i - st_i \geq \sum_j N_{i,j}s_j$  справедливо, поскольку между моментами времени  $st_i$  и  $mt_i$  должно загрузиться  $i$ -е требование из списка  $\pi$ . Неравенство  $st_i \geq mt_{i-1}$  выполняется, так как список  $\pi$  определяет порядок загрузки требований. Из-за чередования приборов должно выполняться неравенство  $st_i \geq ft_{i-2}$ .

Итак, модель M2 можно описать следующим образом:

$$\sum_{i=1}^n ft_i \rightarrow \min;$$

$$\sum_{i=1}^n N_{i,j} = 1 \text{ для } j=1, \dots, n;$$

$$\sum_{j=1}^n N_{i,j} = 1 \text{ для } i=1, \dots, n;$$

$$ft_i - st_i \geq \sum_{j=1}^n N_{i,j}(s_j + p_j) \text{ для } i=1, \dots, n;$$

$$mt_i - st_i \geq \sum_{j=1}^n N_{i,j}s_j \text{ для } i=1, \dots, n;$$

$$st_i \geq mt_{i-1} \text{ для } i = 2, \dots, n;$$

$$st_i \geq ft_{i-2} \text{ для } i = 3, \dots, n.$$

Переменные  $N_{ij}$  являются бинарными;  $ft_i$ ,  $st_i$  и  $mt_i$  – положительные переменные;  $i = 1, \dots, n$  и  $j = 1, \dots, n$ .

## 2. Алгоритм имитации отжига

Опишем алгоритм имитации отжига (SA) для задачи  $P2, S1 \parallel \sum C_j$ . На каждой итерации для заданного расписания случайным образом генерируется соседнее. В случае если значение критерия для сгенерированного соседа оказывается меньше заданного, соседнее расписание принимается за новое заданное расписание. В противном случае соседнее расписание принимается за новое заданное расписание с некоторой вероятностью.

В алгоритме SA для генерации соседних расписаний используются следующие операции:

*Перестановка*: переставляются два случайно выбранные требования. Пусть для  $\pi_0 = (3, 1, 4, 2, 5)$  из примера (см. разд. 1) две случайно выбранные позиции будут  $a = 3$  и  $b = 5$ ; применяя перестановку, получаем  $Swap(\pi_0, a, b) = Swap(\pi_0, 3, 5) = (3, 1, 5, 2, 4)$ .

*Перестановка соседних позиций*: переставляются два случайно выбранные соседние требования. Пусть для  $\pi_0 = (3, 1, 4, 2, 5)$  две выбранные позиции будут  $a = 3$  и  $b = 4$ . Получим  $SwapAdj(\pi_0, a, b) = SwapAdj(\pi_0, 3, 4) = (3, 1, 2, 4, 5)$ .

*Перестановка блока*: случайно выбранный блок, состоящий из  $\ell$  требований, переставляется с другим случайно выбранным блоком, содержащим  $\ell$  требований. Длина блока  $\ell$  случайно выбирается из множества  $\{2, 3, \dots, \lfloor \frac{n}{2} \rfloor\}$ . После этого случайно определяется положение  $a$  первого

требования в первом блоке и положение  $b$  первого требования во втором блоке. При этом  $a + \ell \leq b \leq n - \ell + 1$ . Для  $\pi_0 = (3, 1, 4, 2, 5)$  из примера возможен выбор  $\ell = 2$ ,  $a = 2$  и  $b = 4$ . Тогда  $SwapBlock(\pi_0, a, b, \ell) = SwapBlock(\pi_0, 2, 4, 2) = (3, 2, 5, 1, 4)$ .

*Подстановка*: случайно выбранное требование удаляется с первоначальной позиции  $a$  и помещается в новую, случайно выбранную, позицию  $b$ . Для  $\pi_0 = (3, 1, 4, 2, 5)$  из примера две случайно выбранные позиции  $a = 2$  и  $b = 4$  дают  $Insert(\pi_0, a, b) = Insert(\pi_0, 2, 4) = (3, 4, 2, 1, 5)$ .

*Подстановка блока*: случайно выбранный блок удаляется с позиции  $a$  и помещается в позицию  $b$ . Длина блока  $\ell$  случайно выбирается из множества  $\{2, 3, \dots, n - b\}$ . После выбора  $\ell$  определяются позиции  $a$  и  $b$ , причем  $b + \ell - 1 \leq n$ . Для  $\pi_0 = (3, 1, 4, 2, 5)$  из примера возможен выбор  $\ell = 2$ ,  $a = 2$  и  $b = 4$ . Тогда  $InsertBlock(\pi_0, a, b, \ell) = InsertBlock(\pi_0, 2, 4, 2) = (3, 2, 5, 1, 4)$ .

*Поворот блока*: поворачивается  $\ell$  элементов последовательности. Для  $\pi_0 = (3, 1, 4, 2, 5)$  из примера возможное значение  $\ell = 3$  и  $a = 2$ . Тогда  $ReverseBlock(\pi_0, a, \ell) = ReverseBlock(\pi_0, 2, 3) = (3, 2, 4, 1, 5)$ .

*Подстановка блока и поворот*: после подстановки блока производится поворот. Для  $\pi_0 = (3, 1, 4, 2, 5)$  из примера возможно  $\ell = 2$ ,  $a = 2$  и  $b = 4$ . Тогда  $InsertReverseBlock(\pi_0, a, b, \ell) = InsertReverseBlock(\pi_0, 2, 4, 2) = (3, 2, 5, 4, 1)$ .

На каждой итерации алгоритма имитации отжига используются все семь операций, генерируя семь соседних расписаний. Затем выделяется соседнее расписание с наименьшим значением критерия. Полученное соседнее расписание принимается за новое по правилу, описанному выше.

Результаты тестирования нескольких схем тушения (законов изменения температуры) показали, что геометрическая схема слегка превосходит остальные, поэтому выбрали ее. Геометрическая схема тушения уменьшает температуру по правилу  $T_k = \alpha T_{k-1}$ ,  $k = 1, 2, \dots$ , где  $0 < \alpha < 1$ . Было установлено, что начальная температура должна выбираться таким образом, чтобы в 25 % случаев худшее расписание принималось за новое. Начальная температура в рассматриваемом алгоритме  $T = 15$ . Конечная температура должна быть достаточно низкой, чтобы худшие решения перестали заменять лучшие. Экспериментально было подобрано значение

$\alpha = 0,999$ . После каждой итерации температура в алгоритме изменяется. Фактически температура должна изменяться пропорционально отведенному времени, т. е. когда лимит времени  $TL$  исчерпан, температура должна быть близка к 0. В соответствии с геометрической схемой тушения на последнем шаге алгоритма  $T_N = \alpha^N T_0$ . Получаем  $N = \log_{\alpha} \frac{T_N}{T_0}$ . Выбираем  $T_N = 0,0005$ , при таком выборе алгоритм содержит  $N = 10\,304$  итераций.

Начальное расписание генерируется случайным образом. Критерием остановки служит первое из возможных событий:

- исчерпано выделенное время;
- за последние 2000 итераций значение целевой функции не улучшалось;
- для целевой функции  $BestSumC$  выполняется  $BestSumC - LB < 1$ ;
- текущая температура ниже 0,0005.

В алгоритме  $SA\ Rand(0,1)$  обозначает некое число из интервала  $(0,1)$ , случайно выбранное и равномерно распределенное.

BEGIN

Генерируем начальное расписание  $X$  и вычисляем значение целевой функции  $SumC(\pi)$ ;  
 $BestSol := \pi$ ;  $BestSumC := SumC(\pi)$

$T :=$  начальная температура;

WHILE (критерий остановки не выполнен) DO

$\pi' :=$  лучший сосед среди сгенерированных соседей для расписания  $\pi$ ;

$\Delta C := SumC(\pi') - SumC(\pi)$ ;

$prob := Rand(0,1)$ ;

IF (( $\Delta C \leq 0$ ) или ( $prob < e^{-\Delta C/T}$ )) THEN

$\pi := \pi'$ ;  $SumC(\pi) := SumC(\pi')$ ;

IF ( $SumC(\pi) < BestSumC$ ) THEN

$BestSumC := SumC(\pi)$ ;  $BestSol := \pi$ ;

END IF

END IF

$T := Update(T)$ ;

END WHILE

Output  $BestSol$  и значение  $SumC$  ;

END.

### 3. Результаты тестирования

Эффективность моделей M1, M2 и алгоритма SA тестировалась на данных, генерируемых подобно тому, как описано в [9]. Предельное время установлено равным  $(300/8)n$  с для примеров с  $n \in \{8, 20, 50\}$  и 3600 с для остальных примеров.

Для  $n \in \{8, 20\}$  и для каждого  $L \in \{0,1, 0,2, \dots, 2\}$  значение  $s_j$  распределено равномерно в интервале  $(0, 100L)$ . Для каждого значения  $L$  10 примеров генерируются со значениями  $p_j$ , равномерно распределенными в интервале  $(0, 100)$ .

Для  $n \in \{50, 100, 200, 250\}$  пять примеров генерируется для каждого  $L \in \{0,1, 0,5, 0,8, 1, 1,5, 1,8, 2\}$ . При этом значения  $p_j$  равномерно распределены в интервале  $(0, 100)$ , а значения  $s_j$  – в интервале  $(0, 100L)$ .

Для работы с моделями M1 и M2 использовался пакет CPLEX 10.1, при этом выделялась рабочая память 2 Гб. Алгоритм SA реализован на языке java с использованием среды JDK 1.3.0, используемая оперативная память 2 Гб, используемый процессор Intel(R) Core(TM) i5-2430M 2.4 GHz.

В табл. 1–4 первый столбец содержит число требований  $n$ , второй показывает значение  $L$ , третий указывает на тестируемый объект, столбцы с четвертого по шестой указывают минимальное, среднее и максимальное время счета, в столбцах с седьмого по девятый указаны ми-

нимальное, среднее и максимальное значения отношения  $\sum C_j/LB$  либо  $\sum C_j/\sum C_j^*$ , если оптимальное значение  $\sum C_j^*$  удастся определить.

Таблица 1

$n$	$L$	mod/alg	min time, s	average time, s	max time, s	min $\sum C_j/\sum C_j^*$	average $\sum C_j/\sum C_j^*$	max $\sum C_j/\sum C_j^*$
8	0,1	M1	33,0	48,4	59,7	1,00	1,00	1,00
		M2	0,0	0,1	0,2	1,00	1,02	1,05
		SA	9,5	10,5	13,4	1,00	1,00	1,00
8	0,5	M1	2,4	20,8	58,3	1,00	1,00	1,00
		M2	0,0	0,1	0,1	1,00	1,01	1,02
		SA	10,6	13,2	15,4	1,00	1,00	1,00
	0,8	M1	1,5	9,1	18,8	1,00	1,00	1,00
		M2	0,0	0,1	0,1	1,00	1,01	1,03
		SA	11,0	13,8	15,5	1,00	1,00	1,00
	1,0	M1	2,6	8,2	18,4	1,00	1,00	1,00
		M2	0,0	0,0	0,1	1,00	1,00	1,00
		SA	13,2	14,3	15,7	1,00	1,00	1,00
	1,5	M1	0,6	2,8	6,0	1,00	1,00	1,00
		M2	0,0	0,0	0,1	1,00	1,02	1,06
		SA	13,0	15,0	15,9	1,00	1,00	1,00
	1,8	M1	0,3	2,5	6,5	1,00	1,00	1,00
		M2	0,1	1,1	2,2	1,00	1,00	1,01
		SA	15,5	15,0	11,7	1,00	1,00	1,00
	2,0	M1	0,3	2,4	6,0	1,00	1,00	1,00
		M2	0,0	0,0	0,0	1,00	1,00	1,00
		SA	12,5	14,9	16,5	1,00	1,00	1,00

Из табл. 1 видно, что модель M1 и алгоритм SA находят оптимум достаточно быстро. Решения модели M2 часто не точны, однако эта модель работает очень быстро. Заметно, что время работы модели M1 уменьшается с ростом  $L$ , а алгоритма SA, наоборот, увеличивается.

Таблица 2

$n$	$L$	mod/alg	min time, s	ave time, s	max time, s	min $\sum C_j/LB$	ave $\sum C_j/LB$	max $\sum C_j/LB$
20	0,1	M1	750,0	750,1	750,4	1,04	1,07	1,13
		M2	0,0	0,1	0,5	1,00	1,01	1,02
		SA	41,4	44,1	48,2	1,00	1,00	1,01
	0,5	M1	750,0	750,6	752,7	1,07	1,09	1,13
		M2	0,4	28,4	96,0	1,01	1,05	1,13
		SA	55,5	60,4	65,1	1,01	1,03	1,07
	0,8	M1	750,0	750,0	750,0	1,05	1,09	1,11
		M2	0,2	10,3	29,7	1,02	1,05	1,11
		SA	56,3	62,7	71,6	1,01	1,03	1,06
	1,0	M1	750,0	750,0	750,2	1,09	1,14	1,19
		M2	10,6	91,5	340,9	1,05	1,08	1,10
		SA	57,6	64,9	68,6	1,05	1,06	1,08
	1,5	M1	750,0	750,1	750,3	1,10	1,14	1,20
		M2	2,1	8,7	17,2	1,03	1,07	1,10

		SA	61,4	66,0	67,6	1,03	1,06	1,10
	1,8	M1	750,0	750,0	750,0	1,06	1,10	1,16
		M2	0,1	1,1	2,2	1,00	1,04	1,06
		SA	52,7	58,4	64,2	1,00	1,03	1,05

Окончание таблицы 2

$n$	$L$	mod/alg	min time, s	ave time, s	max time, s	min $\sum C_j/LB$	ave $\sum C_j/LB$	max $\sum C_j/LB$
20	2,0	M1	750,0	750,4	751,9	1,02	1,06	1,15
		M2	0,0	0,9	3,1	1,01	1,03	1,08
		SA	62,3	70,9	79,4	1,01	1,03	1,07

Для  $n = 20$  небольшой проигрыш в качестве полученных решений для модели M2 компенсируется в большинстве случаев значительным опережением по времени. По времени M2 опережает и SA, и M1.

Для  $n = 50, 100, 200$  модель M1 оказалась не в состоянии за отведенное время сгенерировать допустимое решение для большинства примеров, поэтому далее сравнивается лишь эвристика SA с моделью M2.

Таблица 3

$n$	$L$	mod/alg	min time, s	ave time, s	max time, s	min $\sum C_j/LB$	ave $\sum C_j/LB$	max $\sum C_j/LB$
50	0,1	M2	0	77	238	1,00	1,00	1,01
		SA	369	459	586	1,00	1,00	1,00
	0,5	M2	177	980	1875	1,00	1,04	1,06
		SA	372	547	734	1,00	1,01	1,02
	0,8	M2	1069	1680	1876	1,03	1,06	1,11
		SA	463	510	619	1,02	1,03	1,04
	1	M2	1025	1622	1875	1,06	1,08	1,11
		SA	444	529	759	1,05	1,07	1,09
	1,5	M2	544	983	1531	1,03	1,05	1,10
		SA	331	436	509	1,03	1,05	1,08
	1,8	M2	577	664	782	1,03	1,05	1,07
		SA	336	410	540	1,02	1,05	1,07
	2,0	M2	531	609	691	1,02	1,06	1,11
		SA	338	420	519	1,02	1,05	1,10

Для  $n = 50$  алгоритм SA оказывается лучшим и по времени, и по качеству, однако для значения  $L = 0,1$  модель M2 предпочтительнее.

Таблица 4

$n$	$L$	mod/alg	min time, s	ave time, s	max time, s	min $\sum C_j/LB$	ave $\sum C_j/LB$	max $\sum C_j/LB$
100	0,1	M2	15	233	633	1,00	1,00	1,00
		SA	1792	1865	1960	1,00	1,00	1,00
	0,5	M2	711	1039	1400	1,01	1,02	1,04
		SA	1843	2065	2212	1,00	1,01	1,01
	0,8	M2	1718	1847	2064	1,03	1,05	1,07
		SA	1592	1764	2086	1,01	1,02	1,04
	1	M2	1702	1861	2080	1,05	1,07	1,11
		SA	1925	2041	2332	1,02	1,04	1,07
	1,5	M2	1733	1837	1944	1,03	1,06	1,09

		SA	1525	1778	1886	1,02	1,05	1,07
	1,8	M2	1310	1683	1998	1,04	1,05	1,06
		SA	1561	1779	2432	1,03	1,04	1,05
	2,0	M2	920	1123	1354	1,01	1,03	1,05
		SA	1561	1833	2349	1,01	1,03	1,05

Для  $n = 100$  и  $L = 0,8, 1, 1,5$  алгоритм SA показывает лучшие результаты, чем модель M2. Однако для  $L = 0,1, 0,5, 2,0$  модель M2 предпочтительнее, чем SA.

В табл. 5 и 6 предельное время задано во втором столбце.

Таблица 5

$n$	time, s	$L$	Mod/alg	min $\sum C_j/LB$	ave $\sum C_j/LB$	max $\sum C_j/LB$
200	3600	0,1	M2	1,00	1,00	1,00
			SA	1,00	1,00	1,00
		0,5	M2	1,02	1,02	1,03
			SA	1,01	1,01	1,01
		0,8	M2	1,03	1,05	1,08
			SA	1,01	1,02	1,02
		1	M2	1,04	1,06	1,08
			SA	1,02	1,02	1,04
		1,5	M2	1,06	1,08	1,11
			SA	1,05	1,07	1,09
		1,8	M2	1,03	1,04	1,06
			SA	1,02	1,03	1,05
		2,0	M2	1,02	1,05	1,07
			SA	1,02	1,04	1,06

Для  $n = 200$  алгоритм SA оказывается лучше, чем модель M2, во всех случаях.

Таблица 6

$n$	time, s	$L$	mod/alg	min $\sum C_j/LB$	ave $\sum C_j/LB$	max $\sum C_j/LB$
250	3600	0,1	M2	1,00	1,00	1,00
			SA	1,00	1,00	1,00
		0,5	M2	1,01	1,04	1,06
			SA	1,01	1,01	1,02
		0,8	M2	1,04	1,06	1,08
			SA	1,01	1,02	1,02
		1	M2	1,06	1,07	1,08
			SA	1,02	1,02	1,03
		1,5	M2	1,04	1,07	1,12
			SA	1,03	1,05	1,09
		1,8	M2	1,03	1,05	1,06
			SA	1,02	1,03	1,04
		2,0	M2	1,02	1,05	1,07
			SA	1,02	1,04	1,05

Для  $n = 250$  алгоритм SA превосходит модель M2.

### Заключение

Для примеров с числом требований, не превосходящим 100, модель M2 показывает очень хорошие результаты. Учитывая предельно простую структуру строящихся расписаний, этот факт выглядит любопытным. Между тем критерий  $\sum C_j$  более сложен для SA по сравнению с критерием  $C_{\max}$ . Метод имитации отжига является достаточно эффективным для задачи  $P2, S1 || C_{\max}$  и работает с примерами, содержащими до 1000 требований [9].

### Список литературы

1. Hall, N. Parallel machine scheduling with a common server / N. Hall, C. Potts, C. Sriskandarajah // *Discrete Applied Mathematics*. – 2000. – Vol. 102. – P. 223–243.
2. Complexity results for parallel machine problems with a single server / P. Brucker [et al.] // *Journal of Scheduling*. – 2002. – Vol. 5. – P. 429–457.
3. Kravchenko, S.A. A heuristic algorithm for minimizing mean flow time with unit setups / S.A. Kravchenko, F. Werner // *Information Processing Letters*. – 2001. – Vol. 79. – P. 291–296.
4. Wang, G. An approximation algorithm for parallel machine scheduling with a common server / G. Wang, T.C.E. Cheng // *J. of the Operational Research Society*. – 2001. – Vol. 52. – P. 234–237.
5. Weng, M.X. Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective / M.X. Weng, J. Lu, H. Ren // *International J. of Production Economics*. – 2001. – Vol. 70. – P. 215–226.
6. Dunstall, S. Heuristic methods for the identical parallel machine flowtime problem with set-up times / S. Dunstall, A. Wirth // *Computers & Operations Research*. – 2005. – Vol. 32. – P. 2479–2491.
7. Azizoglu, M. Scheduling parallel machines to minimize weighted flowtime with family set-up times / M. Azizoglu, S. Webster // *International J. of Production Research*. – 2003. – Vol. 41. – P. 1199–1215.
8. Guirchoun, S. Total completion time minimization in a computer system with a server and two parallel processors / S. Guirchoun, P. Martineau, J.-C. Billaut // *Computers & Operations Research*. – 2005. – Vol. 32. – P. 599–611.
9. Hasani, K. Two heuristics for minimizing the makespan for the two-machine scheduling problem with a single server / K. Hasani, S.A. Kravchenko, F. Werner. – Magdeburg, 2013. – 20 p. – (Preprint / Otto-von-Guericke-University Magdeburg, Faculty of Mathematics ; № 08/13).

Поступила 12.07.2013

<sup>1</sup>Университет Магдебурга, Германия

<sup>2</sup>Объединенный институт проблем информатики НАН Беларуси,  
Минск, Сурганова, 6  
e-mail: [kravch@newman.bas-net.by](mailto:kravch@newman.bas-net.by)

<sup>3</sup>Исламский университет Азад, Иран

**F. Werner, S.A. Kravchenko, K. Hasani**

### MINIMIZING MEAN FLOW TIME FOR THE TWO-MACHINE SCHEDULING PROBLEM WITH A SINGLE SERVER

The problem of scheduling jobs on two parallel machines to minimize the sum of completion times is considered. Each job requires a setup which is done by a single server. It is known that this problem is strongly NP-hard. Two mixed integer linear programming models and a simulated annealing algorithm are proposed. The performance of these algorithms is evaluated for instances with up to 250 jobs.

УДК 004.942:539.371

К.С. Курочка

## КОНЕЧНОЭЛЕМЕНТНОЕ МОДЕЛИРОВАНИЕ ПРОГИБОВ ТОНКИХ КРУГЛЫХ ТРЕХСЛОЙНЫХ ПЛАСТИН

*Представляется математическая модель тонкой круглой трехслойной пластины под действием вертикальной нагрузки на основе метода конечных элементов с использованием осесимметричного конечного элемента, достоинством которого является малая размерность получаемой матрицы жесткости при обеспечении достаточной для практического использования точности расчетов. В ходе исследования находятся аналитические выражения для вычисления локальных матриц жесткости, что позволяет значительно ускорить процесс формирования глобальных матриц жесткости и увеличить точность проводимых расчетов.*

*Разрабатывается соответствующее программное обеспечение и проводится его верификация. Расхождение результатов исследования предлагаемой математической модели и имеющихся результатов расчетов по аналитическим формулам для однородных круглых тонких трехслойных пластин не превышает 7 %.*

### Введение

В настоящее время все большее применение находят трехслойные элементы конструкций. Совместное использование материалов с существенно различающимися термомеханическими характеристиками позволяет получать в рамках конструкции новые полезные свойства, недостижимые при применении однородных элементов. Прочные и жесткие несущие слои обеспечивают необходимые значения деформаций, а внутренние слои, перераспределяя усилия между несущими слоями, могут также выполнять и ряд других функций, например тепло- и звукоизоляцию, демпфирование и снижение вибраций и т. п. [1, 2].

Одним из распространенных трехслойных элементов технических систем является тонкая пластина, нашедшая широкое применение в строительных конструкциях [2, 3]. В современном градостроительстве часто возводятся здания каркасного типа, основным элементом которых служат диски перекрытий, состоящие из многопустотных железобетонных плит, связевых ригелей и несущих колонн [3]. Все данные элементы в плане являются неоднородными, содержат пустоты или включения арматуры, что требует их рассмотрения как многослойных.

Повсеместное внедрение энергосберегающих технологий привело к широкому использованию шумо- и теплоизоляционных материалов, которые конструктивно размещаются на несущих элементах зданий и сооружений, таким образом превращая их в многослойные элементы.

Прочностные расчеты подобных конструкций нельзя проводить без учета их слоистой структуры, так как это может привести к значительным погрешностям. Одним из эффективных методов исследования подобных конструкций, позволяющих учесть все особенности их поведения, является компьютерное моделирование посредством построения и исследования соответствующих математических моделей [4–7].

Для построения математической модели прогибов тонкой круглой трехслойной пластины воспользуемся методом конечных элементов [4–6].

### 1. Постановка задачи

Рассмотрим тонкую круглую трехслойную пластину под действием вертикальной поперечной нагрузки. Воспользуемся цилиндрической системой координат. Начало координат поместим в центр пластины. Пластина состоит из внешних несущих слоев 1 и 2 соответственно толщиной  $h_1$  и  $h_2$  и жесткого заполнителя толщиной  $h_0$ , воспринимающего нагрузку в тангенциальном направлении (рис. 1). На контуре пластины предполагается наличие жесткой диафрагмы, препятствующей относительному сдвигу слоев ( $\gamma_{rz} = 0$  при  $r = 1$ ), при этом пластина может

быть свободно оперта, шарнирно оперта или закреплена [4]. Перемещения в каждом слое будут определяться из уравнения Софи Жермен – Лагранжа в полярной системе координат [1]:

$$w^{IV} + \frac{2}{r} w''' - \frac{1}{r^2} w'' + \frac{1}{r^3} w' = \frac{q(r)}{D},$$

где  $w$  – прогиб срединной поверхности;  $q(r)$  – поперечная нагрузка;  $D = \frac{Eh^3}{12(1-\nu^2)}$  – цилиндрическая жесткость слоя пластины;  $E$  – модуль упругости;  $h$  – толщина слоя пластины;  $\nu$  – коэффициент Пуассона.

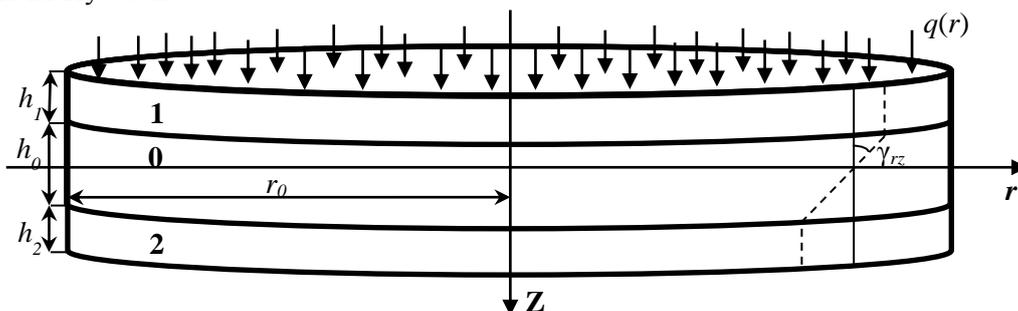


Рис. 1. Круглая трехслойная пластина под действием вертикальной поперечной нагрузки

Для слоев пластины положим справедливыми гипотезы Кирхгофа [1, 4, 8], из которых аналогично, как и в случае декартовых координат [9], можно получить

$$w = w(r); \quad u = u(r, z) = -z \frac{dw(r)}{dr}; \quad \vartheta = 0,$$

где  $u$  – радиальные перемещения;  $\vartheta$  – угловые перемещения.

Формулы Коши [1, 4, 9] для ненулевых компонент деформаций в этом случае примут вид

$$\varepsilon_r = \frac{du}{dr} = -z \frac{d^2w}{dr^2}; \quad \varepsilon_\varphi = \frac{u}{r} = -z \frac{1}{r} \frac{dw}{dr}; \quad \gamma_{rz} = -z \frac{dw}{dr}. \quad (1)$$

В качестве искомой величины рассмотрим прогиб тонкой круглой пластины  $w = w(r)$ , для моделирования которого воспользуемся осесимметричными конечными элементами с двумя узлами по две степени свободы в каждом  $\{g_0\}^T = \{w \ \theta\}$ , где  $\{g_0\}$  – вектор узловых степеней свободы конечного элемента;  $\theta = dw/dr$  – угол поворота в радиальном направлении (рис. 2).

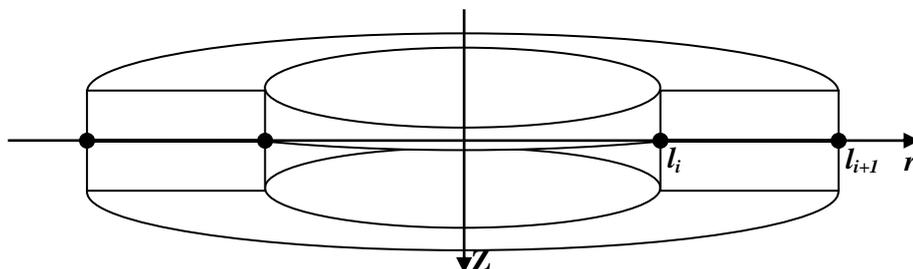


Рис. 2. Осесимметричный конечный элемент для моделирования прогибов тонких круглых пластин

Для построения математической модели применим вариационный принцип Лагранжа [1, 4, 8], который в случае круглой тонкой пластины можно переписать в виде

$$\begin{Bmatrix} - \\ g \end{Bmatrix}^T \{R\} = \int_{l_i}^{l_{i+1}} \int_0^{2\pi} \int_{-\frac{h}{2}}^{\frac{h}{2}} \begin{Bmatrix} - \\ \varepsilon \end{Bmatrix}^T \{\sigma\} dz d\varphi dr, \quad (2)$$

где  $\{g\}^T = \{w_i \ \theta_i \ w_{i+1} \ \theta_{i+1}\}$  – вектор перемещений;  $\{R\}^T = \{R_{zi} \ R_{\theta i} \ R_{zi+1} \ R_{\theta i+1}\}$  – вектор внешних усилий;  $\{\varepsilon\}^T = \{\varepsilon_r \ \varepsilon_\varphi \ \gamma_{rz}\}$  – вектор деформаций;  $\{\sigma\}^T = \{\sigma_r \ \sigma_\varphi \ \tau_{rz}\}$  – вектор напряжений;  $l_i$  – координата  $i$ -го узла; черта над переменной означает вариацию признака.

Будем аппроксимировать значения искомой функции  $w(r)$  и ее первой производной  $\theta$  следующими полиномами, обеспечивающими существование функционала вариационной задачи (2) [6]:

$$w(r) = a_1 + a_2 r + a_3 r^2 + a_4 r^3; \quad \theta(r) = a_2 + 2a_3 r + 3a_4 r^2. \quad (3)$$

Так как соотношения (3) справедливы для всех точек конечного элемента, для его узлов получим

$$\{g\} = [A]\{a\},$$

$$\text{где } \{a\}^T = \{a_1 \ a_2 \ a_3 \ a_4\}; \quad [A] = \begin{bmatrix} 1 & l_i & l_i^2 & l_i^3 \\ 0 & 1 & 2l_i & 3l_i^2 \\ 1 & l_{i+1} & l_{i+1}^2 & l_{i+1}^3 \\ 0 & 1 & 2l_{i+1} & 3l_{i+1}^2 \end{bmatrix}.$$

Отсюда следует

$$\{a\} = [A]^{-1}\{g\}. \quad (4)$$

## 2. Осесимметричный конечный элемент для моделирования прогибов тонкой круглой пластины

Для тонкой круглой пластины будут отсутствовать сдвиговые деформации, т. е. векторы деформаций и напряжений будут содержать только по две компоненты. Тогда принцип возможных перемещений (2) можно будет переписать следующим образом:

$$\left\{ \bar{g} \right\}^T \{R\} = 2\pi \int_{-\frac{h}{2}}^{\frac{h}{2}} \int_{l_i}^{l_{i+1}} \left( \bar{\varepsilon}_r \sigma_r + \bar{\varepsilon}_\varphi \sigma_\varphi \right) r dr dz. \quad (5)$$

Закон Гука [1, 6] примет вид

$$\{\sigma\} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu \\ \nu & 1 \end{bmatrix} \{\varepsilon\}. \quad (6)$$

Подставляя (6) в (5), получим

$$\begin{aligned} \left\{ \bar{g} \right\}^T \{R\} &= 2\pi \frac{E}{1-\nu^2} \int_{-\frac{h}{2}}^{\frac{h}{2}} \int_{l_i}^{l_{i+1}} \left( \bar{\varepsilon}_r (\varepsilon_r + \nu \varepsilon_\varphi) + \bar{\varepsilon}_\varphi (\varepsilon_\varphi + \nu \varepsilon_r) \right) r dr dz = \\ &= 2\pi \frac{E}{1-\nu^2} \int_{-\frac{h}{2}}^{\frac{h}{2}} \int_{l_i}^{l_{i+1}} \left( \bar{\varepsilon}_r \varepsilon_r + \nu \bar{\varepsilon}_r \varepsilon_\varphi + \bar{\varepsilon}_\varphi \varepsilon_\varphi + \nu \bar{\varepsilon}_\varphi \varepsilon_r \right) r dr dz. \end{aligned} \quad (7)$$

Воспользовавшись формулами Коши (1) и соотношениями (3), получим

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_r \\ \varepsilon_\varphi \end{Bmatrix} = -z \begin{Bmatrix} \frac{d^2 w}{dr^2} \\ \frac{1}{r} \theta_r \end{Bmatrix} = -z \begin{bmatrix} Cr \\ \frac{1}{r} C\varphi \end{bmatrix} \{a\}, \quad (8)$$

где  $\{Cr\} = \{0 \ 0 \ 2 \ 6r\}$ ;  $\{C\varphi\} = \{0 \ 1 \ 2r \ 3r^2\}$ .

Подставляя (4) в (8), можно получить выражение

$$\{\varepsilon\} = -z \begin{bmatrix} Cr \\ \frac{1}{r} C\varphi \end{bmatrix} [A]^{-1} \{g\}$$

или

$$\varepsilon_r = -z \{Q_r\} \{g\}; \quad \varepsilon_\varphi = -z \frac{1}{r} \{Q_\varphi\} \{g\}, \quad (9)$$

где

$$[Q_r] = \begin{bmatrix} \frac{6(l_1 + l_2 - 2x)}{(l_1 - l_2)^3} & -\frac{2(l_1 + 2l_2 - 3x)}{(l_1 - l_2)^2} & -\frac{6(l_1 + l_2 - 2x)}{(l_1 - l_2)^3} & -\frac{2(2l_1 + l_2 - 3x)}{(l_1 - l_2)^2} \end{bmatrix};$$

$$[Q_\varphi] = \begin{bmatrix} -\frac{6(l_1 - x) \cdot (12 - x)}{(l_1 - l_2)^3} & \frac{(l_2 - x) \cdot (2l_1 + l_2 - 3x)}{(l_1 - l_2)^2} & \frac{6(l_1 - x) \cdot (l_2 - x)}{(l_1 - l_2)^3} & \frac{(l_1 - x) \cdot (l_1 + 2l_2 - 3x)}{(l_1 - l_2)^2} \end{bmatrix}.$$

Подставляя (9) в (7) и выполняя необходимые преобразования, несложно вычислить

$$\{R\} = 2\pi \frac{Eh^3}{12(1 - \nu^2)} \left[ \int_{l_1}^{l_{i+1}} \left( r \{Q_r\}^T \{Q_r\} + \nu \{Q_r\}^T \{Q_\varphi\} + \nu \{Q_\varphi\}^T \{Q_r\} + \frac{1}{r} \{Q_\varphi\}^T \{Q_\varphi\} \right) dr \right] \{g\}. \quad (10)$$

Перепишем (10) в виде

$$\{R\} = [k] \{g\}, \quad (11)$$

где  $[k] = D_\varphi \left( [K_{rr}] + \nu [K_{r\varphi}] + \nu [K_{\varphi r}] + [K_{\varphi\varphi}] \right)$  – локальная матрица жесткости;  $D_\varphi = 2\pi \frac{Eh^3}{12(1 - \nu^2)}$ ;

$$[K_{rr}] = \int_{l_1}^{l_{i+1}} r \{Q_r\}^T \{Q_r\} dr; \quad [K_{r\varphi}] = \int_{l_1}^{l_{i+1}} \{Q_r\}^T \{Q_\varphi\} dr; \quad [K_{\varphi r}] = \int_{l_1}^{l_{i+1}} \{Q_\varphi\}^T \{Q_r\} dr; \quad [K_{\varphi\varphi}] = \int_{l_1}^{l_{i+1}} \frac{1}{r} \{Q_\varphi\}^T \{Q_\varphi\} dr.$$

Все интегралы в (11) могут быть вычислены точно:

$$K_{rr} = \begin{bmatrix} -\frac{6l_1 + 6l_2}{(l_1 - l_2)^3} & \frac{4l_1 + 2l_2}{(l_1 - l_2)^2} & \frac{6l_1 + 6l_2}{(l_1 - l_2)^3} & \frac{2l_1 + 4l_2}{(l_1 - l_2)^2} \\ \frac{4l_1 + 2l_2}{(l_1 - l_2)^2} & 1 - \frac{4l_1}{l_1 - l_2} & -\frac{4l_1 + 2l_2}{(l_1 - l_2)^2} & 1 - \frac{2l_1}{l_1 - l_2} \\ \frac{6l_1 + 6l_2}{(l_1 - l_2)^3} & -\frac{4l_1 + 2l_2}{(l_1 - l_2)^2} & \frac{6l_1 + 6l_2}{(l_1 - l_2)^3} & -\frac{2l_1 + 4l_2}{(l_1 - l_2)^2} \\ \frac{2l_1 + 4l_2}{(l_1 - l_2)^2} & 1 - \frac{2l_1}{l_1 - l_2} & -\frac{2l_1 + 4l_2}{(l_1 - l_2)^2} & \frac{4l_2}{l_1 - l_2} - 1 \end{bmatrix};$$

$$K_{\varphi\varphi} = \begin{bmatrix} k_{r\varphi11} & k_{r\varphi12} & k_{r\varphi13} & k_{r\varphi14} \\ k_{r\varphi21} & k_{r\varphi22} & k_{r\varphi23} & k_{r\varphi24} \\ k_{r\varphi31} & k_{r\varphi32} & k_{r\varphi33} & k_{r\varphi34} \\ k_{r\varphi41} & k_{r\varphi42} & k_{r\varphi43} & k_{r\varphi44} \end{bmatrix},$$

где

$$k_{r\varphi11} = -k_{r\varphi31} = -k_{r\varphi13} = k_{r\varphi33} = -\frac{3 \cdot (8 \cdot l_1 \cdot l_2^3 - 8 \cdot l_1^3 \cdot l_2 + l_1^4 - l_2^4 + 12 \cdot l_1^2 \cdot l_2^2 \cdot \ln(l_1) - 12 \cdot l_1^2 \cdot l_2^2 \cdot \ln(l_2))}{(l_1 - l_2)^6},$$

$$k_{r\varphi21} = k_{r\varphi12} = -k_{r\varphi32} = -k_{r\varphi23} = -\frac{9 \cdot l_1^2 \cdot l_2 - l_1^3 + 27 \cdot l_1 \cdot l_2^2 + l_2^3}{2 \cdot (l_1 - l_2)^4} - \frac{6 \cdot l_1 \cdot l_2^2 \cdot (2 \cdot l_1 + l_2)}{(l_1 - l_2)^5} \cdot (\ln(l_2) - \ln(l_1));$$

$$k_{r\varphi41} = -k_{r\varphi43} = -k_{r\varphi34} = k_{r\varphi14} = -\left[ \frac{l_1^3 + 27 \cdot l_1^2 \cdot l_2 + 9 \cdot l_1 \cdot l_2^2 - l_2^3}{2 \cdot (l_1 - l_2)^4} + \frac{6 \cdot l_1^2 \cdot l_2 \cdot (l_1 + 2l_2)}{(l_1 - l_2)^5} \cdot (\ln(l_2) - \ln(l_1)) \right];$$

$$k_{r\varphi22} = k_{r\varphi44} = \frac{7 \cdot l_1^2 \cdot l_2 - l_1^3 + 19 \cdot l_1 \cdot l_2^2 + l_1 \cdot l_2^3}{4 \cdot (l_1 - l_2)^3} + \frac{l_2^2 \cdot (2 \cdot l_1 + l_2)^2}{(l_1 - l_2)^4} \cdot (\ln(l_2) - \ln(l_1));$$

$$k_{r\varphi42} = k_{r\varphi24} = \frac{l_1^3 + 17 \cdot l_1^2 \cdot l_2 + 17 \cdot l_1 \cdot l_2^2 + l_2^3}{4 \cdot (l_1 - l_2)^3} + \frac{l_1 \cdot l_2 \cdot (l_1 + 2 \cdot l_2) \cdot (2 \cdot l_1 + l_2)}{(l_1 - l_2)^4} \cdot (\ln(l_2) - \ln(l_1));$$

$$K_{r\varphi} = \begin{pmatrix} 0 & \frac{1}{l_1 - l_2} & 0 & -\frac{1}{l_1 - l_2} \\ -\frac{1}{l_1 - l_2} & -\frac{1}{2} & \frac{1}{l_1 - l_2} & \frac{1}{2} \\ 0 & -\frac{1}{l_1 - l_2} & 0 & \frac{1}{l_1 - l_2} \\ \frac{1}{l_1 - l_2} & -\frac{1}{2} & -\frac{1}{l_1 - l_2} & \frac{1}{2} \end{pmatrix}; K_{\varphi r} = \begin{pmatrix} 0 & -\frac{1}{l_1 - l_2} & 0 & \frac{1}{l_1 - l_2} \\ \frac{1}{l_1 - l_2} & -\frac{1}{2} & -\frac{1}{l_1 - l_2} & -\frac{1}{2} \\ 0 & \frac{1}{l_1 - l_2} & 0 & -\frac{1}{l_1 - l_2} \\ -\frac{1}{l_1 - l_2} & \frac{1}{2} & \frac{1}{l_1 - l_2} & \frac{1}{2} \end{pmatrix}.$$

Локальная матрица жесткости в (11) может быть вычислена по следующей формуле:

$$[k] = D_r \begin{bmatrix} -\frac{12}{(l_1 - l_2)^3} & \frac{6}{(l_1 - l_2)^2} & \frac{12}{(l_1 - l_2)^3} & \frac{6}{(l_1 - l_2)^2} \\ \frac{6}{(l_1 - l_2)^2} & -\frac{4}{l_1 - l_2} & -\frac{6}{(l_1 - l_2)^2} & -\frac{2}{l_1 - l_2} \\ \frac{12}{(l_1 - l_2)^3} & -\frac{6}{(l_1 - l_2)^2} & -\frac{12}{(l_1 - l_2)^3} & -\frac{6}{(l_1 - l_2)^2} \\ \frac{6}{(l_1 - l_2)^2} & -\frac{2}{l_1 - l_2} & -\frac{6}{(l_1 - l_2)^2} & -\frac{4}{l_1 - l_2} \end{bmatrix}, \quad (12)$$

где  $D_r = \frac{5\pi r E h^3}{36(1 - \nu^2)}$ .

После вычисления локальных матриц жесткости осуществлялось построение глобальной матрицы жесткости по формуле

$$[K_{2j,2j}] = [K_{2j,2j}] + [k_j], \quad j = \overline{1, N}, \quad (13)$$

где  $[K_{2j,2j}]$  – подматрица глобальной матрицы жесткости;  $[k_j]$  – локальная матрица жесткости  $j$ -го конечного элемента, вычисляемая по формулам (11) или (12);  $N$  – количество конечных элементов.

После построения глобальной матрицы жесткости формируется вектор узловых усилий  $\{R\}$ , содержащий суммарные нагрузки, действующие на всю круглую пластину. В случае рав-

номерно распределенной нагрузки интенсивностью  $q$  компоненты вектора узловых усилий  $\{R\}$  будут вычисляться по формуле [8]

$$R_{2i} = -q \cdot \pi \cdot [(i+1)^2 - (i)^2] \cdot \Delta r^2, \quad (14)$$

где  $q$  – интенсивность равномерно распределенной нагрузки;  $i$  – номер конечного элемента;  $\Delta r$  – радиус осесимметричного конечного элемента.

После формирования вектора внешних узловых нагрузок учитываются граничные условия и решается система линейных алгебраических уравнений вида (11) с глобальной матрицей, вычисленной по формуле (13).

### 3. Конечнэлементное моделирование трехслойной круглой пластины

При моделировании прогибов тонкой круглой трехслойной пластины для несущих слоев примем гипотезы Кирхгофа [1, 4, 9], как и для случая круглой тонкой пластины. В заполнителе в отличие от несущих слоев будут присутствовать сдвиговые деформации  $\gamma_{rz}$ .

Воспользуемся принципом возможных перемещений (2), который в случае трехслойной пластины перепишем в виде

$$\begin{aligned} \left\{ \bar{g} \right\}^T \{R\} = & \int_0^{2\pi} \int_{l_j}^{l_{j+1}} \int_{\frac{h_0}{2}}^{\frac{h_0}{2}+h_1} \left\{ \bar{\varepsilon}^{-1} \right\}^T \left\{ \sigma^1 \right\} dz d\varphi r dr + \int_0^{2\pi} \int_{l_j}^{l_{j+1}} \int_{\frac{h_0}{2}}^{\frac{h_0}{2}} \left\{ \bar{\varepsilon}^0 \right\}^T \left\{ \sigma^0 \right\} dz d\varphi r dr + \\ & + \int_0^{2\pi} \int_{l_j}^{l_{j+1}} \int_{\frac{h_0}{2}-h_2}^{\frac{h_0}{2}} \left\{ \bar{\varepsilon}^{-2} \right\}^T \left\{ \sigma^2 \right\} dz d\varphi r dr, \end{aligned}$$

где цифра в индексе означает номер слоя пластины, 0 соответствует заполнителю.

Последнее выражение можно переписать в виде

$$\left\{ \bar{g} \right\}^T \{R\} = 2\pi \left[ \int_{l_j}^{l_{j+1}} \int_{\frac{h_0}{2}}^{\frac{h_0}{2}+h_1} \left( \bar{\varepsilon}_r^{-1} \sigma_r^1 + \bar{\varepsilon}_\varphi^{-1} \sigma_\varphi^1 \right) dz r dr + \int_{l_j}^{l_{j+1}} \int_{\frac{h_0}{2}}^{\frac{h_0}{2}} \left( \bar{\varepsilon}_r^0 \sigma_r^0 + \bar{\varepsilon}_\varphi^0 \sigma_\varphi^0 + \bar{\gamma}_{rz}^0 \tau_{rz}^0 \right) dz r dr + \int_{l_j}^{l_{j+1}} \int_{\frac{h_0}{2}-h_2}^{\frac{h_0}{2}} \left( \bar{\varepsilon}_r^{-2} \sigma_r^2 + \bar{\varepsilon}_\varphi^{-2} \sigma_\varphi^2 \right) dz r dr \right]. \quad (15)$$

Аналогично выражениям (9) можно найти

$$\gamma_{rz} = -z \{Q_\varphi\} \{g\}. \quad (16)$$

Для несущих слоев закон Гука будет иметь вид (6), для заполнителя – вид

$$\left\{ \sigma^0 \right\} = \frac{E_0}{1-\nu_0^2} \begin{bmatrix} 1 & \nu_0 & 0 \\ \nu_0 & 1 & 0 \\ 0 & 0 & \frac{1-\nu_0}{2} \end{bmatrix} \left\{ \varepsilon^0 \right\}. \quad (17)$$

Подставляя (6) и (17) в (15), получим

$$\left\{ \bar{g} \right\}^T \{R\} = 2\pi \left[ \int_{l_j}^{l_{j+1}} \int_{\frac{h_0}{2}}^{\frac{h_0}{2}+h_1} \frac{E_1}{1-\nu_1^2} \left( \bar{\varepsilon}_r^{-1} \varepsilon_r^1 + \nu_1 \bar{\varepsilon}_\varphi^{-1} \varepsilon_\varphi^1 + \bar{\varepsilon}_\varphi^{-1} \varepsilon_r^1 + \nu_1 \bar{\varepsilon}_r^{-1} \varepsilon_\varphi^1 \right) dz r dr + \right.$$

$$\begin{aligned}
 & + \int_{l_j}^{l_{j+1}} \int_{-\frac{h_0}{2}}^{\frac{h_0}{2}} \frac{E_0}{1-\nu_0^2} \left( \varepsilon_r \varepsilon_r^0 + \nu_0 \varepsilon_r^0 \varepsilon_\varphi^0 + \varepsilon_\varphi \varepsilon_\varphi^0 + \nu_0 \varepsilon_\varphi^0 \varepsilon_r^0 + \frac{1-\nu_0}{2} \gamma_{rz}^0 \tau_{rz}^0 \right) dz r dr + \\
 & \left. + \int_{l_j}^{l_{j+1}} \int_{-\frac{h_0}{2}-h_2}^{-\frac{h_0}{2}} \frac{E_2}{1-\nu_2^2} \left( \varepsilon_r \varepsilon_r^{-2} + \nu_2 \varepsilon_r^{-2} \varepsilon_\varphi^{-2} + \varepsilon_\varphi \varepsilon_\varphi^{-2} + \nu_2 \varepsilon_\varphi^{-2} \varepsilon_r^{-2} \right) dz r dr \right].
 \end{aligned}$$

С помощью (9) и (16), интегрируя по  $z$ , последнее выражение преобразуется к виду (11), где

$$\begin{aligned}
 [k] = 2\pi \left[ \sum_{i=1}^2 \left( \frac{E_i}{1-\nu_i^2} \frac{(h_0 + 2h_i)^3 - h_0^3}{24} \left( [K_{rr}] + \nu_i [K_{r\varphi}] + \nu_i [K_{\varphi r}] + [K_{\varphi\varphi}] \right) \right) \right. \\
 \left. \frac{E_0}{1-\nu_0^2} \frac{h_0^3}{12} \left( [K_{rr}] + \nu_0 [K_{r\varphi}] + \nu_0 [K_{\varphi r}] + [K_{\varphi\varphi}] + \frac{1-\nu_0}{2} [K_{rz}] \right) \right]; \tag{18} \\
 [K_{rz}] = \int_{l_i}^{l_{i+1}} r \{Q_\varphi\}^T \{Q_\varphi\} dr
 \end{aligned}$$

или

$$[K_{rz}] = \begin{bmatrix} -\frac{3l_1 + 3l_2}{5l_1 - 5l_2} & \frac{l_2}{10} & \frac{3l_1 + 3l_2}{5l_1 - 5l_2} & \frac{l_1}{10} \\ \frac{l_2}{10} & -\frac{(l_1 - l_2)(3l_1 + l_2)}{30} & -\frac{l_2}{10} & \frac{l_1^2}{60} - \frac{l_2^2}{60} \\ \frac{3l_1 + 3l_2}{5l_1 - 5l_2} & -\frac{l_2}{10} & -\frac{3l_1 + 3l_2}{5l_1 - 5l_2} & -\frac{l_1}{10} \\ \frac{l_1}{10} & \frac{l_1^2}{60} - \frac{l_2^2}{60} & -\frac{l_1}{10} & -\frac{(l_1 - l_2)(l_1 + 3l_2)}{30} \end{bmatrix}.$$

Остальные матрицы вычисляются по формулам (11).

После вычисления локальных матриц жесткости (18) по формуле (13) формируется глобальная матрица жесткости. Вектор внешних узловых усилий вычисляется по формуле (14).

#### 4. Верификация предложенной математической модели

Для верификации предложенной математической модели рассмотрим ряд задач, для которых известны аналитические решения.

*Модельная задача 1.* Рассмотрим круглую тонкую пластину, находящуюся под действием равномерно распределенной нагрузки.

Как известно [1], прогиб в случае шарнирного опирания будет вычисляться по формуле

$$w(r) = -\frac{qR^4}{64D} \left( 1 - \left( \frac{r}{R} \right)^2 \right) \left( \frac{5+\nu}{1+\nu} - \left( \frac{r}{R} \right)^2 \right),$$

а в случае защемления – по формуле

$$w(r) = -\frac{qR^4}{64D} \left( 1 - \left( \frac{r}{R} \right)^2 \right)^2.$$

Примем радиус пластины  $r = 1$  м, толщину  $h = 0,05$  м, модуль упругости  $E = 0,829 \cdot 10^5$  МПа, коэффициент Пуассона  $\nu = 0,3$ . Интенсивность равномерно распределенной нагрузки  $q = 1$  МПа. Будем дискретизировать пластину на восемь конечных элементов. Резуль-

таты решения показаны на рис. 3. Локальную матрицу жесткости будем вычислять по формулам (11).

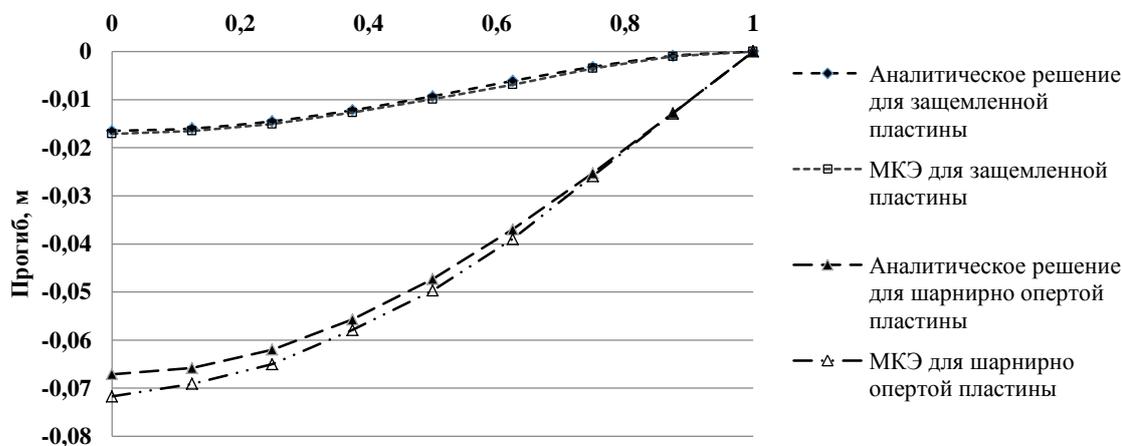


Рис. 3. Прогиб тонкой круглой пластины под действием равномерно распределенной нагрузки

Максимальный прогиб, найденный с помощью предложенного алгоритма, составил для защемленной пластины 0,0171 м, с помощью точного решения – 0,0165 м. Для шарнирно опертой пластины были получены значения 0,0717 и 0,0671 м соответственно. Таким образом, максимальная погрешность не превышала 7 % при небольшом числе конечных элементов.

*Модельная задача 2.* Рассмотрим защемленную по контуру трехслойную круглую пластину радиусом  $r = 1$  м под действием равномерно распределенной нагрузки интенсивностью 10 МПа [1]. В качестве несущих слоев стержня используется алюминиевый сплав Д16Т толщиной 0,04 м, модулем упругости  $E=0,1156 \cdot 10^6$  МПа и модулем объемной деформации  $K=0,8 \cdot 10^5$  МПа. Заполнителем является политетрафторэтилен толщиной 0,2 м, модулем упругости  $E=0,482 \cdot 10^4$  МПа и модулем объемной деформации  $K=0,47 \cdot 10^4$  МПа. Пластина дискретизировалась восемью осесимметричными конечными элементами.

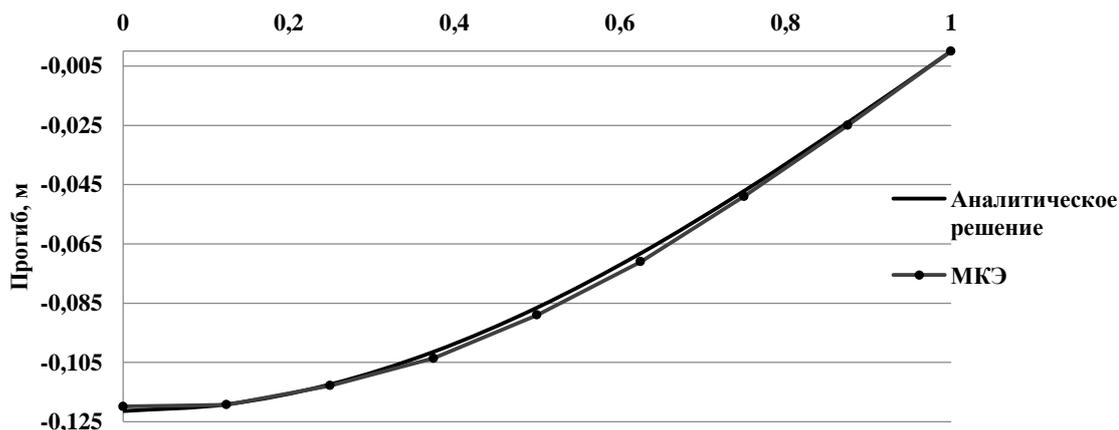


Рис. 4. Прогиб трехслойной круглой пластины под действием равномерно распределенной нагрузки

Решение с помощью предложенного алгоритма сравнивалось с решением из [1] (рис. 4). Матрица жесткости вычислялась по формулам (18). Максимальная погрешность решений не превышала 3 %.

Кроме того, верификация дополнительно осуществлялась на задачах моделирования напряженно-деформированного состояния трехслойных стержней. Погрешность найденных значений их прогиба не превышала 5 %.

### 5. Вычислительный эксперимент

С помощью созданной математической модели и разработанного на ее основании программного обеспечения проведем исследование влияния толщины заполнителя на максимальный прогиб. Для этого рассмотрим круглую трехслойную пластину, описанную в модельной задаче 2. Будем варьировать толщину заполнителя от 0 до 0,5 м. Результаты вычислительного эксперимента представлены на рис. 5.

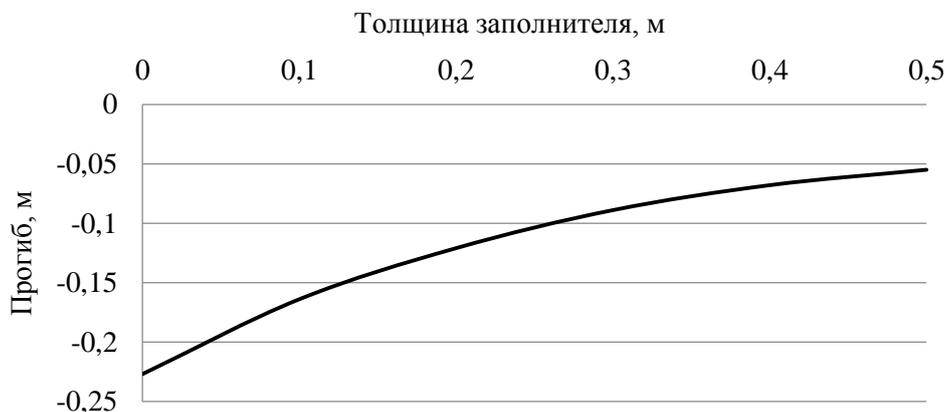


Рис. 5. Зависимость максимального прогиба тонкой трехслойной круглой пластины от толщины заполнителя

Из рис. 5 видно, что максимальный прогиб нелинейно зависит от толщины заполнителя, размеры которого влияют как на массу конструкции в целом, так и на ее термофизические характеристики, позволяя минимизировать эффект термосиловых напряжений и обеспечивая необходимые прочностные характеристики.

### Заключение

Согласно результатам проведенного моделирования предлагаемая математическая модель и численный алгоритм ее реализации могут быть использованы для исследования напряженно-деформированного состояния тонких трехслойных круглых пластин.

Получены соотношения для аналитического вычисления локальных матриц жесткости, что минимизирует погрешность при нахождении глобальной матрицы жесткости и позволяет ускорить процесс ее формирования.

Достоинством предлагаемой математической модели и методики ее применения является использование осесимметричных конечных элементов, позволяющих для дискретизации исследуемой пластины применять меньшее число узлов, чем при использовании элементов других типов. Кроме того, глобальная матрица жесткости имеет ленточную структуру с четырьмя ненулевыми диагоналями, что в целом позволяет экономить ресурсы оперативной памяти и уменьшить время нахождения решения по сравнению с такими широко известными конечно-элементными программными комплексами, как КОМПАС-3D, ANSYS, SolidWorks [7, 10, 11].

### Список литературы

1. Старовойтов, Э.И. Деформирование трехслойных элементов конструкций на упругом основании / Э.И. Старовойтов, А.В. Яровая, Д.В. Леоненко. – М. : Физматлит, 2006. – 379 с.
2. Леоненко, Д.В. Собственные и вынужденные колебания трехслойных элементов конструкций, связанных с упругой средой / Д.В. Леоненко // Автореф. дис. ... докт. техн. наук : 01.02.04 – механика деформируемого твердого тела. – Минск : БНТУ, 2011. – 45 с.
3. Горшков, А.Г. Теория упругости и пластичности : учеб. для вузов / А.Г. Горшков, Э.И. Старовойтов, Д.В. Тарлаковский. – М. : Физматлит, 2002. – 416 с.

4. Андреев, А.Н. Многослойные анизотропные оболочки и пластины: изгиб, устойчивость, колебания / А.Н. Андреев, Ю.В. Немировский. – Новосибирск : Наука, 2001. – 288 с.
5. Быховцев, В.Е. Компьютерное моделирование прогиба диска перекрытия в структуре каркасного здания / В.Е. Быховцев, А.В. Быховцев, К.С. Курочка // Вестник ГГТУ им. П.О. Сухого. – 2001. – № 2. – С. 43–48.
6. Журавков, М.А. Математическое моделирование деформационных процессов в твердых деформируемых средах / М.А. Журавков. – Минск : БГУ, 2002. – 456 с.
7. Большаков, В.П. Основы 3D-моделирования. Изучаем работу в AutoCAD, КОМПАС-3D, SolidWorks, Inventor : учеб. курс / В.П. Большаков, А.Л. Бочков. – СПб. : Питер, 2013. – 304 с.
8. Быховцев, В.Е. Компьютерное моделирование систем нелинейной механики грунтов / В.Е. Быховцев, А.В. Быховцев, В.В. Бондарева. – Гомель : ГГУ им. Ф. Скорины, 2002. – 215 с.
9. Журавков, М. А. Математическое моделирование деформационных процессов в твердых деформируемых средах / М. А. Журавков. – Минск : БГУ, 2002. – 456 с.
10. Zienkiewicz, O.C. The finite element method for solid and structural mechanics. Sixth edition / O.C. Zienkiewicz, R.L. Taylor. – Oxford : Elsevier, 2005. – 631 p.
11. Курочка, К.С. Моделирование вязкоупругого деформирования неоднородных в плане тонких плит сложной конфигурации / К.С. Курочка // Инженерно-физический журнал. – 2008. – Т. 81, № 4. – С. 778–788.

Поступила 5.08.2013

*Гомельский государственный технический  
университет им. П.О. Сухого,  
Гомель, пр. Октября, 48  
e-mail: kurochka@gstu.by*

**K.S. Kurachka**

### **FINITE ELEMENT MODELING OF THIN CIRCULAR SANDWICH PLATES DEFLECTION**

A mathematical model of a thin circular sandwich plate being under the vertical load is proposed. The model employs the finite element method and takes advantage of an axisymmetric finite element that leads to the small dimension of the resulting stiffness matrix and sufficient accuracy for practical calculations. The analytical expressions for computing local stiffness matrices are found, which can significantly speed up the process of forming the global stiffness matrix and increase the accuracy of calculations. A software is under development and verification. The discrepancy between the results of the mathematical model and those of analytical formulas for homogeneous thin circular sandwich plates does not exceed 7%.

УДК 004.3

Л.И. Кульбак

## ИНТЕРВАЛЬНАЯ ОЦЕНКА РАСЧЕТНЫХ ПОКАЗАТЕЛЕЙ НАДЕЖНОСТИ ОБЪЕКТА

*Приводится методика интервальной оценки показателей надежности объекта, имеющего последовательную структурную схему расчета надежности и исходные данные для расчета показателей надежности в виде интервальных оценок. Методика базируется на модели косвенных измерений.*

### Введение

Термин «интервальная оценка показателей надежности» отсутствует в основных стандартах по терминологии в области надежности [1–2]. Воспользуемся определением интервальной оценки, приведенной в [3], для формулировки данного термина.

Интервальная оценка показателя надежности представляет собой числовой интервал  $[X_{\text{CP}} - \Delta X, X_{\text{CP}} + \Delta X]$ , в который с определенной вероятностью  $\alpha$  попадает истинное значение показателя надежности. В обозначенном интервале  $X_{\text{CP}}$  – среднее значение показателя надежности,  $\Delta X$  – отклонение показателя надежности от среднего значения.

Условимся интервал показателя надежности представлять в виде  $[X_{\text{H}}, X_{\text{CP}}, X_{\text{B}}]\alpha$ , где  $X_{\text{H}}$ ,  $X_{\text{B}}$  – соответственно нижнее и верхнее значения интервала, а  $\alpha$  – доверительная вероятность данного интервала.

Работы, посвященные интервальным оценкам показателей надежности, встречаются достаточно редко. Данная тематика не нашла отражения и в справочниках по надежности (например, в [4–6]). В опубликованных работах по этой тематике, как правило, оцениваются результаты испытаний на надежность элементов и технических объектов.

Авторы статьи [7] утверждают, что каких-либо работ, в которых непосредственно рассматривалась бы проблема учета в расчетах надежности неопределенности, вызванной наличием доверительных интервалов при оценивании надежности элементов по схеме Бернулли, им найти не удалось. Однако такие работы имеются. К наиболее фундаментальным из них следует отнести РД 50-476-84 «Методические указания. Надежность в технике. Интервальная оценка надежности технического объекта по результатам испытаний составных частей. Общие положения» [8], в разработке которого принимали участие академики АН СССР Н.Н. Шереметьевский и Н.Н. Моисеев. Методы [8] распространяются на объекты, для которых структурная схема надежности представляет собой все сочетания последовательного и параллельного соединения элементов, в том числе многофункциональные с монотонной структурой, и результаты испытаний элементов должны быть представлены как испытания по биномиальному плану [8]. Определение интервальной оценки показателей надежности сводится к построению односторонних доверительных границ.

В РД 50-476-84 под объектом с монотонной структурой понимается объект, надежность которого не улучшается при ухудшении надежности его элементов, а под биномиальными планами испытаний понимаются планы испытаний, при проведении которых не фиксируются моменты отказов, а определяется лишь количество отказавших элементов за время испытаний объекта.

К развитию методов интервальной оценки расчетных показателей надежности объектов следует отнести статью [9]. В ней описан статистический метод получения интервальных оценок показателей надежности структурно-сложной системы, не приводимой к последовательно-параллельной или параллельно-последовательной структуре.

В работе [4] рассматривается определение вычислительной погрешности расчета надежности в ограниченном числе структур невосстанавливаемых систем с элементами, имеющими одинаковые показатели надежности и одинаковые доверительные вероятности интервалов значений этих показателей.

- В настоящей работе приняты следующие допущения и ограничения:
- исследованию подлежат лишь два свойства надежности – безотказность и ремонтпригодность;
  - оценке подлежат следующие показатели надежности восстанавливаемых объектов:  $\Lambda$  – интенсивность отказов,  $T_0$  – средняя наработка на отказ,  $P(t)$  – вероятность безотказной работы,  $T_B$  – среднее время восстановления,  $K_\Gamma$  – коэффициент готовности,  $K_{O,\Gamma}$  – коэффициент оперативной готовности;
  - наработка на отказ (до отказа) объектов и комплектующих их составных частей (элементов) имеет экспоненциальное распределение;
  - время восстановления составных частей объекта распределено по экспоненциальному закону;
  - показатель надежности объекта можно представить как функцию, аргументами которой являются показатели надежности его составных частей;
  - значения показателей надежности составных частей объекта представлены в виде интервалов с достаточно малым отклонением от среднего значения;
  - доверительная вероятность интервальных оценок у всех составных частей объекта должна быть единой;
  - структурная схема расчета надежности объекта может быть приведена к последовательной.

### 1. Общие положения

Расчет надежности согласно [10] – это процедура определения значений показателей надежности объекта с использованием методов, основанных на их вычислении по справочным данным о надежности элементов объекта, о надежности объектов-аналогов, о свойствах материалов и другой информации, имеющейся к моменту расчета.

В качестве примера приведем формулы расчета показателей надежности при последовательной структурной схеме расчета надежности для восстанавливаемого объекта:

$$\Lambda = \sum_{i=1}^m \Lambda_i, \quad (1)$$

где  $\Lambda$  – интенсивность отказа объекта,  $m$  – количество типов элементов в объекте,  $\Lambda_i$  – интенсивность отказа  $i$ -го элемента структурной схемы надежности объекта;

$$T_0 = \frac{1}{\Lambda}, \quad (2)$$

где  $T_0$  – средняя наработка на отказ объекта;

$$P(t) = e^{-\Lambda t}, \quad (3)$$

где  $P(t)$  – вероятность безотказной работы объекта в течение наработки  $t$ ;

$$T_B = \sum_{i=1}^m \frac{T_{B,i} \Lambda_i}{\Lambda}, \quad (4)$$

где  $T_B$  – среднее время восстановления объекта,  $T_{B,i}$  – среднее время восстановления элемента  $i$ -го типа объекта;

$$K_\Gamma = \frac{T_0}{T_0 + T_B} = \frac{1}{1 + \Lambda T_B}, \quad (5)$$

где  $K_\Gamma$  – коэффициент готовности объекта;

$$K_{O,\Gamma} = K_\Gamma P(t), \quad (6)$$

где  $K_{0,Г}$  – коэффициент оперативной готовности объекта.

Из приведенных формул следует, что для оценки надежности восстанавливаемого объекта достаточно знать интенсивность отказов ( $\Lambda$ ) и среднее время восстановления ( $T_B$ ) объекта.

Существенной особенностью элементов объектов является то, что значения их показателей надежности известны лишь в виде интервальных оценок с установленной доверительной вероятностью. Такое положение обусловлено тем, что показатели надежности элементов находятся лишь в процессе испытаний при ограниченной их продолжительности.

Приведем примеры вычисления показателей надежности элементов в результате определительных испытаний или по данным реальной эксплуатации объекта.

В общем случае наработка элемента до отказа является случайной величиной, а показатели безотказности являются числовыми характеристиками случайной величины (математическим ожиданием и вероятностным значением). В принципе, в процессе эксперимента точно установить эти характеристики невозможно за время, меньшее бесконечности. Поэтому всегда следует использовать показатели, полученные с установленной точностью или находящиеся в определенном интервале.

Допустим, что оцениваются показатели надежности в виде средней наработки на отказ и интенсивности отказов элементов, объект подлежит испытаниям в течение наработки  $T_H$  и по исследуемому элементу зафиксировано  $m$  отказов. Оценку показателей надежности исследуемого элемента следует проводить по следующим формулам [11]:

$$T_{0,оп} = \frac{T_H}{m}; \quad (7)$$

$$T_{0,в} = r_1 T_{0,оп}; \quad (8)$$

$$T_{0,н} = r_2 T_{0,оп}, \quad m \neq 0; \quad (9)$$

$$T_{0,н} = \frac{T_H}{r_0}, \quad m = 0; \quad (10)$$

$$\Lambda_{оп} = \frac{1}{T_{0,оп}}; \quad (11)$$

$$\Lambda_H = \frac{\Lambda_{оп}}{r_1}; \quad (12)$$

$$\Lambda_B = \frac{\Lambda_{оп}}{r_3}; \quad (13)$$

$$r_1 = \frac{2m}{x_{1-\alpha}(2m)}; \quad (14)$$

$$r_2 = \frac{2m}{x_\alpha(2m+2)}; \quad (15)$$

$$r_3 = \frac{2m}{x_\alpha(2m)}; \quad (16)$$

$$r_0 = -\ln(1-\alpha), \quad (17)$$

где  $T_{0,оп}$  – опытное (точечное) значение средней наработки на отказ элемента;

$T_H$  – наработка элемента (объекта);

$m$  – число отказов элемента за наработку  $T_H$ ;

$\Lambda_{оп}$  – опытное (точечное) значение интенсивности отказов элемента;

$\Lambda_B, \Lambda_H$  – верхнее и нижнее значения интенсивности отказов соответственно;  
 $\chi_{1-\alpha}(2m), \chi_{\alpha}(2m+2), \chi_{\alpha}(2m)$  – квантили распределения  $\chi^2$ -квадрат, соответствующие вероятности  $(1-\alpha), \alpha, \alpha$  и числу степеней свободы  $2m, (2m+2)$  и  $2m$ .  
 Значения коэффициентов  $r_0, r_1, r_2, r_3$  даны в табл. 1–4.

Таблица 1  
 Значения коэффициента  $r_0$

$\alpha$	0,80	0,90	0,95
$r_0$	1,61	2,80	3,00

Таблица 2

Значения коэффициента  $r_1$ 

$\alpha$	Количество отказов $m$									
	1	2	3	4	5	6	8	10	15	20
0,8	4,48	2,42	1,95	1,74	1,62	1,54	1,43	1,37	1,28	1,24
0,9	9,50	3,33	2,73	2,29	2,05	1,90	1,72	1,61	1,46	1,37

Таблица 3

Значения коэффициента  $r_2$ 

$\alpha$	Количество отказов $m$									
	1	2	3	4	5	6	8	10	15	20
0,8	0,33	0,47	0,55	0,60	0,63	0,66	0,70	0,73	0,78	0,80
0,9	0,26	0,38	0,45	0,50	0,54	0,57	0,62	0,65	0,70	0,74

Таблица 4

Значения коэффициента  $r_3$ 

$\alpha$	Количество отказов $m$									
	1	2	3	4	5	6	8	10	15	20
0,8	0,62	0,67	0,70	0,73	0,75	0,76	0,78	0,80	0,83	0,85
0,9	0,43	0,51	0,57	0,60	0,62	0,65	0,68	0,70	0,74	0,77

В результате определительных испытаний по наработке на отказ получают опытное (точечное) значение и доверительный интервал. При допущении экспоненциального распределения наработки на отказ опытное (точечное) значение соответствует средней наработке на отказ.

Следует отметить, что имеется принципиальная возможность пересчета интервала показателя надежности по данным его граничных значений и доверительной вероятности  $\alpha_1$  в интервал показателя надежности с другой доверительной вероятностью  $\alpha_2$ . С этой целью определяется опытное (точечное) значение показателя надежности по формуле

$$T_{0,оп} = T_{0,н} + \frac{T_{0,в} - T_{0,н}}{2}. \quad (18)$$

Затем из формулы (8) определяется

$$r_1 = \frac{T_{0,в}}{T_{0,оп}}, \quad (19)$$

а из таблицы коэффициентов  $r_1$  – значение числа отказов  $m$ , по которым находим интервал показателя надежности для доверительной вероятности  $\alpha_1$ . Зная значения  $T_{0,оп}$  и число отказов  $m$ , по формулам (8) и (9) определяются граничные значения интервала для доверительной вероятности  $\alpha_2$ .

## 2. Модель интервальной оценки показателей надежности объекта

Модель расчета показателей надежности объекта можно представить как косвенное вычисление функции ряда измеряемых аргументов. В качестве функции выступает формула расчета определенного показателя надежности объекта с аргументами в виде показателей надежности ее компонентов.

Косвенные измерения регламентируются нормативным документом [12].

Согласно [12] искомое значение физической величины  $A$  находят на основании результатов измерений аргументов  $a_1, \dots, a_i, \dots, a_m$ , связанных с искомой величиной уравнением

$$A = f(a_1, a_2, \dots, a_m). \quad (20)$$

Функция  $f$  должна быть известна из теоретических предпосылок или установлена экспериментально с погрешностью, которой можно пренебречь.

Результаты измерений аргументов и оценки их погрешностей могут быть получены из прямых, косвенных, совокупных и совместных измерений. Сведения об аргументах могут быть взяты из справочной литературы и технической документации.

Основные положения [12] устанавливаются для оценивания косвенно измеряемой величины и погрешностей результата измерения при линейной зависимости и отсутствии корреляции между погрешностями измерений аргументов и при нелинейной зависимости и отсутствии корреляции между погрешностями измерений аргументов.

Рассмотрим косвенные измерения при линейной зависимости, когда искомое значение  $A$  связано с  $m$  измеряемыми аргументами  $a_1, a_2, \dots, a_m$  уравнением

$$A = b_1 a_1 + b_2 a_2 + \dots + b_m a_m, \quad (21)$$

где  $b_1, b_2, \dots, b_m$  – постоянные коэффициенты при аргументах  $a_1, a_2, \dots, a_m$  соответственно.

*Примечание [12].* Если коэффициенты  $b_1, b_2, \dots, b_m$  определяют экспериментально, то задачу определения результата измерения величины решают поэтапно: сначала оценивают каждое слагаемое  $b_i a_i$  как косвенно измеряемую величину, полученную в результате произведения двух измеряемых величин, а потом находят оценку измеряемой величины  $A$ .

Результат косвенного измерения функции  $A$  вычисляют по формулам

$$A = A_{\text{CP}} \pm \Delta A; \quad (22)$$

$$A_{\text{CP}} = \sum_{i=1}^m b_i a_{\text{CP},i}; \quad (23)$$

$$\Delta A = \sum_{i=1}^m b_i \Delta a_i, \quad (24)$$

где  $A_{\text{CP}}$  – среднее значение измеряемой величины  $A$ ;

$\Delta A$  – отклонение измеряемой величины  $A$  от среднего значения (погрешность косвенного измерения);

$a_{\text{CP},i}$  – среднее значение  $i$ -го аргумента;

$\Delta a_i$  – отклонение значения  $i$ -го аргумента от среднего значения (погрешность измерения  $i$ -го аргумента).

Рассмотрим косвенные измерения при нелинейной зависимости. Согласно [12] при отсутствии корреляции между аргументами косвенно измеряемую функцию можно линеаризовать методом разложения в ряд Тейлора в виде

$$\Lambda = f(\lambda_1, \lambda_2, \dots, \lambda_m) = f(\lambda_1^*, \lambda_2^*, \dots, \lambda_m^*) \pm \sum_{i=1}^m \frac{df}{d\lambda_i} \Delta \lambda_i + R, \quad (25)$$

где  $f(\lambda_1, \lambda_2, \dots, \lambda_m)$  – функциональная зависимость измеряемой величины  $\Lambda$  от измеряемых аргументов  $\lambda_i$ ;

$\lambda_1^*, \lambda_2^*, \dots, \lambda_m^*$  – средние значения измеренных аргументов;

$\frac{df}{d\lambda_i}$  – первая производная от функции  $f$  по аргументу  $\lambda_i$ , вычисленная в точке  $\lambda_1^*, \lambda_2^*, \dots, \lambda_m^*$ ;

$\Delta\lambda_i$  – отклонение измеренного значения аргумента  $\lambda_i$  от его среднего значения (погрешность измеренного аргумента);

$R$  – остаточный член ряда (он мал относительно значения функции, и в связи с малостью погрешностей аргументов по отношению к их значениям им можно пренебречь).

Установлено, что корреляция между аргументами чаще всего возникает в тех случаях, когда их измерения проводятся одновременно и подвергаются одинаковому влиянию внешних условий (температуры, влажности, напряжения питающей сети, помех и т. п.), что не характерно при определении показателей надежности элементов.

Заметим, что абсолютную погрешность функции  $\Lambda$  можно вычислять по формуле Лагранжа [13]:

$$\Delta\Lambda = |\Lambda - \Lambda_{\text{CP}}| \leq \sum_{i=1}^m \left| \frac{df}{d\lambda_i} \right| \Delta\lambda_i, \quad (26)$$

где  $\Delta\Lambda$  – абсолютная погрешность функции  $\Lambda$ ;  $\Lambda_{\text{CP}}$  – среднее значение функции  $\Lambda$ ;  $\Delta\lambda_i$  – абсолютная погрешность значения аргумента  $\lambda_i$ .

В соответствии с [12] косвенно измеряемая функция  $\Lambda$  вычисляется по формулам

$$\Lambda = \Lambda_{\text{CP}} \pm \Delta\Lambda; \quad (27)$$

$$\Lambda_{\text{CP}} = f(\lambda_1^*, \lambda_2^*, \dots, \lambda_m^*), \quad (28)$$

где  $\Lambda_{\text{CP}}$  – среднее значение косвенно измеряемой функции  $\Lambda$ ;

$$\Delta\Lambda = \sum_{i=1}^m \frac{df}{d\lambda_i} \Delta\lambda_i, \quad (29)$$

где  $\Delta\Lambda$  – погрешность косвенно измеряемой функции  $\Lambda$ .

### 3. Методика интервальной оценки показателей безотказности объекта

Распространим положения по косвенным измерениям на расчет показателей надежности объекта при наличии интервальных оценок показателей надежности его составных частей с заданным единым уровнем доверительной вероятности  $\alpha$ .

Для оценки показателей надежности объекта в нормативной документации рекомендуется разработать структурную схему расчета надежности.

Установлено, что подавляющее число структур объектов, показатели надежности которых подлежат расчету, можно привести к последовательной схеме расчета надежности (рис. 1).



Рис. 1. Обобщенная структурная схема расчета надежности объекта

В соответствии с рис. 1 и согласно формуле (1) интенсивность отказов объекта  $\Lambda$  в этом случае следует вычислять по формуле

$$\Lambda = \sum_{i=1}^m \Lambda_{\mathcal{E}_i}, \quad (30)$$

где  $\Lambda_{\mathcal{E}_i}$  – интенсивность отказов  $i$ -го компонента структурной схемы надежности объекта.

Функция  $\Lambda$  будет линейной, если будут линейными ее компоненты. Для линеаризации формулы (30) потребуется линеаризация формул всех ее компонентов.

После линеаризации компоненты формулы (30) примут следующий вид:

$$\Lambda_{\Delta,i} = \Lambda_{\text{CP},\Delta,i} + \Delta\Lambda_{\Delta,i}, \quad i = 1, 2, \dots, m; \quad (31)$$

$$\Lambda_{\text{CP},\Delta,i} = f_i(\lambda_1^*, \lambda_2^*, \dots, \lambda_{m,i}^*), \quad i = 1, 2, \dots, m; \quad (32)$$

$$\Delta\Lambda_{\Delta,i} = \sum_{j=1}^{m_i} \frac{df_i}{d\lambda_{i,j}} \Delta\lambda_{i,j}, \quad i = 1, 2, \dots, m, \quad (33)$$

где  $\Delta\Lambda_{\Delta,i}$  – абсолютная погрешность компонентов  $\Lambda_{\Delta,i}$ ;

$\Lambda_{\text{CP},\Delta,i}$  – среднее значение интенсивности отказов компонентов  $\Lambda_{\Delta,i}$ ;

$\Delta\lambda_{i,j}$  – абсолютная погрешность значения аргумента  $\lambda_{i,j}$ ;

$f_i$  – функция  $i$ -го компонента структурной схемы расчета надежности объекта;

$m_i$  – количество аргументов в функции.

Формула (30) может иметь достаточно много разнообразных компонентов. В табл. 5 приведены наиболее часто встречаемые компоненты нерезервируемых систем. Резервируемые системы подлежат отдельному рассмотрению ввиду сложности изложения.

Таблица 5

Часто встречаемые компоненты нерезервируемых систем

Функция компонента	Среднее значение	Погрешность
$n\Lambda^*$	$n\Lambda_{\text{CP}}$	$\Delta\Lambda$
$n\Lambda^* \kappa^* \text{ и } \varepsilon^*$	$n\Lambda_{\text{CP}} \kappa_{\text{CP},\text{и}\varepsilon}$	$n(\kappa_{\text{CP},\text{и}\varepsilon} \Delta\Lambda + \Lambda_{\text{CP}} \Delta\kappa_{\text{и}\varepsilon})$
$\frac{n}{T_0^*}$	$\frac{n}{T_{0,\text{CP}}}$	$\frac{n\Delta T_0}{T_0^2}$

Интервальная оценка интенсивности отказов объекта с последовательной схемой расчета надежности (см. рис. 1) вычисляется по формулам

$$\Lambda_{\text{CP,OB}} = \sum_{i=1}^m \Lambda_{\text{CP},\Delta,i}; \quad (34)$$

$$\Delta\Lambda_{\text{OB}} = \sum_{i=1}^m \Delta\Lambda_{\Delta,i}; \quad (35)$$

$$\Lambda_{\text{OB,Н}} = \Lambda_{\text{CP,OB}} - \Delta\Lambda_{\text{OB}}; \quad (36)$$

$$\Lambda_{\text{OB,В}} = \Lambda_{\text{CP,OB}} + \Delta\Lambda_{\text{OB}}. \quad (37)$$

В соответствии с формулами (2), (28) и (29) интервальная оценка средней наработки на отказ объекта вычисляется по формулам

$$T_{0,\text{CP,OB}} = \frac{1}{\Lambda_{\text{CP,OB}}}; \quad (38)$$

$$\Delta T_{0,\text{OB}} = \frac{\Delta\Lambda_{\text{OB}}}{\Lambda_{\text{OB}}^2}; \quad (39)$$

$$T_{\text{OB,Н}} = 1 / \Lambda_{\text{CP,OB}} - \Delta T_{0,\text{OB}}; \quad (40)$$

$$T_{\text{OB,В}} = 1 / \Lambda_{\text{CP,OB}} + \Delta T_{0,\text{OB}}. \quad (41)$$

В соответствии с формулами (3), (28) и (29) интервальная оценка вероятности безотказной работы объекта вычисляется по формулам

$$P(t)_{\text{ОБ,СР}} = e^{-\Lambda_{\text{СР,ОБ}}t}; \quad (42)$$

$$\Delta P(t)_{\text{ОБ}} = e^{-\Lambda_{\text{СР,ОБ}}t} t \Delta \Lambda_{\text{ОБ}}; \quad (43)$$

$$P(t)_{\text{ОБ,Н}} = e^{-\Lambda_{\text{СР,ОБ}}t} (1 - t \Delta \Lambda_{\text{ОБ}}); \quad (44)$$

$$P(t)_{\text{ОБ,В}} = e^{-\Lambda_{\text{СР,ОБ}}t} (1 + t \Delta \Lambda_{\text{ОБ}}). \quad (45)$$

#### 4. Методика интервальной оценки показателя ремонтпригодности и комплексных показателей

В соответствии с формулами (4), (28) и (29) интервальная оценка среднего времени восстановления объекта вычисляется по формулам

$$T_{\text{В,СР,ОБ}} = \sum_{i=1}^m \frac{T_{\text{В,СР,Э},i} \Lambda_{\text{СР,Э},i}}{\Lambda_{\text{ОБ}}}; \quad (46)$$

$$\Delta T_{\text{В,ОБ}} = \sum_{i=1}^m \frac{\Lambda_{\text{СР,Э},i} \Delta T_{\text{В,СР,Э},i}}{\Lambda_{\text{ОБ}}}; \quad (47)$$

$$T_{\text{В,ОБ,Н}} = T_{\text{В,СР,ОБ}} - \Delta T_{\text{В,ОБ}}; \quad (48)$$

$$T_{\text{В,ОБ,В}} = T_{\text{В,СР,ОБ}} + \Delta T_{\text{В,ОБ}}. \quad (49)$$

В соответствии с формулами (5), (28) и (29) интервальная оценка коэффициента готовности объекта вычисляется по формулам

$$K_{\text{Г,ОБ,СР}} = \frac{T_{\text{О,ОБ,СР}}}{T_{\text{О,ОБ,СР}} + T_{\text{В,ОБ,СР}}}; \quad (50)$$

$$\Delta K_{\text{Г,ОБ}} = \frac{2\Delta T_{\text{О,ОБ}}}{(T_{\text{О,ОБ,СР}} + T_{\text{В,ОБ,СР}})} + \frac{T_{\text{О,ОБ,СР}} \Delta T_{\text{В,ОБ}}}{(T_{\text{О,ОБ,СР}} + T_{\text{В,ОБ,СР}})^2}; \quad (51)$$

$$K_{\text{Г,ОБ,Н}} = K_{\text{Г,СР,ОБ}} - \Delta K_{\text{Г,ОБ}}; \quad (52)$$

$$K_{\text{Г,ОБ,В}} = K_{\text{Г,СР,ОБ}} + \Delta K_{\text{Г,ОБ}}. \quad (53)$$

В соответствии с формулами (6), (28) и (29) интервальная оценка коэффициента оперативной готовности объекта вычисляется по формулам

$$K_{\text{О,Г,ОБ,СР}} = K_{\text{Г,ОБ,СР}} P(t)_{\text{ОБ,СР}}; \quad (54)$$

$$\Delta K_{\text{О,Г,ОБ}} = K_{\text{Г,ОБ,СР}} \Delta P(t)_{\text{ОБ}} + P(t)_{\text{ОБ,СР}} \Delta K_{\text{Г,ОБ}}; \quad (55)$$

$$K_{\text{О,Г,ОБ,Н}} = K_{\text{О,Г,СР,ОБ}} - \Delta K_{\text{О,Г,ОБ}}; \quad (56)$$

$$K_{\text{О,Г,ОБ,В}} = K_{\text{О,Г,СР,ОБ}} + \Delta K_{\text{О,Г,ОБ}}. \quad (57)$$

## 5. Алгоритм интервальной оценки показателей надежности объекта

Для интервальной оценки показателей надежности объекта рекомендуется следующий алгоритм:

*Шаг 1.* Составить последовательную структурную схему расчета надежности объекта.

*Шаг 2.* Записать формулы расчета интенсивности отказов каждого элемента структурной схемы.

*Шаг 3.* Выписать исходные данные, необходимые для расчета интенсивности отказов каждого элемента структурной схемы.

*Шаг 4.* Собрать для всех исходных данных интервальные оценки и привести их к единому значению доверительной вероятности.

*Шаг 5.* По формулам (32) и (33) определить средние значения интенсивностей отказов и отклонения от средних значений для всех элементов структурной схемы.

*Шаг 6.* По формулам (34) – (57) определить интервальные оценки всех показателей надежности объекта.

## Заключение

Приведенная методика интервальной оценки показателей надежности объекта, имеющего последовательную структурную схему расчета надежности и исходные данные для расчета показателей надежности в виде интервальных оценок, базируется на модели косвенных измерений и публикуется впервые в данной работе. В качестве измеряемой величины выступает показатель надежности объекта, представленный формулой, аргументами которой являются показатели надежности составных частей объекта, имеющих интервальную оценку.

Работа будет продолжена в направлении линеаризации формул резервируемых структур.

## Список литературы

1. Надежность в технике. Основные понятия. Термины и определения : ГОСТ 27.002–89. – М. : Изд-во стандартов, 1990. – 37 с.
2. Надежность в технике. Основные понятия. Термины и определения : ГОСТ Р 27.002–2009. – М. : Изд-во стандартов, 2009. – 75 с.
3. Теория погрешностей. Измерение физических величин [Электронный ресурс]. – 2013. – Режим доступа : <http://rudocs.exdat.com/docs/index-27707.html>. – Дата доступа : 20.07.2013.
4. Козлов, Б.А. Краткий справочник по расчету надежности радиоэлектронной аппаратуры / Б.А. Козлов, И.А. Ушаков. – М. : Советское радио, 1966. – 492 с.
5. Справочник. Надежность технических систем / под ред. проф. И.А. Ушакова. – М. : Радио и связь, 1985. – 606 с.
6. Надежность и эффективность в технике : справ. в 10 т. / под ред. д.т.н. А.И. Рембезы. – М. : Машиностроение, 1986.
7. Дубницкий, В.Ю. Определение вычислительной погрешности расчета надежности [Электронный ресурс] / В.Ю. Дубницкий, А.М. Кобылин, А.И. Ходырев. – 2013. – Режим доступа : <http://www.khai.csp/nauchpotal/Arhiv/REKS712/jubnits>. – Дата доступа : 20.07.2013.
8. Методические указания. Надежность в технике. Интервальная оценка надежности технического объекта по результатам испытаний составных частей. Общие положения : РД 50-476-84. – М. : Изд-во стандартов, 1985. – 87 с.
9. Струков, А.В. Интервальная оценка показателей надежности структурно-сложной системы по результатам биномиальных испытаний ее компонентов / А.В. Струков [Электронный ресурс]. – Режим доступа : [http://www.szma.com/strukov%27s\\_lecture.pdf](http://www.szma.com/strukov%27s_lecture.pdf). – Дата доступа : 29.07.2013.
10. Надежность в технике. Расчет надежности. Общие положения : ГОСТ 27.301–95. Минск : Белорус. гос. ин-т стандартизации и сертификации, 1997. – 16 с.
11. Шор, Я.Б. Таблицы для анализа и контроля надежности / Я.Б. Шор, Ф.И. Кузьмин. – М. : Советское радио, 1968. – 289 с.

12. Рекомендация. Государственная система обеспечения единства измерений. Измерения косвенные. Определение результатов измерений и оценивание их погрешностей : МИ 2083–90. – М. : Комитет стандартизации и метрологии СССР, 1991.

13. Основы теории погрешности. Лекция 1[Электронный ресурс]. – Режим доступа : [http://wiki.auditory.ru/Лекция\\_1:\\_Основы\\_теории\\_погрешностей](http://wiki.auditory.ru/Лекция_1:_Основы_теории_погрешностей). – Дата доступа : 29.07.2013.

Поступила 23.07.13

*Объединенный институт проблем  
информатики НАН Беларуси,  
Минск, Сурганова,6  
e-mail: lkulbak@yandex.ru*

**L.I. Kulbak**

### **INTERVAL ESTIMATION OF OBJECT RELIABILITY INDICES**

The paper describes a method of interval estimation of reliability indices of an object having a consistent block diagram of reliability calculation and input data for calculation of reliability indices in the form of interval estimates. The technique is based on the model of indirect measurements.

УДК 517.958:537.8

В.Т. Ерофеевко

## ЭЛЕКТРОДИНАМИЧЕСКАЯ МОДЕЛЬ РАСЧЕТА ЭФФЕКТИВНЫХ ПАРАМЕТРОВ КОМПОЗИТОВ ИЗ СФЕРИЧЕСКИХ БИЗОТРОПНЫХ ЧАСТИЦ

*Разрабатывается метод вычисления эффективных материальных параметров кирального композита, который состоит из разреженной системы дискретных рассеивателей, заполненных материалом с произвольными комплексными диэлектрической и магнитной проницаемостями и комплексными параметрами биизотропности. В полученных расчетных аналитических формулах используется точное аналитическое решение задачи дифракции на биизотропной сфере электромагнитных волн с длиной волны, значительно превосходящей диаметр частиц.*

### Введение

В настоящее время определяющее значение в научных исследованиях электродинамики приобретают исследования электродинамических свойств композитных материалов различных категорий [1, 2]. Композитные материалы представляют собой структурно неоднородную среду с большим разнообразием структурных элементов, отличающихся геометрией, химическим составом и уровнем линейных размеров. К ним относятся метаматериалы [3, 4], киральные среды [5], квадрупольные материалы [1], наноструктурные композиты [6] и др. Моделирование таких композитов сводится к эквивалентной замене структурно неоднородных материалов однородными средами [1, 7]. При этом выделяются основные материальные параметры идеальной среды, называемые эффективными параметрами. Разработан ряд методов моделирования эффективных параметров: метод Максвелла – Гарнетта для композитов, когда длина волны значительно больше размеров неоднородностей, а объемные коэффициенты заполнения малы [8]; метод интегральных уравнений Поклингтона для композитов с проволочными включениями [9–11]; методика [12, 13] для киральных композитов со сферическими частицами из магнитоэлектрических материалов с винтовой проводимостью поверхности. Обсуждаются подходы к моделированию эффективных параметров [14].

В настоящей работе предложен аналитический метод моделирования композитов, которые состоят из однородных сферических биизотропных частиц одинаковых радиусов, случайно распределенных в вакууме. Используется методика усреднения электрических и магнитных индукций и полей по неоднородной области больших размеров при воздействии на структуры плоской электромагнитной волны.

### 1. Композит из биизотропных частиц

В матрице, заполненной вакуумом с электрической и магнитной постоянными  $\epsilon_0, \mu_0$ , случайным образом размещено большое число биизотропных сферических частиц радиуса  $R$ , характеризуемых материальными параметрами  $\epsilon, \mu, G, Z$ .

Выделим частицу  $D_R(0 \leq r < R)$  с центром  $O$  в начале декартовых координат  $Oxuz$  ( $Or\theta\varphi$  – сферические координаты). Обозначим области:  $D_P(0 \leq r < P)$  – шаровая область достаточно большого радиуса  $P$ ;  $D_{RP}(R < r < P)$  – шаровой слой вокруг частицы  $D_R$ ;  $D_0$  – область между частицами,  $D_0 \subset D_P$ ;  $D_S$  – область внутри  $S$ -частицы,  $D_P = \bigcup_{S=1}^N D_S \cup D_0$ ,  $D_1 = D_R$ .

Будем предполагать, что

$$R \ll P \ll \lambda, \quad (1)$$

где  $\lambda$  – длина электромагнитной волны  $\vec{E}_0, \vec{H}_0$ , распространяющейся в вакууме;  $f \leq 10$  ГГц,  $R \leq 4$  мкм,  $P \approx 30R$ . Введем обозначения:  $N$  – число частиц в области  $D_P$ ;  $V_R = \frac{4\pi}{3}R^3$  – объем частицы,  $V_P$  – объем области  $D_P$ ;  $\nu = \frac{N}{V_P}$  – концентрация частиц (число частиц в единице объема матрицы),  $\tau = \nu V_R$  – объемный коэффициент заполнения матрицы сферическими частицами (суммарный объем сферических частиц в единице объема матрицы).

Построим математическую модель композита из биизотропных частиц, используя теорию биизотропных сред, электромагнитные поля в которых описываются уравнениями

$$\text{rot } \vec{E} = i\omega \vec{D}; \text{rot } \vec{H} = -i\omega \vec{D}, \quad (2)$$

где  $\vec{B} = \mu_{\text{эф}} \vec{H} + Z_{\text{эф}} \vec{E}$ ,  $\vec{D} = \varepsilon_{\text{эф}} \vec{E} + G_{\text{эф}} \vec{H}$ ,  $\omega = 2\pi f$ .

Проблема состоит в вычислении эффективных параметров  $\varepsilon_{\text{эф}}, \mu_{\text{эф}}, G_{\text{эф}}, Z_{\text{эф}}$ . Для построения модели композит подвергается воздействию электромагнитного поля  $\vec{E}_0, \vec{H}_0$  и исследуется отраженное поле. Сформируем вспомогательную краевую задачу дифракции поля  $\vec{E}_0, \vec{H}_0$  на изолированной частице  $D_R$ . Обозначим:  $D'_R (R < r < \infty)$  – внешняя к шару  $D_R$  область,  $\Gamma (r = R)$  – сферическая поверхность шара. В  $D'_R$  распространяется первичное электромагнитное поле  $\vec{E}_0, \vec{H}_0$ , колеблющееся с круговой частотой  $\omega$  и временной зависимостью  $\exp(-i\omega t)$ . Обозначим поля:  $\vec{E}_R^0, \vec{H}_R^0$  – поле внутри шара  $D_R$ ;  $\vec{E}'_R, \vec{H}'_R$  – отраженное поле в  $D'_R$ ;  $\vec{E}_2 = \vec{E}_0 + \vec{E}'_R, \vec{H}_2 = \vec{H}_0 + \vec{H}'_R$  – суммарное поле в области  $D'_R$ .

**Краевая задача.** Требуется для заданного первичного поля  $\vec{E}_0, \vec{H}_0$  определить электромагнитные поля  $\vec{E}_R^0, \vec{H}_R^0 \in C^1(D_R) \cap C(\bar{D}_R)$ ,  $\vec{E}'_R, \vec{H}'_R \in C^1(D'_R) \cap C(\bar{D}'_R)$ , которые удовлетворяют уравнениям Максвелла

$$\text{rot } \vec{E}_R^0 = i\omega(\mu \vec{H}_R^0 + Z \vec{E}_R^0), \text{rot } \vec{H}_R^0 = -i\omega(\varepsilon \vec{E}_R^0 + G \vec{H}_R^0), 0 \leq r < R, \quad (3)$$

$$\text{rot } \vec{E}'_R = i\omega \mu_0 \vec{H}'_R, \text{rot } \vec{H}'_R = -i\omega \varepsilon_0 \vec{E}'_R, r > R; \quad (4)$$

граничным условиям

$$\vec{E}_{R\tau}^0 \Big|_{r=R} = \vec{E}_{2\tau} \Big|_{r=R}, \vec{H}_{R\tau}^0 \Big|_{r=R} = \vec{H}_{2\tau} \Big|_{r=R} \quad (5)$$

и условию излучения на бесконечности

$$\lim_{r \rightarrow \infty} r \left( \frac{\partial E'_R}{\partial r} - ik_0 E'_R \right) = 0. \quad (6)$$

## 2. Решение задачи дифракции на биизотропной сферической частице

В качестве первичного поля, действующего на частицы, выберем монохроматическое электромагнитное поле, которое представим в виде рядов по базисным сферическим полям [15, с. 188]:

$$\begin{aligned} \vec{E}_0 &= \sum_{n,m} [a_{nm} \vec{m}_{nm}(\vec{r}; k_0) + b_{nm} \vec{n}_{nm}(\vec{r}; k_0)], \\ \vec{H}_0 &= h_0 \sum_{n,m} [a_{nm} \vec{n}_{nm}(\vec{r}; k_0) + b_{nm} \vec{m}_{nm}(\vec{r}; k_0)], \end{aligned} \quad (7)$$

где  $a_{mn}, b_{mn}$  – заданные коэффициенты разложений,  $\sum_{n,m} = \sum_{n=1}^{\infty} \sum_{m=-n}^n$ ,  $h_0 = \frac{k_0}{i\omega\mu_0}$ ,  $k_0 = \frac{\omega}{c}$ ,  
 $c$  – скорость света в вакууме;  $\vec{n}_{mn} = f_n(k_0 r) Y_n^m(\theta, \varphi) \vec{e}_r + g_n(k_0 r) \vec{\Pi}_{mn}(\theta, \varphi)$ ,  
 $\vec{m}_{mn} = j_n(k_0 r) \vec{T}_{mn}(\theta, \varphi)$ ,  $Y_n^m = P_n^m(\cos\theta) e^{im\varphi}$ ,  $f_n(k_0 r) = \frac{n(n+1)}{k_0 r} j_n(k_0 r)$ ,  $g_n(k_0 r) = \frac{1}{k_0 r} \frac{d}{dr} (r j_n(k_0 r))$ ,  
 $\vec{\Pi}_{mn} = \left( \frac{d}{d\theta} P_n^m(\cos\theta) \vec{e}_\theta + \frac{im}{\sin\theta} P_n^m(\cos\theta) \vec{e}_\varphi \right) e^{im\varphi}$ ,  $\vec{T}_{mn} = \left( \frac{im}{\sin\theta} P_n^m(\cos\theta) \vec{e}_\theta - \frac{d}{d\theta} P_n^m(\cos\theta) \vec{e}_\varphi \right) e^{im\varphi}$ ,  
 $n = 1, 2, \dots, m = 0, \pm 1, \pm 2, \dots, \pm n$ ;  $j_n(x)$  – сферические функции Бесселя.

**Лемма 1.** Значение любого поля вида (7) в начале координат определяется формулами

$$\begin{aligned} \vec{E}_0(0) &= \vec{E}_0^0 = E_x^0 \vec{e}_x + E_y^0 \vec{e}_y + E_z^0 \vec{e}_z = E_+^0 \vec{e}_+ + E_-^0 \vec{e}_- + E_z^0 \vec{e}_z = \\ &= \sum_{m=-1}^1 b_m \vec{n}_{m1}(0) = -\frac{2}{3} b_{11} \vec{e}_+ - \frac{1}{3} b_{-11} \vec{e}_- + \frac{2}{3} b_{01} \vec{e}_z, \\ \vec{H}_0(0) &= \vec{H}_0^0 = H_x^0 \vec{e}_x + H_y^0 \vec{e}_y + H_z^0 \vec{e}_z = H_+^0 \vec{e}_+ + H_-^0 \vec{e}_- + H_z^0 \vec{e}_z = \\ &= h_0 \sum_{m=-1}^1 a_m \vec{n}_{m1}(0) = h_0 \left( -\frac{2}{3} a_{11} \vec{e}_+ - \frac{1}{3} a_{-11} \vec{e}_- + \frac{2}{3} a_{01} \vec{e}_z \right), \end{aligned} \quad (8)$$

где  $\vec{e}_+ = i\vec{e}_y + \vec{e}_x$ ,  $\vec{e}_- = i\vec{e}_y - \vec{e}_x$ ,  $a_{01} = \frac{3}{2h_0} H_z^0$ ,  $a_{11} = \frac{3}{4h_0} (iH_y^0 - H_x^0)$ ,  $a_{-11} = \frac{3}{2h_0} (iH_y^0 + H_x^0)$ ,

$$b_{01} = \frac{3}{2} E_z^0, \quad b_{11} = \frac{3}{4} (iE_y^0 - E_x^0), \quad b_{-11} = \frac{3}{2} (iE_y^0 + E_x^0).$$

Доказательство следует из формул  $\vec{n}_{mn}(0) = 0$  при  $n = 2, 3, \dots$ ;  $\vec{n}_{01}(0) = \frac{2}{3} \vec{e}_z$ ,  
 $\vec{n}_{11}(0) = -\frac{2}{3} \vec{e}_+$ ,  $\vec{n}_{-11}(0) = -\frac{1}{3} \vec{e}_-$ ,  $\vec{m}_{mn}(0) = 0$  при  $n = 1, 2, \dots$ ;  $m = 0, \pm 1, \dots, \pm n$ . ■

Отраженное электромагнитное поле выразим через базисные сферические поля, которые удовлетворяют условию излучения (6) и уравнениям (4):

$$\begin{aligned} \vec{E}'_R &= \sum_{n,m} \left[ x_{mn}^{(2)} \vec{\tilde{m}}_{mn}(\vec{r}, k_0) + y_{mn}^{(2)} \vec{\tilde{n}}_{mn}(\vec{r}, k_0) \right], \quad r > R, \\ \vec{H}'_R &= h_0 \sum_{n,m} \left[ x_{mn}^{(2)} \vec{\tilde{n}}_{mn}(\vec{r}, k_0) + y_{mn}^{(2)} \vec{\tilde{m}}_{mn}(\vec{r}, k_0) \right], \end{aligned} \quad (9)$$

где  $\vec{\tilde{n}}_{mn} = f_n^{(1)}(k_0 r) Y_n^m(\theta, \varphi) \vec{e}_r + g_n^{(1)}(k_0 r) \vec{\Pi}_{mn}(\theta, \varphi)$ ,  $\vec{\tilde{m}}_{mn} = h_n^{(1)}(k_0 r) \vec{T}_{mn}(\theta, \varphi)$ ,

$$f_n^{(1)}(k_0 r) = \frac{n(n+1)}{k_0 r} h_n^{(1)}(k_0 r), \quad g_n^{(1)}(k_0 r) = \frac{1}{k_0 r} \frac{d}{dr} (r h_n^{(1)}(k_0 r)),$$

$h_n^{(1)}(x)$  – сферические функции Ханкеля.

Электромагнитное поле внутри шара  $D_R$  представим через базисные сферические поля в композитных средах, которые удовлетворяют уравнениям (3) [15, с. 122]:

$$\begin{aligned}\bar{E}_R^0 &= \sum_{n,m} \left[ x_{mn}^{(1)} \bar{K}_{mn}^{(1)}(\bar{r}, k_1) + y_{mn}^{(1)} \bar{K}_{mn}^{(2)}(\bar{r}, k_2) \right], \quad 0 \leq r < R, \\ \bar{H}_R^0 &= \sum_{n,m} \left[ x_{mn}^{(1)} p_1 \bar{K}_{mn}^{(1)}(\bar{r}, k_1) + y_{mn}^{(1)} p_2 \bar{K}_{mn}^{(2)}(\bar{r}, k_2) \right],\end{aligned}\tag{10}$$

где коэффициенты  $x_{mn}^{(1)}$ ,  $y_{mn}^{(1)}$ ,  $x_{mn}^{(2)}$ ,  $y_{mn}^{(2)}$  определены в работе [15, с. 275];

$$\bar{K}_{mn}^{(j)} = \bar{n}_{mn}(\bar{r}, k_j) - q_j \bar{m}_{mn}(\bar{r}, k_j),$$

$$k_j = \sqrt{g + \frac{1}{2}a^2 + af_j}, \quad 0 \leq \arg k_j < \pi, \quad q_j = \frac{g}{k_j g_j},$$

$$p_j = \frac{1}{\mu} \left( \frac{ig}{\omega g_j} - Z \right), \quad f_j = (-1)^j f_0, \quad f_0 = \sqrt{\omega^2 \epsilon \mu - b^2},$$

$$0 \leq \arg f_0 < \pi, \quad g = \omega^2(\epsilon \mu - ZG), \quad g_j = f_j - \frac{1}{2}a,$$

$$a = i\omega(G - Z), \quad b = \frac{\omega}{2}(G + Z), \quad j = 1, 2.$$

Удовлетворяя граничным условиям (5), получим систему алгебраических уравнений относительно  $x_{mn}^{(1)}$ ,  $y_{mn}^{(1)}$ ,  $x_{mn}^{(2)}$ ,  $y_{mn}^{(2)}$ :

$$x_{mn}^{(2)} h_n^{(1)}(\xi_0) + a_{mn} j_n(\xi_0) = -x_{mn}^{(1)} q_1 j_n(\xi_1) - y_{mn}^{(1)} q_2 j_n(\xi_2),\tag{11}$$

$$y_{mn}^{(2)} h_n^{(1)}(\xi_0) + b_{mn} j_n(\xi_0) = -x_{mn}^{(1)} \bar{p}_1 q_1 j_n(\xi_1) - y_{mn}^{(1)} \bar{p}_2 q_2 j_n(\xi_2);\tag{12}$$

$$x_{mn}^{(2)} g_n^{(1)}(\xi_0) + a_{mn} g_n(\xi_0) = x_{mn}^{(1)} \bar{p}_1 g_n(\xi_1) + y_{mn}^{(1)} \bar{p}_2 g_n(\xi_2),\tag{13}$$

$$y_{mn}^{(2)} g_n^{(1)}(\xi_0) + b_{mn} g_n(\xi_0) = x_{mn}^{(1)} g_n(\xi_1) + y_{mn}^{(1)} g_n(\xi_2),\tag{14}$$

где  $\bar{p}_j = p_j/h_0$ ,  $\xi_s = k_s R$ ,  $s = 0, 1, 2$ .

Из уравнения (11) с помощью (13) исключим  $x_{mn}^{(2)}$ , а из уравнения (12) —  $y_{mn}^{(2)}$ . Используя формулу  $h_n^{(1)}(\xi_0)g_n(\xi_0) - j_n(\xi_0)g_n^{(1)}(\xi_0) = \frac{1}{i\xi_0^2}$ , получим систему

$$x_{mn}^{(1)} \left( \bar{p}_1 g_n(\xi_1) h_n^{(1)}(\xi_0) + q_1 j_n(\xi_1) g_n^{(1)}(\xi_0) \right) + y_{mn}^{(1)} \left( \bar{p}_2 g_n(\xi_2) h_n^{(1)}(\xi_0) + q_2 j_n(\xi_2) g_n^{(1)}(\xi_0) \right) = \frac{a_{mn}}{i\xi_0^2};\tag{15}$$

$$x_{mn}^{(1)} \left( g_n(\xi_1) h_n^{(1)}(\xi_0) + \bar{p}_1 q_1 j_n(\xi_1) g_n^{(1)}(\xi_0) \right) + y_{mn}^{(1)} \left( g_n(\xi_2) h_n^{(1)}(\xi_0) + \bar{p}_2 q_2 j_n(\xi_2) g_n^{(1)}(\xi_0) \right) = \frac{b_{mn}}{i\xi_0^2}.$$

Полагая  $n = 1$ , запишем систему уравнений (11), (12), (15) в матричном виде:

$$\bar{X}_m^{(2)} = -\left(j_1(\xi_0)\hat{E}\bar{a}_m + \hat{Q}\bar{X}_m^{(1)}\right)/h_1^{(1)}(\xi_0); \quad (16)$$

$$i\xi_0^2\hat{P}\bar{X}_m^{(1)} = \bar{a}_m, \quad (17)$$

где  $\bar{X}_m^{(j)} = \begin{pmatrix} x_{m1}^{(j)} \\ y_{m1}^{(j)} \end{pmatrix}$ ,  $\bar{a}_m = \begin{pmatrix} a_{m1} \\ b_{m1} \end{pmatrix}$ ,  $\hat{E} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ ,  $\hat{Q} = \begin{pmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{pmatrix}$ ,  $\hat{P} = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix}$ ,

$$q_{11} = q_1 j_1(\xi_1), \quad q_{12} = q_2 j_1(\xi_2), \quad q_{21} = \bar{p}_1 q_1 j_1(\xi_1), \quad q_{22} = \bar{p}_2 q_2 j_1(\xi_2),$$

$$p_{11} = \bar{p}_1 g_1(\xi_1) h_1^{(1)}(\xi_0) + q_1 j_1(\xi_1) g_1^{(1)}(\xi_0), \quad p_{12} = \bar{p}_2 g_1(\xi_2) h_1^{(1)}(\xi_0) + q_2 j_1(\xi_2) g_1^{(1)}(\xi_0), \quad (18)$$

$$p_{21} = g_1(\xi_1) h_1^{(1)}(\xi_0) + \bar{p}_1 q_1 j_1(\xi_1) g_1^{(1)}(\xi_0), \quad p_{22} = g_1(\xi_2) h_1^{(1)}(\xi_0) + \bar{p}_2 q_2 j_1(\xi_2) g_1^{(1)}(\xi_0);$$

$$j_1(x) = \frac{\sin x}{x^2} - \frac{\cos x}{x}, \quad h_1^{(1)}(x) = -\left(\frac{1}{x} + \frac{i}{x^2}\right) e^{ix},$$

$$g_1(x) = \frac{1}{x} \left( \frac{\cos x}{x} + \left(1 - \frac{1}{x^2}\right) \sin x \right), \quad g_1^{(1)}(x) = \frac{1}{x} \left( \frac{i}{x^2} + \frac{1}{x} - i \right) e^{ix}.$$

Подставив (17) в (16), получим соотношения

$$\bar{X}_m^{(2)} = -\hat{K}\bar{X}_m^{(1)}, \quad \hat{K} = \left(i\xi_0^2 j_1(\xi_0)\hat{P} + \hat{Q}\right)/h_1^{(1)}(\xi_0), \quad m = 0, \pm 1, \quad (19)$$

которые будут использоваться в дальнейшем.

### 3. Вычисление специальных интегралов

Вычислим вспомогательные интегралы вида

$$\bar{J}_{mn}^Y = \int_0^{2\pi} \int_0^\pi Y_n^m(\theta, \varphi) \bar{e}_r \sin \theta d\theta d\varphi, \quad \bar{J}_{mn}^\Pi = \int_0^{2\pi} \int_0^\pi \bar{\Pi}_{mn}(\theta, \varphi) \sin \theta d\theta d\varphi,$$

$$\bar{J}_{mn}^\Gamma = \int_0^{2\pi} \int_0^\pi \bar{\Gamma}_{mn}(\theta, \varphi) \sin \theta d\theta d\varphi, \quad n = 1, 2, \dots, \quad m = 0, \pm 1, \pm 2, \dots \pm n.$$

**Лемма 2.** Имеют место формулы

$$\bar{J}_{mn}^\Gamma = 0 \quad \text{при } n = 1, 2, \dots; \quad m = 0, \pm 1, \pm 2, \dots \pm n; \quad (20)$$

$$\bar{J}_{mn}^{Y,\Pi} = 0 \quad \text{при } n \geq 2; \quad m = 0, \pm 1, \pm 2, \dots \pm n;$$

$$\bar{J}_{01}^Y = \frac{4\pi}{3} \bar{e}_z, \quad \bar{J}_{01}^\Pi = \frac{8\pi}{3} \bar{e}_z, \quad \bar{J}_{11}^Y = -\frac{4\pi}{3} \bar{e}_+, \quad \bar{J}_{11}^\Pi = -\frac{8\pi}{3} \bar{e}_+, \quad \bar{J}_{-11}^Y = -\frac{2\pi}{3} \bar{e}_-, \quad \bar{J}_{-11}^\Pi = -\frac{4\pi}{3} \bar{e}_-.$$

Доказательство. Для примера вычислим интеграл, используя формулу [15, с. 283]:

$$\begin{aligned}
\bar{J}_{01}^Y &= \int_0^\pi \int_0^{2\pi} \cos \theta \sin \theta \bar{e}_r \, d\theta \, d\varphi = \\
&= \int_0^\pi \cos \theta \sin \theta \int_0^{2\pi} (\sin \theta \cos \varphi \bar{e}_x + \sin \theta \sin \varphi \bar{e}_y + \cos \theta \bar{e}_z) \, d\varphi \, d\theta = \\
&= 2\pi \int_0^\pi \cos^2 \theta \sin \theta \, d\theta \bar{e}_z = \frac{4\pi}{3} \bar{e}_z.
\end{aligned}$$

Аналогично вычисляются остальные интегралы (20).

**Лемма 3.** Для объемных интегралов

$$\begin{aligned}
\tilde{I}_{mn}^\Pi &= \int_{D_{RP}} \tilde{n}_{mn}(\vec{r}, k_0) \, dV = \int_R^P (f_n^{(1)}(k_0 r) \bar{J}_{mn}^Y + g_n^{(1)}(k_0 r) \bar{J}_{mn}^\Pi) r^2 \, dr, \\
\tilde{I}_{mn}^\Gamma &= \int_{D_{RP}} \tilde{m}_{mn}(\vec{r}, k_0) \, dV = \int_R^P h_n^{(1)}(k_0 r) \bar{J}_{mn}^\Gamma r^2 \, dr
\end{aligned}$$

имеют место формулы

$$\begin{aligned}
\tilde{I}_{mn}^\Gamma &= 0 \text{ при } n=1, 2, \dots; \quad m=0, \pm 1, \pm 2, \dots \pm n; \\
\tilde{I}_{mn}^\Pi &= 0 \text{ при } n \geq 2; \quad m=0, \pm 1, \dots \pm n; \\
\tilde{I}_{01}^\Pi &= -2\bar{f}_0 \bar{e}_z, \quad \tilde{I}_{11}^\Pi = 2\bar{f}_0 \bar{e}_+, \quad \tilde{I}_{-11}^\Pi = \bar{f}_0 \bar{e}_-,
\end{aligned} \tag{21}$$

где в асимптотическом приближении  $\bar{f}_0 = \frac{4\pi i}{3k_0^3} + \frac{V_R}{\xi_0} h_1^{(1)}(\xi_0)$ .

Доказательство. Для доказательства используются формулы (20). Вычислим для примера интеграл

$$\tilde{I}_{01}^\Pi = \frac{4\pi}{3} \int_R^P (f_1^{(1)}(k_0 r) + 2g_1^{(1)}(k_0 r)) r^2 \, dr \bar{e}_z = -2\bar{f}_0 \bar{e}_z,$$

$$\begin{aligned}
\text{где } \bar{f}_0 &= -\frac{2\pi}{3} \int_R^P (f_1^{(1)} + 2g_1^{(1)}) r^2 \, dr = -\frac{4\pi}{3k_0} \int_R^P \left( h_1^{(1)}(k_0 r) + \frac{\partial}{\partial r} (r h_1^{(1)}(k_0 r)) \right) r \, dr = \\
&= \frac{4\pi}{3k_0} (R^2 h_1^{(1)}(k_0 R) - P^2 h_1^{(1)}(k_0 P)).
\end{aligned}$$

Преобразуем величину  $\bar{f}_0$  с учетом допущения (1)  $\frac{1}{k_0 P} \ll \frac{1}{k_0^2 P^2}$ . Тогда для сферической функции Ханкеля получим асимптотическую формулу  $h_1^{(1)}(k_0 P) \approx -\frac{i}{k_0^2 P^2}$ . В результате

$$\bar{f}_0 = \frac{4\pi i}{3k_0^3} + \frac{V_R}{k_0 R} h_1^{(1)}(k_0 R).$$

**Лемма 4.** Для объемных интегралов

$$\vec{I}_{mn}^{\Pi}(k_j) = \int_{D_R} \vec{n}_{mn}(\vec{r}, k_j) dV = \int_0^R (f_n(k, r) \vec{J}_{mn}^Y + g_n(k, r) \vec{J}_{mn}^{\Pi}) r^2 dr,$$

$$\vec{I}_{mn}^{\Gamma}(k_j) = \int_{D_R} \vec{m}_{mn}(\vec{r}, k_j) dV = \int_0^R j_n(k_j r) \vec{J}_{mn}^{\Gamma} r^2 dr$$

имеют место формулы

$$\begin{aligned} \vec{I}_{mn}^{\Gamma}(k_j) &= 0 \text{ при } n = 1, 2, \dots; \quad m = 0, \pm 1, \dots, \pm n; \\ \vec{I}_{mn}^{\Pi}(k_j) &= 0 \text{ при } n \geq 2; \quad m = 0, \pm 1, \dots, \pm n; \end{aligned} \quad (22)$$

$$\vec{I}_{01}^{\Pi}(k_j) = -2\bar{f}_j \vec{e}_z, \quad \vec{I}_{11}^{\Pi}(k_j) = 2\bar{f}_j \vec{e}_+, \quad \vec{I}_{-11}^{\Pi}(k_j) = \bar{f}_j \vec{e}_-,$$

где  $\bar{f}_j = -\frac{V_R}{k_j R} j_1(k_j R)$ .

**Доказательство.** Для доказательства используются формулы (20). Вычислим для примера интеграл

$$\vec{I}_{01}^{\Pi} = \frac{4\pi}{3} \int_0^R (f_1(k_j r) + 2g_1(k_j r)) r^2 dr \vec{e}_z = -2\bar{f}_j \vec{e}_z,$$

где  $\bar{f}_j = -\frac{2\pi}{3} \int_0^R (f_1 + 2g_1) r^2 dr = -\frac{4\pi}{3k_j} \int_0^R \left( j_1(k_j r) + \frac{d}{dr}(rj_1(k_j r)) \right) r dr = -\frac{4\pi}{3k_j} R^2 j_1(k_j R)$ .

#### 4. Усредненные электрические и магнитные поля в композите

В результате взаимодействия поля  $\vec{E}_0, \vec{H}_0$  с частицами  $D_S$  композита в пространстве образуются отраженные поля  $\vec{E}'_S, \vec{H}'_S$  и поля  $\vec{E}_S, \vec{H}_S$ , проникшие внутрь частиц  $D_S$ . Вычислим суммарные поля в области  $D_P$ :

$$\vec{E}_{\text{сум}} = \begin{cases} \vec{E}_0 + \sum_{S=1}^N \vec{E}'_S & \text{в } D_0, \\ \vec{E}_S & \text{в } D_S; \end{cases} \quad \vec{H}_{\text{сум}} = \begin{cases} \vec{H}_0 + \sum_{S=1}^N \vec{H}'_S & \text{в } D_0, \\ \vec{H}_S & \text{в } D_S. \end{cases} \quad (23)$$

Усредним поля (23) по объему  $D_P$ :

$$\begin{aligned} \vec{E} &= \frac{1}{V_P} \int_{D_P} \vec{E}_{\text{сум}} dV = \frac{1}{V_P} \left[ \int_{D_P} \vec{E}_0 dV + \sum_{S=1}^N \left( \int_{D_0} \vec{E}'_S dV + \int_{D_S} \vec{E}_S dV - \int_{D_S} \vec{E}_0 dV \right) \right], \\ \vec{H} &= \frac{1}{V_P} \int_{D_P} \vec{H}_{\text{сум}} dV = \frac{1}{V_P} \left[ \int_{D_P} \vec{H}_0 dV + \sum_{S=1}^N \left( \int_{D_0} \vec{H}'_S dV + \int_{D_S} \vec{H}_S dV - \int_{D_S} \vec{H}_0 dV \right) \right]. \end{aligned} \quad (24)$$

Так как частицы находятся в одинаковых электродинамических условиях и выполнено условие (1), будем предполагать

$$\int_{D_S} \vec{E}_0 dV = \int_{D_R} \vec{E}_0 dV = V_R \vec{E}_0^0, \quad \int_{D_S} \vec{E}_S dV = \int_{D_R} \vec{E}_R dV;$$

$$\int_{D_S} \vec{H}_0 dV = \int_{D_R} \vec{H}_0 dV = V_R \vec{H}_0^0, \quad \int_{D_S} \vec{H}_S dV = \int_{D_R} \vec{H}_R dV,$$

пренебрегая малыми величинами  $(k_0 P)^l$ ,  $l \geq 2$  ( $k_0 P < 0,025$ ).

Поскольку частицы расположены на достаточно больших расстояниях друг от друга по сравнению с радиусами самих частиц, будем предполагать, что отраженные поля  $\vec{E}'_S, \vec{H}'_S$  взаимодействуют с окружающими частицами, но повторными отраженными от частиц полями будем пренебрегать. В рамках такого приближения получим

$$\int_{D_0} \vec{E}'_S dV = \int_{D_0} \vec{E}'_R dV = \int_{D_{RP}} \vec{E}'_R dV - \sum_{s=2}^N \int_{D_S} \vec{E}'_R dV = \int_{D_{RP}} \vec{E}'_R dV - V_R \sum_{s=2}^N \vec{E}'_R(\vec{r}_s);$$

$$\int_{D_0} \vec{H}'_S dV = \int_{D_{RP}} \vec{H}'_R dV - V_R \sum_{s=2}^N \vec{H}'_R(\vec{r}_s),$$

где  $\vec{r}_s$  – координаты центра частицы  $D_S$ . Суммы, рассматривая их как интегральные, заменим на интегралы

$$\sum_{s=2}^N \vec{E}'_R(\vec{r}_s) = \nu \int_{D_{RP}} \vec{E}'_R(\vec{r}) dV, \quad \sum_{s=2}^N \vec{H}'_R(\vec{r}_s) = \nu \int_{D_{RP}} \vec{H}'_R(\vec{r}) dV.$$

Поля (24) примут вид

$$\vec{E} = \bar{\tau} \vec{E}_0^0 + \nu \left( \bar{\tau} \int_{D_{RP}} \vec{E}'_R dV + \int_{D_R} \vec{E}_R dV \right), \quad (25)$$

$$\vec{H} = \bar{\tau} \vec{H}_0^0 + \nu \left( \bar{\tau} \int_{D_{RP}} \vec{H}'_R dV + \int_{D_R} \vec{H}_R dV \right), \quad \bar{\tau} = 1 - \tau,$$

где  $\vec{E}_R = \vec{E}_R^0 + \vec{E}_R^{(1)}$ ,  $\vec{H}_R = \vec{H}_R^0 + \vec{H}_R^{(1)}$ ;  $\vec{E}_R^0, \vec{H}_R^0$  – поле (10), возбужденное внутри частицы  $D_R$  первичным полем (7);  $\vec{E}_R^{(1)}, \vec{H}_R^{(1)}$  – поле, возбужденное внутри частицы  $D_R$  суммой  $\vec{E}_R''(\vec{r}) = \sum_{s=2}^N \vec{E}'_S(\vec{r})$ ,  $\vec{H}_R''(\vec{r}) = \sum_{s=2}^N \vec{H}'_S(\vec{r})$  отраженных от соседних частиц полей.

Учитывая соотношения (9), (21), вычислим интегралы, входящие в формулы (25):

$$\int_{D_{RP}} \vec{E}'_R dV = \sum_{m=-1}^1 y_{m1}^{(2)} \int_{D_{RP}} \tilde{n}_{m1}(\vec{r}, k_0) dV = \bar{f}_0 \left( 2y_{11}^{(2)} \vec{e}_+ + y_{-11}^{(2)} \vec{e}_- - 2y_{01}^{(2)} \vec{e}_z \right);$$

$$\int_{D_{RP}} \vec{H}'_R dV = \sum_{m=-1}^1 x_{m1}^{(2)} \int_{D_{RP}} \tilde{n}_{m1}(\vec{r}, k_0) dV = h_0 \bar{f}_0 \left( 2x_{11}^{(2)} \vec{e}_+ + x_{-11}^{(2)} \vec{e}_- - 2x_{01}^{(2)} \vec{e}_z \right). \quad (26)$$

Для вычисления интегралов  $\int_{D_{RP}} \vec{E}_R dV$ ,  $\int_{D_{RP}} \vec{H}_R dV$  определим поле  $\vec{E}_R^{(1)}, \vec{H}_R^{(1)}$ , представив

его в виде (7). Для использования леммы 1 вычислим значение поля  $\vec{E}_R'', \vec{H}_R''$  в начале координат:

$$\vec{E}_R''(0) = \sum_{S=2}^N \vec{E}_S'(0) = \sum_{S=2}^N \vec{E}_R'(\vec{r}_S) \approx \nu \int_{D_{RP}} \vec{E}_R'(\vec{r}) dV = \nu \bar{f}_0 \left( 2y_{11}^{(2)} \vec{e}_+ + y_{-11}^{(2)} \vec{e}_- - 2y_{01}^{(2)} \vec{e}_z \right),$$

$$\vec{H}_R''(0) = \nu h_0 \bar{f}_0 \left( 2x_{11}^{(2)} \vec{e}_+ + x_{-11}^{(2)} \vec{e}_- - 2x_{01}^{(2)} \vec{e}_z \right).$$

В результате с учетом (8) получим представление поля в окрестности частицы  $D_1$ :

$$\vec{E}_R''(\vec{r}) = \sum_{m=-1}^1 \left( a_{m1}^{(1)} \vec{m}_{m1}(\vec{r}) + b_{m1}^{(1)} \vec{n}_{m1}(\vec{r}) \right) + \dots,$$

$$\vec{H}_R''(\vec{r}) = h_0 \sum_{m=-1}^1 \left( a_{m1}^{(1)} \vec{n}_{m1}(\vec{r}) + b_{m1}^{(1)} \vec{m}_{m1}(\vec{r}) \right) + \dots,$$

где  $a_{m1}^{(1)} = -3\nu \bar{f}_0 x_{m1}^{(2)}$ ,  $b_{m1}^{(1)} = -3\nu \bar{f}_0 y_{m1}^{(2)}$ .

Это поле возбуждает внутри частицы  $D_R$  поле вида (10):

$$\vec{E}_R^{(1)}(\vec{r}) = \sum_{m=-1}^1 \left( x_{m1}^{(3)} \vec{K}_{m1}^{(1)}(\vec{r}, k_1) + y_{m1}^{(3)} \vec{K}_{m1}^{(2)}(\vec{r}, k_2) \right) + \dots,$$

$$\vec{H}_R^{(1)}(\vec{r}) = \sum_{m=-1}^1 \left( x_{m1}^{(3)} p_1 \vec{K}_{m1}^{(1)}(\vec{r}, k_1) + y_{m1}^{(3)} p_2 \vec{K}_{m1}^{(2)}(\vec{r}, k_2) \right) + \dots,$$

где в соответствии с формулами (17), (19) получаем

$$\vec{X}_m^{(3)} = \frac{1}{i\xi_0^2} \hat{P}^{-1} \vec{a}_m^{(1)} = -\frac{3\nu \bar{f}_0}{i\xi_0^2} \hat{P}^{-1} \vec{X}_m^{(2)} = \frac{3\nu \bar{f}_0}{i\xi_0^2} \hat{P}^{-1} \hat{K} \vec{X}_m^{(1)}.$$

Суммарное поле внутри частицы  $D_R$  примет вид

$$\vec{E}_R(\vec{r}) = \sum_{m=-1}^1 \left( x_{m1}^{(4)} \vec{K}_{m1}^{(1)}(\vec{r}, k_1) + y_{m1}^{(4)} \vec{K}_{m1}^{(2)}(\vec{r}, k_2) \right) + \dots,$$

$$\vec{H}_R(\vec{r}) = \sum_{m=-1}^1 \left( x_{m1}^{(4)} p_1 \vec{K}_{m1}^{(1)}(\vec{r}, k_1) + y_{m1}^{(4)} p_2 \vec{K}_{m1}^{(2)}(\vec{r}, k_2) \right) + \dots,$$

где  $\vec{X}_m^{(4)} = \vec{X}_m^{(1)} + \vec{X}_m^{(3)} = \hat{K}_1 \vec{X}_m^{(1)}$ ,  $\hat{K}_1 = \hat{E} + \frac{3\nu \bar{f}_0}{i\xi_0^2} \hat{P}^{-1} \hat{K}$ .

Учитывая (22), получим

$$\begin{aligned} \int_{D_R} \vec{K}_{mn}^{(j)} dV &= \int_{D_R} \vec{n}_{mn}(\vec{r}, k_j) dV, \\ \int_{D_R} \vec{E}_R dV &= \sum_{m=-1}^1 \left( x_{m1}^{(4)} \int_{D_R} \vec{n}_{m1}(\vec{r}, k_1) dV + y_{m1}^{(4)} \int_{D_R} \vec{n}_{m1}(\vec{r}, k_2) dV \right) = 2 \left( \bar{f}_1 x_{11}^{(4)} + \bar{f}_2 y_{11}^{(4)} \right) \vec{e}_+ + \\ &+ \left( \bar{f}_1 x_{-11}^{(4)} + \bar{f}_2 y_{-11}^{(4)} \right) \vec{e}_- - 2 \left( \bar{f}_1 x_{01}^{(4)} + \bar{f}_2 y_{01}^{(4)} \right) \vec{e}_z, \end{aligned} \quad (27)$$

$$\int_{D_R} \vec{H}_R dV = \sum_{m=1}^1 \left( x_{m1}^{(4)} p_1 \int_{D_R} \vec{n}_{m1}(\vec{r}, k_1) dV + y_{m1}^{(4)} p_2 \int_{D_R} \vec{n}_{m1}(\vec{r}, k_2) dV \right) = 2(\bar{f}_1 p_1 x_{11}^{(4)} + \bar{f}_2 p_2 y_{11}^{(4)}) \vec{e}_+ + \\ + (\bar{f}_1 p_1 x_{-11}^{(4)} + \bar{f}_2 p_2 y_{-11}^{(4)}) \vec{e}_- - 2(\bar{f}_1 p_1 x_{01}^{(4)} + \bar{f}_2 p_2 y_{01}^{(4)}) \vec{e}_z.$$

**Теорема 1.** Усредненные по области  $D_p$  в композитном материале со сферическими би-изотропными частицами радиуса  $R$  электрическое и магнитное поля в базисе  $\vec{e}_+$ ,  $\vec{e}_-$ ,  $\vec{e}_z$  при выполнении условия (1) определяются формулами

$$\begin{aligned} \vec{E} &= -E_- \vec{e}_+ - E_+ \vec{e}_- + E_z \vec{e}_z = E_x \vec{e}_x + E_y \vec{e}_y + E_z \vec{e}_z = \\ &= \left( -\frac{2\bar{\tau}}{3} b_{11} + 2\nu \left( \bar{\tau} \bar{f}_0 y_{11}^{(2)} + \bar{f}_1 x_{11}^{(4)} + \bar{f}_2 y_{11}^{(4)} \right) \right) \vec{e}_+ + \\ &+ \left( -\frac{\bar{\tau}}{3} b_{-11} + \nu \left( \bar{\tau} \bar{f}_0 y_{-11}^{(2)} + \bar{f}_1 x_{-11}^{(4)} + \bar{f}_2 y_{-11}^{(4)} \right) \right) \vec{e}_- + \\ &+ \left( \frac{2\bar{\tau}}{3} b_{01} - 2\nu \left( \bar{\tau} \bar{f}_0 y_{01}^{(2)} + \bar{f}_1 x_{01}^{(4)} + \bar{f}_2 y_{01}^{(4)} \right) \right) \vec{e}_z, \end{aligned} \quad (28)$$

$$\begin{aligned} \vec{H} &= -H_- \vec{e}_+ - H_+ \vec{e}_- + H_z \vec{e}_z = H_x \vec{e}_x + H_y \vec{e}_y + H_z \vec{e}_z = \\ &= \left( -\frac{2}{3} h_0 \bar{\tau} a_{11} + 2\nu \left( \bar{\tau} h_0 \bar{f}_0 x_{11}^{(2)} + \bar{f}_1 p_1 x_{11}^{(4)} + \bar{f}_2 p_2 y_{11}^{(4)} \right) \right) \vec{e}_+ + \\ &+ \left( -\frac{1}{3} h_0 \bar{\tau} a_{-11} + \nu \left( \bar{\tau} h_0 \bar{f}_0 x_{-11}^{(2)} + \bar{f}_1 p_1 x_{-11}^{(4)} + \bar{f}_2 p_2 y_{-11}^{(4)} \right) \right) \vec{e}_- + \\ &+ \left( \frac{2}{3} h_0 \bar{\tau} a_{01} - 2\nu \left( \bar{\tau} h_0 \bar{f}_0 x_{01}^{(2)} + \bar{f}_1 p_1 x_{01}^{(4)} + \bar{f}_2 p_2 y_{01}^{(4)} \right) \right) \vec{e}_z, \end{aligned}$$

где  $E_+ = \frac{1}{2}(iE_y + E_x)$ ,  $E_- = \frac{1}{2}(iE_y - E_x)$ .

Для доказательства достаточно подставить выражения (8), (26), (27) в формулы (25).

**Следствие 1.** Имеют место матричные соотношения для компонент полей (28)

$$\vec{V}_+ = \frac{1}{2} \hat{L} \vec{X}_{-1}^{(1)}, \quad \vec{V}_- = \hat{L} \vec{X}_1^{(1)}, \quad \vec{V}_z = \hat{L} \vec{X}_0^{(1)}, \quad (29)$$

где

$$\hat{L} = i \frac{2}{3} \xi_0^2 \bar{\tau} \hat{H} \hat{P} + 2\nu \left( \bar{\tau} \bar{f}_0 \hat{H} \hat{K} - \hat{F} \hat{K}_1 \right), \quad (30)$$

$$\vec{V}_+ = \begin{pmatrix} E_+ \\ H_+ \end{pmatrix}, \quad \vec{V}_- = \begin{pmatrix} E_- \\ H_- \end{pmatrix}, \quad \vec{V}_z = \begin{pmatrix} E_z \\ H_z \end{pmatrix}, \quad \hat{H} = \begin{pmatrix} 0 & 1 \\ h_0 & 0 \end{pmatrix}, \quad \hat{F} = \begin{pmatrix} \bar{f}_1 & \bar{f}_2 \\ p_1 \bar{f}_1 & p_2 \bar{f}_2 \end{pmatrix}.$$

Доказательство. Приравняв  $z$ -е компоненты полей (28), получим равенства

$$E_z = \frac{2}{3} \bar{\tau} b_{01} - 2\nu \left( \bar{\tau} \bar{f}_0 y_{01}^{(2)} + \bar{f}_1 x_{01}^{(4)} + \bar{f}_2 y_{01}^{(4)} \right),$$

$$H_z = \frac{2}{3} h_0 \bar{\tau} a_{01} - 2\nu (\bar{\tau} h_0 \bar{f}_0 x_{01}^{(2)} + \bar{f}_1 p_1 x_{01}^{(4)} + \bar{f}_2 p_2 y_{01}^{(4)}).$$

Запишем их в матричном виде

$$\vec{V}_z = \frac{2\bar{\tau}}{3} \hat{H} \vec{a}_0 - 2\nu (\bar{\tau} \hat{f}_0 \hat{H} \vec{X}_0^{(2)} + \hat{F} \vec{X}_0^{(4)}).$$

Используя равенства (17), (19) при  $m=0$ , получим третью формулу (29). Аналогично получим остальные формулы (29).

### 5. Усредненные электрическая и магнитная индукции

Запишем суммарные индукции в области  $D_p$ :

$$\vec{D}_{\text{сум}} = \begin{cases} \varepsilon_0 \left( \vec{E}_0 + \sum_{S=1}^N \vec{E}'_S \right) & \text{в } D_0, \\ \varepsilon \vec{E}_S + G \vec{H}_S & \text{в } D_S; \end{cases} \quad \vec{B}_{\text{сум}} = \begin{cases} \mu_0 \left( \vec{H}_0 + \sum_{S=1}^N \vec{H}'_S \right) & \text{в } D_0, \\ \mu \vec{H}_S + Z \vec{E}_S & \text{в } D_S. \end{cases} \quad (31)$$

Усредненные индукции определяются формулами

$$\vec{D} = \frac{1}{V_p} \int_{D_p} \vec{D}_{\text{сум}} dV; \quad \vec{B} = \frac{1}{V_p} \int_{D_p} \vec{B}_{\text{сум}} dV.$$

По аналогии с формулами (25) получим

$$\begin{aligned} \vec{D} &= \varepsilon_0 \bar{\tau} \vec{E}_0^0 + \nu \left( \varepsilon_0 \bar{\tau} \int_{D_{RP}} \vec{E}'_R dV + \int_{D_R} (\varepsilon \vec{E}_R + G \vec{H}_R) dV \right); \\ \vec{B} &= \mu_0 \bar{\tau} \vec{H}_0^0 + \nu \left( \mu_0 \bar{\tau} \int_{D_{RP}} \vec{H}'_R dV + \int_{D_R} (\mu \vec{H}_R + Z \vec{E}_R) dV \right). \end{aligned} \quad (32)$$

**Теорема 2.** Усредненные по области  $D_p$  в композитном материале с биизотропными частями электрическая и магнитная индукции в базисе  $\vec{e}_+$ ,  $\vec{e}_-$ ,  $\vec{e}_z$  определяются выражениями

$$\begin{aligned} \vec{D} &= -D_- \vec{e}_+ - D_+ \vec{e}_- + D_z \vec{e}_z = D_x \vec{e}_x + D_y \vec{e}_y + D_z \vec{e}_z = \\ &= \left( -\frac{2}{3} \varepsilon_0 \bar{\tau} b_{11} + 2\nu (\varepsilon_0 \bar{\tau} \bar{f}_0 y_{11}^{(2)} + f_1^{(1)} x_{11}^{(4)} + f_2^{(1)} y_{11}^{(4)}) \right) \vec{e}_+ + \\ &+ \left( -\frac{1}{3} \varepsilon_0 \bar{\tau} b_{-11} + \nu (\varepsilon_0 \bar{\tau} \bar{f}_0 y_{-11}^{(2)} + f_1^{(1)} x_{-11}^{(4)} + f_2^{(1)} y_{-11}^{(4)}) \right) \vec{e}_- + \\ &+ \left( \frac{2}{3} \varepsilon_0 \bar{\tau} b_{01} - 2\nu (\varepsilon_0 \bar{\tau} \bar{f}_0 y_{01}^{(2)} + f_1^{(1)} x_{01}^{(4)} + f_2^{(1)} y_{01}^{(4)}) \right) \vec{e}_z; \\ \vec{B} &= -B_- \vec{e}_+ - B_+ \vec{e}_- + B_z \vec{e}_z = B_x \vec{e}_x + B_y \vec{e}_y + B_z \vec{e}_z = \\ &= \left( -\frac{2}{3} \mu_0 h_0 \bar{\tau} a_{11} + 2\nu (\mu_0 \bar{\tau} h_0 \bar{f}_0 x_{11}^{(2)} + f_1^{(2)} x_{11}^{(4)} + f_2^{(2)} y_{11}^{(4)}) \right) \vec{e}_+ + \end{aligned} \quad (33)$$

$$\begin{aligned}
& + \left( -\frac{1}{3} \mu_0 h_0 \bar{\tau} a_{-11} + \nu \left( \mu_0 \tau h_0 \bar{f}_0 x_{-11}^{(2)} + f_1^{(2)} x_{-11}^{(4)} + f_2^{(2)} y_{-11}^{(4)} \right) \right) \bar{e}_- + \\
& + \left( \frac{2}{3} \mu_0 h_0 \bar{\tau} a_{01} - 2\nu \left( \mu_0 \tau h_0 \bar{f}_0 x_{01}^{(2)} + f_1^{(2)} x_{01}^{(4)} + f_2^{(2)} y_{01}^{(4)} \right) \right) \bar{e}_z,
\end{aligned}$$

где  $D_+ = \frac{1}{2}(iD_y + D_x)$ ,  $D_- = \frac{1}{2}(iD_y - D_x)$ ,  $B_+ = \frac{1}{2}(iB_y + B_x)$ ,  $B_- = \frac{1}{2}(iB_y - B_x)$ ,

$$f_1^{(1)} = \bar{f}_1(\varepsilon + Gp_1), \quad f_2^{(1)} = \bar{f}_2(\varepsilon + Gp_2), \quad f_1^{(2)} = \bar{f}_1(Z + \mu p_1), \quad f_2^{(2)} = \bar{f}_2(Z + \mu p_2).$$

Для доказательства достаточно подставить выражения (8), (26), (27) в формулы (32).

**Следствие 2.** *Имеют место матричные соотношения для компонент индукций (33)*

$$\bar{W}_+ = \frac{1}{2} \hat{N} \bar{X}_{-1}^{(1)}, \quad \bar{W}_- = \hat{N} \bar{X}_1^{(1)}, \quad \bar{W}_z = \hat{N} \bar{X}_0^{(1)}, \quad (34)$$

где  $\hat{N} = i \frac{2}{3} \bar{\tau} \xi_0^2 \hat{H}_1 \hat{P} + 2\nu (\bar{\tau} \hat{f}_0 \hat{H}_1 \hat{K} - \hat{F}_1 \hat{K}_1)$ , (35)

$$\bar{W}_+ = \begin{pmatrix} D_+ \\ B_+ \end{pmatrix}, \quad \bar{W}_- = \begin{pmatrix} D_- \\ B_- \end{pmatrix}, \quad \bar{W}_z = \begin{pmatrix} D_z \\ B_z \end{pmatrix}, \quad \hat{H}_1 = \begin{pmatrix} 0; & \varepsilon_0 \\ \mu_0 h_0; & 0 \end{pmatrix}, \quad \hat{F}_1 = \begin{pmatrix} f_1^{(1)} & f_2^{(1)} \\ f_1^{(2)} & f_2^{(2)} \end{pmatrix}.$$

Доказательство проводится по аналогии с доказательством следствия 1.

## 6. Вычисление эффективных материальных параметров композита

Выразим векторы электрической и магнитной индукции (33) через векторы электрической и магнитной напряженностей поля. Воспользовавшись формулами (29), получим

$$\bar{X}_{-1}^{(1)} = 2\hat{L}^{-1}\bar{V}_+, \quad \bar{X}_1^{(1)} = \hat{L}^{-1}\bar{V}_-, \quad \bar{X}_0^{(1)} = \hat{L}^{-1}\bar{V}_z.$$

Подставим полученные выражения в (34), тогда

$$\bar{W}_+ = \hat{C}\bar{V}_+, \quad \bar{W}_- = \hat{C}\bar{V}_-, \quad \bar{W}_z = \hat{C}\bar{V}_z, \quad \hat{C} = \hat{N}\hat{L}^{-1}, \quad (36)$$

где  $\hat{C} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$ .

Запишем (36) в развернутом виде:

$$D_+ = c_{11}E_+ + c_{12}H_+, \quad D_- = c_{11}E_- + c_{12}H_-; \quad (37)$$

$$B_+ = c_{21}E_+ + c_{22}H_+, \quad B_- = c_{21}E_- + c_{22}H_-; \quad (38)$$

$$D_z = c_{11}E_z + c_{12}H_z, \quad B_z = c_{21}E_z + c_{22}H_z. \quad (39)$$

Складывая и вычитая (37) (аналогично складывая и вычитая (38)), получим

$$\begin{aligned}
D_x &= c_{11}E_x + c_{12}H_x, & D_y &= c_{11}E_y + c_{12}H_y; \\
B_x &= c_{21}E_x + c_{22}H_x, & B_y &= c_{21}E_y + c_{22}H_y.
\end{aligned} \quad (40)$$

Равенства (39), (40) запишем в векторном виде:

$$\vec{D} = c_{11}\vec{E} + c_{12}\vec{H}, \quad \vec{B} = c_{21}\vec{E} + c_{22}\vec{H}. \quad (41)$$

**Теорема 3.** Композитный материал, состоящий из случайно распределенных биизотропных сферических частиц радиуса  $R$  и заполненный вакуумом в областях между частицами, является биизотропной средой. Эффективные параметры композита определяются формулами

$$\varepsilon_{\text{эф}} = c_{11}, \quad G_{\text{эф}} = c_{12}, \quad \mu_{\text{эф}} = c_{22}, \quad Z_{\text{эф}} = c_{21}, \quad (42)$$

где  $\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} = \hat{N}\hat{L}^{-1}$ .

Для доказательства сравним индукции (41), построенные на основе разработанной методики моделирования, с индукциями уравнений (2). Получим формулы (42).

### Заключение

Разработан алгоритм вычисления эффективных материальных параметров композита, состоящего из случайно распределенных в вакууме сферических биизотропных частиц с одинаковыми диаметрами. Эффективные параметры, зависящие от частоты поля, описывают однородную биизотропную среду, эквивалентную исходному дискретно неоднородному композиту. Параметры представлены в виде аналитических формул. При разработке модели рассмотрен случай, когда частицы находятся на достаточно больших расстояниях друг от друга и при этом их электродинамическое взаимодействие не учитывается. Для взаимодействия поля со структурными частицами композита использовалось точное решение задачи дифракции на биизотропном шаре. Разработанная модель позволяет рассчитывать композиты, состоящие из киральных частиц, частиц из материала Федорова – Борна и метаматериалов.

Работа выполнена по заданию ГПНИ «Информатика и космос».

### Список литературы

1. Виноградов, А.П. Электродинамика композитных материалов / А.П. Виноградов. – М. : Эдиториал УРСС, 2001. – 206 с.
2. Магнитоэлектрические материалы / М.И. Бичурин [и др.]. – М. : Акад. естествознания, 2006. – 296 с.
3. Лагарьков, А.Н. Радиопоглощающие материалы на основе метаматериалов / А.Н. Лагарьков, В.Н. Кисель, В.Н. Семенов // Радиотехника и электроника. – 2012. – Т. 57, № 10. – С. 1119–1127.
4. Cui, Tie Jun. Metamaterials. Theory, Design and Applications / Tie Jun Cui, David R. Smith, Ruopeng Liu. – Springer, 2009. – 367 p.
5. Неганов, В.А. Современное состояние электродинамики искусственных киральных сред (обзор) / В.А. Неганов, О.В. Осипов // Физика волновых процессов и радиотехнические системы. – 2005. – Т. 8, № 1 – С.7–33.
6. Федосюк, В.М. Наноструктурные пленки и нанопроволоки / В.М. Федосюк. – Минск : Изд. центр БГУ, 2006. – 310 с.
7. Памятных, Е.А. Основы электродинамики материальных сред в переменных и неоднородных полях / Е.А. Памятных, Е.А. Туров. – М. : Наука. Физматлит, 2000. – 240 с.
8. Исмару, А. Распространение и рассеяние волн в случайно неоднородных средах. Т. 1. Однократное рассеяние / А. Исмару. – М. : Мир, 1981. – 280 с.
9. Guerin, F. Scattering of electromagnetic waves by helices and application to the modeling of chiral composites / F.Guerin // J. Phys. D: Appl. Phys. – 1995. – Vol. 28. – P. 623–642.

10. Демидчик, В.И. Математическое моделирование характеристик рассеяния проволочных частиц произвольной конфигурации и композитных материалов на их основе / В.И. Демидчик, Р.В. Корнеев, П.Д. Кухарчик // Докл. НАН Беларуси. – 2005. – Т. 49, № 2. – С. 38–41.
11. Демидчик, В.И. Анализ микроволновых свойств проволочных киральных рассеивателей методом интегральных уравнений / В.И. Демидчик, Р.В. Корнев // Вестник БГУ. Сер. 1. – 2004. – № 1. – С. 100–103.
12. Шатров, А.Д. Модель биизотропной среды из резонансных сферических частиц с идеальной смешанной проводимостью поверхности вдоль спиральных линий / А.Д. Шатров // Радиотехника и электроника. – 2000. – Т. 45, № 10. – С.1168–1170.
13. Костин, М.В. К теории киральной среды на основе сферических спирально проводящих частиц / М.В. Костин, В.В. Шевченко // Радиотехника и электроника. – 1998. – Т. 43, № 8. – С. 921–926.
14. Виноградов, А.П. К вопросу об эффективных параметрах метаматериалов / А.П. Виноградов, А.В. Дорофеев, С. Зухди // Успехи физических наук. – 2008. – Т. 178, № 5. – С. 514–518.
15. Ерофеев, В.Т. Аналитическое моделирование в электродинамике / В.Т. Ерофеев, И.С. Козловская. – Минск : БГУ, 2010. – 304 с.
16. Козлов, И.П. Дифракция электромагнитных волн на двух сферах / И.П. Козлов // Известия высших учебных заведений. Радиофизика. – 1975. – Т. XVIII, № 7. – С. 997–1008.

Поступила 29.12.2013

*НИИ прикладных проблем  
математики и информатики БГУ,  
Минск, пр. Независимости, 4  
e-mail: bsu\_erofeenko@tut.by*

**V.T. Erofeenko**

**ELECTRODYNAMICAL MODEL FOR CALCULATING  
EFFECTIVE PARAMETERS OF COMPOSITES FROM SPHERICAL  
BI-ISOTROPIC PARTICLES**

A method for calculation of the effective material parameters of chiral composites is developed. The composites consist of the disperse system of discrete reflectors filled with the material of arbitrary complex dielectric and magnetic penetrability and complex bi-isotropism. A strict analytical solution of the problem of diffraction on the bi-isotropic sphere of electromagnetic waves with wave length much greater than particles diameter is used in the formulas.

УДК 681.513.5

А.Г. Стрижнев, Г.В. Ледник, А.Н. Русакович

## СИНТЕЗ ЦИФРОВОГО РЕГУЛЯТОРА ДЛЯ ОБЪЕКТА УПРАВЛЕНИЯ ПЯТОГО ПОРЯДКА С ПРИМЕНЕНИЕМ ОПТИМАЛЬНЫХ УПРАВЛЯЮЩИХ ВОЗДЕЙСТВИЙ

Осуществляется синтез цифрового регулятора для электрогидравлической системы с астатическим объектом управления пятого порядка, содержащим в своем составе два колебательных звена. С помощью математического моделирования проводится проверка работы системы с данным объектом управления и рассчитанным для него цифровым регулятором. Полученные результаты рекомендуется использовать при практической реализации аналогичных цифровых систем автоматического управления.

### Введение

Для получения требуемой точности и качества переходного процесса системы автоматического управления (САУ) необходимо выбрать корректирующее устройство. При выборе корректирующего устройства предпочтение отдают тому из них, синтез которого может быть осуществлен аналитическим способом, а его реализация выполнена программно. Примером такого подхода может служить расчет цифровых регуляторов (ЦР), включенных последовательно с объектами управления (ОУ), имеющими различные непрерывные математические модели [1, 2]. В указанных работах осуществлен расчет цифровых регуляторов для объектов управления невысокого порядка (не выше четвертого). Однако на практике не всегда удается получить математическую модель ОУ, которую можно представить в упрощенном (редуцированном) виде. Примером может служить электрогидравлическая силовая следящая система с исполнительным гидроцилиндром [3], передаточная функция которой может быть четвертого, пятого и даже более высокого порядка. В связи с этим возникла необходимость расчета ЦР для астатического ОУ пятого порядка, который содержит два колебательных звена.

### 1. Функциональная схема САУ с объектом управления пятого порядка

Передаточная функция ОУ электрогидравлической системы с исполнительным гидроцилиндром имеет вид

$$G(s) = \alpha [s(s^2 + bs + a)(s^2 + ds + c)]^{-1}, \quad (1)$$

где  $\alpha = 30\,700\text{ с}^{-5}$ ;  $a = 225\text{ с}^{-2}$ ;  $b = 27\text{ с}^{-1}$ ;  $c = 250\text{ с}^{-2}$ ;  $d = 1,6\text{ с}^{-1}$ .

Обобщенная функциональная схема САУ с ОУ  $G(s)$  [2] показана на рис. 1. Сигналы входа  $u(t)$ , выхода  $x(t)$  и рассогласования  $\theta(t) = u(t) - x(t)$  в системе являются непрерывными функциями времени, поэтому при использовании ЦР необходимы аналого-цифровой (АЦП) и цифроаналоговый (ЦАП) преобразователи. ЦАП обычно представляет собой фиксатор нулевого порядка, а АЦП – квантователь с шагом  $h$ .

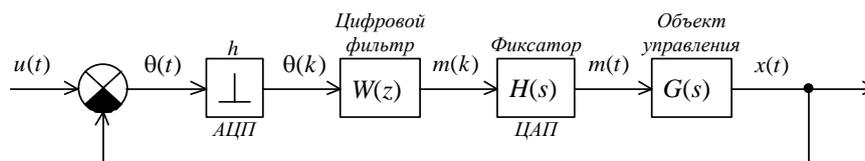


Рис. 1. Функциональная схема САУ

Наличие в системе ЦР позволяет при входном воздействии типа ступенчатой функции и нулевых начальных условиях осуществить оптимальный переходной процесс без перерегулирования за конечное и минимальное время. Для получения такого процесса в системе необходимо определить требуемую передаточную функцию  $W(z)$  ЦР. Для ОУ невысокого порядка (не выше четвертого) наиболее просто функция  $W(z)$  определяется численным методом переменного коэффициента усиления, изложенным в работах [1, 4]. Определение передаточной функции  $W(z)$  для системы с ОУ пятого порядка по данной методике является достаточно трудоемкой задачей. Значительно проще можно осуществить расчет ЦР, если использовать оптимальные управляющие воздействия на линейные ОУ, рассчитанные по методике [5].

## 2. Расчет цифрового регулятора

Для расчета составим схему аналогового моделирования (рис. 2), на которой ЦР представлен усилителем с переменным коэффициентом усиления. Такой усилитель располагается после фиксатора, причем согласно [1, 4] в любой момент времени  $t = \nu h^+$ ,  $\nu = 0, 1, 2, \dots$ , вход  $u_2$  и выход  $u'_2$  этого усилителя связаны линейным соотношением  $u'_2(\nu h^+) = K_\nu u_2(\nu h^+)$ , где  $K_\nu$  – постоянный коэффициент усиления усилителя в  $(\nu + 1)$ -м периоде прерывания  $h$  мгновенного ключа.

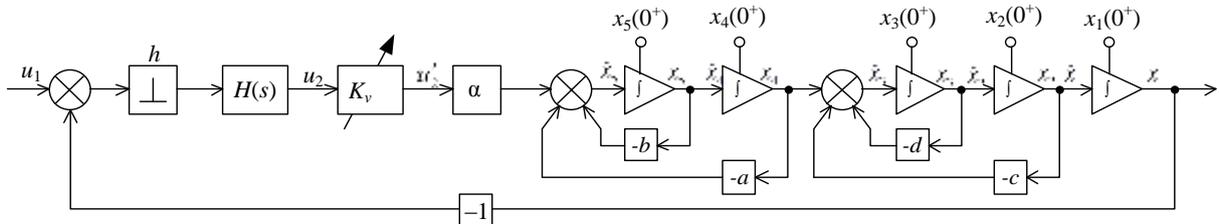


Рис. 2. Схема моделирования САУ с цифровым регулятором и аналоговым объектом управления

Запишем передаточную функцию ЦР в общем виде [1]:

$$W(z) = \frac{U'_2(z)}{U_2(z)} = \frac{K_0 u_2(0^+) + K_1 u_2(h^+) z^{-1} + K_2 u_2(2h^+) z^{-2} + K_3 u_2(3h^+) z^{-3} + K_4 u_2(4h^+) z^{-4}}{u_2(0^+) + u_2(h^+) z^{-1} + u_2(2h^+) z^{-2} + u_2(3h^+) z^{-3} + u_2(4h^+) z^{-4}} =$$

$$= K_0 \frac{1 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}}, \quad (2)$$

где

$$u_2(0^+) = 1; \quad a_1 = u_2(h^+); \quad a_2 = u_2(2h^+); \quad a_3 = u_2(3h^+); \quad a_4 = u_2(4h^+);$$

$$b_1 = \frac{K_1}{K_0} u_2(h^+); \quad b_2 = \frac{K_2}{K_0} u_2(2h^+); \quad b_3 = \frac{K_3}{K_0} u_2(3h^+); \quad b_4 = \frac{K_4}{K_0} u_2(4h^+).$$

Для ОУ (1) в работе [1] приведены оптимальные управляющие воздействия  $u'_2(\nu h^+)$  в виде коэффициентов передаточной функции (2):

$$K_0 = \frac{ac}{\alpha h(1 - 2\sqrt{B} \cos kh + B)(1 - 2\sqrt{D} \cos \mu h + D)}; \quad b_1 = -2(\sqrt{B} \cos kh + \sqrt{D} \cos \mu h);$$

$$b_2 = B + D + 4\sqrt{BD} \cos kh \cos \mu h; \quad b_3 = -2(B\sqrt{D} \cos \mu h + D\sqrt{B} \cos kh); \quad b_4 = BD; \quad (3)$$

$$B = e^{-bh}; \quad D = e^{-dh}; \quad \mu = \sqrt{c - \frac{d^2}{4}}; \quad k = \sqrt{a - \frac{b^2}{4}}.$$

Следовательно, задача синтеза ЦР сводится к определению недостающих коэффициентов  $a_1, a_2, a_3, a_4$  передаточной функции (2).

Используя схему аналогового моделирования, запишем дифференциальные уравнения состояния и уравнения переходных состояний (без учета переменного коэффициента усиления  $K_v$ ):

$$\begin{aligned} \dot{u}_1 &= 0; \quad \dot{x}_1 = x_2; \quad \dot{x}_2 = x_3; \quad \dot{x}_3 = x_4 - dx_3 - cx_2; \quad \dot{x}_4 = x_5; \quad \dot{x}_5 = \alpha u_2 - ax_4 - bx_5; \quad \dot{u}_2 = 0; \\ u_1(vh^+) &= u_1(vh); \quad x_1(vh^+) = x_1(vh); \quad x_2(vh^+) = x_2(vh); \quad x_3(vh^+) = x_3(vh); \\ x_4(vh^+) &= x_4(vh); \quad x_5(vh^+) = x_5(vh); \quad u_2(vh^+) = u_1(vh) - x_1(vh). \end{aligned}$$

Переписывая уравнения в векторно-матричной форме  $\dot{\mathbf{v}}(\tau) = \mathbf{A}\mathbf{v}(\tau)$  и  $\mathbf{v}(vh^+) = \mathbf{B}\mathbf{v}(vh)$ , находим

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -c & -d & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -a & -b & \alpha \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad \mathbf{v} = \begin{bmatrix} u_1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ u_2 \end{bmatrix}; \quad \mathbf{v}(0) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Методом комплексной плоскости [6] по матрице  $\mathbf{A}$  определим матрицу перехода  $\Phi(t) = e^{\mathbf{A}t} = L^{-1} \{ [s\mathbf{I} - \mathbf{A}]^{-1} \}$ , где  $\mathbf{I}$  – единичная матрица.

Для матрицы  $[s\mathbf{I} - \mathbf{A}]$  запишем выражение для нахождения обратной матрицы:

$$[s\mathbf{I} - \mathbf{A}]^{-1} = \frac{1}{\det|s\mathbf{I} - \mathbf{A}|} \mathbf{C}^T,$$

где  $\det|s\mathbf{I} - \mathbf{A}| = s^3(s^2 + sb + a)(s^2 + sd + c)$ ;  $\mathbf{C} = \begin{bmatrix} a_{11} & \dots & a_{1j} \\ \dots & \dots & \dots \\ a_{i1} & \dots & a_{ij} \end{bmatrix}$  – матрица алгебраических дополнений;  $a_{ij}$  – алгебраические дополнения ( $i=1..7, j=1..7$ ) матрицы  $[s\mathbf{I} - \mathbf{A}]$ .

После нахождения матрицы  $\mathbf{C}$  и подстановки получим обратную матрицу:

$$[s\mathbf{I} - \mathbf{A}]^{-1} = \begin{bmatrix} 1/s & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/s & (s+d)/sA_1 & 1/sA_1 & (s+b)/sA_1A_2 & 1/sA_1A_2 & \alpha/s^2A_1A_2 \\ 0 & 0 & (s+d)/A_1 & 1/A_1 & (s+b)/A_1A_2 & 1/A_1A_2 & \alpha/sA_1A_2 \\ 0 & 0 & -c/A_1 & s/A_1 & s(s+b)/A_1A_2 & s/A_1A_2 & \alpha/A_1A_2 \\ 0 & 0 & 0 & 0 & (s+b)/A_2 & 1/A_2 & \alpha/sA_2 \\ 0 & 0 & 0 & 0 & -a/A_2 & s/A_2 & \alpha/A_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/s \end{bmatrix}, \quad (4)$$

где  $A_1 = s^2 + sd + c$ ;  $A_2 = s^2 + sb + a$ .

Применим к матрице (4) обратное преобразование Лапласа [7] и с учетом усилителя с переменным коэффициентом усиления определим дискретную матрицу перехода в виде

$$\Phi(h, K_v) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & J & H & T & V & \alpha UK_v \\ 0 & 0 & F & N & S & K & \alpha VK_v \\ 0 & 0 & -cN & G & P & O & \alpha KK_v \\ 0 & 0 & 0 & 0 & L & M & \alpha RK_v \\ 0 & 0 & 0 & 0 & -aM & C & \alpha MK_v \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

где  $B_1 = \frac{c}{a}(b^3 - b^2d - 2ab + bc + ad)$ ;  $B_2 = \frac{c}{a}(b^4 - b^3d - 4ab^2 + b^2c + 3abd + 2a^2 - 2ac)$ ;

$$D_1 = \frac{a}{c}(d^3 - bd^2 - 2cd + bc + ad)$$
;  $D_2 = \frac{a}{c}(d^4 - bd^3 - 4cd^2 + ad^2 + 3bcd + 2c^2 - 2ac)$ ;

$$M = \frac{1}{k}\sqrt{B} \sin kh$$
;  $N = \frac{1}{\mu}\sqrt{D} \sin \mu h$ ;  $H = \frac{1}{c}\left[1 - \sqrt{D}\left(\cos \mu h + \frac{d}{2\mu} \sin \mu h\right)\right]$ ;

$$J = \frac{d}{c}\left[1 - \sqrt{D}\left(\cos \mu h + \frac{d^2 - 2c}{2d\mu} \sin \mu h\right)\right]$$
;

$$P = aI\sqrt{B}\left[(d-b)\cos kh - \frac{b^2 - bd - 2(a-c)}{2k} \sin kh\right] +$$

$$+ I\sqrt{D}\left[a(b-d)\cos \mu h + \frac{2b^2c - 2ac - 2bcd + 2c^2 + ad^2 - abd}{2\mu} \sin \mu h\right]$$
;

$$I = \frac{1}{(c-a)^2 + (bc-ad)(b-d)}$$
;  $C = \sqrt{B}\left(\cos kh - \frac{b}{2k} \sin kh\right)$ ;

$$K = I\sqrt{B}\left[(b-d)\cos kh + \frac{b^2 - bd - 2(a-c)}{2k} \sin kh\right] + I\sqrt{D}\left[(d-b)\cos \mu h + \frac{d^2 - bd - 2(c-a)}{2\mu} \sin \mu h\right]$$
;

$$O = I\sqrt{B}\left[(c-a)\cos kh + \frac{2ad - b(a+c)}{2k} \sin kh\right] + I\sqrt{D}\left[(a-c)\cos \mu h + \frac{2bc - d(c+a)}{2\mu} \sin \mu h\right]$$
;

$$F = \sqrt{D}\left(\cos \mu h + \frac{d}{2\mu} \sin \mu h\right)$$
;  $G = \sqrt{D}\left(\cos \mu h - \frac{d}{2\mu} \sin \mu h\right)$ ;

$$V = \frac{-I\sqrt{B}}{a}\left[(b^2 - bd + c - a)\cos kh + \frac{b^3 - b^2d - 3ab + bc + 2ad}{2k} \sin kh\right] -$$

$$- \frac{I\sqrt{D}}{c}\left[(d^2 - bd - c + a)\cos \mu h + \frac{d^3 - bd^2 + ad - 3cd + 2bc}{2\mu} \sin \mu h\right] + \frac{1}{ac}$$
;

$$S = I\sqrt{B}\left[(b^2 - bd + c - a)\cos kh + \frac{b^3 - b^2d - 3ab + bc + 2ad}{2k} \sin kh\right] -$$

$$-I\sqrt{D} \left[ (b^2 - bd + c - a)\cos\mu h + \frac{b^2d - bd^2 + cd - 2ab + ad}{2\mu} \sin\mu h \right];$$

$$T = \frac{-I\sqrt{B}}{c} \left[ B_1 \cos kh + \frac{B_2}{2k} \sin kh \right] + \frac{I\sqrt{D}}{c} \left[ (b^2d - bd^2 - ab + cd)\cos\mu h + \left( \frac{2c(a - c + bd - b^2) + cd^2}{2\mu} + \frac{b(bd^2 - ad - d^3)}{2\mu} \right) \sin\mu h \right] + \frac{b}{ac};$$

$$U = \frac{I\sqrt{B}}{ac} \left[ B_1 \cos kh + \frac{B_2}{2k} \sin kh \right] + \frac{I\sqrt{D}}{ac} \left[ D_1 \cos\mu h + \frac{D_2}{2\mu} \sin\mu h \right] + \frac{ach - bc - ad}{a^2c^2}; R = \frac{1}{a} \left[ 1 - \sqrt{B} \left( \cos kh + \frac{b}{2k} \sin kh \right) \right]; L = \sqrt{B} \left( \cos kh + \frac{b}{2k} \sin kh \right).$$

Далее последовательно определяем требуемые векторы состояния  $\mathbf{v}(vh^+)$  в интервалах прерывания мгновенного ключа:

$$\mathbf{v}^T(0^+) = (\mathbf{B}\mathbf{v}(0))^T = [1; 0; 0; 0; 0; 0; 1];$$

$$\mathbf{v}^T(h) = (\Phi(h, K_0)\mathbf{v}(0^+))^T = [1; \alpha UK_0; \alpha VK_0; \alpha KK_0; \alpha RK_0; \alpha MK_0; 1];$$

$$\mathbf{v}^T(h^+) = (\mathbf{B}\mathbf{v}(h))^T = [1; \alpha UK_0; \alpha VK_0; \alpha KK_0; \alpha RK_0; \alpha MK_0; Q];$$

$$\mathbf{v}^T(2h) = (\Phi(h, K_1)\mathbf{v}(h^+))^T = [1; \alpha F_1K_0 + \alpha UQK_1; \alpha F_2K_0 + \alpha VQK_1;$$

$$\alpha F_3K_0 + \alpha KQK_1; \alpha F_4K_0 + \alpha RQK_1; \alpha M(F_5K_0 + QK_1); Q];$$

$$\mathbf{v}^T(2h^+) = (\mathbf{B}\mathbf{v}(2h))^T = [1; \alpha F_1K_0 + \alpha UQK_1; \alpha F_2K_0 + \alpha VQK_1;$$

$$\alpha F_3K_0 + \alpha KQK_1; \alpha F_4K_0 + \alpha RQK_1; \alpha M(F_5K_0 + QK_1); W];$$

$$\mathbf{v}^T(3h) = (\Phi(h, K_2)\mathbf{v}(2h^+))^T = [1; \alpha S_1K_0 + \alpha F_1QK_1 + \alpha UWK_2; \alpha S_2K_0 + \alpha F_2QK_1 + \alpha VWK_2;$$

$$\alpha S_3K_0 + \alpha F_3QK_1 + \alpha KWK_2; \alpha S_4K_0 + \alpha F_4QK_1 + \alpha RWK_2; \alpha M(S_5K_0 + F_5QK_1 + WK_2); W];$$

$$\mathbf{v}^T(3h^+) = (\mathbf{B}\mathbf{v}(3h))^T = [1; \alpha S_1K_0 + \alpha F_1QK_1 + \alpha UWK_2; \alpha S_2K_0 + \alpha F_2QK_1 + \alpha VWK_2;$$

$$\alpha S_3K_0 + \alpha F_3QK_1 + \alpha KWK_2; \alpha S_4K_0 + \alpha F_4QK_1 + \alpha RWK_2; \alpha M(S_5K_0 + F_5QK_1 + WK_2); Y];$$

$$\mathbf{v}(4h) = \Phi(h, K_3)\mathbf{v}(3h^+) = \begin{bmatrix} 1 \\ \alpha L_1K_0 + \alpha S_1QK_1 + \alpha F_1WK_2 + \alpha UYK_3 \\ \alpha L_2K_0 + \alpha S_2QK_1 + \alpha F_2WK_2 + \alpha VYK_3 \\ \alpha L_3K_0 + \alpha S_3QK_1 + \alpha F_3WK_2 + \alpha KYK_3 \\ \alpha L_4K_0 + \alpha S_4QK_1 + \alpha F_4WK_2 + \alpha RYK_3 \\ \alpha M(L_5K_0 + S_5QK_1 + F_5WK_2 + YK_3) \\ Y \end{bmatrix};$$

$$\mathbf{v}(4h^+) = \mathbf{B}\mathbf{v}(4h) = \begin{bmatrix} 1 \\ \alpha L_1 K_0 + \alpha S_1 Q K_1 + \alpha F_1 W K_2 + \alpha U Y K_3 \\ \alpha L_2 K_0 + \alpha S_2 Q K_1 + \alpha F_2 W K_2 + \alpha V Y K_3 \\ \alpha L_3 K_0 + \alpha S_3 Q K_1 + \alpha F_3 W K_2 + \alpha K Y K_3 \\ \alpha L_4 K_0 + \alpha S_4 Q K_1 + \alpha F_4 W K_2 + \alpha R Y K_3 \\ \alpha M(L_5 K_0 + S_5 Q K_1 + F_5 W K_2 + Y K_3) \\ Z \end{bmatrix},$$

где

$$\begin{aligned} F_1 &= U + JV + HK + TR + VM; F_2 = FV + NK + SR + KM; F_3 = -cNV + GK + PR + OM; \\ F_4 &= LR + M^2; F_5 = -aR + C; S_1 = F_1 + JF_2 + HF_3 + TF_4 + VMF_5; S_2 = FF_2 + NF_3 + SF_4 + KMF_5; \\ S_3 &= -cNF_2 + GF_3 + PF_4 + OMF_5; S_4 = LF_4 + M^2 F_5; S_5 = -aF_4 + CF_5; \\ L_1 &= S_1 + JS_2 + HS_3 + TS_4 + VMS_5; L_2 = FS_2 + NS_3 + SS_4 + KMS_5; \\ L_3 &= -cNS_2 + GS_3 + PS_4 + OMS_5; L_4 = LS_4 + M^2 S_5; L_5 = -aS_4 + CS_5; Q = 1 - \alpha UK_0; \\ W &= 1 - \alpha F_1 K_0 - \alpha U Q K_1; Y = 1 - \alpha S_1 K_0 - \alpha F_1 Q K_1 - \alpha U W K_2; \\ Z &= 1 - \alpha L_1 K_0 - \alpha S_1 Q K_1 - \alpha F_1 W K_2 - \alpha U Y K_3. \end{aligned} \quad (5)$$

Из векторов состояний  $\mathbf{v}(0^+)$ ,  $\mathbf{v}(h^+)$ ,  $\mathbf{v}(2h^+)$ ,  $\mathbf{v}(3h^+)$ ,  $\mathbf{v}(4h^+)$  находим

$$u_2(0^+) = 1; a_1 = u_2(h^+) = Q; a_2 = u_2(2h^+) = W; a_3 = u_2(3h^+) = Y; a_4 = u_2(4h^+) = Z. \quad (6)$$

Перепишем коэффициенты  $b_1$ ,  $b_2$ ,  $b_3$  выражения (2) с учетом выражений (6):

$$b_1 = \frac{K_1}{K_0} Q; b_2 = \frac{K_2}{K_0} W; b_3 = \frac{K_3}{K_0} Y. \quad (7)$$

Из формул (7) выразим коэффициенты

$$K_1 = \frac{b_1 K_0}{Q}; K_2 = \frac{b_2 K_0}{W}; K_3 = \frac{b_3 K_0}{Y}. \quad (8)$$

Подставляя  $K_1$ ,  $K_2$ ,  $K_3$  из (8) и  $Q$ ,  $W$ ,  $Y$ ,  $Z$  из (5) в (6), запишем коэффициенты

$$\begin{aligned} a_1 &= 1 - \alpha UK_0; \\ a_2 &= 1 - \alpha F_1 K_0 - \alpha U Q K_1 = 1 - \alpha F_1 K_0 - \alpha U b_1 K_0 = 1 - \alpha K_0 (F_1 + b_1 U); \\ a_3 &= 1 - \alpha S_1 K_0 - \alpha F_1 Q K_1 - \alpha U W K_2 = 1 - \alpha S_1 K_0 - \alpha F_1 b_1 K_0 - \alpha U b_2 K_0 = \\ &= 1 - \alpha K_0 (S_1 + b_1 F_1 + b_2 U); \\ a_4 &= 1 - \alpha L_1 K_0 - \alpha S_1 Q K_1 - \alpha F_1 W K_2 - \alpha U Y K_3 = 1 - \alpha L_1 K_0 - \alpha S_1 b_1 K_0 - \alpha F_1 b_2 K_0 - \alpha U b_3 K_0 = \\ &= 1 - \alpha K_0 (L_1 + b_1 S_1 + b_2 F_1 + b_3 U). \end{aligned} \quad (9)$$

По формулам (9) после ряда преобразований найдем коэффициенты  $a_1, a_2, a_3, a_4$  передаточной функции ЦР (2) в окончательном виде:

$$a_1 = 1 - \frac{\alpha K_0}{ac} \left[ \sqrt{BI} \left( B_1 \cos kh + \frac{B_2}{2k} \sin kh \right) + \sqrt{DI} \left( D_1 \cos \mu h + \frac{D_2}{2\mu} \sin \mu h \right) - \frac{ad + bc}{ac} + h \right];$$

$$\begin{aligned}
 a_2 &= 1 - \frac{\alpha K_0}{ac} \left[ \sqrt{BI} \left( (B_1 + 2D_1) \cos kh - \frac{B_2}{2k} \sin kh \right) + \sqrt{DI} \left( (D_1 + 2B_1) \cos \mu h - \frac{D_2}{2\mu} \sin \mu h \right) - I (BB_1 + \right. \\
 &\quad \left. + DD_1) - 2\sqrt{BDI} \left( \left( B_1 \cos kh + \frac{B_2}{2k} \sin kh \right) \cos \mu h + \left( D_1 \cos \mu h + \frac{D_2}{2\mu} \sin \mu h \right) \cos kh \right) + h(1 + b_1) \right]; \quad (10) \\
 a_3 &= 1 - \frac{\alpha K_0}{ac} \left[ \sqrt{BDI} \left( (B_1 + 2D_1) \cos kh + \frac{B_2}{2k} \sin kh \right) + B\sqrt{DI} \left( (D_1 + 2B_1) \cos \mu h + \frac{D_2}{2\mu} \sin \mu h \right) - \right. \\
 &\quad \left. - I (BD_1 + DB_1) - 2\sqrt{BDI} \left( \left( B_1 \cos kh - \frac{B_2}{2k} \sin kh \right) \cos \mu h + \left( D_1 \cos \mu h - \frac{D_2}{2\mu} \sin \mu h \right) \cos kh \right) + h(1 + b_1 + b_2) \right]; \\
 a_4 &= 1 - \frac{\alpha K_0}{ac} \left[ \sqrt{BDI} \left( B_1 \cos kh - \frac{B_2}{2k} \sin kh \right) + B\sqrt{DI} \left( D_1 \cos \mu h - \frac{D_2}{2\mu} \sin \mu h \right) - BD \frac{ad + bc}{ac} + h(1 + b_1 + b_2 + b_3) \right].
 \end{aligned}$$

Из полученных выражений видно, что недостающие коэффициенты передаточной функции ЦР определяются непосредственно через параметры  $\alpha$ ,  $a$ ,  $b$ ,  $c$  и  $d$  передаточной функции ОУ (1) и шаг квантования  $h$ .

### 3. Определение параметров цифрового регулятора и моделирование работы системы

Для объекта управления (1) по формулам (3) и (10) определим параметры передаточной функции оптимального ЦР (2) при шаге квантования  $h = 0,1с$  :

$$W(z) = 14,9640 \frac{1 - 0,3962z^{-1} + 0,9130z^{-2} - 0,3497z^{-3} + 0,0573z^{-4}}{1 + 0,9773z^{-1} + 0,6462z^{-2} + 0,1352z^{-3} + 0,0034z^{-4}}. \quad (11)$$

Проверку работы системы (см. рис. 2) осуществим путем моделирования переходных процессов в среде Simulink пакета MATLAB.

Схема моделирования (рис. 3) состоит из АЦП, представленного фиксатором нулевого порядка (ZOH), ЦР ( $K_0, W(z)$ ), усилителей Alfa,  $a, b, c, d$  и интеграторов Int1 – Int5, реализующих передаточную функцию (1). Блоки ZOH и  $W(z)$  работают с шагом квантования  $h$ .

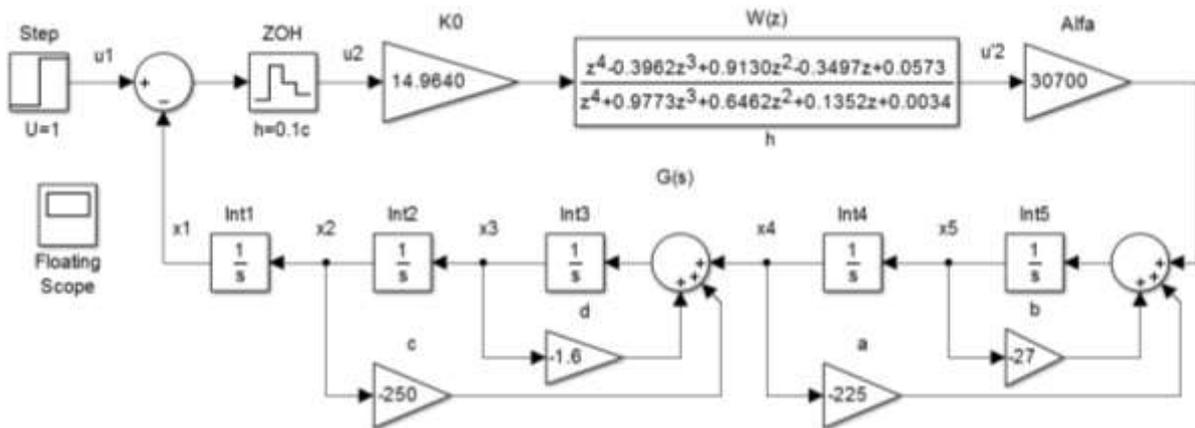


Рис. 3. Схема моделирования работы системы с оптимальным цифровым регулятором

На рис. 4 показаны переходные процессы в системе с объектом  $G(s)$  и ЦР  $W(z)$  при единичном ступенчатом воздействии.

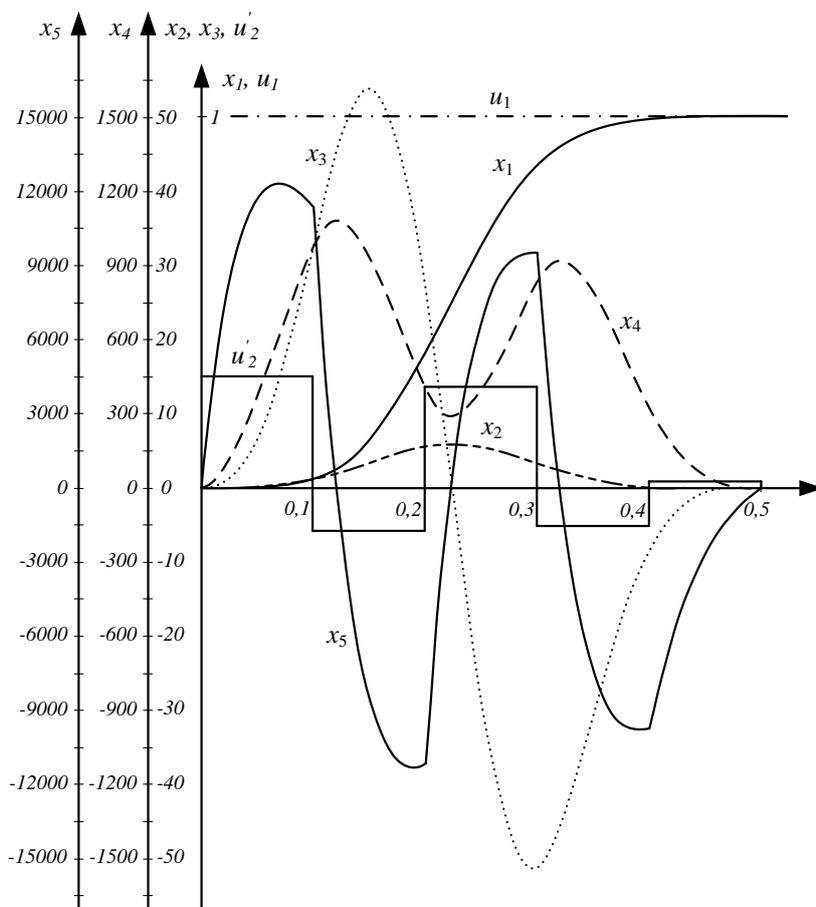


Рис. 4. Переходные процессы в системе с оптимальным ЦР  $W(z)$

Реакция системы (см. рис. 3) на единичное ступенчатое воздействие длится пять шагов квантования  $h = 0,1$  с, и переходной процесс заканчивается за время, равное 0,5 с.

### Заключение

В работе осуществлен синтез ЦР  $W(z)$  для электрогидравлической силовой следящей системы с исполнительным гидроцилиндром, передаточная функция  $G(s)$  которого представлена астатическим ОУ пятого порядка, содержащим в своем составе два колебательных звена. Для синтеза регулятора применен численный метод переменного коэффициента усиления с использованием оптимальных управляющих воздействий. В результате расчетов получены аналитические выражения (2) и (10), которые позволяют определить недостающие коэффициенты передаточной функции ЦР через параметры ОУ и шаг квантования  $h$ . Данный метод может быть использован для различных передаточных функций линейных ОУ. Для проверки работы системы с ОУ  $G(s)$  и рассчитанным для него ЦР  $W(z)$  осуществлено математическое моделирование. При подаче на вход ступенчатой функции получена оптимальная (без перерегулирования) реакция на выходе системы, длительность которой составляет пять шагов квантования  $h$ . Синтезированный ЦР достаточно прост в реализации и может быть рекомендован к практическому применению в аналогичных цифровых САУ, имеющих ОУ пятого порядка, которые содержат интегрирующее и два колебательных звена, но с другими параметрами.

### Список литературы

1. Гостев, В.И. Системы автоматического управления с цифровыми регуляторами : справочник / В.И. Гостев, В.К. Стеклов. – Киев : Радиоаматор, 1998. – 704 с.

2. Гостев, В.И. Синтез цифровых регуляторов систем автоматического управления / В.И. Гостев, Д.А. Худолий, А.А. Баранов. – Киев : Радиоаматор, 2000. – 400 с.
3. Баунин, В.Г. Моделирование цифровой электрогидравлической следящей системы с силовым гидроцилиндром в среде MATLAB / В.Г. Баунин, Н.В. Швецов // Труды Второй Всерос. науч. конф. «Проектирование инженерных и научных приложений в среде MATLAB». Секция 3. Проектирование систем автоматического управления и регулирования. Часть 1. – М. : ИПУ РАН, 2004. – С. 841–858.
4. Ту, Ю. Современная теория управления / Ю. Ту. – М. : Машиностроение, 1971. – 472 с.
5. Гостев, В.И. Определение оптимальных управляющих воздействий на линейные объекты регулирования в системах управления с цифровыми регуляторами / В.И. Гостев // Автоматика. – 1987, № 3. – С. 56–61.
6. Козырев, В.Д. Применение цифровых ЭВМ при исследовании автоматических систем РЭС / В.Д. Козырев. – Киев : КВИРТУ ПВО, 1976. – 183 с.
7. Корн, Г.А. Справочник по математике для научных сотрудников и инженеров / Г.А. Корн, Т.М. Корн. – М. : Наука, 1974. – 832 с.

Поступила 09.09.13

*НПООО «ОКБ Техносоюзпроект»,  
Минск, пр. Независимости, 115  
e-mail: lednikg@yandex.ru*

**A.G. Stryzhniou, H.V. Lednik, A.N. Rusakovich**

### **SYNTHESIS OF OPTIMAL DIGITAL CONTROLLER FOR THE FIFTH ORDER CONTROLLED OBJECT**

Synthesis of optimal digital controller for an electro-hydraulic system with the fifth order non-static controlled object containing oscillatory elements has been carried out. The system performance with such object and the designed optimal digital controller are verified via mathematical simulation. The achieved results are recommended to be used in practical implementation of similar digital automatic control systems.

## ОБРАБОТКА СИГНАЛОВ, ИЗОБРАЖЕНИЙ И РЕЧИ

УДК [004.522+004.934+004.91]:004.89

Ю.С. Гецэвіч, Т.І. Окрут, Б.М. Лабанаў

АЛГАРЫТМЫ ІДЭНТЫФІКАЦЫІ РЭПЛІК СА СЛОВАМІ АЎТАРА  
Ў ЭЛЕКТРОННЫХ ТЭКСТАХ НА БЕЛАРУСКАЙ МОВЕ

*Разглядаюцца асноўныя этапы стварэння аўтаматызаваных алгарытмаў для ідэнтыфікацыі рэплік з устаўкамі слоў аўтара, прапанованаецца іх дапрацоўка ў мованезалежным напрамку. Прыводзяцца вынікі ацэнкі працы распрацаваных мадэляў на трэніровачным і тэставым тэкстах з дакладнасцю ў тэрмінах сярэдняй гарманічнай меры больш за 90 %.*

## Уводзіны

У цяперашні час інфармацыя выконвае вельмі значную ролю ў жыцці чалавека, пранізвае ўсе сферы чалавечай дзейнасці. Яе шырокае прымяненне складае неабходную аснову функцыянавання сучаснага грамадства. Пры гэтым асабліва цэняцца аб'ём і хуткасць яе атрымання, чаму значна садзейнічае з'яўленне інтэрнэта, з дапамогай якога даступна вялікая колькасць электронных кніг, у тым ліку і ў агульным выглядзе. Аднак чытанне тэксту з экрана можа сапсаваць зрок, а дыктарская (акторская) агучка кнігі патрабуе вялікіх выдаткаў часу і сродкаў для стварэння гукавага запісу і яго карэктыроўкі.

Альтэрнатыўным спосабам агучвання тэксту кнігі выступае выкарыстанне сістэм сінтэзу маўлення па тэксце (ССМТ). За кароткі час такія сістэмы здольны стварыць электронны гукавы файл па ўваходным электронным тэксце. Для таго каб сінтэзаванае маўленне было разборлівым, прымяняюцца сістэмы перадапрацоўкі літарна-сімвальных канструкцый, пошуку невядомых слоў і зняцця шматзначнасці слоў у электронных тэкстах [1].

Тэкст кнігі можа ўключаць не толькі словы аўтара, але і словы двух ці больш моўцаў (персанажаў). Гэта абгрунтоўвае матываванае жаданне стваральніка аўдыёкніг выкарыстаць розных дыктараў ці сінтэзаваных галасы, каб аўдыёкніга была больш набліжанай да адлюстравання ўнікальных характарыстык маўлення персанажаў. Разам з тым час разметкі рэплік для чытання іх рознымі дыктарамі ці сінтэзатарамі маўлення застаецца вялікім з-за патрэбы рэдактарскай прачыткі ўсяго тэксту перад агучваннем для абазначэння слоў персанажаў.

Калі рабіць агляд, то на сённяшні дзень ужо існуюць некаторыя напрацоўкі, звязаныя з гэтым напрамкам аналізу тэксту. Так, напрыклад, анлайн-сістэма Text Analysis Demo дазваляе ідэнтыфікаваць персанажаў у тэксце і іх выказванні, што ў адносінах да ССМТ з'яўляецца важным момантам у вызначэнні роду моўцы [2, 3]. Таксама групай еўрапейскіх навукоўцаў былі распрацаваны алгарытмы па ідэнтыфікацыі персанажаў і аўтаматычнаму вызначэнню іх ролі ў творы з дапамогай сінтаксічных граматык NooJ [4]. Што датычыцца славянскіх моў, можна адзначыць працу харвацкіх навукоўцаў па вызначэнні простага мовы тэксту [5], але ў ёй не разглядаецца праблема вызначэння роду моўцаў. У гэтым напрамку распрацоўваюцца такія сродкі стварэння аўдыёкніг, як праграмы MP3book2005 і AUDIOBOOK [6, 7]. У іх убудаваны спецыяльныя блокі лагічнага аналізу дыялогаў, якія рэалізуюць разметку слоў персанажа і слоў аўтара ў дыялагічным тэксце. У AUDIOBOOK былі ажыццёўлены крокі ў бок рэалізацыі чытання па ролях, але праграма ахоплівае не ўсе выпадкі. У ёй ігнаруецца структура афармлення простага мовы, калі ёсць словы моўцы з некалькімі аўтарскімі устаўкамі. Таксама не разглядаецца магчымасць вызначыць род моўцы па іншых вызначальніках, напрыклад па спалучэннях тыпу «дзеяслоў + назоўнік у мужчынскім родзе» ва ўстаўках аўтара:

– Надо написать «ять», – отвечает ученик.

Варта адзначыць, што праграма AUDIOBOOK лепш працуе з рускамоўнымі і англамоўнымі маўленчымі рухавічкамі, а блокі лагічнага аналізу дыялогаў увогуле не прадугледжваюць працу з іншымі мовамі, акрамя рускай.

Такім чынам, дадзены артыкул працягвае распачатую ў артыкуле [8] тэму распрацоўкі алгарытмаў ідэнтыфікацыі прастай мовы і вызначэння роду моўцаў па ўстаўках слоў аўтара ў электронным тэксце. Перад аўтарамі ставіцца задача распрацоўкі мованезалежных мадэляў і стварэння асобных лінгвістычных рэсурсаў для іх з мэтай фармалізацыі як можна большай колькасці сінтаксічных структур дыялагічных рэплік і вызначэння роду моўцаў (персанажаў) па ўстаўленых у простую мову словах аўтара. Разглядаюцца варыянты выкарыстання распрацаваных алгарытмаў у ССМТ, для стварэння аўдыёкніг ці ў якасці дапаможнага прыкладання ў працы дыктараў (рэдактараў).

### 1. Збор і сістэматызацыя матэрыялу для вызначэння роду моўцаў

Для распрацоўкі алгарытмаў экспертамі быў выбраны твор Уладзіміра Караткевіча «Каласы пад сярпом тваім». Спачатку ў якасці трэніровачнага матэрыялу выкарыстоўваліся першыя 12 раздзелаў (падрабязную інфармацыю можна знайсці ў артыкуле [8]), далей былі дабаўлены астатнія раздзелы – усяго 32).

Усе алгарытмы будаваліся з дапамогай беларускага модуля міжнароднай лінгвістычнай праграмы NooJ [9]. Гэта праграма дазваляе распрацоўваць сінтаксічныя і марфалагічныя граматыкі і тэставаць іх на вялікай колькасці тэкстаў. З іх дапамогай можна потым ствараць сінтаксічныя анатацыі і экспертаваць размечаны тэкст як файл XML для далейшай апрацоўкі [10].

Заўважым, што NooJ апрацоўвае кожны наступны абзац тэксту асобна ад папярэдняга. Таму кожны абзац (усяго іх 5748) трэніровачнага тэкставага корпусу быў асобна прааналізаваны экспертам у наступным парадку:

- пазначэнне абзацаў з прастай мовай;
- пазначэнне рэплік з устаўкамі слоў аўтара;
- пазначэнне рэплік моўцаў (персанажаў) мужчынскага і жаночага роду (адпаведна падкрэслены шрыфт для мужчынскага роду і курсіўны шрыфт для жаночага роду (табл. 1)).

Табліца 1

Фрагмент трэніровачнага матэрыялу з ручной разметкай для абзацаў з прастай мовай

Паметы прастай мовы	Простая мова з устаўкамі слоў аўтара	Тэкст твору «Каласы пад сярпом тваім» па абзацах
1	1	– <i>Бацька вады, – шэптам сказала Майка.</i>
1	1	– <u>Бацька вод, – паправіў Алесь. – Вось так і Дняпро пачынаецца недзе.</u>
1	1	– <i>Жывая вада, – сказала Яня.</i>
0	0	І яна апусцілася на калені і зламала пальчыкамі крышталную паверхню.
1	0	– <i>Піце. Будзеце жыць сто год...</i>

Ручная апрацоўка паказала, што тэкст налічвае 2805 абзацаў з прастай мовай (з іх 2580 простая мова, якая пачынаецца з працяжніка, 225 – без працяжніка) і 1281 рэплік з устаўкамі слоў аўтара (сярод іх 1075 рэплік моўцаў мужчынскага роду, 197 – жаночага роду і 9 – невыразных моўцаў).

На пачатковым этапе на аснове аналізу дыялагічных рэплік былі выведзены ніжэй прыведзены структуры афармлення прастай мовы дыялогаў з наступнымі абазначэннямі: М – словы моўцы; А – словы аўтара; дужкі ( , ) – пачатак і завяршэнне набору варыяцый знакаў прыпынку; сімвал вертыкальнай лініі | – выкарыстоўваецца для раздзялення магчымых варыяцый знакаў прыпынку (аператар «або»).

1. Словы моўцы без слоў аўтара:

– М (! | !! | !!! | ? | ?! | ... | ).

2. Словы моўцы са словамі аўтара ў канцы:

–  $M(, | ! | ! | ! | ! | ? | ? | ? | \dots | . ) - A( \dots | . )$ .

3. Словы моўцы з некалькімі аўтарскімі ўстаўкамі:

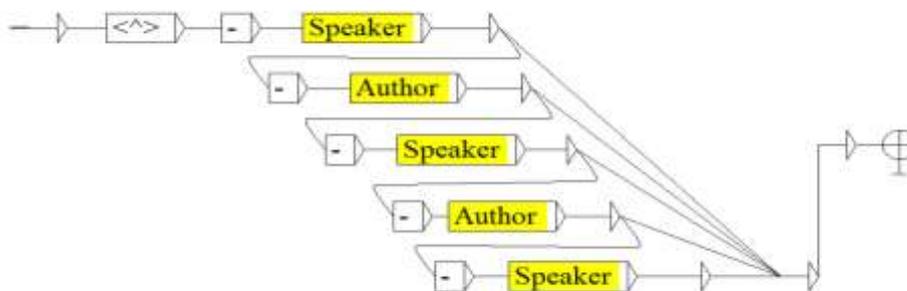
–  $M(, | ! | ! | ! | ! | ? | ? | ? | \dots | . ) - A(, | \dots | . | : | . ) - M(, | ! | ! | ! | ! | ? | ? | ? | \dots | . )$

(–  $A(, | \dots | . | : | . ) - M(, | ! | ! | ! | ! | ? | ? | ? | \dots | . )$ ).

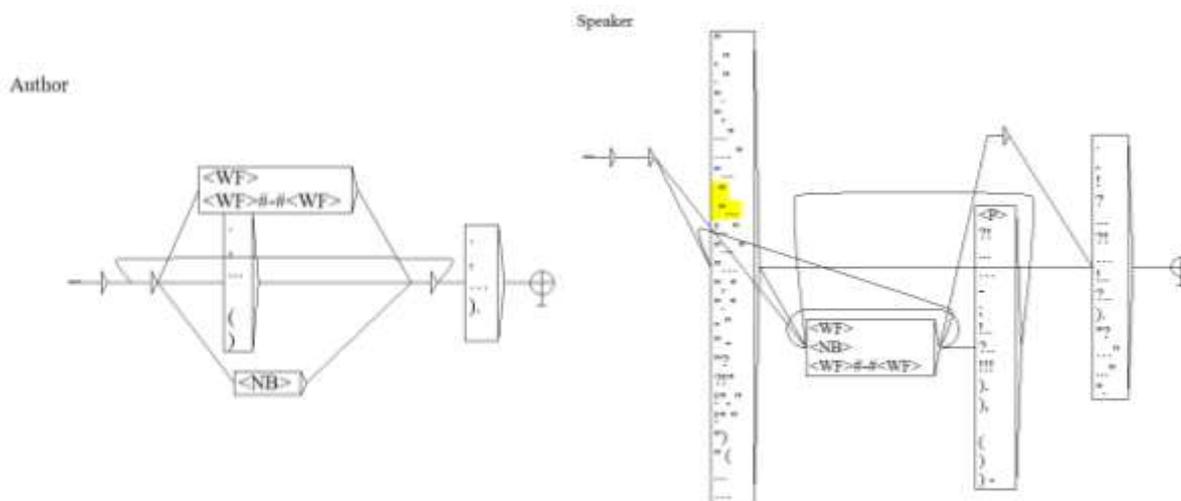
Варта адзначыць, што падчас трэніроўкі і тэсціроўкі распрацаваных алгарытмаў дадзеныя структуры амаль не падвергліся зменам, былі толькі дададзены некалькі спалучэнняў з двукоссем і правай круглай дужкай (унутраныя знакі прыпынку ў словах аўтара або моўцы разглядаюцца асобна).

## 2. Распрацоўка аўтаматызаваных алгарытмаў выяўлення слоў моўцаў і слоў аўтара

На аснове табл. 1 і фармалізаваных структур афармлення простаі мовы была распрацавана сінтаксічная граматыка NooJ з назвай DS\_All для аўтаматызаванага вызначэння ўсіх абзацаў з простаі мовай (мал. 1). Яе галоўныя часткі-падграфы Speaker і Author ідэнтыфікуюць адпаведна словы моўцы (персанажа) і словы аўтара (мал. 2).



Мал. 1. Агульны выгляд сінтаксічнай граматыкі DS\_All



Мал. 2. Падграфы Speaker і Author граматыкі DS\_All, дзе WF – любая словаформа, NB – любая паслядоўнасць лічбаў

Спрошчана апішам працу граматыкі для ідэнтыфікацыі першай і другой фармалізаваных структур простаі мовы, а пасля абагульнім яе працу для трэцяй структуры. Паводле граматыкі DS\_All простая мова пачынаецца з працяжніка, далей ідуць словы моўцы Speaker (лікі, розныя формы слоў са знакамі прыпынку), якія, у сваю чаргу, могуць скончыцца на кропку, коску, клічнік, пыталынік або іх камбінацыю (таксама былі дабаўлены некаторыя спалучэнні з двукоссем). Калі пасля слоў моўцы не ідуць словы аўтара, то граматыка завяршае працу. Так будзе ідэнтыфікавана

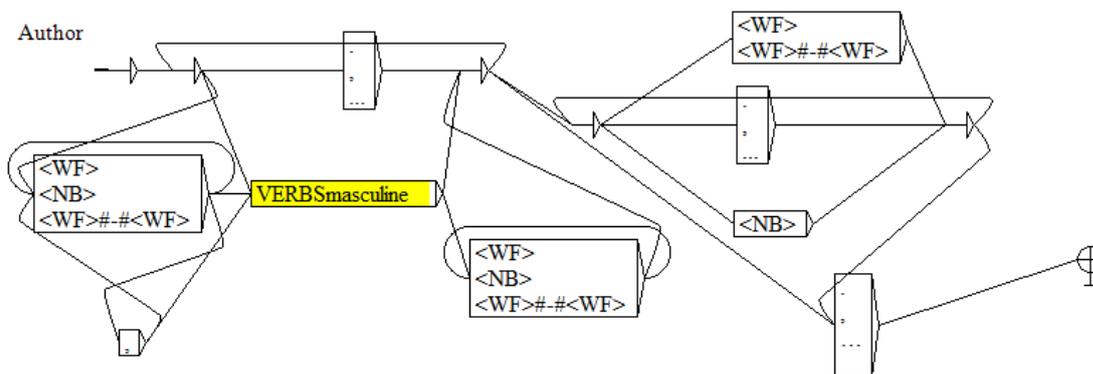
структура першага тыпу. Калі пасля іх зноў ідзе працяжнік, то граматыка задзейнічае падграф Author (таксама ўключае розныя формы слоў, такія знакі прыпынку, як коска, кропка, шматкроп'е і дужкі). Так будзе ідэнтыфікавана структура другога тыпу.

Аналагічна далей граматыка можа ідэнтыфікаваць ад адной да двух уставак слоў аўтара ў простую мову персанажа праз спрацоўванне паслядоўных падграфаў Speaker → Author → Speaker. Так будзе ідэнтыфікавана структура трэцяга тыпу.

### 3. Распрацоўка аўтаматызаваных алгарытмаў ідэнтыфікацыі роду моўцаў па словах аўтара

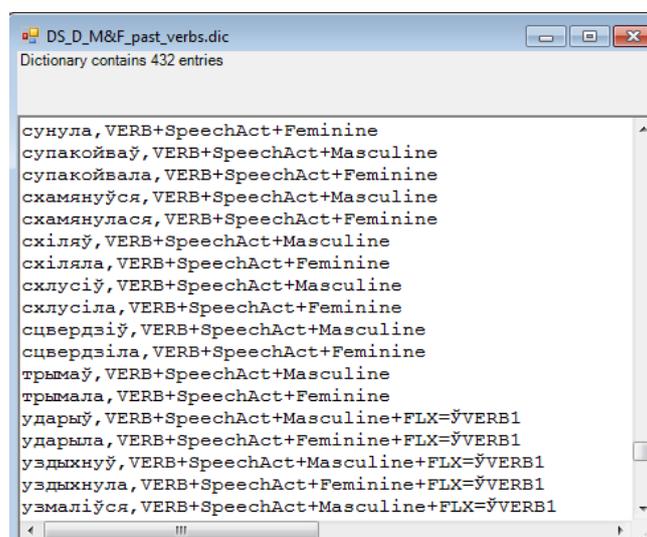
Найбольш прыдатнымі для ідэнтыфікацыі роду моўцы апынуліся дзеясловы дзеяння мінулага часу (паправіў, сказала) персанажаў.

На аснове родазалежнай разметкі тэксту (гл. табл. 1) былі пабудаваны граматыкі для абазначэння моўцаў мужчынскага і жаночага роду. Для гэтага граф Author граматыкі DS\_All быў дапрацаваны. Спачатку ў яго быў даданы родазалежны падграф адпаведна для мужчынскага роду – VERBSmasculine (мал. 3), аналагічна для жаночага – VERBSfeminine. У выніку на аснове графа DirectSpeech былі пабудаваны дзве асобныя граматыкі: DirectSpeechMasculine і DirectSpeechFeminine.



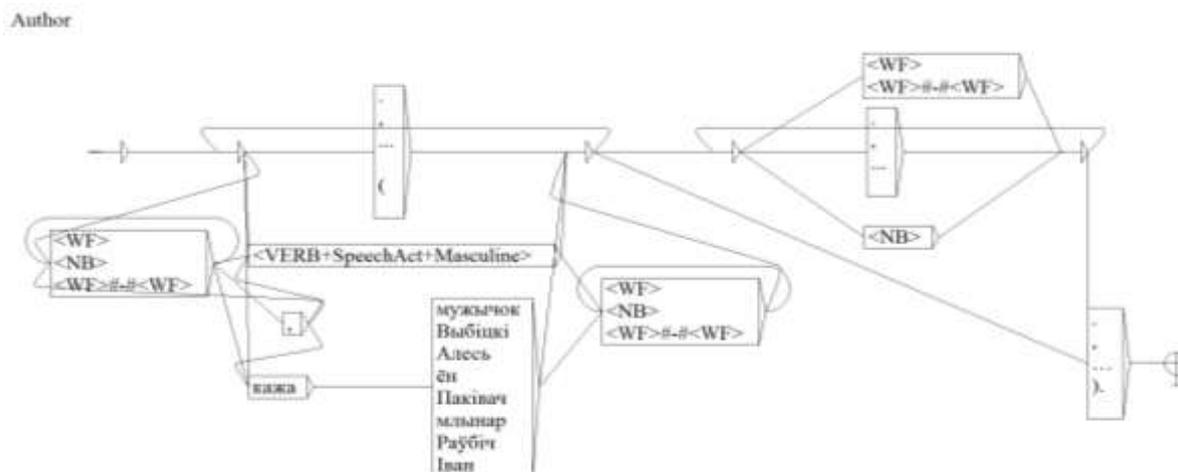
Мал. 3. Падграф Author граматыкі DirectSpeechMasculine

На далейшым этапе па меры папаўнення спісу дзеясловаў – паказчыкаў роду быў створаны асобны слоўнік. У ім парамі прадстаўлены дзеясловы мінулага часу ў формах для жаночага і мужчынскага роду. Для дзеясловаў, якія пачынаюцца з «у», была створана спецыяльная парадыгма ЎVERB1, яна ўлічвае пераход на «ў» пасля галосных (мал. 4).



Мал. 4. Слоўнік дзеясловаў-паказчыкаў

На мал. 5 можна заўважыць, як выкарыстоўваецца атрыманы слоўнік: замест падграфу VERBSmasculine цяпер прымяняюцца спецыяльныя тэгі (катэгорыі) SpeechAct (семантычная пазнака для дзеясловаў – каментарыяў прастай мовы) і Masculine, адпаведна для падграфу VERBSfemenine – тэгі SpeechAct і Feminine.



Мал. 5. Падграф Author граматыкі DirectSpeechMasculine пасля дапаўнення

У ходзе распрацоўкі алгарытмаў таксама была даследавана праблема граматычных амаформаў. Так, дзеяслоўная форма «кажа» (форма цяперашняга часу для дзеяслова казаць) можа адносіцца і да мужчынскага і да жаночага роду, у выніку чаго такія формы не могуць асобна выкарыстоўвацца для ідэнтыфікацыі роду моўцы. Для вырашэння дадзенай задачы была створана дадатковая звязка графаў «дзеяслоў – назоўнік», дзе першы граф уключае граматычныя амаформы дзеясловаў маўлення, другі – спіс назоўнікаў – паказчыкаў роду моўцы (персанажа). Графы адлюстраваны на мал. 5.

Атрыманыя граматыкі могуць прымяняцца паслядоўна да мэтавага тэксту праз NooJ, прычым вынікі пазнак першай граматыкі захоўваюцца другой граматыкай. Такім чынам будзе атрымана разметка тэксту на рэплікі для мужчынскага і жаночага галасоў. Слоўнік дзеясловаў – паказчыкаў роду на дадзены момант змяшчае 432 запісы.

Функцыя праграмы NooJ *Locate Pattern* дазваляе праглядзець знойдзеныя абзацы прастай мовы ў тэксце граматыкай DirectSpeech у выглядзе канкарданса (мал. 6).

Text	Before	Seq.	After
Kalasy_08.not	ў бацькі:	- Быў жа, здаецца, святы з мурынаў? Ці, можа, не?	- Быў, - сказаў
Kalasy_08.not	можа, не?	- Быў, - сказаў бацька. - Здаецца, Хведар-мурын.	- Ну вось
Kalasy_08.not	па сходах.	- Каго яшчэ няма, Georges? - спытала матухна.	- Раўбічаў няма
Kalasy_08.not	спытала матухна.	- Раўбічаў няма. Кроера няма. Старога Вежы няма.	- Добра. Хай
Kalasy_08.not	Вежы няма.	- Добра. Хай ідуць дзеці. - уздыхнула Загорская.	Алесь бег

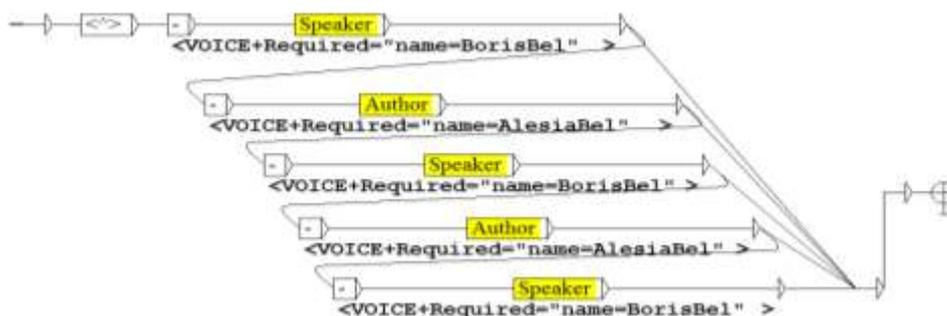
Мал. 6. Простая мова, якую знайшла граматыка DS\_All

Каб працу гэтых граматык можна было выкарыстоўваць у CCMT пад стандарт SAPI 5.1, неабходна прывесці тэксты да выгляду SAPI TTS XML [11]. Таму для выбару неабходнага сінтэзатара маўлення сінтаксічная анатацыя, якую генерыруе граматыка, павінна быць адаптаваная пад наступны код:

```
<VOICE Required="name=[Назва голасу ў сістэме]">
...Тэкст для агучкі...
</VOICE>
```

Для гэтага ў DirectSpeechMasculine і DirectSpeechFemenine былі дададзены маркеры абазначэння шляхоў, якія спрацавалі ў граматыках (мал. 7). Маркеры наладжаны так, што не

пазначаны тэкст твору і словы аўтара чытаюцца голасам AlesiaBel, мужчынскія рэплікі – голасам BorisBel, а жаночыя – ElenaBel.



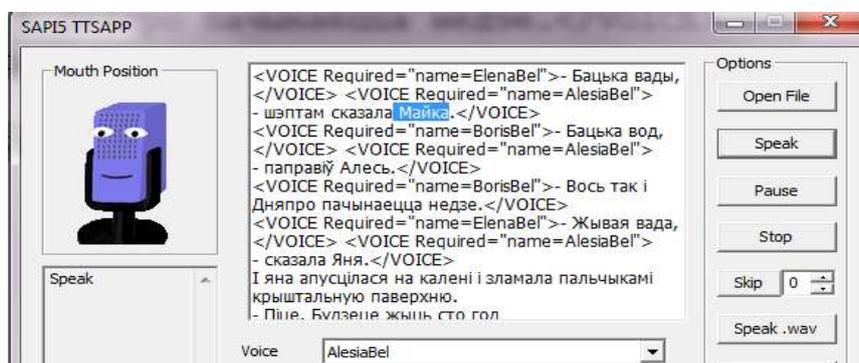
Мал. 7. Прыклад прастай мовы, якую знайшла граматыка DS\_All, з абазначанымі адпаведнымі галасамі для словаў моўцы і аўтара

Для сказаў з табл. 1 афармленне тэксту твору пасля спрацоўвання граматык будзе выглядаць як на прыведзеным ніжэй прыкладзе. Словы аўтара і моўцаў змяшчаюцца ў тэгі VOICE з атрыбутам Required са значэннямі, якія адпавядаюць назвам галасоў сінтэзатараў. Як і планавалася, для слоў аўтара выкарыстоўваецца голас AlesiaBel, а для мужчынскіх і жаночых рэплік моўцаў адпаведна – галасы BorisBel і ElenaBel.

Анатаваны тэкст праз тэгі VoiceXML для аўтаматычнага пераклучэння сінтэзатараў у залежнасці ад роду моўцы і слоў аўтара

```
<VOICE Required="name=ElenaBel">- Бацька вады,</VOICE> <VOICE Required="name=AlesiaBel">
- шэптам сказала Майка.</VOICE>
<VOICE Required="name=BorisBel">- Бацька вод,</VOICE> <VOICE Required="name=AlesiaBel">
- направіў Алясь.</VOICE>
<VOICE Required="name=BorisBel">- Вось так і Дняпро пачынаецца недзе.</VOICE>
<VOICE Required="name=ElenaBel">- Жывая вада,</VOICE> <VOICE Required="name=AlesiaBel">
- сказала Яня.</VOICE>
І яна апусцілася на калені і зламала пальчыкамі крышталную паверхню.
- Піце. Будзеце жыць сто год.
```

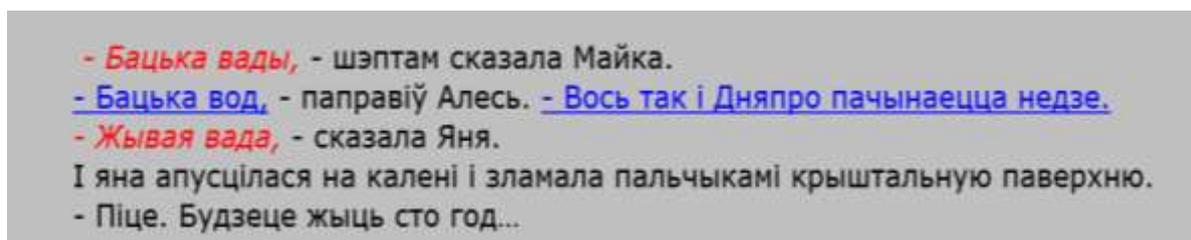
Пасля гэтага такі размечаны тэкст можна падаць на ўваход ССМТ. На мал. 8 паказана, як праграма SAPI5 TTSAPP аўтаматычна пераклучае пастаўленыя ў сістэме галасы AlesiaBel, BorisBel, ElenaBel пры выбранай опцыі Process XML.



Мал. 8. Агучванне тэксту трыма сінтэзатарамі маўлення, які папярэдне быў аўтаматычна размечаны граматыкамі DirectSpeechMasculine і DirectSpeechFemenine

Вынікі працы створаных граматык таксама магчыма выкарыстоўваць і ў мэтах візуальнага абзначэння слоў аўтара і слоў іншых персанажаў у залежнасці ад патрэбнага роду голасу. Гэта можа быць выкарыстана рэдактарам для хуткага прагляду колькасці персанажаў, якія гавораць у творы, каб падабраць неабходную колькасць сінтэзатараў маўлення ці дыктараў.

Для апрацоўкі файлаў VoiceXML была распрацавана спецыяльная праграма VoiceXmlToColorReplacer, якая дазваляе канвертаваць тэгі пазнак неабходнага голасу сінтэзатара ў тэгі HTML з рознымі стылямі выяўлення рэплік (мал. 9).



Мал. 9. Аўтаматычна размечаны тэкст, які зручна агучваць дыктарам

Так, словы аўтара друкуюцца простым шрыфтам чорнага колеру, мужчынскія рэплікі – сінім падкрэсленым, а жаночы голас – чырвоным курсіўным шрыфтам.

#### 4. Ацэнка працы распрацаваных алгарытмаў

У выніку трэніроўкі граматык на 32 раздзелах твору «Каласы пад сярпом тваім» агульная граматыка DS\_All знаходзіць 2574 з 2580 рэплік, з іх 2562 правільныя. Граматыка для вызначэння мужчынскага роду DirectSpeechMasculine знаходзіць 1045 з 1075 рэплік (рэплікі з устаўкамі слоў аўтара) і граматыка для вызначэння жаночага роду DirectSpeechFemenine – 183 з 197. Для больш дэталізаванай ацэнкі выкарыстоўваюцца значэнні дакладнасці ( $P$ ), паўнаты ( $R$ ) і іх сярэдняй гарманічнай велічыні (табл. 2). Пры гэтым  $M$  – правільна знойдзеныя граматыкай рэплікі,  $L$  – усе знойдзеныя граматыкай рэплікі і  $N$  – вызначаныя экспертам правільныя рэплікі ў тэксце.

Табліца 2

Ацэнка працы сінтаксічных граматык NooJ па вызначэнню сказаў з простаю мовай і роду персанажаў па словах аўтара (трэніровачны корпус)

Назвы граматык	Дакладнасць $P = M/L$	Паўната $R = M/N$	Сярэдняя гарманічная велічыня (F1-measure), % $2 * P * R * 100 / (P + R)$
DirectSpeech	$2562/2574 = 0,995$	$2562/2580 = 0,993$	99,4
DirectSpeechMasculine	$1029/1045 = 0,984$	$1029/1075 = 0,957$	97
DirectSpeechFemenine	$181/183 = 0,989$	$181/197 = 0,919$	98,9

У якасці тэставага матэрыялу быў выкарыстаны корпус з мастацкай літаратурай, куды ўвайшлі некаторыя раздзелы твору Якуба Коласа «На ростанях», а таксама «У гарах дажджы» Івана Мележа, «Жалезная кнопка» Людмілы Рублеўскай і «Асеннія лісты» Цёткі. Тэксты падбіраліся метадам выпадковай выбаркі і разам налічваюць 23 867 словаўваходжанняў (першыя 32 раздзелы твору «Каласы пад сярпом тваім» налічваюць 106 217 словаўваходжанняў).

Па падліках эксперта, створаны корпус усяго ўтрымлівае 481 рэпліку (рэплікі з працяжніка), 165 рэплік моўцаў мужчынскага роду і 68 рэплік моўцаў жаночага роду.

Дэталёвыя вынікі тэсціроўкі створаных алгарытмаў на тэставым корпусе адлюстроўваюцца ў табл. 3.

Табліца 3

Ацэнка працы сінтаксічных граматык NooJ па вызначэнню сказаў з прастай мовай і роду персанажаў па словах аўтара (тэставы корпус)

Назвы граматык	Дакладнасць $P = M/L$	Паўната $R = M/N$	Сярэдняя гарманічная велічыня (F1-measure), % $2 * P * R * 100 / (P + R)$
DirectSpeech	461/462 = 0,995	461/481 = 0,958	97,6
DirectSpeechMasculine	143/145 = 0,986	143/165 = 0,866	92,2
DirectSpeechFemenine	57/58 = 0,982	57/68 = 0,838	90,4

### Заклучэнне

Такім чынам, была пастаўлена і вырашана задача па паляпшэнню распрацаваных алгарытмаў і лінгвістычных рэсурсаў для ідэнтыфікацыі прастай мовы ў тэксце і вызначэння роду моўцаў па ўстаўках слоў аўтара з дакладнасцю больш за 90 %. У выніку распрацоўкі слоўніка з паказчыкамі роду атрыманых алгарытмы могуць разглядацца як рэсурсанезалежныя і выкарыстоўвацца для апрацоўкі тэкстаў на іншых славянскіх мовах, неабходна толькі папаўняць адпаведныя слоўнікі. У той жа час распрацаваныя мадэлі паказалі даволі добрыя вынікі ў спалучэнні з ССМТ і ў далейшым могуць быць выкарыстаны ў пабудове дадатковага блоку аўтаматычнага выбару мужчынскага ці жаночага голасу ССМТ. Распрацаваны ў такі спосаб шматгалосы сінтэзатар маўлення па тэксце можа быць добрым прыстасаваннем для хуткага стварэння аўдыёкніг, якія агучаны з захаваннем унікальных асаблівасцей персанажаў твораў.

Нягледзячы на станоўчыя вынікі ацэнкі працы атрыманых алгарытмаў на трэніровачным тэксце, падчас апрацоўкі тэставага корпусу былі высветленыя наступныя праблемы:

- разнастайнасць сімвальных кадыровак выклікае парушэнні ў працэсе ідэнтыфікацыі знакаў прыпынку;

- уніфікацыя алфавітаў (калі літары розных алфавітаў пазначаюцца аднолькавымі сімваламі) не дазваляе граматыкам знаходзіць тыя словы-пазначальнікі, у якіх ёсць устаўкі лацінскіх сімвалаў замест кірылічных;

- спалучэнні сімвалаў, якія пазначаюць пераход ад слоў моўцы да слоў аўтара ці наадварот, часам выкарыстоўваюцца ў дыялагічным тэксце для іншых мэт;

- непаўната слоўнікавых рэсурсаў для вызначэння роду моўцаў памяншае прадукцыйнасць распрацаваных алгарытмаў.

Для вырашэння пералічаных вышэй праблем патрабуецца праца над папаўненнем базы знакаў прыпынку і даданнем слоўнікавых рэсурсаў. У далейшыя планы аўтараў таксама ўваходзіць распрацоўка алгарытмаў для ідэнтыфікацыі роду моўцаў непасрэдна па словах моўцаў і для вызначэння ўзросту моўцаў па словах аўтара і моўцаў.

### Спіс літаратуры

1. Гецэвіч, Ю.С. Аўтаматызаваная апрацоўка сімвальных выразаў у тэкстах для сістэмы сінтэзу беларускага маўлення / Ю.С. Гецэвіч // Інфарматыка. – 2011. – № 4. – С. 82–93.
2. AlchemyAPI Interactive Text Analysis Demo // AlchemyAPI [Electronic resource]. – 2013. – Mode of access : <http://www.alchemyapi.com/api/demo.html>. – Date of access : 23.07.2013.
3. Quotations Extraction // AlchemyAPI [Electronic resource]. – 2013. – Mode of access : <http://www.alchemyapi.com/api/entity/quotations.html>. – Date of access : 23.07.2013.
4. Assignment of Character and Action Types in Folk Tales / P. Lendvai [et al.] // Formalising Natural Languages with NooJ : Selected Papers from the NooJ 2010 Intern. Conf. / eds. Z. Gavriilidou, E. Chatzipapa, L. Papadopoulou, M. Silberzstein. – Greece: Democritus University of Thrace, 2010. – P. 102–111.
5. Jurić, T. Direct Speech Recognition in Text / T. Jurić, M. Stupar, D. Boras // Automatic Processing of Various Levels of Linguistic Phenomena: Selected Papers from the NooJ 2011 Intern.

Conf. / eds. K. Vučković, B. Bekavac, M. Silberztein. – Newcastle : Cambridge Scholars Publishing, 2012. – P. 122–127.

6. Праграма камп'ютарнага запісу аўдыёкніг [Электронны рэсурс]. – Рэжым доступу : <http://mp3book2005.ru/1.htm>. – Дата доступу : 25.10.2013

7. Праграма AUDIOBOOK [Электронны рэсурс]. – 2013. – Рэжым доступу : [http://kompas.narod.ru/audiobook\\_net.htm](http://kompas.narod.ru/audiobook_net.htm). – Дата доступу : 25.10.2013

8. Гецэвіч, Ю.С. Аўтаматызацыя шматгаласавога стварэння аўдыёкніг на беларускай мове з дапамогай сінтэзатару маўлення па тэксце / Ю.С. Гецэвіч, Т.І. Окрут, Б.М. Лабанаў // Развитие информатизации и государственной системы научно-технической информации (РИНТИ-2013) : доклады XII Междунар. конф. (Минск, 20 ноября 2013 г.). – Минск : ОИПИ НАН Беларуси, 2013. – С. 60–67.

9. Hetsevich, Y. Belarusian Module for NooJ / Y. Hetsevich, S. Hetsevich, Ya. Yakubovich // NooJ web-site [Electronic resource]. – 2012. – Mode of access : <http://www.nooj4nlp.net/pages/belarusian.html>. – Date of access : 16.03.2012.

10. Лінгвістычны працэсар NooJ [Электронны рэсурс]. – 2002. – Рэжым доступу : <http://www.nooj4nlp.net/pages/nooj.html>. – Дата доступу : 01.07.2013.

11. XML TTS Tutorial (SAPI 5.3) // Microsoft Developer Network [Electronic resource]. – 2013. – Mode of access : <http://msdn.microsoft.com/en-us/library/ms717077%28v=vs.85%29.aspx>. – Date of access : 29.07.2013.

**Паступіла 12.11.2013**

*Аб'яднаны інстытут праблем  
інфарматыкі НАН Беларусі,  
Мінск, Сурганава, 6  
e-mail: yury.hetsevich@gmail.com,  
tatberrie@gmail.com,  
lobanov@newman.bas-net.by,*

**Y.S. Hetsevich, T.I. Okrut, B.M. Lobanov**

### **ALGORITHMS FOR IDENTIFICATION OF CUES WITH AUTHORS' TEXT INSERTIONS IN BELARUSIAN ELECTRONIC BOOKS**

The main stages of algorithms for characters' gender identification in Belarusian electronic texts are described. The algorithms are based on punctuation marking and gender indicators detection, such as past tense verbs and nouns with gender attributes. For indicators, special dictionaries are developed, thus making the algorithms more language-independent and allowing to create dictionaries for cognate languages. Testing showed the following results: the mean harmonic quantity for masculine gender detection makes up 92,2 %, and for feminine gender detection – 90,4%.

УДК 004

Б.А. Залесский, Э.Н. Середин

## РЕАЛИЗАЦИЯ АЛГОРИТМОВ ОТСЛЕЖИВАНИЯ ОБЪЕКТОВ НА ОСНОВЕ ТЕХНОЛОГИИ CUDA

*Предлагаются быстрые версии корреляционных алгоритмов отслеживания объектов на видеопоследовательностях, снятых нестабилизированной камерой. Алгоритмы предназначены для реализации на CUDA. Они основываются на сравнении корреляции области изображения, содержащей объект интереса, с соответствующими ей по размеру областями текущего кадра видеопоследовательности вместе с анализом межкадровой информации. Применение технологии программирования видеокарты CUDA позволило добиться выполнения алгоритмов в режиме реального времени. Для повышения точности и устойчивости предложенных версий алгоритмов используется робастная версия фильтра Калмана. Показывается работоспособность предложенных версий даже в случае видеопоследовательностей, снятых нестабилизированной камерой.*

### Введение

За последние годы алгоритмы отслеживания объектов получили широкое применение в различных сферах деятельности общества, включая охрану окружающей среды, картографию, навигацию, безопасность движения и т. д.

Для отслеживания объектов на видеопоследовательностях, снятых в разных условиях, используются различные алгоритмы, поэтому рассмотрим случай, когда видеопоследовательность снята движущейся нестабилизированной камерой, например, установленной на борту летательного аппарата. Задача состоит в том, чтобы отследить в реальном времени объект на кадрах видеопоследовательности. Сложность данного случая заключается в невозможности предсказать заранее положение объекта (и, следовательно, ограничить его область поиска) на текущем кадре, а также в том, что обычно при такой съемке на кадрах присутствуют шумы разных типов.

В большом числе статей по данной тематике содержится множество разнообразных алгоритмов, предназначенных для решения сформулированной задачи. Среди них алгоритмы, основанные на сравнении гистограмм яркостей и ориентированных градиентов, оптических потоков, корреляций, текстур, активных контуров и др. [1–6]. Одна из главных трудностей, общая для всех известных подходов, – малое время, имеющееся для оценки положения объекта на текущем кадре, обычно ограниченное 25–40 мс. Некоторые надежные и точные алгоритмы невозможно выполнить на компьютере за столь короткое время. До недавнего времени это касалось и корреляционных алгоритмов, достаточно надежных и точных, но, к сожалению, требующих вычисления большого объема операций. Непосредственное оценивание координат объекта, отслеживаемого на видеопоследовательности стандартного размера с помощью корреляции Пирсона (при поиске по всему кадру), требует нескольких минут вычислений на современном процессоре. Даже использование быстрого преобразования Фурье и SSE-инструкций для подсчета корреляции не позволяет достичь реального времени сопровождения.

Однако ситуация изменилась с появлением технологии программирования CUDA, позволяющей производить параллельные вычисления на видеокартах, выпускаемых компанией Nvidia. Новые архитектуры технологии CUDA, такие как Fermi и Kepler, позволяют выполнить корреляционные алгоритмы отслеживания объектов движущейся нестабилизированной видеокамерой в режиме реального времени на видеокартах GeForce шестой серии и выше.

В статье описаны такого вида реализации вместе со сравнением корреляционного подхода к решению задачи отслеживания с некоторыми известными подходами. Представлено также описание робастной версии фильтра Калмана, использованной для повышения точности и стабильности трекинга.

### 1. Краткое описание алгоритма отслеживания

Первоначально для решения задачи отслеживания объектов на видеопоследовательностях, снятых движущейся нестабилизированной видеокамерой, были реализованы и протестированы на реальных данных (рис. 1) несколько известных алгоритмов, позволяющих решать задачу в режиме реального времени, и неоптимизированные корреляционные алгоритмы в тестовом режиме. Корреляционные алгоритмы оказались более точными и надежными. Для повышения надежности и точности сопровождения была использована робастная версия фильтра Калмана.



Рис. 1. Пример кадра видеопоследовательности, полученной с борта самолета

Обозначим через  $\mathbf{I}$  –  $m \times n$  полутоновое или RGB-изображение с пикселями  $p = (x, y)$  и яркостями  $I_p$  или цветом  $(I_{R,p}, I_{G,p}, I_{B,p})$ . Кадр видеопоследовательности в момент времени  $t \in \{0, 1, \dots\}$  обозначим  $\mathbf{I}_t$ .

Процесс сопровождения начинается с выделения объекта интереса. Пусть  $\mathbf{O}_0(p_{obj})$  –  $k \times k$ -часть начального кадра  $\mathbf{I}_0$ , содержащая отслеживаемый объект, с центром в пикселе  $p_{obj}$ . Пусть  $\mathbf{O}_t(p, \alpha)$  – часть текущего кадра  $\mathbf{I}_t$ , повернутого на угол  $\alpha$ . Она представляет собой  $k \times k$ -квадрат со сторонами, параллельными сторонам неповернутого кадра, и центром  $p$  относительно исходного неповернутого изображения.

*Замечание.* В данной версии алгоритма не описывается применение масштабной пирамиды, позволяющей вести слежение за объектом при изменении высоты видеокамеры, так как проведенные эксперименты показали, что в рассматриваемом случае в каждый момент времени достаточно использовать только три уровня масштабной пирамиды: актуальный уровень предыдущего кадра и ближайшие к нему верхний и нижние уровни.

Два типа корреляции были использованы для оценки положения объекта:

$$r_1(p, \alpha) = r_1(\mathbf{O}_0, \mathbf{O}_t(p, \alpha)) = \frac{1}{k^2} \sum_{x,y=0}^k O_{0,x,y} O_{t,x,y}(p, \alpha) - \bar{O}_0 \bar{O}_t(p, \alpha),$$

где  $\bar{O}_0$  и  $\bar{O}_t(p, \alpha)$  – средние значения изображений  $\mathbf{O}_0$  и  $\mathbf{O}_t(p, \alpha)$ , и

$$r_2(p, \alpha) = r_2(\mathbf{O}_0, \mathbf{O}_t(p, \alpha)) = \frac{r_1(p, \alpha)}{SSQ(\mathbf{O}_0)SSQ(\mathbf{O}_t(p, \alpha))},$$

где  $SSQ(\mathbf{I})$  – стандартное отклонение яркостей (или цвета) изображения  $\mathbf{I}$ .

Оценки положения  $p_t$  отслеживаемого объекта в момент времени  $t$  вычисляются как

$$\hat{p}_{1,t} = \arg \max_{p, \alpha} r_1(p, \alpha) \text{ и } \hat{p}_{2,t} = \arg \max_{p, \alpha} r_2(p, \alpha).$$

Максимум берется по всем пикселям изображения  $\mathbf{I}$  и всем допустимым углам  $\alpha$ .

Данные оценки хорошо известны не только в обработке изображений, поэтому их использование не является новшеством. Новизна заключается, во-первых, в быстрой программной реализации оценок  $\hat{p}_{1,t}$ ,  $\hat{p}_{2,t}$ , основанной на применении технологии программирования видеокарты CUDA, которая сделала возможным выполнение алгоритма в режиме реального времени, и, во-вторых, в использовании робастной версии фильтра Калмана, которая сделала процесс отслеживания более устойчивым и точным. Ключевые моменты аппаратной оптимизации предложенного подхода приведены в следующем разделе.

Опишем кратко применяемую робастную версию фильтра Калмана. Традиционно схема линейного фильтра Калмана без управления задается следующими уравнениями:

$$\begin{aligned} \hat{x}_t^- &= F_{k-1} \hat{x}_{t-1}^+; \\ \hat{x}_t^+ &= \hat{x}_t^- + K_t (y_t - H_t \hat{x}_t^-), \end{aligned} \quad (1)$$

где  $\hat{x}_t^-$  – предсказание (априорная оценка);  $\hat{x}_t^+$  – апостериорная оценка;  $y_t$  – входной зашумленный сигнал. Назначение и вид матриц  $F_{k-1}$ ,  $K_t$  и  $H_t$  приведены в деталях в [7].

При использовании классической версии фильтра Калмана для сглаживания и стабилизации оценок  $\hat{p}_{1,t}$ ,  $\hat{p}_{2,t}$  возникают грубые погрешности в случаях, когда оценка  $\hat{p}_{i,t}$  указывает на ошибочное положение объекта на текущем кадре  $\mathbf{I}_t$ . Погрешности можно уменьшить следующим образом: для квадрата пикселей  $S(\hat{x}_{t-1})$  с центром в  $\hat{x}_{t-1}$  заранее фиксированного размера проверить условие попадания  $\hat{p}_{i,t}$  в  $S(\hat{x}_{t-1})$  и выбрать в качестве входного значения

$$y_t = \begin{cases} \hat{p}_{i,t}, & \text{если } \hat{p}_{i,t} \in S(\hat{x}_{t-1}); \\ \hat{x}_t^-, & \text{в противном случае.} \end{cases}$$

Кроме того, для заранее выбранного целого числа  $N_{out}$  следует вычислить, сколько последовательных оценок  $\hat{p}_{i,t}$  оказалось вне  $S(\hat{x}_t^-)$ . Как только число таких критических оценок станет больше чем  $N_{out}$ , следует инициализировать новый фильтр Калмана следующим образом:

- 1) в качестве решения положить  $\hat{x}_t^+ = \hat{p}_{i,t}$ ;
- 2) для некоторого заранее выбранного целого числа  $N_{in}$  проверить, превосходит ли количество последовательных оценок  $\hat{p}_{i,t+k}$ , оказавшихся в квадрате  $S(\hat{x}_t^+)$ , число  $N_{in}$ . Если превосходит, использовать описанную выше робастную версию фильтра Калмана, иначе повторить п. 1.

Общая схема работы предложенного алгоритма отслеживания объектов на видеопоследовательностях, снятых нестабилизированной видеокамерой, показана на рис. 2.

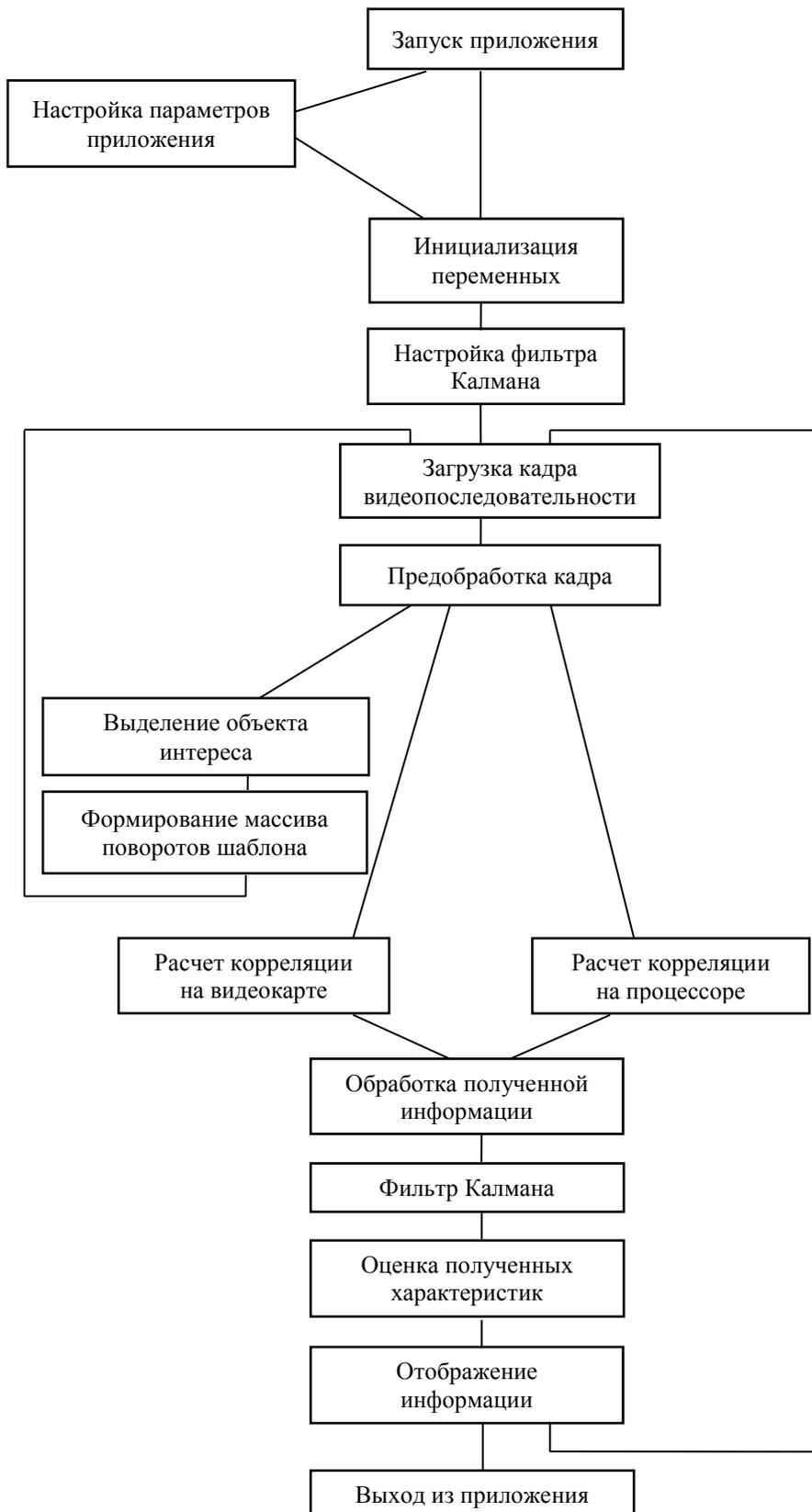


Рис. 2. Общая схема работы приложения по отслеживанию объектов

## 2. Особенности реализации алгоритма

Корреляционные алгоритмы требуют большого числа вычислений, поэтому их применение в задачах отслеживания объектов до недавнего времени было ограничено из-за невозможности выполнения в режиме реального времени. Для преодоления этого недостатка была использована технология программирования CUDA, которая обеспечила необходимый прирост производительности вычислений за счет параллельной обработки данных.

Реализация на CUDA корреляционных алгоритмов, предназначенных для отслеживания объектов на видеопоследовательностях в режиме реального времени, – нетривиальная задача. Для ее решения первоначально был формализован в терминах технологии CUDA последовательный алгоритм вычисления большого числа корреляций, необходимых для отслеживания объекта на текущем кадре. Затем на каждом шаге построения параллельной версии алгоритма выделялись его критические блоки, вычисление которых занимало большую часть времени. Производилось распараллеливание и оптимизация выполнения выделенных блоков до уровня, позволяющего устранить их критичность для достижения режима реального времени. После получения версии алгоритма, допускающей выполнение в режиме реального времени, были распараллелены оставшиеся блоки алгоритма для достижения максимального быстродействия построенной версии.

Практическая реализация алгоритма показала, что наилучшее быстродействие достигается при использовании 2D-текстур и разделяемой памяти. Одной из особенностей работы с текстурами является то, что на видеокarte они доступны в режиме для чтения, а запись в них может производить только процессор.

Эффективность использования текстур проявляется в том, что работа с данными в них происходит через текстурные выборки, содержание которых кешируется. Использование разделяемой памяти объясняется тем, что она является самой быстрой памятью на видеокarte наравне с регистрами и может использоваться для хранения промежуточных вычислений. Расчет на видеокarte всей совокупности корреляций, необходимой для отслеживания объекта интереса на текущем кадре, состоит из следующих этапов:

- инициализации данных, необходимых для запуска ядра на видеокarte;
- заполнения текстуры **source** данными из кадра видеопоследовательности, на котором будет происходить поиск объекта;
- заполнения **n** текстур **template** данными из массива, в котором находятся **n** вариантов поворота объекта поиска;
- выделения памяти на видеокarte под массив выходных значений;
- запуска ядра на видеокarte для выполнения вычислений с использованием специально разработанной топологии взаимодействия блоков и нитей (определение этих терминов дается в [8, 9]), в которой каждому пикселу текстуры **source** ставится в соответствие отдельный блок. Количество нитей для блока выбирается исходя из размеров текстуры **template** для обеспечения наиболее быстрого прохода по ячейкам текстуры;
- создания переменных для индексации внутри видеокарты;
- проверки на наличие выхода переменных за пределы допустимых диапазонов значений;
- запуска **n** раз для каждого пиксела текстуры **source** функции **f()**, вычисляющей корреляцию между областью изображения с центром в данном пикселе, размер которой соответствует размеру окна, выделенного оператором на исходном кадре, и каждой текстурой **template**, где **n** – количество заданных поворотов объекта поиска (рис. 3);
- занесения результатов в выходной массив, находящийся в глобальной памяти видеокарты, с учетом предложенной авторами индексации внутри видеокарты, что позволяет исключить конфликты доступа к ячейкам данного массива;
- копирования выходного массива, в котором содержатся значения корреляции для каждого пиксела текущего кадра видеопоследовательности и для каждого заданного угла поворота, из памяти видеокарты в оперативную память.

Быстрая реализация функции **f()** может быть представлена в виде следующих последовательно выполняющихся шагов:

- считывания нитями блока в параллельном режиме данных из текстуры **source** и текстуры **template** в регистры видеокарты для выполнения арифметических операций, необходимых для вычисления корреляций;
- выполнения в параллельном режиме операций, необходимых для вычисления скалярных произведений и средних значений для расчета корреляций  $r_1(p, \alpha)$  и  $r_2(p, \alpha)$ , с использованием специализированных математических функций технологии CUDA;
- занесения результатов вычислений в разделяемую память;
- использования при больших размерах текстуры **template** для ускорения вычислений приема работы с разделяемой памятью, называемого редукцией [9];
- непосредственного вычисления корреляций  $r_1(p, \alpha)$  и  $r_2(p, \alpha)$ ;
- возвращения из функции значения рассчитанных корреляций  $r_1(p, \alpha)$  и  $r_2(p, \alpha)$ .

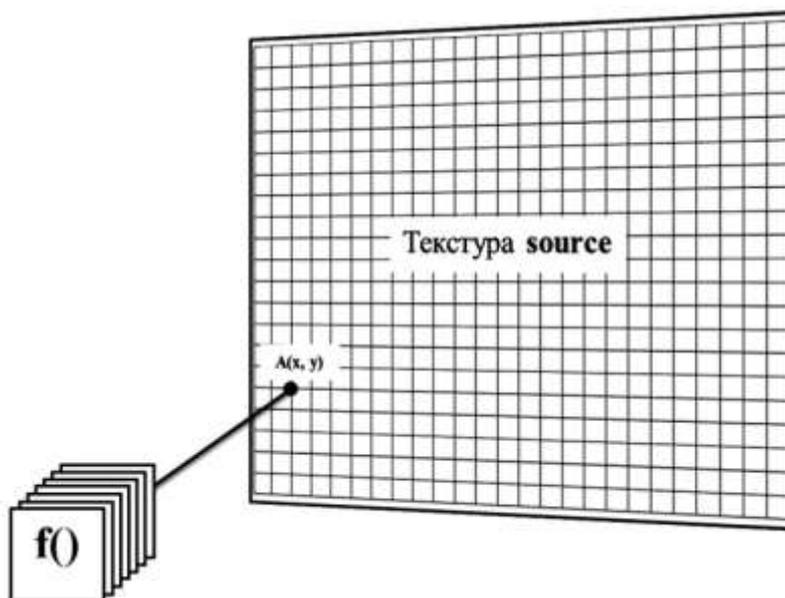


Рис. 3. Пример расчета корреляции на видеокарте в точке **A** с координатами  $(x, y)$

Укажем некоторые особенности программной реализации алгоритма. Функция **f()**, вычисляющая корреляции  $r_1(p, \alpha)$  и  $r_2(p, \alpha)$ , написана на расширенном языке CUDA C без использования сторонних библиотек и является платформенно-независимой. Для операционных систем Windows созданы отдельные 32- и 64-битные библиотеки в виде файлов \*.dll, которые могут быть использованы в любых языках программирования.

Размер текстуры **source** может быть произвольным, а размер текстуры **template** желательно, чтобы был кратным двойке (8x8, 16x16, 32x32, 64x64, 128x128). При заполнении текстур **template** используются заранее подготовленные таблицы соответствия пикселей исходной и повернутой текстур, позволяющие поворачивать шаблон на произвольный угол без каких-либо расчетов и интерполяции.

В описываемом алгоритме не используется масштабирование изображений при поиске объекта. В замечании уже упоминалось, что добавление двух дополнительных уровней масштабной пирамиды – верхнего и нижнего – и расчет корреляции на них не вносят принципиальных трудностей, так как дополнительные вычисления на видеокарте могут проводиться параллельно.

### 3. Результаты тестирования алгоритма

Тестирование алгоритма проводилось на реальных, снятых с борта летательного аппарата, и модельных видеопоследовательностях. Результаты экспериментов приведены в табл. 1.

Приблизительно 90–95 % времени обработки одного кадра видеопоследовательности тратится на расчеты, которые производятся на видеокарте, а остальная часть – на загрузку изображения, предварительную обработку и т. д.

Таблица 1

Время обработки одного кадра видеопоследовательности, мс (шаблон 8x8 пикселей)

Устройство	Разрешение изображения, пикселей						
	256x256	640x480	768x576	1024x768	1440x1080	1920x1080	4000x4000
GPU GTX 260	14	44	58	106	205	271	2069
GPU GTX 550Ti	9	35	46	86	169	225	1730
GPU GTX 650Ti	7	22	29	53	104	138	1049

В табл. 1 время расчетов одного кадра видеопоследовательности указано для коэффициента масштабирования, равного 1, хотя в разработанном алгоритме применяются и другие коэффициенты масштабирования – 4, 16. Как показало тестирование, масштабирование не сказывается на точности работы алгоритма. При масштабировании одного кадра видеопоследовательности время расчетов уменьшается кратно коэффициенту сжатия, что позволяет использовать видеопоследовательности с более высокими разрешениями. Еще одной особенностью реализации алгоритма с помощью технологии CUDA является то, что при увеличении размера шаблона время работы алгоритма растет незначительно. Например, при увеличении размера шаблона в 16 раз время обработки одного кадра возрастает на 20–25 % в отличие от процессорной версии алгоритма, время работы которой возрастает в 16 раз.

Для сравнения точности разработанного алгоритма выбраны другие алгоритмы поиска объекта интереса на кадрах видеопоследовательности, работающие в режиме реального времени и основанные на сравнении гистограмм ориентированного градиента [2, 10] и текстурных признаков областей изображений [1]. В табл. 2 приведены проценты правильного нахождения ранее заданного объекта интереса на кадрах видеопоследовательности плохого качества. Под правильным понимается нахождение такой области изображения, которая содержит не менее 50 % площади искомого объекта интереса.

Таблица 2

Процент нахождения объекта интереса на кадрах видеопоследовательности

Алгоритм	Точность, %
Гистограмма ориентированного градиента	38
Текстура по трем точкам	60
Текстура по двум точкам	74
Описываемый в статье алгоритм	97

Из табл. 2 видно, что корреляционный алгоритм превосходит по точности другие протестированные алгоритмы на видеопоследовательностях низкого качества (пример кадра такой видеопоследовательности приведен на рис. 1).

### Заключение

Представлено описание корреляционного алгоритма отслеживания объектов на кадрах видеопоследовательностей, полученных нестабилизированной камерой, и его быстрой программной реализации на CUDA, позволяющей проводить вычисления в режиме реального времени.

Приведены результаты сравнения на тестовых и реальных видеопоследовательностях точностей предложенного алгоритма с алгоритмами, основанными на сравнении гистограмм ориентированного градиента и текстурных признаков областей изображений. Точность предложенного корреляционного алгоритма оказалась выше, что соответствует результатам, полученным другими авторами.

Созданы 32- и 64-битные библиотеки для операционной системы Windows по расчету корреляций  $r_1(p, \alpha)$  и  $r_2(p, \alpha)$ , реализованные с использованием технологии CUDA без каких-либо дополнительных сторонних библиотек.

В будущем планируется усовершенствовать алгоритм, включив в него шаги, связанные с быстрой сегментацией кадров видеопоследовательностей.

### Список литературы

1. Yilmaz, A. Object tracking: A survey / A. Yilmaz, O. Javed, M. Shah // ACM Computing Surveys. – 2006. – Vol. 38, № 4.
2. Marimon, D. Orientation histogram-based matching for region tracking / D. Marimon, T. Ebrahimi // Proc. 8th Int. Workshop on Image Analysis for Multimedia Interactive Services WIAMIS. – Santorini, 2007. – P. 8–12.
3. Lowe, D. Object recognition from local scale invariant features / D. Lowe // Proc. Int. Conf. on Computer Vision ICCV. – Corfu, 1999. – P. 1150–1157.
4. Bay, H. Surf: Speeded up robust features / H. Bay, T. Tuytelaars, L. Van Gool // Proc. 9<sup>th</sup> Europ. Proc. 8th Int. on Computer Vision ECCV. – Graz, 2006. – P. 404–417.
5. Altmann, J. A Fast Correlation Method for Scale-and Translation-Invariant Pattern Recognition / J. Altmann, H.J. Reitböck // IEEE Trans. Pattern Anal. Mach. Intell. – 1984. – Vol. 6, № 1. – P. 46–57.
6. Object Tracking by Particle Filtering Techniques in Video Sequences / L. Mihaylova [et al.] // Advances and Challenges in Multisensor Data and Information. NATO Security Through Science Series. – Netherlands : IOS Press, 2007. – P. 260–268.
7. Simon. D. Optimal State Estimation. Kalman, H $\infty$ , and Nonlinear Approaches / D. Simon. – New Jersey : John Wiley & Sons, 2006. – 526 p.
8. Сандерс, Дж. Технология CUDA в примерах: введение в программирование графических процессоров / Дж. Сандерс, Э. Кэндрот. – М. : ДМК Пресс, 2011. – 232 с.
9. Боресков, А.В. Основы работы с технологией CUDA / А.В. Боресков, А.А. Харламов. – М. : ДМК Пресс, 2010. – 232 с.
10. Zalesky, B.A. Scale invariant algorithm to match regions on aero and satellite images / B.A. Zalesky, P.V. Lukashovich // Proc. 11<sup>th</sup> Int. Conf. on Pattern recognition and information processing PRIP 2011. – Minsk, 2011. – P. 25–31.

Поступила 3.12.2013

*Объединенный институт проблем  
информатики НАН Беларуси,  
Минск, Сурганова, 6  
e-mail: eduard.seredin@tut.by*

**B.A. Zalesky, E.N. Seredin**

### **IMPLEMENTATION OF OBJECT TRACKING ALGORITHMS ON THE BASIS OF CUDA TECHNOLOGY**

A fast version of correlation algorithm to track objects on video-sequences made by a non-stabilized camcorder is presented. The algorithm is based on comparison of local correlations of the object image and regions of video-frames. The algorithm is implemented in programming technology CUDA. Application of CUDA allowed to attain real time execution of the algorithm. To improve its precision and stability, a robust version of the Kalman filter has been incorporated into the flowchart. Tests showed applicability of the algorithm to practical object tracking.

УДК 621.391

А.А. Журавлёв, В.Ю. Цветков

## МАСОЧНО-ФАЗОВЫЙ МЕТОД ЛОКАЛИЗАЦИИ ПРЯМЫХ ЛИНИЙ НА ИЗОБРАЖЕНИИ

*Предлагается масочно-фазовый метод локализации прямых линий на изображении, основанный на предварительном грубом квантовании отрезков по ориентации с использованием библиотеки бинарных масок и уточнении прямых линий с помощью ориентированного преобразования Хафа. Метод позволяет повысить скорость и качество локализации прямых линий в условиях изменения яркости, контраста и зашумления изображения.*

### Введение

Для решения задач совместной обработки нескольких изображений (наложения, сравнения, построения панорам и т. д.) необходима локализация реперных элементов. Наиболее эффективные методы локализации реперов, такие как SIFT [1] и SURF [2], ориентированы на использование характерных точек. Это позволяет снизить вычислительную сложность по сравнению с корреляционными методами [3], однако приводит к низкой устойчивости результатов локализации реперов в условиях изменения яркости, контраста и зашумления изображения. Прямые линии обладают значительно большей устойчивостью по сравнению с точками. Для их локализации на изображениях широко используются методы, основанные на преобразовании Хафа [4], масочном поиске [5], вычислении градиента [6] и квантовании по ориентации [7]. Однако эти методы требуют значительных вычислительных ресурсов, имеют низкое быстродействие (преобразование Хафа, масочный поиск) и не обеспечивают качественную локализацию прямых, характеризуемую вероятностью ложной локализации и устойчивостью к изменению яркости, контраста и зашумлению изображения (вычисление градиента, квантование по ориентации). Устранение данных недостатков возможно за счет двухэтапной обработки. Сначала осуществляется предварительное грубое квантование отрезков по ориентации с использованием малоразмерных масок. Затем производится уточнение ориентации этих отрезков и их объединение в прямые в фазовом пространстве пространственно-ориентированного преобразования Хафа, реализуемого в ограниченном секторе, который определяется результатами предварительного квантования отрезков по ориентации.

В настоящей статье разработан метод вычислительно простой и качественной локализации прямых линий на изображении с использованием масок и пространственно-ориентированного преобразования Хафа в условиях изменения яркости, контраста и зашумления изображения.

### 1. Локализация прямых линий в фазовом пространстве преобразования Хафа

Преобразование Хафа позволяет находить объекты, принадлежащие определенному классу фигур, в результате применения процедуры голосования к фазовому пространству параметров. Через точку двумерного пространства может проходить бесконечное количество прямых. Если точка имеет координаты  $(x_0, y_0)$ , то все прямые, проходящие через нее, описываются уравнением  $\rho(\theta) = x_0 \cos \theta + y_0 \sin \theta$ , где  $\rho(\theta)$  – длина радиуса-вектора ближайшей к началу координат точки на прямой, а  $\theta$  – угол между радиусом-вектором и осью ординат. Это соответствует синусоиде в фазовом пространстве  $(\rho, \theta)$  преобразования Хафа, которая уникальна для данной точки. Ряд точек, принадлежащих прямой линии, определяет синусоиды, которые пересекаются в одной точке фазового пространства. Это позволяет свести проблему обнаружения прямых линий к обнаружению пересекающихся синусоид. Вычислительная сложность локализации прямых линий на изображении с использованием преобразования Хафа определяется количеством прямых линий, формируемых для каждого контурного пиксела изображения. Она стремительно растет с

повышением точности определения направления линий и уменьшением порога контурной обработки. Это делает невозможным использование преобразования Хафа в большинстве практических задач, требующих локализации прямых линий в реальном масштабе времени.

## 2. Масочно-фазовая локализация прямых линий на изображении

Предлагается масочно-фазовый метод локализации прямых линий на изображении, основанный на библиотеке бинарных масок и пространственно-ориентированном преобразовании Хафа. Сущность метода состоит в использовании библиотеки из четырех бинарных масок размером  $3 \times 3$  пиксела для предварительного грубого квантования по ориентации контурных элементов изображения длиной в три пиксела и последующем уточнении их принадлежности прямым линиям с помощью пространственно-ориентированного преобразования Хафа, применяемого к каждому квантованному контурному элементу в ограниченном секторе, определяемом ориентацией этого элемента. Каждая бинарная маска состоит из трех единичных пикселей, образующих прямолинейный отрезок с угловой ориентацией  $\alpha = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$  (рис. 1). Метод обеспечивает повышение устойчивости и снижение вычислительной сложности локализации прямых линий на изображении по сравнению с методом Хафа за счет предварительной фильтрации прямолинейных отрезков при квантовании контурных элементов изображения по ориентации и уменьшения сектора анализа при уточнении ориентации контурных элементов в фазовом пространстве преобразования Хафа.

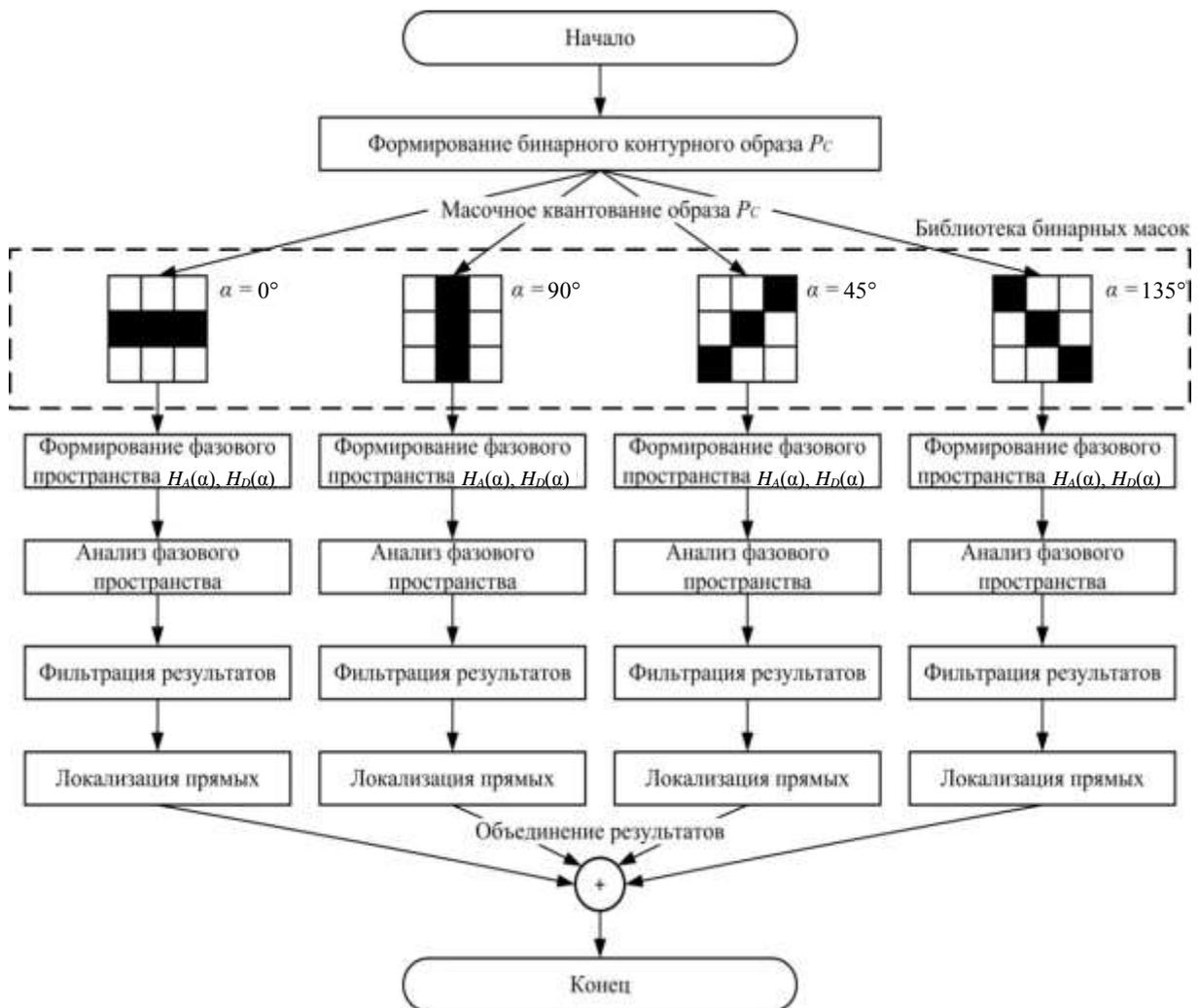


Рис. 1. Блок-схема масочно-фазового метода локализации прямых линий на изображении

Алгоритм, реализующий предлагаемый метод, состоит из следующих шагов (см. рис. 1):

1. Формирование бинарного контурного образа исходного изображения (рис. 2). В результате контурной обработки исходного изображения размером  $Y \times X$  пикселей формируется бинарный контурный образ  $P_C = \|p_C(y, x)\|_{(y=0, \overline{Y-1}; x=0, \overline{X-1})}$ , где  $p_C(y, x)=1$  для контурных пикселей,  $p_C(y, x)=0$  для неконтурных пикселей. Наилучшие результаты локализации прямых получаются при использовании для выделения контуров алгоритма Canny.

2. Предварительное масочное квантование элементов бинарного контурного образа  $P_C$  по ориентации. Окрестность  $3 \times 3$  пиксела каждого элемента  $p_C(y, x)=1$  бинарного контурного образа  $P_C$  проверяется на соответствие четырем контурным маскам. Число масок определяется исходя из их размера и ограничений на возможные варианты расположения в них контурных пикселей. Для маски размером  $3 \times 3$  пиксела существует только четыре варианта размещения трех контурных пикселей в прямую линию, проходящую через центр маски. При предварительном квантовании отрезков по ориентации это обеспечивает относительно малую вычислительную сложность (повышается с увеличением размера маски) и низкую вероятность ошибочной локализации прямых линий (повышается с уменьшением размера маски). Если обнаружено соответствие окрестности элемента  $p_C(y, x)=1$  бинарной маске с угловой ориентацией  $\alpha$ , элементу  $c(y, x, \alpha)$  бинарного образа  $C(\alpha) = \|p_C(y, x, \alpha)\|_{(y=0, \overline{Y-1}; x=0, \overline{X-1})}$  присваивается значение 1. В результате данного шага формируются четыре бинарных контурных образа  $\{C(\alpha)\}_{(\alpha=\{0^\circ, 45^\circ, 90^\circ, 135^\circ\})}$ , соответствующих четырем угловым ориентациям бинарных масок (см. рис. 1). При инициализации алгоритма образы  $\{C(\alpha)\}_{(\alpha=\{0^\circ, 45^\circ, 90^\circ, 135^\circ\})}$  обнуляются.

3. Пространственно-ориентированное преобразование Хафа над бинарными образами  $\{C(\alpha)\}_{(\alpha=\{0^\circ, 45^\circ, 90^\circ, 135^\circ\})}$ . Для каждого элемента  $c(y, x, \alpha)=1$  каждого бинарного образа  $\{C(\alpha)\}_{(\alpha=\{0^\circ, 45^\circ, 90^\circ, 135^\circ\})}$  с помощью преобразования  $T_H$  Хафа вычисляются параметры Хафа  $\{h_A(y, x, n_V, \alpha), h_D(y, x, n_V, \alpha)\}_{(n_V=0, \overline{N_V-1})}$  с помощью выражения

$$\forall c(y, x, \alpha) \left( (c(y, x, \alpha) = 1) \rightarrow \left( \{h_A(y, x, n_V, \alpha), h_D(y, x, n_V, \alpha)\}_{(n_V=0, \overline{N_V-1})} \Leftarrow T_H(y, x) \right) \right), \quad (1)$$

$$y = \overline{0, Y-1}, \quad x = \overline{0, X-1}, \quad \alpha = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\},$$

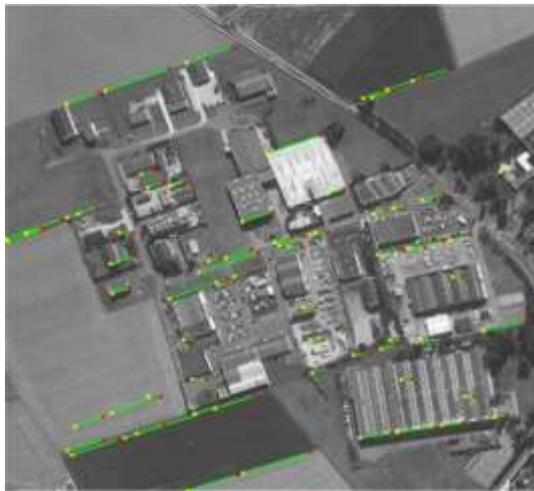
где  $h_A(y, x, n_V, \alpha)$  – угол между осью абсцисс и перпендикуляром к виртуальной прямой с индексом  $n_V$ , проходящей через элемент  $c(y, x, \alpha)$ ;  $h_D(y, x, n_V, \alpha)$  – длина этого перпендикуляра;  $N_V = 2\Delta_K / \Delta_V$  – число виртуальных прямых, проходящих через элемент  $c(y, x, \alpha)$  под углами  $h \in [\alpha - \Delta_K, \alpha + \Delta_K]$  с угловой разностью  $\Delta_V$ ;  $2\Delta_K$  – диапазон применения пространственно-ориентированного преобразования Хафа. Угловая разность  $\Delta_V$  определяет качество локализуемых прямых. Диапазон  $2\Delta_K$  определяется числом ориентаций бинарных масок и определяется с помощью выражения  $2\Delta_K = 180/4 = 45^\circ$  (для обеспечения перекрытия секторов  $\Delta_K = 25^\circ$  установлено экспериментально). В результате данного шага формируются матрицы  $H_A(\alpha) = \left\| \{h_A(y, x, n_V, \alpha)\}_{(n_V=0, \overline{N_V-1})} \right\|_{(y=0, \overline{Y-1}; x=0, \overline{X-1})}$  и  $H_D(\alpha) = \left\| \{h_D(y, x, n_V, \alpha)\}_{(n_V=0, \overline{N_V-1})} \right\|_{(y=0, \overline{Y-1}; x=0, \overline{X-1})}$  параметров Хафа, графическим представлением которых является фазовое пространство.



а)



б)



в)



г)



д)



е)

Рис. 2. Формирование бинарного контурного образа исходного изображения и локализация прямых линий на бинарных образах  $\{C(\alpha)\}_{(\alpha=\{0^\circ, 45^\circ, 90^\circ, 135^\circ\})}$ : а) исходное изображение; б) бинарный контурный образ исходного изображения; в) локализация прямых при  $\alpha = 0^\circ$ ; г) локализация прямых при  $\alpha = 90^\circ$ ; д) локализация прямых при  $\alpha = 45^\circ$ ; е) локализация прямых при  $\alpha = 135^\circ$

При инициализации алгоритма матрицы  $\{H_A(\alpha)\}_{(\alpha=\{0^\circ, 45^\circ, 90^\circ, 135^\circ\})}$  и  $\{H_D(\alpha)\}_{(\alpha=\{0^\circ, 45^\circ, 90^\circ, 135^\circ\})}$  обнуляются.

4. Анализ фазового пространства преобразования Хафа и выделение прямых контурных линий на бинарных образах  $\{C(\alpha)\}_{(\alpha=\{0^\circ, 45^\circ, 90^\circ, 135^\circ\})}$ . Формируются матрица  $P_S(\alpha) = \|p_S(y, x, \alpha)\|_{(y=\overline{0, Y-1}, x=\overline{0, X-1})}$  принадлежности контурных пикселей прямым контурным линиям и матрица  $P_P(\alpha) = \|p_P(n_S, \alpha)\|_{(n_S=\overline{1, N_S(\alpha)})}$  вероятностей прямых контурных линий, значения элементов которых определяются в результате анализа параметров фазовых пространств  $\{H_A(\alpha)\}_{(\alpha=\{0^\circ, 45^\circ, 90^\circ, 135^\circ\})}$  и  $\{H_D(\alpha)\}_{(\alpha=\{0^\circ, 45^\circ, 90^\circ, 135^\circ\})}$  с помощью выражений

$$\forall h_A(y, x, n_V, \alpha) \forall h_D(y, x, n_V, \alpha) \left( (h_A(y, x, n_V, \alpha) = h_A(n_S, \alpha)) \wedge (h_D(y, x, n_V, \alpha) = h_D(n_S, \alpha)) \right) \rightarrow \\ \rightarrow \left( (p_S(y, x, \alpha) \Leftarrow n_S), (p_P(n_S, \alpha) \Leftarrow p_P(n_S, \alpha) + 1) \right), \quad (2)$$

$$y = \overline{0, Y-1}, \quad x = \overline{0, X-1}, \quad \alpha = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}, \quad n_V = \overline{0, N_V-1},$$

где  $N_S(\alpha)$  – число контурных линий, выделенных на бинарном образе  $C(\alpha)$ ;  $\{h_A(n_S, \alpha), h_D(n_S, \alpha)\}$  – параметры прямой линии с индексом  $n_S$ ;  $h_A(n_S, \alpha)$  – угол между осью абсцисс и перпендикуляром к прямой с индексом  $n_S$ ;  $h_D(n_S, \alpha)$  – длина этого перпендикуляра.

В результате выполнения данного шага значение каждого элемента  $p_S(y, x, \alpha) \neq 0$  матрицы  $P_S(\alpha)$  принадлежности, соответствующего элементу  $p_C(y, x) = 1$  бинарного контурного образа  $P_C$ , указывает на прямую линию с индексом  $n_S$  и параметрами  $\{h_A(n_S, \alpha), h_D(n_S, \alpha)\}$ . Число выявленных прямых линий на бинарном образе  $C(\alpha)$  определяется числом ненулевых элементов матрицы  $P_P(\alpha)$  вероятностей. При инициализации алгоритма матрицы  $P_S(\alpha)$  и  $P_P(\alpha)$  обнуляются.

5. Удаление коротких и прерывистых прямых контурных линий на бинарных образах  $\{C(\alpha)\}_{(\alpha=\{0^\circ, 45^\circ, 90^\circ, 135^\circ\})}$ . За счет предварительного масочного квантования элементов бинарного контурного образа  $P_C$  на шаге 2 для всех ненулевых значений матрицы  $P_P(\alpha)$  вероятностей выполняется условие  $p_P(n_S, \alpha) \geq 3$ , т. е. все выделенные на шаге 4 прямые контурные линии имеют длину не менее трех пикселей. Для дополнительных ограничений на длину локализуемых контурных линий снизу и их непрерывность используется выражение

$$\forall p_P(n_S, \alpha) \left( (p_P(n_S, \alpha) \geq 0) \wedge \left( (p_P(n_S, \alpha) < P_T) \vee (L(n_S, \alpha) / p_P(n_S, \alpha) > L_T) \right) \right) \rightarrow (p_P(n_S, \alpha) \Leftarrow 0), \quad (3)$$

где  $L(n_S, \alpha)$  – длина контурной линии с индексом  $n_S$ , ограниченной максимально удаленными элементами  $p_S(y_1, x_1, \alpha) = n_S$  и  $p_S(y_2, x_2, \alpha) = n_S$ ;  $P_T$  – порог, определяющий минимальное число контурных элементов, образующих прямую контурную линию;  $L_T$  – порог, определяющий количество и общую величину разрывов контурной линии на длине  $L(n_S, \alpha)$ .

6. Локализация прямых линий на бинарных образах  $\{C(\alpha)\}_{(\alpha=\{0^\circ, 45^\circ, 90^\circ, 135^\circ\})}$  (см. рис. 2). Значения элементов матрицы  $P_S(\alpha)$  принадлежности приводятся в соответствие со значениями элементов матрицы  $P_P(\alpha)$  вероятностей с помощью выражения

$$\forall p_S(y, x, \alpha) \left( (p_S(y, x, \alpha) = n_S) \wedge (p_P(n_S, \alpha) = 0) \right) \rightarrow (p_S(y, x, \alpha) \leftarrow 0) \quad (4)$$

при  $y = \overline{0, Y-1}$ ,  $x = \overline{0, X-1}$ ,  $\alpha = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ .

В результате выполнения данного шага значение каждого элемента  $p_S(y, x, \alpha) \neq 0$  матрицы  $P_S(\alpha)$  принадлежности указывает на прямую контурную линию с индексом  $n_S$ , проходящую через пиксел с координатами  $(y, x)$  бинарного контурного образа  $P_C$  и имеющую параметры  $\{h_A(n_S, \alpha), h_D(n_S, \alpha)\}$ .

7. Объединение результатов поиска прямых контурных линий на бинарных образах  $\{C(\alpha)\}_{(\alpha=\{0^\circ, 45^\circ, 90^\circ, 135^\circ\})}$ . Формируются результирующее множество  $\{h_A(n_R), h_D(n_R)\}_{(n_R=\overline{1, N_R})}$  параметров прямых контурных линий и результирующая матрица  $P_R = \|p_R(y, x)\|_{(y=\overline{0, Y-1}, x=\overline{0, X-1})}$  принадлежности контурных пикселов бинарного контурного образа  $P_C$  прямым контурным линиям, значения которых вычисляются с помощью выражений

$$h_A(n_R) = \begin{cases} h_A(n_S, 0^\circ) \text{ при } n_R \leq N_S(0^\circ), \\ h_A(n_S, 45^\circ) \text{ при } N_S(0^\circ) < n_R \leq (N_S(0^\circ) + N_S(45^\circ)), \\ h_A(n_S, 90^\circ) \text{ при } (N_S(0^\circ) + N_S(45^\circ)) < n_R \leq (N_S(0^\circ) + N_S(45^\circ) + N_S(90^\circ)), \\ h_A(n_S, 135^\circ) \text{ при } (N_S(0^\circ) + N_S(45^\circ) + N_S(90^\circ)) < n_R \end{cases} \quad (5)$$

при  $n_R = \overline{1, N_R}$ ;

$$h_D(n_R) = \begin{cases} h_D(n_S, 0^\circ) \text{ при } n_R \leq N_S(0^\circ), \\ h_D(n_S, 45^\circ) \text{ при } N_S(0^\circ) < n_R \leq (N_S(0^\circ) + N_S(45^\circ)), \\ h_D(n_S, 90^\circ) \text{ при } (N_S(0^\circ) + N_S(45^\circ)) < n_R \leq (N_S(0^\circ) + N_S(45^\circ) + N_S(90^\circ)), \\ h_D(n_S, 135^\circ) \text{ при } (N_S(0^\circ) + N_S(45^\circ) + N_S(90^\circ)) < n_R \end{cases} \quad (6)$$

при  $n_R = \overline{1, N_R}$ ;

$$p_R(y, x) = p_S(y, x, 0^\circ) \cup (p_S(y, x, 45^\circ) + N_S(0^\circ)) \cup (p_S(y, x, 90^\circ) + N_S(0^\circ) + N_S(45^\circ)) \cup (p_S(y, x, 135^\circ) + N_S(0^\circ) + N_S(45^\circ) + N_S(90^\circ)) \quad (7)$$

при  $y = \overline{0, Y-1}$ ,  $x = \overline{0, X-1}$ ,

где  $N_R = N_S(0^\circ) + N_S(45^\circ) + N_S(90^\circ) + N_S(135^\circ)$  – общее число выделенных на шаге 4 контурных линий на образах  $\{C(\alpha)\}_{\alpha=\{0^\circ, 45^\circ, 90^\circ, 135^\circ\}}$ . Выражение (7) описывает объединение матриц  $P_S(\alpha)$  принадлежности, в результате которого значения элементов  $p_R(y, x)$  результирующей матрицы  $P_R$  принадлежности распределены в диапазоне  $p_R(y, x) \in [1, N_R]$ .

В результате выполнения данного алгоритма формируются множество  $\{h_A(n_R), h_D(n_R)\}_{(n_R=1, N_R)}$  параметров прямых контурных линий и матрица  $P_R$  принадлежности контурных пикселей бинарного контурного образа  $P_C$  прямым контурным линиям, ненулевые элементы которой указывают на индексы  $n_R$  прямых контурных линий с параметрами  $\{h_A(n_R), h_D(n_R)\}$ .

### **3. Оценка эффективности масочно-фазовой локализации прямых линий на изображении**

Для оценки эффективности предложенного масочно-фазового метода локализации прямых линий на изображении проведено его сравнение с методом локализации прямых на основе преобразования Хафа. В качестве критериев эффективности использованы вычислительная сложность локализации прямых линий на изображении, вероятность ошибочной локализации и устойчивость результатов локализации к изменению яркости, контраста и зашумлению изображения.

Оценка вычислительной сложности методов локализации прямых линий получена исходя из числа угловых ориентаций виртуальных прямых, проводимых через контурные пиксели. Поскольку сектор параметризации точек в преобразовании Хафа составляет  $180^\circ$ , а в масочно-фазовом методе  $2\Delta_\kappa = 50^\circ$ , при одинаковой плотности виртуальных прямых вычислительная сложность масочно-фазового метода в 3,6 раза меньше, чем метода на основе преобразования Хафа, что подтверждается экспериментально (время локализации прямых на процессоре Intel i5 2.27 ГГц в MatLab 2012b составляет 0,75 с для масочно-фазового метода и 2,67 с для преобразования Хафа). Однако масочно-фазовый метод незначительно уступает по быстродействию методу LSD [6] (время локализации прямых на процессоре Intel i5 2.27 ГГц в MatLab 2012b составляет 0,28 с для метода LSD, что в 2,6 раза лучше по сравнению с масочно-фазовым методом).

Оценка вероятности ошибочной локализации прямых линий получена на основе отношения числа ошибочно локализованных прямых линий к общему числу локализованных прямых линий на изображении. В результате эксперимента над 20 изображениями, содержащими множество мелких деталей, и 20 изображениями, содержащими преимущественно крупные детали, установлено, что вероятность ошибочной локализации прямых масочно-фазовым методом меньше на 43 и 56 % для изображений, содержащих множество мелких и преимущественно крупных деталей соответственно, по сравнению с методом на основе преобразования Хафа. Для оценки устойчивости результатов локализации прямых линий использовано отношение числа прямых, выделяемых на изображении при изменении яркости, контраста и зашумлении, к числу прямых, выделяемых на исходном изображении. На рис. 3 показаны характеристики устойчивости результатов локализации прямых линий с помощью масочно-фазового метода и метода на основе преобразования Хафа для тестового изображения, приведенного на рис. 2. Изменение уровня яркости, контраста и зашумления изображения производилось средствами графического редактора Adobe Photoshop.

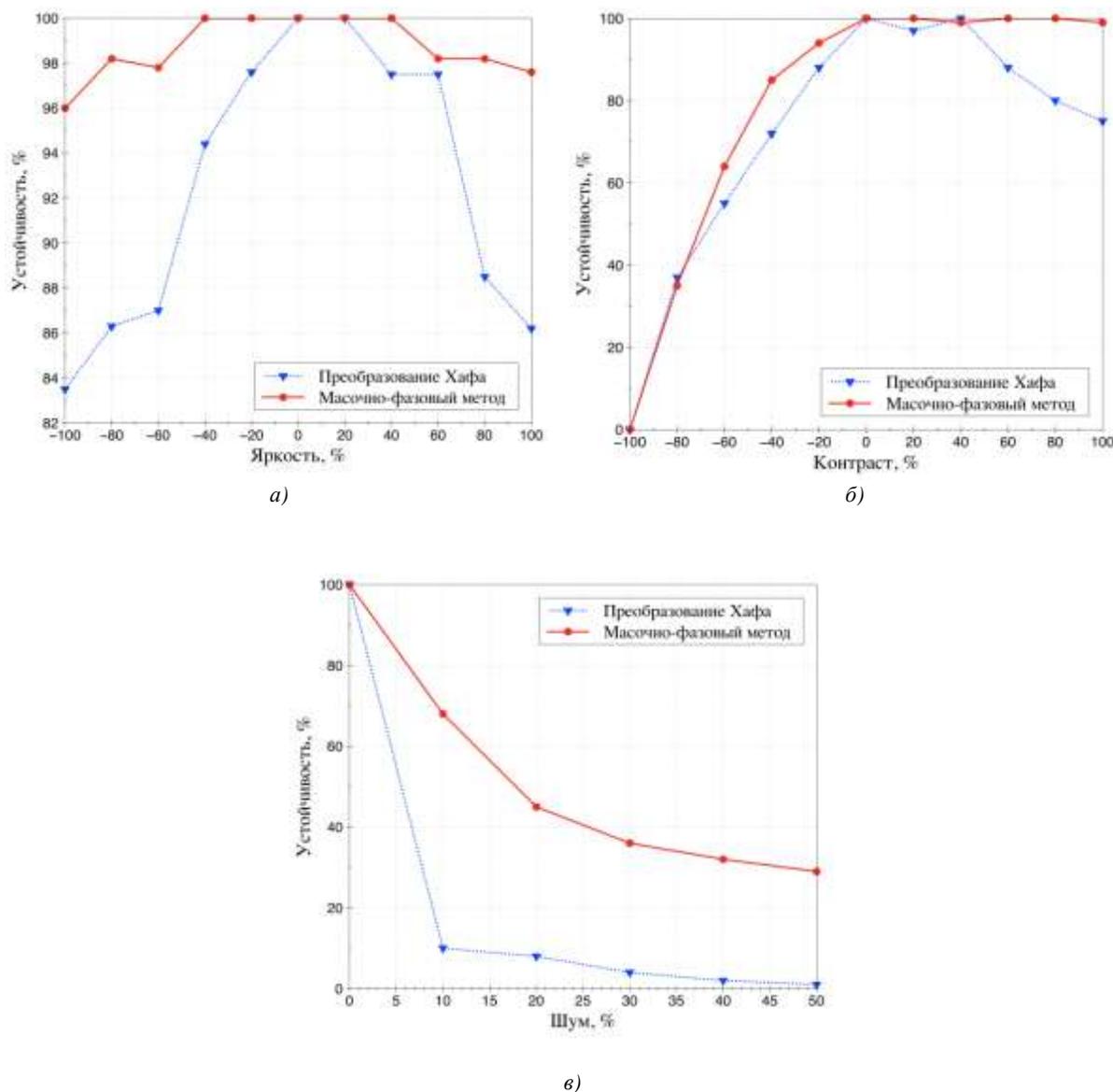


Рис. 3. Оценки устойчивости результатов локализации прямых линий на изображении: а) при изменении яркости; б) при изменении контраста; в) при зашумлении

На рис. 3, а и б видно, что масочно-фазовый метод локализации прямых превосходит по стабильности метод на основе преобразования Хафа на 12 % при изменении яркости и на 22 % при изменении контраста. Масочно-фазовый метод позволяет более чем на 35 % повысить устойчивость результатов локализации прямых к шуму по сравнению с преобразованием Хафа (рис. 3, в) за счет фильтрации бинарного контурного образа на основе использования предварительного масочного квантования его элементов. Установлено, что масочно-фазовый метод одинаково эффективен для реальных изображений и «синтезированных» (полученных в результате совмещения нескольких реальных изображений). На рис. 4 показаны результаты локализации прямых линий на реальных аэрокосмических изображениях, а на рис. 5 – на «синтезированных» изображениях земной поверхности (интернет-ресурс Google Maps).

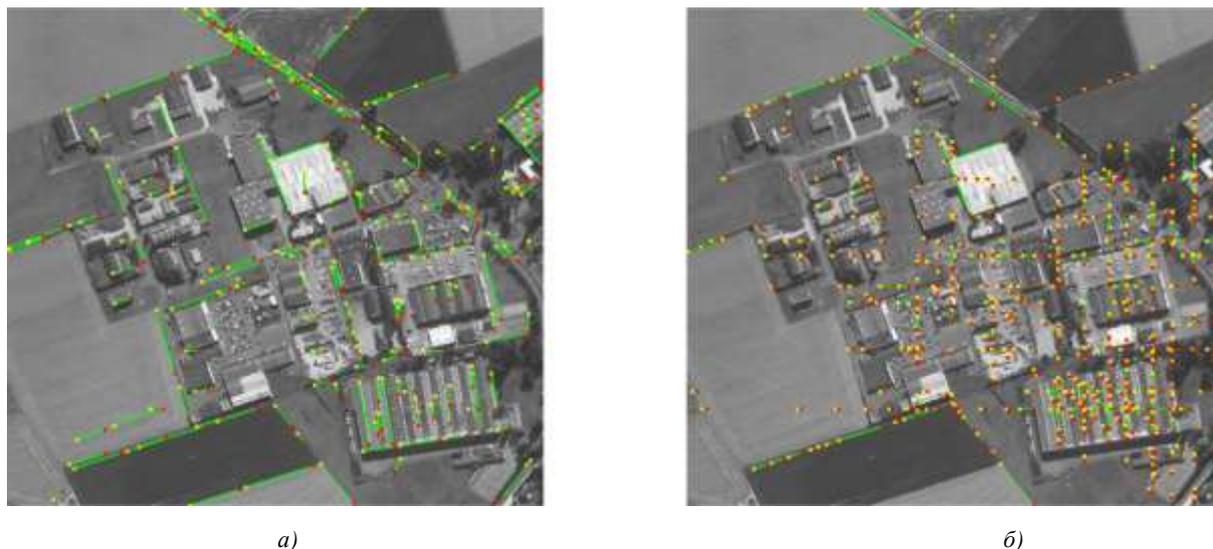


Рис. 4. Локализация прямых линий на реальном изображении:  
*а)* масочно-фазовый метод; *б)* классическое преобразование Хафа

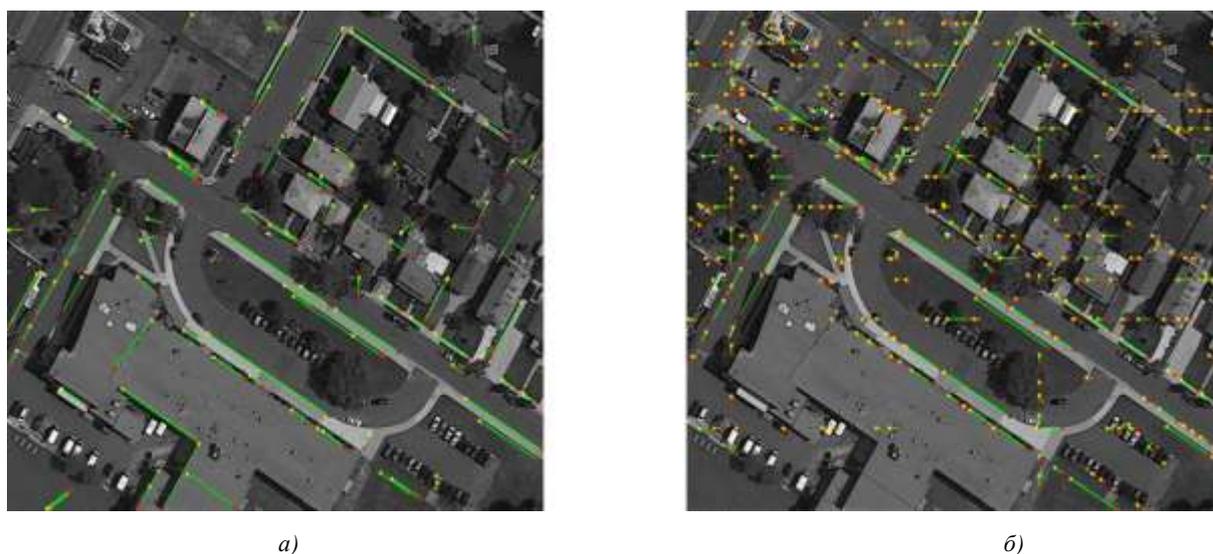


Рис. 5. Локализация прямых линий на «синтезированном» изображении:  
*а)* масочно-фазовый метод; *б)* классическое преобразование Хафа

Экспериментально установлено, что прямые линии, выделяемые с помощью масочно-фазового метода, достаточно устойчивы к изменению яркости (рис. 6, *г–е* – увеличение яркости на 50 %), контраста (рис. 6, *ж–и* – увеличение контраста на 50 %) и зашумления (рис. 6, *к–м* – увеличение уровня шума на 10 %) изображения. В таблице приведены характеристики устойчивости локализации линий методами LSD и масочно-фазовым для изображений, представленных на рис. 6, при изменении яркости, контраста и уровня шума. Из таблицы и рис. 6 следует, что масочно-фазовый метод выделяет меньшее количество прямых линий (рис. 6, *а–в*) из-за более строгих ограничений на их структуру (LSD захватывает прямые отрезки ломаных контурных линий). Вместе с тем масочно-фазовый метод обеспечивает большую стабильность местоположения локализуемых линий на изображении при изменении характеристик изображений: на 41 % при увеличении яркости на 50 %, на 29 % при увеличении контраста на 50 % и на 28 % при увеличении уровня шума на 10 %.

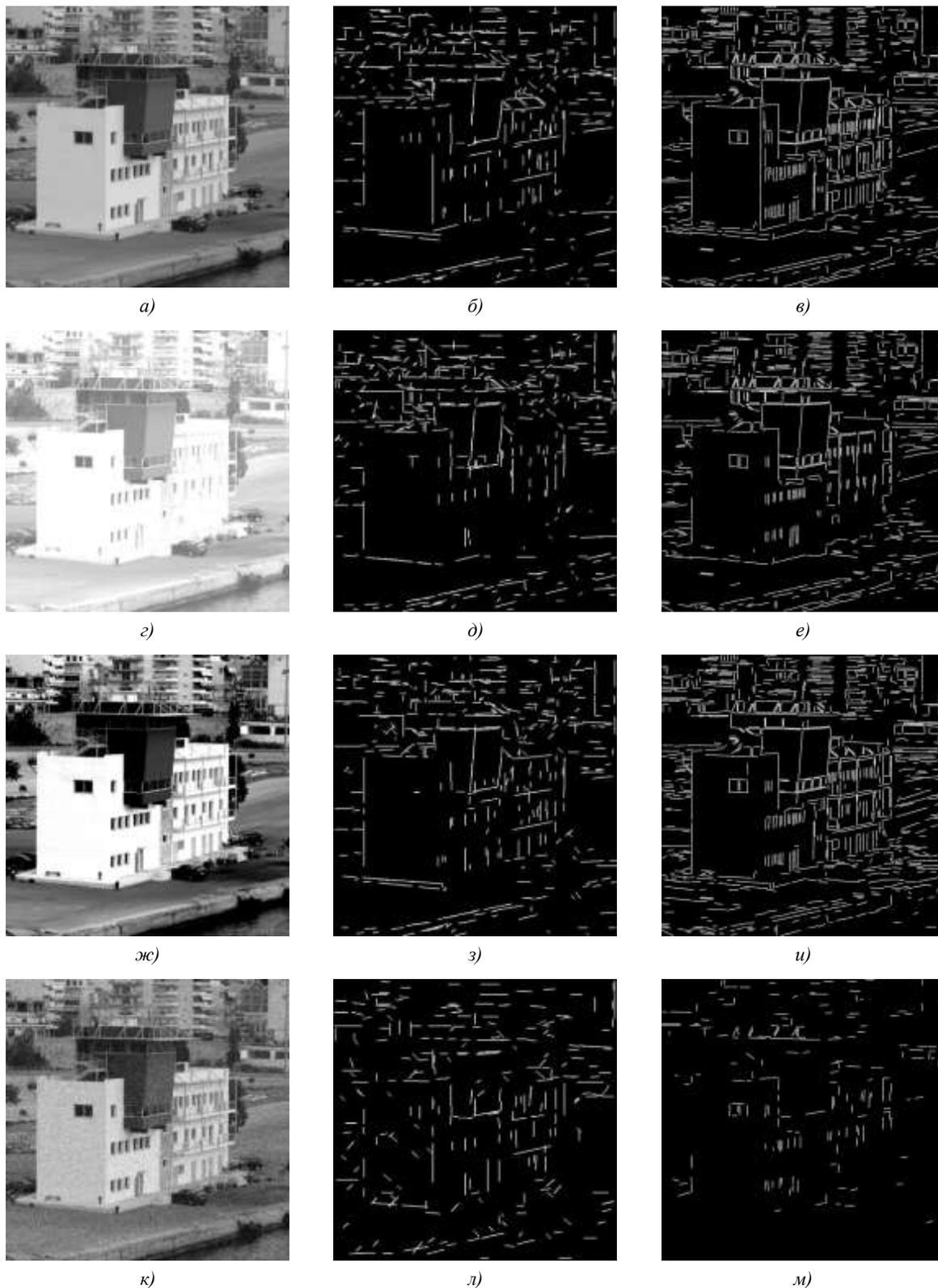


Рис. 6. Локализация прямых линий на изображении: *а)* исходное изображение; *б)* локализация масочно-фазовым методом на исходном изображении; *в)* методом LSD на исходном изображении; *г)* засвеченное изображение (яркость увеличена на 50 %); *д)* локализация масочно-фазовым методом на засвеченном изображении; *е)* методом LSD на засвеченном изображении; *ж)* высококонтрастное изображение (контраст увеличен на 50 %); *з)* локализация масочно-фазовым методом на высококонтрастном изображении; *и)* методом LSD на высококонтрастном изображении; *к)* зашумленное изображение (интенсивность шума 10 %); *л)* локализация масочно-фазовым методом на зашумленном изображении; *м)* методом LSD на зашумленном изображении

Влияние увеличения яркости, контраста и уровня шума изображения на стабильность линий

Метод	Исходное число линий	Число линий, сохранивших местоположение, при увеличении		
		яркости на 50 %	контраста на 50 %	уровня шума на 10 %
LSD	847	297 (35 %)	324 (38 %)	134 (16 %)
Масочно-фазовый	367	278 (76 %)	247 (67 %)	163 (44 %)

Локализация линий методом LSD основана на вычислении градиентов. Поэтому при изменении характеристик изображения линии могут смещаться в пределах нескольких пикселей относительно исходного положения, что приводит к уменьшению стабильности локализации местоположения линий.

### **Заключение**

В работе предложен масочно-фазовый метод локализации прямых линий на изображении, основанный на библиотеке бинарных масок и пространственно-ориентированном преобразовании Хафа. Суть метода состоит в использовании библиотеки малоразмерных бинарных масок для предварительного грубого квантования контурных отрезков по ориентации и последующем уточнении их принадлежности прямым линиям с помощью пространственно-ориентированного преобразования Хафа, применяемого к каждому квантованному контурному отрезку в ограниченном секторе, определяемом ориентацией этого отрезка. Установлено, что по сравнению с методом локализации прямых на основе преобразования Хафа предложенный масочно-фазовый метод обеспечивает сокращение вычислительной сложности локализации в 3,6 раза, уменьшение вероятности ложной локализации прямых на 43 и 56 % для изображений, содержащих множество мелких и преимущественно крупных деталей соответственно, повышение устойчивости результатов локализации на 12 % к изменению яркости, на 22 % к изменению контраста и более чем на 35 % к зашумлению изображения. Показано, что по сравнению с методом LSD предложенный масочно-фазовый метод обеспечивает повышение стабильности локализации прямых на 41 % при увеличении яркости, на 29 % при увеличении контраста и на 28 % при увеличении уровня шума и уменьшении быстродействия в 2,6 раза.

### **Список литературы**

1. Lowe, D. Distinctive image features from scale invariant keypoints / D. Lowe // International J. of Computer Vision. – 2004. – Vol. 60, № 2. – P. 91–110.
2. Bay, H. SURF : Speeded up robust features / H. Bay // Proc. of the 9th European Conference on Computer Vision. – 2006. – Vol. 3951. – P. 404–410.
3. Hirschmuller, H. Realtime correlation-based stereo vision with reduced border errors / H. Hirschmuller, J.M. Garibaldi // International J. of Computer Vision. – 2002. – Vol. 47, № 1–3. – P. 229–246.
4. Duda, R.O. Use of the Hough Transformation to Detect Lines and Curves in Pictures / R.O. Duda // Communication of the ACM. – 1972. – Vol. 15, № 1. – P. 229–246.
5. Anver, M.M. Fuzzy edge detection using competition between multiple masks / M.M. Anver, R.J. Stonier // Proc. of the 2nd International Conference on Computational Intelligence, Robotics and Autonomous Systems, CIRAS 2003. – Singapore, 2003. – P. 344–348.
6. LSD: A Fast Line Segment Detector with a False Detection Control / R. Grompone von Gioi [et al.] // IEEE Transactions on Pattern. Analysis and Machine Intelligence. – 2010. – Vol. 32, № 4. – P. 722–732.

7. Chan, T.S. Line detection algorithm / T.S. Chan, K.K. Raymond // Proc. of 13<sup>th</sup> Int. Conference on Pattern Recognition, ICPR 1996. – Vienna, 1996. – P. 126–130.

Поступила 18.06.2013

*Белорусский государственный университет  
информатики и радиоэлектроники,  
Минск, ул. П. Бровки, 6  
e-mail: vtsvet@bsuir.by*

**A.A. Zhuravlyov, V.Yu. Tsviatkou**

**MASK-PHASE METHOD OF LOCALIZATION  
OF STRAIGHT LINES IN AN IMAGE**

A mask-phase method of localization of straight lines in an image is proposed. It is based on a preliminary quantization of line segments with respect to orientation by using a library of binary masks and rectification of straight lines by using oriented Hough transform. The method allows improving the speed and quality of straight lines localization if there are changes of brightness, contrast and image noise.

УДК 004.931; 004.932

А.С. Поляков, В.Е. Самсонов

## ОЦЕНКА ХАРАКТЕРИСТИК АППАРАТНОЙ РЕАЛИЗАЦИИ НЕКОТОРЫХ АЛГОРИТМОВ ФИЛЬТРАЦИИ ИЗОБРАЖЕНИЙ

*Приводятся данные о быстродействии и требуемых объемах оборудования при аппаратной реализации в базисе микросхем типа FPGA двух часто применяемых алгоритмов фильтрации изображений. Рассматривается возможность использования алгоритмов для обработки стандартных кадров видеопоследовательностей.*

### Введение

В настоящее время задачи отслеживания динамических объектов на видеопоследовательностях, снятых в видимом и инфракрасном диапазонах, являются объектом интереса огромного числа специалистов ведущих исследовательских центров и IT-компаний. Одной из основных трудностей, возникающих при решении указанной задачи, является малое время (0,03–0,04 с), отведенное на обработку каждого кадра видеопоследовательности стандартного размера 768x576 пикселей. Наиболее сложные известные алгоритмы, предназначенные для обработки видеопоследовательностей, непригодны для работы в режиме реального времени [1–3]. В этих условиях сокращение времени выполнения алгоритмов является одной из актуальных проблем в области обработки изображений, особенно при обработке видеопоследовательностей в реальном масштабе времени. Следует отметить, что изображения, полученные видеокамерами, как правило, содержат шум и поэтому нуждаются в предварительной обработке, улучшающей их качество. Поэтому одним из основных требований к алгоритмам обнаружения, отслеживания и распознавания объектов на полутоновых и цветных видеопоследовательностях является высокое быстродействие, достаточное для обработки каждого кадра получаемой цифровой видеoinформации в режиме реального времени при обеспечении должного качества.

Для улучшения качества изображений, получаемых видеокамерами в режиме реального времени, используются оконные линейные фильтры сверточного типа и медианные фильтры, которые считаются одними из лучших в области обработки изображений. Однако их применение при обработке видеопоследовательностей затруднено из-за значительной вычислительной сложности, приводящей к недопустимо большим временным затратам [2–4].

В связи с необходимостью выполнения сложных алгоритмов отслеживания динамических объектов в реальном масштабе времени при сохранении высокого качества видеопоследовательности актуальным является решение задачи повышения скорости выполнения алгоритмов фильтрации. Для увеличения их быстродействия наряду с оптимизацией применяются многопроцессорные системы или аппаратная реализация [5], которая является одним из самых эффективных способов увеличения производительности алгоритмов. К сожалению, вопросу обработки изображений путем их аппаратной реализации на основе современных микроэлектронных средств в отечественной литературе уделяется мало внимания. Настоящая статья посвящена частичному устранению пробела в этой области. В ней представлены данные о характеристиках аппаратной реализации в базисе микросхем типа FPGA [5–7] для двух часто применяемых на практике алгоритмов фильтрации изображений в терминах объемно-временных характеристик, которые используются при оценке результатов проектирования во всех САПР для FPGA и под которыми подразумеваются затраты оборудования, необходимые для реализации алгоритмов, и время их выполнения (производительность, быстродействие).

### 1. Краткое описание исследуемых алгоритмов

*Фильтр «медиана». Медианой дискретной последовательности  $n$  элементов при нечетном  $n$  называется элемент, для которого существует не менее  $(n-1)/2$  элементов, меньших или рав-*

ных ему, и не менее  $(n-1)/2$  элементов, больших или равных ему по величине. При рассмотрении элементарной матрицы размером  $3 \times 3$  элемента для нахождения медианы требуется произвести не менее 19 операций сравнения [3], которые при программной реализации выполняются последовательно, т. е. требуется не менее 19 тактов вычислительного процесса. С учетом возможностей распараллеливания операций при аппаратной реализации алгоритма можно обойтись меньшим количеством тактов.

Рассмотрим элементарные матрицы  $\mathbf{M}$ ,  $\mathbf{M}_1$ ,  $\mathbf{M}_2$  размером  $3 \times 3$  элемента (рис. 1). Видно, что при фильтрации изображений производится переычисление только одного элемента, расположенного на пересечении диагоналей матрицы.

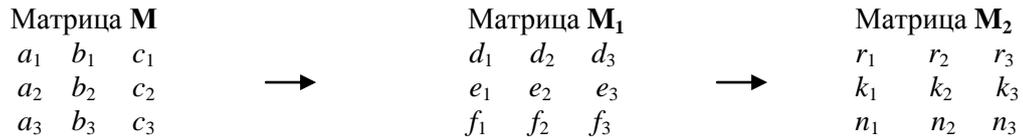


Рис. 1. Матрицы  $\mathbf{M}$ ,  $\mathbf{M}_1$  и  $\mathbf{M}_2$

Для нахождения медианы необходимо упорядочить элементы матрицы по возрастанию (точнее, по неубыванию) значений. Пятый элемент упорядоченной таким образом последовательности из элементов  $a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3$  является медианой рассматриваемого множества элементов. Для ее получения используем алгоритм, ориентированный на аппаратную реализацию.

1. Упорядочим элементы в столбцах матрицы по возрастанию (точнее, по неубыванию) значений. Это означает, что в каждом из столбцов матрицы предыдущий элемент равен или меньше следующего. Получим матрицу  $\mathbf{M}_1$ , элементы которой находятся в отношениях  $d_1 \leq e_1 \leq f_1, d_2 \leq e_2 \leq f_2, d_3 \leq e_3 \leq f_3$ .

2. Упорядочим по неубыванию значений элементы в строках матрицы  $\mathbf{M}_1$ . Получим матрицу  $\mathbf{M}_2$ , в которой  $r_1 \leq r_2 \leq r_3, k_1 \leq k_2 \leq k_3, n_1 \leq n_2 \leq n_3$ .

3. Поскольку упорядочивание элементов матриц  $\mathbf{M}_1$  и  $\mathbf{M}_2$  производилось по неубыванию значений сверху вниз и слева направо, значения элементов в матрице  $\mathbf{M}_2$  возрастают слева направо и сверху вниз, поэтому элементы  $r_1, r_2$  и  $k_1$  занимают первые три места в упорядоченной по неубыванию значений последовательности элементов матрицы  $\mathbf{M}$ . Следовательно, медианой матрицы является тот из элементов  $n_1, k_2, r_3$ , который имеет среднее значение среди них.

С целью минимизации затрат для определения порядка следования сравниваемых элементов в столбцах и строках матриц  $\mathbf{M}_1$  и  $\mathbf{M}_2$  соответственно воспользуемся следующим приемом. Результат отношения неубывания будем описывать с помощью логических переменных  $p_1, p_2, p_3$ :

$$\begin{aligned}
 p_1 &= 1, \text{ если } a_1 \leq a_2; \quad p_1 = 0, \text{ если } a_1 > a_2; \\
 p_2 &= 1, \text{ если } a_1 \leq a_3; \quad p_2 = 0, \text{ если } a_1 > a_3; \\
 p_3 &= 1, \text{ если } a_2 \leq a_3; \quad p_3 = 0, \text{ если } a_2 > a_3.
 \end{aligned} \tag{1}$$

На основании анализа значений переменных  $p_1, p_2$  и  $p_3$  можно установить порядок расположения элементов  $a_1, a_2, a_3$  в последовательности по неубыванию значений. Результаты анализа представлены в табл. 1.

Таблица 1

Упорядочивание элементов (все отношения)

$p_1$	$p_2$	$p_3$	Порядок следования элементов по неубыванию значений
0	0	0	$a_1 \leq a_2 \leq a_3$
0	0	1	$a_1 \leq a_3 \leq a_2$
0	1	0	условия противоречивые
0	1	1	$a_3 \leq a_1 \leq a_2$
1	0	0	$a_2 \leq a_1 \leq a_3$
1	0	1	условия противоречивые
1	1	0	$a_2 \leq a_3 \leq a_1$
1	1	1	$a_3 \leq a_2 \leq a_1$

После удаления строк, содержащих противоречивые условия, получаем табл. 2.

Таблица 2

Упорядочивание элементов (без противоречивых условий)

$p_1$	$p_2$	$p_3$	Порядок следования элементов по неубыванию значений
0	0	0	$a_1 \leq a_2 \leq a_3$
0	0	1	$a_1 \leq a_3 \leq a_2$
0	1	1	$a_3 \leq a_1 \leq a_2$
1	0	0	$a_2 \leq a_1 \leq a_3$
1	1	0	$a_2 \leq a_3 \leq a_1$
1	1	1	$a_3 \leq a_2 \leq a_1$

Линейный *фильтр «свертка»* обеспечивает преобразование исходной матрицы  $\mathbf{M}$  в результирующую матрицу  $\underline{\mathbf{S}}$  с помощью задаваемой пользователем матрицы  $\mathbf{S}$  поправочных коэффициентов. Рассматриваемые матрицы  $\mathbf{S}$ ,  $\mathbf{M}$  и  $\underline{\mathbf{S}}$  представлены ниже:

$$\mathbf{S} = \begin{vmatrix} s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,1} & s_{3,2} & s_{3,3} \end{vmatrix} \times \mathbf{M} = \begin{vmatrix} m_{1,1} & m_{1,2} & m_{1,3} \\ m_{2,1} & m_{2,2} & m_{2,3} \\ m_{3,1} & m_{3,2} & m_{3,3} \end{vmatrix} \rightarrow \underline{\mathbf{S}} = \begin{vmatrix} \underline{s}_{1,1} & \underline{s}_{1,2} & \underline{s}_{1,3} \\ \underline{s}_{2,1} & \underline{s}_{2,2} & \underline{s}_{2,3} \\ \underline{s}_{3,1} & \underline{s}_{3,2} & \underline{s}_{3,3} \end{vmatrix}$$

Значение элемента  $m_{2,2}$  матрицы  $\mathbf{M}$  преобразуется в значение элемента  $\underline{s}_{2,2}$  матрицы  $\underline{\mathbf{S}}$  следующим образом:

– вычисляется

$$\begin{aligned} K = & s_{1,1} * m_{1,1} + s_{1,2} * m_{1,2} + s_{1,3} * m_{1,3} + s_{2,1} * m_{2,1} + s_{2,2} * m_{2,2} + \\ & + s_{2,3} * m_{2,3} + s_{3,1} * m_{3,1} + s_{3,2} * m_{3,2} + s_{3,3} * m_{3,3}; \end{aligned} \quad (2)$$

– в зависимости от величины  $K$  определяется значение  $\underline{s}_{2,2}$  – результат «свертки» матрицы  $\mathbf{M}$ :  $\underline{s}_{2,2} = 255$ , если  $K \geq 255$ ;  $\underline{s}_{2,2} = 0$ , если  $K \leq 0$ ;  $\underline{s}_{2,2} = K$  в остальных случаях.

## 2. Аппаратная реализация алгоритмов

Аппаратную реализацию алгоритмов можно выполнить в виде синхронной последовательностной схемы либо асинхронной комбинационной схемы. Комбинационные схемы обеспечивают большее быстродействие, поэтому с помощью системы проектирования фирмы Xilinx Ise 9.2i были разработаны проекты реализации рассматриваемых алгоритмов фильтрации изображений в виде комбинационных схем.

В алгоритме «медиана» основными являются операции сравнения. Поскольку при вычислении значений средних элементов исходных матриц порядок и очередность выполнения операций сравнения не имеют значения, при аппаратной реализации все операции сравнения могут выполняться параллельно, что легко увидеть на структурной схеме (рис. 2).

Обозначения переменных на рис. 2–4 соответствуют обозначениям, используемым в рассматриваемых выше матрицах  $\mathbf{M}$ ,  $\mathbf{M}_1$  и  $\mathbf{M}_2$ . Показанные блоки comp1–comp4 и median реализуют функции сравнения трех величин по отношению неубывания и их упорядочивания в соответствии с табл. 2. Все блоки аналогичны представленному на рис. 2 блоку comp1, но в отличие от comp1 блоки comp2a–comp4 и median вычисляют только одно значение из трех возможных: блок comp2a – наибольшее значение, comp3 и median – среднее, comp4 – минимальное значение из входных переменных. Результат выполнения алгоритма представлен на выходе блока median в виде вектора **med** разрядностью 8 бит.

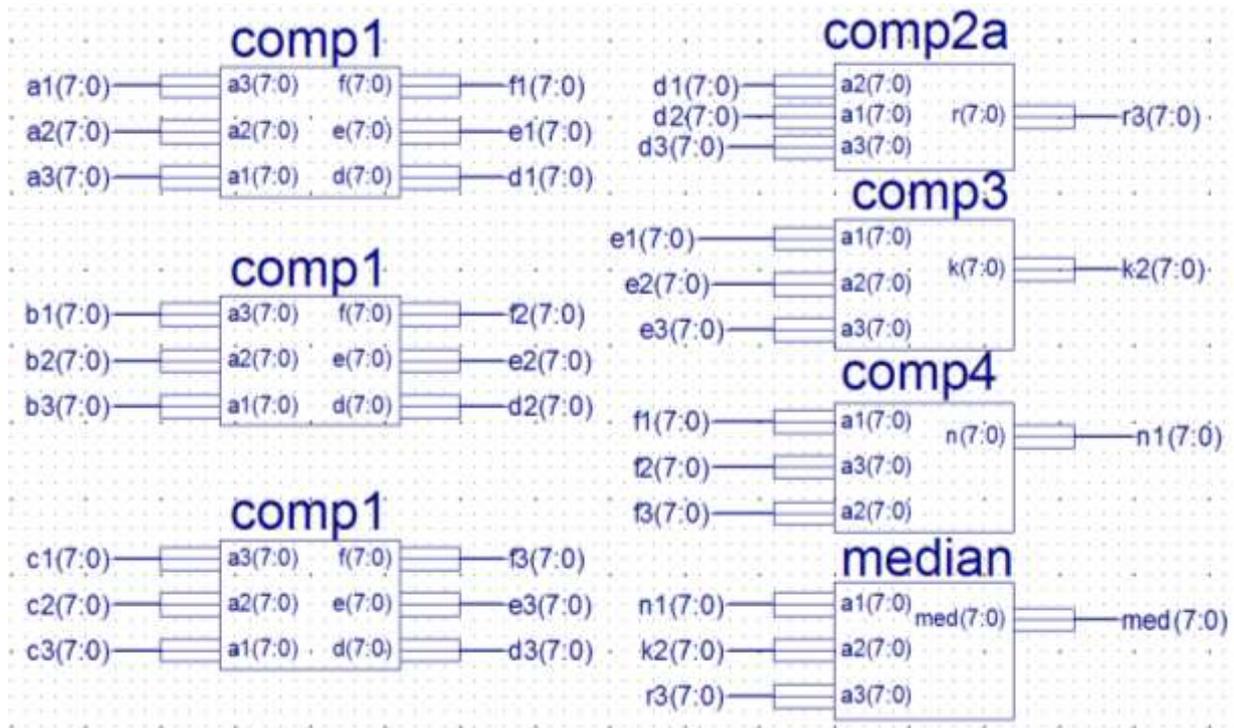
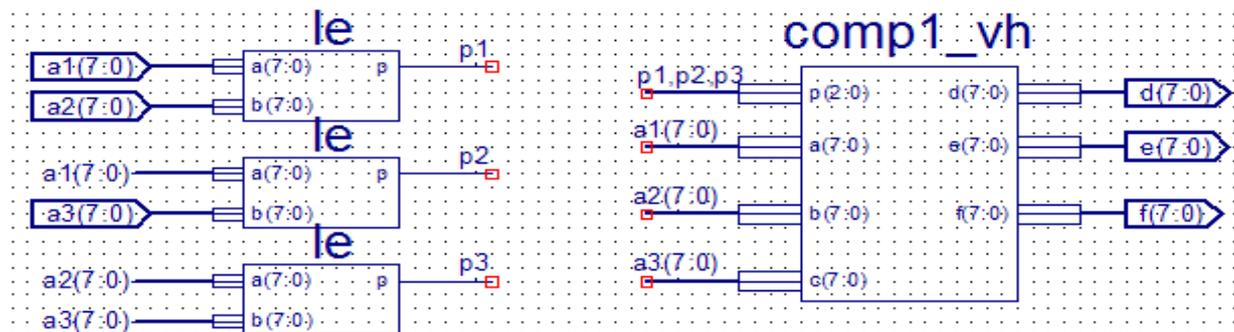


Рис. 2. Комбинационная схема, реализующая алгоритм «медиана»

Рис. 3. Блок **comp1**, реализующий упорядочивание элементов матрицы **M**

Представленный на рис. 3 блок **le** производит сравнение двух величин по отношению убывания и присваивает значение переменной  $p$  в соответствии с уравнением 1; блок **comp1\_vh** в зависимости от значений  $p_1, p_2, p_3$  в соответствии с табл. 2 определяет порядок следования переменных  $a_1, a_2, a_3$  по отношению убывания значений.

Аппаратная реализация алгоритма «свертка» имеет более сложную структуру (рис. 4), в которой сначала выполняются операции умножения элементов матрицы **M** на соответствующие элементы матрицы **S** с помощью умножителей **mult**, а затем с помощью четырехступенчатой схемы сумматоров и блока **res** вычисляется величина **K**, в зависимости от значения которой в соответствии с уравнением 2 определяется значение переменной **svert**, представляющей собой результат выполнения алгоритма «свертка» для рассматриваемой матрицы.

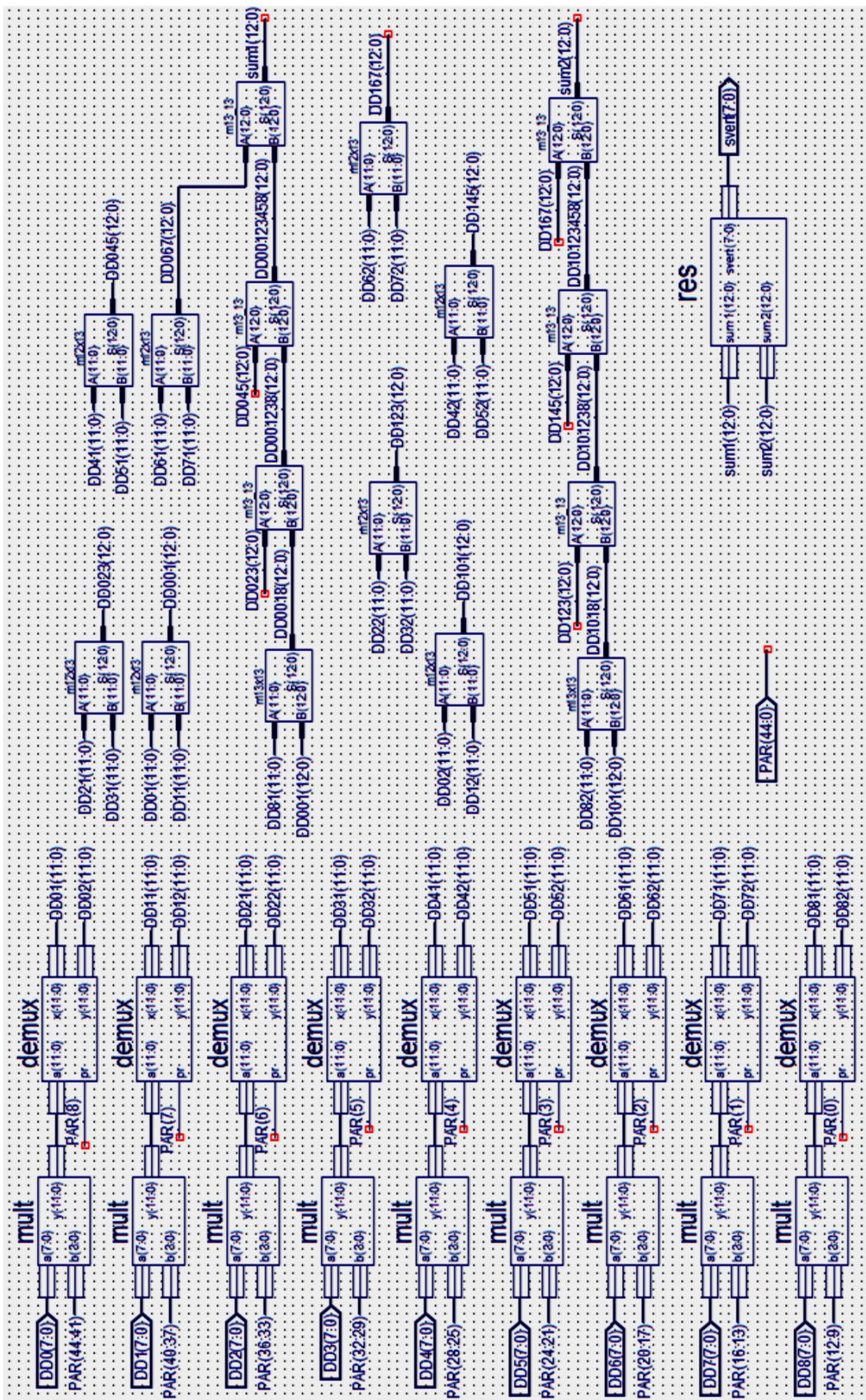


Рис. 4. Комбинаторная схема, реализующая алгоритм «свертка»

На рис. 4 используются следующие обозначения: demux – демультимплексоры; m12x12, m12x13, m13x13 – сумматоры; логические переменные PAR(0)– PAR(8) представляют знаки (+ или –) элементов матрицы **S**; PAR(9)–PAR(44) – значения элементов матрицы **S**; блок res вычисляет значение svert в зависимости от значений sum1 (положительные значения) и sum2 (отрицательные значения) в соответствии с уравнением 2.

### 3. Объемно-временные характеристики аппаратной реализации алгоритмов

Оценка характеристик аппаратной реализации исследуемых алгоритмов фильтрации изображений производилась с использованием результатов выполнения разработанных проектов в системе проектирования фирмы Xilinx ISE 9.2i (этапы Synthesize, Translate, Map) и логического проектирования с помощью системы моделирования ModelSim 6.2f.

Затраты оборудования на реализацию проекта, предусматривающего обработку одной элементарной матрицы размером 3x3 элемента, т. е. на вычисление значения среднего элемента матрицы, приведены в табл. 3 и 4. Значения затрат оборудования приведены в процентах от ресурса, имеющегося в микросхемах.

Таблица 3

Затраты оборудования на реализацию алгоритмов в микросхемах серии Spartan 3, %

Показатель	Тип микросхемы			
	xc3s200-5ft256		xc3s400-5fg456	
	медиана	свертка	медиана	свертка
Slices	14,5	16,2	7,8	8,7
4 Input LUTs	11,0	12,0	6,0	6,5
Flip Flops	5,5	3,0	3,0	1,6

Таблица 4

Затраты оборудования на реализацию алгоритмов в микросхемах серии Virtex 4

Показатель	Тип микросхемы					
	xc4vlx15-12sf363		xc4vlx25-12sf363		xc4vlx40-12ff668	
	медиана	свертка	медиана	свертка	медиана	свертка
Slices	4,5	7,1	2,6	4,2	1,5	2,4
4 Input LUTs	3,4	5,7	2,5	3,2	1,1	1,9
Flip Flops	1,8	0,9	1,0	0,5	0,6	0,3

*Примечание к табл. 3 и 4:* LUT (look-up table) – логическая таблица, представляющая собой однобитовое ОЗУ на 16 ячеек; Slice – единица оборудования, состоящая из двух триггеров и двух LUT, Flip-Flop – триггер.

Из табл. 3 и 4 видно, что затраты оборудования на реализацию алгоритмов для обработки одной элементарной матрицы составляют незначительную часть от возможностей микросхем. Это позволяет разработать проекты, предусматривающие загрузку и параллельную обработку матриц большего размера, что существенно в случае фильтрации видеопоследовательностей. Подробнее возможность аппаратной обработки видеопоследовательностей будет рассмотрена ниже.

Поскольку аппаратные реализации алгоритмов «медиана» и «свертка» выполнены в виде комбинационных схем, оценку их временных характеристик можно произвести путем вычисления времени прохождения сигнала от входа до выхода комбинационной схемы, соответствующее времени обработки исходной матрицы **M**. Для этой цели были использованы материалы отчетов о результатах выполнения проектов в системе проектирования Xilinx, а также справочные материалы о характеристиках микросхем серий Spartan 3 и Virtex 4 [5–7]. Результаты представлены в табл. 5.

Таблица 5

Время выполнения алгоритмов (задержка прохождения сигнала через схему), нс

Серия и тип микросхемы	Алгоритм		Максимальная частота, МГц
	«медиана»	«свертка»	
Spartan 3 xc3s200-5ft256 xc3s400-5fg456	32,317 32,317	25,280 25,280	270
Virtex 4 xc4vlx15-12sf363 xc4vlx25-12sf363 xc4vlx40-12ff668	17,407 17,407 17,407	13,493 13,493 13,493	584

Следует учитывать, что ко времени выполнения алгоритма, указанному в табл. 5, нужно добавить еще время одного такта, необходимого для записи результатов обработки элементарной матрицы в выходной регистр.

Используя данные табл. 5, можно определить количество тактов, необходимое для выполнения алгоритма. Например, если выбрать микросхему серии Spartan 3, работающую на частоте 100 МГц (такт = 10 нс), то для обработки одной элементарной матрицы алгоритмом «медиана» потребуются 32,317 нс:  $10 \text{ нс} = 4 + 1 = 5$  тактов, а алгоритмом «свертка» – четыре такта (25,280 нс:  $10 \text{ нс} = 3 + 1$ ). Аналогично определяем, что при использовании микросхемы серии Virtex 4, работающей на частоте 100 МГц, для выполнения алгоритмов «медиана» и «свертка» потребуются три такта.

#### 4. Обработка видеопоследовательностей в режиме реального времени

На основании данных табл. 3 и 4 можно рассчитать возможности аппаратной обработки видеопоследовательностей в режиме реального времени. Поскольку затраты оборудования на обработку одной элементарной матрицы составляют небольшую часть объема микросхемы, можно загрузить в микросхему и одновременно обработать некоторую часть (блок) матрицы изображений. Загружая и обрабатывая поочередно блоки матрицы изображений, можно получить результат фильтрации всей матрицы.

Рассмотрим возможность обработки матрицы изображений размером 768 x 576 пикселей (байтов) с помощью алгоритма «медиана». Например, при обработке изображений с помощью микросхемы xc4vlx40-12ff668 можно одновременно загрузить блок матрицы, состоящий из 60 элементарных матриц, т. е. из трех строк длиной 60 элементов каждая. При этом будет задействовано  $1,5 \% \times 60 = 90 \%$  оборудования микросхемы. Оставшаяся часть аппаратуры может быть использована для организации операций ввода данных и вывода результатов обработки.

Рассматриваемый блок данных содержит  $3 \times 60 = 180$  элементов (байтов). В результате обработки такого блока будут получены новые значения 60 элементов средней строки. Затраты времени на обработку такого блока будут составлять: 180 тактов для ввода данных из ПЭВМ; 60 тактов для вывода результатов обработки; 3 такта для обработки данных; всего 243 такта. Если принять рабочую частоту микросхемы в 100 МГц (Clock, такт = 10 нс), то на обработку блока данных потребуется  $243 \text{ такта} \times 10 \text{ нс} = 2430 \text{ нс} = 2,43 \text{ мкс}$ .

Для обработки одной строки рассматриваемой матрицы изображений необходимо обработать  $768 : 60 = 12,8$ , т. е. 13, блоков данных, а для обработки всей матрицы  $13 \times 574 = 7462$  блока (из 576 строк матрицы изображений значения первой и последней строк остаются без изменения). Следовательно, обработка матрицы размером 768 x 576 пикселей будет произведена за время, равное  $2,43 \text{ мкс} \times 7462 \text{ блока} = 18132,66 \text{ мкс} = 18,2 \text{ мс}$ , что значительно меньше 40 мс, через которые происходит смена кадров изображений при обработке видеопоследовательностей.

### Заключение

Представленные результаты объемно-временных характеристик аппаратной реализации двух алгоритмов фильтрации изображений позволяют сформулировать ее преимущества перед программной реализацией:

1. Обеспечивается высокая скорость выполнения алгоритмов, позволяющая производить обработку видеопоследовательностей в реальном масштабе времени.
2. Максимальный размер обрабатываемой матрицы изображений определяется только логическими возможностями используемой микросхемы, что позволяет выбирать наиболее приемлемый по технико-экономическим показателям вариант аппаратной реализации алгоритмов.
3. Возможности современных микросхем и приведенные выше характеристики аппаратной реализации рассматриваемых алгоритмов фильтрации изображений позволяют производить одновременно обработку изображения двумя алгоритмами фильтрации в реальном масштабе времени.
4. В случае реализации алгоритма с помощью комбинационной схемы появляется возможность применения более дешевых низкоскоростных микросхем FPGA.

### Список литературы

1. Программно-аппаратный комплекс слежения в реальном масштабе времени за движущимися объектами / Б.А. Залесский [и др.] // Информационные технологии программы Союзного государства «Триада». Основные результаты и перспективы. – Минск : ОИПИ НАН Беларуси, 2010. – С. 262–269.
2. Кравчонок, А.И. Алгоритмы медианной фильтрации с окном 3x3 на основе неполных сортирующих сетей / А.И. Кравчонок // Информатика. – 2009. – № 1(21). – С. 91–102.
3. Кравчонок, А.И. Алгоритмы медианной фильтрации с окном 3x3 при помощи MMX- и SSE2-инструкций процессоров семейства x86 / А.И. Кравчонок // Информатика. – 2010. – № 1(25). – С. 41–57.
4. Залесский, Б.А. Отслеживание динамических объектов и их распознавание с помощью графовых алгоритмов / Б.А. Залесский, А.И. Кравчонок // Информатика. – 2006. – № 2 (10). – С. 17–26.
5. Кузелин, М.О. Современные семейства ПЛИС фирмы Xilinx : справочное пособие / М.О. Кузелин, Д.А. Кнышев, В.Ю. Зотов. – М. : Горячая линия – Телеком, 2004. – 440 с.
6. Spartan-3. FPGA Family Data Sheet: DC and Switching Characteristics [Electronic resource]. – Mode of access : <http://www.xilinx.com/support/index.htm>. – Date of access : 23.10.2013.
7. Virtex-4. FPGA Family Data Sheet: DC and Switching Characteristics [Electronic resource]. – Mode of access : [www.xilinx.com/support/documentation/data\\_sheets/ds302.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds302.pdf). – Date of access : 23.10.2013.

Поступила 23.10.2013

*Объединенный институт проблем  
информатики НАН Беларуси,  
Минск, Сурганова, 6  
e-mail: alexpolja@tut.by*

**A.S. Poljakov, V.E. Samsonov**

### EVALUATION OF CHARACTERISTICS OF HARDWARE IMPLEMENTATION OF SOME IMAGE FILTERING ALGORITHMS

Processing speed and hardware requirements of two popular image filtering algorithms in hardware implementation based on microchips of FPGA type are provided. A possibility of using algorithms for processing of standard frames of video sequences is considered.

## ЗАЩИТА ИНФОРМАЦИИ

УДК 004.421.5-519.217.2

И.Б. Бережной, Ю.С. Харин

ВЕРОЯТНОСТНАЯ МОДЕЛЬ ДИНАМИКИ ПАМЯТИ  
КРИПТОГРАФИЧЕСКИХ ГЕНЕРАТОРОВ МАКЛАРЕНА – МАРСАЛЬИ

*Рассматривается класс криптографических генераторов псевдослучайных последовательностей Макларена – Марсальи. Исследуется вероятностная модель динамики памяти генераторов Макларена – Марсальи. Показываются результаты компьютерных экспериментов.*

## Введение

В криптографической защите информации важную роль играют последовательности случайных чисел [1, 2]. На практике для их генерации в основном используют программные генераторы, когда с помощью детерминированного алгоритма создается псевдослучайная последовательность, как можно больше похожая на случайную. В последнее время особое внимание уделяется генераторам с памятью. Так, в конкурсе eSTREAM, целью которого было создание европейских стандартов для поточных систем шифрования, два из четырех финалистов в профиле «поточные шифры для программного применения с большой пропускной способностью» являются такими генераторами [3]. Современные генераторы в подавляющем большинстве являются комплексами комбинированных определенным образом модулей, представляющих собой различные криптографические примитивы [1, 2]. Данные примитивы для эффективной работы всего генератора должны иметь достаточно простую структуру и при этом обладать хорошими криптографическими свойствами. Одним из таких широко известных примитивов является класс генераторов Макларена – Марсальи [4], который сочетает свойство простоты реализации с достаточно высоким качеством выходной последовательности [5]. Интерес к данному классу генераторов не угасает: один из участников конкурса eSTREAM – поточный шифр Ямб [6] – содержит модуль аналогичной структуры.

Ранее класс генераторов Макларена – Марсальи изучался авторами в статье [7]. Для него получена формула для периода выходной последовательности, уточняющая приведенную в [2] формулу, доказаны некоторые вероятностные свойства, на основании которых построены формулы и оценки, характеризующие влияние генератора на некоторые статистические характеристики выходной последовательности. В настоящей статье продолжено исследование класса генераторов Макларена – Марсальи: исследуется вероятностная модель динамики памяти, найдены некоторые свойства и вероятностные характеристики предельного распределения памяти, приведены результаты компьютерных экспериментов.

## 1. Математическая модель генераторов Макларена – Марсальи

Генераторы Макларена – Марсальи, исследуемые в данной статье, имеют структуру, представленную на рисунке. Любой генератор данного семейства состоит из модифицируемой памяти  $X$  размера  $L$  и двух простейших генераторов псевдослучайных последовательностей  $G_1$  и  $G_2$ . Генератор  $G_1$  порождает «заполняющую» (исходную) последовательность  $\{\xi_t\}$  над некоторым конечным множеством  $V$ , генератор  $G_2$  – «управляющую» последовательность  $\{\eta_t\}$  над множеством  $A = \{0, 1, \dots, L - 1\}$ . Результирующей (выходной) последовательностью является последовательность  $\{y_t\}$  над  $V$ . Имеется некоторое начальное заполнение памяти – вектор-столбец  $X_0 = (x_{0,1}, x_{0,2}, \dots, x_{0,L})^T \in V^L$  ( $T$  – знак транспонирования).

Определим функцию  $Y = \chi(X, v, a)$ , где  $X, Y \in V^L$ ,  $v \in V$ ,  $a \in A$ , следующим образом:

$$Y = (y_1, y_2, \dots, y_L), \quad y_i = \begin{cases} x_i, & \text{если } i \neq a; \\ v, & \text{если } i = a. \end{cases} \quad (1)$$

Другими словами, функция  $\chi(X, v, a)$  заменяет в векторе  $X$  значение элемента с номером  $a$  на значение  $v$ .

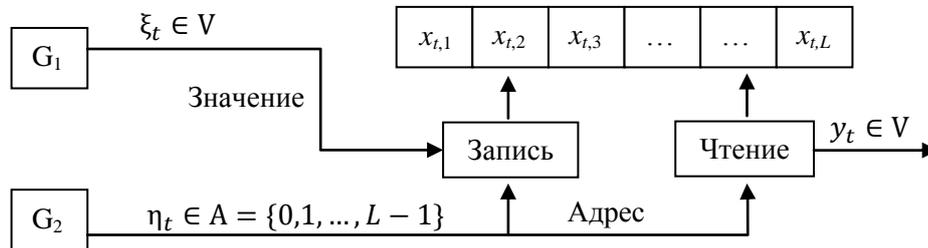


Схема генератора Макларена – Марсальи

Пусть  $\{X_t: t = 1, 2, \dots\}$  – последовательность состояний памяти  $X$ ;  $X_0$  – некоторое начальное заполнение памяти. Тогда динамика генератора на тактах  $t = 1, 2, \dots$  опишется формулами

$$\begin{aligned} y_t &= x_{t-1, \xi_t}; \\ X_t &= \chi(X_{t-1}, \xi_t, \eta_t). \end{aligned} \quad (2)$$

Таким образом, на каждом такте в выходную последовательность считывается элемент из памяти  $X$  по адресу, определяемому генератором  $G_2$ , затем по этому адресу в память заносится новый элемент из последовательности, порождаемой генератором  $G_1$ .

## 2. Вероятностная модель динамики памяти

Пусть  $\{\xi_t\}$  и  $\{\eta_t\}$  – определенные на некотором вероятностном пространстве  $(\Omega, \mathcal{F}, \mathcal{P})$  последовательности независимых в совокупности случайных величин с распределениями вероятностей

$$P\{\xi_t = i\} = \lambda_i \neq 0, \quad i \in V, \quad \sum_{i \in V} \lambda_i = 1; \quad P\{\eta_t = j\} = \gamma_j \neq 0, \quad j \in A, \quad \sum_{j \in A} \gamma_j = 1. \quad (3)$$

**Теорема 1.** Пусть  $\{\xi_t\}$  и  $\{\eta_t\}$  — определенные на некотором вероятностном пространстве  $(\Omega, \mathcal{F}, \mathcal{P})$  последовательности независимых в совокупности случайных величин с распределениями вероятностей (3). Тогда последовательность состояний памяти  $\{X_t\}$  представляет собой регулярную цепь Маркова первого порядка с пространством состояний  $V^L$  и матрицей переходных вероятностей  $P = (p_{J,K})$ :

$$p_{J,K} = P\{X_{t+1} = K | X_t = J\} = \sum_{i \in V} \sum_{j \in A} \lambda_i \gamma_j \delta_{K, \chi(J, i, j)}, \quad J, K \in V^L, \quad (4)$$

где  $\delta_{J,K} = \begin{cases} 1, & J = K \\ 0, & J \neq K \end{cases}$  – символ Кронекера.

Доказательство. В силу формулы (2) и отсутствия зависимости между элементами последовательностей  $\{\xi_t\}$  и  $\{\eta_t\}$  последовательность  $\{X_t\}$  представляет собой цепь Маркова первого порядка, так как выполняется марковское свойство [8]:

$$\begin{aligned} &P\{X_t = W_t | X_{t-1} = W_{t-1}, X_{t-2} = W_{t-2}, \dots, X_0 = W_0\} = \\ &= P\{\chi(X_{t-1}, \xi_t, \eta_t) = W_t | X_{t-1} = W_{t-1}, X_{t-2} = W_{t-2}, \dots, X_0 = W_0\} = \\ &= P\{\chi(W_{t-1}, \xi_t, \eta_t) = W_t | X_{t-1} = W_{t-1}\} = P\{X_t = W_t | X_{t-1} = W_{t-1}\}, \end{aligned}$$

где  $W_0, W_1, \dots, W_t \in V^L, t \geq 1$ .

На основании формулы (1), а также согласно формуле полной вероятности и в силу отсутствия зависимости между элементами последовательностей  $\{\xi_t\}$  и  $\{\eta_t\}$  переходная вероятность находится из выражения

$$\begin{aligned} P\{X_t = K | X_{t-1} = J\} &= \sum_{i \in V} \sum_{j \in A} P\{\xi_t = i, \eta_t = j\} P\{X_t = K | X_{t-1} = J, \xi_t = i, \eta_t = j\} \\ &= \sum_{i \in V} \sum_{j \in A} \lambda_i \gamma_j P\{X_t = K | X_t = \chi(X_{t-1}, i, j)\} = \sum_{i \in V} \sum_{j \in A} \lambda_i \gamma_j \delta_{K, \chi(i, j)}. \end{aligned}$$

Так как  $P\{X_t = K | X_{t-1} = J\}$  не зависит от  $t$ , то  $\{X_t\}$  – однородная цепь Маркова. Для ее регулярности необходимо и достаточно существование такого  $n$ , что  $P^n$  не содержит нулевых элементов [8]. Рассмотрим матрицу  $P^L = (p_{J,K}^{(L)})$ . Допустим, что существуют такие  $U, W \in V^L$ , что  $p_{U,W}^{(L)} = 0$ . Построим последовательность векторов  $\{U_0, U_1, U_2, \dots, U_L\}$  следующим образом:

$$U_0 = U, U_1 = \chi(U_0, w_1, 1), U_2 = \chi(U_1, w_2, 2), \dots, U_L = \chi(U_{L-1}, w_L, L) = W,$$

где  $W = (w_1, w_2, \dots, w_L)^T$ . Для последовательности  $\{U_i\}$  по построению справедливо соотношение  $p_{U_i, U_{i+1}} > 0, i = 0, 1, \dots, L-1$ .

Заметим, что в силу обобщенной формулы умножения вероятностей для однородной цепи Маркова первого порядка  $\{X_t\}$  справедлива следующая цепочка равенств:

$$\begin{aligned} &P\{X_t = U_0, X_{t+1} = U_1, \dots, X_{t+L} = U_L\} = \\ &= P\{X_t = U_0, X_{t+1} = U_1, \dots, X_{t+L-1} = U_{L-1}\} \times \\ &\times P\{X_{t+L} = U_L | X_t = U_0, X_{t+1} = U_1, \dots, X_{t+L-1} = U_{L-1}\} = \\ &= P\{X_t = U_0, X_{t+1} = U_1, \dots, X_{t+L-1} = U_{L-1}\} \times P\{X_{t+L} = U_L | X_{t+L-1} = U_{L-1}\} = \\ &= P\{X_t = U_0, X_{t+1} = U_1, \dots, X_{t+L-1} = U_{L-1}\} \times p_{U_{L-1}, U_L} = \dots = \\ &= P\{X_t = U_0\} \cdot p_{U_0, U_1} \cdot p_{U_1, U_2} \cdot \dots \cdot p_{U_{L-1}, U_L}. \end{aligned}$$

Тогда согласно уравнению Колмогорова – Чепмена и свойствам условной и совместной вероятностей справедлива цепочка неравенств

$$\begin{aligned} p_{U,W}^{(L)} &= P\{X_{t+L} = W | X_t = U\} = \frac{P\{X_t = U, X_{t+L} = W\}}{P\{X_t = U\}} \geq \\ &\geq \frac{P\{X_t = U_0, X_{t+1} = U_1, \dots, X_{t+L} = U_L\}}{P\{X_t = U_0\}} = p_{U_0, U_1} \cdot p_{U_1, U_2} \cdot \dots \cdot p_{U_{L-1}, U_L} > 0. \end{aligned}$$

Таким образом, получено противоречие. Следовательно, в матрице  $P^L$  нет нулевых элементов, т. е.  $\{X_t\}$  – регулярная цепь Маркова. ■

Обозначим через  $H(U, W) = \sum_{i=1}^L (1 - \delta_{u_i, w_i})$  количество несовпадающих элементов векторов  $U = (u_1, u_2, \dots, u_L), W = (w_1, w_2, \dots, w_L) \in V^L$ .

**Следствие 1.** При выполнении условий (3) элементы матрицы  $P$  переходных вероятностей имеют следующий вид:

$$p_{J,K} = \begin{cases} \sum_{i=1}^L \gamma_i \lambda_{j_i}, & \text{если } H(J, K) = 0, \text{ т. е. } K = J = (j_1, j_2, \dots, j_L); \\ \gamma_i \lambda_s, & \text{если } H(J, K) = 1, \text{ т. е. } K = \chi(J, s, i), s \neq j_i; \\ 0, & \text{если } H(J, K) \geq 2. \end{cases} \quad (5)$$

Доказательство непосредственно следует из формулы (4). ■

**Теорема 2.** Пусть  $\{\xi_t\}$  и  $\{\eta_t\}$  – определенные на некотором вероятностном пространстве  $(\Omega, F, P)$  последовательности независимых в совокупности случайных величин с распределениями вероятностей (3). Тогда стационарное распределение вероятностей  $\alpha$  цепи Маркова  $\{X_t\}$  существует и имеет мультипликативный вид

$$\alpha_K = \lambda_{k_1} \cdot \lambda_{k_2} \cdot \dots \cdot \lambda_{k_L}, K = (k_1, k_2, \dots, k_L) \in V^L. \quad (6)$$

Доказательство. Формулировка теоремы корректна, так как если  $\{\xi_t\}$  и  $\{\eta_t\}$  – последовательности независимых в совокупности случайных величин, то согласно теореме 1  $\{X_t\}$  – регулярная цепь Маркова, а для регулярных цепей Маркова всегда существует стационарное распределение, причем оно – единственное вероятностное решение уравнения  $\alpha^T P = \alpha^T$  [8]. Формула (6) описывает корректное распределение вероятностей, так как в силу (3) выполняется условие нормировки  $\sum_{K \in V^L} \lambda_{k_1} \cdot \lambda_{k_2} \cdot \dots \cdot \lambda_{k_L} \equiv 1$ .

В данном случае уравнение  $\alpha^T P = \alpha^T$  можно согласно следствию 1 представить в виде

$$\begin{aligned} \forall K \in V^L \quad \alpha_K &= \sum_J \alpha_J p_{J,K} = \alpha_K \sum_{i=1}^L \gamma_i \lambda_{k_i} + \sum_{J: \exists i: K=\chi(J, k_i, i), j_i \neq k_i} \gamma_i \lambda_{k_i} \alpha_J = \\ &= \alpha_K \sum_{i=1}^L \gamma_i \lambda_{k_i} + \sum_{i=1}^L (\gamma_i \lambda_{k_i} \sum_{J: K=\chi(J, k_i, i), j_i \neq k_i} \alpha_J), \end{aligned} \quad (7)$$

где  $K = (k_1, k_2, \dots, k_L)$ ,  $J = (j_1, j_2, \dots, j_L) \in V^L$ .

В силу формул (1) и (3) для распределения вероятностей  $\alpha$ , определяемого формулой (6), справедливы равенства

$$\begin{aligned} \sum_{J: K=\chi(J, k_i, i), j_i \neq k_i} \alpha_J &= \sum_{J: K=\chi(J, k_i, i), j_i \neq k_i} \lambda_{j_1} \cdot \lambda_{j_2} \cdot \dots \cdot \lambda_{j_L} = \sum_{S \in V, S \neq k_i} \lambda_{k_1} \cdot \dots \cdot \lambda_{k_{i-1}} \cdot \lambda_S \cdot \lambda_{k_{i+1}} \cdot \dots \cdot \lambda_{k_L} = \\ &= \lambda_{k_1} \cdot \dots \cdot \lambda_{k_{i-1}} \cdot \lambda_{k_{i+1}} \cdot \dots \cdot \lambda_{k_L} \cdot \sum_{S \in V, S \neq k_i} \lambda_S = \lambda_{k_1} \cdot \dots \cdot \lambda_{k_{i-1}} \cdot (1 - \lambda_{k_i}) \cdot \lambda_{k_{i+1}} \cdot \dots \cdot \lambda_{k_L}. \end{aligned}$$

Отсюда для любого  $\alpha_K$ , определяемого формулой (6), получаем следующие равенства:

$$\begin{aligned} \alpha_K \sum_{i=1}^L \gamma_i \lambda_{k_i} + \sum_{i=1}^L \left( \gamma_i \lambda_{k_i} \sum_{J: K=\chi(J, k_i, i), j_i \neq k_i} \alpha_J \right) &= \alpha_K \sum_{i=1}^L \gamma_i \lambda_{k_i} + \sum_{i=1}^L \left( \gamma_i \lambda_{k_i} (1 - \lambda_{k_i}) \prod_{j=1, j \neq i}^L \lambda_{k_j} \right) = \\ &= \alpha_K \sum_{i=1}^L \gamma_i \lambda_{k_i} + \sum_{i=1}^L \gamma_i (1 - \lambda_{k_i}) \alpha_K = \alpha_K \sum_{i=1}^L \gamma_i = \alpha_K. \end{aligned}$$

Таким образом, если распределение вероятностей  $\alpha$  определяется формулой (6), для него выполняются соотношения (7), а значит,  $\alpha$  является решением уравнения  $\alpha^T P = \alpha^T$ . Так как данное уравнение имеет единственное вероятностное решение, то  $\alpha$  совпадает со стационарным распределением вероятностей цепи Маркова  $\{X_t\}$ . ■

**Следствие 2.** Стационарное распределение вероятностей (6) состояний памяти  $\{X_t\}$  не зависит от распределения вероятностей  $\{\gamma_i\}$  управляющей последовательности  $\{\eta_t\}$ .

Доказательство непосредственно следует из формулы (6). ■

Следствие 2 показывает, что при моделировании достаточно продолжительной динамики генератора нет необходимости жестко контролировать вероятностные характеристики управляющей последовательности: даже если будут искажения в распределении вероятностей ее элементов, на частоту встречаемости состояний памяти это не повлияет.

**Следствие 3.** Пусть выполняются условия теоремы 2. Стационарное распределение вероятностей  $\alpha$  состояний памяти  $\{X_t\}$  является равномерным (т. е.  $\forall K \in V^L \quad \alpha_K = \frac{1}{|V|^L}$ ) тогда и только тогда, когда распределение вероятностей  $\{\lambda_i\}$  элементов заполняющей последовательности  $\{\xi_t\}$  равномерное ( $\lambda_i \equiv \frac{1}{|V|}$ ).

Доказательство. Согласно теореме 2 стационарное распределение вероятностей  $\alpha$  цепи Маркова  $\{X_t\}$  существует и определяется формулой (6).

Пусть  $\lambda_i = \frac{1}{|V|}$ ,  $i \in V$ . Тогда в силу (6) выполняется требуемое соотношение

$$\alpha_K = \lambda_{k_1} \cdot \lambda_{k_2} \cdot \dots \cdot \lambda_{k_L} = \frac{1}{|V|^L} = \text{const}, K = (k_1, k_2, \dots, k_L) \in V^L.$$

Пусть  $\alpha$  – равномерное стационарное распределение вероятностей. Тогда выполняется условие (7), которое можно представить в следующем виде:

$$\begin{aligned} 1 &= \sum_{i=1}^L \gamma_i \lambda_{k_i} + \sum_{i=1}^L \left( \gamma_i \lambda_{k_i} \sum_{j: K=\chi(j, k_i, i), j_i \neq k_i} 1 \right) = \\ &= \sum_{i=1}^L \gamma_i \lambda_{k_i} + \sum_{i=1}^L (\gamma_i \lambda_{k_i} (|V| - 1)) = \sum_{i=1}^L (\gamma_i \lambda_{k_i} |V|). \end{aligned} \quad (8)$$

Условие (8) должно выполняться для любого  $K \in V^L$ . Выберем два вектора  $U = (u_1, u_2, \dots, u_L)$ ,  $W = (w_1, w_2, \dots, w_L) \in V^L$ , такие, что  $\forall i \in A \setminus \{s\} u_i = w_i$ ,  $u_s \neq w_s$ , т. е. отличающиеся в одном элементе под номером  $s$ . Тогда для данных векторов в силу (8) выполняется соотношение

$$\gamma_s \lambda_{u_s} |V| = 1 - \sum_{i=1, i \neq s}^L (\gamma_i \lambda_{u_i} |V|) = 1 - \sum_{i=1, i \neq s}^L (\gamma_i \lambda_{w_i} |V|) = \gamma_s \lambda_{w_s} |V|.$$

Из этого следует, что  $\lambda_{u_s} = \lambda_{w_s}$ . Так как  $u_s$  может принимать любое значение из множества  $V \setminus \{w_s\}$ , получаем искомое требование

$$\lambda_i = \text{const} = \frac{1}{|V|}, \quad i \in V. \blacksquare$$

Следствие 3 определяет, что отклонение стационарного распределения вероятностей последовательности  $\{X_t\}$  от равномерного свидетельствует о наличии искажений в распределении вероятностей последовательности  $\{\xi_t\}$ . Поэтому, если имеется возможность оценить стационарное распределение вероятностей  $\{X_t\}$ , можно получить сведения о параметрах исходной последовательности  $\{\xi_t\}$ .

Введем следующие обозначения:

$u_K^{(n)}$  – частота вектора  $K \in V^L$  в последовательности состояний памяти  $\{X_t\}$ ,  $t \in \{1, 2, \dots, n\}$ :

$$u_K^{(n)} = \frac{1}{n} \sum_{i=1}^n \delta_{X_i, K};$$

$f_K$  – время первого возврата в состояние, равное  $K$ :

$$f_K = n \in \mathbb{N}: X_n = X_0 = K, \forall i = \overline{1, n-1} X_i \neq K.$$

**Следствие 4.** Пусть выполняются условия теоремы 2. Тогда верны следующие утверждения:

1) вне зависимости от начального распределения математическое ожидание  $u_K^{(n)}$  при  $n \rightarrow \infty$  удовлетворяет предельному соотношению

$$\mathbf{E} \left\{ u_K^{(n)} \right\} \rightarrow \lambda_{k_1} \cdot \lambda_{k_2} \cdot \dots \cdot \lambda_{k_L}, \quad K = (k_1, k_2, \dots, k_L) \in V^L; \quad (9)$$

2) математическое ожидание времени возврата в исходное состояние памяти  $K$  определяется формулой

$$\mathbf{E} \{ f_K \} = \frac{1}{\lambda_{k_1} \cdot \lambda_{k_2} \cdot \dots \cdot \lambda_{k_L}}. \quad (10)$$

Доказательство. Согласно [8] для регулярных цепей Маркова выполняются соотношения

$$\mathbf{E} \left\{ u_K^{(n)} \right\} \xrightarrow{n \rightarrow \infty} \alpha_K, \quad \mathbf{E} \{ f_K \} = \frac{1}{\alpha_K}.$$

Подставляя в данные соотношения формулу (6), получаем искомые утверждения. ■

Следствие 4 описывает частотные характеристики динамики памяти генератора. С их помощью можно оценить, как часто элемент выходной последовательности выбирался из заданного состояния памяти и имел соответствующее условное распределение вероятностей.

**Теорема 3.** Пусть  $\{\xi_t\}$  и  $\{\eta_t\}$  – определенные на некотором вероятностном пространстве  $(\Omega, \mathcal{F}, P)$  последовательности случайных величин с распределениями вероятностей (3), причем элементы  $\{\eta_t\}$  независимы в совокупности, а  $\{\xi_t\}$  представляет собой цепь Маркова первого порядка с матрицей переходных вероятностей  $Q = (q_{u,v})$ ,  $u, v \in V$ . Тогда случайная последовательность состояний памяти  $\{X_t\}$  представляет собой однородную цепь Маркова второго порядка с матрицей переходных вероятностей  $P = (p_{I,J,K})$ :

$$\begin{aligned} p_{I,J,K} &= P\{X_{t+1} = K | X_t = J, X_{t-1} = I\} = \\ &= \frac{\sum_{i \in V} \sum_{j \in A} (\lambda_i \gamma_j \delta_{J, \chi(I,i,j)} \sum_{l \in V} \sum_{s \in A} q_{j,l} \gamma_s \delta_{K, \chi(l,s)})}{\sum_{i \in V} \sum_{j \in A} \lambda_i \gamma_j \delta_{J, \chi(I,i,j)}}, \quad I, J, K \in V^L. \end{aligned} \quad (11)$$

Доказательство. Определим функцию изменяющего элемента  $v = \psi(Y, X, a) \in V$ , где  $Y, X \in V^L, a \in A$ , следующим образом:

$$v = \psi(Y, X, a) \Leftrightarrow Y = \chi(X, v, a).$$

В силу формулы (2) и представления  $\{\xi_t\}$  как цепи Маркова первого порядка для последовательности  $\{X_t\}$  справедливы соотношения

$$\begin{aligned} P\{X_t = U\} &= P\{\chi(X_{t-1}, \xi_t, \eta_t) = U\} = f(U, X_{t-1}, \eta_t, \xi_t) = \\ &= g(U, X_{t-1}, \eta_t, \xi_{t-1}) = g(U, X_{t-1}, \eta_t, \psi(X_{t-1}, X_{t-2}, \eta_{t-1})) = \\ &= h(U, X_{t-1}, X_{t-2}, \eta_t, \eta_{t-1}), \end{aligned} \quad (12)$$

где  $f, g, h$  – некоторые вероятностные функции.

В связи с тем что  $\{\eta_t\}$  – последовательность независимых в совокупности одинаково распределенных случайных величин, из (12) следует обобщенное марковское свойство [8]:

$$P\{X_t = W_t | X_{t-1} = W_{t-1}, X_{t-2} = W_{t-2}, \dots, X_0 = W_0\} = P\{X_t = W_t | X_{t-1} = W_{t-1}, X_{t-2} = W_{t-2}\},$$

где  $W_0, W_1, \dots, W_t \in V^L, t \geq 2$ . Оно доказывает, что  $\{X_t\}$  представляет собой цепь Маркова второго порядка.

Для  $\{X_t\}$  на основании формулы (2), а также согласно формуле полной вероятности и в силу отсутствия попарной зависимости между элементами последовательностей  $\{\xi_t\}$  и  $\{\eta_t\}$  выполняется следующее соотношение:

$$\begin{aligned} &P\{X_t = K | X_{t-1} = J\} = \\ &= \sum_{i \in V} \sum_{j \in A} P\{\xi_t = i, \eta_t = j\} P\{X_t = K | X_{t-1} = J, \xi_t = i, \eta_t = j\} = \\ &= \sum_{i \in V} \sum_{j \in A} \lambda_i \gamma_j P\{X_t = K | X_t = \chi(X_{t-1}, i, j)\} = \sum_{i \in V} \sum_{j \in A} \lambda_i \gamma_j \delta_{K, \chi(I,i,j)}. \end{aligned} \quad (13)$$

Переходная вероятность на основании формул (2) и (13), а также согласно формулам полной и условной вероятностей и в силу свойств последовательностей  $\{\xi_t\}$  и  $\{\eta_t\}$  определяется выражением

$$\begin{aligned} P\{X_t = U | X_{t-1} = W, X_{t-2} = Z\} &= \frac{P\{X_t = U, X_{t-1} = W, X_{t-2} = Z\}}{P\{X_{t-1} = W, X_{t-2} = Z\}} = \\ &= \frac{P\{X_t = U, X_{t-1} = W | X_{t-2} = Z\}}{P\{X_{t-1} = W | X_{t-2} = Z\}} = \end{aligned}$$

$$\begin{aligned}
 &= \frac{\sum_{j \in V} \sum_{k \in A} \lambda_j \cdot \gamma_k \cdot P\{X_t = U, \chi(Z, j, k) = W \mid X_{t-2} = Z, \xi_{t-1} = j, \eta_{t-1} = k\}}{\sum_{i \in V} \sum_{j \in A} \lambda_i \gamma_j \delta_{W, \chi(Z, i, j)}} = \\
 &= \frac{1}{\sum_{i \in V} \sum_{j \in A} \lambda_i \gamma_j \delta_{W, \chi(Z, i, j)}} \sum_{j \in V} \sum_{k \in A} (\lambda_j \gamma_k \delta_{W, \chi(Z, j, k)} P\{\chi(W, \xi_t, \eta_t) = U \mid \xi_{t-1} = j\}) = \\
 &= \frac{\sum_{j \in V} \sum_{k \in A} (\lambda_j \gamma_k \delta_{W, \chi(Z, j, k)} \sum_{l \in V} P\{\chi(W, \xi_t, \eta_t) = U \mid \xi_t = l\} P\{\xi_t = l \mid \xi_{t-1} = j\})}{\sum_{i \in V} \sum_{j \in A} \lambda_i \gamma_j \delta_{W, \chi(Z, i, j)}} = \\
 &= \frac{1}{\sum_{i \in V} \sum_{j \in A} \lambda_i \gamma_j \delta_{W, \chi(Z, i, j)}} \sum_{j \in V} \sum_{k \in A} \left( \lambda_j \gamma_k \delta_{W, \chi(Z, j, k)} \sum_{l \in V} P\{\chi(W, l, \eta_t) = U\} \cdot q_{j,l} \right) = \\
 &= \frac{1}{\sum_{i \in V} \sum_{j \in A} \lambda_i \gamma_j \delta_{W, \chi(Z, i, j)}} \sum_{j \in V} \sum_{k \in A} \left( \lambda_j \gamma_k \delta_{W, \chi(Z, j, k)} \sum_{l \in V} \left( q_{j,l} \cdot \sum_{s \in A} \gamma_s P\{\chi(W, l, s) = U\} \right) \right) = \\
 &= \frac{1}{\sum_{i \in V} \sum_{j \in A} \lambda_i \gamma_j \delta_{W, \chi(Z, i, j)}} \sum_{j \in V} \sum_{k \in A} \left( \lambda_j \gamma_k \delta_{W, \chi(Z, j, k)} \sum_{l \in V} \sum_{s \in A} q_{j,l} \gamma_s \delta_{U, \chi(W, l, s)} \right).
 \end{aligned}$$

Так как  $P\{X_t = U \mid X_{t-1} = W, X_{t-2} = Z\}$  не зависит от  $t$ , то  $\{X_t\}$  – однородная цепь Маркова второго порядка. ■

**Следствие 5.** При выполнении условий теоремы 3 элементы матрицы переходных вероятностей  $P = (p_{I,J,K})$  имеют следующий вид:

$$p_{I,J,K} = \begin{cases} \frac{\sum_{s \in A} (\gamma_s \lambda_{k_s} \sum_{r \in A} (\gamma_r q_{k_s, k_r}))}{\sum_{s \in A} \gamma_s \lambda_{k_s}}, & \text{если } K = J = I; \\ \frac{\sum_{s \in A} (\gamma_s \lambda_{j_s} \cdot \gamma_r q_{j_s, k_r})}{\sum_{s \in A} \gamma_s \lambda_{j_s}}, & \text{если } K = \chi(J, k_r, r), k_r \neq j_r, J = I; \\ \sum_{r \in A} (\gamma_r q_{j_s, k_r}), & \text{если } K = J = \chi(I, j_s, s), j_s \neq i_s; \\ \gamma_r \cdot q_{j_s, k_r}, & \text{если } K = \chi(J, k_r, r), k_r \neq j_r, J = \chi(I, j_s, s), j_s \neq i_s; \\ 0, & \text{если } H(I, J) \geq 2 \text{ или } H(J, K) \geq 2, \end{cases} \quad (14)$$

где  $K = (k_1, k_2, \dots, k_L)$ ,  $J = (j_1, j_2, \dots, j_L)$ ,  $I = (i_1, i_2, \dots, i_L)$ .

Доказательство непосредственно следует из формулы (11). ■

**Следствие 6.** Если в условиях теоремы 3 одномерные распределения вероятностей  $\lambda$  и  $\gamma$  элементов последовательностей  $\{\xi_t\}$  и  $\{\eta_t\}$  равномерные, то элементы матрицы переходных вероятностей  $P = (p_{I,J,K})$  имеют следующий вид:

$$p_{I,J,K} = \begin{cases} \frac{1}{L^2} \sum_{s \in A} \sum_{r \in A} q_{k_s, k_r}, & \text{если } K = J = I; \\ \frac{1}{L^2} \sum_{s \in A} q_{j_s, k_r}, & \text{если } K = \chi(J, k_r, r), k_r \neq j_r, J = I; \\ \frac{1}{L} \sum_{r \in A} q_{j_s, k_r}, & \text{если } K = J = \chi(I, j_s, s), j_s \neq i_s; \\ \frac{1}{L} q_{j_s, k_r}, & \text{если } K = \chi(J, k_r, r), k_r \neq j_r, J = \chi(I, j_s, s), j_s \neq i_s; \\ 0, & \text{если } H(I, J) \geq 2 \text{ или } H(J, K) \geq 2. \end{cases} \quad (15)$$

Доказательство непосредственно следует из формулы (14). ■

Следствие 6 наглядно иллюстрирует связь между матрицей переходных вероятностей  $P = (p_{I,J,K})$  последовательности состояний памяти  $\{X_t\}$  и матрицей переходных вероятностей  $Q = (q_{u,v})$  заполняющей последовательности  $\{\xi_t\}$ . Формула (15) также позволяет проводить

оценку параметров заполняющей последовательности  $\{\xi_t\}$  по выборке из последовательности  $\{X_t\}$ .

### 3. Компьютерные эксперименты

Известно [8], что для регулярных цепей Маркова с матрицей переходных вероятностей  $P$  и стационарным распределением  $\alpha$  выполняются следующие асимптотические соотношения:

$$\alpha_K = \lim_{n \rightarrow \infty} P_{j,K}^{(n)},$$

причем  $\exists b, r: 0 < r < 1, P_{j,K}^{(n)} = \alpha_K + e_{j,K}^{(n)}$ , где  $|e_{j,K}^{(n)}| < b \cdot r^n$ .

Другими словами, имеет место сходимость матрицы  $P^n$  к предельной матрице  $M = E_L \alpha^T$  ( $E_L \in V^L$  – вектор-столбец, все  $L$  элементов которого равны 1) при  $n \rightarrow \infty$  с экспоненциальной скоростью. Поэтому стационарное распределение вероятностей с высокой точностью возможно вычислить путем возведения  $P$  в достаточно большую степень. В табл. 1 и 2 приведены результаты соответствующих компьютерных экспериментов. Для различной мощности алфавита  $V$  и размера памяти  $L$  определялись распределения вероятностей  $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_{|V|}\}$  элементов заполняющей последовательности и  $\gamma = \{\gamma_1, \gamma_2, \dots, \gamma_L\}$  управляющей последовательности. Затем на основании формулы (4) вычислялась матрица  $P^{2^{10}}$ , являющаяся весьма точной ( $|e_{j,K}^{(n)}| < O(r^{1024})$ ) аппроксимацией предельной матрицы  $M$ .

Таблица 1

Аппроксимация стационарного распределения  $\alpha$  при  $L = 2$  для различных  $\lambda$  и  $\gamma$

$\gamma$	$\lambda$		
	{0,1, 0,3, 0,6}	{0,2, 0,8}	{0,5, 0,5}
{0,25, 0,75}	{0,01, 0,03, ..., 0,36}	{0,04, 0,16, 0,16, 0,64}	$\alpha_K \equiv 0,25$
{0,5, 0,5}	{0,01, 0,03, ..., 0,36}	{0,04, 0,16, 0,16, 0,64}	$\alpha_K \equiv 0,25$
{0,001, 0,999}	{0,01, 0,03, ..., 0,36}	{0,04, 0,16, 0,16, 0,64}	$\alpha_K \equiv 0,25$

Таблица 2

Аппроксимация стационарного распределения  $\alpha$  при  $L = 3$  для различных  $\lambda$  и  $\gamma$

$\gamma$	$\lambda$		
	{0,1, 0,3, 0,6}	{0,2, 0,8}	{0,5, 0,5}
{0,01, 0,01, 0,98}	{0,001, 0,003, ..., 0,216}	{0,008, 0,032, ..., 0,128, 0,512}	$\alpha_K \equiv 0,125$
{0,2, 0,5, 0,3}	{0,001, 0,003, ..., 0,216}	{0,008, 0,032, ..., 0,128, 0,512}	$\alpha_K \equiv 0,125$
{0,33, 0,34, 0,33}	{0,001, 0,003, ..., 0,216}	{0,008, 0,032, ..., 0,128, 0,512}	$\alpha_K \equiv 0,125$

Данные результаты подтверждают теоретическую формулу (6) для стационарного распределения и его независимость от распределения вероятностей управляющей последовательности.

Для оценивания среднего времени возврата в заданное состояние  $X$  моделировалось поведение генератора Макларена – Марсальи с размером таблицы, равным  $L$ . Последовательности  $\{\xi_t\}$  и  $\{\eta_t\}$  строились как последовательности независимых в совокупности случайных величин с помощью встроенного в C++ генератора  $\text{rand}()$ , при этом элементы  $\{\eta_t\}$  имели равномерное распределение вероятностей, а  $\{\xi_t\}$  – заданное распределение вероятностей  $\lambda$ . Имитировались  $n$  реализаций динамики памяти с начальным заполнением памяти  $X_0 = K$  и высчитывалось среднее арифметическое времени возврата, являющееся несмещенной оценкой для  $E\{f_K\}$ . Результаты для состояния  $K = (0, 0, 0)$  при  $L = 3$  указаны в табл. 3. Видно, что статистики по выборкам стремятся с увеличением  $n$  к теоретическому значению, вычисленному согласно формуле (10). Для других значений  $K$  и  $L$  получены аналогичные результаты экспериментов.

Таблица 3

Среднее арифметическое времени возврата в состояние  $K = (0, 0, 0)$  при  $L = 3$  для различных  $\lambda$  и  $n$

$\lambda$	$n = 1000$	$n = 10\ 000$	$n = 100\ 000$	$n = 1\ 000\ 000$	Теоретическое значение по формуле (10)
{0,501, 0,499}	7,994	8,037	8,040	7,942	7,952
{0,51, 0,49}	6,927	7,209	7,503	7,517	7,538
{0,55, 0,45}	5,580	5,989	5,946	6,002	6,0105
{0,6, 0,4}	5,125	4,577	4,624	4,625	4,630
{0,3, 0,7}	37,548	37,190	36,963	37,102	37,037
{0,99, 0,01}	1,021	1,0295	1,031	1,03	1,0306

Также были проведены эксперименты по определению частот встречаемости вектора  $K$  в последовательности состояний памяти  $\{X_t\}$  длины  $n$ . Для этого моделировалось поведение генератора Макларена – Марсальи с размером таблицы, равным  $L$ . Последовательности  $\{\xi_t\}$  и  $\{\eta_t\}$  строились как последовательности независимых в совокупности случайных величин с помощью встроенного в C++ генератора `rand()`, при этом элементы  $\{\eta_t\}$  имели равномерное распределение вероятностей, а  $\{\xi_t\}$  – заданное распределение вероятностей  $\lambda$ . Проводились 100 экспериментов по оценке частоты встречаемости состояния  $K$  на выборке  $\{X_t\}$  длиной  $n$  и вычислялось среднее арифметическое. Результаты для состояния  $K = (0, 0, 0)$  при  $L = 3$  указаны в табл. 4. Видно, что выборочные частоты стремятся с увеличением  $n$  к теоретическому значению, вычисленному согласно формуле (9). Для других значений  $K$  и  $L$  получены аналогичные результаты экспериментов.

Таблица 4

Среднее арифметическое частот встречаемости состояния  $K = (0, 0, 0)$  при  $L=3$  для различных  $\lambda$  и  $n$

$\lambda$	$n = 1000$	$n = 10\ 000$	$n = 100\ 000$	$n = 1\ 000\ 000$	Теоретическое значение по формуле (9)
{0,495, 0,505}	0,119790	0,121143	0,121439	0,121271	0,121287
{0,49, 0,51}	0,117020	0,117806	0,117335	0,117614	0,117649
{0,45, 0,55}	0,094080	0,090817	0,091141	0,091124	0,091125
{0,4, 0,6}	0,064480	0,064663	0,064030	0,063984	0,064000
{0,3, 0,7}	0,026510	0,027010	0,026933	0,027030	0,027000
{0,1, 0,9}	0,000890	0,001000	0,001009	0,001003	0,001000
{0,05, 0,95}	0,00010	0,000127	0,000121	0,000123	0,000125

Для исследования случая, когда  $\{\xi_t\}$  является цепью Маркова первого порядка, проводились следующие эксперименты. Моделировалось поведение генератора Макларена – Марсальи с размером таблицы, равным  $L$ . Последовательность  $\{\eta_t\}$  строилась как последовательность независимых в совокупности равномерно распределенных на  $\{1, 2, \dots, L\}$  случайных величин с помощью встроенного в C++ генератора `rand()`. Последовательность  $\{\xi_t\}$  строилась как двоичная стационарная однородная цепь Маркова первого порядка с бистochasticкой матрицей переходных вероятностей  $Q = \begin{pmatrix} 1 - \epsilon & \epsilon \\ \epsilon & 1 - \epsilon \end{pmatrix}$  ( $\epsilon \in (0, 1)$  – задаваемый уровень) и стационарным начальным распределением вероятностей  $\pi = (0, 5, 0, 5)$ . Генерировалась выборка  $\{X_t\}$  длиной  $n$ , по которой оценивалась матрица переходных вероятностей для цепи Маркова второго порядка. Результаты для выборок длиной  $n = 10^4$  и  $n = 10^7$  при  $L = 2$  и уровне  $\epsilon = 0,1$  приведены в табл. 5, в которой в столбце Т указано соответствующее теоретическое значение, вычисленное согласно формуле (15).

Таблица 5

Выборочная матрица переходных вероятностей при  $L = 2$  и  $\epsilon = 0, 1$  для  $n = 10^4$  и  $10^7$ 

$X_{t-2}$	$X_{t-1}$	$X_t$											
		(0, 0)			(0, 1)			(1, 0)			(1, 1)		
		$10^4$	$10^7$	T									
(0, 0)	(0, 0)	0,897	0,900	0,9	0,049	0,049	0,05	0,053	0,049	0,05			
(0, 1)	(0, 0)	0,896	0,899	0,9	0,052	0,050	0,05	0,052	0,050	0,05			
(1, 0)	(0, 0)	0,857	0,899	0,9	0,066	0,050	0,05	0,076	0,049	0,05			
(1, 1)	(0, 0)												
(0, 0)	(0, 1)	0,034	0,050	0,05	0,554	0,498	0,5				0,412	0,451	0,45
(0, 1)	(0, 1)	0,264	0,249	0,25	0,500	0,500	0,5				0,235	0,249	0,25
(1, 0)	(0, 1)												
(1, 1)	(0, 1)	0,460	0,451	0,45	0,490	0,498	0,5				0,049	0,049	0,05
(0, 0)	(1, 0)	0,041	0,050	0,05				0,452	0,501	0,5	0,507	0,448	0,45
(0, 1)	(1, 0)												
(1, 0)	(1, 0)	0,243	0,249	0,25				0,533	0,500	0,5	0,223	0,250	0,25
(1, 1)	(1, 0)	0,412	0,451	0,45				0,511	0,498	0,5	0,077	0,050	0,05
(0, 0)	(1, 1)												
(0, 1)	(1, 1)				0,056	0,050	0,05	0,051	0,049	0,5	0,892	0,899	0,9
(1, 0)	(1, 1)				0,039	0,051	0,05	0,026	0,050	0,5	0,934	0,898	0,9
(1, 1)	(1, 1)				0,047	0,049	0,05	0,053	0,049	0,05	0,898	0,900	0,9

Видно, что статистики стремятся с увеличением  $n$  к теоретическому значению, вычисленному согласно формуле (15). Для других значений  $L$  и  $\epsilon$  получены аналогичные результаты экспериментов.

Полученные результаты иллюстрируют соответствие результатов компьютерного моделирования теоретическим результатам анализа динамики памяти генератора Макларена – Марсальи.

### Заключение

В статье построена и исследуется вероятностная модель динамики памяти генераторов Макларена – Марсальи. Доказан критерий равномерности стационарного распределения вероятностей памяти, получена общая формула для предельного распределения. На основании данной формулы найдены некоторые вероятностные характеристики динамики памяти. Также найдены формулы, позволяющие проводить оценку параметров заполняющей последовательности по выборке из последовательности состояний памяти. Результаты компьютерных экспериментов подтверждают теоретические результаты.

### Список литературы

1. Математические и компьютерные основы криптологии / Ю.С. Харин [и др.]. – Минск : Новое знание, 2003. – 382 с.
2. Основы криптографии / А.П. Алферов [и др.]. – М. : Гелиос АРВ, 2005. – 480 с.
3. eSTREAM: the ECRYPT Stream Cipher Project [Electronic resource]. – Mode of access : <http://www.ecrypt.eu.org/stream/project.html>. – Date of access : 10.04.2013.
4. MacLaren, M. Uniform Random Number Generators / M. MacLaren, G. Marsaglia // J. of the Association for Computing Machinery. – 1965. – Vol. 12(1). – P.83–89.
5. Кнут, Д. Искусство программирования. Получисленные алгоритмы = The Art of Computer Programming. Vol. 2. Seminumerical Algorithms. – 3-е изд. – М. : Вильямс, 2001. – Т. 2. – С. 45–47.
6. Starodubtzev, S.A. «Yamb», LAN Crypto Submission to the ECRYPT Stream Cipher Project / S.A. Starodubtzev, A.N. Lebedev, A.A. Volchkov [Electronic resource]. – Mode of access : <http://www.ecrypt.eu.org/stream/yamb.html>. – Date of access : 10.04.2013.

7. Бережной, И.Б. О периодичности и вероятностных свойствах генератора Макларена – Марсальи / И.Б. Бережной, Ю.С. Харин // Материалы XI Междунар. науч.-практ. конф. «Информационная безопасность–2010». – Таганрог, 2010. – Ч. 3. – С. 83–85.

8. Кемени, Дж.Дж. Конечные цепи Маркова / Дж.Дж. Кемени, Дж.Л. Снелл. – М. : Наука, 1970. – 272 с.

Поступила 9.07.2013

*НИИ прикладных проблем математики и информатики  
Белорусского государственного университета,  
Минск, пр. Независимости, 4  
e-mail: berezhnoy@tut.by*

**I.B. Berezhnoy, Yu.S. Kharin**

**PROBABILISTIC MODEL OF MEMORY DYNAMICS  
OF MACLAREN – MARSAGLIA CRYPTOGRATIC GENERATORS**

A family of the Maclaren–Marsaglia cryptographic generators for pseudorandom sequences is considered. A probabilistic model of the memory dynamics for the Maclaren–Marsaglia generators is proposed and analyzed. The results of computer experiments are presented.

УДК 519.681.3/004.41

В.В. Сергейчик, А.А. Иванюк

## ОСОБЕННОСТИ ОБФУСКАЦИИ VHDL-ОПИСАНИЙ И МЕТОДЫ ОЦЕНКИ ЕЕ СЛОЖНОСТИ

*Формализуется лексическая и функциональная обфускация. Обфускация – совокупность методик и средств, направленных на затруднение анализа исходных кодов программ. Дается краткий обзор способов лексической обфускации и изучаются их недостатки применительно к описаниям на языке VHDL. Рассматриваются методы оценки сложности описаний на языке VHDL. Приводится оценка сложности для различных вариантов VHDL-описаний одного цифрового устройства.*

### Введение

В настоящее время быстрыми темпами растет объем производства цифровых устройств, и в связи с этим особую остроту приобретает проблема нарушения авторских прав. По некоторым данным, ущерб от пиратства и других угроз в области производства аппаратного обеспечения составляет около 1 млрд долл. в день [1], что примерно в 10 раз превышает ущерб от пиратства в области ПО [2]. Кроме пиратства, появляются и новые виды угроз. Например, известны атаки на аппаратные реализации криптографических алгоритмов (так называемые Side-Channel Attacks [3]). В ходе этих атак осуществляется измерение токов и магнитных полей в некоторых частях схемы в моменты переключения значений, что позволяет получать значения секретных ключей. Также существует угроза внедрения аппаратных троянов, которые могут изменять функционирование устройства, снижать уровень его защиты, нарушать работоспособность и передавать секретную информацию из него. Поэтому сейчас огромное значение имеет разработка методов защиты от подобных угроз и борьбы с пиратством. Одним из таких методов является обфускация.

*Обфускация* (от англ. *obfuscate* – делать неочевидным, запутанным, сбивать с толку) – широко известная методика защиты исходных кодов программ от обратного проектирования [4]. Основной целью обфускации является затруднение понимания функционирования программы. В идеале сложность и временные затраты на обратное проектирование должны оказаться близки к затратам на разработку системы с нуля. Обфускация очень часто используется совместно с другими методами защиты от пиратства; например, ее применяют для сокрытия водяных знаков и отпечатков пальцев. Следует отметить, что обфускация используется также для защиты внедренного вредоносного кода (malware) от сканеров и при разработке вирусов.

Существует большое число методов обфускации, разработанных для различных языков программирования. Однако эти методы теряют свою актуальность в случае языка VHDL, так как результаты их применения не приводят к изменению конечного результата синтеза, т. е. схемы устройств до и после обфускации выглядят одинаково. Поэтому применительно к языку VHDL следует рассмотреть еще одну разновидность обфускации – функциональную [5]. Ее суть состоит в получении эквивалентной схемы.

### 1. Лексическая обфускация

Попробуем формализовать лексическую обфускацию для языка VHDL. Для этого представим исходный язык описания как

$$\langle V \rangle = \{T; S; D; K\},$$

где  $T$  – множество терминалов;  $S$  – множество выражений;  $D$  – множество объявлений;  $K$  – множество комментариев.

Схему цифрового устройства  $Sch$  можно описать следующим образом [5]:

$$Sch = \{IP; OP; B; L\},$$

где  $IP$  – множество входных портов;  $OP$  – множество выходных портов;  $B$  – множество функциональных блоков, из которых состоит схема устройства;  $L$  – множество проводящих линий, соединяющих внутренние блоки и порты.

Синтез – процесс интерпретации исходного описания на языке  $V$  в схему:

$$DD(V) = DD(\{T; S; D; K\}) = \{IP; OP; B; L\} = Sch.$$

Тогда лексическое эквивалентное преобразование – это замена одного фрагмента кода  $V_1 = \{T; S; D; K\}$ , результат синтеза которого

$$DD(V_1) = DD(\{T; S; D; K\}) = \{IP; OP; B; L\} = Sch_1,$$

другим фрагментом

$$V_2 = \{T'; S'; D'; K'\},$$

где  $V_1 \neq V_2$ , результат синтеза которого идентичен предыдущему:

$$DD(V_2) = DD(\{T'; S'; D'; K'\}) = \{IP; OP; B; L\} = Sch_1 = DD(V_1).$$

Сложность описания можно представить как функцию, зависящую от внутренней структуры  $V$ :

$$C(V) = C(\{T; S; D; K\}).$$

Лексическое обфусцирующее преобразование – это лексическое эквивалентное преобразование, для которого дополнительно выполняется свойство «сложность результирующего фрагмента  $V_2$  больше, чем сложность исходного  $V_1$ »:

$$C(V_2) > C(V_1), DD(V_2) = DD(V_1).$$

Лексическая обфускация – процесс применения лексических обфусцирующих преобразований к исходному описанию  $V$  с целью получения более сложного  $V^*$ , но при сохранении неизменным результата синтеза.

Частными случаями лексических обфусцирующих преобразований будут следующие:

1) преобразования, удаляющие или добавляющие избыточные конструкции в описание:

$$V' = V \pm V_r,$$

где  $V_r$  – избыточные конструкции;  $V$  – исходное описание;  $V'$  – результирующее описание.

При этом  $DD(V') = DD(V \pm V_r) = DD(V) \pm DD(V_r) = Sch \pm \emptyset = Sch$ . Примером таких преобразований будет внедрение комбинаций сигналов, минимизируемых при синтезе; удаление комментариев; разрушение форматирования; переименование идентификаторов.

2) преобразования, переупорядочивающие операции и не оказывающие влияние на семантику:

$$\begin{aligned} V &= \{p_1, \dots, p_n\}; \\ V' &= \text{permutation}(\{p_1, \dots, p_n\}); \\ DD(V) &= DD(V') = Sch. \end{aligned}$$

Примером служат преобразования переупорядочивания параллельных выражений.

## 2. Функциональная обфускация

Функциональность устройства  $Sch$  можно определить как зависимость значений на выходных портах  $OP$  от значений на входных портах  $IP$ , внутренних блоков  $B$  и проводящих линий  $L$  [5]:

$$OP = F_{Sch}(IP; B; L).$$

Пусть существует другое устройство  $Sch'$ , схема которого описывается выражением

$$Sch' = \{IP; OP; B'; L'\},$$

при этом множество блоков  $B'$  и множество линий  $L'$  не совпадают с аналогичными множествами устройства  $Sch$ .

Устройства  $Sch$  и  $Sch'$  будут функционально эквивалентными, если выполняется следующее равенство [5]:

$$F_{Sch}(IP; B; L) = F_{Sch'}(IP; B'; L').$$

Функциональное эквивалентное преобразование – это замена одного фрагмента кода  $V_1 = \{T; S; D; K\}$ , результат синтеза которого

$$DD(V_1) = DD(\{T; S; D; K\}) = \{IP; OP; B; L\} = Sch_1,$$

другим фрагментом  $V_2 = \{T'; S'; D'; K'\}$ , результат синтеза которого

$$DD(V_2) = DD(\{T'; S'; D'; K'\}) = \{IP; OP; B'; L'\} = Sch_2.$$

При этом  $Sch_2$  не совпадает с  $Sch_1$ . Однако наблюдаемое поведение [4] обеих схем идентично:

$$F_{Sch_1}(IP; B; L) = F_{Sch_2}(IP; B'; L').$$

Тогда преобразование функциональной обфускации – это функциональное эквивалентное преобразование, для которого дополнительно выполняется свойство «сложность результирующей схемы  $Sch_2$  выше, чем сложность исходной  $Sch_1$ »:

$$C_{Sch_1}(IP; OP; B; L) < C_{Sch_2}(IP; OP; B'; L').$$

Функциональная обфускация – процесс применения функциональных обфусцирующих преобразований к схеме  $Sch$  с целью получения более сложной для понимания схемы  $Sch^*$ , имеющей эквивалентную функциональность:

$$Obfusc(Sch) = Sch^*, F_{Sch}(IP; B; L) = F_{Sch^*}(IP; B^*; L^*).$$

Примером функциональной обфускации может быть мультиплексор, первый вход которого подключен к константе 0 (рис. 1). Такая схема эквивалентна вентилю  $and$ . С помощью некоторых ухищрений можно добиться того, что синтезатор не распознает в данной схеме  $and$ .

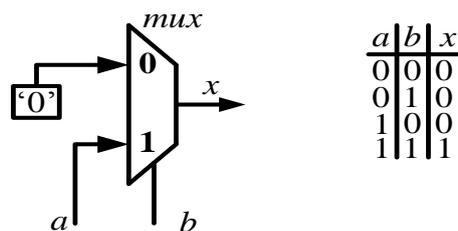


Рис. 1. Обфусцированная схема вентиля  $and$

Возникает необходимость в оценке качества проводимых функциональных преобразований. В связи с этим можно обратиться к методике [6], которая учитывает вносимые преобразованиями издержки в результирующую схему: увеличение использования аппаратуры и уменьшение скорости работы. Кроме того, методика позволяет оценить сложность обфусцированного описания. Однако в случае функциональных преобразований необходимо также оценивать не только сложность описания, но и сложность обфусцированной схемы по сравнению с исходной.

Понятие «сложность для понимания» подразумевает наличие человека и поэтому является достаточно субъективным и размытым. В связи с этим можно использовать подходы для оценки сложности логических цепей [7]: число вентилях в цепи, число вентилях на самом длинном пути. Очевидным недостатком упомянутых подходов является то, что они не учитывают число прово-

дящих линий и число соединений между блоками; кроме того, они не приспособлены для последовательных цепей. Эти методы также не учитывают разную квалификацию специалистов: одна и та же схема может иметь разную сложность для разных людей. Для простоты можно определить сложность схемы  $Sch$  для понимания как некоторую функцию  $C_{Sch}(IP; OP; B; L)$ , подразумевая, что сложность определяется только внутренним устройством схемы.

### 3. Обзор существующих методов обфускации

В настоящее время разработано большое количество методов лексической обфускации. Оригинальная классификация описана в работе [4]. Ниже приводятся основные группы преобразований с примерами согласно этой классификации: преобразования размещения, преобразования потока управления, преобразования данных. Исследуется также их применимость для VHDL.

#### 1. Преобразования размещения.

*Удаление комментариев.*

*Переименование идентификаторов* – замена имен в программе на последовательности символов, не несущие смысловой нагрузки. Для усложнения задачи анализа кода человеком имеет смысл генерировать длинные имена, близкие по написанию [5]. Этого можно достичь, используя похожие символы: I, l, 1, o, 0. Пример имен: oo01iooollo001oioiilol00o101ol001 и oo01iooollo001oioillol00o101ol001.

*Разрушение форматирования* – удаление отступов, табуляции, разграничений смысловых частей.

Данные преобразования являются достаточно слабыми даже для обычных языков программирования. Их можно считать предельным случаем плохого стиля кода.

#### 2. Преобразования потока управления.

*Преобразования агрегирования.*

Преобразования функций направлены на разрушение абстракций, создаваемых функциями, а также на порождение ложных абстракций. Примеры: встраивание функций, выделение функций, объединение функций.

Преобразования переупорядочивания циклов применяются для оптимизации в компиляторах. Примеры: разворачивание циклов, расщепление циклов, объединение циклов.

Приведенные преобразования агрегирования обладают достаточно малой стойкостью.

*Преобразования упорядочивания.*

Переупорядочивание выражений направлено на разрушение локальности расположения программистом связанных по смыслу операций. Преобразование обладает малой силой.

*Вычислительные преобразования.*

Преобразование потока управления из сократимого в несократимый основано на различии в описательной силе исходного языка программирования и представления, в которое он компилируется. Например, в java нет оператора goto, а в байт-коде он есть. Поэтому возможно представление с помощью байт-кода потока управления, который описывается на java на другом уровне абстракции, что создает трудности для деобфускаторов при попытке воссоздания исходного кода [4, 8].

Интерпретация таблиц – замена части кода исходного языка на код для виртуальной машины другого языка. Преобразование обладает высокой стойкостью, но требует больших затрат времени выполнения. Рекомендуется использовать его только для самых ценных частей кода.

#### 3. Преобразования данных.

*Изменение кодирования.* Примером может служить замена счетчика цикла  $i$  на выражение  $i' = c_1 + i * c_2$ , где  $c_1, c_2$  – некоторые константы. Преобразование обладает низкой стойкостью, легко удаляется оптимизирующим компилятором (это так называемые аффинные преобразования индексов [9]).

*Преобразование статических данных в процедурные* заменяет строки автоматами, генерирующими такие строки.

*Объединение скалярных переменных.* Суть данного преобразования заключается в замене нескольких переменных  $var_1 \dots var_n$  на одну –  $var$ , которая включает объединенные диапазоны  $var_1 \dots var_n$ .

*Изменение структуры массивов* разрушает абстракции, создаваемые массивами. Примеры: разделение массива на несколько независимых массивов, объединение двух и более массивов в один, сворачивание массива – увеличение числа измерений массива, сплющивание массива – уменьшение числа измерений массива. Данные преобразования используются в компиляторах для оптимизаций [9], преобразования обладают низкой стойкостью.

**Прочие преобразования.**

*Динамическая мутация кода* [10] – основана на замене кода обфусцируемой процедуры на вызов шаблона процедуры. В шаблоне часть инструкций заменена на бессмысленный набор байтов; кроме того, присутствует вызов редактирующего механизма, который конструирует процедуру из данного шаблона и последовательности байтов инструкций исходной процедуры. Затем происходит передача управления этой сконструированной процедуре. Метод обладает очень высокой стойкостью, так как процесс конструирования происходит во время выполнения программы.

*Внедрение сигналов, минимизируемых при синтезе.* Представляет собой специфичное для HDL-языков преобразование. Суть этого лексического преобразования заключается во внедрении в описание комбинаций сигналов, которые будут заведомо удалены при синтезе. Преобразование можно считать разновидностью метода введения избыточного кода (*Dead Code Insertion*). Данная трансформация реализуется следующим образом: сначала выбирается выражение, всегда вычисляемое в константу. Например:

$$(A + \bar{A}) * (B + \bar{B}) = '1' \Rightarrow A * B + \bar{A} * B + A * \bar{B} + \bar{A} * \bar{B} = '1',$$

где  $A$  и  $B$  – произвольные сигналы.

Выражение всегда принимает значение логической единицы. После раскрытия скобок получается выражение в правой части. Порядок дальнейшего вычисления показан на рис. 2.

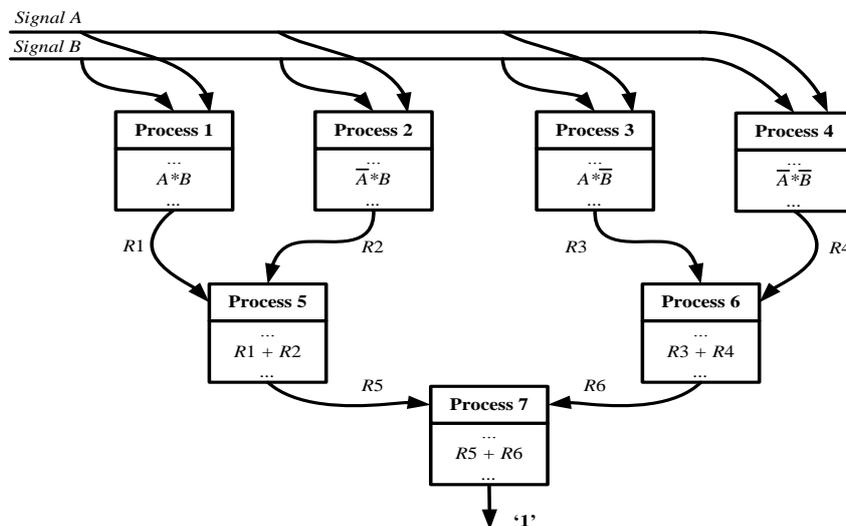


Рис. 2. Порядок вычисления значений методом минимизируемых сигналов

Каждое слагаемое выносится в отдельный процесс и вычисляется в нем, промежуточные результаты вычисления (сигналы  $R_1 - R_6$ ) попарно складываются в других процессах. Для реализации преобразования в виде, показанном на рис. 2, необходимы  $2^{x+1}-1$  процессов, где  $x$  – число независимых сигналов ( $A, B$  в данном примере), хотя возможны и другие реализации. Чтобы преобразование не слишком отличалось от остального кода, имеет смысл для вычислений использовать уже существующие в описании процессы. Это также позволит снизить связность отдельных процессов и увеличить сцепление между ними, что в итоге затруднит понимание.

Все приведенные преобразования в случае языка VHDL обладают низкой стойкостью. Причина этого заключается в том, что результат синтеза (схема) до и после обфускации остается неизменным. Именно поэтому простейшим способом атаки на лексические методы будет

синтез. Однако некоторую ценность эти методы все же представляют. Они ставят первый барьер на пути понимания системы, разрушая исходные высокоуровневые абстракции, известные идиомы, скрывая назначение кода за различными ложными операциями и абстракциями. Кроме того, обфускация добавляет дополнительную маскировку для других методов защиты, например для водяных знаков и отпечатков пальцев. В конечном итоге лексические методы приводят к необходимости привлечения более квалифицированных специалистов, увеличению времени, необходимого для обратного проектирования, а следовательно, снижают экономическую эффективность последнего.

#### 4. Пример лексической обфускации

Рассмотрим пример мультиплексора. Существует много вариантов его реализации на VHDL (рис. 3). Описания 1 – 4 отражают поведение устройства. Они являются эквивалентными, однако трудно выделить среди них наиболее сложное. Поэтому в данном случае нельзя сказать, что замена одного из описаний на другое в проектном коде приведет к лексической обфускации.

Самым низкоуровневым будет представление в виде логических элементов (описание 5). Такое описание немного труднее для восприятия, что связано с раскрытием низкоуровневых деталей реализации и разрушением абстракции переключателя. По этой причине его можно считать не просто очередным эквивалентным описанием, а лексической обфускацией описания.

Кроме того, мультиплексор можно представить в виде элемента памяти, адресуемого входными сигналами (описание 6). Данное описание сложнее для понимания, чем предыдущие. Его можно считать примером лексической обфускации, так как в нем создается фальшивая абстракция ROM и осуществляется работа с этой абстракцией.

Возможность реализации функций вместе с возможностью их перегрузки (в том числе и операторов, таких как «+», «-», «\*») позволяет создавать сбивающие с толку абстракции, которые «подражают» знакомым и хорошо известным, но отличаются поведением. Иллюстрацией служит описание 7.

- |    |   |    |   |
|----|---|----|---|
| 1) | <pre>process (a, b) begin   if (sel = '1') then     x &lt;= a;   else     x &lt;= b;   end if; end process;</pre>                     | 6) | <pre>process(a,b,sel)   variable index : integer;   variable temp : std_logic_vector(3 downto 1);   variable rom : std_logic_vector (7 downto 0) := "11001010"; begin   temp := sel &amp; a &amp; b;   index := conv_integer(temp);   x &lt;= rom(index); end process;</pre>  |
| 2) | <pre>process (a, b) begin   case sel is     when '1' =&gt; x &lt;= a;     when others =&gt; x &lt;= b;   end case; end process;</pre> | 7) | <pre>process (a, b, sel)   subtype pseudo_std_logic is std_logic range 'X' to '-';   FUNCTION "+" (v : std_logic_vector; selector : pseudo_std_logic)   RETURN std_logic IS     variable res : std_logic;   BEGIN     if (selector = '1') then       res := v(v'high);     else       res := v(v'low);     end if;     return res;   END "+"; begin   x &lt;= (a&amp;b) + sel; end process;</pre> |
| 3) | <pre>x &lt;= a when sel = '1' else b;</pre>   |    |   |
| 4) | <pre>with sel select   x &lt;= a when '1',   b when others;</pre>   |    |   |
| 5) | <pre>x &lt;= (sel and a) or (not sel and b);</pre>  |    |   |

Рис. 3. Некоторые варианты описаний мультиплексора

В описании 7 используется фальшивая абстракция операции сложения. Кроме того, здесь задействовано преобразование агрегирования (объединение переменных в вектор).

Результат применения лексических обфусцирующих преобразований размещения к описанию 7 показан на рис. 4. Из рисунка видно, что переименование идентификаторов и разрушение форматирования усложняют процесс анализа кода.

```
process (o10ll0110oo0o0oi0o10o0oooi0iilll0,o10ll0110oo0o0oi0i0o0oooi0iilll0
,o10ll0110oo0o0oi0ooloo0oooi0iilll0)subtype o10ll0111oo0o0oi0i0o0oooi0iilll0
to '-';function "+"(o10ll0110oo0o0oi0ioiooo0oooi0iilll0: is std_logic range 'X'
in std_logic_vector ;o10ll0110oo0o0oi0ioiooo0oooi0iilll0:in
o10ll0111oo0o0oi0i0o0o0oooi0iilll0)return std_logic is Variable
o10ll0110oo0o0oi0ioiooo000oi0iilll0:std_logic;begin if
(o10ll0110oo0o0oi0ioiooo0oooi0iilll0='1')then o10ll0110oo0o0oi0ioiooo000oi0iilll0
:=o10ll0110oo0o0oi0ioiooo0oooi0iilll0(o10ll0110oo0o0oi0ioiooo0oooi0iilll0 'high');
else o10ll0110oo0o0oi0ioiooo000oi0iilll0:=o10ll0110oo0o0oi0ioiooo0oooi0iilll0
(o10ll0110oo0o0oi0ioiooo0oooi0iilll0'low');end if;return
o10ll0110oo0o0oi0ioiooo000oi0iilll0;end;begin o10ll0110oo0o0oi0ioi0o0o0oooi0iilll0
<=(o10ll0110oo0o0oi0o10o0o0oooi0iilll0&o10ll0110oo0o0oi0i0o0o0oooi0iilll0)
+o10ll0110oo0o0oi0ooloo0oooi0iilll0;end process ;
```

Рис. 4. Результат лексической обфускации описания 7

Результат лексической обфускации описания 1 (рис. 5) в дальнейшем будем рассматривать как описание 8. В нем использованы преобразования разрушения форматирования, переименования параллельных операторов, переименования идентификаторов и внедрения комбинаций сигналов, минимизируемых при синтезе. Для осуществления последнего преобразования дополнительно введены еще девять сигналов и шесть процессов, взаимодействия между которыми также придется изучать, чтобы понять код. Часть действий с исчезающими сигналами производится внутри процесса, существовавшего в исходном описании, за счет внедрения в него дополнительных выражений.

```
process(o10iol0111ooli000i0011o0111lll1i,o10iol0ll0oli000i0011o0111lll1i,o10iol01i10oli000i0011o0111lll1i)begin
o10iol0111ooli000i0011o0111lll1i<=(not o10iol0ll0oli000i0011o0111lll1i)AND o10iol01i10oli000i0011o0111lll1i
;end process;process(o10iol0111ooli000i0011o0111lll1i,o10iol0ll0oli000i0011o0111lll1i,o10iol01i10oli000i0011o0111lll1i)begin
o10iol0111ooli000i0011o0111lll1i<=(o10iol0ll0oli000i0011o0111lll1i)AND o10iol01i10oli000i0011o0111lll1i;end process
;process(o10iol0111ooli000i0011o0111lll1i,o10iol01i10oli000i0011o0111lll1i,
o10iol01i10oli000i0011o0111lll1i)begin o10iol01i10oli000i0011o0111lll1i<=(o10iol01i10oli000i0011o0111lll1i)
OR o10iol01i10oli000i0011o0111lll1i;end process;process(o10iol0111ooli000i0011o0111lll1i,o10iol0ll0oli000i0011o0111lll1i,
o10iol01i10oli000i0011o0111lll1i)begin o10iol01i10oli000i0011o0111lll1i<=(
o10iol0ll0oli000i0011o0111lll1i)AND not o10iol01i10oli000i0011o0111lll1i;end process;process(o10iol0111ooli000i0011o0111lll1i
,o10iol01i10oli000i0011o0111lll1i,o10iol01i10oli000i0011o0111lll1i)begin o10iol01i10oli000i0011o0111lll1i<=
(o10iol01i10oli000i0011o0111lll1i)OR o10iol01i10oli000i0011o0111lll1i;end process;
process(o10iol0111ooli000i0011o0111lll1i,o10iol01i10oli000i0011o0111lll1i,o10iol01i10oli000i0011o0111lll1i,
o10iol01i10oli000i0011o0111lll1i,o10iol0ll0oli000i0011o0111lll1i
o10iol01i10oli000i0011o0111lll1i ,o10iol01i10oli000i0011o0111lll1i)begin if (o10iol01i10oli000i0011o0111lll1i='1')then
<=o10iol01i10oli000i0011o0111lll1i;else o10iol01i10oli000i0011o0111lll1i<=o10iol01i10oli000i0011o0111lll1i;end
if;o10iol01i10oli000i0011o0111lll1i<=(not o10iol0ll0oli000i0011o0111lll1i)AND not o10iol01i10oli000i0011o0111lll1i;
end process;process(o10iol01i10oli000i0011o0111lll1i,o10iol01i10oli000i0011o0111lll1i,o10iol01i10oli000i0011o0111lll1i
)begin o10iol01i10oli000i0011o0111lll1i<=(o10iol01i10oli000i0011o0111lll1i)OR o10iol01i10oli000i0011o0111lll1i; end process;end;
```

Рис. 5. Результат лексической обфускации описания 1

Синтез всех приведенных описаний дает одинаковый результат. Отсюда следует простейшая атака на лексические методы – синтез. Кроме того, у специалиста может быть в распоряжении только результат синтеза, в таком случае он даже не узнает, что обфускация имела место.

Описания различаются по сложности для восприятия. Для оценки сложности описаний можно воспользоваться различными методами. Интуитивным подходом будет оценка сложности по числу терминалов, входящих в состав описания, или по числу операций. Однако многие идиомы описаний на VHDL могут содержать достаточно много терминалов и при этом быть простыми для понимания и узнаваемыми.

Метод, использующий одновременно несколько метрик оценки и весовые коэффициенты, представлен в работе [6], где приведена методика оценки обфусцирующих преобразований для VHDL-описаний с учетом издержек, вносимых обфусцирующими преобразованиями в проект.

Более подробно методы оценки сложности рассмотрены в следующем разделе.

Приведенным эквивалентным преобразованиям можно найти еще одно полезное применение помимо запутывания. Их можно использовать для внедрения водяных знаков в исходный код. Например, если выбрать  $2^n$  различных описаний и каждому из них поставить в соответствие индекс  $n$ -битовой последовательности символов, водяной знак можно кодировать заменой описаний мультиплексоров в коде на эквивалентные, но с номерами, выбранными согласно требуемым битам водяного знака. Подобный метод для последовательностей эквивалентных ассемблерных команд приведен в работе [11]. Такой водяной знак будет достаточно слабым в случае VHDL: его легко удалить переупорядочиванием параллельных операторов. В операторах наподобие `if` из описания 1, где возможны несколько эквивалентных равнозначных вариантов (`if(sel = '1')` или `if(sel = '0')`), можно дополнительно закодировать 1 бит информации так: если указано сравнение с 1, то бит водяного знака равен 1, иначе 0. Способ имеет низкую стойкость.

## 5. Метрики оценки сложности

Для оценки сложности исходных кодов используются различные метрики. Метрики оценки сложности показывают качество исходного кода. Эти же метрики очень удобно использовать для оценки эффективности обфусцирующих преобразований, однако в случае обфускации целью является не минимизация значений метрик, а, наоборот, максимизация [4]. Рассмотрим несколько наиболее известных метрик и способов оценки сложности.

Метрика Маккейба показывает сложность потока управления программы [12]. В случае VHDL этого недостаточно: метрика никак не учитывает сложность параллельных взаимодействий.

Метрика Холстеда [13] оценивает сложность потока данных, игнорируя сложность потока управления.

Метрика числа строк кода (LOC) [13] в случае обфускации мало применима из-за преобразований, разрушающих форматирование (обычно они представляют все описание в виде одной строки кода). Вместо числа строк кода можно воспользоваться числом операторов.

Связность – степень направленности методов класса на решение только одной проблемы [14]. Эта метрика носит, скорее, экспертный характер и поэтому трудна для автоматического вычисления.

Сцепление – число зависимостей между классами либо модулями [13]. Чем выше сцепление, тем выше энтропия и тем труднее понять систему и ее поведение. В случае языка VHDL сцепление можно понимать двояко. Во-первых, в качестве модулей можно применять пакеты или архитектуры. Во-вторых, можно рассмотреть сцепление параллельных операторов посредством одинаковых сигналов: чем больше параллельных операторов зависят от одного и того же сигнала, тем сложнее понять их взаимодействие.

Пространственная сложность – сложность, связанная с расположением элементов в описании или программе, их расстоянием от места объявления и предыдущего использования [15]. В том же источнике предложены несколько метрик, нацеленных в первую очередь на объектно-ориентированное ПО. Например, метрика пространственной сложности атрибута класса (CASC) и общая пространственная сложность класса (TCASC) вычисляются следующим образом:

$$CASC = \frac{1}{N} \cdot \sum_{k=1}^N Dist(k); TCASC = \frac{1}{q} \cdot \sum_{i=1}^q CASC(i),$$

где  $Dist(k)$  – расстояние от места данного использования атрибута до предыдущего, выраженное в строках кода;  $N$  – число использований атрибута;  $q$  – число атрибутов. Вариации подобных метрик могут быть применены и для оценки обфусцирующих преобразований в случае VHDL. Величину «расстояние», которая там определена как число строк кода, для оценки обфусцирующих преобразований можно вычислить через число операторов или выражений.

Для оценки сложности приведенных эквивалентных преобразований предлагаются следующие метрики:

$M_1$  – число операторов;

$M_2$  – среднее число операторов в параллельном выражении;

$M_3$  – число параллельных выражений;

$M_4$  – число сигналов и переменных;

$M_5$  – сцепление параллельных операторов (связанность параллельных операторов посредством одинаковых сигналов в списке чувствительности);

$M_6$  – средний размер списка чувствительности параллельных операторов;

$M_7$  – число объявлений (число объявленных типов, сигналов, переменных и т. д.);

$M_8$  – TCASC.

Для расчета общей сложности описания используем формулу

$$C(V) = \sum_{i=1}^n a_i \cdot M_i,$$

где  $a_i$  – весовой коэффициент, выбранный экспертами для метрики;  $M_i$  – рассчитанное значение  $i$ -й метрики.

В таблице приведены результаты расчета метрик для описаний мультиплексора из предыдущего раздела при  $a_i = 1$  для всех метрик  $M_i$ .

Результаты оценки уровня сложности

Метрика	Вариант							
	1	2	3	4	5	6	7	8
$M_1$	4	4	3	3	5	8	13	28
$M_2$	3	3	3	3	5	7	12	3
$M_3$	1	1	1	1	1	1	1	7
$M_4$	4	4	4	4	4	7	5	13
$M_5$	1	1	1	1	1	1	1	1,84
$M_6$	3	3	3	3	3	3	3	3,42
$M_7$	0	0	0	0	0	3	2	9
$M_8$	1,875	1,875	1,75	2	2,5	2,67	5,52	7,73
$C(V)$	17,875	17,875	16,75	17	23,5	32,67	43,52	71,69

Результаты оценки показывают, что уровень сложности повышается при разрушении исходных абстракций. При замене абстракций более низкоуровневыми сложность возрастает, но не очень сильно. Наибольший прирост сложности наблюдается у преобразований, создающих новые абстракции. Количественно это проявляется в увеличении числа объявлений типов и структур данных, а также способов взаимодействия между ними (например, выражений, функций или порядка адресации). Дополнительно следует отметить, что минимальные значения сложности  $C(V)$  могут быть свидетельством хорошего качества и стиля VHDL-кода.

### Заключение

Обфускация представляет собой одно из новых, но перспективных направлений в защите программных кодов и описаний. В случае языка VHDL традиционные методы обфускации теряют свою силу в связи с возможностью синтеза, при котором результаты преобразований будут удалены. Язык VHDL позволяет описывать одни и те же схемы различными способами, поэтому возникает необходимость в оценке уровня сложности проводимых преобразований. Функциональная обфускация позволяет преодолеть недостатки лексической обфускации: не только исходное описание, но также результат синтеза до и после обфускации выглядят по-разному, однако поведение схемы остается одинаковым. Недостатком функциональной обфускации является внесение дополнительных аппаратных и временных издержек в схему устройства. Имеет смысл

применять лексическую и функциональную обфускацию совместно, чтобы обеспечить высокий уровень защиты на всех стадиях жизненного цикла устройства.

### Список литературы

1. Hardware Security Mechanisms for Authentication and Trust // GLSVLSI 2011 [Electronic resource]. – Mode of access : [http://www.glsvlsi.org/archive/glsvlsi11/Koushanfar\\_MeteringGLS-VLSI.pdf](http://www.glsvlsi.org/archive/glsvlsi11/Koushanfar_MeteringGLS-VLSI.pdf). – Date of access : 9.04.2013.
2. Active Hardware Metering for Intellectual Property Protection and Security // Usenix [Electronic resource]. – Mode of access : <https://www.usenix.org/conference/16th-usenix-security-symposium/active-hardware-metering-intellectual-property-protection>. – Date of access : 5.04.2013.
3. Majzoobi, M. Introduction to hardware security and trust / M. Majzoobi, F. Koushanfar, M. Potkonjak. – N. Y. : Springer, 2011. – 427 p.
4. Collberg, C. A Taxonomy of Obfuscating Transformations / C. Collberg, C. Thomborson, D. Low. – Auckland : Department of Computer Science, 1997. – 36 p.
5. Иванюк, А.А. Проектирование встраиваемых цифровых устройств и систем : монография / А.А. Иванюк. – Минск : Бестпринт, 2012. – 337 с.
6. Brzozowski, M. Obfuscation quality in hardware designs / M. Brzozowski, V.N. Yarmolik // Zeszyty Naukowe Politechniki Bialostockiej. Informatyka. – 2009. – № 4. – P. 19–29.
7. Circuit\_complexity // Wikipedia [Electronic Resource]. – Mode of Access : [http://www.en.wikipedia.org/wiki/Circuit\\_complexity](http://www.en.wikipedia.org/wiki/Circuit_complexity). – Date of Access : 8.09.2013.
8. Hou, T. Three control flow obfuscation methods for Java software / T.W. Hou, H.Y. Chen, M.H. Tsai // Software IEE Proceedings. – 2006. – Vol. 153(2). – P. 80–86.
9. Компиляторы: принципы, технологии, инструменты / А. Ахо [и др.]. – 2-е изд. – СПб. : Вильямс, 2008. – 1184с.
10. Software Protection Through Dynamic Code Mutation / M. Madou [et al.] // Information Security Applications : 6<sup>th</sup> International Workshop WISA–2005. – Jiju Island, Korea, 2005. – P. 194–206.
11. Ярмолик, В.Н. Криптография, стеганография и охрана авторского права / В.Н. Ярмолик, С.С. Портянко, С.В. Ярмолик. – Минск : Изд. центр БГУ, 2007. – 240 с.
12. Software Complexity Measurement / J. Kearney [et al.] // Communications of the ACM. – 1986. – Vol. 29. – P. 1044–1050.
13. Sheng, Y. A Survey on Metric of Software Complexity / Y. Sheng, Z. Shijie // Information Management and Engineering (ICIME). – Chengdu, China, 2010. – P. 352–356.
14. Макконелл, С. Совершенный код / С. Макконелл. – СПб. : Питер, 2005. – 893 с.
15. Gupta, V. Object-oriented cognitive-spatial complexity measures / V. Gupta, K. Chhabra // International J. of Computer Engineering & Science. – 2009. – Vol. 3. – P. 122–129.

Поступила 4.11.2013

*Белорусский государственный университет  
информатики и радиоэлектроники,  
Минск, ул. П. Бровки, 6  
e-mail: vovasq@mail.ru*

**V.V. Sergeichik, A.A. Ivaniuk**

### **FEATURES OF OBFUSCATION OF VHDL-DESIGNS AND ITS COMPLEXITY EVALUATION METHODS**

Lexical and functional obfuscation is formalized. Brief survey of methods of lexical obfuscation is given and their drawbacks are investigated when applied to specifications in VHDL language. Complexity evaluation methods for specifications in VHDL language are considered. Complexity evaluation for different variants of VHDL specifications of a given digital device is presented.

КОНФЕРЕНЦИИ



28-30 октября 2014 г. Минск, Беларусь

ОРГАНИЗАТОРЫ

Национальная академия наук Беларуси

Объединенный институт проблем информатики

ЦЕЛИ КОНГРЕССА

Обсуждение новых достижений в космической области, определение приоритетных направлений развития космической тематики

Укрепление международного сотрудничества в области космических исследований

Расширение совместных усилий белорусских и зарубежных деловых кругов, фирм, компаний, ученых и специалистов, направленных на выполнение космических программ, анализ хода их реализации и выбор перспективных направлений дальнейших исследований

Развитие партнерства правительственных и общественных организаций, предприятий и организаций разных форм собственности, заинтересованных в исследованиях и использовании космического пространства в мирных целях

ТЕМАТИКА КОНГРЕССА

Перспективные приборы и материалы для космической техники

Спутники, целевая и научная аппаратура

Обработка изображений земной поверхности

Геоинформационные системы и их применение

Космические технологии и образование

Навигационные космические системы

**РАБОЧИЕ ЯЗЫКИ КОНГРЕССА**

Русский, английский

**ВАЖНЫЕ ДАТЫ**

Представление доклада	–	<b>до 30 мая 2014 г.</b>
Извещение о приеме	–	<b>до 5 сентября 2014 г.</b>
Оплата оргвзноса	–	<b>до 30 сентября 2014 г.</b>

Для зарубежных участников и участников из стран СНГ возможна оплата оргвзноса при регистрации.

**ОБЩИЕ ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ДОКЛАДА**

Доклады на конгресс представляются на русском или английском языке в двух экземплярах окончательно оформленных работ и в электронном виде. Электронная версия доклада может быть передана по электронной почте. Белорусские участники также предоставляют в одном экземпляре экспертное заключение. Один экземпляр доклада будет послан на рецензирование. Если представленная работа будет принята, то второй экземпляр будет непосредственно использован для тиражирования трудов конгресса, которые будут изготовлены к началу работы конгресса. Объем представляемых докладов – до четырех полных нумерованных страниц формата А4.

Текст должен быть размещен в пределах рабочего поля 160x240 (верх/низ/лево/право – 25/32/25/25) мм, в одну колонку, с одинарным межстрочным интервалом, шрифт 12 pt Times New Roman. Доклад должен включать: УДК (в левом верхнем углу), название (прописными буквами), Ф.И.О. авторов, полное название организации, город, аннотацию (не более 100 слов, шрифт 10 pt Times New Roman), основной текст доклада.

Детальные требования к оформлению текста доклада на русском языке будут прилагаться к информационному письму отдельно на примере представления текста самих требований и будут размещены в Интернете.

В оргкомитет конгресса также необходимо предоставить сведения о всех авторах, которые должны включать: Ф.И.О., ученое звание и степень, занимаемую должность, место работы, почтовый адрес, телефоны, факс и e-mail для оперативной связи. Должна быть указана фамилия докладчика.

**ИНФОРМАЦИЯ О КОНГРЕССЕ**

Шестой Белорусский космический конгресс будет проводиться в Минске в Объединенном институте проблем информатики Национальной академии наук Беларуси в период с 28 по 30 октября 2014 г.

Для участия в работе конгресса необходимо **до 30 сентября 2014 г.** оплатить организационный взнос. Информация об уплате организационного взноса и реквизиты для оплаты будут предоставлены авторам принятых докладов.

Бронирование гостиницы для иногородних участников будет осуществляться по их заявкам, поступившим не позднее **1 сентября 2014 г.** Резервирование места в гостинице оплачивает участник конгресса.

**АДРЕС КОНГРЕССА**

Объединенный институт проблем информатики Национальной академии наук Беларуси,  
ул. Сурганова, 6, Минск, 220012, Республика Беларусь

**Тел.:** +375-17-284 20 74    Кореняко Сергей Анатольевич  
          +375-17-284 20 98    Ляткевич Ирина Антоновна  
          +375-17-284 26 02    Железнова Татьяна Владимировна

**Факс:** +375-17-284 21 75

**E-mail:** [cosmos6@newman.bas-net.by](mailto:cosmos6@newman.bas-net.by)

**Официальный сайт:** <http://cosmos.basnet.by/belcosm6/>

**ХIII МЕЖДУНАРОДНАЯ КОНФЕРЕНЦИЯ  
«РАЗВИТИЕ ИНФОРМАТИЗАЦИИ И ГОСУДАРСТВЕННОЙ СИСТЕМЫ  
НАУЧНО-ТЕХНИЧЕСКОЙ ИНФОРМАЦИИ» (РИНТИ–2014)  
20 ноября 2014 г., Минск, Беларусь**

**ЦЕЛИ КОНФЕРЕНЦИИ**

Анализ состояния и проблем развития информатизации и государственной системы научно-технической информации в Республике Беларусь; разработка новых подходов к научно-методическому обеспечению развития информатизации, информационно-коммуникационных технологий и информационной инфраструктуры Беларуси; реализация Стратегии развития информационного общества в Республике Беларусь на период до 2015 г. и Национальной программы ускоренного развития услуг в сфере информационно-коммуникационных технологий на 2011–2015 гг.

**ТЕМАТИКА КОНФЕРЕНЦИИ**

• Анализ состояния и проблем развития единого информационного пространства, методов и средств обеспечения его формирования, использования и развития в интересах граждан, общества и государства, в том числе:

– развития общегосударственной автоматизированной информационной системы в части объединения информационных ресурсов органов государственного управления и обеспечения доступа к ним, организации межведомственного информационного взаимодействия;

– развития информационной инфраструктуры;

– развития и совершенствования информационно-коммуникационных технологий и формирования экспортно ориентированной отрасли индустрии информационных технологий;

– совершенствования законодательной базы и системы государственного регулирования в сфере информатизации и системы научно-технической информации;

– развития системы подготовки и переподготовки специалистов по информационно-коммуникационным технологиям и квалифицированных пользователей;

– развития культуры и средств массовой информации путем внедрения информационно-коммуникационных технологий;

– совершенствования системы информационной безопасности республики.

• Развитие государственной системы научно-технической информации.

• Оказание электронных услуг для населения и бизнеса.

• Формирование интегрированной системы стандартизации (унификации), сертификации, экспертизы и оценки эффективности программ, проектов и результатов информатизации.

• Развитие научно-информационных компьютерных сетей и технологий высокоскоростного информационного обмена между организациями с доступом в международные научно-образовательные сети.

• Корпоративная сеть библиотек Республики Беларусь и ее информационные ресурсы, электронные библиотеки, генерация собственных информационных ресурсов, обеспечение доступа к национальным и зарубежным информационным ресурсам.

• Региональная информатизация и информационная инфраструктура.

**РАБОЧИЕ ЯЗЫКИ КОНФЕРЕНЦИИ**

В качестве рабочих языков устанавливаются русский и английский.

## ПОРЯДОК ПРЕДСТАВЛЕНИЯ И ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ДОКЛАДОВ

Представление докладов – до 30 июля 2014 г.

Доклады представляются на русском или английском языках по электронной почте [vengerov@basnet.by](mailto:vengerov@basnet.by) (запасной адрес – [gsnti@basnet.by](mailto:gsnti@basnet.by)), а также один экземпляр вместе с актом экспертизы в оргкомитет конференции нарочным по адресу: 220072, г. Минск, ул. Академическая, 25, к. 711 (к. 704 и 707) или письмом на почтовый адрес конференции.

Акт экспертизы (экспертное заключение) с указанием возможности опубликования доклада в открытой печати белорусские участники представляют в одном экземпляре в обязательном порядке. Без такого документа доклад оргкомитетом не рассматривается.

Доклады будут опубликованы в сборнике докладов и предоставлены всем авторам в день выступления на конференции.

Авторам докладов необходимо зарегистрироваться на сайте конференции и указать контактные данные для оперативной связи с ними.

Объем представляемых докладов – до шести нумерованных страниц формата А4.

Доклад должен включать: УДК (в левом верхнем углу); название доклада; И.О.Ф. автора; полное название организации, города, страны; аннотацию (не более 100 слов); текст доклада (с обязательными ссылками на литературу и рисунки); список литературы (если имеется), оформленный согласно приложению 2 к инструкции ВАК по оформлению диссертации (<http://www.vak.org.by>).

Поля текста: верхнее – 2,5 см, нижнее – 3,2 см, левое – 2,5 см, правое – 2,5 см, абзацный отступ – 1,0 см. Интервал между строками и абзацами – одинарный. Текст должен входить в рамку 24 × 16 см и набираться с переносами. При подготовке текста доклада используются шрифты семейства Times New Roman Суг следующих размеров и типов:

УДК – прописные 12 пт; **НАЗВАНИЕ ДОКЛАДА** – прописные полужирные 12 пт; И.О.Ф. – строчные 12 пт; страна, город – строчные 12 пт; *аннотация* – курсив 10 пт (отступ слева и справа – 1 см); **заголовки разделов** – строчные полужирные 12 пт; **заголовки подразделов** – строчные полужирные 12 пт; основной текст – 12 пт. без нумерации страниц. Основной текст может включать введение (тогда обязательно и заключение). Все заголовки должны быть выровнены по левому краю.

## АДРЕС КОНФЕРЕНЦИИ

220012, Республика Беларусь, Минск, ул. Сурганова, 6, ОИПИ НАН Беларуси (<http://www.uiip.basnet.by>, тел.+375 17 284-25-26).

### Председатель программного комитета

Тузиков Александр Васильевич +375 17 284-21-75 [tuzikov@newman.bas-net.by](mailto:tuzikov@newman.bas-net.by)

### Заместители председателя программного комитета:

Алюшкевич Виктор Бернардович +375 17 284-09-85 [aliushkevich@newman.bas-net.by](mailto:aliushkevich@newman.bas-net.by)

Григянец Ромуальд Брониславович +375 17 284-25-26 [griganec@bas-net.by](mailto:griganec@bas-net.by)

### Ответственные секретари:

Венгеров Виктор Николаевич +375 17 284-12-08 [vengerov@basnet.by](mailto:vengerov@basnet.by)

Мисякова Галина Тихоновна +375 17 217-74-06 [gsnti@basnet.by](mailto:gsnti@basnet.by)

**Факс:** +375 17 284-07-22

**E-mail:** [vengerov@basnet.by](mailto:vengerov@basnet.by), [gsnti@basnet.by](mailto:gsnti@basnet.by)

**Сайт:** <http://opac.bas-net.by/opacpage/rinti>

## ПРАВИЛА ДЛЯ АВТОРОВ

1. Статьи принимаются в редакцию через электронную систему подачи по адресу <http://jinfo.bas-net.by> в формате файлов текстовых редакторов Microsoft Word 97 и Word 2000 для Windows. Основной текст статьи набирается с переносами шрифтом Times New Roman 11 пт, интервал между строками – одинарный, абзацный отступ 1 см, поля по 2,5 см со всех сторон.

2. Объем статьи не должен превышать 12 страниц (включая таблицы, иллюстрации, список литературы), количество иллюстраций – не больше пяти. Допускаются краткие сообщения до трех страниц.

3. Статья должна иметь индекс УДК (универсальная десятичная классификация).

4. Название статьи, фамилии всех авторов и аннотация должны быть переведены на английский язык. Для каждого из авторов приводится развернутое название учреждения с полным почтовым адресом, а также номер телефона и электронный адрес (e-mail) для связи с редакцией.

5. Формулы, иллюстрации, таблицы, встречающиеся в статье, должны быть пронумерованы в соответствии с порядком цитирования в тексте. Ссылки на рисунки и таблицы в тексте обязательны. Необходимо избегать повторения одних и тех же данных в таблицах, графиках и тексте статьи.

Рисунки должны быть представлены в виде файлов формата .cdr, .ai, .wmf, .psd, .jpg, .tif (.tiff) и выполнены с хорошим разрешением в масштабе, позволяющем четко различать надписи и обозначения. Подрисовочные подписи с расшивкой всех позиций, представленных на рисунке, набираются шрифтом гарнитуры основного текста, размер символов 9 пт. Цветные иллюстрации печатаются только в том случае, когда это необходимо для понимания излагаемого материала.

6. Набор формул выполняется в формульных редакторах Microsoft Equation или Math Type и должен быть единообразным по применению шрифтов и знаков по всей статье.

Прямо ( ) набираются: греческие и русские буквы; математические символы ( $\sin$ ,  $\lg$ ,  $\infty$ ); символы химических элементов (C, Cl,  $\text{CHCl}_3$ ); цифры (римские и арабские); векторы; индексы (верхние и нижние), являющиеся сокращениями слов.

Курсивом (~) набираются: латинские буквы – переменные, символы физических величин (в том числе и в индексе).

7. Сокращения в тексте статьи (за исключением единиц измерения) могут быть использованы только после упоминания полного термина. Единицы измерения физических величин следует приводить в Международной системе СИ.

8. Литература приводится автором общим списком в конце статьи. Ссылки на литературу в тексте идут по порядку и обозначаются цифрой в квадратных скобках. Ссылаться на неопубликованные работы не допускается. С примерами оформления библиографического описания в списке литературы можно ознакомиться в приложении 2 к *Инструкции по оформлению диссертации, автореферата и публикаций по теме диссертации* на сайте Высшей аттестационной комиссии Республики Беларусь <http://vak.org.by>.

9. Поступившие в редакцию статьи направляются на рецензирование специалистам. Основным критерием целесообразности публикации является новизна и информативность статьи. Если по рекомендациям рецензента статья возвращается автору на доработку, а переработанная рукопись вновь рассматривается редколлегией, датой поступления считается день получения редакцией ее окончательного варианта. Статьи не по профилю журнала возвращаются авторам после заключения редколлегии.

10. Статьи, направляемые на доработку, должны быть возвращены в исправленном виде с ответами на все вопросы.

11. Редакция журнала предоставляет возможность первоочередного опубликования статей, представленных лицами, которые осуществляют послевузовское обучение (аспирантура, докторантура, соискательство) в год завершения обучения.

12. Авторы несут ответственность за направление в редакцию статей, уже опубликованных ранее, или статей, принятых к публикации другими изданиями.

13. Редакция оставляет за собой право на редакционные изменения, не искажающие основное содержание статьи.

***Журнал «Информатика» включен Высшей аттестационной комиссией Республики Беларусь в список научных изданий для опубликования результатов диссертационных исследований.***

## **Индексы**

**00827**

для индивидуальных  
подписчиков

**008272**

для предприятий и  
организаций