

ISSN 1816-0301

ИНФОРМАТИКА

4(48)

ОКТЯБРЬ-ДЕКАБРЬ
2015

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, 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. Ярмолик
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
<http://uiip.bas-net.by>

ОБЪЕДИНЕННЫЙ ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ
НАЦИОНАЛЬНОЙ АКАДЕМИИ НАУК БЕЛАРУСИ

ИНФОРМАТИКА

ЕЖЕКВАРТАЛЬНЫЙ НАУЧНЫЙ ЖУРНАЛ

Издается с января 2004 г.

№ 4(48) • октябрь-декабрь 2015

СОДЕРЖАНИЕ

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

- Малюгин В.И.** Алгоритмы тестирования циклических структурных изменений в моделях векторной авторегрессии с переключением состояний5
- Ерофеев В.Т.** Модель вычисления эффективных параметров матричного композита из биизотропных частиц с учетом многократных переотражений электромагнитного поля17
- Врублевский А.С., Шлык В.А.** Вычисление вершин политопов разбиений чисел34
- Гривачевский А.Г., Гривачевский В.Д., Ковалев М.Я., Сухомлинов А.И.** Планирование процесса изготовления секции корпуса судна как трехстадийная задача Джонсона49
- Гущинский Н.Н., Зданович В.Е., Розин Б.М.** Оптимизация размещения детали на многопозиционном поворотном столе агрегатного станка57

ОБРАБОТКА СИГНАЛОВ, ИЗОБРАЖЕНИЙ И РЕЧИ

- Лисица Е.В., Яцков Н.Н., Апанасович В.В., Апанасович Т.В.** Программный пакет CellDataMiner для анализа люминесцентных изображений раковых клеток73
- Петровский Н.А., Станкевич А.В., Петровский А.А.** CORDIC-техника для фиксированного угла вращения в операции умножения кватернионов85

НАДЕЖНОСТЬ СИСТЕМ

Кульбак Л.И., Алюшкевич В.Б., Золотой С.А. Оценка надежности бортовой аппаратуры малых космических аппаратов в процессе их полета109

Редактор Г.Б. Гончаренко
Корректор А.А. Михайлова
Компьютерная верстка О.Б. Бутевич

Сдано в набор 02.11.2015. Подписано в печать 01.12.2015.
Формат 60×84 1/8. Бумага офсетная. Гарнитура Таймс. Ризография.
Усл. печ. л. 13,7. Уч.-изд. л. 13,4. Тираж 70 экз. Заказ 14.

Государственное научное учреждение «Объединенный институт проблем информатики Национальной академии наук Беларуси».
Свидетельство о государственной регистрации издателя, изготовителя, распространителя печатных изданий № 1/274 от 04.04.2014.
ЛП № 02330/444 от 18.12.13.
Ул. Сурганова, 6, 220012, Минск.

© Объединенный институт проблем информатики
Национальной академии наук Беларуси, 2015

INFORMATICS

PUBLISHED QUATERLY

Issued since 2004

№ 4(48) • October-December 2015

CONTENTS

MATHEMATICAL MODELING

- Malugin V.I.** Testing of cyclic structural changes in switching regime vector autoregressive models5
- Erofeenko V.T.** Model for calculating effective parameters of matrix composites from bi-isotropic particles with regard multiple reflections of electromagnetic field.....17
- Vroublevski A.S., Shlyk V.A.** Computing vertices of integer partition polytopes34
- Grivachevsky A.G., Grivachevsky V.D., Kovalyov M.Y., Suhomlinov A.I.** Planning vessel body section production49
- Guschinski N.N., Zdanovich V.E., Rozin B.M.** Optimizing the placement of a work-piece at a multi-position rotary table of transfer machine with vertical multi-spindle head57

SIGNAL, IMAGE AND SPEECH PROCESSING

- Lisitsa Y.U., Yatskou M.M., Apanasovich V.V., Apanasovich T.V.** The software package CellDataMiner for data analysis of fluorescent images of cancer cells.....73
- Petrovsky N.A., Stankevich A.V., Petrovsky A.A.** CORDIC techniques for fixed angle of rotation in multiplying operation of quaternions.....85

SYSTEMS RELIABILITY

Kulbak L.I., Aliushkevich V.B., Zolotoy S.A. Assessment of reliability of the onboard equipment of small spacecrafts during their flight109

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

УДК 519.237: 681.3

В.И. Малюгин

АЛГОРИТМЫ ТЕСТИРОВАНИЯ ЦИКЛИЧЕСКИХ СТРУКТУРНЫХ ИЗМЕНЕНИЙ В МОДЕЛЯХ ВЕКТОРНОЙ АВТОРЕГРЕССИИ С ПЕРЕКЛЮЧЕНИЕМ СОСТОЯНИЙ

Для векторных авторегрессионных моделей с циклическими переключениями состояний предлагается метод исключения краткосрочных колебаний состояния системы, который основывается на последовательном применении двух алгоритмов, реализующих подстановочное байесовское решающее правило поточечной классификации и статистический критерий проверки гипотезы о значении ожидаемой вероятности ошибки классификации. Результаты компьютерного моделирования демонстрируют работоспособность предлагаемого метода.

Введение

Под структурными изменениями, или структурными сдвигами, в статистических моделях понимаются скачкообразные изменения параметров модели сложной системы, обусловленные внешними «шоковыми» воздействиями [1]. Структурные изменения, как правило, не затрагивают фундаментальные зависимости между используемыми в модели переменными, т. е. не приводят к «разрушению» структуры модели, а отражаются только на значениях ее параметров. Целью анализа таких систем обычно является построение статистической модели с учетом установленных моментов структурных изменений с помощью соответствующих фиктивных переменных [2]. Методам анализа структурных изменений в статистических моделях при различных модельных предположениях посвящено значительное число исследований. Обзор наиболее важных результатов дается в [3].

Во многих приложениях актуальными являются сложные системы с фиксированным числом классов состояний и структурными сдвигами, приводящими к последовательной смене состояний. Такие сдвиги будем называть циклическими структурными изменениями. Для описания сложных систем в указанных условиях используются многомерные статистические модели с циклическими переключениями состояний (Regime Switching Models – RS-Models) [4]. Основными задачами анализа подобных систем являются: установление циклического характера смены классов состояний, оценивание моментов переключения состояний (датировка поворотных точек), а также оценивание параметров моделей [5].

Циклический характер смены состояний сложных систем обуславливает зависимость классов состояний. Традиционно предполагается, что эта зависимость описывается ненаблюдаемой однородной цепью Маркова. Это приводит к моделям с марковскими переключениями классов состояний (Markov Switching states model – MS model) [6]. В случае неизвестной модели зависимости либо высокой неопределенности относительно состояния системы в рамках марковской модели целесообразно использовать модели с независимыми классами состояний (Independent Switching states model – IS model) [7]. При наличии обучающих реализаций временных рядов, соответствующих различным классам состояний, для оценивания классов состояний могут применяться методы дискриминантного анализа указанных моделей [8]. Для решения перечисленных выше задач анализа при отсутствии классифицированной обучающей выборки активно используются EM-алгоритмы (expectation-maximization algorithms) расщепления смесей распределений наблюдений, описываемых многомерными статистическими моделями [5, 9, 10]. EM-алгоритмы обеспечивают совместное решение таких задач, как классификация многомерных наблюдений, описываемых статистическими моделями, на заданное число классов состояний и оценивание параметров моделей для различных классов состояний. Полученная в результате классификации наблюдений последовательность классов используется да-

лее для установления циклического характера изменения состояния и определения моментов переключения (поворотных точек).

Решение перечисленных выше задач с помощью методов статистической классификации временных рядов может осложняться недостаточной точностью лежащих в их основе статистических решающих правил классификации [10]. Вследствие этого возможно появление значительного числа «ложных сигналов», соответствующих изменениям, приводящим не к смене классов состояний, а к случайным колебаниям состояния системы, которые не должны приниматься во внимание. Доля ложных сигналов при использовании статистических решающих правил классификации ограничена снизу вероятностью ошибки соответствующего байесовского решающего правила, оптимального в смысле минимума вероятности ошибки [2]. Необходимость исключения краткосрочных колебаний может быть также обусловлена ограничениями, имеющими содержательную интерпретацию. Например, при анализе макроэкономических циклов с двумя классами состояния экономики (рост и рецессия) по квартальным временным рядам для перехода из состояния «рост» в состояние «рецессия» может устанавливаться ограничение: не менее двух кварталов пребывания экономической системы в состоянии «рецессия» [8].

Таким образом, при анализе сложных систем с циклической сменой состояний с помощью алгоритмов статистической классификации актуальной является задача тестирования установленных с помощью указанных алгоритмов моментов переключения состояний с целью исключения краткосрочных колебаний, не имеющих содержательной интерпретации. Решение данной задачи с помощью специальных статистических алгоритмов позволяет автоматизировать процесс обработки данных, не прибегая к экспертному анализу. В статье предлагается метод решения данной задачи, описываются реализующие его базовые алгоритмы и приводятся результаты экспериментального исследования эффективности алгоритмов на основе компьютерных модельных экспериментов.

1. Модели RS-VARX и задача тестирования циклических структурных изменений

Пусть сложная система в момент времени t характеризуется случайным вектором наблюдений $\mathbf{y}_t \in \mathfrak{R}^n$, определенным на вероятностном пространстве $(\Omega, \mathfrak{F}, \mathbf{P})$, где Ω – пространство элементарных объектов ($\omega \in \Omega$ – элементарный объект); \mathfrak{F} – σ -алгебра подмножеств из Ω ; \mathbf{P} – вероятностная мера: $\mathbf{P}(A) = \mathbf{P}\{\omega \in A\}$, $A \in \mathfrak{F}$. Пусть $\{\Omega_0, \dots, \Omega_{L-1}\}$ – разбиение Ω на конечное число непустых непересекающихся подмножеств, таких что $\Omega_l \in \mathfrak{F}$, $\mathbf{P}\{\Omega_l\} = \mathbf{P}(\{\omega \in \Omega_l\}) > 0$, $\Omega_l \in \mathfrak{F}$, $\mathbf{P}\{\Omega_l\} = \mathbf{P}(\{\omega \in \Omega_l\}) > 0$, $\bigcup_{l \in S(L)} \Omega_l = \Omega$, $S(L) = \{0, \dots, L-1\}$. Подмножества $\{\Omega_l\} (l \in S(L))$

называются *классами состояний сложной системы*, число которых равно L .

Предполагается, что случайный вектор наблюдений допускает разбиение $\mathbf{y}_t = (\mathbf{x}'_t, \mathbf{z}'_t)' \in \mathfrak{X} \times \mathfrak{Z} \subseteq \mathfrak{R}^n$ (где $'$ – знак транспонирования, $n = N + M$, $N \geq 1$, $M \geq 1$) на подвекторы эндогенных переменных $\mathbf{x}_t = (x_{ij}) \in \mathfrak{X} \subseteq \mathfrak{R}^N$, характеризующих состояние сложной системы, и экзогенных переменных $\mathbf{z}_t = (z_{ik}) \in \mathfrak{Z} \subseteq \mathfrak{R}^M$, описывающих внешние воздействия на систему. В качестве базовой модели наблюдений будем использовать модель векторной авторегрессии с неоднородной циклически изменяющейся эндогенно-экзогенной структурой [4]. В общем случае будем полагать, что временной ряд $\mathbf{y}_t = (\mathbf{x}'_t, \mathbf{z}'_t)'$ описывается моделью RS-VARX $(p)(p \geq 1)$ вида

$$\mathbf{x}_t = \sum_{i=1}^p A_{d(t),i} \mathbf{x}_{t-i} + B_{d(t)} \mathbf{z}_t + \boldsymbol{\eta}_{d(t),t}, \quad t=1, \dots, T, \quad (1)$$

где $\mathbf{x}_{1-p}, \dots, \mathbf{x}_0 \in \mathfrak{R}^N$ – заданные начальные значения; $\boldsymbol{\eta}_{d(t),t} \in \mathfrak{R}^N$ – определенные на $(\Omega, \mathfrak{F}, \mathbf{P})$ случайные величины, соответствующие ошибкам наблюдения; $d(t) \equiv d_t \in S(L) = \{0, \dots, L-1\}$ – номер класса состояния; $\mathbf{D} = (d_t) \in S^T(L)$ – вектор состояний сложной системы на заданном временном интервале.

Относительно модели (2) сделаем следующие предположения:

М.1. Матрицы коэффициентов авторегрессии $\{A_{l,i}\}$ ($i=1,\dots,p$) для каждого класса состояний $l \in S(L)$ удовлетворяют условию стационарности модели VAR(p) [1].

М.2. Ошибки наблюдения $\{\boldsymbol{\eta}_{l,r}\}$ ($t=1,\dots,T$) являются независимыми в совокупности гауссовскими случайными векторами с нулевым вектором математического ожидания и ковариационной матрицей $\Sigma_l : \mathbf{E}\{\boldsymbol{\eta}_{l,r}\} = \mathbf{0}_N \in \mathfrak{R}^N$, $\mathbf{E}\{\boldsymbol{\eta}_{l,r}\boldsymbol{\eta}_{l,s}'\} = \delta_{r,s}\Sigma_l$ ($r,s=1,\dots,T, l \in S(L)$), $\delta_{r,s}$ – символ Кронекера.

М.3. Значения экзогенных переменных $\mathbf{z}_t = (z_{t1}, \dots, z_{tM})' \in \mathfrak{Z} \subseteq \mathfrak{R}^M$ являются заданными.

М.4. Выполняется условие структурной параметрической неоднородности модели, которое означает, что для разных классов состояний $\{\Omega_0, \dots, \Omega_{L-1}\}$ параметры модели частично либо полностью различны: $A_l \neq A_k$ и (или) $B_l \neq B_k \quad \forall k \neq l, k, l \in S(L)$.

При выполнении предположений М.1–М.4 модель RS-VARX вида (1) является кусочно-стационарной [3]. Будем рассматривать кусочно-стационарные модели с L классами состояний, для которых $2 \leq L < s+1$, где $s \geq 1$ – число структурных изменений, соответствующих смене классов состояний, в неизвестные моменты времени $1 < \tau_1 < \dots < \tau_s < T$. Относительно номеров классов состояний $d_t \in S(L)$ ($t=1,\dots,T$) возможны два типа предположений:

d.1. $\{d_t\}$ ($t=1,\dots,T$) – ненаблюдаемые независимые в совокупности случайные величины с распределением вероятностей $P\{d_t=l\} = \pi_l > 0$ ($l \in S(L)$), $\sum_{l \in S(L)} \pi_l = 1$.

d.2. $\{d_t\}$ ($t=1,\dots,T$) – ненаблюдаемая однородная эргодическая цепь Маркова (ОЦМ) с распределением, которое определяется вектором вероятностей начального состояния $\boldsymbol{\pi}$ и матрицей вероятностей одношаговых переходов P соответственно:

$$\boldsymbol{\pi} = (\pi_l), \pi_l = \mathbf{P}\{d_1=l\} > 0 \quad (l \in S(L)), \sum_{l \in S(L)} \pi_l = 1;$$

$$P = (p_{kl}), p_{kl} = \mathbf{P}\{d_{t+1}=l \mid d_t=k\} \geq 0 \quad (k,l \in S(L)), \sum_{l \in S(L)} p_{kl} = 1, k \in S(L).$$

В случае двух классов состояний ($L=2$) будем использовать следующие предположения относительно параметров вероятностной модели для $\{d_t\}$:

$$d.1. \pi_0 = \pi_1 = 0,5; \quad d.2. \pi_0 = \pi_1 = 0,5; \quad P = \begin{pmatrix} 1-\omega & \omega \\ \omega & 1-\omega \end{pmatrix} \quad (0 < \omega < 0,5). \quad (2)$$

При условии d.1 модель (1) будем называть *моделью VARX с независимыми случайными переключениями состояний (IS-VARX)*, а в случае d.2 – *моделью VARX с марковскими переключениями состояний (MS-VARX)*.

Истинные значения параметров $\{A_l, B_l, \Sigma_l\}$ ($l \in S(L)$), $\boldsymbol{\pi}, P$, а также моменты структурных изменений $1 < \tau_1 < \dots < \tau_s < T$ в тестируемом временном ряде $\mathbf{y}_t = (\mathbf{x}'_t, \mathbf{z}'_t)' \in \mathfrak{X} \times \mathfrak{Z} \subseteq \mathfrak{R}^n$ ($t=1,\dots,T$) неизвестны. Имеются обучающие реализации временных рядов для каждого класса состояний $\mathbf{Y}_l = (\mathbf{X}'_l, \mathbf{Z}'_l)' \in \mathfrak{X}^{T^l} \times \mathfrak{Z}^{T^l} \subseteq \mathfrak{R}^{nT^l}$ ($l \in S(L)$), где $\mathbf{X}'_l = (\mathbf{x}'_{l1}, \dots, \mathbf{x}'_{lT_l}) \in \mathfrak{X}^{T_l}$, $\mathbf{Z}'_l = (\mathbf{z}'_{l1}, \dots, \mathbf{z}'_{lT_l}) \in \mathfrak{Z}^{T_l}$, T_l – длина временных рядов для класса Ω_l , $T^L = T_1 + \dots + T_L$ – суммарная длина временных рядов. Будем называть совокупность наблюдений $\{\mathbf{X}_l, \mathbf{Z}_l\}$ ($l \in S(L)$) *классифицированной обучающей выборкой* наблюдений объема T^L .

Имеют место следующие основные задачи анализа рассматриваемых моделей:

1) оценивание вектора состояний сложной системы $\mathbf{D} = (d_i) \in S^T(L)$ для тестируемого временного ряда $\mathbf{y}_t = (\mathbf{x}'_t, \mathbf{z}'_t)' \in \mathcal{X} \times \mathcal{Z} \subseteq \mathfrak{R}^n$ ($t=1, \dots, T$) с помощью решающих правил статистической классификации и нахождения оценок числа \hat{s} и моментов структурных изменений $\{\hat{\tau}_i\}$ ($i=1, \dots, \hat{s}$) на основании полученной оценки $\hat{\mathbf{D}} = (\hat{d}_i) \in S^T(L)$;

2) тестирование оценок моментов структурных изменений $\{\hat{\tau}_i\}$ ($i=1, \dots, \hat{s}$) с целью исключения краткосрочных колебаний состояния сложной системы и определения точек переключения состояний $\{\tilde{\tau}_i\}$ ($i=1, \dots, \tilde{s}$, $\tilde{s} \leq \hat{s}$).

Предполагается, что классифицированная обучающая выборка при ее отсутствии может быть сформирована на основе ретроспективных данных с помощью EM-алгоритмов для рассматриваемых моделей RS-VARX [2] и экспертного анализа. Обучающая выборка $\{\mathbf{X}_l, \mathbf{Z}_l\}$ ($l \in S(L)$) используется для построения «подстановочного» решающего правила классификации и оценивания его условных вероятностей ошибок в рамках задачи 1, которые применяются в предлагаемом методе тестирования краткосрочных колебаний при решении задачи 2.

2. Метод и алгоритмы тестирования структурных изменений в модели RS-VARX

Метод тестирования моментов циклических структурных изменений основан на последовательном решении задач классификации многомерных авторегрессионных наблюдений и тестирования моментов структурных изменений с помощью предлагаемого ниже статистического критерия проверки гипотезы о значении ожидаемой вероятности ошибки классификации для применяемого решающего правила классификации. Детальное описание и исследование используемых решающих правил поточечной и групповой классификации для моделей RS-VARX приводится в [9]. Поэтому далее дается лишь краткое описание применяемого решающего правила, а основное внимание уделяется обоснованию предлагаемого статистического критерия.

2.1. Решающее правило поточечной классификации

Задача поточечной классификации векторных авторегрессионных наблюдений, описываемых моделью (1), заключается в оценивании номера класса состояния сложной системы $d_i \in S(L)$ по наблюдениям $(\mathbf{x}_t, \mathbf{z}_t)$ при заданных значениях экзогенных переменных $\mathbf{x}_{t-p}, \dots, \mathbf{x}_{t-1}$ ($t=1, \dots, T$).

В работе [9] установлено, что если модель (1) удовлетворяет предположениям М.1–М.4, $d.1$, а $\{A_{l,i}\}$, $\{B_l\}$, $\{\pi_l\}$ ($i=1, \dots, p$; $l \in S(L)$) – истинные значения параметров модели, то оптимальное в смысле минимума вероятности ошибки байесовское решающее правило (БРП) поточечной классификации наблюдений (x_t, z_t) ($t=1, \dots, T$) единственное и определяется соотношениями

$$d_i \equiv d(\mathbf{x}_t, \mathbf{z}_t) = \arg \min_{l \in S(L)} \left\{ \text{tr}(\Sigma_l^{-1} S_l) + \ln |\Sigma_l| - 2 \ln \pi_l \right\}, \quad S_l = \bar{\boldsymbol{\eta}}_{l,t} \bar{\boldsymbol{\eta}}'_{l,t}, \quad \bar{\boldsymbol{\eta}}_{l,t} = \mathbf{x}_t - \sum_{i=1}^p A_{l,i} \mathbf{x}_{t-i} - B_l \mathbf{z}_t, \quad (3)$$

где для класса Ω_l $\boldsymbol{\eta}_{l,t} \in \mathfrak{R}^N$ – случайный вектор отклонений в момент t значений эндогенных переменных от модельных значений; S_l – матричная сумма квадратов отклонений.

В предположениях М.1–М.4, $d.1$ для модели (1) могут быть получены состоятельные оценки максимального правдоподобия параметров $\{\hat{A}_{l,i}\}$, $\{\hat{B}_l\}$, $\{\hat{\pi}_l\}$ ($i=1, \dots, p$, $l \in S(L)$) по классифицированной обучающей выборке $\{\mathbf{X}_l, \mathbf{Z}_l\}$ ($l \in S(L)$) [2]. Путем подстановки оценок параметров в БРП (3) получается состоятельное в смысле минимума вероятности ошибки подстановочное байесовское решающее правило (ПБРП) поточечной классификации наблюдений $(\mathbf{x}_t, \mathbf{z}_t)$: $\hat{d}_i \equiv \hat{d}(\mathbf{x}_t, \mathbf{z}_t)$ ($t=1, \dots, T$). В качестве характеристик точности решающих правил классификации используются условные вероятности ошибок для альтернативных классов состоя-

ний. Для простоты описания далее рассматривается случай двух классов, т. е. полагается, что $L=2$, $d_t \in S(2) = \{0, 1\}$ ($t=1, \dots, T$).

Введем обозначения: $r_0^{(l)} = \mathbf{P}\{d_t = 1 - l \mid d_t^0 = l\}$ – условная вероятность ошибки при классификации наблюдений из класса Ω_l ($l \in \{0, 1\}$), где $d_t^0 \in \{0, 1\}$ – истинный номер класса состояния системы в момент времени t ; $r_1^{(l)}$ ($r_1^{(l)} > r_0^{(l)}$) – недопустимое значение вероятности ошибки классификации для класса Ω_l , которое интерпретируется как переход системы в альтернативный класс Ω_{1-l} и, соответственно, определяется по формуле

$$r_1^{(l)} = \mathbf{P}\{d_t = 1 - l \mid d_t^0 = 1 - l\} = 1 - \mathbf{P}\{d_t = l \mid d_t^0 = 1 - l\} = 1 - r_0^{(1-l)}. \quad (4)$$

Условные вероятности ошибок $\{r_0^{(l)}, r_1^{(l)}\}$ ($l \in \{0, 1\}$) неизвестны и могут быть оценены по обучающей выборке $\{\mathbf{X}_l, \mathbf{Z}_l\}$ ($l \in S(L)$) при построении ПБРП [9].

2.2. Алгоритм тестирования моментов циклических структурных изменений

Пусть имеются тестируемый временной ряд $\mathbf{y}_t = (\mathbf{x}'_t, \mathbf{z}'_t)' \in \mathcal{X} \times \mathcal{Z} \subseteq \mathfrak{R}^n$ ($t=1, \dots, T$), описываемый моделью RS-VARX (1), М.1–М.4 в предположениях $d.1$ или $d.2$, и классифицированная обучающая выборка $\{\mathbf{X}_l, \mathbf{Z}_l\}$ ($l \in S(L)$). Предлагаемый алгоритм тестирования структурных изменений основан на использовании ПБРП поточечной классификации, а также оценок его условных вероятностей ошибок по выборке $\{\mathbf{X}_l, \mathbf{Z}_l\}$ ($l \in S(L)$). Он реализуется посредством выполнения следующих этапов:

1. Построение ПБРП и оценивание условных вероятностей ошибок. По обучающей выборке строится ПБРП поточечной классификации $\hat{d}_t \equiv \hat{d}(\mathbf{x}_t, \mathbf{z}_t)$ вида (3). Оценки условных вероятностей ошибок $\{\hat{r}_0^{(l)}\}$ вычисляются по формуле

$$\hat{r}_0^{(l)} = 1 - \frac{1}{T_l} \sum_{t=1}^{T_l} \delta_{l, \hat{d}_t} \quad (l \in S(2)). \quad (5)$$

2. Классификация тестируемого временного ряда. С помощью оцененного на этапе 1 решающего правила ПБРП $\hat{d}(\cdot; \cdot)$ осуществляется поточечная классификация тестируемого временного ряда $\mathbf{y}_t = (\mathbf{x}'_t, \mathbf{z}'_t)' \in \mathcal{X} \times \mathcal{Z} \subseteq \mathfrak{R}^n$ ($t=1, \dots, T$), в результате чего находится оценка вектора состояний сложной системы $\hat{\mathbf{D}} = (\hat{d}_1, \dots, \hat{d}_T)' \in S^T(2)$. Для упрощения записи далее для оценок и истинных значений рассматриваемых характеристик используются одинаковые обозначения $\{r_0^{(l)}\}$ и $\{d_t\}$.

3. Оценивание условных вероятностей ошибок в сериях наблюдений. Случайная последовательность $d_1, \dots, d_T \in \{0, 1\}$ разбивается на $\gamma = T/m$ серий $\mathbf{D}_j = (d_{(j-1)m+1}, \dots, d_{jm})' \in \mathfrak{R}^m$ ($j=1, \dots, \gamma$) фиксированной длины m ($p+1 < m \ll T$). Для простоты описания алгоритма считается, что T кратно m . Будем полагать, что в момент наблюдения серии \mathbf{D}_j Ω_l – текущее состояние системы (исходное), а Ω_{1-l} – альтернативное ($l \in \{0, 1\}$). Для каждой серии \mathbf{D}_j вычисляются число решений в пользу альтернативного класса состояния Ω_{1-l} , обозначаемое $k_j^{(l)}(m) \in \{0, 1, \dots, m\}$, а также доля таких решений (оценка условной вероятности ошибки) $k_{j,m}^{(l)} \in (0, 1)$ по формулам

$$k_{j,m}^{(l)} = \frac{k_j^{(l)}}{m}, \quad k_j^{(l)} = \sum_{i=(j-1)m+1}^{jm} \delta_{d_i, 1-l} \quad (j=1, \dots, \gamma). \quad (6)$$

4. Тестирование моментов структурных изменений. Тестирование осуществляется с помощью статистического критерия на основе сравнения ожидаемой и оцененной условной веро-

ятностей ошибки для текущего состояния системы. Опишем соответствующую задачу статистической проверки гипотез. Пусть Ω_l ($l \in \{0, 1\}$) – текущее состояние системы, $r^{(l)}$ – неизвестное фактическое значение условной вероятности ошибки в пользу альтернативного класса Ω_{1-l} при оценивании состояния системы по серии D_j ; $r_0^{(l)}$ – условная вероятность ошибки ПБРП (3), которая интерпретируется как ожидаемое значение вероятности ошибки для состояния Ω_l ; $r_1^{(l)}$ – недопустимое значение вероятности ошибки для класса Ω_l .

Задача проверки гипотезы об отсутствии циклического структурного изменения, приводящего к изменению состояния системы, формулируется следующим образом: в состоянии системы Ω_l по серии наблюдений D_j ($j=1, \dots, \gamma$) для заданного значения вероятности ошибки первого рода α_0 ($0 < \alpha_0 < 0,5$) проверить гипотезу

$$H_0 : r^{(l)} = r_0^{(l)} \text{ при альтернативе } H_1 : r^{(l)} = r_1^{(l)} > r_0^{(l)}. \quad (7)$$

Имеет место следующая интерпретация гипотез в (7). Если доля решений в пользу альтернативного класса состояния системы Ω_{1-l} соответствует вероятности ошибки решающего правила для класса Ω_l , то гипотеза H_0 не отклоняется и, следовательно, нет оснований предполагать наличие смены состояния на тестируемом отрезке временного ряда. В этом случае считается, что момент переключения состояния $\tau > jm$ и осуществляется переход к тестированию следующей $(j+1)$ -й серии. Если доля решений в пользу альтернативного класса больше условной вероятности ошибки решающего правила для класса Ω_l , то это может быть следствием перехода системы из состояния Ω_l в альтернативное состояние Ω_{1-l} в момент времени $\bar{\tau} \in [(j-1)m+1, \dots, jm]$. В этом случае гипотеза H_0 отклоняется.

5. Выявление ложных сигналов» о смене класса состояния системы. Как отмечалось ранее, в ходе поточечной классификации временных рядов возможно появление «ложных сигналов» о смене класса состояния системы. В рамках предлагаемой процедуры сигнал о смене состояния Ω_l на Ω_{1-l} в некоторый момент времени τ считается ложным, если при тестировании очередной серии наблюдений было принято решение о возврате системы из состояния Ω_{1-l} в состояние Ω_l . Такие изменения состояния системы называются *случайными краткосрочными колебаниями*. В противном случае при сохранении нового состояния системы Ω_{1-l} в течение двух и более серий момент времени τ считается истинным моментом смены класса состояния (моментом циклического структурного изменения). Таким образом, в одной серии наблюдений длины m ($m \ll T$) не может быть более одного циклического структурного изменения, что исключает зависимость классов состояний внутри серий и дает возможность использовать решающее правило (3) поточечной классификации для случая независимых переключений состояний сложной системы.

2.3. Статистический критерий циклических структурных изменений

Сформулируем статистический критерий для решения задачи (7). Обозначим через $B(r_0^{(l)}, m, 1-\alpha_0)$ квантиль уровня $1-\alpha_0$ биномиального закона распределения с параметрами $m, r_0^{(l)}$.

Теорема. Если тестируемый временной ряд $\mathbf{y}_t = (\mathbf{x}'_t, \mathbf{z}'_t)' \in \mathcal{X} \times \mathcal{Z} \subseteq \mathcal{R}^n$ ($t=1, \dots, T$) описывается моделью RS-VARX (1), M.1–M.4 в предположениях d.1 или d.2, а описанный выше алгоритм тестирования циклических структурных изменений удовлетворяет следующим условиям: 1) для поточечной классификации наблюдений $\{\mathbf{x}_t, \mathbf{z}_t\}$ ($t=1, \dots, T$) используется ПБРП вида (3) для независимых классов состояний с условными вероятностями ошибок $\{r_0^{(l)}\}$ ($l \in S(2)$); 2) серии наблюдений $\mathbf{D}_j = (d_{(j-1)m+1}, \dots, d_{jm})' \in \mathcal{R}^m$ ($j=1, \dots, \gamma$) имеют длину m ($p+1 < m \ll T$), исключая возможность существования нескольких циклических структурных изменений

внутри серии, то статистический критерий проверки гипотез H_0, H_1 вида (7), имеющий заданный размер α_0 , формулируется следующим образом:

$$\text{гипотеза } H_0 \begin{cases} \text{не отклоняется, если } k_j^{(l)}(m) < B(r_0^{(l)}, m, 1 - \alpha_0); \\ \text{отклоняется, если } k_j^{(l)}(m) \geq B(r_0^{(l)}, m, 1 - \alpha_0). \end{cases} \quad (8)$$

Доказательство. Число ошибочных решений $k_j^{(l)}(m) \in \{0, 1, \dots, m\} (l \in S(2))$ при оценке состояния системы по серии $\mathbf{D}_j (j=1, \dots, \gamma)$ соответствует числу наступлений некоторого события в серии из m независимых экспериментов при условии, что вероятность наступления события равна некоторой величине $r^{(l)}$. Это означает, что при верной гипотезе H_0 случайная величина $k_j^{(l)}(m) \in \{0, 1, \dots, m\}$ имеет биномиальный закон распределения с параметрами $m, r_0^{(l)}$ ($l \in S(2)$). Тогда задача (7) заключается в проверке гипотезы H_0 о значении параметра биномиального закона распределения при простой альтернативе H_1 по серии наблюдений длиной m . В этом случае в соответствии с определением квантиля уровня $1 - \alpha_0$ распределения случайной величины $k_j^{(l)}(m)$ получаем

$$\mathbf{P}\{k_j^{(l)}(m) \leq B(r_0^{(l)}, m, 1 - \alpha_0) | H_0\} = \mathbf{P}\{H_0 | H_0\} = 1 - \alpha_0,$$

откуда следует, что размер критерия (7) равен $\mathbf{P}\{H_1 | H_0\} = \alpha_0$. ■

Следствие 1. Для заданного значения $1 - \beta_0$ мощности критерия (8) длина серии m находится из уравнения

$$\pi(m) = 1 - \mathbf{P}\{H_0 | H_1\} = \mathbf{P}\{H_1 | H_1\} = \mathbf{P}\{k_j^{(l)}(m) \geq B(r_0^{(l)}, m, 1 - \alpha_0) | H_1\} = 1 - \beta_0. \quad (9)$$

Доказательство. Формула (9) следует из определения мощности критерия. ■

Для вычисления квантилей биномиального закона могут применяться как точные формулы, основанные на использовании бета-распределения [11], так и приближенные формулы, для получения которых применяются различные аппроксимации биномиального распределения. На практике часто удобнее использовать аппроксимацию биномиального распределения нормальным законом. В этом случае асимптотическое распределение случайной величины $k_j^{(l)}(m) (j=1, \dots, \gamma)$ вида (6) при $m \rightarrow \infty$ и верной нулевой гипотезе в (7) согласно формуле Муавра – Лапласа удовлетворяет следующему соотношению [12]:

$$\mathbf{P}\{k_j^{(l)}(m) \leq k/m | H_0\} = \sum_{h=0}^k \binom{m}{h} (r_0^{(l)})^h (1 - r_0^{(l)})^{m-h} = \Phi\left(\frac{k/m - r_0^{(l)} + c}{\sqrt{r_0^{(l)}(1 - r_0^{(l)})/m}}\right) + O\left(\frac{1}{\sqrt{m}}\right), \quad (10)$$

где $\Phi(\cdot)$ – функция распределения стандартного нормального закона.

Следствие 2. Если используется аппроксимация (10) биномиального закона нормальным распределением, для критерия (8) справедливо представление

$$\text{гипотеза } H_0 \begin{cases} \text{не отклоняется, если } k_j^{(l)} < k^{(l)}; \\ \text{отклоняется, если } k_j^{(l)} \geq k^{(l)}, \end{cases} \quad (11)$$

где для заданных значений $\alpha_0, \beta_0 (0 < \alpha_0, \beta_0 < 0,5)$ статистика критерия $k_j^{(l)}$ вычисляется согласно (6), пороговое значение $k^{(l)}$ критерия находится из ограничения на вероятность ошибки первого рода $\mathbf{P}(H_1 | H_0) = \mathbf{P}(d_l = 1 - l | d_l^0 = l) = \alpha_0$ и имеет вид

$$k^{(l)} = r_0^{(l)} + \frac{0,5}{m} + \Phi^{-1}(1 - \alpha_0) \frac{\sigma_0}{\sqrt{m}}, \quad \sigma_0 = \sqrt{r_0^{(l)}(1 - r_0^{(l)})}; \quad (12)$$

длина серии m , необходимая для достижения требуемого значения мощности критерия $\pi(m) = 1 - \mathbf{P}(H_0 | H_1) = 1 - \mathbf{P}(d_i = l | d_i^0 = 1 - l) = 1 - \beta_0$ при заданном значении $r_1^{(l)}$, определяется по формуле

$$m = \left(\frac{\sigma_0 \Phi^{-1}(1 - \alpha_0) + \sigma_1 \Phi^{-1}(1 - \beta_0)}{\rho} \right)^2, \quad \sigma_i = \sqrt{r_i^{(l)}(1 - r_i^{(l)})} (i = 0, 1), \quad \rho = r_1^{(l)} - r_0^{(l)}. \quad (13)$$

Доказательство. С учетом введенных обозначений при использовании аппроксимации (10) выражения (11)–(13) непосредственно следуют из (8) и (9) согласно [12]. ■

Нормальная аппроксимация вида (10) биномиального распределения имеет скорость сходимости $O(1/\sqrt{m})$ и, как отмечается в [11], является удовлетворительной в смысле точности вычисления значений функции распределения, если $mr^{(l)}(1 - r^{(l)}) \geq 9$. В то же время, как показывает экспериментальное исследование предлагаемых алгоритмов [2], нормальная аппроксимация биномиального распределения обеспечивает приемлемую точность вычисления характеристик в контексте решаемой задачи в условиях структурных изменений, приводящих к существенным различиям в значениях параметров модели для различных классов состояний.

2.4. Задание и интерпретация управляемых параметров

Пусть в алгоритме тестирования структурных изменений используется ПБРП с оценками условных вероятностей ошибок $\{\tilde{r}_0^{(l)}\} (l \in \{0, 1\})$ по обучающим выборкам объема $\{T_l^L\} (l \in S(L))$. Выбор значений управляемых параметров описанной выше процедуры основывается на минимаксной стратегии, в соответствии с которой

$$r_0^{(l)} = \begin{cases} \tilde{r}_0^{(l)}, & \text{если } T_l^L > 500, \\ \delta_l^+(q, T_l^L), & \text{если } T_l^L \leq 500; \end{cases} \quad r_1^{(l)} = \begin{cases} 1 - \tilde{r}_0^{(1-l)}, & \text{если } T_{1-l}^L > 500, \\ 1 - \delta_{1-l}^+(q, T_{1-l}^L), & \text{если } T_{1-l}^L \leq 500, \end{cases} \quad (14)$$

где $[\delta_l^-(q, T_l^L), \delta_l^+(q, T_l^L)] (l \in \{0, 1\})$ – доверительные интервалы с доверительной вероятностью q для условных вероятностей ошибок ПБРП $\{r_0^{(l)}\}: \mathbf{P}\{r_0^{(l)} \in [\delta_l^-(q, \tilde{T}_l), \delta_l^+(q, \tilde{T}_l)]\} = q$. Для вычисления границ доверительных интервалов в случае аппроксимации (10) удобно применять приближенную формулу [11]

$$\delta_l^\mp(q, \tilde{T}_l) = \frac{1}{\tilde{T}_l + \phi_q^2} \times \left(k_l + \frac{\phi_q^2}{2} \mp \phi_q \sqrt{\frac{k_l(\tilde{T}_l - k_l) + \phi_q^2}{\tilde{T}_l} + \frac{\phi_q^2}{4}} \right) (l \in S(L)), \quad (15)$$

где k_l – число ошибок при поточечной классификации \tilde{T}_l наблюдений из класса Ω_l ; $\phi_q \equiv \Phi^{-1}(q)$ – квантиль стандартного нормального распределения уровня q .

В контексте рассматриваемой проблемы параметры α_0, β_0 имеют важную содержательную интерпретацию. Размер α_0 критерия (8) определяет среднюю длину серий (СДС) наблюдений между ложными сигналами о структурном изменении $L_0 = 1/\alpha_0$, а значение $1 - \beta_0$ мощности данного критерия – среднее число серий наблюдений, необходимое для обнаружения структурного изменения, если оно наступило, $L_1 = 1/(1 - \beta_0)$. Таким образом, параметры α_0, β_0 задают ограничения соответственно на число ложных сигналов о структурном изменении и скорость обнаружения структурного изменения, определяемую необходимыми для этого значениями СДС. Параметр q существенно влияет на ширину доверительного интервала (15) и согласно (13) – на длину серии m : чем больше q , тем больше значение m . Очевидно, большие значения m могут приводить к более позднему обнаружению структурного изменения в реальном времени. Для рассматриваемых задач целесообразными представляются значения $L_0 \in \{10, 20\}, L_1 \in \{1, 05; 1, 1\}, q \in \{0, 95; 0, 975\}$.

3. Результаты компьютерного моделирования

Для описания циклических изменений временных рядов $\mathbf{y}_t = (\mathbf{x}'_t, \mathbf{z}'_t)' \in \mathfrak{R}^n$, $\mathbf{x}_t = (x_{ij}) \in \mathfrak{X} \subseteq \mathfrak{R}^N$, $\mathbf{z}_t = (z_{ik}) \in \mathfrak{Z} \subseteq \mathfrak{R}^M$ ($n = N + M$, $t = 1, \dots, T$) используется модель MS-VARX с двумя ($L = 2$) чередующимися классами состояний Ω_0 и Ω_1 , которым соответствуют рост и снижение значений анализируемых эндогенных переменных. В задачах анализа циклических изменений экономических систем указанные состояния интерпретируются как фазы роста и спада экономического цикла. Предполагается, что смена состояний происходит в случайные моменты времени $\{\tau_i\}$ ($i = 1, \dots, s$), причем количество s циклических структурных изменений, приводящих к смене состояния системы, априори неизвестно. Для классификации временных рядов $\{y_t\}$ используется алгоритм EM MS-VARX [10].

Приведем результаты численных экспериментов, целью которых является исследование эффективности предлагаемого метода тестирования циклических структурных изменений на тестовых модельных данных.

3.1. Описание тестовых модельных данных

Рассматриваются модели MS-VARX вида (1) с циклическими структурными изменениями в матрице регрессионных коэффициентов. При статистическом моделировании тестовых данных предполагается, что вектор экзогенных переменных $\mathbf{z}_t = (z_{ij}) \in \mathfrak{Z} \subseteq \mathfrak{R}^M$ является случайным и имеет равномерное распределение в области $\mathfrak{Z} = Z^M \in \mathfrak{R}^M$, $Z = [1, 10]$. Рассматриваются различные варианты спецификации модели MS-VARX, различающиеся сложностью задачи обнаружения структурных изменений. Степень сложности задачи характеризуется значением межклассового расстояния $\Delta(\tilde{\mathbf{z}}) = \sqrt{\tilde{\mathbf{z}}'(B_1 - B_0)' \Sigma^{-1} (B_1 - B_0) \tilde{\mathbf{z}}}$, $\tilde{\mathbf{z}} = (5,5; 5,5; 5,5)'$, и параметром ОЦМ ω . В работе [10] было показано, что, если $\Delta(\tilde{\mathbf{z}}) \rightarrow 0$, $\omega \rightarrow 0,5$, задача классификации усложняется и точность алгоритма классификации EM MS-VARX снижается. С учетом этих особенностей используются девять тестовых примеров, определяемых следующими наборами параметров модели:

- размерность модели: $L = 2$, $N = 2$, $M = 3$; число оцениваемых параметров 29;
- матрицы параметров VARX и переходных вероятностей:

$$A_1 = A_0 = \begin{pmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{pmatrix}, \Sigma_0 = \Sigma_1 = \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 5 \end{pmatrix};$$

- матрицы коэффициентов регрессии:

$$B_0 = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 0 & 3 \end{pmatrix}, \quad B_1 = B_0 + H,$$

$$1) H = \begin{pmatrix} 0 & 0 & 0 \\ -0,5 & 0 & 0 \end{pmatrix}, \quad 2) H = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 1 & 1 \end{pmatrix}, \quad 3) H = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & -1 \end{pmatrix},$$

т. е. структурное изменение касается только параметров второго уравнения.

В табл. 1 для вариантов модели, определяемых матрицей H , приводятся значения межклассового расстояния $\Delta(\tilde{\mathbf{z}})$, условных вероятностей ошибок классификации $r_0(\tilde{\mathbf{z}})$ для БРП, не учитывающего марковскую зависимость номеров классов состояний, а также ожидаемое число «ложных сигналов» на 1000.

Таблица 1

Характеристики сложности задачи классификации

Вариант	$\Delta(\tilde{\mathbf{z}})$	$r_0(\tilde{\mathbf{z}})$	Ожидаемое число «ложных сигналов»
1	1,23	0,269	269
2	2,46	0,109	109
3	4,92	0,007	7

3.2. Тестирование моментов смены состояний

Для каждого варианта модели используются по три тестовых примера (ТП), различающихся значениями параметра ОЦМ $\{d_t\}(t=1,\dots,T)$: $\omega \in \{0,02; 0,05; 0,1\}$. В табл. 2 приведены вычисленные по обучающим реализациям временных рядов длиной $T_0^L = T_1^L = 300$ точечные оценки условных вероятностей ошибок $\{\hat{r}_0^{(l)}\}(l \in \{0, 1\})$ и границы доверительных интервалов (15) для доверительной вероятности $q=0,95$. Полученные оценки применяются для расчета длин серий и пороговых значений критериев. При вычислении пороговых значений критерия (8), а также длины серий m полагается $\alpha_0=0,05$ ($L_0=20$), $\beta_0=0,1$ ($L_1=1,1$). В формулах (12), (13) им соответствуют значения квантилей стандартного нормального распределения: $\Phi^{-1}(0,95)=1,64$, $\Phi^{-1}(0,9)=1,29$. Результаты расчетов представлены в табл. 3.

Таблица 2
Оценки вероятностей ошибок и числа циклических структурных изменений

Тестовый пример	δ_0^-	$\hat{r}_0^{(0)}$	δ_0^+	δ_1^-	$\hat{r}_0^{(1)}$	δ_1^+
$\omega = 0,02, T^E = 200, s = 5$						
1.1	0,023	0,041	0,067	0,009	0,025	0,055
2.1	0,000 07	0,003	0,015	0,050	0,017	0,043
3.1	0,000 07	0,003	0,015	0,000	0,000	0,016
$\omega = 0,05, T^E = 300, s = 12$						
1.2	0,056	0,084	0,012	0,033	0,057	0,091
2.2	0,003	0,012	0,032	0,002	0,011	0,031
3.2	0,0008	0,006	0,022	0,020	0,004	0,002
$\omega = 0,1, T^E = 100, s = 11$						
1.3	0,129	0,174	0,226	0,038	0,061	0,091
2.3	0,009	0,024	0,051	0,018	0,035	0,060
3.3	0,002	0,012	0,034	0,000 07	0,003	0,016

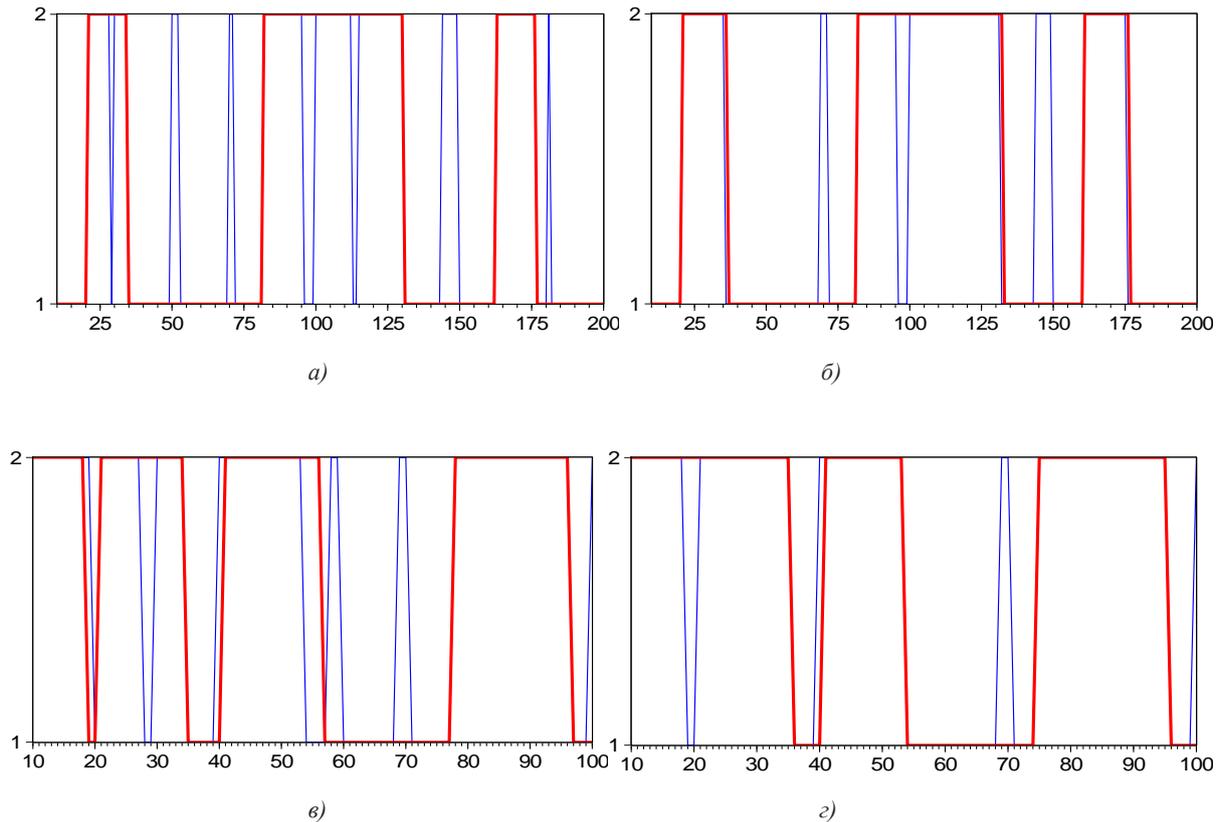
Таблица 3
Расчет длины серии и порогового значения критерия (текущее состояние Ω_0)

Тестовый пример	$r_0^{(0)} \equiv \delta_0^+$	$r_1^{(0)} \equiv 1 - \delta_1^+$	ρ	σ_0	σ_1	$[m]+1$	κ_+^1
$\omega = 0,02, T^E = 200, s = 5$							
1.1	0,067	0,945	0,878	0,25	0,227	4	0,398
2.1	0,015	0,957	0,942	0,122	0,202	4	0,241
3.1	0,015	0,984	0,969	0,122	0,124	4	0,241
$\omega = 0,05, T^E = 300, s = 12$							
1.2	0,121	0,909	0,788	0,325	0,288	4	0,513
2.2	0,032	0,969	0,937	0,175	0,173	4	0,301
3.2	0,022	0,980	0,958	0,148	0,139	4	0,269
$\omega = 0,1, T^E = 100, s = 11$							
1.3	0,226	0,909	0,683	0,418	0,288	4	0,696
2.3	0,051	0,940	0,889	0,220	0,237	4	0,357
2.4	0,034	0,984	0,950	0,182	0,125	4	0,309

Анализ условных вероятностей ошибок ПБРП $\hat{r}_0^{(l)}(l \in \{0, 1\})$ из табл. 3, полученных для различных вариантов значений $\Delta(\bar{z})$ и ω , подтверждает установленную в [10] закономерность: снижение точности классификации при уменьшении межклассового расстояния и ослаблении

зависимости классов наблюдений. В то же время наличие марковской зависимости номеров классов обуславливает достаточно малые значения межклассовых расстояний и минимально допустимую длину серий во всех рассматриваемых экспериментах.

На рисунке показаны результаты работы предлагаемого алгоритма тестирования моментов структурных изменений для четырех из описанных выше тестовых примеров, имеющих место при $\Delta(\tilde{z}) \in \{2,46; 4,92\}$ и $\omega \in \{0,05; 0,1\}$. По оси абсцисс откладывается порядковый номер наблюдения t ($t=1, \dots, T^E$), а по оси ординат – переменные $\hat{d}_t + 1, \tilde{d}_t + 1$ ($t=1, \dots, T^E$), указывающие на класс состояний сложной системы.



Результаты тестирования моментов структурных изменений:

- а) ТП 2.2: $\Delta(\tilde{z}) = 2,46$, $\omega = 0,05$, $\hat{s} = 22$, $\tilde{s} = 6$; б) ТП 3.2: $\Delta(\tilde{z}) = 4,92$, $\omega = 0,05$, $\hat{s} = s = 12$, $\tilde{s} = 6$;
 в) ТП 2.3: $\Delta(\tilde{z}) = 2,46$, $\omega = 0,1$, $\hat{s} = 15$, $\tilde{s} = 6$; г) ТП 3.3: $\Delta(\tilde{z}) = 4,92$, $\omega = 0,1$, $\hat{s} = s = 11$, $\tilde{s} = 6$

Тонкие линии на графиках соответствуют значениям номеров классов $\{\hat{d}_t\}$, полученных с помощью алгоритма EM MS-VARX, жирные линии – номерам классов состояний $\{\tilde{d}_t\}$ ($t=1, \dots, T^E$) после исключения краткосрочных колебаний с помощью предлагаемого алгоритма тестирования. Значения переменных \hat{s} и \tilde{s} указывают на число структурных изменений до и после исключения краткосрочных колебаний.

Заключение

Модели с циклическими переключениями состояний семейства RS-VARX в настоящее время применяются для анализа экономических циклов и циклических изменений в экономических системах, вызванных шокowymi воздействиями. Предлагаемые в статье алгоритмы классификации и тестирования моментов переключения состояний моделируемой системы позволяют отличать краткосрочные колебания в состоянии системы от долгосрочных циклических изменений. В основе алгоритмов лежат подстановочное байесовское решающее правило

поточечной классификации и статистический критерий проверки гипотезы о значении ожидаемой вероятности ошибки классификации.

Список литературы

1. Харин, Ю.С. Эконометрическое моделирование / Ю.С. Харин, В.И. Малюгин, А.Ю. Харин. – Минск : БГУ, 2003. – 318 с.
2. Малюгин, В.И. Методы анализа многомерных эконометрических моделей с неоднородной структурой / В.И. Малюгин. – Минск : БГУ, 2014. – 351 с.
3. Perron, P. Dealing with structural breaks / P. Perron // *Palgrave handbook of econometrics*. – Vol. 1: *Econometric Theory*. – Basingstoke : Palgrave Macmillan, 2006. – P. 278–352.
4. Hamilton, J.D. Regime switching models / J.D. Hamilton // *New Palgrave Dictionary of Economics*. – 2nd edition. – Basingstoke : Palgrave Macmillan, 2008. – P. 1755–1804.
5. Krolzig, H.-M. Markov switching vector autoregressions. Modelling statistical inference and application to business cycle analysis / H.-M. Krolzig. – Berlin : Springer, 1997. – 166 p.
6. Hamilton, J.D. A new approach to the economic analysis of nonstationary time series and the business cycle / J.D. Hamilton // *Econometrica*. – 1989. – Vol. 57(2). – P. 357–384.
7. Малюгин, В.И. Об оптимальности классификации случайных наблюдений, различающихся уравнениями регрессии / В.И. Малюгин, Ю.С. Харин // *Автоматика и телемеханика*. – 1986. – № 7. – С. 61–69.
8. Малюгин, В.И. Дискриминантный анализ многомерных авторегрессионных моделей с неоднородной структурой / В.И. Малюгин // *Известия НАН Беларуси. Сер. 1: Физ. Мат. Информ.* – 2013. – № 3. – С. 43–53.
9. Малюгин, В.И. Статистический анализ смесей распределений регрессионных наблюдений / В.И. Малюгин // *Информатика*. – 2008. – № 4(20). – С. 79–88.
10. Малюгин, В.И. Анализ многомерных статистических моделей с неоднородной структурой в случае скрытой марковской зависимости состояний / В.И. Малюгин, А.Ю. Новопольцев // *Известия НАН Беларуси. Сер. 1: Физ. Мат. Информ.* – 2015. – № 2. – С. 26–36.
11. Большев, Л.Н. Таблицы математической статистики / Л.Н. Большев, Н.В. Смирнов. – М. : Наука, 1983. – 416 с.
12. Браунли, К.А. Статистический анализ и методология в науке и технике / К.А. Браунли. – М. : Наука, 1977. – 407 с.

Поступила 31.07.2015

*НИИ прикладных проблем математики
и информатики Белорусского
государственного университета,
Минск, пр. Независимости, 4
e-mail: Malugin@bsu.by*

V.I. Malugin

TESTING OF CYCLIC STRUCTURAL CHANGES IN SWITCHING REGIME VECTOR AUTOREGRESSIVE MODELS

For vector autoregressive models RS-VARX with cyclic regime switching of states the method of excluding of short-term system state fluctuations is proposed. The method is based on a sequential application of two algorithms, realizing the Bayesian “plug-in” decision rule of point wise classification and a statistical test for expected probability of misclassification. Accuracy of the approach is examined by means of computer simulation experiments.

УДК 517.958:537.8

В.Т. Ерофеев

МОДЕЛЬ ВЫЧИСЛЕНИЯ ЭФФЕКТИВНЫХ ПАРАМЕТРОВ МАТРИЧНОГО КОМПОЗИТА ИЗ БИИЗОТРОПНЫХ ЧАСТИЦ С УЧЕТОМ МНОГОКРАТНЫХ ПЕРЕОТРАЖЕНИЙ ЭЛЕКТРОМАГНИТНОГО ПОЛЯ

Разрабатывается метод вычисления эффективных материальных параметров матричного композита, состоящего из магнитодиэлектрической проводящей матрицы с множеством равномерно и случайно распределенных биизотропных сферических частиц одинаковых радиусов. При расчете параметров, учитывающем многократные рассеяния поля между частицами, используется новый тип теорем сложения, связывающих базисные сферические электромагнитные поля, отнесенные к разным частицам. Разработанная методика позволяет рассчитывать эффективные параметры для композитов с ансамблем несоприкасающихся частиц.

Введение

В настоящее время актуальным является исследование электродинамических свойств композитных материалов различных типов, представляющих собой структурно-неоднородные материалы, состоящие из однородной магнитопроводящей матрицы с большим числом структурных включений. Включения (частицы) характеризуются геометрической формой, химическим составом и размерами по отношению к длине электромагнитной волны, воздействующей на материал. Представляет интерес случай, когда частицы также выполнены из композитных материалов. В последнее время проводятся исследования метаматериалов, киральных, биизотропных и бианизотропных сред [1–3]. В частности, исследуются композиты со сферическими частицами [4–7]. Один из методов моделирования электромагнитных свойств неоднородных композитов сводится к замене структурно-неоднородного материала однородным материалом с сохранением характерных свойств исходного композита. Переход к однородному материалу базируется на использовании различных методов усреднения полей и материальных параметров композита. Усредненные материальные параметры, называемые эффективными параметрами, с высокой точностью характеризуют неоднородный композит. В статье [6] разработана модель вычисления эффективных параметров равномерно распределенных биизотропных частиц, расположенных в вакууме. В [7] вычисляются эффективные параметры композита с биизотропными частицами, расположенными в магнитопроводящей матрице. Учитывается однократное рассеяние поля на частицах. В книге [8] обсуждаются статистические модели композитов с использованием многократных рассеяний поля на частицах из обычных материалов. В настоящей статье разработана аналитическая модель матричного композита со сферическими биизотропными частицами с учетом многократных переотражений поля между частицами на основе теории дифракции волн на многих телах [9]. Алгоритм вычисления эффективных параметров композита представлен в виде последовательности аналитических формул. Для построения математической модели композита используются следующие основные принципы:

- рассматривается ансамбль сферических биизотропных частиц с одинаковыми радиусами и комплексными материальными параметрами;
- модель композита строится в рамках биизотропной среды;
- используется оригинальный вариант теорем сложения для вычисления взаимных переотражений полей между частицами;
- при вычислении эффективных параметров композита в разложениях полей учитываются только дипольные слагаемые;
- усреднение напряженностей и индукций полей производится по окрестности центральной частицы;
- эффективные параметры вычисляются для бесконечно протяженной композитной среды без учета внешних границ.

1. Структура матричного композита

Рассмотрим матрицу, заполненную однородным магнитодиэлектрическим материалом с диэлектрической и магнитной проницаемостями $\epsilon_M = \epsilon_m \epsilon_0$, $\mu_M = \mu_m \mu_0$. В матрице распределены случайным образом сферические частицы D_k ($k=1, 2, \dots$) радиуса R из биизотропного материала с проницаемостями ϵ, μ и параметрами биизотропности G, Z . Радиус частиц $R < \lambda_0/30$, где λ_0 – длина волны в вакууме [6]. Частицу с номером $k=1$, $D_1 = D_R$ поместим в начало O декартовой системы координат $Oxyz$. Пусть $Or\theta\varphi$ – сферические координаты; \vec{r}_k^0 – вектор, направленный в центр O_k частицы D_k ; $\{r_k^0, \theta_k^0, \varphi_k^0\}$ – сферические координаты точки O_k . Введем обозначения: D_p ($0 \leq r \leq P$) – шаровая область достаточно большого радиуса P ($R \ll P$); N – число частиц в области D_p ; $V_R = \frac{4\pi}{3} R^3$ – объем частицы D_R ; V_p – объем области D_p ; $v = \frac{N}{V_p}$ – концентрация частиц (число частиц в единице объема матрицы); $\tau = vV_R$ – объемный коэффициент заполнения матрицы сферическими частицами (суммарный объем сферических частиц в единице объема матрицы).

Определим радиус $R_\tau > R$ с помощью соотношения $\frac{V_R}{V_{R_\tau}} = \tau$, тогда $R_\tau = R/\sqrt[3]{\tau}$ при заданном τ . Опишем вокруг шара D_R сферу Γ_τ радиуса R_τ . Обозначим: D_τ – шар внутри сферы Γ_τ ; $D_{R\tau} = D_\tau/D_R$ – шаровой слой вокруг частицы D_R ; $D_{\tau p} = D_p/D_\tau$ – шаровой слой, который содержит частицы D_k с номерами $k=2, 3, \dots, N$; $D_p^0 = D_p \setminus \bigcup_{k=1}^N D_k$ – матрица без сферических частиц.

Будем считать, что в матричном композите распространяется плоское первичное поле $\vec{E}^{(0)}, \vec{H}^{(0)}$, колеблющееся с круговой частотой $\omega = 2\pi f$. Поле присутствует в области D_p^0 и возбуждает внутри частиц D_k поля $\vec{E}_k^{(1)}, \vec{H}_k^{(1)}$, в частности $\vec{E}_R^{(1)}, \vec{H}_R^{(1)}$ внутри D_R . В области D_p^0 образуется суммарное поле $\vec{E}^{(1)'}, \vec{H}^{(1)'}$, отраженное от всех частиц:

$$\vec{E}^{(1)'} = \sum_{k=1}^N \vec{E}_k^{(1)'}, \quad \vec{H}^{(1)'} = \sum_{k=1}^N \vec{H}_k^{(1)'},$$

где $\vec{E}_k^{(1)'}, \vec{H}_k^{(1)'}$ – поле, отраженное от частицы D_k .

Поле $\vec{E}^{(1)'}, \vec{H}^{(1)'}$ является первым суммарным отражением от частиц композита, которое снова воздействует на частицы D_k . В результате образуется второе отраженное поле $\vec{E}_k^{(2)'}, \vec{H}_k^{(2)'}$ от частицы D_k и суммарное $\vec{E}^{(2)'}, \vec{H}^{(2)'}$. Внутри частицы D_k образуется второе поле $\vec{E}_k^{(2)}, \vec{H}_k^{(2)}$, в частности $\vec{E}_R^{(2)}, \vec{H}_R^{(2)}$ внутри D_R . Суммарное поле $\vec{E}^{(2)'}, \vec{H}^{(2)'}$ снова воздействует на частицы и образуется третье отражение $\vec{E}^{(3)'}, \vec{H}^{(3)'}$ и поле $\vec{E}_k^{(3)}, \vec{H}_k^{(3)}$ внутри D_k и т. д. В итоге получим последовательность отражений $\vec{E}^{(l)'}, \vec{H}^{(l)'}$ и последовательность полей $\vec{E}_k^{(l)}, \vec{H}_k^{(l)}$, $l=1, 2, \dots, L$. Суммируя все отражения, получим результирующее электромагнит-

ное поле $\vec{E}'_{pez} = \sum_{l=1}^L \vec{E}^{(l)'}$, $\vec{H}'_{pez} = \sum_{l=1}^L \vec{H}^{(l)'}$ в L -м приближении в области D_p^0 и результирующее поле $\vec{E}_{pez} = \sum_{l=1}^L \vec{E}_R^{(l)}$, $\vec{H}_{pez} = \sum_{l=1}^L \vec{H}_R^{(l)}$ внутри центральной частицы D_R .

2. Постановка задач

Один из методов моделирования электродинамических свойств неоднородных композитных сред, представляющих собой однородную материальную матрицу, включающую частицы из других материалов, состоит в замене неоднородной среды эквивалентной однородной средой. При этом материальные параметры частиц и матрицы заменяются на постоянные эффективные материальные параметры композита в целом. Имеются различные методы вычисления эффективных параметров. В статье ставится задача построения алгоритма определения эффективных параметров $\varepsilon_{\text{эф}} = \varepsilon_3 \varepsilon_0$, $\mu_{\text{эф}} = \mu_3 \mu_0$, $G_{\text{эф}} = G_3/c$, $Z_{\text{эф}} = Z_3/c$ в рамках биизотропной однородной среды, так как частицы композита выполнены из биизотропных материалов. Для решения данной задачи воспользуемся аналитическим решением краевой задачи дифракции на изолированной сферической частице D_R , размещенной в бесконечно протяженной матрице [2, с. 272]:

$$\begin{aligned} \text{rot } \vec{E}_R^{(1)} &= i\omega(\mu \vec{H}_R^{(1)} + Z \vec{E}_R^{(1)}), \quad \text{rot } \vec{H}_R^{(1)} = -i\omega(\varepsilon \vec{E}_R^{(1)} + G \vec{H}_R^{(1)}), \quad 0 \leq r < R; \\ \text{rot } \vec{E}_R^{(1)'} &= i\omega \mu_M \vec{H}_R^{(1)'}, \quad \text{rot } \vec{H}_R^{(1)'} = -i\omega \varepsilon_M \vec{E}_R^{(1)'}, \quad r > R; \\ (\vec{E}_R^{(1)})_{\tau}|_{r=R} &= (\vec{E}^{(0)} + \vec{E}_R^{(1)'})_{\tau}|_{r=R}, \quad (H_R^{(1)})_{\tau}|_{r=R} = (\vec{H}^{(0)} + \vec{E}_R^{(1)'})_{\tau}|_{r=R}. \end{aligned} \quad (1)$$

Решение задачи (1) представим в виде рядов по сферическим базисным полям:

– первичное поле

$$\begin{aligned} \vec{E}^{(0)} &= \sum_{n,m} [a_{mn}^{(0)} \vec{m}_{mn}(\vec{r}, k_M) + b_{mn}^{(0)} \vec{n}_{mn}(\vec{r}, k_M)], \\ \vec{H}^{(0)} &= h_M \sum_{n,m} [a_{mn}^{(0)} \vec{n}_{mn}(\vec{r}, k_M) + b_{mn}^{(0)} \vec{m}_{mn}(\vec{r}, k_M)], \end{aligned} \quad (2)$$

где $a_{mn}^{(0)}, b_{mn}^{(0)}$ – заданные коэффициенты, $\sum_{n,m} = \sum_{n=1}^{\infty} \sum_{m=-n}^n$, $h_M = \frac{k_M}{i\omega \mu_M}$, $k_M = \omega \sqrt{\varepsilon_M \mu_M}$,

$0 < \arg k_M < \pi$;

– отраженное поле

$$\begin{aligned} \vec{E}_R^{(1)'} &= \sum_{n,m} [\bar{x}_{mn}^{(1)} \vec{m}_{mn}(\vec{r}, k_M) + \bar{y}_{mn}^{(1)} \vec{n}_{mn}(\vec{r}, k_M)], \quad r > R, \\ \vec{H}_R^{(1)'} &= h_M \sum_{n,m} [\bar{x}_{mn}^{(1)} \vec{n}_{mn}(\vec{r}, k_M) + \bar{y}_{mn}^{(1)} \vec{m}_{mn}(\vec{r}, k_M)]; \end{aligned} \quad (3)$$

– поле внутри частицы

$$\begin{aligned} \vec{E}_R^{(1)} &= \sum_{n,m} [x_{mn}^{(1)} \vec{K}_{mn}^{(1)}(\vec{r}, k_1) + y_{mn}^{(1)} \vec{K}_{mn}^{(2)}(\vec{r}, k_2)], \quad 0 \leq r < R, \\ \vec{H}_R^{(1)} &= \sum_{n,m} [x_{mn}^{(1)} p_1 \vec{K}_{mn}^{(1)}(\vec{r}, k_1) + y_{mn}^{(1)} p_2 \vec{K}_{mn}^{(2)}(\vec{r}, k_2)], \end{aligned} \quad (4)$$

где

$$\begin{aligned}\bar{K}_{mm}^{(j)} &= \bar{n}_{mm}(\bar{r}, k_j) - q_j \bar{m}_{mm}(\bar{r}, k_j), \quad j=1, 2, \\ k_j &= \omega \bar{k}_j, \quad \bar{k}_j = \sqrt{\varepsilon \mu - \frac{1}{2}(G^2 + Z^2) + i(G - Z)f_j}, \quad 0 \leq \arg \bar{k}_j < \pi, \\ f_j &= (-1)^j f_0, \quad f_0 = \sqrt{\varepsilon \mu - \frac{1}{4}(G + Z)^2}, \quad 0 \leq \arg f_0 < \pi, \\ g_j &= f_j - \frac{i}{2}(G - Z), \quad q_j = \frac{\varepsilon \mu - ZG}{\bar{k}_j g_j}, \quad p_j = \frac{1}{\mu} \left(i \frac{(\varepsilon \mu - ZG)}{g_j} - Z \right),\end{aligned}$$

а коэффициенты $x_{mm}^{(1)}, y_{mm}^{(1)}, \bar{x}_{mm}^{(1)}, \bar{y}_{mm}^{(1)}$ выражаются через величины $a_{mm}^{(0)}, b_{mm}^{(0)}$ разложения (2). В дальнейшем будут использоваться коэффициенты при $n=1, m=0, \pm 1$, которые определяются формулами [7]

$$\bar{X}_m^{(1)} = \hat{M} \bar{a}_m^{(0)}, \quad \bar{X}_m^{(1)'} = \hat{T} \bar{a}_m^{(0)}, \quad m=0, \pm 1, \quad (5)$$

где

$$\begin{aligned}\hat{M} &= -\frac{i}{\xi_M^2} \hat{P}^{-1}, \quad \hat{T} = -(j_1(\xi_M) \hat{E} + \hat{Q} \hat{M}) / h_1^{(1)}(\xi_M), \\ \bar{X}_m^{(1)} &= \begin{pmatrix} x_{m1}^{(1)} \\ y_{m1}^{(1)} \end{pmatrix}, \quad \bar{X}_m^{(1)'} = \begin{pmatrix} \bar{x}_{m1}^{(1)} \\ \bar{y}_{m1}^{(1)} \end{pmatrix}, \quad \bar{a}_m^{(0)} = \begin{pmatrix} a_{m1}^{(0)} \\ b_{m1}^{(0)} \end{pmatrix}, \quad \hat{Q} = \begin{pmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{pmatrix}, \quad \hat{P} = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix}, \quad \hat{E} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \\ q_{11} &= q_1 j_1(\xi_1), \quad q_{12} = q_2 j_1(\xi_2), \quad q_{21} = \bar{p}_1 q_1 j_1(\xi_1), \quad q_{22} = \bar{p}_2 q_2 j_1(\xi_2); \\ p_{11} &= \bar{p}_1 g_1(\xi_1) h_1^{(1)}(\xi_M) + q_1 j_1(\xi_1) g_1^{(1)}(\xi_M), \quad p_{12} = \bar{p}_2 g_1(\xi_2) h_1^{(1)}(\xi_M) + q_2 j_1(\xi_2) g_1^{(1)}(\xi_M), \\ p_{21} &= g_1(\xi_1) h_1^{(1)}(\xi_M) + \bar{p}_1 q_1 j_1(\xi_1) g_1^{(1)}(\xi_M), \quad p_{22} = g_1(\xi_2) h_1^{(1)}(\xi_M) + \bar{p}_2 q_2 j_1(\xi_2) g_1^{(1)}(\xi_M); \\ \xi_j &= k_j R, \quad \xi_M = k_M R, \quad \bar{p}_j = p_j / h_M, \\ h_0^{(1)}(x) &= -\frac{i}{x} e^{ix}, \quad h_1^{(1)}(x) = -\left(\frac{1}{x} + \frac{i}{x^2} \right) e^{ix}, \quad j_1(x) = \frac{\sin x}{x^2} - \frac{\cos x}{x}; \\ g_1(x) &= \frac{1}{x} \left(\frac{\cos x}{x} + \left(1 - \frac{1}{x^2} \right) \sin x \right), \quad g_1^{(1)}(x) = \frac{1}{x} \left(\frac{i}{x^2} + \frac{1}{x} - i \right) e^{ix}.\end{aligned} \quad (6)$$

Выражения (6) представляют собой сферические функции Бесселя.

3. Теоремы сложения для сферических базисных полей

Для расчета полей, рассеянных на сферических частицах, используются базисные сферические поля, которые записываются аналитически в различных формах. Приведем аналитические формулы специального вида, которые будут использоваться в дальнейшем.

Теорема 1. *Сингулярные сферические базисные поля представляются в декартово-сферических координатах следующим образом:*

$$\begin{aligned} \tilde{n}_{mn}(\vec{r}, k) &= \frac{1}{2(2n+1)} \left\{ \left[n\tilde{U}_{n+1}^{m+1}(\vec{r}, k) - (n+1)\tilde{U}_{n-1}^{m+1}(\vec{r}, k) \right] \vec{e}_- + \right. \\ &+ \left[n(n-m+1)(n-m+2)\tilde{U}_{n+1}^{m-1}(\vec{r}, k) - (n+1)(n+m)(n+m-1)\tilde{U}_{n-1}^{m-1}(\vec{r}, k) \right] \vec{e}_+ + \\ &\left. + 2 \left[n(n-m+1)\tilde{U}_{n+1}^m(\vec{r}, k) + (n+1)(n+m)\tilde{U}_{n-1}^m(\vec{r}, k) \right] \vec{e}_z \right\}, \\ \tilde{m}(\vec{r}, k) &= \frac{i}{2} \left\{ \tilde{U}_n^{m+1}(\vec{r}, k) \vec{e}_- - (n+m)(n-m+1)\tilde{U}_n^{m-1}(\vec{r}, k) \vec{e}_+ - 2m\tilde{U}_n^m(\vec{r}, k) \vec{e}_z \right\}, \\ n &= 1, 2, \dots, \quad m = 0, \pm 1, \pm 2, \dots, \pm n, \end{aligned} \quad (7)$$

где $\tilde{U}_n^m(\vec{r}, k) = h_n^{(1)}(kr) P_n^m(\cos \theta) e^{im\varphi}$, $\vec{e}_+ = i\vec{e}_y + \vec{e}_x$, $\vec{e}_- = i\vec{e}_y - \vec{e}_x$.

Доказательство. Выведем первую формулу (7). Имеем представление [2, с. 118]

$$\tilde{n}_{mn} = \left[\frac{n(n+1)}{kr} h_n^{(1)}(kr) P_n^m(\cos \theta) \vec{e}_r + g_n^{(1)}(kr) \left(\frac{\partial}{\partial \theta} P_n^m(\cos \theta) \vec{e}_\theta + \frac{im}{\sin \theta} P_n^m(\cos \theta) \vec{e}_\varphi \right) \right] e^{im\varphi}. \quad (8)$$

Подставляя формулы

$$\vec{e}_r = \frac{1}{2} \sin \theta (e^{-i\varphi} \vec{e}_+ - e^{i\varphi} \vec{e}_-) + \cos \theta \vec{e}_z, \quad \vec{e}_\theta = \frac{1}{2} \cos \theta (e^{-i\varphi} \vec{e}_+ - e^{i\varphi} \vec{e}_-) - \sin \theta \vec{e}_z,$$

$$\vec{e}_\varphi = -\frac{i}{2} (e^{-i\varphi} \vec{e}_+ + e^{i\varphi} \vec{e}_-), \quad \frac{1}{x} h_n^{(1)}(x) = \sigma_n (h_{n-1}^{(1)}(x) + h_{n+1}^{(1)}(x)),$$

$$g_n^{(1)} = \sigma_n \left((n+1)h_{n-1}^{(1)}(x) - nh_{n+1}^{(1)}(x) \right), \quad \sigma_n = \frac{1}{2n+1}$$

в (8) и группируя слагаемые, получим соотношение

$$\tilde{n}_{mn} = \frac{1}{2} \left(A^{(+)} e^{i(m-1)\varphi} \vec{e}_+ - A^{(-)} e^{i(m+1)\varphi} \vec{e}_- + A^{(0)} e^{im\varphi} \vec{e}_z \right), \quad (9)$$

где

$$A^{(-)} = \sigma_n \left((n+1)G_{mn}^{(-)} h_{n-1}^{(1)} + nZ_{mn}^{(-)} h_{n+1}^{(1)} \right),$$

$$A^{(+)} = \sigma_n \left((n+1)G_{mn}^{(+)} h_{n-1}^{(1)} + nZ_{mn}^{(+)} h_{n+1}^{(1)} \right),$$

$$A^{(0)} = 2\sigma_n \left((n+1)G_{mn}^{(0)} h_{n-1}^{(1)} + nZ_{mn}^{(0)} h_{n+1}^{(1)} \right);$$

$$G_{mn}^{(-)} = (n+1) \sin \theta P_n^m(\cos \theta) + \frac{\partial}{\partial \theta} (\cos \theta P_n^m(\cos \theta)) - \sigma_n (\alpha + \beta) \frac{m}{\sin \theta} P_n^m(\cos \theta),$$

$$Z_{mn}^{(-)} = n \sin \theta P_n^m(\cos \theta) - \frac{\partial}{\partial \theta} (\cos \theta P_n^m(\cos \theta)) + \frac{m}{\sin \theta} P_n^m(\cos \theta),$$

$$G_{mn}^{(+)} = (n+1) \sin \theta P_n^m(\cos \theta) + \frac{\partial}{\partial \theta} (\cos \theta P_n^m(\cos \theta)) + \sigma_n (\bar{\alpha} + \bar{\beta}) \frac{m}{\sin \theta} P_n^m(\cos \theta), \quad (10)$$

$$Z_{mn}^{(+)} = n \sin \theta P_n^m(\cos \theta) - \frac{\partial}{\partial \theta} (\cos \theta P_n^m(\cos \theta)) - \frac{m}{\sin \theta} P_n^m(\cos \theta),$$

$$G_{mn}^{(0)} = n \cos \theta P_n^m(\cos \theta) - \sin \theta \frac{\partial}{\partial \theta} P_n^m(\cos \theta), \quad Z_{mn}^{(0)} = (n+1) \cos \theta P_n^m(\cos \theta) + \sin \theta \frac{\partial}{\partial \theta} P_n^m(\cos \theta),$$

$$\sigma_n(\alpha + \beta) = 1, \quad \alpha = n - m, \quad \beta = n + m + 1, \quad \sigma_n(\bar{\alpha} + \bar{\beta}) = 1, \quad \bar{\alpha} = 3n + m + 2, \quad \bar{\beta} = -(n + m + 1).$$

Для упрощения выражений (10) воспользуемся серией рекуррентных формул [2, с. 287]

$$\begin{aligned} \frac{\partial}{\partial \theta} (\cos \theta P_n^m(\cos \theta)) &= \frac{\sigma_n}{2} \left[(n + m) P_{n-1}^{m+1}(\cos \theta) - (n + m)(n + m - 1)(n - m) P_{n-1}^{m-1}(\cos \theta) + \right. \\ &\quad \left. + (n - m + 1) P_{n+1}^{m+1}(\cos \theta) - (n - m + 1)(n + m + 1)(n - m + 2) P_{n+1}^{m-1}(\cos \theta) \right], \\ \sin \theta P_n^m(\cos \theta) &= \sigma_n \left[P_{n-1}^{m+1}(\cos \theta) - P_{n+1}^{m+1}(\cos \theta) \right]. \end{aligned}$$

Для вычисления $G_{mn}^{(-)}$ при коэффициенте α используется формула

$$\frac{m}{\sin \theta} P_n^m(\cos \theta) = -\frac{1}{2} \left[(n + m)(n + m - 1) P_{n-1}^{m-1}(\cos \theta) + P_{n-1}^{m+1}(\cos \theta) \right],$$

а при коэффициенте β –

$$\frac{m}{\sin \theta} P_n^m(\cos \theta) = -\frac{1}{2} \left[(n - m + 1)(n - m + 2) P_{n+1}^{m-1}(\cos \theta) + P_{n+1}^{m+1}(\cos \theta) \right],$$

аналогично и для коэффициентов $\bar{\alpha}, \bar{\beta}$ при вычислении $G_{mn}^{(+)}$.

В результате после сокращения слагаемых получим упрощенные формулы

$$G_{mn}^{(-)} = P_{n-1}^{m+1}(\cos \theta), \quad G_{mn}^{(+)} = -(n + m)(n + m - 1) P_{n-1}^{m-1}(\cos \theta).$$

Далее воспользуемся соотношениями

$$\begin{aligned} G_{mn}^{(-)} + Z_{mn}^{(-)} &= (2n + 1) \sin \theta P_n^m(\cos \theta) = P_{n-1}^{m+1}(\cos \theta) - P_{n+1}^{m+1}(\cos \theta), \\ G_{mn}^{(+)} + Z_{mn}^{(+)} &= (2n + 1) \sin \theta P_n^m(\cos \theta) = \\ &= (n - m + 1)(n - m + 2) P_{n+1}^{m-1}(\cos \theta) - (n + m)(n + m - 1) P_{n-1}^{m-1}(\cos \theta). \end{aligned}$$

Тогда

$$Z_{mn}^{(-)} = -P_{n+1}^{m+1}(\cos \theta), \quad Z_{mn}^{(+)} = (n - m + 1)(n - m + 2) P_{n+1}^{m-1}(\cos \theta).$$

Аналогично преобразуем

$$\begin{aligned} G_{mn}^{(0)} &= \sigma_n \left[n \left((n + m) P_{n-1}^m + (n - m + 1) P_{n+1}^m \right) - \left(n(n - m + 1) P_{n+1}^m - (n + 1)(n + m) P_{n-1}^m \right) \right] = (n + m) P_{n-1}^m(\cos \theta), \\ Z_{mn}^{(0)} &= (n - m + 1) P_{n+1}^m(\cos \theta). \end{aligned}$$

Подставляя преобразованные формулы (10) в (9), получим первую формулу (7). Для вывода второй формулы (7) имеем представление

$$\begin{aligned} \tilde{m}_{mn} &= h_n^{(1)}(kr) \left(\frac{im}{\sin \theta} P_n^m(\cos \theta) \vec{e}_\theta - \frac{\partial}{\partial \theta} P_n^m(\cos \theta) \vec{e}_\varphi \right) \sin^{im\varphi} = \\ &= \frac{i}{2} h_n^{(1)} \left[\left(\frac{m \cos \theta}{\sin \theta} P_n^m + \frac{\partial}{\partial \theta} P_n^m \right) e^{-i\varphi} \vec{e}_+ + \left(\frac{\partial}{\partial \theta} P_n^m - \frac{m \cos \theta}{\sin \theta} P_n^m \right) e^{i\varphi} \vec{e}_- - 2m P_n^m \vec{e}_z \right] e^{im\varphi}. \end{aligned}$$

Воспользуемся формулами [2, с. 117, с. 287]

$$\frac{\partial}{\partial \theta} P_n^m(\cos \theta) = \frac{1}{2} \left(P_{n-1}^{m+1}(\cos \theta) - (n + m)(n - m + 1) P_n^{m-1}(\cos \theta) \right),$$

$$\frac{m \cos \theta}{\sin \theta} P_n^m(\cos \theta) = -\frac{1}{2} \left(P_n^{m+1}(\cos \theta) + (n+m)(n-m+1) P_n^{m-1}(\cos \theta) \right).$$

В результате получим вторую формулу (7). ■

Для решения задач дифракции электромагнитных волн на нескольких сферах используются теоремы сложения, связывающие базисные сферические поля, отнесенные к сдвинутым относительно друг друга системам сферических координат [2, с. 150]. Рассмотрим основную систему координат $O r \theta \varphi$ и сдвинутую систему $O_1 r_1 \theta_1 \varphi_1$, где точка O_1 имеет сферические координаты r_0, θ_0, φ_0 в системе O . В системе $O_1 r_1 \theta_1 \varphi_1$ рассмотрим базисные сферические поля $\tilde{n}_{mn}(\vec{r}_1, k), \tilde{m}_{mn}(\vec{r}_1, k)$, которые представим через базисные поля специального вида в системе координат $O r \theta \varphi$.

Теорема 2. Сингулярные базисные сферические поля в системе сферических координат $O_1 r_1 \theta_1 \varphi_1$ представляются в виде рядов через регулярные базисные поля в смешанных декартово-сферических координатах для системы координат $O x y z$, сдвинутой относительно первой системы координат:

$$\tilde{n}_{mn}(\vec{r}_1, k) = \sum_{s=0}^{\infty} \sum_{p=-s}^s \left(A_{ps}^{+mn}(\vec{r}_0, k) \vec{e}_+ + A_{ps}^{-mn}(\vec{r}_0, k) \vec{e}_- + A_{ps}^{0mn}(\vec{r}_0, k) \vec{e}_z \right) U_s^p(\vec{r}, k), \quad (11)$$

$$\tilde{m}_{mn}(\vec{r}_1, k) = \sum_{s=0}^{\infty} \sum_{p=-s}^s \left(B_{ps}^{+mn}(\vec{r}_0, k) \vec{e}_+ + B_{ps}^{-mn}(\vec{r}_0, k) \vec{e}_- + B_{ps}^{0mn}(\vec{r}_0, k) \vec{e}_z \right) U_s^p(\vec{r}, k), \quad 0 \leq r < r_0,$$

$$n = 1, 2, \dots, \quad m = 0, \pm 1, \pm 2, \dots, \pm n,$$

где

$$U_s^p(\vec{r}, k) = j_s(kr) P_s^p(\cos \theta) e^{ip\varphi}, \quad x = r \sin \theta \cos \varphi, \quad y = r \sin \theta \sin \varphi, \quad z = r \cos \theta,$$

$$A_{ps}^{+mn} = \frac{\sigma_n}{2} \left[n(n-m+1)(n-m+2) C_{ps}^{-m-1, n+1}(r_0, \theta_0) - (n+1)(n+m)(n+m-1) C_{ps}^{-m-1, n-1}(r_0, \theta_0) \right] e^{i(m-p-1)\varphi_0},$$

$$A_{ps}^{-mn} = \frac{\sigma_n}{2} \left[n C_{ps}^{-m+1, n+1}(r_0, \theta_0) - (n+1) C_{ps}^{-m+1, n-1}(r_0, \theta_0) \right] e^{i(m-p+1)\varphi_0},$$

$$A_{ps}^{0mn} = \sigma_n \left[n(n-m+1) C_{ps}^{-m, n+1}(r_0, \theta_0) + (n+1)(n+m) C_{ps}^{-m, n-1}(r_0, \theta_0) \right] e^{i(m-p)\varphi_0}, \quad (12)$$

$$B_{ps}^{+mn} = -\frac{i}{2} (n+m)(n-m+1) C_{ps}^{-m-1, n}(r_0, \theta_0) e^{i(m-p-1)\varphi_0},$$

$$B_{ps}^{-mn} = \frac{i}{2} C_{ps}^{-m+1, n}(r_0, \theta_0) e^{i(m-p+1)\varphi_0}, \quad B_{ps}^{0mn} = -im C_{ps}^{-mn}(r_0, \theta_0) e^{i(m-p)\varphi_0};$$

$$C_{ps}^{-mn}(r_0, \theta_0) = \frac{(2s+1)(s-p)!}{(s+p)!} \sum_{\sigma=|s-n|}^{s+n} i^{\sigma+s-n} b_{\sigma}^{(nmsp)} (-1)^{\sigma} h_{\sigma}^{(1)}(kr_0) P_{\sigma}^{m-p}(\cos \theta_0), \quad (13)$$

$$C_{ps}^{-mn}(r_0, \theta_0) = 0 \quad \text{при} \quad |m| > n.$$

Доказательство. Воспользуемся теоремой сложения для базисных сферических решений скалярного уравнения Гельмгольца [10, с. 196]:

$$\begin{aligned} \tilde{U}_n^m(\vec{r}_1, k) &= \sum_{s=0}^{\infty} \sum_{p=-s}^s \bar{C}_{ps}^{mn}(r_0, \theta_0) e^{i(m-p)\varphi_0} U_s^p(\vec{r}, k), \quad 0 \leq r < r_0, \\ n &= 0, 1, 2, \dots, \quad m = 0, \pm 1, \pm 2, \dots \pm n. \end{aligned} \quad (14)$$

Заметим, что в формуле (13) присутствует дополнительный множитель $(-1)^\sigma$ по сравнению с [10, с. 196]. Это связано с тем, что теорема сложения (14) раскладывает в ряд функции $\tilde{U}_n^m(\vec{r}_1, k)$ в системе координат O_1 по функциям $U_s^p(\vec{r}, k)$ в системе координат O . Подставим ряд (14) в (7), тогда

$$\begin{aligned} \tilde{n}_{mn}(\vec{r}_1, k) &= \sum_{s=0}^{\infty} \sum_{p=-s}^s \frac{\sigma_n}{2} \left\{ \left[n C_{ps}^{-m+1, n+1} - (n+1) C_{ps}^{-m+1, n-1} \right] e^{i(m-p+1)\varphi_0} \vec{e}_- + \right. \\ &+ \left. \left[n(n-m+1)(n-m+2) C_{ps}^{-m-1, n+1} - (n+1)(n+m)(n+m-1) C_{ps}^{-m-1, n-1} \right] e^{i(m-p-1)\varphi_0} \vec{e}_+ + \right. \\ &\left. + 2 \left[n(n-m+1) C_{ps}^{-m, n+1} + (n+1)(n+m) C_{ps}^{-m, n-1} \right] e^{i(m-p)\varphi_0} \vec{e}_z \right\}. \end{aligned}$$

Отсюда получаем требуемые выражения для коэффициентов (12). Аналогично доказывается вторая теорема сложения (11). Заметим, что формулы (11) отличаются по виду записи от аналогичных теорем сложения [2, с. 150]. ■

4. Суммарное поле, отраженное от частиц

Рассмотрим сферическую частицу D_k , расположенную в области D_p . При воздействии первичного поля (2) на частицу D_k образуется отраженное поле (3), которое запишем в локальной системе координат $O_k r_k \theta_k \varphi_k$ с центром в точке O_k :

$$\begin{aligned} \vec{E}_k^{(1)'} &= \sum_{n,m} \left[\bar{x}_{mn}^{(1)} \tilde{m}_{mn}(\vec{r}_k, k_M) + \bar{y}_{mn}^{(1)} \tilde{n}_{mn}(\vec{r}_k, k_M) \right], \quad r_k > R, \\ \vec{H}_k^{(1)'} &= h_M \sum_{n,m} \left[\bar{x}_{mn}^{(1)} \tilde{n}_{mn}(\vec{r}_k, k_M) + \bar{y}_{mn}^{(1)} \tilde{m}_{mn}(\vec{r}_k, k_M) \right]. \end{aligned} \quad (15)$$

Суммируя поля (15), получим суммарное поле, отраженное от всех частиц в области D_p :
 $\vec{E}^{(1)} = \sum_{k=2}^N \vec{E}_k^{(1)'}$, $\vec{H}^{(1)} = \sum_{k=2}^N \vec{H}_k^{(1)'}$. Здесь исключено поле $\vec{E}_1^{(1)'} = \vec{E}_R^{(1)'}$, $\vec{H}_1^{(1)'} = \vec{H}_R^{(1)'}$, отраженное от центральной частицы D_R .

Представим поля (15) в единой сферической системе координат $O r \theta \varphi$, подставляя (11) в (15). При этом в (15) оставим слагаемые с номерами $n = 1, m = 0, \pm 1$, так как в статье разрабатывается дипольная модель материала. Слагаемые при $n = 1, m = 0, \pm 1$ описывают электрические и магнитные диполи частицы D_k , другие слагаемые определяют квадруполь и т. д., которые опустим. Волновое число k заменим на k_M . В результате

$$\begin{aligned} \vec{E}_k^{(1)'} &= \sum_{s=0}^{\infty} \sum_{p=-s}^s \left(A_{ps}^{(+)}(\vec{r}_k^0, k_M) \vec{e}_+ + A_{ps}^{(-)}(\vec{r}_k^0, k_M) \vec{e}_- + A_{ps}^{(0)}(\vec{r}_k^0, k_M) \vec{e}_z \right) U_s^p(\vec{r}, k_M), \\ \vec{H}_k^{(1)'} &= h_M \sum_{s=0}^{\infty} \sum_{p=-s}^s \left(B_{ps}^{(+)}(\vec{r}_k^0, k_M) \vec{e}_+ + B_{ps}^{(-)}(\vec{r}_k^0, k_M) \vec{e}_- + B_{ps}^{(0)}(\vec{r}_k^0, k_M) \vec{e}_z \right) U_s^p(\vec{r}, k_M), \end{aligned} \quad (16)$$

где

$$\begin{aligned}
 A_{ps}^{(+)} &= \sum_{m=-1}^1 \left(\bar{x}_{m1}^{(1) + m1} B_{ps}(\vec{r}_k^0, k_M) + \bar{y}_{m1}^{(1) + m1} A_{ps}(\vec{r}_k^0, k_M) \right), \\
 A_{ps}^{(-)} &= \sum_{m=-1}^1 \left(\bar{x}_{m1}^{(1) - m1} B_{ps}(\vec{r}_k^0, k_M) + \bar{y}_{m1}^{(1) - m1} A_{ps}(\vec{r}_k^0, k_M) \right), \\
 A_{ps}^{(0)} &= \sum_{m=-1}^1 \left(\bar{x}_{m1}^{(1) 0 m1} B_{ps}(\vec{r}_k^0, k_M) + \bar{y}_{m1}^{(1) 0 m1} A_{ps}(\vec{r}_k^0, k_M) \right), \\
 B_{ps}^{(+)} &= \sum_{m=-1}^1 \left(\bar{y}_{m1}^{(1) + m1} B_{ps}(\vec{r}_k^0, k_M) + \bar{x}_{m1}^{(1) + m1} A_{ps}(\vec{r}_k^0, k_M) \right), \\
 B_{ps}^{(-)} &= \sum_{m=-1}^1 \left(\bar{y}_{m1}^{(1) - m1} B_{ps}(\vec{r}_k^0, k_M) + \bar{x}_{m1}^{(1) - m1} A_{ps}(\vec{r}_k^0, k_M) \right), \\
 B_{ps}^{(0)} &= \sum_{m=-1}^1 \left(\bar{y}_{m1}^{(1) 0 m1} B_{ps}(\vec{r}_k^0, k_M) + \bar{x}_{m1}^{(1) 0 m1} A_{ps}(\vec{r}_k^0, k_M) \right).
 \end{aligned} \tag{17}$$

Суммируя поля (16) по индексу k от $k = 2$ до $k = N$, вычислим суммарное поле, отраженное от частиц D_k . Рассматривая суммы как интегральные, заменим их на интегралы по области $D_{\tau p}$:

$$\begin{aligned}
 \vec{E}^{(1)} &= \sum_{s,p} \left(\vec{a}_{ps}^{(+)} \vec{e}_+ + \vec{a}_{ps}^{(-)} \vec{e}_- + \vec{a}_{ps}^{(0)} \vec{e}_z \right) U_s^p(\vec{r}, k_M), \\
 \vec{H}^{(1)} &= h_M \sum_{s,p} \left(\vec{b}_{ps}^{(+)} \vec{e}_+ + \vec{b}_{ps}^{(-)} \vec{e}_- + \vec{b}_{ps}^{(0)} \vec{e}_z \right) U_s^p(\vec{r}, k_M),
 \end{aligned} \tag{18}$$

где

$$\begin{aligned}
 \vec{a}_{ps}^{(+),(-),(0)} &= \sum_{k=2}^N A_{ps}^{(+),(-),(0)}(\vec{r}_k^0, k_M) \approx \nu \int_{D_{\tau p}} A_{ps}^{(+),(-),(0)}(\vec{r}, k_M) dV, \\
 \vec{b}_{ps}^{(+),(-),(0)} &= \sum_{k=2}^N B_{ps}^{(+),(-),(0)}(\vec{r}_k^0, k_M) \approx \nu \int_{D_{\tau p}} B_{ps}^{(+),(-),(0)}(\vec{r}, k_M) dV.
 \end{aligned} \tag{19}$$

Вычислим интегралы (19), используя соотношения (17), (12):

$$\begin{aligned}
 \vec{a}_{ps}^{(+)} &= \nu \sum_{m=-1}^1 \left(\bar{x}_{m1}^{(1)} \int_{D_{\tau p}}^{+ m1} B_{ps} dV + \bar{y}_{m1}^{(1)} \int_{D_{\tau p}}^{+ m1} A_{ps} dV \right) = \\
 &= \pi \nu \sum_{m=-1}^1 \left[i(m+1)(m-2) \bar{x}_{m1}^{(1)} I_{ps}^{m-1,1} + \frac{1}{3} \bar{y}_{m1}^{(1)} \left((m-2)(m-3) I_{ps}^{m-1,2} - 2m(m+1) I_{ps}^{m-1,0} \right) \right] \delta_{0,m-p-1}, \\
 \vec{a}_{ps}^{(-)} &= \pi \nu \sum_{m=-1}^1 \left[i \bar{x}_{m1}^{(1)} I_{ps}^{m+1,1} + \frac{1}{3} \bar{y}_{m1}^{(1)} \left(I_{ps}^{m+1,2} - 2 I_{ps}^{m+1,0} \right) \right] \delta_{0,m-p+1}, \\
 \vec{a}_{ps}^{(0)} &= -2\pi \nu \sum_{m=-1}^1 \left[i m \bar{x}_{m1}^{(1)} I_{ps}^{m,1} + \frac{1}{3} \bar{y}_{m1}^{(1)} \left((m-2) I_{ps}^{m,2} - 2(m+1) I_{ps}^{m,0} \right) \right] \delta_{0,m-p};
 \end{aligned} \tag{20}$$

$$b_{ps}^{(+)} = \pi v \sum_{m=-1}^1 \left[i(m+1)(m-2)\bar{y}_{m1}^{(1)} I_{ps}^{m-1,1} + \frac{1}{3} \bar{x}_{m1}^{(1)} \left((m-2)(m-3) I_{ps}^{m-1,2} - 2m(m+1) I_{ps}^{m-1,0} \right) \right] \delta_{0,m-p-1},$$

$$b_{ps}^{(-)} = \pi v \sum_{m=-1}^1 \left[i\bar{y}_{m1}^{(1)} I_{ps}^{m+1,1} + \frac{1}{3} \bar{x}_{m1}^{(1)} \left(I_{ps}^{m+1,2} - 2I_{ps}^{m+1,0} \right) \right] \delta_{0,m-p+1},$$

$$b_{ps}^{(0)} = -2\pi v \sum_{m=-1}^1 \left[im \bar{y}_{m1}^{(1)} I_{ps}^{m,1} + \frac{1}{3} \bar{x}_{m1}^{(1)} \left((m-2) I_{ps}^{m,2} - 2(m+1) I_{ps}^{m,0} \right) \right] \delta_{0,m-p},$$

где

$$I_{ps}^{mn} = \int_{R_c}^{\pi} \int_0^{\pi} \bar{C}_{ps}^{mn}(r, \theta) r^2 \sin \theta d\theta dr, \quad \delta_{nm} = \begin{cases} 1, & n = m, \\ 0, & n \neq m. \end{cases}$$

В частном случае $s = 0, p = 0$ коэффициенты (20) имеют достаточно простой вид:

$$a_{00}^{(+)} = 2\pi v \left[-i\bar{x}_{11}^{(1)} I_{00}^{01} + \frac{1}{3} \bar{y}_{11}^{(1)} \left(I_{00}^{02} - 2I_{00}^{00} \right) \right],$$

$$a_{00}^{(-)} = \pi v \left[i\bar{x}_{-11}^{(1)} I_{00}^{01} + \frac{1}{3} \bar{y}_{-11}^{(1)} \left(I_{00}^{02} - 2I_{00}^{00} \right) \right],$$

$$a_{00}^{(0)} = \frac{4\pi}{3} v \bar{y}_{01}^{(1)} \left(I_{00}^{02} + I_{00}^{00} \right); \quad (21)$$

$$b_{00}^{(+)} = 2\pi v \left[-i\bar{y}_{11}^{(1)} I_{00}^{01} + \frac{1}{3} \bar{x}_{11}^{(1)} \left(I_{00}^{02} - 2I_{00}^{00} \right) \right],$$

$$b_{00}^{(-)} = \pi v \left[i\bar{y}_{-11}^{(1)} I_{00}^{01} + \frac{1}{3} \bar{x}_{-11}^{(1)} \left(I_{00}^{02} - 2I_{00}^{00} \right) \right],$$

$$b_{00}^{(0)} = \frac{4\pi}{3} v \bar{x}_{01}^{(1)} \left(I_{00}^{02} + I_{00}^{00} \right).$$

Кроме представления полей (18) рассмотрим также представления через регулярные базисные сферические поля в виде

$$\vec{E}^{(1)} = \sum_{n,m} \left[a_{mn}^{(1)} \vec{m}_{mn}(\vec{r}, k_M) + b_{mn}^{(1)} \vec{n}_{mn}(\vec{r}, k_M) \right], \quad (22)$$

$$\vec{H}^{(1)} = h_M \sum_{n,m} \left[a_{mn}^{(1)} \vec{n}_{mn}(\vec{r}, k_M) + b_{mn}^{(1)} \vec{m}_{mn}(\vec{r}, k_M) \right].$$

Лемма 1. В рамках разрабатываемой дипольной модели для коэффициентов разложений (22) при $n = 1, m = 0, \pm 1$ имеют место формулы

$$a_{m1}^{(1)} \approx \bar{\tau} \bar{f}_M \bar{x}_{m1}^{(1)}, \quad b_{m1}^{(1)} \approx \bar{\tau} \bar{f}_M \bar{y}_{m1}^{(1)}, \quad (23)$$

где

$$\bar{f}_M = \frac{3}{(\xi_M)^3} (k_M R_c + i) e^{ik_M R_c}.$$

Доказательство. Используя (13) и формулу [10, с. 232] для вычисления коэффициентов $b_{\sigma}^{(m_1 n_1 m_2 n_2)}$, получим $I_{00}^{01} = 0, I_{00}^{02} = 0,$

$$I_{00}^{00} = 2 \int_{R_\tau}^P h_0^{(1)}(k_M r) r^2 dr = -\frac{2}{k_M^3} (k_M r + i) e^{ik_M r} \Big|_{R_\tau}^P.$$

Учитывая, что $e^{ik_M P} \approx 0$ при $P \rightarrow \infty$ для проводящей среды, получим приближенную формулу

$$I_{00}^{00} \approx \frac{2}{k_M^3} (k_M R_\tau + i) e^{ik_M R_\tau} = \frac{2}{3} R^3 \bar{f}_M.$$

В результате из (21) следует

$$\begin{aligned} a_{00}^{(+)} &= -\frac{2\tau}{3} \bar{f}_M y_{11}^{(1)}, & a_{00}^{(-)} &= -\frac{\tau}{3} \bar{f}_M y_{-11}^{(1)}, & a_{00}^{(0)} &= \frac{2\tau}{3} \bar{f}_M y_{01}^{(1)}, \\ b_{00}^{(+)} &= -\frac{2\tau}{3} \bar{f}_M x_{11}^{(1)}, & b_{00}^{(-)} &= -\frac{\tau}{3} \bar{f}_M x_{-11}^{(1)}, & b_{00}^{(0)} &= \frac{2\tau}{3} \bar{f}_M x_{01}^{(1)}. \end{aligned} \quad (24)$$

Определим значения полей (18), (22) в начале координат ($r = 0$) и сравним результаты. Учитывая формулы (8) из работы [7, с. 31], получим соотношения

$$\begin{aligned} \vec{E}^{(l)}(0) &= a_{00}^{(+)} \vec{e}_+ + a_{00}^{(-)} \vec{e}_- + a_{00}^{(0)} \vec{e}_z = \frac{1}{3} (-2b_{11}^{(l)} \vec{e}_+ - b_{11}^{(l)} \vec{e}_- + 2b_{01}^{(l)} \vec{e}_z), \\ \vec{H}^{(l)}(0) &= h_M (b_{00}^{(+)} \vec{e}_+ + b_{00}^{(-)} \vec{e}_- + b_{00}^{(0)} \vec{e}_z) = \frac{h_M}{3} (-2a_{11}^{(l)} \vec{e}_+ - a_{-11}^{(l)} \vec{e}_- + 2a_{01}^{(l)} \vec{e}_z). \end{aligned}$$

Приравнивая коэффициенты при векторах $\vec{e}_+, \vec{e}_-, \vec{e}_z$ и используя (24), придем к требуемым формулам (23). ■

Лемма 2. Имеют место обобщения формул (5), (23) на поля последовательности отражений $\vec{E}^{(l)'}, H^{(l)'}; \vec{E}_R^{(l)}, \vec{H}_R^{(l)}; \vec{E}^{(l)}, \vec{H}^{(l)}$ для шара D_R :

$$\vec{X}_m^{(l+1)} = \hat{M} \vec{a}_m^{(l)}, \quad \vec{X}_m^{(l+1)'} = \hat{T} \vec{a}_m^{(l)'}, \quad \vec{a}_m^{(l)} = \tau \bar{f}_M \hat{E} \vec{X}_m^{(l)'}, \quad (25)$$

где

$$\vec{a}_m^{(l)} = \begin{pmatrix} a_{m1}^{(l)} \\ b_{m1}^{(l)} \end{pmatrix}, \quad \vec{X}_m^{(l)'} = \begin{pmatrix} \bar{x}_{m1}^{(l)} \\ \bar{y}_{m1}^{(l)} \end{pmatrix}, \quad \vec{X}_m^{(l)} = \begin{pmatrix} x_{m1}^{(l)} \\ y_{m1}^{(l)} \end{pmatrix}.$$

Для получения формул (25) в формулах (5), (23) верхний индекс заменяется на индекс l для l -й итерации, $l = 0, 1, 2, \dots, L$. ■

5. Усреднение электрического и магнитного полей

В области D_τ рассмотрим электромагнитное поле, которое образовалось при воздействии первичного поля (2) и L взаимных переотражений от частиц композита:

$$\vec{E}_{\text{сум}} = \begin{cases} \vec{E}_{\text{nad}} + \vec{E}_{\text{omp}} & \text{в } D_{R\tau}, \\ \vec{E}_{\text{вн}} & \text{в } D_R, \end{cases} \quad \vec{H}_{\text{сум}} = \begin{cases} \vec{H}_{\text{nad}} + \vec{H}_{\text{omp}} & \text{в } D_{R\tau}, \\ \vec{H}_{\text{вн}} & \text{в } D_R, \end{cases} \quad (26)$$

где $E_{\text{nad}} = \sum_{l=0}^{L-1} \vec{E}^{(l)}$ – суммарное падающее поле на частицу D_R ; $\vec{E}_{o\ m\ p} = \sum_{l=1}^L \vec{E}_R^{(l)'}$ – суммарное

поле, отраженное от частицы D_R ; $\vec{E}_{\text{вн}} = \sum_{l=1}^L \vec{E}_R^{(l)}$ – суммарное поле внутри области D_R .

Представим поля (26) в виде рядов

$$\vec{E}_{\text{nad}} = \sum_{n,m} [a_{nm} \vec{m}_{nm}(\vec{r}, k_M) + b_{nm} \vec{n}_{nm}(\vec{r}, k_M)],$$

$$\begin{aligned}
 \vec{H}_{na\delta} &= h_M \sum_{n,m} [a_{mn} \vec{n}_{mn}(\vec{r}, k_M) + b_{mn} \vec{m}_{mn}(\vec{r}, k_M)]; \\
 \vec{E}_{omp} &= \sum_{n,m} [x'_{mn} \vec{\tilde{m}}_{mn}(\vec{r}, k_M) + y'_{mn} \vec{\tilde{n}}_{mn}(\vec{r}, k_M)], \\
 \vec{H}_{omp} &= h_M \sum_{n,m} [x'_{mn} \vec{\tilde{n}}_{mn}(\vec{r}, k_M) + y'_{mn} \vec{\tilde{m}}_{mn}(\vec{r}, k_M)]; \\
 \vec{E}_{\theta n} &= \sum_{n,m} [x_{mn} \vec{K}_{mn}^{(1)}(\vec{r}, k_1) + y_{mn} \vec{K}_{mn}^{(2)}(\vec{r}, k_2)], \\
 \vec{H}_{\theta n} &= \sum_{n,m} [x_{mn} p_1 \vec{K}_{mn}^{(1)}(\vec{r}, k_1) + y_{mn} p_2 \vec{K}_{mn}^{(2)}(\vec{r}, k_2)],
 \end{aligned} \tag{27}$$

где

$$a_{mn} = \sum_{l=0}^{L-1} a_{mn}^{(l)}, \quad b_{mn} = \sum_{l=0}^{L-1} b_{mn}^{(l)}, \quad x'_{mn} = \sum_{l=1}^L \bar{x}_{mn}^{(l)}, \quad y'_{mn} = \sum_{l=1}^L \bar{y}_{mn}^{(l)}, \quad x_{mn} = \sum_{l=1}^L x_{mn}^{(l)}, \quad y_{mn} = \sum_{l=1}^L y_{mn}^{(l)}. \tag{28}$$

Соотношения (28) запишем в векторном виде при $n=1, m=0, \pm 1$:

$$\vec{a}_m = \sum_{l=0}^{L-1} \vec{a}_m^{(l)}, \quad \vec{X}'_m = \sum_{l=1}^L \vec{X}'_m{}^{(l)}, \quad \vec{X}_m = \sum_{l=1}^L \vec{X}_m^{(l)}, \tag{29}$$

где

$$\vec{X}'_m = \begin{pmatrix} x'_{m1} \\ y'_{m1} \end{pmatrix}, \quad \vec{X}_m = \begin{pmatrix} x_{m1} \\ y_{m1} \end{pmatrix}, \quad \vec{a}_m = \begin{pmatrix} a_{m1} \\ b_{m1} \end{pmatrix},$$

Усредним поля (26) по объему D_τ :

$$\begin{aligned}
 \vec{E} &= \frac{1}{V_{R_\tau}} \int_{D_\tau} \vec{E}_{cym} dv = \frac{1}{V_{R_\tau}} \left[\int_{D_{R_\tau}} (\vec{E}_{na\delta} + \vec{E}_{omp}) dv + \int_{D_R} \vec{E}_{\theta n} dv \right], \\
 \vec{H} &= \frac{1}{V_{R_\tau}} \int_{D_\tau} \vec{H}_{cym} dv = \frac{1}{V_{R_\tau}} \left[\int_{D_{R_\tau}} (\vec{H}_{na\delta} + \vec{H}_{omp}) dv + \int_{D_R} \vec{H}_{\theta n} dv \right].
 \end{aligned} \tag{30}$$

Вычислим интегралы, входящие в (30). Для этого воспользуемся леммами 2, 3 и формулами (29) из статьи [7]. Учитывая (27), получим

$$\begin{aligned}
 \int_{D_{R_\tau}} \vec{E}_{na\delta} dv &= V_{R_\tau} g_M(R, R_\tau) (2b_{11} \vec{e}_+ + b_{-11} \vec{e}_- - 2b_{01} \vec{e}_z), \\
 \int_{D_{R_\tau}} \vec{E}_{omp} dv &= V_{R_\tau} f_M(R, R_\tau) (2y'_{11} \vec{e}_+ + y'_{-11} \vec{e}_- - 2y'_{01} \vec{e}_z), \\
 \int_{D_R} \vec{E}_{\theta n} dv &= -V_{R_\tau} \left[(2(\bar{F}_1 x_{11} + \bar{F}_2 y_{11}) \vec{e}_+ + (\bar{F}_1 x_{-11} + \bar{F}_2 y_{-11}) \vec{e}_- - 2(\bar{F}_1 x_{01} + \bar{F}_2 y_{01}) \vec{e}_z) \right],
 \end{aligned} \tag{31}$$

где

$$\begin{aligned}
 g_M(R, R_\tau) &= \frac{1}{\xi_\tau} \left(\left(\frac{R}{R_\tau} \right)^2 j_1(\xi_M) - j_1(\xi_\tau) \right), \\
 f_M(R, R_\tau) &= \frac{1}{\xi_\tau} \left(\left(\frac{R}{R_\tau} \right)^2 h_1^{(1)}(\xi_M) - h_1^{(1)}(\xi_\tau) \right), \quad \xi_M = k_M R, \quad \xi_\tau = k_M R_\tau, \\
 \bar{F}_j &= \frac{1}{\xi_j} \left(\frac{R}{R_\tau} \right)^3 j_1(\xi_j), \quad \xi_j = k_j R.
 \end{aligned}$$

Аналогично получим

$$\begin{aligned} \int_{D_{R_\tau}} \vec{H}_{nad} dv &= V_{R_\tau} h_M g_M (R, R_\tau) (2a_{11} \vec{e}_+ + a_{-11} \vec{e}_- - 2a_{01} \vec{e}_z), \\ \int_{D_{R_\tau}} \vec{H}_{omp} dv &= V_{R_\tau} h_M f_M (R, R_\tau) (2x'_{11} \vec{e}_+ + x'_{-11} \vec{e}_- - 2x'_{01} \vec{e}_z), \end{aligned} \quad (32)$$

$$\int_{D_R} \vec{H}_{вн} dv = -V_{R_\tau} \left[(2(\bar{F}_1 p_1 x_{11} + \bar{F}_2 p_2 y_{11}) \vec{e}_+ + (\bar{F}_1 p_1 x_{-11} + \bar{F}_2 p_2 y_{-11}) \vec{e}_- - 2(\bar{F}_1 p_1 x_{01} + \bar{F}_2 p_2 y_{01}) \vec{e}_z) \right].$$

Теорема 3. Усредненные по области D_τ электрические и магнитные поля

$$\begin{aligned} \vec{E} &= -E_- \vec{e}_+ - E_+ \vec{e}_- + E_z \vec{e}_z = E_x \vec{e}_x + E_y \vec{e}_y + \vec{E}_z \vec{e}_z, \\ \vec{H} &= -H_- \vec{e}_+ - H_+ \vec{e}_- + H_z \vec{e}_z = H_x \vec{e}_x + H_y \vec{e}_y + H_z \vec{e}_z, \\ E_+ &= \frac{1}{2}(iE_y + E_x), \quad E_- = \frac{1}{2}(iE_y - E_x), \quad H_+ = \frac{1}{2}(iH_y + H_x), \quad H_- = \frac{1}{2}(iH_y - H_x), \end{aligned}$$

в матричном композите со сферическими биизотропными частицами радиуса R , распределенными в проводящей матрице, определяются с помощью соотношений

$$\begin{aligned} E_+ &= -[g_M b_{-11} + f_M y'_{-11} - (\bar{F}_1 x_{-11} + \bar{F}_2 y_{-11})], \\ E_- &= -2[g_M b_{11} + f_M y'_{11} - (\bar{F}_1 x_{11} + \bar{F}_2 y_{11})], \\ E_z &= -2[g_M b_{01} + f_M y'_{01} - (\bar{F}_1 x_{01} + \bar{F}_2 y_{01})]; \\ H_+ &= -[h_M (g_M a_{-11} + f_M x'_{-11}) - (\bar{F}_1 p_1 x_{-11} + \bar{F}_2 p_2 y_{-11})], \\ H_- &= -2[h_M (g_M a_{11} + f_M x'_{11}) - (\bar{F}_1 p_1 x_{11} + \bar{F}_2 p_2 y_{11})], \\ H_z &= -2[h_M (g_M a_{01} + f_M x'_{01}) - (\bar{F}_1 p_1 x_{01} + \bar{F}_2 p_2 y_{01})] \end{aligned}$$

или в векторном виде

$$\begin{aligned} \vec{V}_+ &= -[\hat{H}(g_M \vec{a}_{-1} + f_M \vec{X}'_{-1}) - \hat{F} \vec{X}_{-1}], \\ \vec{V}_- &= -2[\hat{H}(g_M \vec{a}_1 + f_M \vec{X}'_1) - \hat{F} \vec{X}_1], \\ \vec{V}_z &= -2[\hat{H}(g_M \vec{a}_0 + f_M \vec{X}'_0) - \hat{F} \vec{X}_0], \end{aligned} \quad (33)$$

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

Для доказательства достаточно подставить (31), (32) в (30) и воспользоваться обозначениями (29). ■

6. Усреднение электрической и магнитной индукций

Рассмотрим выражения для индукций в области D_τ :

$$\vec{D}_{сум} = \begin{cases} \varepsilon_M (\vec{E}_{nad} + \vec{E}_{omp}) & \text{в } D_{R_\tau}, \\ \varepsilon \vec{E}_{вн} + G \vec{H}_{вн} & \text{в } D_R, \end{cases} \quad \vec{B}_{сум} = \begin{cases} \mu_M (\vec{H}_{nad} + \vec{H}_{omp}) & \text{в } D_{R_\tau}, \\ \mu \vec{H}_{вн} + Z \vec{E}_{вн} & \text{в } D_R. \end{cases}$$

По аналогии с формулами (30) получим усредненные индукции:

$$\begin{aligned}\vec{D} &= \frac{1}{V_{R_\tau}} \int_{D_\tau} \vec{D}_{\text{сум}} dv = \frac{1}{V_{R_\tau}} \left[\varepsilon_M \int_{D_{R_\tau}} (\vec{E}_{\text{над}} + \vec{E}_{\text{опр}}) dv + \int_{D_R} (\varepsilon \vec{E}_{\text{вн}} + G \vec{H}_{\text{вн}}) dv \right], \\ \vec{B} &= \frac{1}{V_{R_\tau}} \int_{D_\tau} \vec{B}_{\text{сум}} dv = \frac{1}{V_{R_\tau}} \left[\mu_M \int_{D_{R_\tau}} (\vec{H}_{\text{над}} + \vec{H}_{\text{опр}}) dv + \int_{D_R} (\mu \vec{H}_{\text{вн}} + Z \vec{E}_{\text{вн}}) dv \right].\end{aligned}\quad (34)$$

Подставляя в правые части формул (34) интегралы (31), (32), приходим к следующему результату:

Теорема 4. Усредненные по области D_τ электрическая и магнитная индукции

$$\vec{D} = -D_- \vec{e}_+ - D_+ \vec{e}_- + D_z \vec{e}_z, \quad \vec{B} = -B_- \vec{e}_+ - B_+ \vec{e}_- + B_z \vec{e}_z$$

в матричном композите со сферическими биизотропными частицами, распределенными в проводящей матрице, определяются с помощью соотношений

$$\begin{aligned}D_+ &= -[\varepsilon_M (g_M b_{-11} + f_M y'_{-11}) - (\bar{F}_1^{(1)} x_{-11} + \bar{F}_2^{(1)} y_{-11})], \\ D_- &= -2[\varepsilon_M (g_M b_{11} + f_M y'_{11}) - (\bar{F}_1^{(1)} x_{11} + \bar{F}_2^{(1)} y_{11})], \\ D_z &= -2[\varepsilon_M (g_M b_{01} + f_M y'_{01}) - (\bar{F}_1^{(1)} x_{01} + \bar{F}_2^{(1)} y_{01})], \\ B_+ &= -[\mu_M h_M (g_M a_{-11} + f_M x'_{-11}) - (\bar{F}_1^{(2)} x_{-11} + \bar{F}_2^{(2)} y_{-11})], \\ B_- &= -2[\mu_M h_M (g_M a_{11} + f_M x'_{11}) - (\bar{F}_1^{(2)} x_{11} + \bar{F}_2^{(2)} y_{11})], \\ B_z &= -2[\mu_M h_M (g_M a_{01} + f_M x'_{01}) - (\bar{F}_1^{(2)} x_{01} + \bar{F}_2^{(2)} y_{01})]\end{aligned}$$

или в векторном виде

$$\begin{aligned}\vec{W}_+ &= -[\hat{H}_1 (g_M \vec{a}_{-1} + f_M \vec{X}'_{-1}) - \hat{F}_1 \vec{X}_{-1}], \\ \vec{W}_- &= -2[\hat{H}_1 (g_M \vec{a}_1 + f_M \vec{X}'_1) - \hat{F}_1 \vec{X}_1], \\ \vec{W}_z &= -2[\hat{H}_1 (g_M \vec{a}_0 + f_M \vec{X}'_0) - \hat{F}_1 \vec{X}_0],\end{aligned}\quad (35)$$

$$\text{где } \vec{W}_+ = \begin{pmatrix} D_+ \\ B_+ \end{pmatrix}, \quad \vec{W}_- = \begin{pmatrix} D_- \\ B_- \end{pmatrix}, \quad \vec{W}_z = \begin{pmatrix} D_z \\ B_z \end{pmatrix}, \quad \hat{H}_1 = \begin{pmatrix} 0 & \varepsilon_M \\ h_M \mu_M & 0 \end{pmatrix}, \quad \hat{F}_1 = \begin{pmatrix} \bar{F}_1^{(1)} & \bar{F}_2^{(1)} \\ \bar{F}_1^{(2)} & \bar{F}_2^{(2)} \end{pmatrix},$$

$$\bar{F}_1^{(1)} = \bar{F}_1(\varepsilon + p_1 G), \quad \bar{F}_2^{(1)} = \bar{F}_2(\varepsilon + p_2 G), \quad \bar{F}_1^{(2)} = \bar{F}_1(Z + p_1 \mu), \quad \bar{F}_2^{(2)} = \bar{F}_2(Z + p_2 \mu).$$

7. Алгоритм вычисления эффективных материальных параметров матричного композита

Построение алгоритма сводится к получению аналитических формул, выражающих усредненные индукции (34) через усредненные электрические и магнитные поля (30).

Приведем вспомогательные аналитические соотношения.

Лемма 3. Коэффициенты разложений в ряды полей l -го отражения от частицы D_R выражаются через коэффициенты $a_{mn}^{(0)}, b_{mn}^{(0)}$ первичного плоского поля (2) с помощью векторных соотношений

$$\vec{X}_m^{(l+1)} = (\tau \bar{f}_m)^l \hat{M} \hat{T}^l \vec{a}_m^{(0)}, \quad \vec{X}_m^{(l+1)'} = (\tau \bar{f}_m)^l \hat{T}^{l+1} \vec{a}_m^{(0)}, \quad \vec{a}_m^{(l)} = (\tau \bar{f}_m \hat{T})^l \vec{a}_m^{(0)}, \quad l = 0, 1, 2, \dots \quad (36)$$

Доказательство. Подставим третье выражение (25) во второе, тогда

$$\vec{X}_m^{(l+1)'} = \tau \bar{f}_M \hat{T} \vec{X}_m^{(l)'}. \quad (37)$$

Последовательно применяя соотношения (37) l раз, получим

$$\vec{X}_m^{(l+1)'} = \dots = (\tau \bar{f}_M \hat{T})^k \vec{X}_m^{(l-k+1)'} = \dots = (\tau \bar{f}_M \hat{T})^l \vec{X}_m^{(1)'}. \quad (38)$$

После подстановки (5) в (38) получим формулу

$$\vec{X}_m^{(l+1)'} = (\tau \bar{f}_M)^l \hat{T}^{l+1} \vec{a}_m^{(0)}. \quad (39)$$

Подставим (39) в третье равенство (25), тогда

$$\vec{a}_m^{(l)} = (\tau \bar{f}_M \hat{T})^l \vec{a}_m^{(0)}. \quad (40)$$

Аналогично после подстановки выражения (40) в первое равенство (25) придем к первой формуле (36). ■

Лемма 4. Коэффициенты разложений (27) при $n = 1, m = 0, \pm 1$, которые представляют собой результирующие поля в окрестности частицы D_R , равные сумме L отражений от частиц матричного композита, определяются векторными соотношениями

$$\vec{X}_m = \hat{M} \hat{T}^{(L)} \vec{a}_m^{(0)}, \quad \vec{X}_m' = \hat{T}^{(L)} \hat{T} \vec{a}_m^{(0)}, \quad \vec{a}_m = \hat{T}^{(L)} \vec{a}_m^{(0)}, \quad (41)$$

где
$$\hat{T}^{(L)} = \sum_{l=0}^{L-1} (\tau \bar{f}_M \hat{T})^l, \quad m = 0, \pm 1.$$

Для доказательства формул (41) достаточно (36) подставить в (29). ■

Лемма 5. Усредненные компоненты (33) электрической и магнитной напряженностей, усредненные компоненты (35) электрической и магнитной индукций с учетом L переотражений поля между частицами выражаются через коэффициенты разложений первичного поля (2) с помощью формул

$$\vec{V}_+ = -\hat{H}^{(L)} \vec{a}_{-1}^{(0)}, \quad \vec{V}_- = -2\hat{H}^{(L)} \vec{a}_1^{(0)}, \quad \vec{V}_z = -2\hat{H}^{(L)} \vec{a}_0^{(0)}; \quad (42)$$

$$\vec{W}_+ = -\hat{B}^{(L)} \vec{a}_{-1}^{(0)}, \quad \vec{W}_- = -2\hat{B}^{(L)} \vec{a}_1^{(0)}, \quad \vec{W}_z = -2\hat{B}^{(L)} \vec{a}_0^{(0)}, \quad (43)$$

где
$$\hat{H}^{(L)} = \hat{H} \hat{T}^{(L)} (g_M \hat{E} + f_M \hat{T}) - \hat{F} \hat{M} \hat{T}^{(L)},$$

$$\hat{B}^{(L)} = \hat{H}_1 \hat{T}^{(L)} (g_M \hat{E} + f_M \hat{T}) - \hat{F}_1 \hat{M} \hat{T}^{(L)}.$$

Для доказательства формул (42), (43) достаточно формулы (41) при $m = 0, \pm 1$ подставить в соотношения (33), (35). ■

Для упрощения математической модели неоднородного матричного композита, состоящего из частиц, заменим композит на однородную биизотропную среду, электромагнитное поле в которой подчиняется уравнениям

$$\text{rot } \vec{E} = i\omega \vec{D}, \quad \text{rot } \vec{H} = -i\omega \vec{D}, \quad (44)$$

где

$$\vec{B} = \mu_{\text{эф}} \vec{H} + Z_{\text{эф}} \vec{E}, \quad \vec{D} = \varepsilon_{\text{эф}} \vec{E} + G_{\text{эф}} \vec{H}. \quad (45)$$

Сформулируем теорему, определяющую усредненные материальные параметры $\varepsilon_{\text{эф}}, \mu_{\text{эф}}, G_{\text{эф}}, Z_{\text{эф}}$ матричного композита.

Теорема 5. Матричный композитный материал, состоящий из случайно распределенных биизотропных сферических частиц и заполненный матричным материалом в областях между частицами, является биизотропной средой с уравнениями (44). Эффективные параметры композита с учетом L взаимных переотражений поля между частицами определяются формулами

$$\varepsilon_{эф} = C_{11}, \quad G_{эф} = C_{12}, \quad \mu_{эф} = C_{22}, \quad Z_{эф} = C_{21}, \quad (46)$$

где

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \hat{C} = \hat{B}^{(L)}(\hat{H}^{(L)})^{-1}.$$

Доказательство. С помощью (42) исключим $\vec{a}_m^{(0)}$ из формул (43), получим

$$\vec{W}_+ = \hat{C}\vec{V}_+, \quad \vec{W}_- = \hat{C}\vec{V}_-, \quad \vec{W}_z = \hat{C}\vec{V}_z. \quad (47)$$

Запишем (47) в виде скалярных уравнений

$$D_m = C_{11}E_m + C_{12}H_m, \quad B_m = C_{21}E_m + C_{22}H_m, \quad m = z, +, -.$$

Для векторов индукций следует

$$\vec{D} = C_{11}\vec{E} + C_{12}\vec{H}, \quad \vec{B} = C_{21}\vec{E} + C_{22}\vec{H}. \quad (48)$$

Сравнивая индукции (48) с индукциями (45), получим требуемые формулы (46). ■

Заключение

В статье представлен алгоритм аналитического вычисления эффективных материальных параметров композита, состоящего из случайно распределенных в магнитоэлектрической проводящей матрице сферических биизотропных частиц с произвольными комплексными диэлектрической и магнитной проницаемостями и параметрами биизотропности. Алгоритм учитывает L переотражений между частицами внешнего электромагнитного поля, воздействующего на композит. При построении алгоритма используются точные решения краевых задач дифракции на сферических частицах, представленные в виде рядов по сферическим базисным полям. Разработан новый тип теорем сложения, связывающих базисные декартово-сферические электромагнитные поля в сдвинутых относительно друг друга сферических системах координат, которые упрощают построение алгоритма.

Работа выполнена по заданию ГПНИ «Информатика и космос».

Список литературы

1. 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.
2. Ерофеенко, В.Т. Аналитическое моделирование в электродинамике / В.Т. Ерофеенко, И.С. Козловская. – М.: КД Либроком, 2014. – 304 с.
3. Engheta, N. Antenna radiation in the presence of a chiral sphere / N. Engheta, M.W. Kowarz // J. Appl. Phys. – 1990. – Vol. 67(2). – P. 639–647.
4. 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.
5. 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.

6. Ерофеенко, В.Т. Электродинамическая модель расчета эффективных параметров композитов из сферических биизотропных частиц / В.Т. Ерофеенко // Информатика. – 2014. – № 1. – С. 45–58.
7. Ерофеенко, В.Т. Экранирование электромагнитных полей экранами из матричных композитов, содержащих биизотропные частицы / В.Т. Ерофеенко, В.Ф. Бондаренко // Информатика. – 2014. – № 3. – С. 28–43.
8. Исимару, А. Распространение и рассеяние волн в случайно неоднородных средах. Т. 2. Многократное рассеяние / А. Исимару. – М. : Мир, 1981. – 322 с.
9. Иванов, Е.А. Дифракция электромагнитных волн на двух телах / Е.А. Иванов. – Минск : Наука и техника, 1968. – 584 с.
10. Ерофеенко, В.Т. Теоремы сложения / В.Т. Ерофеенко. – Минск : Наука и техника, 1989. – 256 с.

Поступила 30.09.2015

*Учреждение БГУ «НИИ прикладных проблем математики и информатики»,
Минск, пр. Независимости, 4
e-mail: bsu_erofeenko@tut.by*

V.T. Erofeenko

MODEL FOR CALCULATING EFFECTIVE PARAMETERS OF MATRIX COMPOSITES FROM BI-ISOTROPIC PARTICLES WITH REGARD MULTIPLE REFLECTIONS OF ELECTROMAGNETIC FIELD

A method for calculation of the effective material parameters of matrix composites, consisting of magnetodielectric conducting matrix with a set of bi-isotropic spherical fractions, is developed. Under the calculations of parameters, taking into account a multiple scattering of the field between particles, a new type of addition theorems, connecting basic spherical electromagnetic fields relating to different particles, is used. The developed method allows to calculate the effective parameters for the composites with sufficiently dense ensemble of the fractions.

УДК 519.116+519.688

А.С. Врублевский¹, В.А. Шлык²**ВЫЧИСЛЕНИЕ ВЕРШИН ПОЛИТОПОВ РАЗБИЕНИЙ ЧИСЕЛ**

*Описывается метод генерирования вершин политопов разбиений чисел, с помощью которого авторами были вычислены все вершины и опорные вершины политопов разбиений всех $n \leq 105$ и все рюкзачные разбиения $n \leq 165$. Метод не требует построения всех разбиений n . Вершины определяются с помощью достаточных и необходимых условий, в трудных случаях применяется известная программа *Polymake*. Подробно излагаются алгоритм проверки критерия, характеризующего разбиения, являющиеся выпуклыми комбинациями двух других; методика применения двух комбинаторных операций, преобразующих известные вершины в новые вершины, и способ применения программы *Polymake* для распознавания небольшого (для малых n) числа разбиений, являющихся выпуклыми комбинациями трех и более разбиений. Представляются результаты вычислений и формулируются новые проблемы, к которым приводят полученные данные о числах вершин и опорных вершин политопов разбиений чисел.*

Введение

Разбиением натурального числа n [1] называется конечная невозрастающая последовательность натуральных чисел $\rho = (\rho_1, \rho_2, \dots, \rho_r)$, для которой $\sum_{j=1}^r \rho_j = n$. Неформально разбиением положительного целого числа n часто называют его представление в виде суммы положительных целых чисел

$$n = n_1 + n_2 + \dots + n_k,$$

где $n_1, n_2, \dots, n_k \in \mathbb{Z}$, $n_1, n_2, \dots, n_k > 0$, и порядок слагаемых не учитывается. Слагаемые n_1, n_2, \dots, n_k называют частями разбиения.

Основы теории разбиений чисел заложил Л. Эйлер [2]. Существенный вклад в ее развитие внесли многие крупнейшие математики [1, 3]. Разбиения чисел играют важную роль в разных областях математики, в статистической механике и современной теоретической физике. Вторым автором предложен полиэдральный подход в теории разбиений чисел [4, 5]. При его применении каждое разбиение n отождествляется с неотрицательной целочисленной точкой $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, являющейся решением уравнения

$$x_1 + 2x_2 + \dots + nx_n = n.$$

Каждая компонента x_i точки x равна числу вхождений части i в разбиение. Так, разбиение $8 = 1 + 1 + 2 + 4$ отождествляется с точкой $(2, 1, 0, 1, 0^4)$, где 0^4 обозначает последовательность из четырех нулей. То, что x является разбиением n , обозначим $x \vdash n$, через $S(x) = \{i \in \{1, \dots, n\} \mid x_i > 0\}$ обозначим множество различных частей разбиения $x \vdash n$, через \mathbb{Z}_+ – множество неотрицательных целых чисел, а через $[n_1, n_2]$ – отрезок натуральных чисел $\{n_1, n_1 + 1, \dots, n_2\}$.

Политоп $P_n \subset \mathbb{R}^n$ разбиений числа n определяется как выпуклая оболочка множества

$$T_n = \{x \in \mathbb{Z}_+^n \mid x_1 + 2x_2 + \dots + nx_n = n\},$$

состоящего из всех точек x , соответствующих разбиениям n [4, 5]:

$$P_n = \text{conv}\{x \in \mathbb{R}_+^n \mid x \vdash n\}.$$

Точка x произвольного полиэдра P является его вершиной тогда и только тогда, когда ее нельзя представить в виде выпуклой комбинации $x = \sum_{j=1}^k \lambda_j y^j$, $\sum_{j=1}^k \lambda_j = 1$, $\lambda_j > 0$, некоторых других точек $y^j \in P$, $1 \leq j \leq k$ [6]. Множество вершин политопа P_n будем обозначать $\text{vert} P_n$.

На рис. 1 представлены политопы P_1, P_2, P_3 и проекция политопа P_4 , лежащего в пространстве \mathbb{R}^4 , на пространство первых трех координат \mathbb{R}^3 . Вершина $(0, 0, 0, 1)$ политопа P_4 , соответствующая разбиению $4 = 4$, не изображена. Из рисунка видно, что при $n < 4$ каждое разбиение n является вершиной P_n , однако при $n = 4$ разбиение $(2, 1, 0, 0) \vdash 4$ вершиной P_4 не является.

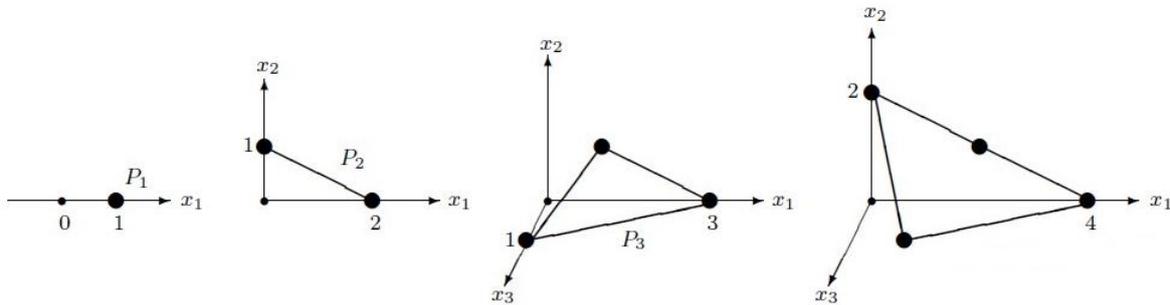


Рис. 1. Политопы P_1, P_2, P_3 и проекция политопа P_4 на пространство \mathbb{R}^3

В работе [5] показано, что T_n есть множество всех целочисленных точек политопа P_n и что аффинная размерность P_n равна $n - 1$.

Вершины P_n порождают множество всех разбиений T_n , поскольку любое разбиение представляется в виде выпуклой комбинации конечного набора вершин. Другими словами, вершины P_n составляют базис множества T_n относительно операции взятия выпуклой комбинации. Поэтому применение полиэдрального подхода позволяет избежать построения всех разбиений n , ограничившись вершинами P_n . В [7] показано, что вершинный базис множества разбиений может быть уменьшен: нет необходимости знать все вершины P_n , поскольку многие из них можно построить из подмножества вершин, названных опорными, при помощи двух комбинаторных операций (см. разд. 2, условие ДЗ).

В теории разбиений число разбиений числа n принято обозначать $p(n)$. Обозначим через $v(n)$ число вершин политопа P_n , а через $s(n)$ – число его опорных вершин.

Вычисление значений функции $p(n)$ является предметом постоянного интереса с момента зарождения теории разбиений [1]. Точная формула для $p(n)$ до сих пор неизвестна, однако работы Харди, Рамануджана и Радемахера привели к представлению $p(n)$ в виде суммы быстро сходящегося ряда, причем точное значение $p(n)$ можно получить, взяв малое число членов ряда и округлив сумму до ближайшего целого. Макмагон вычислил значения $p(n)$ для первых двухсот натуральных чисел, используя рекуррентное соотношение, вытекающее из пентагональной теоремы Эйлера и выражающее число разбиений n через числа разбиений меньших натуральных чисел. В частности, он получил, что $p(200) = 3972999029388$. В [1, 8] приведены таблицы значений $p(n)$ и чисел разбиений из некоторых классов, соответствующих наиболее важным дополнительным ограничениям на части разбиений, для $n \leq 100$. В печатном

виде наиболее полные таблицы значений $p(n)$ для $n \leq 400$ представлены в [9]. В онлайн-энциклопедии Слоана «Целочисленные последовательности» можно найти последовательность значений $p(n)$ для $n \leq 10\,000$ [10]. Последнее число $p(10\,000)$ содержит 107 знаков:

$$p(10\,000) = 36167251325636293988820471890953695495016030339315650 \\ 422081868605887952568754066420592310556052906916435144.$$

Часто требуется знать не только количество разбиений, но и уметь перечислять их в явном виде. Необходимость иметь полные списки разбиений, перестановок, разбиений множеств, деревьев и других основных комбинаторных структур возникает, например, при тестировании программ и алгоритмов, анализе их вычислительной сложности, а также при проверке всех возможных решений задач. Знание полных списков разбиений важно и для теоретических исследований. Они помогают обнаружить новые свойства, высказать и проверить возникающие предположения или построить опровергающие контрпримеры. Так, анализ списков разбиений показал, что частота появления хотя бы одной двойки в качестве части разбиения n , т. е. $x_2 \geq 1$, возрастает с ростом n . В разбиениях $n = 30$ двойка встречается в 66 % разбиений, а для $n = 90$ – в 78 % разбиений. Доказана сходимость $\frac{p(n-2)}{p(n)} \rightarrow 1$ при возрастании n [11].

Основные алгоритмы генерирования разбиений чисел, в том числе с ограничениями, которые чаще всего накладывают на величину и (или) число частей, можно найти в [1, 12–15], см. также [10]. Обзор методов генерирования разбиений представлен Д. Кнутом в материалах [16] для четвертого тома его труда «Искусство программирования». Разработка новых алгоритмов генерирования разбиений продолжается и сейчас [17–19].

В настоящей статье описывается разработанный авторами метод вычисления вершин и опорных вершин политопа P_n . С его помощью построены все вершины и опорные вершины политопа P_n для всех $n \leq 100$. Вычисленные последовательности чисел $v(n)$ и $s(n)$ включены в энциклопедию «Целочисленные последовательности» [20, 21]. Попутно была продолжена до $n = 165$ вычисленная Эренборгом для $n \leq 50$ последовательность чисел рюкзачных разбиений [22].

В разд. 1 статьи изложен алгоритм построения списка $\text{PartList}(n)$ разбиений n , содержащего все вершины политопа P_n и некоторые подозрительные разбиения, о которых в данный момент неизвестно, являются ли они вершинами. Разработанный метод решения проблемы распознавания вершин среди подозрительных разбиений и отсеивания избыточных разбиений, а также основные моменты его программной реализации описаны в разд. 2. Здесь приведены достаточные и (не одновременно) необходимые условия для вершин и охарактеризованы их практическая эффективность и трудоемкость, что обосновывает выбранную стратегию проверки указанных условий. Приведено подробное описание алгоритма проверки наиболее общего и одновременно наиболее трудоемкого условия, характеризующего разбиения, являющиеся выпуклыми комбинациями двух других. Изложена методика применения μ -операций, с помощью которых из известных вершин политопа можно строить новые вершины, а также применения программы Polymake , к которой авторы прибегают для распознавания небольшого (для сравнительно малых n) числа оставшихся разбиений, которые не являются вершинами потому, что они представляют собой выпуклые комбинации более чем двух разбиений. В разд. 3, 4 и 5 изложены соответственно методы построения списков опорных вершин политопа P_n , рюкзачных разбиений и некоторые технические особенности организации вычислений. В разд. 6 изложены результаты проведенных вычислений и сформулированы новые проблемы о структуре множеств разбиений, к которым приводят полученные данные о поведении функций чисел разбиений, вершин и опорных вершин политопа разбиений чисел.

1. Генерирование списка разбиений

Генерирование списка $\text{PartList}(n)$ разбиений каждого числа n производится последовательно по мере роста n , начиная с $n = 1, 2, 3$. При $n = 1$ в список безусловно включается единственное разбиение $x = (1)$. При обращении к очередному политопу P_n используются уже построенные списки $\text{VertList}(n-i)$ вершин политопов P_{n-i} , $1 \leq i \leq \lfloor n/2 \rfloor$, а последним безусловно включается разбиение $x = (0^{n-1}, 1)$. Сгенерированный список содержит все вершины P_n , но в нем присутствуют также некоторые избыточные разбиения, не являющиеся вершинами. Разбиения, о которых в текущий момент неизвестно, вершины они или нет, назовем *подозрительными*.

В списках $\text{PartList}(n)$ разбиения хранятся в лексикографическом порядке. Например, четыре разбиения $n = 4$ хранятся в $\text{PartList}(4)$ в порядке $x^1 = (4, 0, 0, 0)$, $x^2 = (2, 1, 0, 0)$, $x^3 = (1, 0, 1, 0)$, $x^4 = (0, 2, 0, 0)$, где разбиение x^2 избыточное, так как $x^2 = \frac{1}{2}(x^1 + x^4)$. Такой порядок облегчает выполнение используемых далее операций выбора разбиений из $\text{PartList}(n)$ и построение в дальнейшем списка $\text{SuppList}(n)$ опорных вершин P_n .

Первоначальное построение списка $\text{PartList}(n)$ основано на рекуррентном соотношении

$$\text{vert } P_n \subseteq \left(\biguplus_{i=1}^{\lfloor n/2 \rfloor} \Phi_i(\text{vert } P_{n-i}[\geq i]) \right) \cup (0^{n-1}, 1), \quad (1)$$

где \biguplus обозначает объединение непересекающихся множеств; $P_m[\geq i]$ – политоп тех разбиений m , все части которых не меньше i ,

$$P_m[\geq i] = \text{conv}\{x \vdash m \mid x_j = 0, j < i\},$$

и Φ_i , $1 \leq i \leq \lfloor n/2 \rfloor$, – оператор

$$\Phi_i : \mathbb{R}^{n-i} \rightarrow \mathbb{R}^n : \Phi_i(x_1, x_2, \dots, x_{n-i}) = (x_1, x_2, \dots, x_{i-1}, x_i + 1, x_{i+1}, \dots, x_{n-i}, 0^i).$$

Каждый оператор Φ_i есть композиция сдвига на i вдоль оси x_i и вложения пространства \mathbb{R}^{n-i} в \mathbb{R}^n , при котором к каждой точке \mathbb{R}^{n-i} добавляются i нулевых координат $x_{n-i+1} = x_{n-i} = \dots = x_n = 0$. Понятно, что если $y \vdash n-i$, то $x = \Phi_i(y) \vdash n$. Наоборот, если $x \vdash n$, $x \neq (0^{n-1}, 1)$ и $x_i > 0$ для некоторого $1 \leq i \leq \lfloor n/2 \rfloor$, то $\Phi_i^{-1}(x)$ есть разбиение числа $n-i$. В работе [5] доказано, что если вершина $x \in \text{vert } P_n$ содержит часть $i < n$, то $\Phi_i^{-1}(x) \in \text{vert } P_{n-i}$. Поэтому, последовательно включая в $\text{PartList}(n)$ вершины из списков $\text{VertList}(n-i)$, $i = 1, 2, \dots, \lfloor n/2 \rfloor$, состоящие из частей, не меньших i (см. (1)), получим все вершины P_n в указанном выше порядке. Построенный таким образом список $\text{PartList}(n)$ может содержать некоторые избыточные разбиения, которые вершинами P_n не являются. Для построения списка вершин $\text{VertList}(n)$ их необходимо обнаружить и исключить. Применяемый метод позволяет избежать перебора и проверки всех разбиений n .

2. Распознавание вершин и избыточных разбиений

Отсеивание из $\text{PartList}(n)$ избыточных разбиений и построение в результате списка вершин $\text{VertList}(n)$ политопы P_n составляют основную проблему вычислений. Для распознавания

вершин P_n и избыточных разбиений в текущем списке $\text{PartList}(n)$ применяются доказанные в [5, 23–25] достаточные и необходимые условия для вершин. В сложных случаях программа обращается к созданной в Германии универсальной программе Polymake [26].

Распознавание вершин и избыточных разбиений выполняется в три этапа:

1. Распознавание некоторых вершин и разбиений, являющихся выпуклыми комбинациями двух других.

2. Применение μ -операций (условие ДЗ, см. далее в описании этапа 1).

3. Применение программы Polymake .

Проверка условия ДЗ вынесена в отдельный этап, так как для ее выполнения нужен еще один проход по всем разбиениям из списка, полученного на первом этапе. Выделив из списка вершины и подозрительные разбиения, его можно существенно сократить и повысить скорость дисковых операций.

Этап 1. Распознавание некоторых вершин и разбиений, являющихся выпуклыми комбинациями двух других.

Имеются три достаточных условия для того, чтобы разбиение было вершиной P_n [7, 24, 25, 27]:

Д1. Пусть для множества $\{i_1, i_2, \dots, i_k\} \subset [1, n]$ уравнение $i_1 x_1 + i_2 x_2 + \dots + i_k x_k = n$ имеет точно одно или два решения в целых положительных числах. Тогда для каждого решения a_1, a_2, \dots, a_k точка $x \in \mathbb{R}^n$ с ненулевыми компонентами $x_i = a_i$ для $i \in \{i_1, i_2, \dots, i_k\}$ является вершиной P_n .

Д2. Пусть $\{i_1 < i_2 < \dots < i_k\}$ – множество различных частей разбиения $x \vdash n$. Если $x_{i_k} = \lfloor n_k / i_k \rfloor$ для $n_k = n$; $x_{i_{k-1}} = \lfloor n_{k-1} / i_{k-1} \rfloor$ для $n_{k-1} = n_k - x_{i_k} i_k$; \dots ; $x_{i_1} = x_{i_1} = \lfloor n_1 / i_1 \rfloor = n_1$ для $n_1 = n_2 - x_{i_2} i_2$ и $x_i = 0$ для $i \neq i_1, i_2, \dots, i_k$, то $x = (x_1, x_2, \dots, x_n)$ – вершина P_n .

Д3. Разбиения, полученные из вершин политопа P_n с помощью определенных ниже μ -операций слияния частей, являются вершинами P_n .

Операция 1. Пусть $x \vdash n$ и пусть $u, v \in S(x)$, $u \neq v$, – две различные части x . Для определенности можно считать, что $x_u \leq x_v$. Построить точку $y = \mu_{u,v}(x) \in \mathbb{Z}_+^n$ с компонентами $y_u = 0$, $y_v = x_v - x_u$, $y_{u+v} = x_{u+v} + x_u$, $y_j = x_j$ для $j \in [1, n]$, $j \neq u, v, u+v$.

Операция 2. Пусть x – разбиение числа n , в которое некоторая часть $u \in S(x)$ входит более одного раза, т. е. $x_u > 1$. Построить точку $y = \mu_u(x) \in \mathbb{Z}_+^n$ с компонентами $y_u = 0$, $y_{x_u u} = x_{x_u u} + 1$, $y_j = x_j$ для $j \in [1, n]$, $j \neq u, x_u u$.

Для определения в $\text{PartList}(n)$ избыточных разбиений имеются следующие необходимые условия [5, 23–25, 27] принадлежности x множеству $\text{vert} P_n$:

Н1. Ни одно целое $k < n$ вида $k = \sum_{i \in S(x)} q_i i$, $q_i \in \mathbb{Z}_+$, $q_i \leq x_i$, кроме тривиального случая $k = 1 \cdot i$, не является частью x .

Н2. $i x_i < k$ для всех i и k , таких, что $1 \leq i < k \leq n$, i делит k и $x_k > 0$.

Н3. $i x_i < m - k$ для всех троек индексов i, k, m , таких, что $k < m$, i делит $m - k$ и $x_k, x_m > 0$.

Н4. $\prod_{i \in S(x)} (x_i + 1) \leq n + 1$.

Н5. Число различных частей в x не превосходит $\lfloor \log(n+1) \rfloor$, т. е. $|S(x)| \leq \lfloor \log(n+1) \rfloor$.

Н6. Если $x \neq (n, 0^{n-1})$ и $n > 1$, то x содержит не более $\lceil n/2 \rceil$ частей, т. е. $\sum_{i=1}^n x_i \leq \lceil n/2 \rceil$.

Н7. Любые два целочисленных набора $u = \langle u_j; j \in S_1 \rangle$, $v = \langle v_k; k \in S_2 \rangle$, $0 < u_j \leq x_j$, $0 < v_k \leq x_k$, соответствующих подмножествам $S_1, S_2 \subset S(x)$, $S_1 \cap S_2 = \emptyset$, удовлетворяют соотношению

$$\sum_{j \in S_1} u_j j \neq \sum_{k \in S_2} v_k k.$$

Как достаточные, так и необходимые условия неравноценны по своей эффективности. Достаточные условия определяют различное число вершин, а одни необходимые условия отсеивают больше подозрительных разбиений, чем другие. Поэтому общее время, затрачиваемое на их проверку, существенно зависит от порядка проверки условий. Экспериментально подобран такой порядок, при котором первыми проверяются либо самые быстрые, либо самые эффективные условия.

От использования трудоемкого условия Д1 в данной версии программы авторы отказались, хотя при проверке разбиений «вручную» в несколько измененных, но трудноформализуемых вариантах оно часто оказывается полезным. Проверка условия Д2 проста и, как показали вычисления, это условие определяет большую часть вершин, содержащихся среди подозрительных разбиений. Условия Н1–Н4 позволяют быстро отсеивать многие разбиения, не являющиеся вершинами политопов P_n . Например, для $x = (1, 0, 2, 0, 3, 0^{17}) \vdash 22$ имеем $2 \cdot 3 \cdot 4 = 24 > 23$, откуда ввиду Н4 заключаем, что $x \notin \text{vert } P_{22}$. Наиболее результативными оказались необходимые условия Н2 и Н3, причем применение Н2 даже в примитивном варианте

$$x \in \text{vert } P_n, i \in S(x), q \leq x_i \Rightarrow qx_i \notin S(x)$$

отсеивает массу подозрительных разбиений. Условия Н4 и Н6, напротив, оказались малопродуктивными.

Условие Н7 выражает доказанный в [23] критерий того, что разбиение является выпуклой комбинацией двух других разбиений того же числа. Условия Н1–Н5 представляют собой его частные случаи. Для ускорения вычислений их было решено проверять первыми, а намного более трудоемкое условие Н7 – последним.

Для малых n сложная задача распознавания разбиений, оставшихся подозрительными после проверки условий Д2 и Н1–Н5, решается достаточно быстро, но уже при $n > 50$ она требует значительных временных затрат. Тем не менее проверять условие Н7 все еще выгоднее, чем обращаться к программе Polymake.

Распознанные вершины помечаются в списке PartList(n) специальным ключом. Отсеиваемые подозрительные разбиения также не удаляются всякий раз из списка, поскольку удаление записи связано с обращением к внешней памяти и не может быть выполнено быстро. Как и найденные вершины, эти разбиения помечаются специальным признаком, который исключает их из дальнейшей обработки и служит для сбора статистической информации в процессе последующего анализа.

Обработка каждого из включаемых в список PartList(n) разбиений заканчивается проверкой условия Н7. Алгоритм его проверки описан ниже. После завершения первого этапа в списке PartList(n) найдены и помечены большая часть вершин и все разбиения, являющиеся выпуклыми комбинациями двух других.

Алгоритм проверки условия Н7.

Вход: подозрительное разбиение $x \vdash n$ из PartList(n).

Выход: один из двух ответов: «разбиение x не является вершиной P_n » или «разбиение x не является выпуклой комбинацией никаких двух разбиений n ».

Для каждой из различных частей разбиения x вводим битовую маску, использование которой облегчит и ускорит обработку x . Например, для разбиения $x = (0, 1, 2, 3, 0^{16}) \vdash 20$, состоящего из частей 2, 3, 3, 4, 4, 4, используем маски: 001 для части 2, 010 для части 3 и 100 для части 4.

С помощью итерационного процесса строим все возможные комбинации частей разбиения x , соблюдая следующие ограничения. В каждой комбинации любая часть i из x должна повторяться не более x_i раз и комбинации не должны содержать одновременно все различные части x . В комбинациях части упорядочены по возрастанию. Сами комбинации строим по группам: из одной, двух, трех и т. д. частей. Внутри групп комбинации упорядочены лексикографически по возрастанию.

Для каждой комбинации вычисляем:

- сумму, равную сумме всех входящих в нее частей, учитывая повторения;
- битовую маску, являющуюся объединением масок всех входящих в нее различных частей. Ее легко вычислить, применив к маскам частей операцию арифметического OR. Так, маской комбинации $\langle 3, 3, 4 \rangle$ является $(010)OR(100) = (110)$, а ее сумма равна $3 + 3 + 4 = 10$.

Для хранения данных используем четыре массива. В массиве `Masks` хранятся маски всех различных частей x . В массиве `Mas1` хранятся комбинации, состоящие из $k - 1$ частей разбиения x (k – номер итерации), а в массиве `Mas2` – комбинации, состоящие из k частей. Части могут повторяться, k возрастает от 1. В массиве `Sums` хранятся пары (сумма, маска), рассчитанные для построенных комбинаций. `Sums` представляет собой массив, который на языке C++ называют множественным ассоциативным контейнером (`std::multimap Sums`). «Множественный» означает, что одному значению ключа может соответствовать несколько значений данных. В рассматриваемом случае ключом служит сумма комбинаций, а маски комбинаций выступают в роли данных. Одному значению ключа может соответствовать несколько масок, так как одну и ту же сумму можно получить из различных наборов слагаемых. Интернет-ресурсы (cplusplus.com, cppreference.com) указывают, что в сортированных ассоциативных контейнерах `multimap` операции поиска, вставки и удаления элементов имеют логарифмическую сложность.

Сам алгоритм состоит из следующих шагов:

Шаг 1. Инициализация: число частей в комбинациях $k = 1$. В `Mas1` заносим комбинации, состоящие из одной части разбиения x , т. е. каждую из различных частей разбиения x по одному разу включаем в `Mas1`. Сумму каждой комбинации (на этапе инициализации сумма равна самой части) и ее битовую маску заносим в массив `Sums`.

Шаг 2. Итерация $k + 1$: переход от построенных на k -й итерации комбинаций, состоящих из k частей, к комбинациям из $k + 1$ частей.

К каждой комбинации из массива `Mas1` присоединяем по очереди различные не вошедшие в нее части разбиения x . Добавление начинаем с наибольшей из входящих в комбинацию части, так как комбинации с меньшими частями к этому моменту уже построены. Следим за тем, чтобы в новую комбинацию не вошли все различные части x . Новые комбинации из $k + 1$ частей добавляем в массив `Mas2`.

После добавления в `Mas2` очередной комбинации c вычисляем ее сумму и маску и выполняем следующую проверку. Если в `Sums` имеется одна или несколько пар (сумма, маска), для которых сумма равна сумме комбинаций c , а арифметическое AND маски из пары с маской комбинации c равно 0 (это означает, что наборы частей в комбинации c и комбинации, соответствующей рассматриваемой паре из `Sums`, не пересекаются), то можно сделать следующий вывод: разбиение x не является вершиной P_n . *Конец алгоритма.* В противном случае добавляем сумму и битовую маску комбинации c в `Sums` и переходим к построению следующей комбинации.

Когда все комбинации из массива `Mas1` обработаны и исходя из них в `Mas2` построены новые комбинации из $k + 1$ частей, очищаем `Mas1` и перемещаем в него все комбинации из

Mas2. Массив Mas2 пуст и готов к занесению в него на следующей итерации комбинаций из $k + 2$ частей. Если массив Mas1 не пуст, то переходим к итерации $k + 2$, в противном случае (когда добавление новой части к каждой комбинации приводит к использованию всех различных частей x) можно сделать следующий вывод: разбиение x не является выпуклой комбинацией никаких двух разбиений n . *Конец алгоритма.*

Проиллюстрируем работу алгоритма, выполнив проверку одной из комбинаций частей разбиения $x = (2, 5, 3, 0, 1, 0^{21}) \vdash 26$. Пусть $k = 5$ и на данной итерации построена комбинация из пяти частей $\langle 2, 2, 2, 2, 2 \rangle$. Ее сумма равна $2 + 2 + 2 + 2 + 2 = 10$, а маска – 0010. Маска данной комбинации четырехзначная, поскольку в разбиении x только четыре различные части и частям 1, 2, 3, 5 соответствуют маски 0001, 0010, 0100, 1000 соответственно. К моменту построения комбинации x в Sums содержатся пять пар (сумма, маска), соответствующих пяти комбинациям с суммой 10: пара (10, 1110) для комбинации $\langle 2, 3, 5 \rangle$; (10, 1101) для $\langle 1, 1, 3, 5 \rangle$; (10, 1011) для $\langle 1, 2, 2, 5 \rangle$; (10, 0110) для $\langle 2, 2, 3, 3 \rangle$ и (10, 0111) для $\langle 1, 2, 2, 2, 3 \rangle$.

Вычисляя арифметическое AND маски 0010 с маской из первой пары (10, 1110), получаем $0010 \text{ AND } 1110 = 0010$, что не равно 0, следовательно, переходим к обработке следующей пары. Арифметическое AND с ее маской дает $0010 \text{ AND } 1101 = 0000$. Таким образом, соответствующие этим парам комбинации $\langle 2, 2, 2, 2, 2 \rangle$ и $\langle 1, 1, 3, 5 \rangle$ имеют одинаковые суммы, но части данных комбинаций различны и, значит, разбиение $x = (2, 5, 3, 0, 1, 0^{21})$ не является вершиной P_{26} . *Конец алгоритма.*

Этап 2. Применение μ -операций.

На втором этапе с помощью условия ДЗ, использующего μ -операции, удастся обнаружить некоторые из тех подозрительных разбиений из списка PartList(n), которые являются вершинами P_n . Применение μ -операций к разбиению реализовано с помощью специальных функций.

Вершины и подозрительные разбиения извлекаются из исходного файла в отдельный файл. Для него запускается функция, реализующая μ -операции из условия ДЗ. В результате некоторые подозрительные разбиения получают статус «вершина».

Функция, реализующая μ -операции, строит в памяти список вершин, полученных из текущего разбиения-вершины, добавляя их туда в лексикографическом порядке через вспомогательную функцию-сортировщик. Этот список передается другой вспомогательной функции, которая проходит по файлу, находя разбиения из построенного списка и меняя их статус, если они были подозрительными. Затем основная функция переходит к следующей вершине.

Этап 3. Применение программы Polymake.

К началу третьего этапа в PartList(n) найдены и помечены бóльшая часть вершин и все разбиения, являющиеся выпуклыми комбинациями двух других. Однако среди подозрительных разбиений могут присутствовать такие «трудные» разбиения, которые требуют для своего выпуклого представления трех или более других разбиений. Такие разбиения появляются при $n = 15, 21, 24, 27$ и присутствуют далее почти для всех n . Для $n = 15$ существует всего одно «трудное» разбиение $15 = 2 \cdot 3 + 4 + 5$ с частями 3, 4, 5. Это разбиение – не вершина P_{15} , поскольку является выпуклой комбинацией трех разбиений:

$$(0, 0, 2, 1, 1, 0^{10}) = \frac{1}{3}(0, 0, 0, 0, 3, 0^{10}) + \frac{1}{3}(0, 0, 1, 3, 0, 0^{10}) + \frac{1}{3}(0, 0, 5, 0, 0, 0^{10}).$$

Число $n = 21$ обладает уже тремя разбиениями данного типа: $(1, 0, 0, 1, 0, 0, 1, 0, 1, 0^{12})$, $(0, 0, 1, 0, 1, 1, 1, 0^{14})$ и $(0, 0, 3, 0, 1, 0, 1, 0^{14})$. С ростом n такие «трудные» разбиения перестают быть исключениями. Для чисел вида $n = 21 + 6k$ и $n = 21 + 3k$ существуют серии разбиений, являющихся выпуклыми комбинациями трех разбиений n [27].

Для распознавания «трудных» разбиений применяем программу Polymake. Polymake не позволяет определить минимальное число разбиений, необходимых для выпуклого представле-

ния. Однако анализ результатов позволил найти разбиения чисел $n = 43$ и $n = 44$, которые требуют четырех разбиений n для своего выпуклого представления.

Задача «Является ли заданная точка политопа $P \subset \mathbb{R}^n$ выпуклой комбинацией k точек из P , но не является выпуклой комбинацией m , $m < k$, точек, где $k > 2$ – фиксированное целое?» ранее не рассматривалась. Характеризация разбиений, представимых через два разбиения, – пока единственный результат в данном направлении. Поэтому для решения этой задачи используется программа Polymake.

Polymake – мощная программа, которая «умеет» вычислять разнообразные характеристики многогранников. Однако ограничиться ее применением для вычисления вершин P_n невозможно по нескольким причинам. Во-первых, для того чтобы с помощью Polymake строить вершины некоторого политопа, его необходимо задать одним из двух способов: указав явно или все фасеты политопа, или набор точек (можно с избытком), выпуклой оболочкой которых он является. Для политопа P_n явно известны только тривиальные фасеты, а нетривиальные охарактеризованы как вершины другого политопа и явно неизвестны [5]. Поэтому нельзя задать политоп P_n первым способом. Второй способ в рассматриваемом случае также непригоден. Число разбиений n имеет порядок

$$p(n) \sim \frac{1}{4n\sqrt{3}} e^{\pi\sqrt{\frac{2}{3}}\sqrt{n}} \quad (2)$$

и быстро растет [28]. Поэтому перечисление всех разбиений вскоре вынудило бы нас прекратить вычисления, в то время как целью работы является получение информации о вершинах политопа P_n для как можно больших значений n . Более того, желательно максимально сократить даже построенный список разбиений $\text{PartList}(n)$, выпуклой оболочкой которых является P_n . Кроме того, в силу своей универсальности программа Polymake выполняется довольно медленно. Она хороша для разовых вычислений, но малоприспособна для обработки структур, характеризующих серию многогранников (или политопов), особенно если объем их входных данных быстро возрастает. При вычислении вершин полиэдров Polymake многократно использует алгоритмы линейного программирования, а нам необходимо распознать вершины наиболее эффективным методом, используя их комбинаторные свойства.

Отсеивание с помощью Polymake разбиений, требующих $k > 2$ разбиений, удалось ускорить, воспользовавшись связью между вершинами P_n и вершинами политопов ограниченных разбиений $P_n(M)$, таких, что все их части принадлежат заданному $M \subset \{1, 2, \dots, n\}$, т. е. $P_n = \text{conv}\{x \in \mathbb{R}_+^n \mid x \vdash n, S(x) \subseteq M\}$.

Эту связь выражает следующее

Утверждение [5]. *Политоп $P_n(M)$ является гранью политопа P_n и совпадает с пересечением $P_n \cap E(M)$, где $E(M) = \{x \in \mathbb{R}^n : x_i = 0, i \notin M\}$.*

Следовательно, для того чтобы определить, является ли $x \vdash n$ вершиной P_n , необходимо и достаточно проверить, представляется ли x в виде выпуклой комбинации разбиений $y \vdash n$, $y \in E(M)$. Этот прием позволил сократить время расчета на порядок. Для каждого из оставшихся подозрительных разбиений x выгоднее потратить время на предобработку, выбрав и подав на вход программе Polymake только те разбиения из $\text{PartList}(n)$, которые принадлежат $E(M)$, где $M = S(x)$, чем подавать ей весь список. При этом данное подмножество разбиений, кроме проверяемого, может содержать и другие подозрительные разбиения, статус которых – вершины они или нет – будет определен одновременно со статусом проверяемого разбиения. Для n порядка 100 количество одновременно обрабатываемых таким образом подозрительных разбиений порой превышает 10.

Для реализации описанного приема написаны четыре функции на C++, которые подключаются к программе Polymake и вызываются из встроенного в нее интерпретатора языка Perl. Функции разработаны на языке C++, так как Perl не позволяет считывать байты из файла, как из бинарного потока. При этом упрощается и портирование (Polymake создана на платформе Linux) других, уже написанных на C++ функций, необходимость в которых может возникнуть при использовании вычислений в Polymake. Планируется, например, проверить эффективность применения μ -операций к очередным найденным Polymake вершинам.

Первая функция извлекает из исходного файла в отдельный временный файл очередное подозрительное разбиение и все разбиения, которые принадлежат $E(M)$. В имени временного файла отражен номер позиции этого подозрительного разбиения в исходном файле. Вторая функция из полученного временного файла считывает разбиения в объект Polytope программы Polymake, после чего вызывается встроенная функция этого объекта, определяющая вершины соответствующего политопа. Третья функция обновляет статусы разбиений во временном файле в соответствии с полученным Polymake набором вершин. Четвертая функция объединяет информацию из временного файла с исходным общим файлом.

Обработка подозрительных разбиений в исходном файле продолжается далее до тех пор, пока они не будут исчерпаны. На этом построение списка вершин политопа P_n завершается.

3. Построение списка опорных вершин

Опорные вершины политопа P_n – это такие вершины, которые нельзя получить из других вершин с помощью введенных в [7] и определенных в условии ДЗ μ -операций слияния частей. Их существование следует из того, что применение любой из μ -операций к вершине x приводит к вершине y с меньшим числом частей: $\sum_{i=1}^n y_i < \sum_{i=1}^n x_i$.

Все вершины P_n можно построить из опорных вершин, рекурсивно применяя к ним операции слияния частей. Все разбиения n можно затем получить из вершин P_n , вычислив все целочисленные точки \mathbb{R}^n , являющиеся выпуклыми комбинациями вершин. В последние два десятилетия достигнуты значительные успехи в решении проблемы подсчета и перечисления целочисленных точек полиэдров, представляющей интерес для многих областей математики. Барвинок [29] предложил полиномиальный при фиксированной размерности пространства алгоритм для подсчета таких точек. Метод Барвинка реализован Де Лоерой с соавторами (см. [30] и дальнейшие ссылки в этой работе) в программе LattE, применимой для произвольных рациональных политопов.

Таким образом, зная вершинные базисы множеств разбиений чисел, состоящие из вершин или опорных вершин политопов разбиений, можно достаточно эффективно построить все разбиения относительно малых n .

Опорные вершины политопа P_n нетрудно выделить из списка $\text{VertList}(n)$. К каждой вершине $x \in \text{vert}P_n$ последовательно, начиная с начала списка, следует применить следующую процедуру:

- 1) если вершина x не помечена как опорная, пометить ее;
- 2) применить к x все возможные операции слияния частей и пометить полученные вершины y как не опорные;
- 3) перейти к обработке следующей вершины x из $\text{VertList}(n)$.

Порядок расположения вершин в списке $\text{VertList}(n)$ гарантирует, что вершины y расположены в нем после x . Поэтому достаточно выполнить один проход по списку $\text{VertList}(n)$.

В результате в $\text{VertList}(n)$ будут помечены все опорные вершины P_n и из них легко составить отдельный список.

4. Построение списка рюкзачных разбиений

Рюкзачные разбиения были введены Эренборгом и Ридди в [31] при исследовании функции Мебиуса частично упорядоченного множества разбиений множества на блоки с ограничениями на размеры. Они назвали так разбиения, все наборы частей которых дают различные суммы. В [27] доказано, что класс рюкзачных разбиений всех чисел совпадает с классом разбиений, не представимых в виде выпуклой комбинации двух разбиений. Все такие разбиения помечены должным образом в списке $\text{PartList}(n)$ в момент окончания проверки условия H7. Число рюкзачных разбиений $k(n)$ связано с числом вершин $v(n)$ политопа P_n равенством

$$k(n) = v(n) + [\text{число разбиений, для выпуклого представления которых необходимы три или более разбиения}],$$

т. е. $k(n) = v(n) + [\text{число тех разбиений } n \text{ из } \text{PartList}(n), \text{ для которых программа Polymake показывает, что они не являются вершинами } P_n]$.

Эренборг и Ридди представили в энциклопедии «Целочисленные последовательности» первые 50 значений функции $k(n)$. Авторам удалось продолжить эту последовательность до $n = 165$ [22].

5. Технические особенности вычислений

Вычисления в программе Polymake выполнялись на виртуальном сервере с двумя процессорами с частотой 1 ГГц каждый под управлением ОС Linux на платформе x86 (i586).

При выполнении программы списки разбиений $\text{PartList}(n)$ хранятся в байтовом представлении. При n порядка 190 файлы достигли размера более 2 Гб и перестали работать стандартные для языка C++ функции для их обработки, реализованные в системных библиотеках ядра ОС Windows. Поэтому пришлось прекратить генерирование разбиений для последующих n . В будущем это препятствие можно обойти, используя 64-разрядные функции. Следует заметить, однако, что предел 2 Гб для размера файлов был достигнут на начальном этапе вычислений, когда генерируемые файлы содержали все разбиения числа n . В дальнейшем авторы оптимизировали алгоритм и не генерировали разбиения n из не вершин, содержащихся в файлах с разбиениями чисел, меньших n . Расчеты показывают, что при такой организации вычислений вполне хватит 2 Гб для $n \leq 255$. Однако при больших n критическим окажется размер в 1 байт, отведенный для каждого слагаемого. Для обработки $n > 255$ придется модернизировать программу, отведя на каждое слагаемое по 2 байта.

Основной причиной остановки вычислений при $n = 105$, когда файлы с данными имели еще достаточно небольшой размер, послужило возрастание времени, затрачиваемого программой Polymake на распознавание разбиений, оставшихся после второго этапа процедуры распознавания вершин и избыточных разбиений и не являющихся вершинами P_n . В данный момент некоторый ресурс времени еще остается, но для дальнейшего существенного продвижения по n придется найти новые алгоритмические решения данной проблемы. Резюмируя, можно сказать, что главную проблему составляет распознавание разбиений, представимых в виде выпуклой комбинации более чем двух разбиений.

6. Результаты вычислений

При исследовании политопов разбиений чисел первостепенный интерес представляет поведение функций числа вершин и числа опорных вершин. Пока неясно, как вывести для них точные формулы или хотя бы достаточно точные аппроксимации. Поэтому авторы стремились получить их значения для как можно больших n . Диаграмма на рис. 2 демонстрирует рост

функций числа разбиений $p(n)$, числа вершин $v(n)$ и числа опорных вершин $s(n)$ политопов P_n в логарифмической шкале для $n \leq 100$.

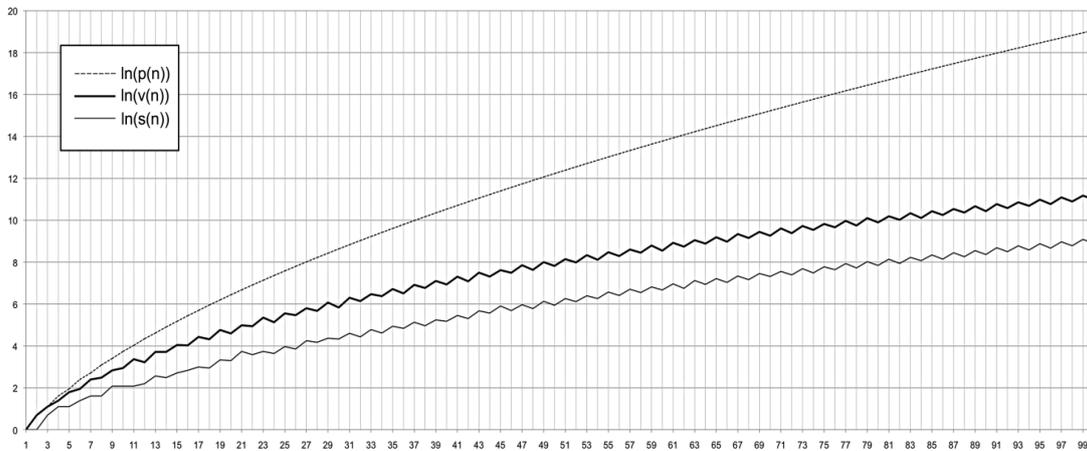


Рис. 2. Диаграмма роста функций $p(n)$, $v(n)$ и $s(n)$

Анализ результатов вычислений приводит к интересным наблюдениям и новым вопросам о структуре множеств разбиений n :

1. Отношения чисел вершин и опорных вершин политопа P_n к числу разбиений n быстро убывают с ростом n . Для $n = 10, 50, 100$ эти числа, соответственно, таковы:

$$p(n) = 42, 204226, 190569292; \quad v(n) = 17, 2488, 59294; \quad s(n) = 8, 380, 7228,$$

а отношения $v(n)/p(n)$ и $s(n)/p(n)$, соответственно, равны 0,45 и 0,19; 0,12 и 0,019; 0,0031 и 0,00038. Легко заметить, что отношение числа опорных вершин к числу всех вершин тоже быстро уменьшается с ростом n . Таким образом, множество всех 190 569 292 разбиений числа $n = 100$ можно построить из множества, содержащего 59 294 вершин политопа P_{100} , и даже всего лишь из 7228 его опорных вершин, что меньше исходного числа разбиений в 330 и более чем в 26 000 раз соответственно. Отметим также, что при $n = 100$ число опорных вершин P_n всего лишь примерно в 72 раза больше самого числа n . При возрастании n от 50 до 100 отношение числа разбиений в PartList, оказавшихся не вершинами P_n , к общему числу разбиений в списке изменяется от 0,52 – 0,64 до 0,63 – 0,73 в зависимости от четности n .

Анализ вычисленных значений показал, что при $n \leq 100$ рост числа вершин вполне удовлетворительно описывается экспоненциальной зависимостью от корня из n , что асимптотически совпадает с формулой (2) для $p(n)$. Более точное определение этой зависимости представляет проблему для дальнейших исследований.

2. Из рис. 2 отчетливо видно, что функции $v(n)$ и $s(n)$ растут не монотонно: начиная с n порядка 10, значение $v(n)$ (так же, как и $s(n)$) для нечетного n существенно больше, чем $v(n+1)$. Можно сказать, что графики функций $v(n)$ и $s(n)$ распадаются на два лежащих один под другим подграфика: для четных и нечетных n , причем расстояние между ними экспоненциально возрастает с ростом n . Объяснение такого «пилообразного» роста исследуемых функций может оказаться очень полезным.

3. При более внимательном рассмотрении обнаруживается, что «подграфик» функции $v(n)$ для нечетных n возрастает тоже неравномерно. Значения $v(n)$ для простых n образуют кривую, лежащую выше значений $v(n)$ для составных нечетных n . Более того, создается впечатление, что величины $v(n)$ для составных нечетных n зависят от свойств делимости чис-

ла n . Каковы именно эти свойства, пока неизвестно. Возможно, основную роль играет число делителей n , или величина наименьшего отличного от нуля делителя n , или некий другой параметр. Число $n = 100$ все еще слишком мало, чтобы сделать определенные выводы. Продолжение вычислений вершин политопов разбиений позволит расширить собранную статистику и многое прояснить в этом вопросе. Если высказанное подозрение оправдается, то основным итогом проделанной работы, по мнению авторов, следует считать обнаружение связи между комбинаторно-геометрическими свойствами политопов разбиений чисел с теоретико-числовыми свойствами разбиваемых чисел.

Заключение

Описанные алгоритм и программа разработаны для продолжения исследования полиэдральной структуры множеств разбиений чисел. Авторы вычислили все вершины и опорные вершины политопов разбиений чисел для $n \leq 105$ в расчете на то, что знание списков вершин поможет обнаружить их новые свойства. Полученные данные уже сейчас привели к постановке новых проблем о связи числовых характеристик политопов разбиений n с теоретико-числовыми свойствами n . Попутно были вычислены все рюкзачные разбиения чисел $n \leq 165$. Вычисленные последовательности чисел вершин и опорных вершин P_n для $n \leq 100$ и чисел рюкзачных разбиений для $n \leq 165$ представлены в энциклопедии «Целочисленные последовательности» [20–22].

Отметим, что вывод приближенных формул для $p(n)$ потребовал длительных и напряженных усилий многих математиков, а также гениальной интуиции Рамануджана. История теории разбиений подтверждает слова Эйлера, сказанные им в 1750 г. в труде «De Partitione Numerorum»: «Всякий, кто намеревается постичь структуру разбиений чисел, обрекает себя на необъятный труд и должен быть готов страдать от постоянного пребывания в состоянии абсолютного внимания, чтобы не оказаться жертвой чудовищного заблуждения» [16]. Маловероятно, что вершины политопов разбиений подчиняются более простым законам, чем сами разбиения.

Список литературы

1. Эндрюс, Г. Теория разбиений / Г. Эндрюс. – М. : Наука, 1982. – 256 с.
2. Эйлер, Л. Введение в анализ бесконечных / Л. Эйлер. – М. : Гос. изд. физ.-мат. лит., 1961. – Т. 1. – 316 с.
3. Dickson, L.E. History of the Theory of Numbers / L.E. Dickson. – Vol. II : Diophantine Analysis. – Washington : Carnegie Inst., 1919. – 520 p.
4. Шлык, В.А. Политопы разбиений чисел / В.А. Шлык // Весці Нац. акад. навук Беларусі. Сер. фіз.-мат. навук. – 1996. – № 3. – С. 89–92.
5. Shlyk, V.A. Polytopes of partitions of numbers / V.A. Shlyk // European J. Combin. – 2005. – Vol. 26, № 8. – P. 1139–1153.
6. Емеличев, В.А. Многогранники, графы, оптимизация (комбинаторная теория многогранников) / В.А. Емеличев, М.М. Ковалев, М.К. Кравцов. – М. : Наука, 1981. – 341 с.
7. Шлык, В.А. Комбинаторные операции порождения вершин политопов разбиений чисел / В.А. Шлык // Докл. Нац. акад. наук Беларусі. – 2009. – Т. 53, № 6. – С. 27–32.
8. Холл, М. Комбинаторика / М. Холл. – М. : Мир, 1970. – 424 с.
9. Gupta, H. Tables of partitions / H. Gupta, C.E. Gwyther, C.P. Winter // Royal society mathematical tables. – Vol. 4. – Cambridge : University Press, 1958. – 132 p.
10. Sloane, N.J.A. a(n) = number of partitions of n (the partition numbers) [Electronic resource] / N.J.A. Sloane. – 2010. – Mode of access : <http://oeis.org/A000041>. – Date of access : 11.10.2015.
11. Zoghbi, A. Fast algorithms for generating integer partitions / A. Zoghbi, I. Stojmenovic // Intern. J. Computer Math. – 1998. – Vol. 70. – P. 319–332.
12. Рейнгольд, Э. Комбинаторные алгоритмы: теория и практика / Э. Рейнгольд, Ю. Нивергельт, Н. Део. – М. : Мир, 1980. – 476 с.

13. Nijenhuis, A. A method and two algorithms on the theory of partitions / A. Nijenhuis, H.S. Wilf // *J. Comb. Theory A.* – 1975. – Vol. 18. – P. 219–222.
14. Riha, W. Efficient algorithms for doubly and multiply restricted partitions / W. Riha, K.R. James // *Algorithm 29. Computing.* – 1976. – Vol. 16. – P. 163–168.
15. Constant time generation of integer partitions / K. Yamanaka [et al.] // *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences.* – 2007. – Vol. E90-A, № 5. – P. 888–895.
16. Knuth, D.E. Generating all partitions. Pre-fascicle 3B of *The Art of Computer Programming. A draft of sections 7.2.1.4-5* [Electronic resource] / D.E. Knuth. – 2004. – Mode of access : <http://www.cs.utsa.edu/~wagner/knuth/fasc3b.pdf>. – Date of access : 11.10.2015.
17. Kelleher, J. Generating all partitions: a comparison of two encodings [Electronic resource] / J. Kelleher, B. O'Sullivan. – 2009. – Mode of access : <http://arxiv.org/pdf/0909.2331v1.pdf>. – Date of access : 11.10.2015.
18. Opdyke, J.D. A unified approach to algorithms generating unrestricted and restricted integer compositions and integer partitions / J.D. Opdyke // *J. Math. Model. Algor.* – 2010. – Vol. 9, № 1. – P. 53–97.
19. Stojmenovic, I. Generating all and random instances of a combinatorial object / I. Stojmenovic // *Handbook of applied algorithms: solving scientific, engineering, and practical problem.* – Hoboken : John Wiley&Sons, 2008. – P. 1–38.
20. Shlyk, V.A. Number of vertices of the integer partition polytope [Electronic resource] / V.A. Shlyk. – 2012. – Mode of access : <http://oeis.org/A203898>. – Date of access : 11.10.2015.
21. Shlyk, V.A. Number of support partitions-vertices [Electronic resource] / V.A. Shlyk. – 2012. – Mode of access : <http://oeis.org/A203899>. – Date of access : 11.10.2015.
22. Ehrenborg, R. Number of knapsack partitions of n [Electronic resource] / R. Ehrenborg. – 2005. – Mode of access : <http://oeis.org/A108917>. – Date of access : 11.10.2015.
23. Шлык, В.А. Критерий представления разбиений чисел в виде выпуклой комбинации двух разбиений / В.А. Шлык // *Вестник БГУ. Сер. 1.* – 2009. – № 2. – С. 109–114.
24. Shlyk, V.A. Integer partitions from the polyhedral point of view / V.A. Shlyk // *Electron. Notes Discrete Math.* – 2013. – Vol. 43. – P. 319–327.
25. Shlyk, V.A. Polyhedral approach to integer partitions / V.A. Shlyk // *Journal of Combinatorial Mathematics and Combinatorial Computing.* – 2014. – Vol. 89. – P. 113–128.
26. Gawrilow, E. Polymake: a framework for analyzing convex polytopes / E. Gawrilow, E.M. Joswig // *Polytopes – combinatorics and computation.* – Basel : Birkhäuser, 2000. – P. 43–73.
27. Шлык, В.А. О вершинах политопов разбиений чисел / В.А. Шлык // *Докл. Нац. акад. наук Беларуси.* – 2008. – Т. 52, № 3. – С. 5–10.
28. *Handbook of Mathematical Functions* / M. Abramowitz, I.A. Stegun [eds]. – Applied Math. Series 55. – Washington : Government Printing Office, 1972. – 1046 p.
29. Barvinok, A.I. Polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed / A.I. Barvinok // *Math. Oper. Res.* – 1994. – Vol. 19. – P. 769–779.
30. Effective lattice point counting in rational convex polytopes / J.A. De Loera [et al.] // *Journal of Symbolic Computation.* – 2004. – Vol. 38, № 4. – P. 1273–1302.
31. Ehrenborg, R. The Möbius function of partitions with restricted block sizes / R. Ehrenborg, M.A. Readdy // *Adv. in Appl. Math.* – 2007. – Vol. 39, № 3. – P. 283–292.

Поступила 07.09.2015

¹ИООО «Управляющая компания "Атлант-М"»,
Минск, Шаранговича, 22-А
e-mail: alvrub@gmail.com

²Командно-инженерный институт МЧС Республики Беларусь,
Минск, Машиностроителей, 25
e-mail: v.shlyk@gmail.com

A.S. Vroublevski, V.A. Shlyk

COMPUTING VERTICES OF INTEGER PARTITION POLYTOPES

The paper describes a method of generating vertices of the polytopes of integer partitions that was used by the authors to calculate all vertices and support vertices of the partition polytopes for all $n \leq 105$ and all knapsack partitions of $n \leq 165$. The method avoids generating all partitions of n . The vertices are determined with the help of sufficient and necessary conditions; in the hard cases, the well-known program Polymake is used. Some computational aspects are exposed in more detail. These are the algorithm for checking the criterion that characterizes partitions that are convex combinations of two other partitions; the way of using two combinatorial operations that transform the known vertices to the new ones; and employing the Polymake to recognize a limited number (for small n) of partitions that need three or more other partitions for being convexly expressed. We discuss the computational results on the numbers of vertices and support vertices of the partition polytopes and some appealing problems these results give rise to.

УДК 519.8, 004.9

А.Г. Гривачевский¹, В.Д. Гривачевский², М.Я. Ковалев¹, А.И. Сухомлинов²

ПЛАНИРОВАНИЕ ПРОЦЕССА ИЗГОТОВЛЕНИЯ СЕКЦИИ КОРПУСА СУДНА

Рассматривается задача планирования процесса изготовления секции корпуса судна, которая сводится к классической в теории расписаний трехстадийной задаче Джонсона. Описывается генетический алгоритм решения и компьютерный эксперимент для сравнения эффективности этого алгоритма и алгоритма полного перебора.

Введение

Составление производственных планов и расписаний является одной из актуальных задач управления деятельностью судостроительного предприятия. В статье рассматривается процесс изготовления деталей секции корпуса судна и предлагается сведение задачи оптимального планирования этого процесса к классической задаче теории расписаний для трех последовательных станков, обозначаемой $F3//C_{\max}$, – трехстадийной задаче Джонсона [1]. Поскольку задача является NP -трудной в сильном смысле [2], для ее решения предлагается генетический алгоритм.

1. Практическая ситуация и трехстадийная задача Джонсона

В основу современной организации производства сварного корпуса судна положен принцип декомпозиции конструкции корпуса на секции и независимого параллельного изготовления этих секций с последующей сборкой их в общий корпус. Рассмотрим задачу планирования процесса изготовления одной секции корпуса судна. Этот процесс включает изготовление необходимых для секции узлов из листового металлопроката и их сварку. Каждый узел состоит из определенного набора деталей. Все детали при изготовлении проходят три производственные стадии:

1. Разметка детали на металлическом прокате.
2. Вырезка детали.
3. Обработка детали после вырезки.

Детали проходят указанные три стадии в одной и той же последовательности: 1, 2, 3. На одной и той же стадии производственные операции над различными деталями выполняются последовательно и не могут пересекаться по времени. Кроме того, операции различных стадий над одной и той же деталью не могут пересекаться во времени, однако операции над различными деталями могут выполняться параллельно во времени на различных стадиях. Предположим, что количество деталей равно n , они обозначаются $j = 1, \dots, n$, и для каждой детали j известны длительности p_{ij} выполнения соответствующей операции на стадии i , $i = 1, 2, 3$.

В табл. 1 приведен пример значений длительностей операций при изготовлении 10 деталей секции корпуса судна на Дальневосточном заводе «Звезда». Объектом планирования является расписание выполнения операций над n деталями на указанных трех стадиях, которое, в свою очередь, определяется указанием моментов времени C_{ij} завершения выполнения операций над деталями j на стадиях i . Пусть C_j обозначает момент завершения последней операции над деталью j , т. е. $C_j = C_{3j}$. Исходя из предположения, что выполнение любой операции может начаться в момент времени ноль, задача заключается в поиске расписания, минимизирующего момент времени C_{\max} завершения выполнения последней операции над всеми деталями, $C_{\max} = \max\{C_{3j}/j=1, \dots, n\}$, называемый также длиной расписания.

Таблица 1

Длительности выполнения операций, мин

Номер детали	Разметка	Вырезка	Обработка
1	4	5	8
2	3	6	7
3	9	1	4
4	6	9	1
5	4	1	6
6	7	8	9
7	1	7	6
8	8	1	8
9	3	1	3
10	4	9	1

Сформулированная задача как классическая задача теории расписаний $F3//C_{\max}$ была впервые поставлена Джонсоном [1], который доказал, что оптимальное расписание находится в классе так называемых перестановочных расписаний, для которых последовательность операций на каждой стадии одинакова и определяется последовательностью деталей.

2. Генетический алгоритм

Для решения задачи $F3//C_{\max}$ предлагается воспользоваться генетическим алгоритмом, суть которого заключается в генерации случайным образом множества (начальной популяции) последовательностей деталей (хромосом) и получении решения (искомой оптимальной последовательности деталей) при помощи генетических операторов скрещивания и мутации. Генетические алгоритмы не гарантируют построение оптимального решения, однако в большинстве случаев позволяют отыскать решение, близкое к оптимальному. Основы генетических алгоритмов описаны Холландом [3]. Анализ литературы показывает эффективность и целесообразность применения данного метода для решения задач планирования работ современных судостроительных предприятий [4, 5].

Хромосомой разрабатываемого генетического алгоритма будем считать последовательность деталей (например, 2, 3, 7, 6, 4, 1, 10, 5, 9, 8).

В качестве фитнес-функции используется целевая функция C_{\max} , которая для любой последовательности деталей может быть вычислена рекурсивно следующим образом. Не ограничивая общности, рассмотрим последовательность $(1, \dots, n)$. Введем вспомогательные величины fin_1 , fin_2 и fin_3 . Вначале полагаем $fin_1 = 0$, $fin_2 = 0$, $fin_3 = 0$. Далее для $j = 1, \dots, n$ рекуррентно вычисляем

$$\begin{aligned} fin_1 &:= fin_1 + p_{1j}, \\ fin_2 &:= fin_2 + p_{2j} + \max\{fin_1 - fin_2, 0\}, \\ fin_3 &:= fin_3 + p_{3j} + \max\{fin_2 - fin_3, 0\}. \end{aligned}$$

Значение фитнес-функции $C_{\max} = fin_3$ при $j = n$.

Скрещивание и мутация осуществляются только для одной и той же хромосомы, так как могут возникнуть ситуации, когда при скрещивании двух различных хромосом или мутации полученная хромосома (последовательность деталей) будет содержать повторяющиеся номера деталей.

При скрещивании для определенной хромосомы выбирается точка между номерами соседних генов (позиций в перестановке деталей), относительно которой происходит обмен местами двух частей родительской хромосомы. Допустим, для хромосомы, представленной на рис. 1, выбирается точка скрещивания между четвертым и пятым генами, в результате скрещивания получаем новую хромосому. При этом в новую популяцию попадают обе хромосомы – «родитель» и «потомок».

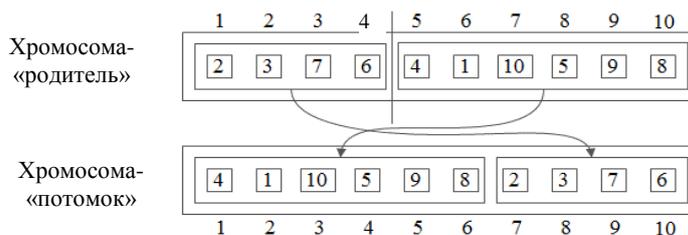


Рис. 1. Скрещивание хромосом в разработанном методе

При мутации для определенной хромосомы выбираются номера генов, которые затем производят обмен значениями между собой. Например, для хромосомы, представленной на рис. 2, выбираются номера генов 3 и 8 для мутации, в результате чего получаем новую хромосому. При этом в новую популяцию попадают обе хромосомы – до и после мутации.

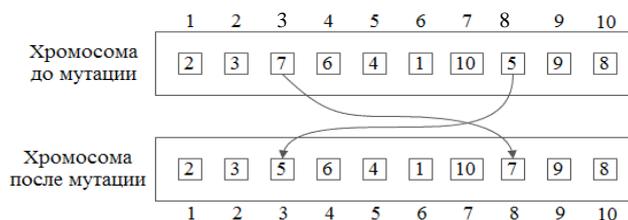


Рис. 2. Мутация хромосом в разработанном методе

Для исходных данных в табл. 1 с помощью описанных операций скрещивания и мутации хромосом были получены пять различных расписаний одинаковой длины $C_{\max} = 59$ мин (рис. 3). В целях практического улучшения получаемых расписаний был введен дополнительный критерий Idle (простой), значением которого является суммарное время промежуточного простоя в минутах для всех трех стадий между моментами начала и завершения работ для каждой стадии. На рис. 3 промежуточные простои обозначаются символом 0.

Заметим, что для любого расписания операции первой стадии могут быть смещены по оси времени максимально влево и операции третьей стадии – максимально вправо так, что расписание останется допустимым, значение C_{\max} не увеличится, а промежуточные простои на первой и третьей стадиях исчезнут. Все построенные алгоритмом расписания трансформируем указанным образом (рис. 4). Для расписания 4, а значение критерия Idle равно 6 мин, для 4, б – 4 мин, для 4, в – 6 мин, для 4, г – 6 мин, для 4, д – 0 мин. Среди расписаний с одинаковой длиной предпочтительным является расписание с наименьшим промежуточным простоем.

Предлагается рассмотреть двухмерную фитнес-функцию (C_{\max} , Idle) с лексикографическим отношением предпочтения, таким, что значение C_{\max} является первичным (более значимым), а значение Idle – вторичным. Функцию Idle необходимо минимизировать на множестве расписаний, минимизирующих C_{\max} .

Для определения экономической выгоды использования разработанного метода был проведен сравнительный анализ наихудшего расписания, соответствующего последовательности 9, 3, 8, 4, 7, 10, 6, 1, 2, 5, и расписания на рис. 4, д, полученного с помощью генетического алгоритма. Сравнительный анализ этих расписаний приведен в табл. 2.

Таблица 2

Сравнение полученного и наихудшего расписаний

Параметр	Наилучший план	Наихудший план
Время окончания работ по разметке деталей, мин	49	49
Время окончания работ по вырезке деталей, мин	52	71
Время окончания работ по обработке деталей, мин	59	89
Итоговое время, мин	59	89
Промежуточный простой при вырезке деталей, мин	0	20

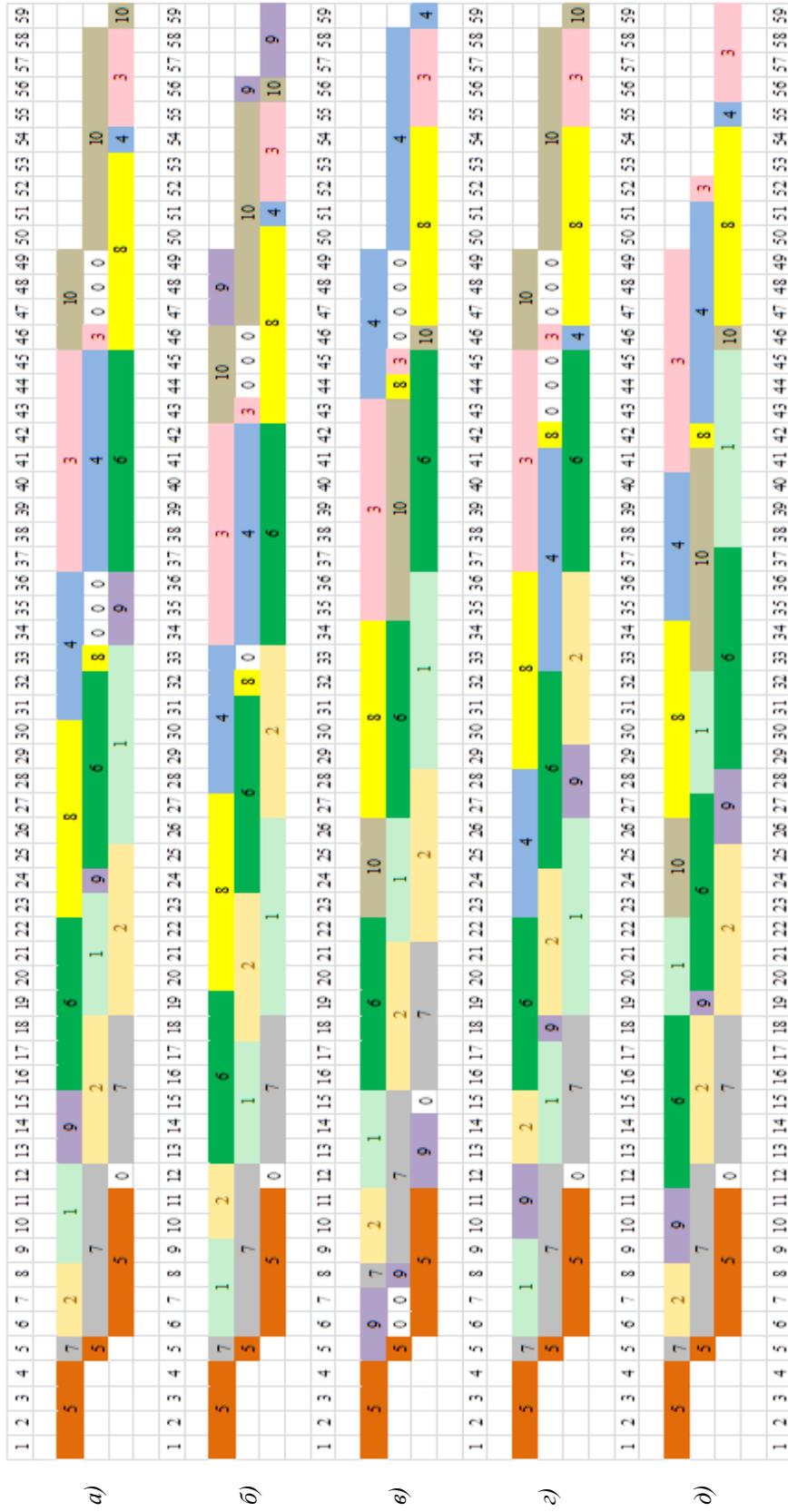


Рис. 3. Диаграммы Ганта для расписаний производственного процесса, полученных с помощью генетического алгоритма

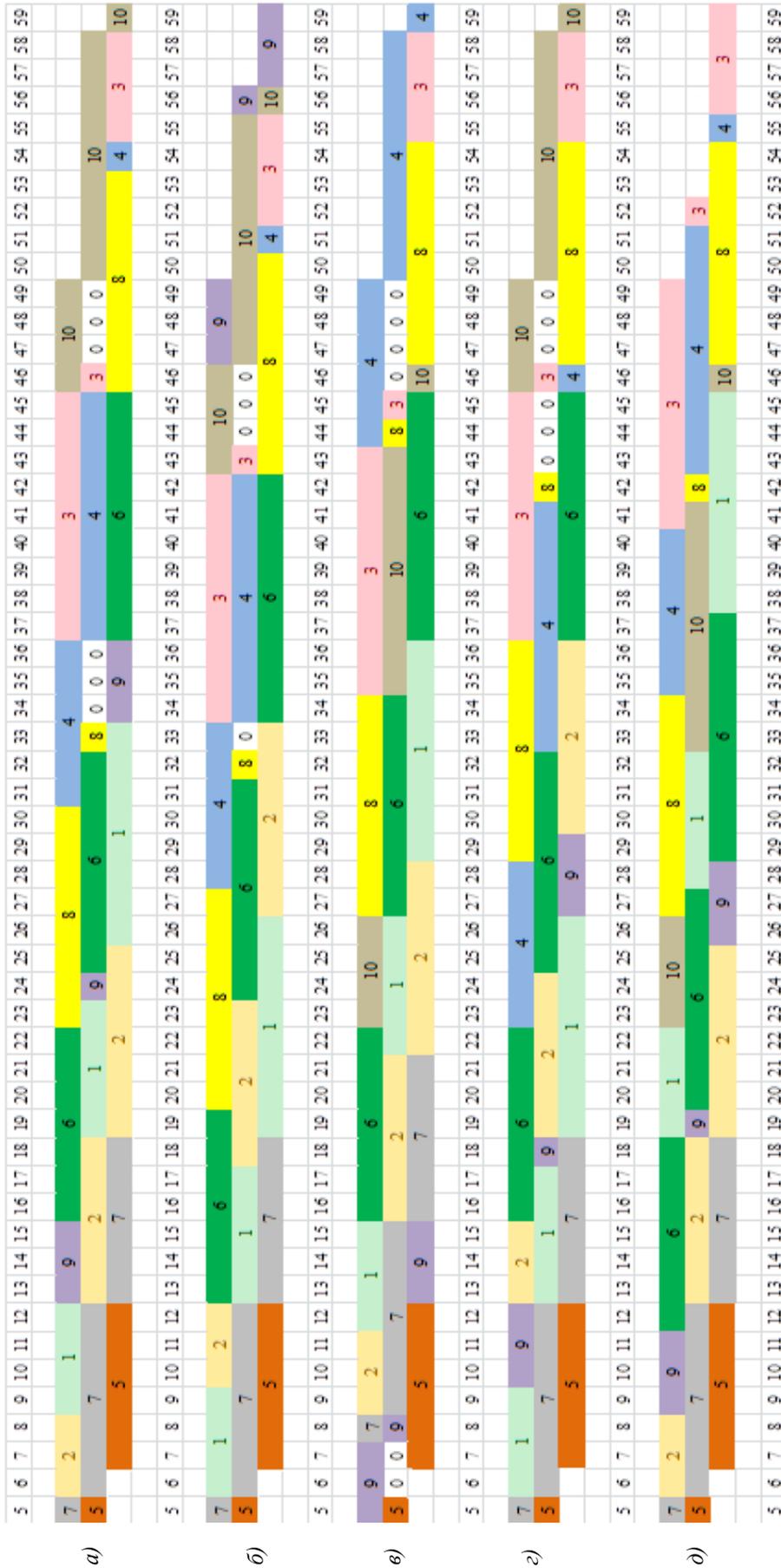


Рис. 4. Диаграммы Ганта для расписаний производственного процесса, полученных с помощью генетического алгоритма, со смещением времени начала работ на первой и третьей стадиях

Исходя из полученных результатов, можно сделать вывод, что по критерию итогового времени наилучшее расписание обеспечивает изготовление всех деталей быстрее на 33,7 %, снижая промежуточный простой с 20 мин до нуля (на 100 %).

Генетический алгоритм и алгоритм полного перебора были разработаны в среде Microsoft Visual Studio 2012 на языках C# и C++ соответственно. Параметры генетического алгоритма: размер популяции – 1000 особей, количество итераций – 1000.

3. Вычислительные эксперименты

Вычислительные эксперименты проводились на персональном компьютере с операционной системой Windows 7 Ultimate SP1 32-bit, центральным процессором Intel Core 2 DUO E8400 3,00 ГГц и оперативной памятью 4 Гб. Было проведено исследование скорости нахождения решения для исходных данных из табл. 1 с помощью генетического алгоритма и алгоритма полного перебора. При заданном количестве деталей n входом являлись детали 1, ..., n . Сравнительные результаты экспериментов представлены в табл. 3. Отметим, что во всех случаях генетический алгоритм нашел оптимальное решение.

Таблица 3

Результаты экспериментов

Количество деталей	Количество перестановок	Генетический алгоритм		Полный перебор	
		Время работы, с	C_{\max} , Idle	Время работы, с	C_{\max} , Idle
4	$4! = 24$	5	29,0	< 1	29,0
5	$5! = 120$	6	33,4	< 1	33,4
6	$6! = 720$	7	42,4	1	42,4
7	$7! = 5040$	8	47,1	1	47,1
8	$8! = 40\,320$	9	55,2	3	55,2
9	$9! = 362\,880$	10	58,4	26	58,4
10	$10! = 3\,628\,800$	11	59,0	260	59,0

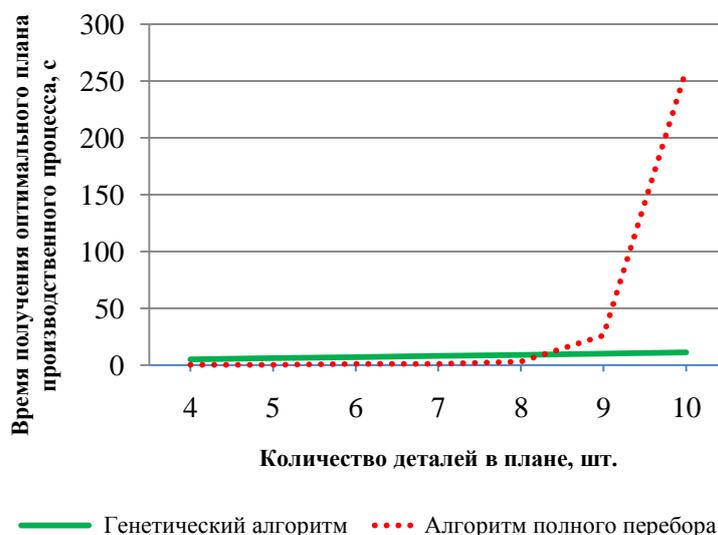


Рис. 5. График зависимости времени получения решения от количества деталей

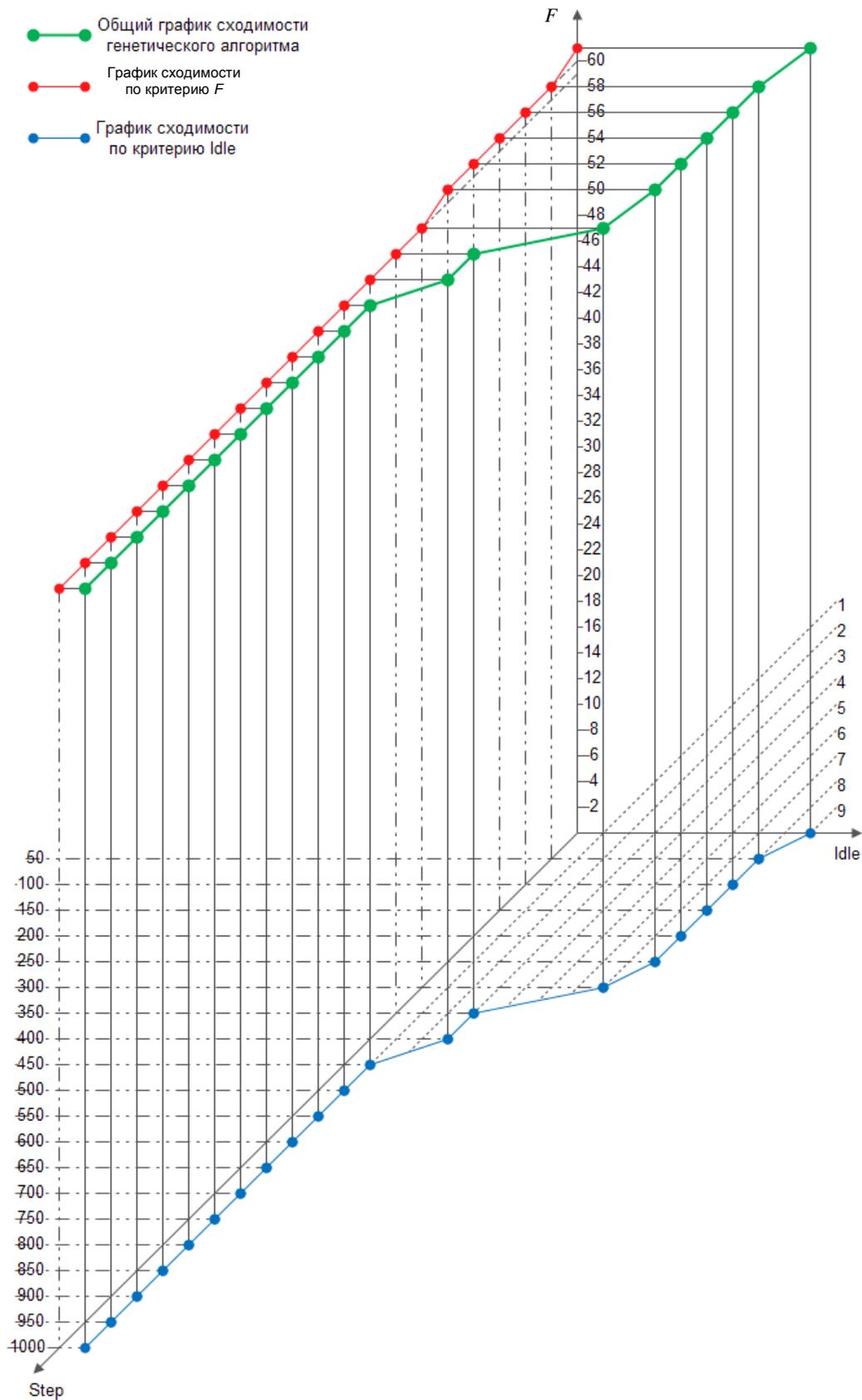


Рис. 6. График сходимости генетического алгоритма

Из рис. 5 видно, что на полный перебор при количестве деталей от 4 до 8 затрачивается в разы меньше времени, чем на реализацию генетического алгоритма. Однако, учитывая, что масштабы производства достаточно велики, при большом количестве деталей генетический алгоритм сохраняет практически линейную зависимость времени получения решения от количества деталей, тогда как время, необходимое для полного перебора, увеличивается на порядок с каждой дополнительной деталью.

На рис. 6 представлен график сходимости генетического алгоритма по критериям C_{\max} (ось F) и $Idle$ в зависимости от номера шага выполнения алгоритма (ось $Step$) для примера на рис. 4, d . Результирующая кривая есть функция от двух переменных. Для удобства восприятия сходимости по каждому из критериев на рис. 6 представлены также проекции общего графика сходимости на плоскость F - $Step$ для критерия C_{\max} и на плоскость $Idle$ - $Step$ для критерия $Idle$. Видно, что решение было получено на 401-м – 450-м шагах выполнения алгоритма.

Заключение

Применение современных методов автоматизированного планирования производственного процесса, основанных на генетических алгоритмах, позволяет получить оптимальные и субоптимальные планы, графики и расписания деятельности предприятия за приемлемое время, что приводит к сокращению затрат временных и материальных ресурсов, тем самым увеличивая эффективность производства и производительность труда инженеров-технологов.

Список литературы

1. Johnson, S.M. Optimal two- and three-stage production schedules with set-up time included / S.M. Johnson // *Naval Research Logistics Quarterly*. – 1954. – Vol. 1. – P. 61–68.
2. Гэри, М. Вычислительные машины и труднорешаемые задачи / М. Гэри, Д. Джонсон. – М. : Наука, 1979.
3. Holland, J. *Adaptation in natural and artificial systems* / J. Holland. – Cambridge : MIT Press, 1992.
4. Xu, X. Computer-aided process planning – A critical review of recent developments and future trends / X. Xu, L. Wang, S.T. Newman // *International Journal of Computer Integrated Manufacturing*. – 2011. – Vol. 24, № 1. – P. 1–31.
5. Qassim, R.Y. Integrated process planning and scheduling and multimode resource constrained project scheduling: ship block assembly application / R.Y. Qassim // *Manufacturing System*. – 2012. – P. 307–314.

Поступила 10.09.2015

¹Объединенный институт проблем информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: griva@newman.bas-net.by

²Дальневосточный федеральный университет,
Владивосток, ул. Суханова, 8

A.G. Grivachevsky, V.D. Grivachevsky, M.Y. Kovalyov, A.I. Suhomlinov

PLANNING VESSEL BODY SECTION PRODUCTION

A problem of planning production of a vessel body section is considered. The problem is reduced to the classic Johnson's tree-machine flow-shop scheduling problem. A genetic algorithm and computer experiment to compare efficiency of this algorithm and the algorithm of full enumeration are described.

УДК 658.512.2:621.9.06

Н.Н. Гущинский, В.Е. Зданович, Б.М. Розин

ОПТИМИЗАЦИЯ РАЗМЕЩЕНИЯ ДЕТАЛИ НА МНОГОПОЗИЦИОННОМ ПОВОРОТНОМ СТОЛЕ АГРЕГАТНОГО СТАНКА

Рассматривается задача минимизации массы агрегатного станка с многопозиционным поворотным столом за счет размещения детали на столе при обработке деталей одного наименования. Предлагаются математическая модель и эвристический алгоритм «роя частиц» для решения данной задачи. Приводятся результаты численных экспериментов по решению предложенным методом двух задач этого типа. Эксперименты показали большую эффективность алгоритма «роя частиц» для решения рассматриваемых задач по сравнению с методами случайного поиска и ЛП-последовательностей.

Введение

Одной из основных задач при проектировании агрегатного станка (АС) является формирование его компоновки. Проектирование компоновки АС при заданных структуре и параметрах технологического процесса (ТП) обработки деталей одного наименования заключается в определении состава, типоразмеров основных силовых и корпусных узлов, а также параметров их взаимного расположения. При этом должны учитываться характерные для рассматриваемого типа АС технологические, конструктивные и эргономические ограничения.

Решению задач автоматизации выбора компоновок различных типов станков, в том числе АС, посвящен ряд публикаций [1–13]. В этих публикациях описана разработанная в ОИПИ НАН Беларуси подсистема «Компоновка» программного комплекса «Агрегат», предназначенная для автоматизации построения компоновок АС, формируемых по блочно-модульному принципу нескольких компоновочных схем с приспособлением для установки деталей на многопозиционных поворотных столах, однопозиционных стационарных и двухпозиционных передвижных столах. Формирование компоновки при помощи этой подсистемы осуществляется в автоматизированном режиме с учетом информации о структуре и параметрах ТП, положении обрабатываемой детали на виртуальном приспособлении АС выбранного типа, данных об основных параметрах приспособления и инструментальной наладки. Для полученного варианта компоновки АС определяются его цена, масса, высота и занимаемая площадь.

Кроме структуры и параметров ТП существенное влияние на формируемую компоновку АС оказывает расположение обрабатываемой детали на позициях обработки АС. При этом в ряде случаев для некоторых узлов АС расположение обрабатываемой детали на приспособлении не влияет на выбор их типоразмеров. К таким узлам относятся, в частности, силовые бабки и вспомогательное оборудование (инструментальные шкафы, стружкосборники, гидростанции). Их типоразмеры выбираются минимальными, удовлетворяющими условию обеспечения выполнения с требуемой точностью технологических переходов и функционирования АС в целом. Типоразмеры других унифицированных узлов и сборочных единиц, таких, например, как многошпиндельные коробки, силовые и крестовые столы, на которых размещаются эти коробки и силовые бабки, либо размеры оригинальных узлов, таких как центральные и боковые станины, стойки и подставки, как правило, существенно зависят от размещения на приспособлении обрабатываемой детали [14].

В подсистеме «Компоновка» установка 3D-модели обрабатываемой детали на загрузочной позиции АС производится в ручном режиме. Для построения компоновки АС пользователю приходится устанавливать на позиции стола 3D-модель детали и проводить расчеты многократно, причем в результате этого процесса не гарантируется построение даже допустимой компоновки. На поиск же оптимальной (по выбранному критерию оптимальности) компоновки может потребоваться значительное время. В связи с этим актуальной является задача автоматизации поиска такого размещения обрабатываемой детали на приспособлении, при котором дос-

тается минимальное значение критерия качества компоновки АС. Критериями качества компоновки в рассматриваемой задаче могут быть выбраны такие характеристики, как материалоемкость, габариты (высота, занимаемая площадь) либо стоимость. Эти характеристики являются, как правило, возрастающими функциями типоразмеров узлов АС. В настоящей работе рассматривается однокритериальная задача оптимизации компоновки АС, в которой в качестве критерия оптимальности выбрана масса станка.

1. Постановка задачи

Рассматривается АС для обработки однородной партии деталей с приспособлением на многопозиционном поворотном делительном столе с вертикальной осью вращения и числом боковых приставок до пяти. Среди боковых приставок имеются горизонтальные приставки с силовыми либо крестовыми столами и установленными на них силовыми бабками и многошпиндельными коробками и может быть одна вертикальная приставка, предназначенная для вертикальной параллельной обработки деталей на нескольких позициях стола (рис. 1). Заданы количество позиций поворотного стола и количество горизонтальных приставок, выполняющих обработку деталей на соответствующих позициях и пристыкованных к фиксированным граням средней станины. Для АС этого типа расположение детали в плоскости стола существенно влияет на типоразмеры выбираемых узлов, особенно узлов вертикальной боковой приставки.

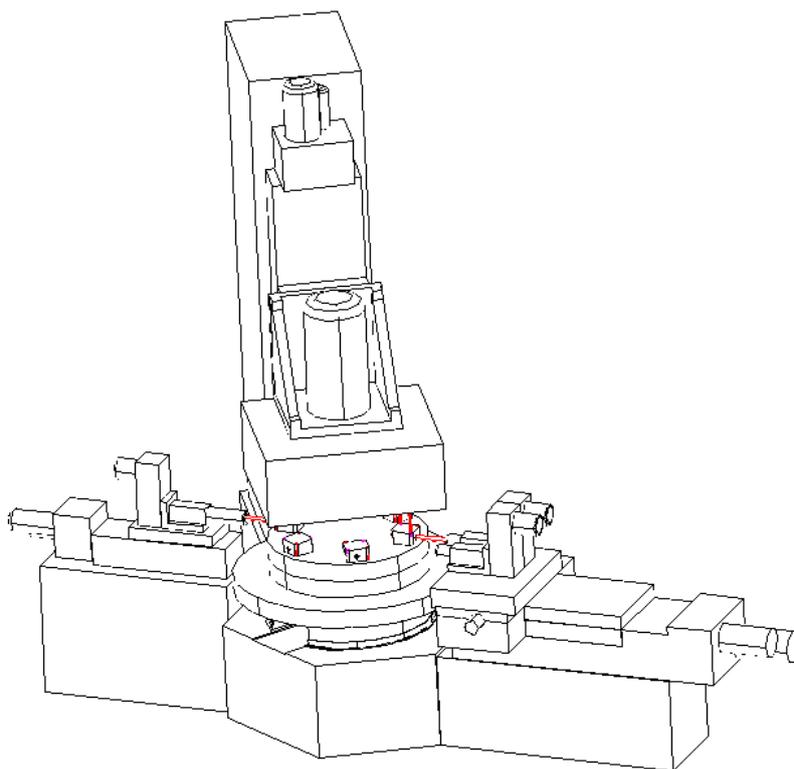


Рис. 1. АС с поворотным столом

Вертикальная приставка может быть размещена на любой из предусмотренных для обработки граней центральной станины АС, не занятой горизонтальными приставками. Основные компоновочные параметры виртуального приспособления для размещения деталей (которое еще предстоит спроектировать) задаются пользователем в режиме диалога. К таким параметрам относятся: диаметр планшайбы поворотного стола, диаметр приспособления, длины баз инструментов и их съема, а также толщина литья шпиндельной коробки для силовых столов, на которых они устанавливаются.

Ограничения, которым должно удовлетворять размещение детали, можно разделить на три группы:

- 1) компоновочные;
- 2) конструктивно-технологические;
- 3) ограничения на размещение зажимного приспособления детали на загрузочной позиции.

К компоновочным относятся ограничения на взаимное расположение узлов каждой приставки и узлов различных приставок между собой. Конструктивно-технологические ограничения для шпиндельной коробки включают условия на минимальные межшпиндельные расстояния (расстояния между осями пары шпинделей в плоскости, ортогональной их осям), ограничения на размещение осей шпинделей относительно границ габаритных размеров литья шпиндельной коробки и относительно направляющих кондукторной плиты. Вторая группа ограничений для других типовых узлов АС основана на опыте их проектирования на ведущих предприятиях станкостроения Беларуси ОАО «Минский завод автоматических линий им. П.М. Машерова» и ОАО «Барановичский завод автоматических линий». Третья группа ограничений включает ограничения невыхода виртуального зажимного приспособления детали за границы сектора позиции поворотного стола. Более подробно первая и вторая группы ограничений описаны в работе [11].

Перечислим более детально конструктивные ограничения, учитываемые при выборе габарита литья шпиндельной коробки:

– межшпиндельное расстояние должно превышать минимально допустимое, зависящее от параметров (усилий, угловых скоростей) выполняемых соответствующими инструментами технологических переходов;

– расстояния от оси шпинделя до краев литья коробки и до направляющих втулок комплектов деталей для крепления штанг кондукторных плит должны превышать заданные табличные значения (рис. 2);

– значения параметров (высота H , ширина B) габарита литья выбираются из ряда пар типовых величин, соответствующих возможным реализациям $tp-isp$ (tp – типоразмер, isp – исполнение) шпиндельной коробки, $tp \in \{УНЗ3111, \dots, УНЗ3114\}$, $isp \in \{01, \dots, 28\}$ (табл. 1).

Замечание. Следует обратить внимание на существенное различие влияния ограничений на минимальное межшпиндельное расстояние для горизонтальных и вертикальной многошпиндельных коробок. Поскольку каждая горизонтальная шпиндельная коробка выполняет обработку на единственной соответствующей ей позиции, нарушение такого ограничения хотя бы для одной пары ее шпинделей приводит к нереализуемости коробки (для заданной структуры технологического процесса), поскольку межшпиндельное расстояние определяется расположением осей отверстий на плоскости детали и набором технологических переходов этой позиции. Для вертикальной же шпиндельной коробки, выполняющей параллельную обработку на нескольких позициях, межшпиндельные расстояния для шпинделей, выполняющих обработку на различных позициях поворотного стола, могут быть увеличены за счет изменения расположения детали на столе.

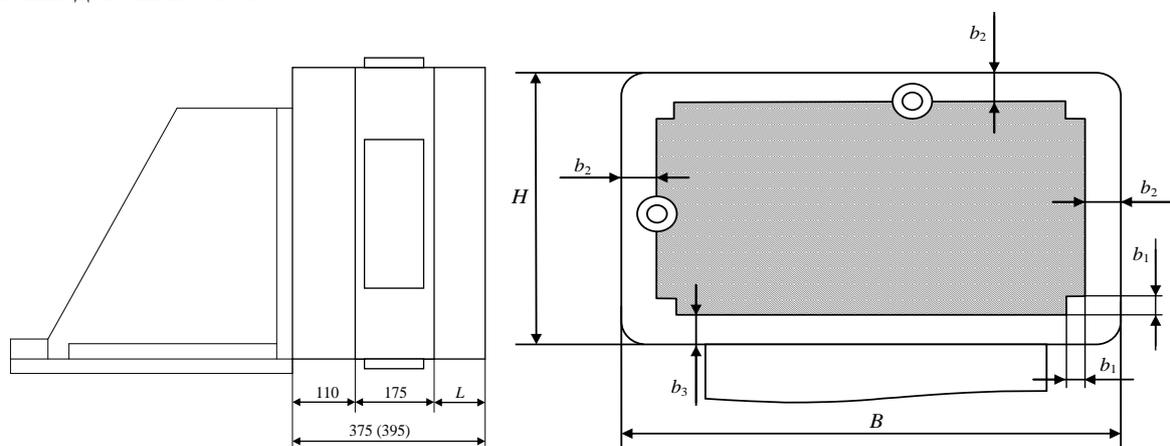


Рис. 2. Упрощенное изображение литья шпиндельной коробки

Таблица 1

Типоразмеры и масса литья шпиндельной коробки

Комплект литья			Габариты литья, мм		Ориентировочная масса, кг
Типоразмер	Исполнение		H	B	
	L, мм				
	90	110			
УНЕЗ111	-01	-02	360	400	240
	-03	-04	400	500	400
	-05	-06	450		440
	-07	-08		540	
УНЕЗ112	-01	-02	500	630	600
	-03	-04		800	800
	-05	-06	560	630	670
	-07	-08		800	830
	-09	-10		1000	1020
	-11	-12	630	800	920
	-13	-14		1000	1200
	-15	-16		800	1020
	-17	-18	710	1000	1250
	-19	-20		1250	1570
	-21	-22	800	1000	1430
	-23	-24		1250	1850
	-25	-26	900	1000	1580
	-27	-28		1250	1950
	УНЕЗ113	-01		-02	1000
-03		-04	1250	2130	
-05		-06	1600	2500	
-07		-08	1120	1250	2370
-09		-10	1250		2800
УНЕЗ114	-01	-02	560	900	950
	-03	-04		1250	1400
	-05	-06	710	1400	1800
	-07	-08	800	800	1300
	-09	-10	900	1400	2150
	-11	-12	1000	1000	1800
	-13	-14	1400	1400	2900

Снижение массы станка достигается за счет уменьшения масс составляющих его узлов или, что эквивалентно, уменьшения их типоразмеров. При этом, основываясь на опыте разработки соответствующего семейства металлообрабатывающих АС, предполагается, что изменение типоразмеров узлов в заданном диапазоне не приводит к снижению точности и надежности обработки. Размещение на загрузочной позиции АС детали при заданной ее ориентации однозначно определяется координатами начала связанной с деталью локальной системы координат (ЛСК) в системе координат АС, начало которой совмещено с центром поворотного стола.

Рассматриваемая в настоящей работе задача сводится к определению допустимых значений координат начала ЛСК обрабатываемой детали на загрузочной позиции поворотного стола с вертикальной осью вращения, а также выбору грани средней станины для установки боковой приставки с вертикальной шпиндельной коробкой, минимизирующих массу АС.

2. Математическая модель

В рассматриваемой задаче управляемыми переменными являются координаты начала ЛСК детали в системе координат АС и номер грани центральной станины для установки боковой приставки с вертикальной шпиндельной коробкой. Начало правой декартовой трехмерной

системы координат АС совмещено с центром поворотного стола, ее ось Oz является нормалью к плоскости поворотного стола, ось Oy направлена вдоль биссектрисы внешнего угла сектора загрузочной позиции.

Предполагается, что расположение ЛСК детали на загрузочной позиции АС отличается от расположения системы координат АС только координатами ее начала. Введем необходимые в дальнейшем обозначения:

$x_{лск}, y_{лск}, z_{лск}$ – координаты начала ЛСК детали, расположенной на загрузочной позиции, в системе координат АС;

$N_{поз}$ – число позиций АС, включая загрузочную;

$N_{зр}$ – число граней средней станины (границ нумеруются по часовой стрелке, начиная с нулевой грани для загрузочной позиции по $N_{зр}-1$);

$N_{зоп}$ – число боковых приставок с горизонтальными силовыми столами;

$\{n_1, n_2, \dots, n_{N_{зоп}}\}$ – номера граней, занятых горизонтальными силовыми столами;

$\{q_1, q_2, \dots, q_0\}$ – номера свободных граней средней станины для размещения вертикальной боковой приставки, $\theta = N_{зр} - N_{зоп}$;

$N_{конт}$ – число точек излома кусочно-линейной ломаной, описывающей контур детали (либо ее зажимное приспособление);

Δ_x^l, Δ_y^l – координаты l -й точки контура детали в плоскости Oxy ЛСК детали, $l=1, 2, \dots, N_{конт}$;

$m_{пр}^k(x_{лск}, y_{лск}, z_{лск})$ – масса k -й горизонтальной боковой приставки, $k=1, 2, \dots, N_{зоп}$;

$m_{пр}^{верт}(x_{лск}, y_{лск}, z_{лск})$ – масса вертикальной боковой приставки;

$m_{ср. ст.}(x_{лск}, y_{лск})$ – масса средней станины АС;

$m_{пов. ст.}(x_{лск}, y_{лск})$ – масса поворотного стола АС.

Для рассматриваемых АС с приспособлением на поворотном делительном столе предусмотрено использование одногранных (четырёхгранных с одной обработанной гранью, предназначенной для установки приставки), четырёхгранных, шестигранных или восьмигранных типовых средних станин под поворотный стол. Наибольшее число боковых приставок для таких средних станин не превышает пяти.

Основные компоновочные параметры (величины баз инструментов боковых приставок, длина съема инструментов, радиус защитного короба приспособления и др.) вводятся пользователем в режиме диалога до выполнения процедуры размещения детали. Первоначальное размещение детали на загрузочной позиции начинается с определения координаты $z_{лск}$ высоты ЛСК детали над плоскостью поворотного стола, которая выбирается, как правило, наименьшей возможной и в дальнейшем не меняется в процессе поиска размещения детали. Вследствие этого координата $z_{лск}$ исключается из состава управляемых переменных задачи.

Далее описываются наиболее существенные ограничения, которым должны удовлетворять координаты $x_{лск}, y_{лск}$ ЛСК детали при ее размещении и которые накладываются проектируемой вертикальной шпиндельной коробкой.

Для описания этих ограничений введем обозначения:

n – число шпинделей шпиндельной коробки;

$I = \{1, 2, \dots, n\}$ – множество индексов шпинделей шпиндельной коробки;

$\alpha(i)$ – номер позиции, на которой осуществляется обработка шпинделем $i \in I$;

x_i, y_i – координаты оси шпинделя $i \in I$ на плоскости поворотного стола в системе координат АС (однозначно определяются по номеру σ конструктивно-технологического элемента (КТЭ), обрабатываемого шпинделем i , номеру $\alpha(i)$ позиции, на которой осуществляется обработка детали этим шпинделем, и координатам $x_{лск}, y_{лск}$ начала ЛСК детали на загрузочной позиции);

$\delta(i, j)$ – минимальное межшпиндельное расстояние для пары шпинделей $i, j \in I$;

b_1, b_2, b_3 – минимальные расстояния оси шпинделя от края литья и комплектов деталей для крепления штанг кондукторных плит (см. рис. 2);

$q \in \{q_1, q_2, \dots, q_0\}$ – номер грани центральной станины, сопоставленной вертикальной боковой приставке;

$\beta = -2\pi q/N_{ep}$ – угол поворота системы координат АС для совмещения оси Oy с осью симметрии грани q центральной станины, на которой располагается вертикальная приставка;

$X_{max}(q, x_{лск}, y_{лск}), X_{min}(q, x_{лск}, y_{лск}), Y_{max}(q, x_{лск}, y_{лск}), Y_{min}(q, x_{лск}, y_{лск})$ – координаты плоскостей, ортогональных соответственно координатным осям Ox, Oy системы координат АС, повернутой относительно вертикальной оси Oz на угол β , которые ограничивают поверхность литья шпиндельной коробки вертикальной боковой приставки на грани q ;

$H=H(q, x_{лск}, y_{лск})$ – высота литья шпиндельной коробки вертикальной приставки;

$B=B(q, x_{лск}, y_{лск})$ – ширина литья шпиндельной коробки вертикальной приставки;

$\mathfrak{R}=\{(H, B)_{tp,isp} \mid tp \in \{УН3111, \dots, УН3114\}, isp \in \{01, \dots, 28\}, \text{ по всем сочетаниям } tp-isp \text{ из табл. 1}\}$ – множество пар габаритных размеров H и B для возможных реализаций литья коробки;

$N_{ктэ}$ – число КТЭ обрабатываемой детали;

$\Delta_{ктэ, x}^{\sigma}, \Delta_{ктэ, y}^{\sigma}$ – координаты x, y σ -го КТЭ в ЛСК детали.

Координаты x_i, y_i оси шпинделя $i \in I$ в системе координат АС, повернутой вокруг оси Oz на угол β (шпиндель i выполняет обработку σ -го КТЭ детали на позиции $\alpha(i)$), определяются следующим образом:

$$x_i = (x_{лск} + \Delta_{ктэ, x}^{\sigma}) \cdot \cos(2\pi\alpha(i)/N_{поз} - \beta) - (y_{лск} + \Delta_{ктэ, y}^{\sigma}) \sin(2\pi\alpha(i)/N_{поз} - \beta);$$

$$y_i = (x_{лск} + \Delta_{ктэ, x}^{\sigma}) \sin(2\pi\alpha(i)/N_{поз} - \beta) + (y_{лск} + \Delta_{ктэ, y}^{\sigma}) \cos(2\pi\alpha(i)/N_{поз} - \beta).$$

Межшпиндельное расстояние $L(i, j, x_{лск}, y_{лск})$ для каждой пары шпинделей i, j должно превышать минимально допустимое для них значение $\delta(i, j)$:

$$L(i, j, x_{лск}, y_{лск}) \geq \delta(i, j), i, j \in I. \quad (1)$$

Значение $\delta(i, j), i, j \in I$, определяется по соответствующей таблице [11] и зависит от диаметров обрабатываемых отверстий и габаритов радиальных подшипников, которые, в свою очередь, зависят от режимов обработки.

В процессе определения габаритов $B=B(q, x_{лск}, y_{лск})$ и $H=H(q, x_{лск}, y_{лск})$ литья шпиндельной коробки вычисляются координаты плоскостей $X_{max}(q, x_{лск}, y_{лск}), X_{min}(q, x_{лск}, y_{лск}), Y_{max}(q, x_{лск}, y_{лск}), Y_{min}(q, x_{лск}, y_{лск})$ с использованием ограничений на размещение осей шпинделей относительно границ литья коробки:

$$X_{max}(q, x_{лск}, y_{лск}) - x_i \geq b_2; \quad (2)$$

$$x_i - X_{min}(q, x_{лск}, y_{лск}) \geq b_2; \quad (3)$$

$$Y_{max}(q, x_{лск}, y_{лск}) - y_i \geq b_2; \quad (4)$$

$$y_i - Y_{min}(q, x_{лск}, y_{лск}) \geq b_2; \quad (5)$$

а также ограничений

$$\begin{cases} X_{max}(q, x_{лск}, y_{лск}) - x_i \geq b_1, \\ x_i - X_{min}(q, x_{лск}, y_{лск}) \geq b_1 \end{cases} \quad \text{при} \quad \begin{cases} Y_{max}(q, x_{лск}, y_{лск}) - y_i \leq b_1, \\ y_i - Y_{min}(q, x_{лск}, y_{лск}) \leq b_1; \end{cases} \quad (6)$$

$$\begin{cases} Y_{max}(q, x_{лск}, y_{лск}) - y_i \geq b_1, \\ y_i - Y_{min}(q, x_{лск}, y_{лск}) \geq b_1 \end{cases} \quad \text{при} \quad \begin{cases} X_{max}(q, x_{лск}, y_{лск}) - x_i \leq b_1, \\ x_i - X_{min}(q, x_{лск}, y_{лск}) \leq b_1, \end{cases} \quad (7)$$

которые обеспечивают непопадание осей шпинделей в квадратные области в углах литья шпиндельной коробки, предназначенные для размещения направляющих втулок крепления штанг кондукторных плит.

Габаритные размеры H и B литья коробки должны быть не меньше чем вычисленные значения $X_{max} - X_{min}$ и $Y_{max} - Y_{min}$ соответственно:

$$B \geq X_{max}(q, x_{лск}, y_{лск}) - X_{min}(q, x_{лск}, y_{лск}); \quad (8)$$

$$H \geq Y_{\max}(q, x_{\text{лск}}, y_{\text{лск}}) - Y_{\min}(q, x_{\text{лск}}, y_{\text{лск}}). \quad (9)$$

При этом значения пар габаритных размеров H и B выбираются из заданного дискретного множества \mathfrak{R} :

$$(H, B) \in \mathfrak{R}. \quad (10)$$

Ограничения на размещение детали (зажимного приспособления) в пределах сектора загрузочной позиции выглядят следующим образом:

$$y_{\text{лск}} + \Delta_y^l - (x_{\text{лск}} + \Delta_x^l) \cdot \text{tg}(\pi / N_{\text{поз}}) \leq 0, \quad l=1, 2, \dots, N_{\text{конт}}; \quad (11)$$

$$y_{\text{лск}} + \Delta_y^l + (x_{\text{лск}} + \Delta_x^l) \cdot \text{tg}(\pi / N_{\text{поз}}) \leq 0, \quad l=1, 2, \dots, N_{\text{конт}}. \quad (12)$$

Кроме описанных ограничений, узлы и сборочные единицы компоновки АС (средняя станина, поворотный стол, боковые приставки со шпиндельными коробками и силовыми бабками на горизонтальных силовых столах) должны удовлетворять еще ряду ограничений, как аналогичных приведенным выше, так и задаваемых алгоритмически [11]. В дальнейшем для сокращения изложения описываемое этими ограничениями множество допустимых значений координат $x_{\text{лск}}, y_{\text{лск}}$ начала ЛСК детали на загрузочной позиции приспособления для фиксированного номера q грани вертикальной боковой приставки будем обозначать $G(q)$:

$$(x_{\text{лск}}, y_{\text{лск}}) \in G(q). \quad (13)$$

Номер q грани вертикальной боковой приставки выбирается из множества номеров свободных граней:

$$q \in \{q_1, q_2, \dots, q_0\}. \quad (14)$$

Масса $F(q, x_{\text{лск}}, y_{\text{лск}})$ АС представляет сумму масс средней станины $m_{\text{ср. ст.}}(x_{\text{лск}}, y_{\text{лск}})$, поворотного стола $m_{\text{пов.ст.}}(x_{\text{лск}}, y_{\text{лск}})$, горизонтальных боковых приставок $m_{\text{пр}}^k(x_{\text{лск}}, y_{\text{лск}})$, $k=1, 2, \dots, N_{\text{гор}}$, вертикальной приставки $m_{\text{пр}}^{\text{верт}}(x_{\text{лск}}, y_{\text{лск}})$ и вспомогательного оборудования. Поскольку состав вспомогательного оборудования не зависит от расположения детали на приспособлении, его масса в дальнейшем без ограничения общности может не учитываться.

Таким образом, рассматриваемая задача заключается в выборе координат $(x_{\text{лск}}, y_{\text{лск}})$ начала ЛСК обрабатываемой детали на загрузочной позиции поворотного стола с вертикальной осью вращения и номера q грани средней станины для размещения вертикальной приставки, удовлетворяющих ограничениям (1)–(14) и минимизирующих массу АС:

$$F(q, x_{\text{лск}}, y_{\text{лск}}) = \sum_{k=1}^{N_{\text{гор}}} m_{\text{пр}}^k(x_{\text{лск}}, y_{\text{лск}}) + m_{\text{пр}}^{\text{верт}}(x_{\text{лск}}, y_{\text{лск}}) + m_{\text{ср.ст.}}(x_{\text{лск}}, y_{\text{лск}}) + m_{\text{пов.ст.}}(x_{\text{лск}}, y_{\text{лск}}) \rightarrow \min. \quad (15)$$

Если система ограничений (1)–(14) несовместна, то полагается $F(q, x_{\text{лск}}, y_{\text{лск}}) = \infty$.

В разд. 3 описывается предложенный метод решения задачи (1)–(15).

3. Метод решения

Система ограничений задачи (1)–(15) на допустимое положение детали состоит из линейных ограничений на координаты $(x_{\text{лск}}, y_{\text{лск}})$ начала ЛСК детали в плоскости Oxy поверхности стола, квадратичных ограничений на межшпиндельные расстояния, а также ограничений, в том числе задаваемых алгоритмически, относящихся к выбору узлов горизонтальных боковых приставок и условиям совместимости приставок со средней станиной и между собой. Можно показать, что только ограничения на межшпиндельные расстояния порождают невыпуклую область допустимых положений детали, в результате чего задача минимизации даже унимодальной функции не может быть решена в общем случае методами локального поиска, в том числе градиентными, а требует применения глобального поиска. Для такой задачи наиболее подходящи-

ми методами решения традиционно являются эвристики и метаэвристики [15], а также методы зондирования области поиска типа ЛП-последовательностей (ЛП) и случайного поиска (СП) [16, 17]. Среди метаэвристик перспективными представляются метод «роя частиц» (PSO – Particle Swarm Optimization Algorithm), генетические алгоритмы, алгоритм имитации отжига, нейронные сети.

В работе для исследования был выбран PSO – эвристический метод численной оптимизации, не требующий специальных свойств оптимизируемой функции. Этот метод нашел применение в большом количестве областей [18]. В частности, он хорошо зарекомендовал себя при решении задач, аналогичных исследуемой в настоящей работе [19, 20]. PSO оптимизирует функцию, поддерживая популяцию возможных решений, называемых частицами, и перемещая эти частицы в области поиска решений согласно итеративно вычисляемой простой формуле. Перемещения частиц учитывают наилучшее найденное в этой области положение, которое изменяется от итерации к итерации при нахождении частицами более выгодных положений. Ниже приведен алгоритм решения рассматриваемой задачи, основанный на методе PSO.

Введем следующие обозначения:

m – число частиц роя в популяции;

$F(q^\gamma, x_{лск}^\gamma, y_{лск}^\gamma)$ – значение целевой функции для γ -й частицы, $\gamma=1, 2, \dots, m$, при размещении вертикальной приставки на грани q^γ , т;

$\bar{p}^\gamma = (\bar{x}^\gamma, \bar{y}^\gamma)$ – положение γ -й частицы, в котором целевая функция принимает наименьшее значение среди всех ее значений для γ -й частицы, рассмотренных к моменту текущей итерации алгоритма, $\gamma=1, 2, \dots, m$;

\tilde{q}^γ – номер грани, на которой достигается наименьшее значение целевой функции в точке \bar{p}^γ ;

$g = (\bar{x}^g, \bar{y}^g)$ – положение частицы роя g^* , в котором целевая функция принимает наименьшее значение среди всех ее значений для всех частиц $\gamma \in \{1, 2, \dots, m\}$, рассмотренных к моменту текущей итерации алгоритма;

q^g – номер грани, на которой достигается наименьшее значение целевой функции для точки g ;

$v^\gamma = (v_x^\gamma, v_y^\gamma)$ – вектор смещения γ -й частицы (ее скорость), $\gamma=1, 2, \dots, m$;

ω – коэффициент, учитывающий влияние на вектор смещения его значения на предыдущем шаге (инерция);

φ_p – коэффициент влияния лучшего положения (отдельно взятой) частицы γ на ее вектор смещения (не зависит от индекса γ);

φ_g – коэффициент влияния наилучшего найденного положения g среди всех частиц роя на вектор смещения каждой из них;

r_p, r_g – вспомогательные коэффициенты, рассматриваемые как равномерно распределенные на интервале $[0, 1]$ случайные величины;

$\bar{\eta}$ – предельное число итераций алгоритма;

T_{CPU} – процессорное время работы алгоритма, с;

\bar{T}_{CPU} – максимально допустимое время работы алгоритма, с;

$[x_{min}, x_{max}]$ – отрезок возможных значений $x_{лск}^\gamma$ для γ -й точки, $\gamma=1, 2, \dots, m$;

$[y_{min}, y_{max}]$ – отрезок возможных значений координаты $y_{лск}^\gamma$ для γ -й точки, $\gamma=1, 2, \dots, m$.

Прямое произведение отрезков $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$ должно включать все возможные начала ЛСК, для которых координаты $x^\gamma + \Delta_x^l, y^\gamma + \Delta_y^l$ всех точек контура детали содержатся внутри сектора загрузочной позиции поворотного стола, $l=1, 2, \dots, N_{конт}$.

Сокращение размера отрезков $[x_{min}, x_{max}]$, $[y_{min}, y_{max}]$, определяющих пространство поиска, позволяет повысить эффективность алгоритма решения задачи. Определение этих отрезков зависит от числа $N_{поз}$ позиций поворотного стола АС, координат Δ_x^l, Δ_y^l точек контура детали

в ЛСК детали, $l=1, 2, \dots, N_{\text{конт}}$, и расположения начала координат ЛСК. В частности, сократить искомые отрезки позволяет определение правильной плоской фигуры (например, круга или прямоугольника), вписанной в контур детали, и исключение таких точек каждого из отрезков (рассматриваемых как геометрическое место координат центра фигуры), в которых эта фигура не вписывается в сектор позиции при любых значениях другой координаты центра фигуры.

Алгоритм PSO:

1. Генерируются m случайных точек $p^{\gamma(0)} = (x^{\gamma(0)}, y^{\gamma(0)})$, $\gamma=1, \dots, m$, начала ЛСК детали, координаты x^γ, y^γ которых равномерно распределены на заданных отрезках $[x_{\min}, x_{\max}]$, $[y_{\min}, y_{\max}]$ соответственно.

2. За положение \bar{p}^γ частицы γ принимается его начальное значение:

$$\bar{p}^\gamma = (\bar{x}^\gamma, \bar{y}^\gamma) = (x^{\gamma(0)}, y^{\gamma(0)}), \quad \gamma = 1, \dots, m.$$

3. Определяется начальное значение $v^{\gamma(0)} = (v_x^{\gamma(0)}, v_y^{\gamma(0)})$ вектора скорости частицы γ посредством выбора случайных равномерно распределенных чисел из интервалов $[-(x_{\max} - x_{\min}); (x_{\max} - x_{\min})]$, $[-(y_{\max} - y_{\min}); (y_{\max} - y_{\min})]$, $\gamma=1, 2, \dots, m$ соответственно.

4. Если частица \bar{p}^γ допустима хотя бы для одного $q \in \{q_1, q_2, \dots, q_0\}$, то вычисляется значение целевой функции $F(\tilde{q}, \bar{p}^\gamma) = \min\{F(q, x^{\gamma(0)}, y^{\gamma(0)}) | q \in \{q_1, q_2, \dots, q_0\}\}$, иначе полагается $F(\tilde{q}, \bar{p}^\gamma) = \infty$, $\gamma = 1, \dots, m$.

5. Среди всех значений $F(\tilde{q}, \bar{p}^\gamma)$ находится наименьшее и определяются соответствующая точка $g = (\bar{x}^g, \bar{y}^g)$ и соответствующая ей грань q^g :

$$F(q^g, g) = \min\{F(\tilde{q}, \bar{p}^{\gamma(0)}) | \gamma = 1, 2, \dots, m\};$$

$$(q^g, g) = \arg\{F(q^g, g)\}.$$

6. Положить $\eta = 1$.

7. Для каждой частицы $\gamma=1, 2, \dots, m$:

7.1. Генерируются случайные числа r_p и r_g из интервала $[0, 1]$.

7.2. Вычисляется новая скорость $v^{\gamma(\eta)} = (v_x^{\gamma(\eta)}, v_y^{\gamma(\eta)})$ частицы γ по формулам

$$v_x^{\gamma(\eta)} = \omega \cdot v_x^{\gamma(\eta-1)} + \phi_p \cdot r_p \cdot (\bar{x}^\gamma - x^\gamma) + \phi_g \cdot r_g \cdot (\bar{x}^\gamma - x^\gamma);$$

$$v_y^{\gamma(\eta)} = \omega \cdot v_y^{\gamma(\eta-1)} + \phi_p \cdot r_p \cdot (\bar{y}^\gamma - y^\gamma) + \phi_g \cdot r_g \cdot (\bar{y}^\gamma - y^\gamma).$$

7.3. Определяется новое положение $p^{\gamma(\eta)} = (\bar{x}^{\gamma(\eta)}, \bar{y}^{\gamma(\eta)})$ частицы γ переносом ее из положения $p^{\gamma(\eta-1)}$ на итерации $\eta-1$ на вектор $v^{\gamma(\eta)}$:

$$x^{\gamma(\eta)} = x^{\gamma(\eta-1)} + v_x^{\gamma(\eta)}, \quad y^{\gamma(\eta)} = y^{\gamma(\eta-1)} + v_y^{\gamma(\eta)}.$$

7.4. Если точка $(x^{\gamma(\eta)}, y^{\gamma(\eta)})$ допустима хотя бы для одного $q \in \{q_1, q_2, \dots, q_0\}$, то вычисляется значение $F(q^{\gamma(\eta)}, x^{\gamma(\eta)}, y^{\gamma(\eta)}) = \min\{F(q, x^{\gamma(\eta)}, y^{\gamma(\eta)}) | q \in \{q_1, q_2, \dots, q_0\}\}$, иначе полагается $F(q^{\gamma(\eta)}, x^{\gamma(\eta)}, y^{\gamma(\eta)}) = \infty$.

7.5. Если $F(q^{\gamma(\eta)}, x^{\gamma(\eta)}, y^{\gamma(\eta)}) < F(\tilde{q}^\gamma, \bar{p}^\gamma)$, то полагается $\bar{p}^\gamma = (x^{\gamma(\eta)}, y^{\gamma(\eta)})$, $\tilde{q}^\gamma = q^{\gamma(\eta)}$ и $F(\tilde{q}^\gamma, \bar{p}^\gamma) = F(q^{\gamma(\eta)}, x^{\gamma(\eta)}, y^{\gamma(\eta)})$.

7.6. Если $F(\tilde{q}^\gamma, \bar{p}^\gamma) < F(q^g, g)$, то полагается $g = \bar{p}^\gamma$, $q^g = \tilde{q}^\gamma$ и $F(q^g, g) = F(\tilde{q}^\gamma, \bar{p}^\gamma)$.

8. Если выполняется любое из условий $\eta = \bar{\eta}$ либо $T_{CPU} \geq \bar{T}_{CPU}$, то переход к п. 9. Иначе полагается $\eta = \eta + 1$ и переход к п. 7.

9. Если $F(q^g, g) < \infty$, то (q^g, g) – решение задачи. В противном случае решение задачи не найдено.

Таким образом, итерации повторяются до тех пор, пока не выполнится хотя бы одно из условий остановки работы алгоритма: либо выполнится заданное количество итераций, либо истечет максимально допустимое время работы алгоритма.

Как показывает практика, работоспособность и эффективность алгоритма существенно зависят от значений его параметров m , $\bar{\eta}$, ω , φ_p , φ_g . Методике выбора этих параметров при решении тестовых задач посвящен следующий раздел.

4. Численный эксперимент

Приведенный в разд. 3 алгоритм PSO реализован на языке C++. Эксперимент по решению описанных ниже двух тестовых задач рассматриваемого типа проводился на персональном компьютере с 64-разрядной операционной системой Windows 7 на базе процессора Intel Core i3-2120 с частотой 3,3 ГГц и оперативной памятью 8 ГБ.

С целью лучшей адаптации алгоритма для решения конкретного класса задач осуществляется подбор его параметров. Наиболее значимыми из них являются число m частиц; предельное число $\bar{\eta}$ итераций; параметр ω , учитывающий инерцию; φ_p – влияние лучшего положения частицы на поиск ее следующего положения; φ_g – влияние известного на каждом конкретном шаге лучшего положения частицы среди всех частиц.

Обозначим:

F^* – наименьшее значение целевой функции, полученное в результате работы алгоритма PSO;

g^* – точка, для которой $F(g^*) = F^*$.

Подбор значений параметров PSO с целью повышения его эффективности зависит от решаемой задачи [15]. Наиболее часто в литературе встречаются положительные значения перечисленных параметров алгоритма. Коэффициент ω , как правило, варьировался в пределах от 0 до 1, а коэффициенты φ_p и φ_g – от 0 до 4. Исходя из этого, были определены следующие множества для поиска наилучших значений параметров алгоритма:

$$m = \{5, 10, 15, 20\}, \bar{\eta} = \{10, 15, 20, 25, 30\}, \omega \in [0,1], \varphi_p \in [0,4], \varphi_g \in [0,4].$$

Работоспособность алгоритма PSO проверялась на двух тестовых примерах.

Пример 1. На приспособлении АС с шестипозиционным поворотным столом требуется разместить для обработки крупногабаритную деталь (рис. 3, а) с размерами H_{dem} (высота) = 550 мм, L_{dem} (длина) = 400 мм и B_{dem} (ширина) = 350 мм. Обработка ведется на четырех позициях: фрезерование двух верхних поверхностей детали осуществляется на второй и четвертой позициях фрезерными бабками, установленными на горизонтальных силовых столах и занимающих соответствующие грани центральной станины; сверление и зенкерование одного отверстия, расположенного на верхней грани детали, осуществляются на пятой и шестой позициях шпиндельной коробкой, установленной на вертикальном силовом столе боковой приставки, которая может быть размещена на третьей, пятой или шестой гранях.

Пример 2. Малогабаритная деталь (рис. 3, б) с размерами $H_{dem} = 75$ мм, $L_{dem} = 128$ мм и $B_{dem} = 128$ мм предназначается для обработки на этом же АС с шестипозиционным поворотным столом. Обработка ведется на четырех позициях: на второй и пятой позициях сверление и растачивание отверстия диаметром 30 мм, расположенного на боковой грани детали, осуществляются горизонтальными силовыми столами, размещенными на соответствующих гранях (второй и пятой) центральной станины; на третьей, четвертой и пятой позициях сверление, зенкерование и развертка четырех отверстий верхней грани детали выполняются шпиндельной коробкой, установленной на вертикальном силовом столе приставки, которая может быть размещена на третьей, четвертой или шестой гранях центральной станины.

На рис. 3, в изображены обе детали в положении «в плане» на поверхности поворотного стола и возможные диаметры его планшайбы – 900, 1120, 1250, 1400, 1600, 2000 мм соответственно.

В настоящей работе была выбрана следующая схема подбора параметров алгоритма PSO. Параметру ω присваивалось среднее значение 0,5 из выбранного интервала. Затем подбирались значения параметров φ_p и φ_g таким образом, чтобы получить наименьшее значение целевой функции за наименьшее процессорное время. На рис. 4 представлена зависимость значения F^* от параметров φ_p и φ_g при $\omega=0,5$, $m=10$, $\bar{\eta}=30$ для примеров 1 и 2 соответственно.

На рис. 5 показана зависимость значения F^* от числа m генерируемых частиц и предельного количества $\bar{\eta}$ итераций при фиксированных значениях параметров $\varphi_p=1,5$, $\varphi_g=3,5$ (для примера 1) и $\varphi_p=4$, $\varphi_g=0,5$ (для примера 2), полученных как наилучшие в предыдущем эксперименте. При этом $F^*=20,8697$ для примера 1 и $F^*=12,5704$ для примера 2 были получены при различных значениях параметров φ_p и φ_g , но были выбраны такие значения этих параметров, для которых значение F^* достигалось за наименьшее процессорное время (3,3228 с).

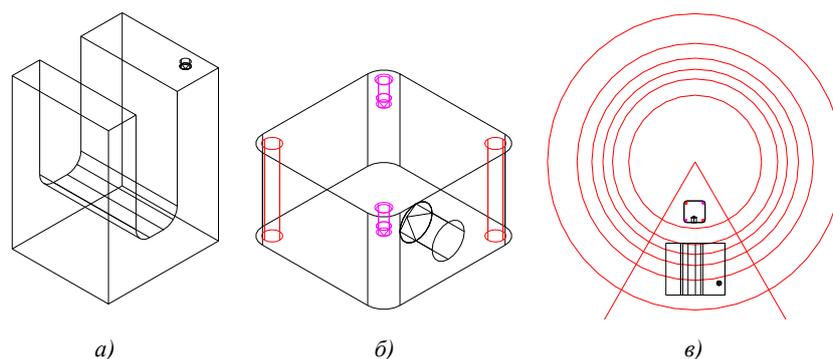


Рис. 3. Примеры обработки на АС с поворотным столом двух деталей различных габаритов

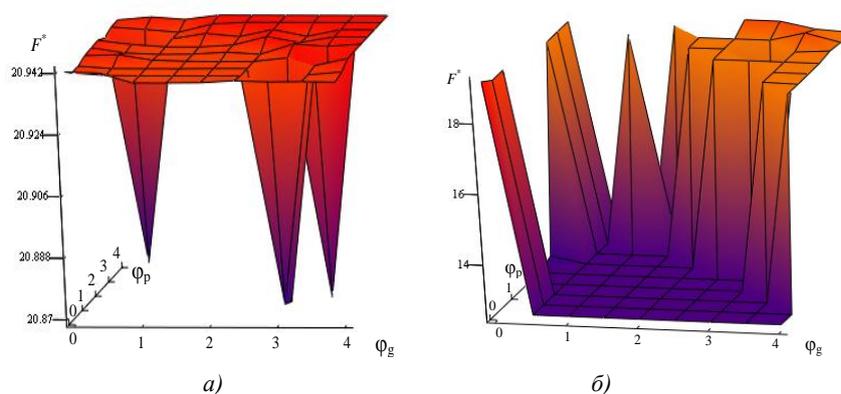


Рис. 4. Зависимость значения F^* от параметров φ_p и φ_g : а) для примера 1; б) для примера 2

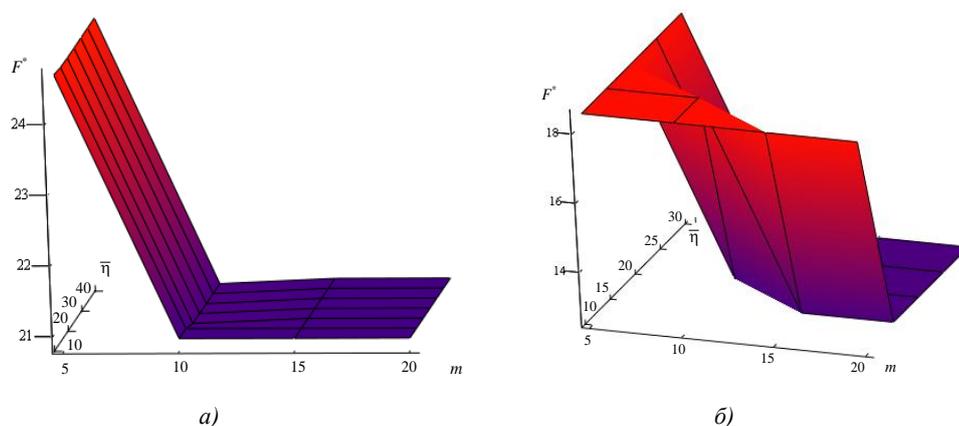


Рис. 5. Зависимость значения F^* от параметров m и $\bar{\eta}$: а) для примера 1; б) для примера 2

Поскольку в каждом из примеров наименьшее значение целевой функции достигалось при различных значениях параметров m и $\bar{\eta}$, были выбраны значения, которым соответствовало минимальное значение T_{CPU} . Для примера 1 $F^*=20,8697$ было получено при $m=10$, $\bar{\eta}=40$. При дальнейшем увеличении $\bar{\eta}$ до 400 значение F^* не улучшалось (рис. 6, а). Для примера 2 минимальное значение функции $F^*=12,5704$ было получено при $m=10$, $\bar{\eta}=30$. При дальнейшем увеличении $\bar{\eta}$ до 400 значение F^* не улучшалось (рис. 6, б).

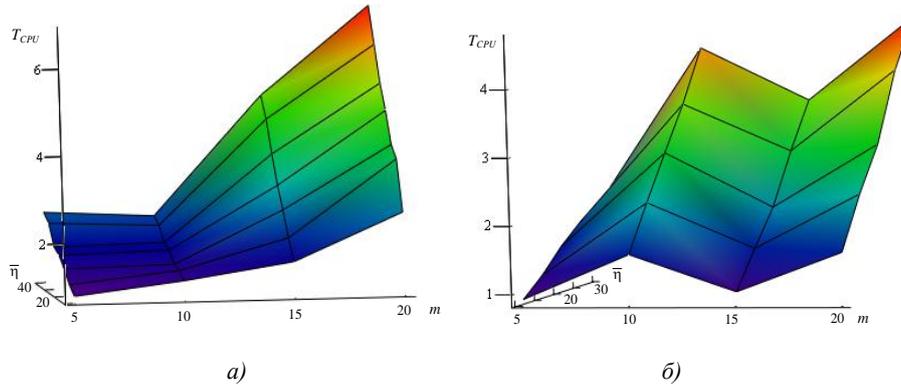


Рис. 6. Зависимость значения T_{CPU} от параметров m и $\bar{\eta}$: а) для примера 1; б) для примера 2

Таким образом, по результатам экспериментов были выбраны следующие значения параметров алгоритма:

- для примера 1: $m=10$, $\bar{\eta}=40$, $\omega=0,5$, $\varphi_p=1,5$, $\varphi_g=3,5$;
- для примера 2: $m=10$, $\bar{\eta}=30$, $\omega=0,5$, $\varphi_p=4$, $\varphi_g=0,5$.

При этих значениях параметров $F^*=20,8693$ при $T_{CPU}=2,1996$ для примера 1 и $F^*=12,5704$ при $T_{CPU}=4,4304$ для примера 2.

С целью проверки целесообразности использования PSO для решения подобных задач было проведено его сравнение с известными методами СП и ЛП. Поскольку термин «итерация» для PSO и этих методов имеет различный смысл, было введено понятие «шаг» – порядковый номер точки, в которой определяется значение функции F . Для методов СП и ЛП число шагов совпадает с количеством итераций, а для PSO число шагов равно произведению числа точек роя и числа итераций. Поскольку для примера 1 с помощью PSO решение было найдено при $m=10$, $\bar{\eta}=40$ (т. е. за 400 шагов), то для СП и ЛП в эксперименте число шагов изменялось в интервале от 10 до 400 (рис. 7, а и табл. 2). Для примера 2 выбран интервал от 0 до 300, поскольку параметры PSO были $m=10$, $\bar{\eta}=30$ (рис. 7, б и табл. 3).

В табл. 2 и 3 N – число шагов сравниваемых алгоритмов и N_F – число шагов, на которых ограничения (1)–(14) совместны.

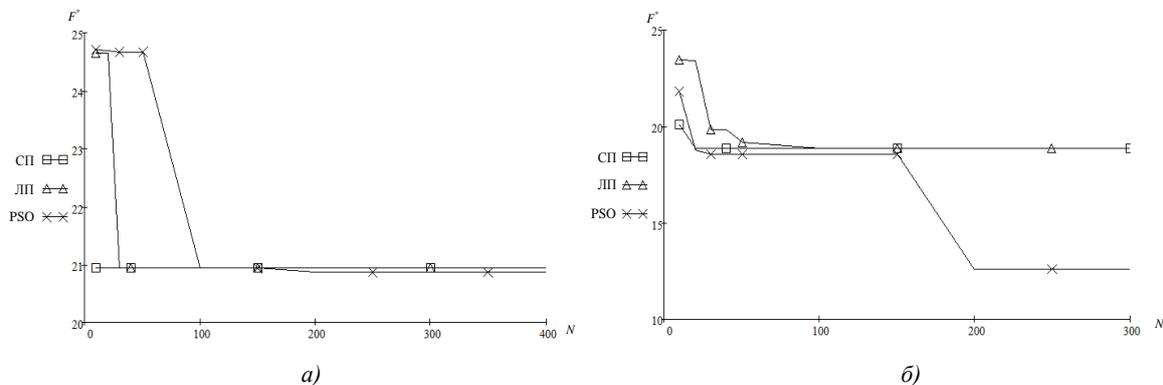


Рис. 7. Зависимость значения F^* от числа N шагов для методов СП, ЛП и алгоритма PSO: а) для примера 1; б) для примера 2

Таблица 2

Результаты решения для примера 1

N	СП			ЛП			PSO		
	F^* , т	N_F	T_{CPU} , с	F^* , т	N_F	T_{CPU} , с	F^* , т	N_F	T_{CPU} , с
10	20,9424	4	0,1404	24,646	5	0,1872	24,7058	4	0,2496
20	20,9424	12	0,3744	24,646	10	0,3588	24,6789	7	0,3588
30	20,9424	18	0,5304	20,9458	15	0,5616	24,6762	10	0,5148
40	20,9424	22	0,7020	20,9458	20	0,8112	24,6762	13	0,7020
50	20,9424	28	0,8424	20,9458	25	0,9360	24,6748	16	0,7800
100	20,9424	53	1,6536	20,9431	54	2,0280	20,9418	21	1,0764
150	20,9424	81	2,5584	20,9431	78	2,8236	20,9418	24	1,2324
200	20,9424	110	3,6660	20,9431	107	4,0404	20,8723	27	1,2948
250	20,9424	133	4,1028	20,9431	132	4,9140	20,8717	29	1,5444
300	20,9424	163	5,3352	20,9431	158	5,8188	20,8697	34	1,6380
350	20,9424	191	6,2088	20,9396	187	6,8952	20,8695	39	1,9500
400	20,9424	213	7,0980	20,9396	209	7,8000	20,8693	44	2,0904
4000	20,8705	2177	70,7152	20,8716	2155	80,6993	20,8693	248	7,1448

Таблица 3

Результаты решения для примера 2

N	СП			ЛП			PSO		
	F^* , т	N_F	T_{CPU} , с	F^* , т	N_F	T_{CPU} , с	F^* , т	N_F	T_{CPU} , с
10	20,0757	4	0,1248	23,4435	5	0,2184	21,8035	4	0,1404
20	18,8904	12	0,3744	23,3778	10	0,3588	18,7664	6	0,2496
30	18,8904	18	0,5304	19,7998	15	0,5304	18,5285	9	0,4212
40	18,8904	22	0,6708	19,7998	20	0,6396	18,5285	12	0,5772
50	18,8904	28	0,8736	19,1396	25	0,8112	18,5285	14	0,9204
100	18,8904	53	1,4976	18,8907	54	2,0592	18,5312	29	1,5756
150	18,8904	80	2,3868	18,8907	77	2,7924	18,5297	38	2,2464
200	18,8863	108	3,2604	18,8899	106	3,8376	12,5728	46	2,8860
250	18,8863	131	3,9156	18,8899	131	4,8360	12,5706	53	3,5100
300	18,8863	161	4,9452	18,8899	156	5,5692	12,5704	62	4,2432
3000	12,5707	1601	73,2424	12,5732	1594	75,6916	12,5704	398	17,6437

На рис. 8 изображены графики зависимости T_{CPU} от количества N шагов для сравниваемых методов. На решение задач с помощью PSO было потрачено меньше процессорного времени, так как число удовлетворяющих ограничениям точек было существенно меньше, чем в алгоритмах СП и ЛП (см. табл. 2 и 3).

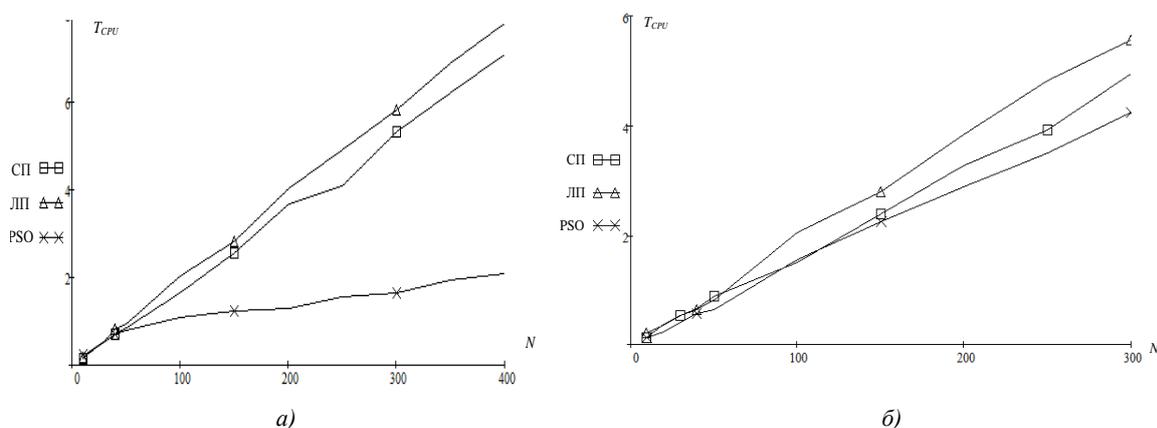


Рис. 8. Зависимость времени T_{CPU} , за которое найдено F^* , от числа N шагов для СП, ЛП и PSO: а) для примера 1; б) для примера 2

По результатам экспериментов можно сделать следующие выводы:

1. Для примеров 1 и 2 с помощью PSO были получены решения с меньшими значениями целевой функции за меньшее процессорное время по сравнению с решениями, полученными методами СП и ЛП. При этом для примера 1 больший выигрыш был по процессорному времени, а для примера 2 – по значению целевой функции.

2. Для того чтобы получить значение F^* , близкое к найденному с помощью PSO, для примера 1 было выполнено по 4000 шагов в методах СП и ЛП. При этом для получения $F^*=20,8705$ методом СП было потрачено в 32 раза больше времени ($T_{CPU} = 70,7152$), а для получения $F^*=20,8716$ методом ЛП – в 36 раз ($T_{CPU} = 80,6993$).

С помощью всех трех методов в качестве оптимальной грани для установки вертикального силового стола была выбрана пятая грань центральной станины.

3. Для того чтобы получить значение F^* , близкое к найденному с помощью PSO, для примера 2 было выполнено по 3000 шагов в методах СП и ЛП. При этом для получения $F^*=12,5707$ методом СП было потрачено более чем в 16 раз больше времени ($T_{CPU} = 73,2424$), а для получения $F^*=12,5732$ методом ЛП – более чем в 17 раз ($T_{CPU} = 75,6916$).

С помощью всех трех методов в качестве оптимальной грани для установки вертикального силового стола была выбрана четвертая грань центральной станины.

4. При решении оптимизационных задач зачастую используется критерий останова алгоритма по заданному числу последовательных итераций с неизменным значением целевой функции. Проведенный эксперимент с алгоритмом PSO для рассматриваемой задачи показал неприменимость этого критерия останова из-за относительно небольшого числа итераций, необходимых для нахождения решения задачи. Благодаря наличию случайных величин в формуле вычисления вектора скорости минимальное значение целевой функции может обновиться (причем существенно) после 10–20 итераций с неизменным ее значением. Таким образом, эти случайные величины играют важную роль, позволяя выводить алгоритм из локального экстремума и находить лучшее решение.

Заключение

В работе исследована задача оптимизации массы АС с поворотным столом и параллельной многопозиционной многшпиндельной обработкой однородной партии деталей вертикальной приставкой за счет выбора грани расположения этой приставки и размещения детали в секторе загрузочной позиции. Выбор грани для размещения приставки осуществляется полным перебором граней, не занятых горизонтальными приставками. Для оптимизации размещения детали предложен эвристический алгоритм «роя частиц» и подобраны наилучшие параметры этого алгоритма для двух тестовых примеров. Проведено экспериментальное сравнение предложенного метода с известными методами случайного поиска и ЛП-последовательностей, показавшее его превосходство как по значению целевой функции, так и затраченному процессорному времени.

В качестве направления дальнейших исследований предполагается рассмотреть задачу оптимизации размещения на приспособлении АС (с приспособлением на поворотном столе либо на передвижном столе) группы различных деталей для групповых процессов серийной обработки. Эта задача является существенно более сложной, чем рассмотренная в работе, поскольку размещаемые детали могут быть пространственно взаимосвязаны, при этом улучшение положения одной детали не всегда приводит к улучшению положения для других деталей группы.

Авторы выражают искреннюю признательность рецензенту за сделанные замечания, позволившие существенно улучшить описание полученных результатов.

Список литературы

1. Гебель, Х. Компоновка агрегатных станков и автоматических линий / Х. Гебель. – М. : Машгиз, 1959. – 189 с.

2. Горанский, Г.К. Алгоритмизация проектирования типовых компоновок агрегатных станков / Г.К. Горанский, Г.М. Левин, Г.И. Цфасман // Вычислительная техника в машиностроении. – 1967. – Июль. – С. 109–121.
3. Горанский, Г.К. Алгоритмизация проектирования компоновочной схемы агрегатного станка с возвратно-поступательным перемещением обрабатываемой детали / Г.К. Горанский, Г.М. Левин, Г.И. Цфасман // Вычислительная техника в машиностроении. – 1967. – Октябрь. – С. 35–41.
4. Хомяков, В.С. Автоматизированное проектирование компоновок металлообрабатывающих станков / В.С. Хомяков, И.И. Давыдов // Станки и инструмент. – 1990. – № 5. – С. 4–7.
5. Клепиков, С.И. Автоматизация проектирования компоновок металлорежущих станков / С.И. Клепиков, А.Г. Ивахненко, Д.В. Картелев // Станки и инструмент. – 2002. – № 7. – С. 3–4.
6. Врагов, Ю.Д. Анализ компоновок металлорежущих станков: основы компонетики / Ю.Д. Врагов. – М. : Машиностроение, 1978. – 208 с.
7. Аверьянов, О.И. Автоматизированное проектирование компоновок многооперационных станков / О.И. Аверьянов, А.Л. Воронов, Я.М. Гельштейн // Станки и инструмент. – 1982. – № 8. – С. 6–7.
8. Аверьянов, О.И. Модульный принцип построения станков с ЧПУ / О.И. Аверьянов. – М. : Машиностроение, 1987. – 232 с.
9. Халдей, М.Б. Автоматизация синтеза и оценки компоновок станков : автореф. дис. ... канд. техн. наук : 05.03.01 / М.Б. Халдей ; Моск. гос. технол. ун-т. – М., 1996. – 21 с.
10. Хомяков, В.С. Информационная система синтеза компоновок станков / В.С. Хомяков, М.Б. Халдей // Конструкторско-технологическая информатика (КТИ-96) : труды Международного конгресса. – М., 1996. – С. 150–151.
11. Комплекс алгоритмов и программных средств формирования компоновок агрегатных металлорежущих станков / Г.М. Левин [и др.]. – Минск, 2005. – 50 с. – (Препринт / Объединенный институт проблем информатики НАН Беларуси; № 4).
12. Автоматизация проектирования агрегатных металлорежущих станков : учеб. пособие / Г.М. Левин [и др.]. – Витебск : ВГТУ, 2008. – 122 с.
13. MIP Approach to Design of Rotary Transfer Machines with Turrets / O. Battaia [et al.]. – Saint Etienne, 2013. – 35 p.
14. Розин, Б.М. К синтезу компоновок агрегатных станков для групповой обработки / Б.М. Розин, В.Е. Зданович // Информатика. – 2014. – № 4(44). – С. 100–116.
15. Скобцов, Ю.А. Метаэвристики / Ю.А. Скобцов, Е.Е. Федоров – Донецк : Ноулидж, 2013. – 426 с.
16. Пантелеев, А.В. Методы оптимизации в примерах и задачах : учеб. пособие / А.В. Пантелеев, Т.А. Летова. – 2-е изд. – М. : Высш. шк., 2005. – 544 с.
17. Соболев, И.М. Выбор оптимальных параметров в задачах со многими критериями / И.М. Соболев, Р.Б. Статников. – М. : Дрофа, 2006. – 175 с.
18. Poli, R. Analysis of the Publications on the Applications of Particle Swarm Optimisation / R. Poli // Journal of Artificial Evolution and Applications. – 2008. – Vol. 2008. – P. 1–10.
19. Shahrajabian, H. Multi-constrained optimization in ball-end machining of carbon fiber-reinforced epoxy composites by PSO / H. Shahrajabian, M. Farahnakian // Cogent Engineering. – 2015. – Vol. 2, iss. 1. – 14 p.
20. Wei, L. A particle swarm optimization approach to a multi-objective reconfigurable machine tool design problem / L. Wei, L. Ming // Proceedings of the 2006 IEEE Congress on Evolutionary Computation. – Vancouver, BC, Canada, 2006. – P. 2222–2229.

Поступила 19.10.2015

*Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: gyshin@newman.bas-net.by
victoria.zdanovich@gmail.com
rozin@newman.bas-net.by*

N.N. Guschinski, V.E. Zdanovich, B.M. Rozin

**OPTIMIZING THE PLACEMENT OF A WORK-PIECE
AT A MULTI-POSITION ROTARY TABLE OF TRANSFER MACHINE
WITH VERTICAL MULTI-SPINDLE HEAD**

The problem of minimizing the weight of transfer machine with a multi-position rotary table by placing of a work-piece at the table for processing of homogeneous batch of work-pieces is considered. To solve this problem the mathematical model and heuristic particle swarm optimization algorithm are proposed. The results of numerical experiments for two real problems of this type are given. The experiments revealed that the particle swarm optimization algorithm is more effective for the solution of the problem compared to the methods of random search and LP-search.

ОБРАБОТКА СИГНАЛОВ, ИЗОБРАЖЕНИЙ И РЕЧИ

УДК: 303.732.4

Е.В. Лисица¹, Н.Н. Яцков¹, В.В. Апанасович¹, Т.В. Апанасович²

ПРОГРАММНЫЙ ПАКЕТ CellDataMiner ДЛЯ АНАЛИЗА ЛЮМИНЕСЦЕНТНЫХ ИЗОБРАЖЕНИЙ РАКОВЫХ КЛЕТОК

Предлагается программный пакет CellDataMiner для анализа люминесцентных изображений раковых клеток. Проводится сравнительный анализ алгоритмов классификации и кластеризации данных с целью реализации в пакете наиболее эффективных из них. Работоспособность программного обеспечения проверяется на экспериментальных данных, представляющих результаты по исследованию опухоли молочной железы.

Введение

Метод люминесцентной микроскопии успешно используется для исследования срезов анатомических тканей в биомедицине [1]. В данном методе регистрируется интенсивность биомаркера объекта исследования (например, маркера онкологического заболевания) в ядрах и цитоплазмах клеток. Биомаркер – это специальное химическое вещество, отражающее протекающие в клетке процессы. В случае если биомаркер не обладает люминесцентными свойствами, то используются специальные красители, которые позволяют идентифицировать наличие биомаркера в образце. Поскольку в клетках одновременно протекает несколько процессов, для их анализа удобно использовать несколько маркеров, отражающих эти процессы. Все это привело к широкому внедрению на практике многоканальных экспериментов. Как правило, в многоканальных экспериментах используются отдельные красители для цитоплазм и ядер, информация о которых регистрируется в отдельном канале изображения. Результатом исследования является цветное RGB-изображение [2]. В большинстве случаев за один эксперимент получается около сотни изображений, которые содержат порядка тысячи объектов. Специалисты могут производить анализ изображений вручную, классифицировать объекты на них, считать их количество. Однако при обработке больших объемов данных такой подход становится неэффективным и приводит как к временным затратам, так и к ошибкам при проведении анализа.

На сегодняшний день существует большое количество программных средств, как коммерческих, так и находящихся в свободном доступе, которые предназначены для обработки и анализа изображений. Среди них, например, можно выделить проект по разработке открытого программного обеспечения (ПО) и форматов данных для обработки и хранения изображений, получаемых оптическим микроскопом в ходе анализа биологических объектов OME (Open Microscopy Environment). В рамках данного проекта реализованы программные пакеты Bio-Formats для чтения и записи данных в виде изображений с помощью стандартизированных открытых форматов и VisBio для визуализации и анализа многомерных данных [3]. Существуют и другие программные продукты для анализа многомерных данных, среди которых можно отметить пакет BioImageXD [4]. На рынке ПО имеется ряд средств для управления процессом получения изображений, одним из них является MicroManager [5]. Другое направление разработки ПО ориентировано на 4D- и 5D-визуализацию данных, получаемых при помощи оптических микроскопов, например FARSIGHT [6].

Одним из широко используемых программных пакетов для обработки изображений биологических объектов является ImageJ [7] и его расширение Fiji [8, 9]. Данное ПО представляет собой программу с открытым программным кодом, написанным на языке Java. Расширение функций осуществляется за счет подключения плагинов, реализующих алгоритмы сегментации и анализа; для устранения повторяющихся действий при анализе предусмотрен специальный язык макросов [10]. Разработаны специальные алгоритмы по обработке гистологических сним-

ков, получаемых с оптического микроскопа [11, 12], однако их применение к люминесцентным изображениям затруднительно. Эффективным программным средством для анализа люминесцентных изображений клеток является группа пакетов CellProfiler и CellProfiler Analyst [13, 14]. Пакет CellProfiler представляет собой набор модулей для сегментации и оценки характеристик сегментированных объектов. Он включает известные методы обработки, такие как пороговые, градиентные, и метод водораздела [15], где пользователю предлагается настраивать параметры методов вручную. CellProfiler Analyst – программный пакет, ориентированный на анализ многомерных данных. В нем интегрированы базовые методы классификации и кластерного анализа [14]. Однако использование стандартных методов, реализованных в программных пакетах, затруднено при сегментации ядер на трехканальных люминесцентных изображениях. Так, например, недостатком пороговых алгоритмов является низкое качество сегментации, характерное для обработки больших объемов данных. Группа градиентных методов имеет ограничения, связанные с чувствительностью к наличию артефактов и к перепадам интенсивности на изображении. Перекрытие цветовых каналов при регистрации изображений приводит к наличию большого количества артефактов на снимках. Широкое распространение для анализа биоданных получила библиотека C++ OpenCV, объединяющая набор алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов [11]. Следует также отметить работы, посвященные разработке алгоритмов анализа данных об экспрессии генов, которые были получены с помощью изображений биоматриц ДНК [16] и гистологических изображений [17].

К основным ограничениям существующего ПО для анализа люминесцентных трехканальных биоизображений можно отнести:

- отсутствие автоматических алгоритмов сегментации;
- невысокую точность локализации биообъектов изображений, связанную с использованием традиционных методов, не адаптированных для анализа многоканальных изображений;
- отсутствие учета статистической взаимосвязи между каналами регистрации многоканальных изображений;
- трудности обработки больших объемов данных;
- анализ изображений без учета характеристик биообъектов (ядра, клетки, мембраны) на пиксельном уровне изображения, что значительно ограничивает точность классификации данных;
- требование достаточно глубокого знания пользователем не только применяемых методов анализа данных, но и определенных навыков программирования для более точной настройки параметров алгоритмов или осуществления взаимодействия между алгоритмами;
- ограниченное представление методов классификации сегментированных объектов изображений.

Возможными методами улучшения автоматической обработки трехканальных биомедицинских изображений являются применение объектно-ориентированной методологии анализа изображений на пиксельном уровне, учет статистической взаимосвязи между различными каналами изображений и последующее использование методов анализа данных для выявления скрытых закономерностей в оценке параметров сегментированных объектов изображений, не поддающихся обнаружению традиционными методами или экспертным путем [18, 19].

В данной работе представлен программный пакет для локализации, сегментации и классификации ядер клеток трехканальных люминесцентных изображений, интегрирующий наиболее эффективные алгоритмы сегментации, объектно-ориентированный анализ биообъектов на пиксельном уровне и методы анализа данных. Для всестороннего исследования разработанного программного пакета используется набор экспериментальных данных, полученных в ходе анализа тканей опухоли молочной железы [20, 21].

1. Экспериментальные данные

В настоящем исследовании рассматриваются микрочипы срезов тканей опухолей молочной железы. Изображения представляют собой популяции клеток, окрашенные в зеленые, синие и красные цвета (трехканальные люминесцентные сигналы в системе RGB). В цитоплазмах раковых клеток регистрируются процессы с участием белка цитокератина. Белок маркируется циа-

ниновым красителем Су3 и регистрируется в зеленом цветовом канале изображения. Красный канал изображения зарезервирован для индикации ядер раковых клеток. В ядрах раковых клеток находится белок эстроген-рецептор, для маркировки которого применяется краситель [20]. Для маркировки ядер используется краситель 4,6-диамидино-2-фенилиндол дигидрохлорид (DAPI, 4',6'-diamidino-2-phenylindole) и зарезервирован синий канал. Размер изображений – 2048×2048 пикселей в каждом из трех каналов, разрешающая способность – 0,2 мкм на пиксел [20–22].

2. Программный пакет CellDataMiner

Разработан и реализован программный пакет CellDataMiner, интегрирующий объектно-ориентированную методологию анализа биообъектов люминесцентных изображений на пиксельном уровне с последующим использованием методов анализа данных для выделения групп раковых клеток или исследования стадий развития онкологического заболевания. На данном этапе оптимальной средой разработки является Matlab. Пакет содержит проверенные и опубликованные библиотеки математических алгоритмов, включая алгоритмы цифровой обработки данных и имитационного моделирования, а также предоставляет возможности для создания графических интерактивных интерфейсов приложений и некоммерческой установки разрабатываемых пакетов.

Созданное ПО CellDataMiner реализует следующий набор принципиально новых функций, обеспечение которых не поддерживается или ограничено в других программных продуктах, находящихся в открытом доступе [23]:

- полностью автоматическую сегментацию ядер на изображениях биологических объектов;
- попиксельный анализ сегментированных объектов (результаты анализа отображаются в виде дополнительной таблицы);
- анализ распределений характеристик сегментированных объектов на пиксельном и интегральном уровнях;
- снижение размерности данных, т. е. выделение основных групп признаков сегментированных объектов для последующей интерпретации и визуализации данных;
- классификацию и кластеризацию раковых клеток;
- табличную и графическую визуализацию промежуточных и итоговых результатов анализа;
- интерактивное взаимодействие пользователя с объектами изображения.

Каждый из функциональных блоков ПО является независимым, что позволяет пользователю пропустить часть из них с целью оптимального выбора схемы анализа данных. В разработанном пакете наборы люминесцентных изображений могут анализироваться последовательно автоматически, без участия оператора, что позволяет значительно упростить анализ данных и вместе с тем сократить затраты на проведение исследований.

2.1. Алгоритм сегментации и методы анализа данных

В основе алгоритма сегментации лежит использование корреляционной зависимости между сигналами флуоресценции в R-, G-, B-каналах изображения. Представленный алгоритм реализуется поэтапно. На первом этапе происходит сегментация маски опухоли и оценка размеров ядер, на втором – сегментация ядер [24]. Сегментация ядер, в свою очередь, является многоэтапным процессом. В результате использования усовершенствованного алгоритма адаптивной сегментации полученное после бинаризации изображение содержит слившиеся объекты, которые разделяются водораздельным фильтром. Для устранения эффекта чрезмерной сегментации водораздельным методом, где полутоновое изображение линий перепадов и водоразделов предварительно сглаживается, применяется медианный фильтр.

В результате выполнения процедур сегментации изображения колоний раковых клеток и квантификации оценок статистических характеристик параметров ядер клеток получены N объектов-ядер n_1, n_2, \dots, n_N , характеризующихся набором из K признаков (измеряемых оценок параметров объектов) X_1, X_2, \dots, X_K .

Выходная характеристика, или зависимая переменная Y , представляет собой экспертную оценку состояния клеток. Например, для объекта n_i $y_i = 1$, если эксперт на основе визуальной оценки считает, что клетка раковая; $y_i = -1$ для нераковой клетки; $y = 0$ для объектов, подверженных значительным экспериментальным искажениям (рис. 1).

В ходе дальнейшего анализа требуется распределить объекты в группы раковых (больных) и нераковых (здоровых) по степени подобия, а также выявить факторы, обуславливающие различия статистических характеристик объектов, для решения задач определения фенотипа клеток, отделения шума от полезной информации, классификации стадий рака и т. д. В общем случае в ходе анализа решаются задачи снижения размерности признаков объектов, кластеризации, классификации и визуализации.

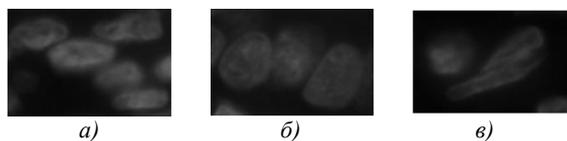


Рис. 1. Примеры ядер: а) здоровые; б) больные; в) искаженные

В качестве характеристик объектов изображений выбраны десять наиболее значимых признаков [25]:

- координаты центра масс (XM, YM);
- угол между осью абсцисс и основной осью характеристического эллипса (Angle);
- эксцентриситет характеристического эллипса (Eccentricity);
- площадь исследуемого объекта совместно с площадью пустот, входящих в него (FilledArea);
- плотность как отношение площади объекта к его выпуклой площади (Solidity);
- средняя интенсивность в красном канале (MEAN_RED);
- средняя интенсивность в синем канале (MEAN_BLUE);
- среднеквадратичное отклонение в красном канале (STD_RED);
- медиана красного канала (MEDIAN_RED);
- среднее значение в зеленом канале (MEAN_GREEN).

Координаты центра масс и угол между осью абсцисс и основной осью характеристического эллипса не являются признаками объектов как таковыми, а описывают их местоположение и ориентацию на изображении. Однако данные параметры характеризуют свойство раковых клеток формировать колонии и различные пространственные ориентации относительно нераковых клеток. Использование координат центра масс в качестве признака позволяет усилить объединение в кластеры близкорасположенных объектов, а учет угла формирует кластеры одинаково ориентированных объектов.

Для оценки качества классификации и кластеризации использовалась ошибка неверно классифицированных объектов ERROR:

$$\text{ERROR} = 100 \% * (\text{NUMBER}_1 + \text{NUMBER}_2) / \text{TOTAL},$$

где NUMBER_1 – количество здоровых объектов, отнесенных к больным; NUMBER_2 – количество раковых объектов, отнесенных к здоровым; TOTAL – размер обучающей выборки.

Сравнение методов анализа проведено на экспериментальных изображениях.

На сегодняшний день не существует оптимального метода для решения задач классификации и кластеризации. В работе [19] проводится сравнение методов дискриминантного анализа, метода на основе опорных векторов и предлагается подход к усилению простых классификаторов путем их комбинирования в различные группы для исследования клеток линий HeLa и HT29. Наилучшие результаты классификации получены для метода опорных векторов (MOB). В [26] для классификации четырех линий раковых клеток Ovar3, MiaPaCa2, MCF7 и MCF7-p53 используются комбинации искусственных нейронных сетей, метод k -ближайших соседей и метод главных компонент.

2.2. Кластеризация и классификация данных

Для кластеризации данных реализованы стандартные методы, такие как иерархический кластерный анализ, метод k -средних и сети Кохонена.

Для решения задачи кластерного анализа с помощью иерархических методов необходимо задать меру сходства, способ кластеризации и число кластеров. Для сравнения двух объектов n_i и n_j используются расстояния: евклидово (d_{euc}), города (d_{city}), Минковского (d_{Mink}), Махалоноби-

са (d_{mah}), косинусное (d_{cos}) и корреляционное (d_{cor}) [27, 28]. Методы иерархического кластерного анализа различаются по способу связывания объектов в кластеры. Наибольшее распространение получили следующие методы связывания: ближнего соседа, дальнего соседа, средней связи, медианной связи [29].

Для определения эффективности степени близости кластеров и методов связывания используется кофенетический корреляционный коэффициент κ [28]. Построение иерархического дерева считается успешным, если кофенетический корреляционный коэффициент близок к единице. Для наиболее успешной кластеризации строится дендрограмма иерархического дерева.

Суть метода k -средних заключается в следующем. Предположим, что заранее определено количество кластеров k , на которые необходимо разбить имеющиеся объекты. В качестве начальных центров кластеров можно выбрать любые k объектов. Для каждого следующего объекта рассчитываются расстояния до центров кластеров и данный объект относится к тому кластеру, расстояние до которого оказалось минимальным. После этого для кластера, в котором увеличилось количество наблюдений, рассчитывается новое положение центра кластера (как среднее по каждому признаку) по всем включенным в кластер объектам.

Нейронная сеть на основе самоорганизующихся карт состоит из компонентов, называемых узлами или нейронами. Изначально задается размерность карты, по ней некоторым образом строится первоначальный вариант карты. В процессе обучения векторы веса узлов приближаются к входным данным. Для каждого наблюдения выбирается наиболее похожий по вектору веса узел и значение его вектора веса приближается к наблюдению. Также к наблюдению приближаются векторы веса нескольких узлов, расположенных рядом. Таким образом, если в множестве входных данных два наблюдения были схожи, на карте им будут соответствовать близкие узлы. Кроме карт Кохонена существует также слой Кохонена, который представляет собой нейронную сеть с конкурирующей активационной функцией. В этом случае в качестве принадлежности к кластеру нейронная сеть выдает номер нейрона ближайшего к исследуемому объекту.

Для классификации данных реализованы наиболее эффективные алгоритмы, такие как дискриминантный анализ, МОВ и метод k -ближайших соседей (k ББ).

В дискриминантном анализе (ДА) рассматривается предположение о нормальном распределении признаков объектов. В линейном ДА (ЛДА) используется линейная комбинация признаков, позволяющая построить классифицирующее правило для объектов. Квадратичный ДА (КДА) работает аналогично с той лишь разницей, что предварительно производится преобразование пространства и в новом пространстве ищется линейная комбинация признаков. В исходном же пространстве эта комбинация будет нелинейной (квадратичной).

В основе работы метода k ББ лежит положение о том, что объект присваивается классу, который является самым распространенным среди его соседей. Для расчета расстояний между объектами можно использовать расстояния d_{euc} , d_{city} , d_{Mink} , d_{mah} , d_{cos} , d_{cor} .

2.3. Снижение размерности данных

Для визуализации данных и выделения значимой информации рассмотрен метод главных компонент [30]. В частности, для представления в наглядной форме измеренных характеристик и соответствующих им групп объектов (ядер и клеток) используются первые две компоненты, что представляет собой ортогональное проецирование многомерной системы на двухмерную плоскость.

2.4. Критерии оценки качества и результаты анализа

Основным критерием для оценки качества классификации и кластеризации выступала ошибка неверно классифицированных объектов. Для того чтобы исследовать, насколько устойчиво могут работать методы на независимых данных, использовался метод перекрестной проверки. За один этап перекрестной проверки происходит разделение имеющихся данных на обучающую часть и тестовый набор. Чтобы получить более точные результаты, разные циклы перекрестной проверки проводятся на разных разбиениях, затем результат усредняется по всем циклам [31].

Методы классификации и кластеризации исследованы на восьми экспериментальных изображениях. В качестве эталонных масок изображений приняты контуры ядер, выделенные

экспертным путем. Рассмотрены следующие размеры обучающей выборки: 15, 30, 50, 70, 100, 150 и 200 объектов. Результаты сравнительного анализа наиболее эффективных алгоритмов кластерного анализа представлены в таблице.

Ошибка кластеризации ERROR для иерархического кластерного анализа и метода k -средних

Метод	Изображение								Среднее значение
	1	2	3	4	5	6	7	8	
Иерархический (d_{cos})	1,4	4,3	28,5	1,7	5,6	3,5	2,8	1,7	6,2
Иерархический (d_{cor})	3	4,3	39,6	4	3,7	4,4	4,8	2,3	8,3
k -средних (d_{euc})	1,9	3,2	30,7	2,2	9	6,7	48,4	1,1	12,9
k -средних (d_{city})	2,8	3,1	42,8	2,5	8,6	8,5	38,5	1,2	13,5
k -средних (d_{cos})	6,9	7,6	47,3	2,2	11,9	22,7	40,5	24,5	20,5
k -средних (d_{cor})	7,1	6,7	49,8	2,7	9,8	22	36,2	23,9	19,8

Для иерархического кластерного анализа две метрики и центроидный метод связывания показали наилучшие результаты с наименьшей ошибкой кластеризации. Наилучшие результаты получены для центроидного метода связывания. Средние значения ошибок кластеризации, вычисленные по восьми изображениям, составляют 6,2 и 8,3 % для косинусного и корреляционного расстояний соответственно.

Наименьшая ошибка для исследуемых изображений – 1,4 %, а максимальная – 39,6 %. Более подробно результаты анализа другими методами вычисления расстояния и связывания показаны в приложении А (<http://www.sstcenter.com/dsa/Staff/Lisitsa/index.html>).

Наименьшая ошибка кластеризации для метода k -средних в два раза больше, чем ошибка для иерархического кластерного анализа, и составляет ERROR = 12,9 %. Две метрики показали сопоставимые результаты кластеризации: евклидово расстояние и метрика города (таблица).

Наихудшие результаты получены для нейронных сетей. Ошибка кластеризации для слоя Кохонена составляет 20,2 %, а для карты Кохонена наименьшая ошибка кластеризации – 20,3 %, что в два раза больше, чем ошибка кластеризации для метода k -средних.

Результаты сравнительного анализа алгоритмов классификации в зависимости от объема обучающей выборки представлены на рис. 2. Для получения достоверных результатов использовался метод перекрестной проверки, из имеющихся данных семь раз независимо выбирались обучающие выборки. Наилучшие результаты получены для линейного дискриминантного классификатора.

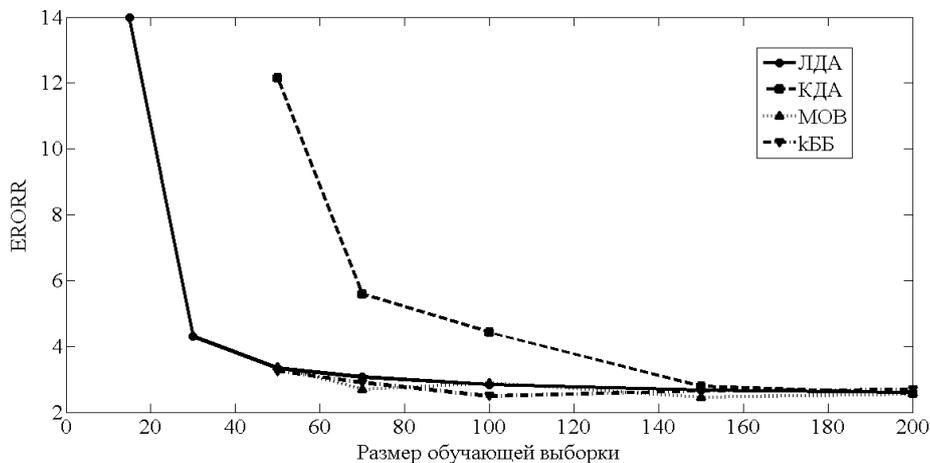


Рис. 2. Зависимость ошибки классификации от размера обучающей выборки для методов линейного и квадратичного дискриминантных анализов, опорных ветров и k -ближайших соседей

Ошибка классификации уменьшается по мере увеличения размера обучающей выборки, при размере выборки от 50 объектов и более она не превосходит 4 %.

Для КДА наименьший размер обучающей выборки должен составлять как минимум 50 объектов. При размере обучающей выборки 150 объектов и более качество классификации методом КДА не отличается от результатов для линейного классификатора.

Использование метода *k*-ближайших соседей является эффективным при размере обучающей выборки от 50 объектов и более и метрики города. В этом случае происходит успешное обучение классификатора, аналогичные результаты получены для линейного метода опорных векторов. Ошибка классификации методами МОВ и *k*ББ варьируется около 3 % (рис. 2).

2.5. Интерфейс CellDataMiner

Главное окно программного пакета CellDataMiner показано на рис. 3. На первом этапе пользователю необходимо загрузить выбранное изображение кнопкой Load Image. Сегментация объектов на изображении начнется при нажатии кнопки Process Image. После выделения объектов на изображении их границы будут показаны в окне программы SegementationAxes, при этом ядра, относящиеся к раковым клеткам, будут подсвечены бирюзовым цветом, а ядра здоровых клеток – фиолетовым.

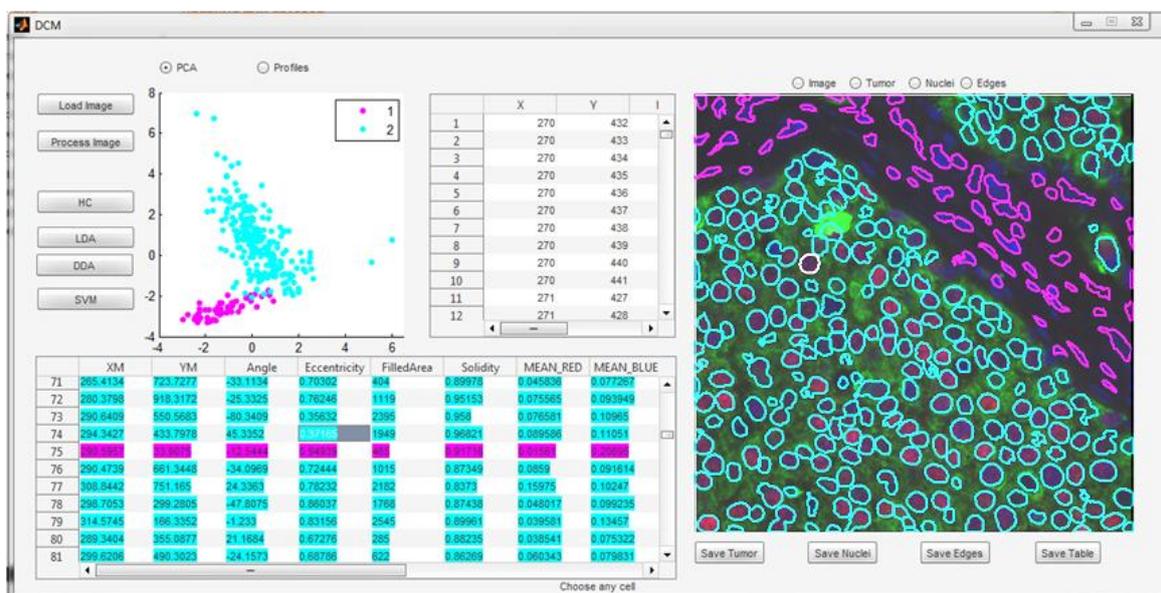


Рис. 3. Общий вид интерфейса программного пакета CellDataMiner

Одновременно в таблице NucleiTable, в столбце PRE-TYPE устанавливаются значения 1 для ядер здоровых клеток и 2 для ядер раковых клеток. В окне отображения границ объектов можно также посмотреть исходное изображение до выделения границ, границы объектов, бинарные изображения сегментации, маски опухоли и ядер (рис. 4).

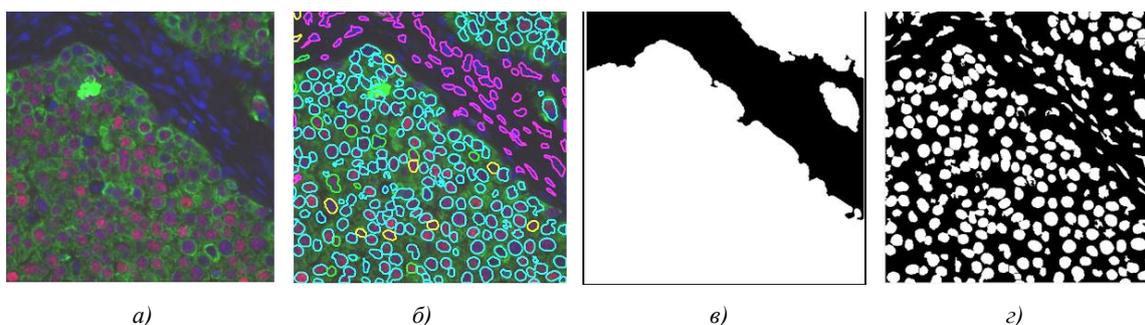


Рис. 4. Примеры изображений в окне SegementationAxes: а) исходное изображение; б) границы объектов; в) маска опухоли; г) маска ядер

В отображении результатов сегментации реализована интерактивная работа пользователя с объектами. Выбрав определенный объект на изображении и щелкнув по нему клавишей мыши, можно посмотреть его параметры в поле Status, а сам объект будет выделен белым цветом. В окне DMAxes можно отобразить сами объекты в пространстве главных компонент, выбрав режим PCA, и посмотреть профили объектов, выбрав режим Profiles.

Интерактивное взаимодействие с пользователем также осуществляется при работе с таблицей NucleiTable. В таблице отображены признаки сегментированных объектов, при этом цветом обозначается принадлежность объектов к классам. При нажатии на объект в таблице он будет подсвечен белым цветом в окне изображения, а значения интенсивностей его пикселей показаны в таблице PixelLevelTable (рис. 3).

Отдельным блоком в ПО вынесены методы анализа данных. С целью разбиения данных на группы реализован метод иерархического кластерного анализа. Для запуска метода необходимо нажать на кнопку HC, после прогона алгоритма индексы классов для объектов можно посмотреть в таблице NucleiTable, объекты классов будут отображены в пространстве главных компонент, расположение объектов классов отобразится на изображении SegmentationAxes. Объекты первого класса отобразятся бирюзовым цветом, объекты второго класса – фиолетовым.

В ПО реализованы два метода кластеризации данных – ЛДА и КДА. Методы требуют ввода обучающей выборки для настройки параметров классификатора. Ввод обучающей выборки осуществляется в таблице NucleiTable в столбце USER-TYPE. При этом для обозначения объектов обучающей выборки первого класса используется цифра 3, сами объекты подсвечиваются на изображении PixelLevelTable желтым цветом, а для обозначения объектов обучающей выборки второго класса используется цифра 4, объекты выделяются зеленым цветом. После того как объекты установлены, запуск анализа осуществляется нажатием кнопки LDA для линейного классификатора и DDA для квадратичного (рис. 3).

Сохранить полученные результаты можно при помощи блока кнопок: SaveTumor – сохранение результатов сегментации опухоли на изображении, SaveNuclei – сохранение результатов сегментации ядер, SaveEdges – сохранение цветного изображения с границами объектов на нем, SaveTable – сохранение результатов анализа данных. Пример исследования изображения показан на рис. 3.

3. Анализ экспериментальных изображений с использованием пакета CellDataMiner

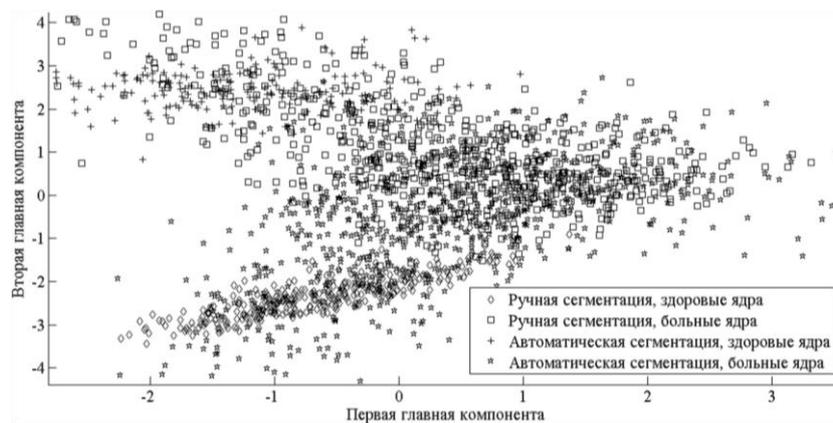
Работоспособность разработанного ПО проверена на примере обработки больших объемов данных, полученных в результате эксперимента по исследованию опухоли молочной железы. Для анализа отобраны 137 изображений. Для каждого экспериментального изображения определено количество нераковых и раковых клеток, а также их профили в пространстве характерных признаков. Общее количество клеток варьируется в пределах 1000–3000, количество раковых клеток – в пределах 100–1700.

Алгоритм успешно сегментировал 120 слайдов. Результат неверной сегментации объектов остальных 17 слайдов вызван недооценкой размеров ядер вследствие наличия малого количества объектов на них при низкой средней интенсивности люминесценции изображения.

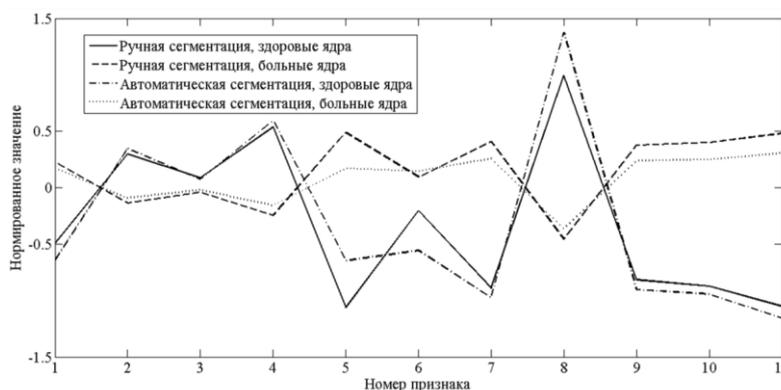
В результате анализа определено, что все виды изображений можно условно разделить на три класса:

1. Изображения, где количество раковых клеток значительно меньше количества здоровых клеток. Количество раковых клеток не превосходит 100.
2. Изображения, где количества двух классов клеток сопоставимы. Количество раковых и здоровых клеток составляет около 500.
3. Изображения, где количество раковых клеток превышает количество здоровых клеток. Количество здоровых клеток не превосходит 100.

На рис. 5 показано отображение найденных объектов в пространстве первых двух главных компонент и средние профили признаков объектов для одного из рассматриваемых изображений, определенные автоматическим методом на основе результатов сегментации маски опухоли и экспертным путем.



а)



б)

Рис. 5. Результаты анализа экспериментальных изображений с использованием пакета CellDataMiner: а) объекты, найденные экспертным путем и автоматическим методом в пространстве первых двух главных компонент; б) средние профили признаков объектов, найденных экспертным путем и автоматическим методом

Отображения объектов в пространстве первых двух главных компонент, найденные автоматическим методом и экспертным путем, имеют различия (рис. 5, а). Это можно объяснить тем, что у автоматического метода существуют ошибки локализации и сегментации, которые вносят искажения. Однако средние профили признаков объектов достаточно близки, за исключением нескольких признаков, для которых характерно значительное количество значений-выбросов.

Результаты анализа демонстрируют, что отобранные признаки позволяют выделять кластеры объектов на изображениях. При этом необходимо отметить, что для классификации объектов достаточным является использование параметров формы. Ядра раковых клеток в большинстве своем обладают бóльшими размерами, чем ядра здоровых клеток. Для нераковых объектов характерна бóльшая экспрессия в синем канале, чем для раковых клеток. Необходимо отметить, что зеленый краситель используется только для цитоплазмы, тем не менее его накопление в ядрах раковых клеток происходит интенсивнее, чем в ядрах здоровых клеток.

Заключение

В работе представлено программное средство для анализа люминесцентных изображений раковых клеток. Разработанный программный пакет CellDataMiner интегрирует объектно-ориентированную методологию анализа биообъектов люминесцентных изображений на пиксельном уровне с последующим использованием методов анализа данных для выделения групп раковых клеток или исследования стадий развития онкологического заболевания.

В качестве личного вклада авторов следует отметить:

- разработку объектно-ориентированной методологии анализа биообъектов на пиксельном уровне изображения;

– разработку и программную реализацию автоматического алгоритма сегментации с учетом статистической взаимосвязи между каналами изображения, позволяющего повысить точность сегментации;

– полную автоматизацию программных средств анализа больших объемов данных;

– реализацию средств интерактивного анализа сегментированных объектов изображений;

– проведение сравнительного анализа наиболее эффективных алгоритмов классификации и кластерного анализа сегментированных объектов биоизображений с целью интеграции наилучших алгоритмов в программный пакет.

Реализованные в ПО методы исследованы на экспериментальных данных. Наилучшие результаты кластеризации получены для иерархического кластерного анализа, наихудшие результаты получены для слоя Кохонена. Для классификации данных наилучшим методом является ЛДА; при больших размерах обучающей выборки КДА, *k*ББ и МОВ показывают такие же результаты, как и ЛДА. Их ошибка классификации не превосходит 5 %.

С использованием разработанного программного средства выполнен анализ 137 изображений раковых клеток, полученных в ходе экспериментов по исследованию микрочипов тканей опухолей молочной железы. Успешная сегментация была выполнена для 120 слайдов. Установлено точное количество раковых и нераковых клеток на каждом из исследуемых изображений. Общее количество клеток варьируется в пределах 1000–3000. Количество раковых клеток изменяется в пределах 100–1700. Рассчитаны основные характеристики раковых клеток: интенсивность биомаркера рака, параметры формы и структуры биообъектов и т. д. Выбранные характеристики позволяют выделить два класса объектов. Для разделения объектов анализируемых изображений на классы достаточно использовать признаки формы. Результаты анализа экспериментальных изображений позволяют сделать следующий вывод: автоматически сегментированные контуры объектов хорошо согласуются с контурами, выделенными вручную, а подобранные параметры объектов могут использоваться для классификации ядер клеток.

Список литературы

1. Spatial quantitative analysis of fluorescently labeled nuclear structures: problems, methods, pitfalls / O. Ronneberger [et al.] // *Chromosome Res.* – 2008. – Vol. 16(3). – P. 523–562.
2. *Molecular Biology of the Cell* / B. Alberts [et al.]. – N. Y. : Garland Science, 2012.
3. Rueden, C. VisBio: a computational tool for visualization of multidimensional biological image data / C. Rueden, K. Eliceiri, J. White // *Traffic.* – 2004. – Vol. 5. – P. 411–417.
4. BioImageXD: an open, general-purpose and high-throughput image-processing platform / P. Kankaanpää [et al.] // *Nat Methods.* – 2012. – Vol. 9(7). – P. 683–689.
5. Computer control of microscopes using μ Manager / A. Edelstein [et al.] // *Current Protocols in Molecular Biology.* – 2010. – Vol. 14(20). – P. 1–17.
6. The FARSIGHT Project: Associative 4D/5D Image Analysis Methods for Quantifying Complex and Dynamic Biological Microenvironments / B. Roysam [et al.] // *Microscopy and Microanalysis.* – 2008. – Vol. 14 (Supplement S2). – P. 60–61.
7. Schneider, C.A. NIH Image to ImageJ: 25 years of image analysis / C.A. Schneider, W.S. Rasband, K.W. Eliceiri // *Nat Methods.* – 2012. – Vol. 9(7). – P. 671–675.
8. Out, W.A. A new method for morphometric analysis of opal phytoliths from plants / W.A. Out, J.F. Pertusa Grau, M. Madella // *Microsc Microanal.* – 2014. – Vol. 20(6). – P. 1876–1887.
9. Fiji: an open-source platform for biological-image analysis / J. Schindelin [et al.] // *Nat Methods.* – 2012. – Vol. 9(7). – P. 676–682.
10. Optimized digital counting colonies of clonogenic assays using ImageJ software and customized macros: comparison with manual counting / Z. Cai [et al.] // *Int. J. Radiat. Biol.* – 2011. – Vol. 87(11). – P. 1135–1146.
11. Designing a wearable navigation system for image-guided cancer resection surgery / P. Shao [et al.] // *Ann Biomed Eng.* – 2014. – Vol. 42(11). – P. 2228–2237.
12. Computer-aided Image Processing of Angiogenic Histological / M. Sprindzuk [et al.] // *J. Clin. Med. Res.* – 2009. – Vol. 1(5). – P. 249–261.

13. Bray, M.A. Using CellProfiler for Automatic Identification and Measurement of Biological Objects in Images / M.A. Bray, M.S. Vokes, A.E. Carpenter // *Current Protocols in Molecular Biology*. – 2015. – Vol. 109. – P. 14 17 1–14 17 13.
14. CellProfiler Analyst: data exploration and analysis software for complex image-based screens / T.R. Jones [et al.] // *BMC Bioinformatics*. – 2008. – Vol. 9. – 482 p.
15. Gonzalez, W. Eddins, Digital Image Processing Using MATLAB / W. Gonzalez. – 2nd edition. – Gatesmark Publishing, 2009.
16. Novoselova, N. Supervised Clustering of Genes for Multi-Class Phenotype Classification / N. Novoselova, I. Tom // *Modeling and Simulation (MS'2012)*. – Minsk : BSU, 2012. – P. 32–36.
17. Alilou, M. Segmentation of cell nuclei in heterogeneous microscopy images: a reshapable templates approach / M. Alilou, V. Kovalev, V. Taimouri // *Comput Med Imaging Graph*. – 2013. – Vol. 37(7–8). – P. 488–499.
18. Segmentation of microscope cell images via adaptive eigenfilters / S. Kumar [et al.] // *Image Proc. ICIP'04. Intern. Conf.* – Singapore, 2004. – Vol. 1. – P. 135–138.
19. Abbas, S.S. A comparative study of cell classifiers for image-based high-throughput screening / S.S. Abbas, T.M. Dijkstra, T. Heskes // *BMC Bioinformatics*. – 2014. – Vol. 15. – 342 p.
20. Quantitative analysis of estrogen receptor heterogeneity in breast cancer / G.G. Chung [et al.] // *Lab. Invest.* – 2007. – Vol. 87(7). – P. 662–669.
21. Camp, R.L. Automated subcellular localization and quantification of protein expression in tissue microarrays / R.L. Camp, G.G. Chung, D.L. Rimm // *Nat. Med.* – 2002. – Vol. 8(11). – P. 1323–1327.
22. Simulation Model for Three-Channel Luminescent Images of Cancer Cell Populations / E.V. Lisitsa [et al.] // *Journal of Applied Spectroscopy*. – 2015. – Vol. 81(6). – P. 996–1003.
23. Review of free software tools for image analysis of fluorescence cell micrographs / V. Wiesmann [et al.] // *Journal of Microscopy*. – 2015. – Vol. 257, iss. 1. – P. 39–53.
24. Алгоритм автоматической сегментации границ ядер раковых клеток на трехканальных люминесцентных изображениях / Е.В. Лисица [и др.] // *Журнал прикладной спектроскопии*. – 2015. – № 82(4). – С. 598–607.
25. Разработка методов цифровой обработки люминесцентных изображений биологических объектов / В.В. Апанасович [и др.]. – Минск : Белорусский фонд фундаментальных исследований, 2013.
26. High-content phenotypic profiling of drug response signatures across distinct cancer cells / P.D. Caie [et al.] // *Mol Cancer Ther.* – 2010. – Vol. 9(6). – P. 1913–1926.
27. Методы и модели анализа данных: OLAP и Data Mining / А.А. Барсегян [и др.]. – СПб. : БХВ-Петербург, 2004. – 336 с.
28. Urangun, B. The discrimination of interaural level difference sensitivity functions: development of a taxonomic data template for modelling / B. Urangun, R. Rajan. – Clayton : Monash University, 2013. – P. 1471–2202.
29. Мандель, И.Д. Кластерный анализ / И.Д. Мандель. – М. : Финансы и статистика, 1988. – 176 с.
30. Прикладная статистика. Классификация и снижение размерности / С.А. Айвазян [и др.]. – М. : Финансы и статистика, 1989. – 607 с.
31. Воронцов, К.В. Комбинаторный подход к оценке качества обучаемых алгоритмов / К.В. Воронцов // *Математические вопросы кибернетики*. – М. : Физматлит, 2004. – Vol. 13. – С. 5–36.

Поступила 21.10.2015

¹Белорусский государственный университет,
Минск, пр. Независимости, 4
e-mail: Lisitsa@bsu.by

²Университет Джорджа Вашингтона,
1922 F str NW, Old Main
e-mail: apanasovich@gwu.edu

Y.U. Lisitsa, M.M. Yatskou, V.V. Apanasovich, T.V. Apanasovich

**THE SOFTWARE PACKAGE CellDataMiner FOR DATA ANALYSIS
OF FLUORESCENT IMAGES OF CANCER CELLS**

The paper presents the software package CellDataMiner for data analysis of luminescent images of cancer cells. The comparative analysis of classification and clustering methods is carried out. The most sufficient of them are implemented in the software. The software package is tested on the dataset of the experimental images of breast cancer.

УДК 004.031.6

Н.А. Петровский, А.В. Станкевич, А.А. Петровский

**CORDIC-ТЕХНИКА ДЛЯ ФИКСИРОВАННОГО УГЛА ВРАЩЕНИЯ
В ОПЕРАЦИИ УМНОЖЕНИЯ КВАТЕРНИОНОВ**

Предлагается ряд решений ключевого элемента параунитарного банка фильтров на основе алгебры кватернионов – умножителя кватернионов с использованием CORDIC (Coordinate Rotation Digital Computer)-техники для фиксированного угла вращения, в которых в отличие от известных решений параметры управления 4D-вращением представляются нелинейной функцией числа сдвигов входных операндов операции микровращения. Предложенный подход проектирования умножителя на кватернион-константу позволяет достигать максимальной производительности схемы умножителя при скромном использовании ресурсов, например FPGA.

Введение

Алгебра кватернионов \mathbb{H} является ассоциативной некоммутативной четырехмерной алгеброй [1] $\mathbb{H} = \{ \mathbf{q} = q_1 + q_2i + q_3j + q_4k \mid q_1, q_2, q_3, q_4 \in \mathbb{R} \}$, где ортогональные мнимые части подчиняются следующим законам умножения: $i^2 = j^2 = k^2 = ijk = -1$, $ij = -ji = k$, $jk = -kj = i$, $ki = -ik = j$. Операции сложения и умножения кватернионов могут быть реализованы с использованием векторно-матричной алгебры. Для этого кватернионы представляются в виде четырехмерных векторов. Умножение кватернионов в векторной форме производится по правилу умножения вектора на матрицу. В связи с тем что операция умножения некоммутативна, продукт умножения определяется операторами умножения «левый» $\mathbf{M}^+(q)$ и «правый» $\mathbf{M}^-(q)$ [2]:

$$qx \Leftrightarrow \underbrace{\begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \\ q_2 & q_1 & -q_4 & q_3 \\ q_3 & q_4 & q_1 & -q_2 \\ q_4 & -q_3 & q_2 & q_1 \end{bmatrix}}_{\mathbf{M}^+(q)} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\mathbf{x}}, \quad xq \Leftrightarrow \underbrace{\begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \\ q_2 & q_1 & q_4 & -q_3 \\ q_3 & -q_4 & q_1 & q_2 \\ q_4 & q_3 & -q_2 & q_1 \end{bmatrix}}_{\mathbf{M}^-(q)} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\mathbf{x}}. \quad (1)$$

Обе матрицы (1) ортогональны до масштабного коэффициента, заданного нормой кватернионов $|q| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2}$: $\mathbf{M}^\pm(q)^{-1} = (1/|q|)\mathbf{M}^\pm(q)^T$, где T – знак транспонирования матрицы. Как следует из структур матриц, их свойства не зависят от значений компонент кватерниона и, следовательно, не чувствительны к их модификации, например квантованию. Матрицы $\mathbf{M}^+(q)$ и $\mathbf{M}^-(q)$ соотносятся между собой как

$$\mathbf{M}^-(q) = \mathbf{D}_C \mathbf{M}^+(q)^T \mathbf{D}_C, \quad (2)$$

где $\mathbf{D}_C = \text{diag}(1, -\mathbf{I}_3)$ описывает в матричной нотации оператор гиперкомплексного сопряжения. Определив сопряженный кватернион $\bar{q} = q_1 - q_2i - q_3j - q_4k$ в векторно-матричном виде как $\bar{\mathbf{q}} = \mathbf{D}_C \mathbf{q}$, можно получить

$$\mathbf{M}^\pm(\bar{q}) = \mathbf{M}^\pm(q)^T, \quad (3)$$

тогда выражение (2) переопределяется следующим образом:

$$\mathbf{M}^-(\bar{q}) = \mathbf{D}_C \mathbf{M}^+(q) \mathbf{D}_C. \quad (4)$$

Последнее равенство показывает, что результаты, полученные для умножения «левого» qx или «правого» xq , могут быть применимы к умножению на сопряженный кватернион. В частности, это доказывает, что вычислительная сложность операторов умножений кватернионов одинакова. Данное обстоятельство позволяет сосредоточить внимание на организации вычисления продукта qx оператора «левого» умножения $\mathbf{M}^+(q)$.

Последние 20 лет в цифровой обработке сигналов алгебра кватернионов рассматривается как новая парадигма, позволяющая осуществлять обработку сигналов непосредственно в многомерном домене. Кватернионы уже с успехом использовались в цифровой обработке сигналов, например адаптивных фильтрах Калмана [3], при оценке спектра по методу MUSIC [4], в сингулярной декомпозиции матриц [5], параунитарных банках фильтров (ПУБФ) – Q-ПУБФ [6–8] и других приложениях [9, 10]. ПУБФ могут рассматриваться как наиболее значимое преобразование среди многоскоростных систем цифровой обработки сигналов [11] для систем мультимедиа. Это обусловлено тем фактом, что подобные банки являются преобразованиями без потерь в дополнение к гарантированной перфективной реконструкции сигнала. Точное соотношение между энергиями во всем частотном диапазоне и в субполосах сильно упрощает теоретические выкладки и делает ПУБФ полезными в применении к компрессии (кодированию) изображений. В работе [8] утверждается, что для каждой ортогональной матрицы размерностью 4×4 существует уникальная (с точностью до знака) пара единичных кватернионов P и Q ($|P|=1$, $|Q|=1$), таких, что $\mathbf{M}^+(P) \cdot \mathbf{M}^-(Q) = \mathbf{M}^-(Q) \cdot \mathbf{M}^+(P)$. Факторизации полифазных передаточных матриц анализа $\mathbf{E}(z)$ и синтеза $\mathbf{D}(z)$ решетчатой структуры Q-ПУБФ в алгебре кватернионов будут соответственно равны [8]:

$$\mathbf{E}(z) = \left(\prod_{i=N-1}^1 \mathbf{M}^+(Q_i) \Lambda(z) \right) \mathbf{M}^+(Q_0) \mathbf{M}^-(P_0); \quad (5)$$

$$\mathbf{D}(z) = \mathbf{M}^-(\bar{P}_0) \mathbf{M}^+(\bar{Q}_0) \left(\prod_{i=1}^{N-1} \hat{\Lambda}(z) \mathbf{M}^+(\bar{Q}_i) \right), \quad (6)$$

где \bar{P}_0 и \bar{Q}_0 – сопряженные кватернионы; $\Lambda(z) = \text{diag}(z^{-1}, \mathbf{I}_{M-1})$ и $\hat{\Lambda}(z) = \text{diag}(1, z^{-1} \mathbf{I}_{M-1})$ – матрицы задержек; \mathbf{I}_{M-1} – $M \times M$ единичная матрица. Данная факторизация всегда соответствует ортогональному преобразованию сигналов, даже если ее компоненты квантованы. Это происходит потому, что столбцы каждой матрицы операторов умножения кватернионов (1) составлены из одних и тех же элементов с точностью до знака. Таким образом, множитель кватернионов является альтернативным модулем построения Q-ПУБФ и может рассматриваться как обобщение планарных вращений Гивенса на 4D-пространство [7, 8]. Данные решетчатые структуры Q-ПУБФ на кватернионах являются системами без потерь, несмотря на квантование коэффициентов. Более того, предложенный подход дает возможность непосредственного выражения условия регулярности первого порядка, которое зафиксировано в значениях коэффициентов решетчатой структуры Q-ПУБФ и легко удовлетворяется даже для арифметики с фиксированной запятой [7, 8].

Следовательно, базовым элементарным преобразованием Q-ПУБФ является умножение кватернионов, где один из сомножителей – кватернион-константа, т. е. кватернион с постоянными параметрами. При этом умножение кватернионов – ключевая операция, от эффективной реализации которой зависят характеристики всего преобразования. Однако прямое умножение матрицы на вектор потребует 16 умножений действительных чисел и 12 алгебраических сложений. В [12] показано, как можно вычислить продукт умножения кватернионов на основе восьми действительных умножений, но данная техника не подходит для случая, когда один операнд есть кватернион-константа. Алгоритм выполнения умножения кватернионов на основе лестничной структуры в арифметике с фиксированной запятой представлен в работе [13] (12 операций умножения на действительные числа), где основной компонент схемы – множитель-накопитель. Реализация множителя в этом случае требует больших затрат компонент структуры ПЛИС типа FPGA. Представляют интерес вычислительные схемы множителя кватернионов без использования устройств действительных умножений. Известно решение реали-

зации множителя кватернионов на распределенной арифметике [14], которое позволяет получать сбалансированные схемные решения как по производительности, так и по потребляемой мощности. Однако для достижения высокой скорости и точности вычислений здесь необходим большой объем памяти.

В настоящее время в области систем мультимедиа достигнут большой прогресс в развитии алгоритмов и архитектурных решений для высокопроизводительных вычислений с малыми аппаратными затратами – CORDIC-алгоритмов [15]. Преимущество CORDIC заключается в том, что с помощью простых операций сдвига и сложения $a \pm b2^{-i}$ можно вычислять ряд задач во многих областях, таких как обработка сигналов и изображений, системы телекоммуникаций, робототехника, 3D-графика. Параллельные и конвейерные структуры CORDIC-процессоров позволяют достигать высокой скорости вычислений [15].

Целью настоящей статьи является представление ряда решений ключевого элемента Q-ПУБФ – множителя кватернионов на базе CORDIC-алгоритмов.

1. Базовая CORDIC-техника

1.1. 2-D-CORDIC-алгоритм

Вычислительный CORDIC-алгоритм, предложенный Волдером [16] и позднее развитый Вальтером [17], представляет собой алгоритм, оперирующий 2D-векторами для вычисления вращений с помощью простых арифметических примитивов: сдвига и сложения. Поворот вектора $\mathbf{x} = [x_1 \ x_2]^T$ на угол φ может быть описан произведением данного вектора на соответствующую матрицу вращения $\mathbf{R}_2(\varphi)$:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix}}_{\mathbf{R}_2(\varphi)} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\mathbf{x}}. \quad (7)$$

Вместе с тем выражение (7) описывает произведение комплексного числа $(x_1 + jx_2)$ на комплексную экспоненту $e^{-j\varphi}$. CORDIC-алгоритм аппроксимирует матрицу вращения $\mathbf{R}_2(\varphi)$ произведением N элементарных вращений $\mathbf{R}_2(i)$:

$$\mathbf{R}_2(\varphi) = \prod_{i=0}^{N-1} \mathbf{R}_2(i); \quad (8)$$

$$\mathbf{R}_2(i) = (1 + \delta t_i^2)^{-1/2} \mathbf{U}_2(i), \quad \mathbf{U}_2(i) = \begin{bmatrix} 1 & -\delta \cdot \sigma(i) t_i \\ \delta \cdot \sigma(i) t_i & 1 \end{bmatrix}, \quad (9)$$

где $\mathbf{U}_2(i)$ – ненормализованная часть $\mathbf{R}_2(i)$; $\delta = 1$, если вращение осуществляется в евклидовом пространстве; $\delta = -1$, если вращение осуществляется в псевдоевклидовом пространстве (гиперболическое вращение); $t_i = 2^{-i}$. Параметры N , $\sigma(i)$, i подбираются таким образом, что результат произведения (8) аппроксимирует вращение на угол

$$\varphi \approx \sum_{i=0}^{N-1} \sigma(i) \cdot 2^{-i}, \quad (10)$$

где N – число вращений, а параметр $\sigma(i) \in \{-1, 1\}$ определяет направление вращения для оставшейся части угла. CORDIC-алгоритм состоит из итерационного процесса вращения и процесса масштабирования. На каждой итерации входной вектор \mathbf{x} поворачивается на угол 2^{-i} , но результат вращения не соответствует фиксированному радиусу: длина вектора увеличивается на величину $(1 + 2^{-2i})^{1/2}$, т. е. необходима нормализация размера вектора, которая выполняется в

процессе масштабирования. Таким образом, на i -й итерации для соответствующего входного вектора $\mathbf{x}(i) = [x_1(i) \ x_2(i)]^T$ выходной вектор $\mathbf{x}(i+1) = [x_1(i+1) \ x_2(i+1)]^T$ вычисляется следующим образом:

$$\mathbf{x}(i+1) = \mathbf{U}_2(i) \cdot \mathbf{x}(i); \quad (11)$$

$$\varphi(i+1) = \varphi(i) - \sigma(i)2^{-i} \quad \text{для } i = \overline{0, N-1}, \quad (12)$$

где $\mathbf{U}_2(i)$ – оператор микровращения на i -й итерации CORDIC-алгоритма; $\varphi(i)$ – оставшийся угол после $(i-1)$ -й итерации. Амплитуда результата нормализуется на этапе масштабирования после окончания итерационного этапа: выходной вектор $\mathbf{x}(N)$ делится на масштабный фактор $T = \prod_{i=0}^{N-1} (1 + 2^{-2i})^{1/2}$. Итак, окончательный результат $\mathbf{x}_{out}(N)$ CORDIC-алгоритма может быть представлен как

$$\mathbf{x}_{out}(N) = \frac{1}{T} \mathbf{x}(N) = \left(\prod_{i=0}^{N-1} (1 + 2^{-2i})^{1/2} \right)^{-1} \mathbf{x}(N). \quad (13)$$

Таким образом, масштабирование требует дополнительной вычислительной операции, но если масштабный фактор представить в виде

$$\frac{1}{T} = \sum_{s=0}^{S-1} \sigma(s)2^{-i(s)}, \quad (14)$$

где $\sigma(s) = \pm 1$ и $0 \leq i(s) \leq N-1$, то процесс масштабирования может быть описан рекурсивным выражением:

пусть $\mathbf{x}(0) = 0$ и $\mathbf{x}(S) = \mathbf{x}_{out}(N)$, тогда

$$\mathbf{x}(s+1) = \mathbf{x}(s) + \sigma(s)2^{-i(s)} \mathbf{x}(N) \quad \text{для } s = 0, 1, \dots, S-1. \quad (15)$$

В стандартной CORDIC-технике параметры алгоритма следующие: $\sigma(i) \in \{-1, 1\}$, $i = \overline{0, N-1}$, и $N = B$, где B – число разрядов в двоичном представлении компонент векторов, т. е. сдвиг и знак оператора направления вращения для обоих процессов (итерационного и процесса масштабирования) определяются в онлайн-режиме на основе входных данных. Выбор соответствующих параметров алгоритма определяет фиксированное число микровращений (итераций).

1.2. Многомерный CORDIC-алгоритм

В работе [18] были предложены быстрые 3D- и 4D-CORDIC-алгоритмы, число итераций которых немного больше, чем для одного 2D-CORDIC-вращения. Данные алгоритмы основываются на фундаментальной теореме Кэли [19]. Формула Кэли устанавливает связь между ортогональными и кососимметрическими операторами в евклидовом пространстве и позволяет любое n D евклидово вращение \mathbf{R}_n , для которого характеристическое число равно -1 , представить как

$$\mathbf{R}_n = (\mathbf{I}_n - \mathbf{T}_n)(\mathbf{I}_n + \mathbf{T}_n)^{-1}, \quad (16)$$

где \mathbf{I}_n – единичная матрица $n \times n$; \mathbf{T}_n – кососимметрическая матрица (оператор), т. е. $\mathbf{T}_n^T = -\mathbf{T}_n$. Например, для i -го элементарного 3D-вращения в [18] предлагается матрицу $\mathbf{T}_{3,i}$ конструировать следующим образом:

$$\mathbf{T}_3(i) = \begin{bmatrix} 0 & -\sigma_1(i)t & \sigma_2(i)t \\ \sigma_1(i)t & 0 & -\sigma_1(i)t \\ -\sigma_2(i)t & \sigma_1(i)t & 0 \end{bmatrix}, \quad \sigma_1(i), \sigma_2(i) \in \{1, -1\}, \quad (17)$$

где $t = 2^{-i}$. Соответствующая элементарная нормализованная матрица вращения согласно (16) определится как

$$\mathbf{R}_3(i) = (\mathbf{I}_3 - \mathbf{T}_{3,i})(\mathbf{I}_3 + \mathbf{T}_{3,i})^{-1} = \frac{1}{\sqrt{1+3 \cdot t^2}} \begin{bmatrix} 1-t^2 & 2(\sigma_1(i)t+t^2) & -2(\sigma_2(i)t+t^2) \\ -2(\sigma_1(i)t+t^2) & 1-t^2 & 2(\sigma_1(i)t+t^2) \\ 2(\sigma_2(i)t+t^2) & -2(\sigma_1(i)t+t^2) & 1-t^2 \end{bmatrix}. \quad (18)$$

Данные вращения выполняются относительно оси $[\sigma_1(i) \ \sigma_2(i) \ \sigma_1(i)]^T$, что соответствует направляющему вектору либо $[1 \ 1 \ 1]^T$, либо $[1 \ -1 \ 1]^T$. Когда вектор $[x_1 \ x_2 \ x_3]^T$ следует привести к направлению $[1 \ 0 \ 0]^T$, т. е. первой координатной оси, знаки направления вращения на i -й итерации $\sigma_1(i)$ и $\sigma_2(i)$ выбираются согласно следующему закону управления: $\sigma_1(i) = \text{sign}(x_{1,i-1} \cdot x_{2,i-1})$, $\sigma_2(i) = -\text{sign}(x_{1,i-1} \cdot x_{2,i-1})$, где $x_{j,i-1}$ определяет j -ю компоненту вектора в начале i -й итерации. Данный метод не является систематическим подходом конструирования nD CORDIC элементарных матриц вращения \mathbf{R}_n для произвольных n . Согласно [18] это его главный недостаток.

CORDIC-алгоритм может быть расширен на многомерный случай с использованием преобразования Хаусхолдера [18], матрица отражения которого определяется следующим образом:

$$\mathbf{H}_n = \mathbf{I}_n - 2 \frac{\mathbf{u} \cdot \mathbf{u}^T}{\mathbf{u}^T \cdot \mathbf{u}}, \quad (19)$$

где \mathbf{u} – n -мерный вектор. Результат преобразования $(\mathbf{H}_n \cdot \mathbf{v})$ есть отражение n -мерного вектора \mathbf{v} по отношению к гиперплоскости с нормалью \mathbf{u} , которая проходит через начало координат. Таким образом, CORDIC-алгоритм на основе отражения Хаусхолдера выполняет операцию векторизации n -мерного вектора к одной из осей. Ненормализованная матрица вращения для n -мерного вектора соответствующей i -й итерации $\mathbf{R}_{Hn}(i)$ определяется как произведение двух простых отражений Хаусхолдера [5]:

$$\mathbf{R}_{Hn}(i) = \left(\mathbf{I}_n - 2 \frac{\mathbf{e}_1 \mathbf{e}_1^T}{\mathbf{e}_1^T \mathbf{e}_1} \right) \left(\mathbf{I}_n - 2 \frac{\mathbf{u}_i \mathbf{u}_i^T}{\mathbf{u}_i^T \mathbf{u}_i} \right), \quad (20)$$

где $\mathbf{e}_1 = [1 \ 0 \ \dots \ 0]^T$ и $\mathbf{u}_i = [1 \ \sigma_1(i) \cdot t_i \ \dots \ \sigma_{n-1}(i) \cdot t_i]^T$, $t_i = \tan \varphi_i = 2^{-i}$, а управляющие сигналы направления вращения выбираются как

$$\sigma_j(i) = \text{sign}(x_{1,i-1} \cdot x_{j+1,i-1}), \quad 1 \leq j \leq n-1. \quad (21)$$

Для $n=4$ нормализованная матрица вращения $\mathbf{R}_{H4}(i)$ согласно (20) задается как функция сдвигов и управляющих сигналов направления вращения [5]:

$$\mathbf{R}_{H4}(i) = \frac{1}{\sqrt{1+3 \cdot t_i^2}} \begin{bmatrix} 1 & -\sigma_1(i)t_i & -\sigma_2(i)t_i & -\sigma_3(i)t_i \\ \sigma_1(i)t_i & 1 & \sigma_3(i)t_i & -\sigma_2(i)t_i \\ \sigma_2(i)t_i & -\sigma_3(i)t_i & 1 & \sigma_1(i)t_i \\ \sigma_3(i)t_i & \sigma_2(i)t_i & -\sigma_1(i)t_i & 1 \end{bmatrix}. \quad (22)$$

Следовательно, CORDIC-алгоритм Хаусхолдера представляет собой обобщение CORDIC-алгоритма на nD -мерный случай для осуществления вращения в евклидовом и псевдо-евклидовом пространствах. Следует также отметить, что nD CORDIC-алгоритм хорошо отображается на параллельные структуры процессоров [18].

В [20] описан 4D-CORDIC-алгоритм умножения кватернионов для сингулярного разложения матриц на основе преобразования Хаусхолдера. Авторы показывают, что матрицы умножения (1) можно записать в виде факторизации:

$$\mathbf{M}^\pm(q) \approx \prod_{n=0}^{N-1} \frac{1}{|\zeta(\tau(i), \sigma_1(i), \sigma_2(i), \sigma_3(i))|} \mathbf{M}^\pm(\zeta(\tau(i), \sigma_1(i), \sigma_2(i), \sigma_3(i))), \quad (23)$$

где кватернион \mathbf{q} на i -й итерации 4D-CORDIC-алгоритма аппроксимируется как $\zeta(\tau(i), \sigma_1(i), \sigma_2(i), \sigma_3(i)) = 1 + \sigma_1(i)2^{\tau(i)}i + \sigma_2(i)2^{\tau(i)}j + \sigma_3(i)2^{\tau(i)}k$, а его матрица умножения, которая описывает i -ю итерацию 4D-CORDIC-алгоритма, представляется следующим образом:

$$\mathbf{M}^\pm(\zeta(\tau(i), \sigma_1(i), \sigma_2(i), \sigma_3(i))) \approx \begin{bmatrix} 1 & -\sigma_1(i)2^{\tau(i)} & -\sigma_2(i)2^{\tau(i)} & -\sigma_3(i)2^{\tau(i)} \\ \sigma_1(i)2^{\tau(i)} & 1 & \mp\sigma_3(i)2^{\tau(i)} & \pm\sigma_2(i)2^{\tau(i)} \\ \sigma_2(i)2^{\tau(i)} & \pm\sigma_3(i)2^{\tau(i)} & 1 & \mp\sigma_1(i)2^{\tau(i)} \\ \sigma_3(i)2^{\tau(i)} & \mp\sigma_2(i)2^{\tau(i)} & \pm\sigma_1(i)2^{\tau(i)} & 1 \end{bmatrix}. \quad (24)$$

Параметр количества двоичных сдвигов $\tau(i)$ зависит от индекса итераций i , т. е. $\tau(i) = -i$, а параметры управления направлением вращения $\sigma_1(i), \sigma_2(i), \sigma_3(i)$ принимают значения 1 или -1 .

Таким образом, 4D-CORDIC-алгоритм умножения кватернионов с матрицей умножения (24), предложенный в [20], определяется как

$$\mathbf{x}_{i+1} = \mathbf{M}^\pm(\zeta(\tau(i), \sigma_1(i), \sigma_2(i), \sigma_3(i))) \cdot \mathbf{x}_i \quad (i = \overline{0, N-1}; \mathbf{x}_0 = \mathbf{x}). \quad (25)$$

Для того чтобы 4D-вектор \mathbf{X} привести к направлению первой канонической оси, сигналы управления направлением вращения на i -й итерации определяются кватернионом \mathbf{X}_i :

$$\sigma_1(i) = -\text{sign}(x_{1,i}) \cdot \text{sign}(x_{2,i}), \quad \sigma_2(i) = -\text{sign}(x_{1,i}) \cdot \text{sign}(x_{3,i}), \quad \sigma_3(i) = -\text{sign}(x_{1,i}) \cdot \text{sign}(x_{4,i}). \quad (26)$$

Норма кватерниона \mathbf{X}_{i+1} , или длина вектора на i -й итерации, согласно (22) равна $|\zeta(\tau(i), \sigma_1(i), \sigma_2(i), \sigma_3(i))| = \sqrt{1 + 3 \cdot 2^{-2\tau(i)}}$. Следовательно, выходной результат итерационного процесса \mathbf{X}_N нормализуется на коэффициент $1/K = \prod_{i=0}^{N-1} 1/\sqrt{1 + 3 \cdot 2^{-2\tau(i)}}$, который может быть

аппроксимирован в соответствии с (14) и тогда умножение на данный коэффициент реализуется отдельно от итерационного процесса как серия сдвигов и сложений (процесс масштабирования). Окончательный результат 4D-CORDIC-алгоритма равен $\mathbf{x}_{out,N} = (1/K)\mathbf{x}_N$. Анализ матрицы (24) показывает, что при увеличении числа итераций i матрица $\mathbf{M}^\pm(\zeta(\tau(i), \sigma_1(i), \sigma_2(i), \sigma_3(i)))$ сходится к единичной матрице 4×4 . Схема вычисления итерации (микровращений) 4D-CORDIC-алгоритма может быть реализована на четырех регистрах сдвига и на четырех четырехходовых сумматорах. Это обуславливает необходимость дополнительной логики контроля переполнения, а при реализации на ПЛИС типа FPGA такие сумматоры требуют повышенного аппаратного ресурса. Схема микровращений 4D-CORDIC-алгоритма не может быть использована на этапе масштабирования результата итерационного процесса, 4D-итерации (25), (26) не всегда сходятся [20]. Как показали экспериментальные исследования, число кватернионов на множестве 10^4 случайно сгенерированных единичных кватернионов, для которых заданная точность недостижима (алгоритм не сходится), изменялось от 6 до 18 % для точности CORDIC-процесса соответственно 10^{-2} и 10^{-6} .

2. 4D-CORDIC-алгоритм умножения на кватернион-константу

2.1. Постановка задачи

Имеется множество алгоритмов, например таких, как дискретное косинусное преобразование или быстрое преобразование Фурье, где параметры преобразования известны априори. Для большинства алгоритмов на основе алгебры кватернионов гиперкомплексные переменные умножаются на постоянные коэффициенты, например Q-ПУБФ [7, 8]. Как было показано выше, в решетчатой структуре Q-ПУБФ (5) и (6) в операции умножения кватернионов (1) операнд x является входной переменной, а коэффициент q – кватернион-константой, компоненты которого определяют характеристики Q-ПУБФ и задают матрицу умножения. Оба операнда нормированы (норма равна единице), и их матрицы умножения различаются от заданных только масштабным коэффициентом. Для данных кватернионов операция умножения на сопряженный кватернион \bar{q} эквивалентна умножению на обратный кватернион:

$$q \cdot \bar{q} = q \cdot q^{-1} = 1 + (0i + 0j + 0k) \Leftrightarrow \mathbf{M}^{\pm}(q) \underbrace{\begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \end{bmatrix}^T}_{\bar{\mathbf{q}}^T} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T}_{\mathbf{e}_1^T}. \quad (27)$$

Матричное выражение (27) может интерпретироваться с различных точек зрения: результат умножения на \bar{q} есть кватернион, у которого мнимые части обнулены, что эквивалентно расположению вектора $\bar{\mathbf{q}}$ вдоль первой канонической оси при умножении на $\mathbf{M}^{\pm}(q)$. Таким образом, единичные кватернионы, у которых матрицы умножения ортонормальны, могут быть использованы для вывода CORDIC-алгоритма умножения кватернионов на основе факторизаций $\mathbf{M}^{\pm}(q)$. Факторизации $\mathbf{M}^{\pm}(q)$ имеют эффект, аналогичный умножению на вектор $\bar{\mathbf{q}}$, при следующих ограничениях:

– все сомножители в факторизации должны иметь структуру, аналогичную матрице $\mathbf{M}^{\pm}(q)$, т. е. произведение любого их множества представляет матрицу умножения кватернионов соответствующего типа:

$$\begin{aligned} \mathbf{M}^+(q_{k-1} \dots q_0) &= \mathbf{M}^+(q_{k-1}) \cdot \dots \cdot \mathbf{M}^+(q_0); \\ \mathbf{M}^-(q_0 \dots q_{k-1}) &= \mathbf{M}^-(q_{k-1}) \cdot \dots \cdot \mathbf{M}^-(q_0); \end{aligned} \quad (28)$$

– сомножители в факторизации (28) должны сходиться к единичной матрице 4×4 , \mathbf{I}_4 , т. е. возможная бесконечная серия сомножителей в факторизации может быть усечена до некоторой аппроксимации исходной матрицы при допустимой ошибке;

– сомножители в факторизации должны быть «простыми» матрицами, умножение на которые можно реализовать с использованием нескольких операций сдвига и сложения.

2.2. 4D-CORDIC-алгоритм умножения кватернионов (4D-CONST-Q-CORDIC) на базе 2D-вращений в 4D-гиперплоскости

Для уменьшения вычислительных ресурсов необходим поиск альтернативных Q-CORDIC-алгоритмов, которые просто проектируются. Предлагается конструировать 4D-CORDIC-алгоритм умножения кватернионов на базе 2D-вращений в 4D-гиперплоскости [21, 22], основываясь на факторизации (28) матрицы умножения кватернионов $\mathbf{M}^{\pm}(q)$, когда единичный кватернион q представляется как произведение $q = d_k d_j d_i$ трех выродившихся единичных кватернионов d_i, d_j, d_k , у которых соответственно одна мнимая часть i, j, k :

$$\begin{aligned} \mathbf{M}^+(q) &= \mathbf{M}^+(d_k) \cdot \mathbf{M}^+(d_j) \cdot \mathbf{M}^+(d_i); \\ \mathbf{M}^-(q) &= \mathbf{M}^-(d_i) \cdot \mathbf{M}^-(d_j) \cdot \mathbf{M}^-(d_k). \end{aligned} \quad (29)$$

Матрицы умножения вырожденных кватернионов $\mathbf{M}^\pm(d_i)$, $\mathbf{M}^\pm(d_j)$ и $\mathbf{M}^\pm(d_k)$ разреженные, каждая из них описывает 2D-вращение в различных плоскостях 4D-гиперплоскости на одинаковый угол, но, возможно, в противоположных направлениях:

$$\begin{aligned}\mathbf{M}^\pm(d_i) &= \text{diag}(\mathbf{R}_2(\varphi_i), \mathbf{R}_2(\pm\varphi_i)); \\ \mathbf{M}^\pm(d_j) &= \mathbf{P}_{1324} \text{diag}(\mathbf{R}_2(\varphi_j), \mathbf{R}_2(\mp\varphi_j)) \mathbf{P}_{1324}^T; \\ \mathbf{M}^\pm(d_k) &= \mathbf{P}_{1423} \text{diag}(\mathbf{R}_2(\varphi_k), \mathbf{R}_2(\pm\varphi_k)) \mathbf{P}_{1423}^T,\end{aligned}\quad (30)$$

где $\mathbf{R}_2(\varphi)$ – матрица вращения Гивенса (7); \mathbf{P}_{1324} и \mathbf{P}_{1423} – матрицы перестановок строк и столбцов в матрице умножения кватернионов в соответствии с индексами. Умножение на каждую разреженную матрицу умножения вырожденного кватерниона равносильно выполнению вращения на одинаковые углы, которое может осуществляться параллельно на двух 2D-CORDIC-блоках для фиксированного угла вращения φ . Поэтому итерации плотно связанных 2D-CORDIC-алгоритмов идентифицируются с одной итерацией 4D-CORDIC-алгоритма. Изменение плоскостей 2D-вращений или переназначение входов и выходов 2D-CORDIC-блоков эквивалентно переключению 4D-CORDIC-алгоритмов среди трех итерационных процессов, описываемых следующими разреженными матрицами умножения вырожденных кватернионов:

$$\mathbf{M}^\pm(\zeta(\tau(i), \sigma_1(i), 0, 0)) \approx \begin{bmatrix} 1 & -\sigma_1(i)2^{-\tau(i)} & 0 & 0 \\ \sigma_1(i)2^{-\tau(i)} & 1 & 0 & 0 \\ 0 & 0 & 1 & \pm\sigma_1(i)2^{-\tau(i)} \\ 0 & 0 & \pm\sigma_1(i)2^{-\tau(i)} & 1 \end{bmatrix}; \quad (31)$$

$$\mathbf{M}^\pm(\zeta(\tau(i), 0, \sigma_2(i), 0)) \approx \begin{bmatrix} 1 & 0 & -\sigma_2(i)2^{-\tau(i)} & 0 \\ 0 & 1 & 0 & \pm\sigma_2(i)2^{-\tau(i)} \\ \sigma_2(i)2^{-\tau(i)} & 0 & 1 & 0 \\ 0 & \pm\sigma_2(i)2^{-\tau(i)} & 0 & 1 \end{bmatrix}; \quad (32)$$

$$\mathbf{M}^\pm(\zeta(\tau(i), 0, 0, \sigma_3(i))) \approx \begin{bmatrix} 1 & 0 & 0 & -\sigma_3(i)2^{-\tau(i)} \\ 0 & 1 & \pm\sigma_3(i)2^{-\tau(i)} & 0 \\ 0 & \pm\sigma_3(i)2^{-\tau(i)} & 1 & 0 \\ \sigma_3(i)2^{-\tau(i)} & 0 & 0 & 1 \end{bmatrix}. \quad (33)$$

Переходя от размерности пространства 4D к 2D и обратно, можно аппроксимировать умножение кватернионов дискретными вырожденными кватернионами, у которых действительные части равны единице, а ненулевые мнимые части представляются отрицательной степенью по основанию два.

В факторизациях (29), (30) выбор углов $\varphi_i, \varphi_j, \varphi_k$ определяет соответствующую тройку вырожденных единичных кватернионов d_i, d_j, d_k , умножение которых на сопряженный кватернион \bar{q} обнуляет мнимые части результирующего произведения, т. е. $d_i \cdot d_j \cdot d_k \cdot \bar{q} = q \cdot \bar{q} = 1 + 0i + 0j + 0k$. Обнуление мнимых частей произведения на основе CORDIC-алгоритма предполагает решение задачи аппроксимации данного процесса с помощью дискретных вырожденных кватернионов (31), (33), каждый из которых выбирается следующим образом:

$$\begin{aligned}d_i &= \cos(\varphi_i) + i \sin(\varphi_i), \quad \varphi_i = \arctg(q_2/q_1) \rightarrow d_i \bar{q} = 0i + \dots; \\ d_j &= \cos(\varphi_j) + j \sin(\varphi_j), \quad \varphi_j = \arctg(q_3/q_1) \rightarrow d_j \bar{q} = \dots + 0j + \dots; \\ d_k &= \cos(\varphi_k) + j \sin(\varphi_k), \quad \varphi_k = \arctg(q_4/q_1) \rightarrow d_k \bar{q} = \dots + 0k.\end{aligned}$$

После умножения на сопряженный кватернион мнимые части могут отличаться от нуля, тогда данный процесс повторяется для другой тройки дискретных вырожденных кватернионов и так до тех пор, пока мнимые части не будут обнулены или по крайней мере их значения не будут меньше некоторой заданной величины. Это объясняется тем, что преобразование ортогональное и не модифицирует норму входных кватернионов. Если одна из мнимых частей увеличивается, то другая мнимая часть должна уменьшаться, так как сумма квадратов мнимых частей единичного кватерниона равна единице минус квадрат действительной части. Таким образом, последовательность умножений на дискретные вырожденные кватернионы путем увеличения реальной части неявно уменьшает значения мнимых частей, даже если их максимальная амплитуда временно увеличивается. Обнуление мнимых частей может осуществляться последовательно, начиная с мнимой части по i , однако фактически необходимо на каждой итерации выбирать для обнуления мнимую часть, у которой наибольшая величина. Хотя оба подхода сходятся, но у второго сходимость процесса выше [21]. Максимум точности операции умножения на кватернион-константу достигается за некоторое число итераций на полученном наборе управляющих параметров $\tau(i)$ и $\sigma(i)$. Значения мнимых компонент по модулю вектора $\mathbf{x}(i)$ операции CORDIC на сопряженный кватернион-константу \bar{q} близки к нулю. Псевдокод программы алгоритма расчета операторов сдвига $\tau(i)$ и направления вращения $\sigma(i)$ для 4D CONST Q-CORDIC-алгоритма умножения кватернионов показан ниже.

Алгоритм 1 расчета параметров управления схемы 4D-CONST-Q-CORDIC умножения кватернионов (кватернион q – константа, т. е. угол вращения фиксированный)

Вход: \bar{q} – сопряженный кватернион-константа;

ε – точность CORDIC-операции.

Выход: $\tau(i)$ – неотрицательное целое число сдвигов на i -й итерации;

$\sigma(i) \in \pm 1$ – направление вращения на i -й итерации.

Начало:

1. Установить: $i = 0$ и $\mathbf{x}(i) = \bar{q}$.

2. Определить индекс $m(i)$ наибольшей мнимой части i -й итерации: $x_{m(i)}(i) = \max_{2 \leq l \leq 4} |x_l(i)|$.

3. Определить значение параметра направления вращения: $\sigma(i) = -\text{sign}(x_{m(i)}(i))$.

4. Вычислить параметр числа сдвигов: $\tau(i) = \text{round}(\log_2 |x_{m(i)}(i)|)$.

5. Сохранить $m(i)$, $\sigma(i)$ и $\tau(i)$ как параметры i -й итерации.

6. Установить $\zeta = 1 + 0i + 0j + 0k$ и $\zeta_{m(i)} = \sigma(i)2^{\tau(i)}$.

7. Выполнить итерацию операции CORDIC, чтобы вычислить $\mathbf{x}(i+1) = \mathbf{M}^+(\zeta)\mathbf{x}(i)$.

8. Нормализовать результат итерации: $\mathbf{x}(i+1) = \mathbf{x}(i+1) / |\zeta|$.

9. Если $\forall_{l=2,4} |x_l(i+1)| \leq \varepsilon$, то переход на *Конец*, иначе $i = i + 1$ и переход к п. 2.

Конец.

В табл. 1 приведен пример расчета параметров управления схемы 4D CONST Q-CORDIC-умножения на кватернион-константу $q = [4, -1, 3, -2] / \sqrt{30}$ для точности CORDIC-операции $\varepsilon < 10^{-3}$.

Таблица 1

Пример расчета коэффициентов для кватерниона $q = [4, -1, 3, -2] / \sqrt{30}$

i	$m(i)$	$\tau(i)$	$\sigma(i)$	β_2	β_1	Значение вектора $\mathbf{x}(i)$			
0	–	–	–	–	–	0,730297	0,182574	–0,547723	0,365148
1	2	1	+1	1	0	0,903696	0,387298	0,129099	0,129099
2	1	2	–1	0	1	0,981495	–0,057735	0,173205	0,057735
3	2	3	–1	1	0	0,994199	–0,070014	–0,070014	0,042008
4	2	4	+1	1	0	0,995206	0,053842	–0,074684	0,033000

Окончание табл. 1

i	$m(i)$	$\tau(i)$	$\sigma(i)$	β_2	β_1	Значение вектора $\mathbf{x}(i)$			
5	1	4	+1	0	1	0,996784	0,057519	0,049333	0,026067
6	1	5	-1	0	1	0,998431	-0,004770	0,050863	0,022939
7	3	5	-1	1	1	0,999659	-0,006192	-0,011517	0,022596
8	2	5	-1	1	0	0,999877	-0,006549	-0,011318	-0,008639
9	2	7	+1	1	0	0,999932	-0,006683	0,004305	-0,008535
10	3	7	+1	1	1	0,999943	-0,006749	0,004200	0,007088
11	3	8	-1	1	1	0,999968	-0,006716	0,004253	-0,000724
12	2	8	-1	1	0	0,999990	0,001096	0,004258	-0,000691
13	2	9	+1	1	0	0,999993	0,001101	-0,003554	-0,000682
14	1	10	-1	0	1	0,999999	0,001099	0,000352	-0,000687
15	1	11	+1	0	1	0,999999	-0,000855	0,000351	-0,000687

В предложенном многомерном CORDIC-алгоритме умножение кватернионов микровращения соответствует умножению вырожденных кватернионов, у которых только одна мнимая часть отличается от нуля и может быть разной для разных CORDIC-итераций. Вместе с тем параметры знака и сдвига итераций выбираются таким образом, чтобы итерации могли пропускаться или повторяться. Из табл. 1 видно, что сходимость алгоритма достигается. Исследования проводились на множестве 10^6 случайно сгенерированных единичных кватернионов, точность CORDIC-операции задавалась в диапазоне от 10^{-2} до 10^{-6} . Среднее число итераций колебалось от 9 до 28, а максимальное число итераций – от 15 до 41. Масштабный множитель здесь меньше, чем у 4D-Q-CORDIC-алгоритма, и при равном количестве итераций для каждой разреженной матрицы определяется как $1/K = \left[\prod_{i=0}^{N-1} 1/\sqrt{1+3 \cdot 2^{-2\tau(i)}} \right]^3$.

2.3. Схема микровращений 4D-CONST-Q-CORDIC-алгоритма

Схема микровращений 4D-CONST-Q-CORDIC-алгоритма умножения кватернионов (рис. 1, а) может быть реализована с помощью четырех двухвходовых сумматоров и схемы коммутации ICN (interconnection network). Схема коммутации ICN применяется для выбора ненулевой мнимой части для текущей итерации. Центральное место схемы 4D-CONST-Q-CORDIC-алгоритма занимает схема коммутации ICN (рис. 1, б). Исходя из структуры разреженных матриц умножения вырожденных кватернионов (31)–(33), достаточно четырех вариантов перестановок данных и можно воспользоваться упрощенной сетью Бенесса [23]. Выбирая режимы работы переключающих схем 2×2 между «прямо» и «накрест» (рис. 1, в), можно получить четыре возможных потока перестановки данных (рис. 2).

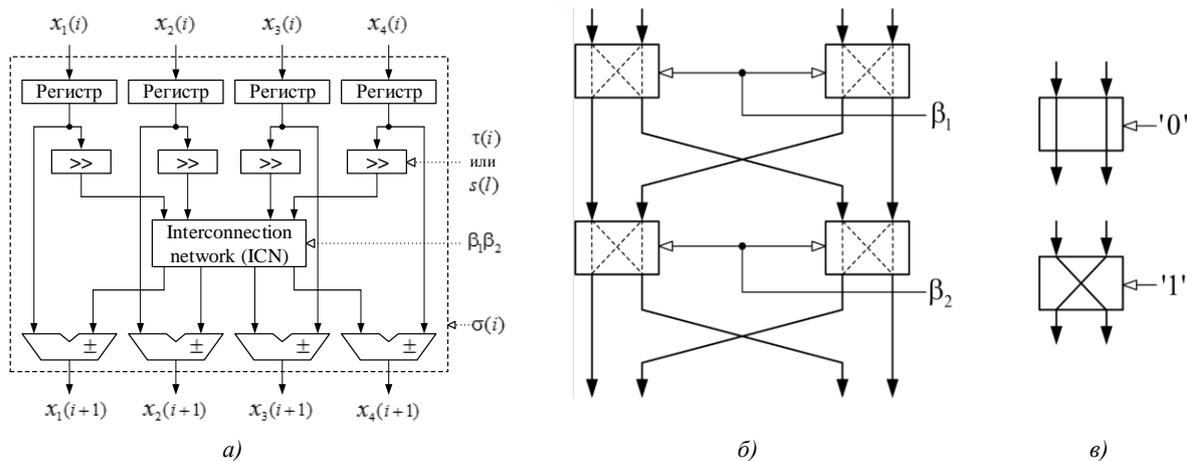


Рис. 1. 4D-CONST-Q-CORDIC-умножения кватернионов: а) схема микровращений 4D-CONST-Q-CORDIC-алгоритма; б) схема коммутации ICN для перестановки данных на основе сети Бенесса; в) состояния переключающих схем 2×2

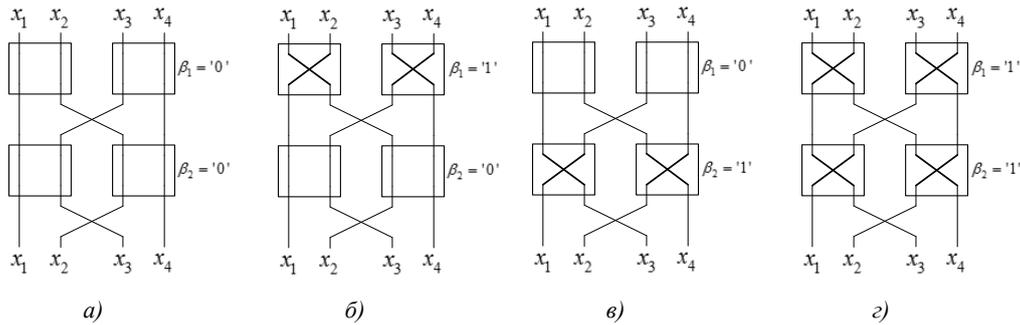


Рис. 2. Возможные варианты перестановки компонентов входного вектора $\mathbf{x}(i)$ на схеме коммутации ICN для перестановки данных: а) путь передачи данных без изменения их порядка; б) операции с мнимой частью i ; в) операции с мнимой частью j ; з) операции с мнимой частью k

Экспериментальные исследования 4D-CONST-Q-CORDIC-алгоритма проводились для аппаратной реализации схемы микровращений на FPGA XC6VLX240T (Xilinx Virtex-6). Для 16-разрядных данных задействовано следующее количество ресурсов FPGA: LUT6-FF – 189, LUT – 105, FF – 81, максимальная частота задающего генератора составляет 315 МГц. Кроме того, если в ICN выбирать путь передачи данных без изменения их порядка (см. рис. 2, а), то схема микровращений 4D-CONST-Q-CORDIC-алгоритма умножения кватернионов может быть использована для реализации этапа масштабирования, который представляется как умножение на вырожденный кватернион с ненулевой действительной компонентой:

$$\mathbf{M}^{\pm}(1 - 2^{-s(l)} + 0i + 0j + 0k) = \begin{bmatrix} 1 - 2^{-s(l)} & 0 & 0 & 0 \\ 0 & 1 - 2^{-s(l)} & 0 & 0 \\ 0 & 0 & 1 - 2^{-s(l)} & 0 \\ 0 & 0 & 0 & 1 - 2^{-s(l)} \end{bmatrix}.$$

Здесь $1 - 2^{-s(l)}$ – аппроксимация масштабного множителя l -й итерации процесса масштабирования, расчет которого приведен в подразд. 3.3. Цикл работы (рис. 3) состоит из последовательно выполняемых шагов итерационного (16 шагов) и масштабирующего (7 шагов) процессов, выбор которых определяется настройкой схемы ICN(β_1, β_2) (значение кода $i_beta(1:0)$). Направление поворота определяет управляющий сигнал $i_sigma(\sigma(i))$, а количество сдвигов – $i_shift(\tau(i))$. Переключение схемы коммутации ICN для выбора соответствующей разреженной матрицы умножения вырожденного кватерниона задает значение кода $i_beta(\beta_1, \beta_2)$. Результат операции микровращения формируется на выходах $i_data0(x_1(i+1))$, $i_data1(x_2(i+1))$, $i_data2(x_3(i+1))$, $i_data3(x_4(i+1))$. Как следует из временной диаграммы, скорость выполнения операции умножения кватернионов обратно пропорциональна суммарному времени итерационного и масштабирующего процессов.

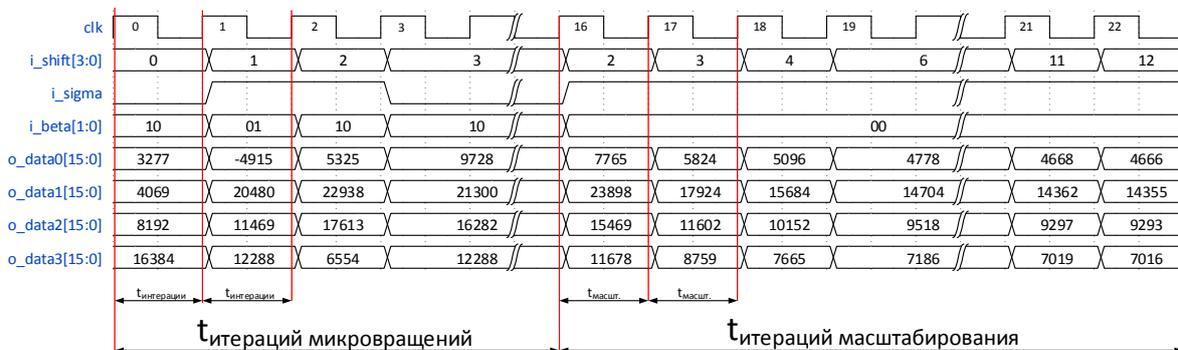


Рис. 3. Временные диаграммы работы схемы микровращений 4D-CONST-Q-CORDIC-алгоритма умножения на кватернион $q = [4, -1, 3, -2] / \sqrt{30}$

3. Целочисленный умножитель на кватернион-константу на основе встроенного 2D-CORDIC-модуля

3.1. Блочная лестничная схемная параметризация оператора умножения кватернионов

Ограничение динамического диапазона арифметики с фиксированной запятой приводит к тому, что в (5) результаты умножений округляются и, следовательно, реализовать обратное преобразование (6) – умножение на сопряженный кватернион – в арифметике с фиксированной запятой не представляется возможным. В результате получается только аппроксимация входной величины. Это является недопустимым для таких приложений, например, как компрессия изображений по схеме L2L (lossless-to-lossy, сжатие и восстановление изображений как без потерь, так и с контролируемым внесением артефактов). Данный недостаток может быть исправлен, если перейти к лестничной схемной параметризации оператора умножения кватернионов, которая позволяет выполнять обратимые «целое к целому» отображения, используя арифметику с фиксированной запятой [13]. Вычислительная сложность операции составит 12 умножений на действительные числа. Однако в настоящей работе не берется во внимание количество операций округления результатов умножений в лестничной структуре, рассматриваемых как белый шум. Число операций округления влияет на компактность энергии в субполосах Q-ПУБФ. Следовательно, количество операций округления должно быть уменьшено в максимально возможной степени для приложений кодирования без потерь. Структура обратимого преобразования на основе блочной лестничной схемной параметризации (рис. 4) представляет собой специальный класс структур обратимых преобразователей с лестничной схемной параметризацией. Количество источников шума здесь сокращается за счет объединения многих операций округления [24].

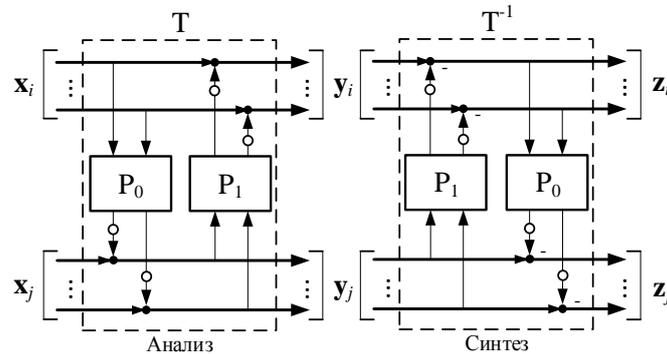


Рис. 4. Структура обратимого преобразователя на основе блочной лестничной схемной параметризации (черные и белые кружки обозначают операции суммирования и округления соответственно)

Векторы \mathbf{x}_i и \mathbf{x}_j на входе и на выходе \mathbf{y}_i и \mathbf{y}_j прямого преобразования, а также векторы \mathbf{z}_i и \mathbf{z}_j на выходе обратного преобразования и блоки преобразования – матрицы \mathbf{P}_0 и \mathbf{P}_1 – соотносятся следующим образом: $\mathbf{y}_j = \mathbf{x}_j + \text{round}(\mathbf{P}_0 \mathbf{x}_i)$, $\mathbf{y}_i = \mathbf{x}_j + \text{round}(\mathbf{P}_1 \mathbf{y}_j)$; $\mathbf{z}_i = \mathbf{y}_i - \text{round}(\mathbf{P}_1 \mathbf{y}_j) = \mathbf{x}_i$, $\mathbf{z}_j = \mathbf{y}_j - \text{round}(\mathbf{P}_0 \mathbf{y}_i) = \mathbf{x}_j$, где $\text{round}(\cdot)$ – оператор округления. В данном случае матрицы преобразования и их инверсные матрицы определяются как

$$\begin{bmatrix} \mathbf{y}_i \\ \mathbf{y}_j \end{bmatrix} = \mathbf{T} \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{bmatrix}, \begin{bmatrix} \mathbf{z}_i \\ \mathbf{z}_j \end{bmatrix} = \mathbf{T}^{-1} \begin{bmatrix} \mathbf{y}_i \\ \mathbf{y}_j \end{bmatrix} = \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{bmatrix},$$

$$\text{где } \mathbf{T} = \mathbf{P}_U \mathbf{P}_L, \mathbf{T}^{-1} = \mathbf{P}_L^{-1} \mathbf{P}_U^{-1}, \mathbf{P}_U = \begin{bmatrix} \mathbf{I} & \mathbf{P}_1 \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \mathbf{P}_U^{-1} = \begin{bmatrix} \mathbf{I} & -\mathbf{P}_1 \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \mathbf{P}_L = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{P}_0 & \mathbf{I} \end{bmatrix}, \mathbf{P}_L^{-1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{P}_0 & \mathbf{I} \end{bmatrix}.$$

Таким образом, следуя философии блочной лестничной схемной параметризации, матрицу оператора левого умножения кватернионов $\mathbf{M}^+(q)$ можно представить в виде блочной матрицы:

$$\mathbf{M}^+(q) = \begin{bmatrix} \mathbf{C}(q) & -\mathbf{S}(q) \\ \mathbf{S}(q) & \mathbf{C}(q) \end{bmatrix}, \quad \mathbf{C}(q) = \begin{bmatrix} q_1 & -q_2 \\ q_2 & q_1 \end{bmatrix}, \quad \mathbf{S}(q) = \begin{bmatrix} q_3 & q_4 \\ q_4 & -q_3 \end{bmatrix}.$$

Далее на основе известной [25] лестничной схемной факторизации двумерной матрицы вращения

$$\mathbf{R}_2(\varphi) = \begin{bmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{bmatrix} = \begin{bmatrix} 1 & \cos \varphi - 1 / \sin \varphi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \sin \varphi & 1 \end{bmatrix} \begin{bmatrix} 1 & \cos \varphi - 1 / \sin \varphi \\ 0 & 1 \end{bmatrix}$$

можно получить факторизацию матрицы умножения кватернионов

$$\mathbf{M}^+(q) = \underbrace{\begin{bmatrix} \mathbf{I}_2 & \mathbf{F}(q) \\ 0 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{U}(q)} \underbrace{\begin{bmatrix} \mathbf{I}_2 & 0 \\ \mathbf{G}(q) & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{L}(q)} \underbrace{\begin{bmatrix} \mathbf{I}_2 & \mathbf{H}(q) \\ 0 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{V}(q)}. \quad (34)$$

Для заданного коэффициента q и матрицы умножения $\mathbf{M}^+(q)$ определяется набор матричных выражений, которые могут быть решены однозначно для $\mathbf{F}(q)$, $\mathbf{G}(q)$ и $\mathbf{H}(q)$ при условии, что матрица $\mathbf{S}(q)$ является несингулярной или, более определенно, ненулевой:

$$\mathbf{F}(q) = (\mathbf{C}(q) - \mathbf{I}_2)\mathbf{S}(q)^{-1}, \quad \mathbf{G}(q) = \mathbf{S}(q), \quad \mathbf{H}(q) = \mathbf{S}(q)^{-1}(\mathbf{C}(q) - \mathbf{I}_2). \quad (35)$$

Элементы данных матриц являются вещественными коэффициентами лестничной схемной параметризации.

В случае если в прямом преобразовании Q-ПУБФ используется умножение на нормированный кватернион ($|q|=1$), обратное преобразование основывается на умножении на сопряженный кватернион \bar{q} . Из равенства $\mathbf{M}^\pm(\bar{q}) = \mathbf{M}^\mp(q)^T$ следует, что $\mathbf{M}^+(\bar{q}) = \mathbf{V}(q)^T \mathbf{L}(q)^T \mathbf{U}(q)^T$, но здесь в отличие от факторизации (34) нижние треугольные матрицы заменяются на верхние треугольные матрицы, и наоборот. Однако, принимая во внимание, что $\mathbf{M}^+(\bar{q}) = -\mathbf{P}\mathbf{M}^+(q)\mathbf{P}$, где $\mathbf{P} = \Gamma_4 \mathbf{J}_4$ и $\mathbf{P}^2 = -\mathbf{I}_4$, факторизация оператора умножения на сопряженный кватернион имеет вид

$$\mathbf{M}^+(\bar{q}) = \underbrace{(-\mathbf{P})\mathbf{V}(q)^T \mathbf{P}}_{\mathbf{U}(\bar{q})} \underbrace{(-\mathbf{P})\mathbf{L}(q)^T \mathbf{P}}_{\mathbf{L}(\bar{q})} \underbrace{(-\mathbf{P})\mathbf{U}(q)^T \mathbf{P}}_{\mathbf{V}(\bar{q})}$$

и упрощается:

$$\mathbf{M}^+(\bar{q}) = \pm \underbrace{\begin{bmatrix} \mathbf{I}_2 & -\mathbf{H}(q) \\ 0 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{U}(\bar{q})} \underbrace{\begin{bmatrix} \mathbf{I}_2 & 0 \\ -\mathbf{G}(q) & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{L}(\bar{q})} \underbrace{\begin{bmatrix} \mathbf{I}_2 & -\mathbf{F}(q) \\ 0 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{V}(\bar{q})}. \quad (36)$$

Из уравнений (5) и (6) факторизации полифазных передаточных матриц анализа $\mathbf{E}(z)$ и синтеза $\mathbf{D}(z)$ решетчатой структуры Q-ПУБФ следует, что необходимо переключение между операторами левого и правого умножений, а не только между оригинальным и сопряженным кватернионами. Принимая во внимание тот факт, что оператор сопряжения кватернионов есть собственно оператор получения обратного кватерниона, можно записать

$$\mathbf{M}^{-}(q) = \underbrace{\mathbf{D}_C \mathbf{U}(\bar{q}) \mathbf{D}_C}_{\mathbf{U}^{-}(q)} \underbrace{\mathbf{D}_C \mathbf{L}(\bar{q}) \mathbf{D}_C}_{\mathbf{L}^{-}(q)} \underbrace{\mathbf{D}_C \mathbf{V}(\bar{q}) \mathbf{D}_C}_{\mathbf{V}^{-}(q)}$$

или в форме блочной лестничной схемной параметризации

$$\mathbf{M}^{-}(q) = \underbrace{\begin{bmatrix} \mathbf{I}_2 & \Gamma_2 \mathbf{H}(q) \\ 0 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{U}^{-}(q)} \underbrace{\begin{bmatrix} \mathbf{I}_2 & 0 \\ \mathbf{G}(q) \Gamma_2 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{L}^{-}(q)} \underbrace{\begin{bmatrix} \mathbf{I}_2 & \Gamma_2 \mathbf{F}(q) \\ 0 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{V}^{-}(q)}.$$

Далее, представляя матрицу $\mathbf{M}^{-}(q)$ в виде блочной матрицы

$$\mathbf{M}^{-}(q) = \begin{bmatrix} \mathbf{C}(q) & -\bar{\mathbf{S}}(q) \\ \bar{\mathbf{S}}(q) & \mathbf{C}(q) \end{bmatrix}, \quad \bar{\mathbf{S}}(q) = \mathbf{S}(q) \text{diag}(1, -1) = \begin{bmatrix} q_3 & -q_4 \\ q_4 & q_3 \end{bmatrix},$$

можно получить ту же схему умножителя кватернионов для правого оператора умножения, что и для левого оператора.

Структура обратимого оператора умножения кватернионов на основе блочной лестничной схемной параметризации (рис. 5) представляет собой каскадное соединение блочных матриц $\mathbf{H}(q)$, $\mathbf{G}(q)$ и $\mathbf{F}(q)$ (прямое преобразование) и их инверсное включение с отрицательным знаком (обратное преобразование). Здесь белые кружки обозначают выходы, результат которых округлен.

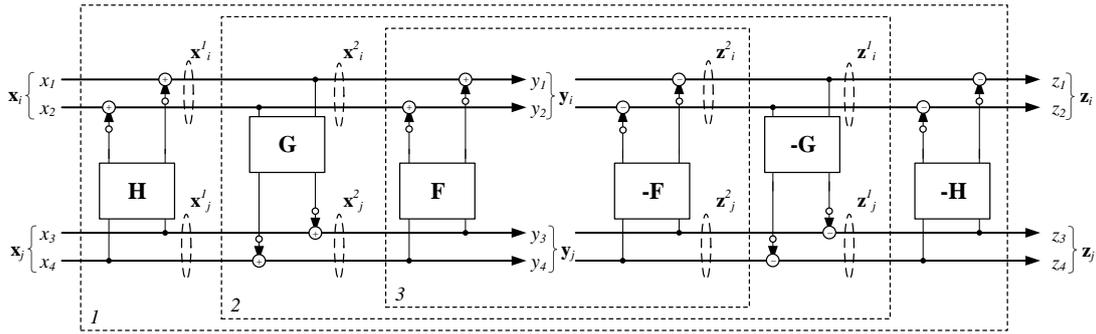


Рис. 5. Структура обратимого оператора умножения кватернионов

В результате прямого преобразования входной вектор $\mathbf{x} = [x_i, x_j]^T$ проходит три стадии обработки для формирования результата преобразования (умножения вектора \mathbf{x} на коэффициент q) $\mathbf{y} = [y_i, y_j]^T$:

$$\mathbf{x}_1 = \begin{bmatrix} \mathbf{I}_2 & \mathbf{H} \\ 0 & \mathbf{I}_2 \end{bmatrix} \mathbf{x} = \begin{bmatrix} x_i + \text{round}[\mathbf{H}x_j] \\ x_j \end{bmatrix} = \begin{bmatrix} x_i^1 \\ x_j^1 \end{bmatrix};$$

$$\mathbf{x}_2 = \begin{bmatrix} \mathbf{I}_2 & 0 \\ \mathbf{G} & \mathbf{I}_2 \end{bmatrix} \mathbf{x}_1 = \begin{bmatrix} x_i^1 \\ \text{round}[\mathbf{G}x_i^1] + x_j^1 \end{bmatrix} = \begin{bmatrix} x_i^2 \\ x_j^2 \end{bmatrix};$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{F} \\ 0 & \mathbf{I}_2 \end{bmatrix} \mathbf{x}_2 = \begin{bmatrix} x_i^2 + \text{round}[\mathbf{F}x_j^2] \\ x_j^2 \end{bmatrix} = \begin{bmatrix} y_i \\ y_j \end{bmatrix}.$$

Выходной результат обратного преобразования $\mathbf{z} = [z_i z_j]^T$ получается также после трех ступенной обработки результата прямого преобразования $\mathbf{y} = [y_i y_j]^T$:

$$\begin{aligned} \mathbf{z}_2 &= \begin{bmatrix} \mathbf{I}_2 & -\mathbf{F} \\ 0 & \mathbf{I}_2 \end{bmatrix} \mathbf{y} = \begin{bmatrix} x_i^1 + \text{round}[\mathbf{F}x_j^2] - \text{round}[\mathbf{F}x_j^2] \\ x_j^2 \end{bmatrix} = \begin{bmatrix} x_i^1 \\ x_j^2 \end{bmatrix} = \mathbf{x}_2; \\ \mathbf{z}_1 &= \begin{bmatrix} \mathbf{I}_2 & 0 \\ -\mathbf{G} & \mathbf{I}_2 \end{bmatrix} \mathbf{z}_2 = \begin{bmatrix} x_i^1 \\ -\text{round}[\mathbf{G}x_i^1] + \text{round}[\mathbf{G}x_i^1] + x_j^2 \end{bmatrix} = \begin{bmatrix} x_i^1 \\ x_j^2 \end{bmatrix} = \mathbf{x}_1; \\ \mathbf{z} &= \begin{bmatrix} \mathbf{I}_2 & -\mathbf{H} \\ 0 & \mathbf{I}_2 \end{bmatrix} \mathbf{z}_1 = \begin{bmatrix} x_i + \text{round}[\mathbf{H}x_j] - \text{round}[\mathbf{H}x_j] \\ x_j \end{bmatrix} = \begin{bmatrix} x_i \\ x_j \end{bmatrix} = \mathbf{x}. \end{aligned}$$

Таким образом, умножение на $1/q$ или на эквивалентный сопряженный кватернион \bar{q} реализуется переупорядочиванием коэффициентов в обратном порядке и изменением их знака. Следует отметить, что округление лестничных коэффициентов и результата соответствующего умножения не влияет на обратимость преобразования. Следовательно, выполнение обратимого «целое к целому» отображения на основе оператора умножения возможно для арифметики с фиксированной запятой.

3.2. Прямая и обратная схемы множителя кватернионов на основе блочной лестничной схемной параметризации со встроенным 2D-CORDIC-модулем

Величины углов в пределах $-\pi \leq \varphi < \pi$, $-\pi/2 \leq \psi \leq \pi/2$ и $-\pi/2 \leq \chi \leq \pi/2$ достаточно, чтобы описать любой кватернион заданного модуля $|q|$ и трех углов φ , ψ , χ в полярной форме: $q = |q|e^{i\varphi}e^{j\psi}e^{k\chi}$, где $q_1 = |q|\cos\varphi$, $q_2 = |q|\sin\varphi\cos\psi$, $q_3 = |q|\sin\varphi\sin\psi\cos\chi$, $q_4 = |q|\sin\varphi\sin\psi\sin\chi$. Для $|q|=1$ расчетные выражения блочной лестничной схемной параметризации (35) оператора умножения кватернионов могут быть представлены следующим образом:

$$\begin{aligned} f_{11}(q) &= -f_{22}(q) = [\cos\chi(\cos\varphi - 1) - \sin\varphi\cos\psi\sin\chi]/d, \\ f_{12}(q) &= f_{21}(q) = [\sin\chi(\cos\varphi - 1) + \sin\varphi\cos\psi\cos\chi]/d, \\ g_{11}(q) &= -g_{22}(q) = d\cos\chi, \\ g_{12}(q) &= g_{21}(q) = d\sin\chi, \\ h_{11}(q) &= -h_{22}(q) = [\cos\chi(\cos\varphi - 1) + \sin\varphi\cos\psi\sin\chi]/d, \\ h_{12}(q) &= h_{21}(q) = [\sin\chi(\cos\varphi - 1) - \sin\varphi\cos\psi\cos\chi]/d, \\ d &= \sin\varphi\sin\psi. \end{aligned} \tag{37}$$

Анализ (37) показывает, что в компонентах $\mathbf{F}(q)$, $\mathbf{G}(q)$ и $\mathbf{H}(q)$ факторизации матрицы умножения кватернионов $\mathbf{M}^+(q)$ (34) есть пары коэффициентов с одинаковым модулем, что не очевидно из (35) и не принималось во внимание, когда создавалась факторизация (34). Следовательно, структуры матриц $\mathbf{F}(q)$, $\mathbf{G}(q)$ и $\mathbf{H}(q)$ имеют структуру, близкую к матрице $\mathbf{R}_2(\varphi)$ вращения Гивенса $\begin{bmatrix} s & c \\ c & -s \end{bmatrix} = \mathbf{J}_2 \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$, где через c и s обозначены косинус и синус соответствующего угла. Например, матрица $\mathbf{F}(q)$ имеет вид

$$\begin{bmatrix} f_{11}(q) & f_{12}(q) \\ f_{12}(q) & -f_{11}(q) \end{bmatrix} = \mathbf{J}_2 \begin{bmatrix} f_{12}(q) & -f_{11}(q) \\ f_{11}(q) & f_{12}(q) \end{bmatrix}.$$

Таким образом, данные матрицы могут быть аппроксимированы с использованием 2D-CORDIC-алгоритма, а зависимости (37) являются расчетными выражениями блочной лестничной схемной параметризации оператора умножения кватернионов и формируют матрицы вращения Гивенса для соответствующих этапов разложения $\mathbf{U}(q)$, $\mathbf{L}(q)$, $\mathbf{V}(q)$.

Динамический диапазон схемы умножителя кватернионов (рис. 6) ограничен диапазоном $-1..1$, что обусловлено форматом фиксированной запятой. Однако коэффициенты, полученные с помощью расчетных выражений блочной лестничной схемной параметризации (37) оператора умножения кватернионов, могут иметь абсолютные значения больше единицы. Приведение параметров умножителя к требуемому динамическому диапазону может быть достигнуто заменой умножения кватерниона q на кватернион \tilde{q} , который представляет собой версию кватерниона q с переставленными и (или) измененными знаками компонент. Матрицы оператора умножения кватернионов $\mathbf{M}^{\pm}(q)$ и $\mathbf{M}^{\pm}(\tilde{q})$ будут соотноситься согласно следующему выражению:

$$\mathbf{M}^{\pm}(q) = \mathbf{P}_{\text{post}} \mathbf{M}^{\pm}(\tilde{q}) \mathbf{P}_{\text{pre}}, \quad (38)$$

где матрицы \mathbf{P}_{pre} и \mathbf{P}_{post} являются матрицами перестановок исходного кватерниона q и кватерниона \tilde{q} с переставленными и (или) измененными знаками компонент кватерниона q . На практике это соответствует лишь выполнению пред- и постобработки входного операнда \mathbf{x} и выходного результата соответственно: $qx = \mathbf{M}^{\pm}(q) \mathbf{x} = \mathbf{P}_{\text{post}} \mathbf{M}^{\pm}(\tilde{q}) \mathbf{P}_{\text{pre}} \mathbf{x}$.

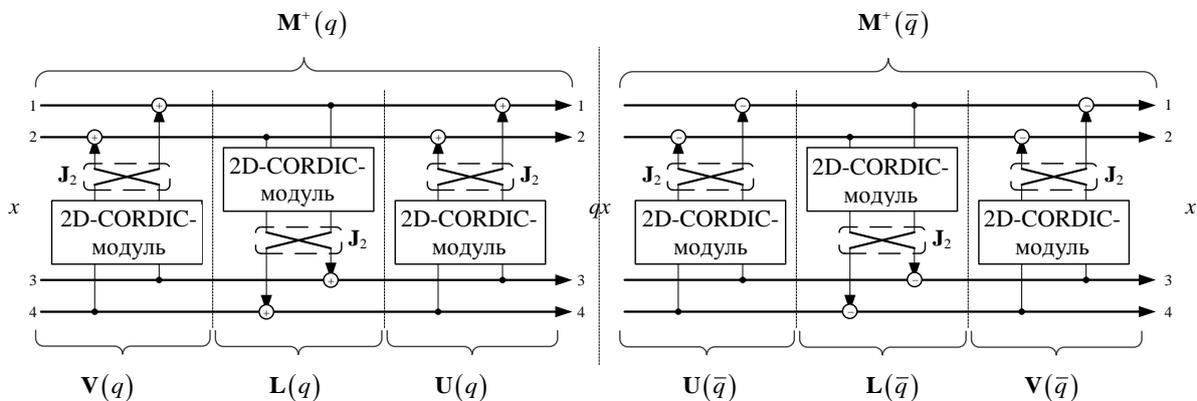


Рис. 6. Прямая и обратная схемы умножителя кватернионов на основе блочной лестничной параметризации со встроенным 2D-CORDIC-модулем

Таким образом, гибридная схема «CORDIC – лестничная параметризация» позволяет интегрировать CORDIC-алгоритм «внутри» лестничной схемы умножителя, заменив действительные умножения на микроповороты CORDIC-алгоритма: сложение и сдвиг, а также получить совершенное преобразование (ПУБФ) и в аппаратной реализации [26]. Неотъемлемым свойством блочных лестничных схем является то, что все результаты между этапами факторизации могут быть вычислены сразу, без использования вспомогательной памяти. Небольшая разница в схемах для прямого и обратного умножения (умножения на сопряженный кватернион) дает возможность проектировать универсальный умножитель. Отличие лишь в знаке для коэффициентов прямого и обратного преобразования, что позволяет иметь одну схему с переключаемыми режимами. Еще одно преимущество предложенной схемы умножителя кватернионов заключается в том, что число операций округления уменьшилось почти в два раза по сравнению со структурой умножителя на основе лестничной схемной параметризации [27].

3.3. Оптимизация параметров встроенного 2D-CONST-CORDIC-модуля

Задача выбора параметров схемы умножителя кватернионов со встроенным 2D-CORDIC-алгоритмом – наиболее сложная часть проекта. Как отмечалось выше, при фиксированном угле вращения φ с целью достижения заданной точности за минимальное число итераций M параметры управления для итерационного и масштабирующего процессов CORDIC-алгоритма могут быть определены и оптимизированы заранее во время проектирования схемы CORDIC-алгоритма:

– фиксированный угол φ предопределяет некоторое множество базисных углов CORDIC-алгоритма $\{\varphi(i) = \arctan(2^{-\tau(i)}) \text{ для } 0 \leq i \leq M-1\}$. Для того чтобы повернуть вектор $[x_1, x_2]^T$ на угол φ для получения вектора $[x_1^{out}, x_2^{out}]^T$, CORDIC-алгоритм (8)–(15) может быть модифицирован следующим образом:

$$\begin{bmatrix} x_1(i+1) \\ x_2(i+1) \end{bmatrix} = \begin{bmatrix} 1 & -\sigma(i)2^{-\tau(i)} \\ \sigma(i)2^{-\tau(i)} & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1(i) \\ x_2(i) \end{bmatrix}, \quad \begin{bmatrix} x_1^{out} \\ x_2^{out} \end{bmatrix} \leftarrow \frac{1}{K} \begin{bmatrix} x_1(M) \\ x_2(M) \end{bmatrix}; \quad (39)$$

– масштабный фактор $1/K$ сейчас зависит от множества базисных углов $\{\varphi(i)\}$, а точность алгоритма определяет, как близко фактический угол вращения φ_A , полученный за M микровращений, аппроксимирует заданный угол вращения φ . Это, в свою очередь, определяет отклонение фактического вектора вращения от требуемого.

Произвольное значение масштабирующего фактора $1/K = \prod_{i=0}^{M-1} (1 + 2^{-2\tau(i)})^{-1/2}$ при $1/K < 1$ может быть аппроксимировано выражением [14]

$$\frac{1}{K} \approx \prod_{l=0}^{L-1} [1 - 2^{-s(l)}], \quad (40)$$

где L и $s(l) \in \mathbb{N} \setminus \{0\}$ выбраны как продукт аппроксимации $1/K$, а процесс масштабирования описывается рекурсивным выражением

$$\mathbf{x}(l+1) = \mathbf{x}(l) - 2^{-s(l)} \mathbf{x}(l), \quad l = 0, 1, \dots, L-1, \quad \mathbf{x}(0) = \mathbf{x}(M), \quad \mathbf{x}(L) = \mathbf{x}_{out}(M). \quad (41)$$

Таким образом, для фиксированного угла вращения параметры управления для итерационного и масштабирующего процессов CORDIC-алгоритма могут быть оптимизированы за два шага:

1. Приведение угла к заданным ограничениям. Преобразования (8), (9) или (39), (40) применимы только для вращений на углы, для которых $\cos \varphi \geq 0$ и $|\cos \varphi| \geq |\sin \varphi|$, т. е. на углы $|\varphi| \leq \frac{\pi}{4}$ [27]. Так, для осуществления вращения на произвольный угол φ , не удовлетворяющий данным условиям, всегда могут быть реализованы простой пред- и постобработкой матрицы вращений \mathbf{R}_2 с подходящим углом $\tilde{\varphi}$ [14]:

$$\mathbf{R}_2(\varphi) = \mathbf{P}_{post}^{CORDIC} \mathbf{R}_2(\tilde{\varphi}) \mathbf{P}_{pre}^{CORDIC}, \quad (42)$$

в частности

$$\mathbf{R}_2(\varphi) = \mathbf{\Gamma}_2 \mathbf{R}_2(-\varphi) \mathbf{\Gamma}_2, \quad \mathbf{R}_2(\varphi) = \mathbf{J}_2 \mathbf{R}_2(-\varphi) \mathbf{\Gamma}_2, \quad \mathbf{R}_2(\varphi) = -\mathbf{J}_2 \mathbf{R}_2(\pi - \varphi) \mathbf{J}_2, \quad (43)$$

т. е. достаточно поменять местами величины и (или) их знаки. Здесь \mathbf{J}_2 – обратная единичная матрица 2×2 , а $\mathbf{\Gamma}_2 = \text{diag}(1, -1)$.

2. Минимизация числа базисных углов в множестве $\{\varphi(i)\}$ для достижения требуемой точности вычисления преобразования: определяются параметры $\tau(i)$, M и $s(l)$, L .

Простой псевдокод программы алгоритма расчета операторов сдвига $\tau(i)$ и направления вращения $\sigma(i)$ для схемы 2D-CONST-Q-CORDIC при фиксированном угле вращения показан

ниже. Заданная максимальная точность ε_ϕ определяется как максимальная ошибка Δ_ϕ между матрицей вращения \mathbf{M} , определенной заданным углом – входным параметром $\mathbf{q} = [q_1, q_2]^T$, и ее аппроксимацией $\mathbf{IR}_2(i)$. Алгоритм оптимизации поиска параметров $\tau(i)$ и $\sigma(i)$ базируется на минимизации целевой функции $\max|\Delta_\phi|$. Последовательность матриц приближения $\mathbf{IR}_2(i)$, $i = 0, 1, 2, \dots$, определяется CORDIC-операцией, для которой матрица вращения формируется из параметров управления $\tau(i)$ и $\sigma(i)$, рассчитанных на i -й итерации. Для $i = 0$ существует единичная матрица вращения вектора, лежащего на оси абсцисс, вокруг центра декартовых координат на угол, определенный вектором $\mathbf{q} = [q_1, q_2]^T$ ($\text{Re } q = q_1, \text{Im } q = q_2$). При этом входной вектор $\mathbf{x}(i) = [x_1(i), x_2(i)]^T$ равен комплексно-сопряженному вектору $\bar{\mathbf{q}} = [q_1, -q_2]^T$ ($|\mathbf{q}| = |\bar{\mathbf{q}}|$).

Алгоритм 2 расчета параметров управления $\tau(i)$ и $\sigma(i)$ 2D-CONST-CORDIC-схемы

Вход: $\mathbf{q} = [q_1, q_2]^T$ ($\text{Re } q = q_1, \text{Im } q = q_2$) – комплексное число-константа, т. е. угол вращения фиксированный;

ε_ϕ – точность аппроксимации заданного угла вращения.

Выход: $\tau(i)$ – неотрицательное целое число сдвигов на i -й итерации;

$\sigma(i) \in \pm 1$ – направление вращения на i -й итерации.

Начало:

1. Сформировать матрицы вращения $\mathbf{M} = \begin{bmatrix} q_1 & -q_2 \\ q_2 & q_1 \end{bmatrix}$ и единичную $\mathbf{IR}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

2. Установить: $i = 0$, $\mathbf{x}(i) = \bar{\mathbf{q}}$, $\mathbf{x}(i) = [x_1(i), x_2(i)]^T$ и $\mathbf{IR}_2(i) = \mathbf{IR}_2$.

3. Определить значение параметра направления вращения $\sigma(i) = -\text{sign}[x_2(i)]$.

4. Вычислить параметр числа сдвигов $\tau(i) = \text{round}(\log_2 |x_2(i)/x_1(i)|)$.

5. Сохранить $\sigma(i)$ и $\tau(i)$ как параметры i -й итерации.

6. Сформировать $\chi = 1 + j\sigma(i)2^{\tau(i)}$, $j = \sqrt{-1}$.

7. Выполнить итерацию аппроксимации: $\mathbf{IR}_2(i+1) = \mathbf{R}_2(\chi)\mathbf{IR}_2(i)$, где $\mathbf{R}_2(\chi)$ – матрица вращения (9).

8. Нормализовать результат итерации: $\mathbf{IR}_2(i+1) = \mathbf{IR}_2(i+1) / |\chi|$.

9. Вычислить ошибку аппроксимации $\Delta_\phi = \mathbf{IR}_2(i+1) - \mathbf{M}$.

10. Если $\max|\Delta_\phi| \leq \varepsilon_\phi$, то $M = i$, иначе выполнить итерацию CORDIC-операции $\mathbf{x}(i+1) = \mathbf{R}_2(\chi)\mathbf{x}(i)$ и $i = i + 1$, перейти к п. 3.

Конец.

Алгоритм 3 описывает задачу оптимизации поиска параметров L и $s(l)$ процесса масштабирования 2D-CONST-CORDIC-схемы. Масштабный множитель, обусловленный выполнением итерационного процесса с параметрами $\tau(i)$ и $\sigma(i)$, которые получены в результате выполнения алгоритма 2, вычисляется как $1/K = \prod_{i=0}^{M-1} (1 + 2^{-2\tau(i)})^{-1/2}$. Целевая функция ΔK определяет, на сколько отношение масштабного множителя $1/K$ к его аппроксимированному значению $(1/K)_A$ отличается от единицы, т. е. $\Delta K = |(1/K)/(1/K)_A - 1|$. Алгоритм начинает работу с вычисления аппроксимированного значения масштабного множителя для одного сдвига, а затем число масштабирующих сдвигов увеличивается на единицу до тех пор, пока ΔK не станет больше заданной точности ε_K операции масштабирования. Величина ε_K равна величине

не ε_ϕ в алгоритме 2, потому что ошибки ΔK и Δ_ϕ вносят равный вклад в общую ошибку 2D-CONST-CORDIC-схемы.

Алгоритм 3 расчета параметров L и $s(l)$ процесса масштабирования

Вход: $1/K = \prod_{i=0}^{M-1} (1 + 2^{-2\tau(i)})^{-1/2}$ – масштабный множитель;

ε_K – точность операции масштабирования.

Выход: L и $s(l) \in \mathbb{N} \setminus \{0\}$ – неотрицательное целое число сдвигов на l -й итерации процесса масштабирования.

Начало:

1. Установить $A=1, k=1, l=0$.
2. Вычислить $(1/K)_A = A(1-2^{-k})$.
3. Если $(1/K)_A \geq (1/K)$, то сохранить число сдвигов на l -й итерации процесса масштабирования $s(l) = k$ и $A = (1/K)_A$, иначе $k = k+1$ и переход к п. 2.
4. Оценить точность аппроксимации $\Delta K = |(1/K)/(1/K)_A - 1|$.
5. Если $\Delta K \leq \varepsilon_K$, то $L = k$, иначе переход к п. 2.

Конец.

В отличие от стандартного CORDIC-алгоритма в операции масштабирования параметры $\tau(i)$, M и L представляются произвольными значениями. Например, выбор количества сдвигов $\tau(i) \in \mathbb{N}$ не зависит от номера итерации i , но при этом может быть получена требуемая точность за меньшее число микровращений [28].

Анализ схем на рис. 7 показывает, что итерационный и масштабирующий процессы могут быть реализованы на одной коммутируемой во времени CORDIC-схеме. При этом затраты оборудования будут минимальны, а производительность данного процессора будет обратно пропорциональна сумме числа микровращений и итераций процесса масштабирования. С целью увеличения скорости работы 2D-CORDIC-процессора вычисления (39) и (41) могут осуществляться по конвейерной CORDIC-схеме [15].

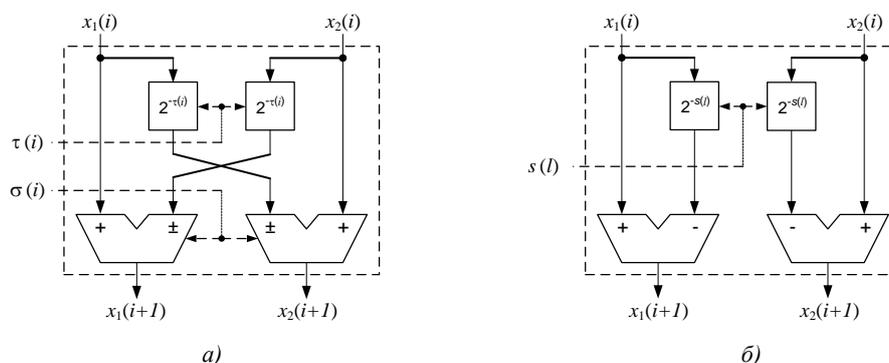


Рис. 7. Схемы для реализации: а) микровращений; б) итераций масштабирования 2D-CORDIC-алгоритма для априори известного угла

Процесс оптимизации параметров встроенной 2D-CONST-CORDIC-схемы выполняется в три этапа:

- 1) для кватерниона q рассматриваются все модификации гиперкомплексного числа, потенциально определяющие вычислительную схему умножителя, в которой все коэффициенты лестничной схемы находятся в диапазоне $-1...1$;
- 2) для каждого такого модифицированного кватерниона \tilde{q} осуществляется формирование матриц $\mathbf{P}_{pre}^{CORDIC}$ и $\mathbf{P}_{post}^{CORDIC}$ (42)–(43), рассчитываются параметры и оценивается число итераций 2D-CONST-CORDIC-алгоритма (число микровращений итерационного процесса и количе-

ство итераций процесса масштабирования), гарантирующее заданную точность операции умножения на кватернион-константу;

3) выбирается соответствующий кандидат из множества модифицированных кватернионов \tilde{q} , для которого параметры схемы наиболее близко согласуются с принятой платформой реализации (число итераций должно быть минимальным). Это обеспечивает максимальную скорость выполнения операции умножения.

Например, для кватерниона $q = q_1 + q_2i + q_3j + q_4k = 1/\sqrt{30}(4 - i + 3j - 2k)$ процесс настройки параметров умножителя на основе 2D-CONST-CORDIC-модуля выглядит следующим образом. На первом этапе для обеспечения заданного динамического диапазона из 24 возможных модификаций исходного кватерниона $q = q_1 + q_2i + q_3j + q_4k$ только у семи кватернионов \tilde{q} , представляющих собой версии кватерниона q с переставленными и (или) измененными знаками компонент, коэффициенты блочной лестничной схемной параметризации (37) попадают в динамический диапазон $-1 \dots 1$. У модифицированного кватерниона $\tilde{q} = q_3 + q_2i + q_1j + q_4k$ коэффициенты блочной лестничной факторизации (37) следующие: $f_{11}(q) = -f_{22}(q) = -0,395$, $f_{12}(q) = f_{21}(q) = 0,488$, $g_{11}(q) = -g_{22}(q) = 0,730$, $g_{12}(q) = g_{21}(q) = 0,365$, $h_{11}(q) = -h_{22}(q) = -0,595$, $h_{12}(q) = h_{21}(q) = 0,048$. Далее, на втором и третьем этапах процесса оптимизации параметров встроенного 2D-CONST-CORDIC-алгоритма расчет числа итераций (алгоритмы 2 и 3) 2D-CORDIC-модуля для каждого этапа разложения $\mathbf{U}(q)$, $\mathbf{L}(q)$, $\mathbf{V}(q)$ схемы умножителя кватернионов (см. рис. 6) показал, что у данного модифицированного кватерниона $\tilde{q} = q_3 + q_2i + q_1j + q_4k$ для точности ε_ϕ и ε_k порядка 10^{-5} требуется минимальное число микровращений итерационного процесса 7 ($M_u = 3$, $M_L = 1$, $M_V = 3$) и итераций процесса масштабирования 17 ($L_u = 7$, $L_L = 5$, $L_v = 5$). В табл. 2 для кватерниона \tilde{q} приведены параметры управления $\tau(i)$ и $\sigma(i)$, а также L и $s(l)$ процесса масштабирования для каждой компоненты разложения $\mathbf{U}(q)$, $\mathbf{L}(q)$, $\mathbf{V}(q)$. Для реализации умножения на кватернион q через умножение на кватернион \tilde{q} должна быть выполнена следующая пред- и постобработка (38), представленная в виде матриц перестановок:

$$\mathbf{P}_{\text{pre}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}, \quad \mathbf{P}_{\text{post}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Таблица 2

Параметры управления 2D-CONST-CORDIC-модулями схемы умножения кватернионов на основе блочной лестничной параметризации

Параметр управления	Компоненты разложения		
	\mathbf{U} $M=3, L=7$	\mathbf{L} $M=1, L=5$	\mathbf{V} $M=3, L=5$
$\sigma(0)$	1	-1	1
$\tau(0)$	0	1	4
$\sigma(1)$	-1		1
$\tau(1)$	4		6
$\sigma(2)$	1		1
$\tau(2)$	11		9
$s(0)$	1	2	2
$s(1)$	3	6	3
$s(2)$	5	7	4

$s(3)$	8	9	6
$s(4)$	10	10	6
$s(5)$	12		
$s(6)$	13		
$\mathbf{P}_{pre}^{CORDIC}$	$-\Gamma_2$	\mathbf{I}_2	$-\mathbf{I}_2$
$\mathbf{P}_{post}^{CORDIC}$	\mathbf{J}_2	\mathbf{I}_2	\mathbf{I}_2

Рассмотрим структурную схему операции микровращения 2D-CONST-CORDIC-алгоритма (рис. 8). Ядро схемы образуют два сумматора, работающих в режиме суммирования и вычитания, два регистра сдвига вправо и модуль коммутации ICN, основное назначение которого заключается в переключении выходов регистров сдвига в соответствии с режимами работы 2D-CONST-CORDIC-схемы: итерационным процессом (выходы коммутируются накрест) и масштабирующим процессом (выходы коммутируются прямо). Компоненты $x_1(i)$ и $x_2(i)$ загружаются во входные регистры и передаются в сумматоры согласно схеме. Содержимое сумматоров $x_1(i+1)$, $x_2(i+1)$ загружается во входные регистры для следующей CORDIC-итерации. Для повышения производительности 2D-CORDIC-модуля в схеме умножителя кватернионов на основе блочной лестничной параметризации можно использовать конвейерное включение 2D-CONST-CORDIC-схем микровращения. При этом латентность операции будет определяться числом схем микровращения в конвейере и производительность модуля возрастет во столько же раз. Прототипирование схемы осуществлялось на ПЛИС с архитектурой FPGA XC6VLX240T (Xilinx Virtex-6).

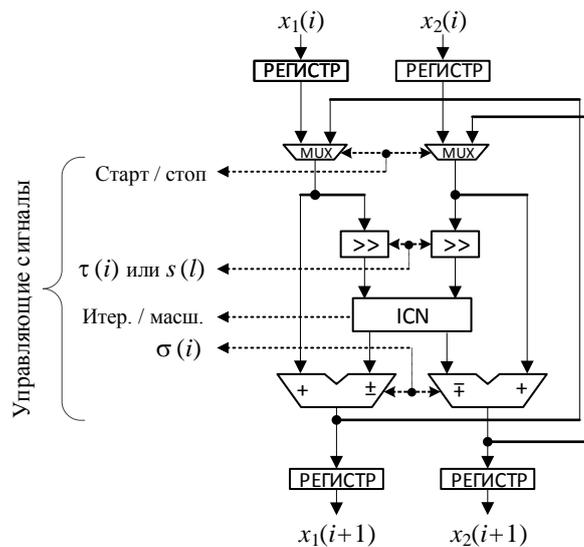


Рис. 8. Структурная 2D-CONST-CORDIC-схема операции микровращения

Аппаратные затраты умножителя кватернионов 4D-CONST-Q-CORDIC (см. рис. 2) приблизительно на 50 % меньше, чем у умножителя кватернионов с блочной лестничной структурой на основе 2D CONST CORDIC (см. рис. 6). Это обусловлено дополнительными аппаратными затратами на встраивание 2D-CONST-CORDIC-модуля в схему умножителя для динамического диапазона обработки данных в пределах диапазона $-1...1$ и вращения на произвольный угол (реализации матриц перестановок (38) и (43)). Однако только схема умножения кватернионов на основе блочной лестничной параметризации со встроенными 2D-CONST-CORDIC-модулями дает возможность реализации умножителя кватернионов как обратимого оператора умножения кватернионов в целочисленной арифметике. Данную схему можно рассматривать как структурное решение для цифровой обработки медиаданных без потерь, когда данные и алгоритмы представляются в алгебре кватернионов. В работе [28] предложен кон-

вейерный процессор умножителя кватернионов с блочной лестничной параметризацией на основе 2D-CONST-CORDIC-процессора, производительность которого выше, чем у 4D-CONST-Q-CORDIC-процессора, который по определению не допускает организации поточной обработки данных. Сбалансированного решения между производительностью умножителя, ядром которого является 2D-CONST-CORDIC, и аппаратными затратами, как показано в [29], можно достигнуть в схеме рекурсивного процессора умножителя кватернионов.

Моделирование в программной среде MATLAB с помощью инструментального модуля Fixed Point системы анализа-синтеза с линейной фазочастотной характеристикой четырехканального Q-ПУБФ на основе умножителя кватернионов из конвейера 2D-CONST-CORDIC-модулей осуществлялось для следующих параметров банка фильтров: эффективность субполосного кодирования $CG = 8,41$ дБ; ослабление в полосе непропускания более 21дБ; уровень постоянной составляющей не менее -331 дБ, т. е. отсутствует эффект DC-leakage; 2D-CORDIC-арифметика 16-разрядная, погрешность сходимости менее 10^{-5} .

Анализ результатов моделирования показывает, что амплитудно-частотная характеристика системы анализа-синтеза Q-ПУБФ представляет собой прямолинейную функцию, т. е. анализ-синтез Q-ПУБФ со встроенными 2D-CONST-CORDIC-модулями обладает свойством совершенной реконструкции входных данных и вносит только задержку. Например, у реализации решетчатой структуры Q-ПУБФ на основе планарных оборотов Гивенса амплитудно-частотная характеристика системы анализа-синтеза соответствует нелинейной функции с максимальной неравномерностью $\pm 0,2$ дБ. Таким образом, Q-ПУБФ на основе 2D-CONST-CORDIC-схем представляет собой преобразователь «целое к целому» и является системой без потерь.

Заключение

В работе сформулирована задача оптимизации схемных решений CORDIC-алгоритма для фиксированного угла поворота как для 2D-пространства, так и многомерного (4D) евклидова вращения. Результатом решения задачи является оптимизированное множество микровращений для фиксированного и известного углов поворота. При этом итерации микровращения и масштабирования выполняются последовательно друг за другом на одном аппаратном модуле. Эффективные CORDIC-схемы получаются благодаря тому, что параметры сдвига представляются нелинейной функцией числа сдвигов входных операндов операции микровращения в отличие от стандартного CORDIC-алгоритма, где параметры сдвига являются последовательностью однобитных сдвигов. В предложенной структуре параметры схемы подбираются каждый раз под соответствующий кватернион-константу, на который выполняется умножение, т. е. для каждого постоянного сомножителя-кватерниона константы получается разное количество сдвигов и разные масштабные множители, а также свой набор операторов направления вращения, которые должны храниться в памяти. Это цена за оптимизацию итерационного процесса и общая проблема к подходам построения более гибких CORDIC-алгоритмов [14, 28, 30].

Список литературы

1. Кантор, И.Л. Гиперкомплексные числа / И.Л. Кантор, А.С. Солодовников. – М. : Наука, 1973. – 145 с.
2. Parfieniuk, M. Quaternionic building block for paraunitary filter banks / M. Parfieniuk, A. Petrovsky // Proc. 12th European Signal Processing Conf. (EUSIPCO). – Vienna, Austria, 2004. – P. 1237–1240.
3. Choukroun, D. Novel quaternion kalman filter / D. Choukroun, I. Bar-Itzhack, Y. Oshman // IEEE Trans. Aerosp. Electron. Syst. – 2006. – Vol. 42, № 1. – P. 174–190.
4. Miron, S. Quaternion-music for vector-sensor array processing / S. Miron, N. Le Bihan, J. Mars // IEEE Trans. Signal Process. – 2006. – Vol. 54, № 4. – P. 1218–1229.
5. Hsiao, S.F. Parallel singular value decomposition of complex matrices using multidimensional CORDIC algorithms / S.F. Hsiao, J.M. Delosme // IEEE Trans. Signal Process. – 1996. – Vol. 44, № 3. – P. 685–697.

6. Sercov, V.V. Digital hypercomplex all-pass filters: A novel filters bank building block / V.V. Sercov, A.A. Petrovsky, D.V. Lushtyk // Proc. 6th Int. Workshop on Systems, Signals and Image Proc. (IWSSIP). – Bratislava, Slovakia, 1999. – P. 181–184.
7. Парфенюк, М. Параунитарные банки фильтров на основе алгебры кватернионов: теория и применение / М. Парфенюк, А.А. Петровский // Цифровая обработка сигналов. – 2008. – № 1. – С. 22–36.
8. Parfieniuk, M. Inherently lossless structures for eight- and six-channel linear-phase paraunitary filter banks based on quaternion multipliers / M. Parfieniuk, A. Petrovsky // Signal Process. – 2010. – Vol. 90. – P. 1755–1767.
9. Alexiadis, D. Estimation of motions in color image sequences using hypercomplex Fourier transforms / D. Alexiadis, G. Sergiadis // IEEE Trans. Image Process. – 2009. – Vol. 18, № 1. – P. 168–187.
10. Karney, C. Quaternions in molecular modeling / C. Karney // J. Molecular Graphics and Modelling. – 2007. – Vol. 25, № 5. – P. 595–604.
11. Vaidyanathan, P.P. Multirate Systems and Filter Banks / P.P. Vaidyanathan. – Englewood Cliffs : Prentice-Hall, 1993. – 911 p.
12. Howell, T.D. The complexity of the quaternion product: Tech. Rep. TR 75-245 / T.D. Howell, J.C. Lafon [Электронный ресурс]. – 1975. – Mode of access : <http://www.theworld.com/~sweetser/quaternions/ps/cornellcstr75-245.pdf>. – Date of access : 05.10.2015.
13. Parfieniuk, M. Quaternion multiplier inspired by the lifting implementation of plane rotations / M. Parfieniuk, A. Petrovsky // IEEE Trans. Circuits Syst. I. – 2010. – Vol. 57, № 10. – P. 2708–2717.
14. Parfieniuk, M. Rapid Prototyping of Quaternion Multiplier: From Matrix Notation to FPGA-Based Circuits / M. Parfieniuk, N. Petrovsky, A. Petrovsky // Rapid Prototyping Technology: Principles and Functional Requirements. – Vienna : InTech, 2011. – P. 227–246.
15. 50 years of CORDIC: Algorithms, architectures, and applications / P. Meher [et al] // IEEE Trans. Circuits Syst. I. – 2009. – Vol. 56, № 9. – P. 1893–1907.
16. Volder, J.E. The CORDIC trigonometric computing technique / J. E. Volder // IRE Trans. Electron. Comput. – 1959. – Vol. 8. – P. 330–334.
17. Walther, J.S. A unified algorithm for elementary functions / J.S. Walther // Proceedings of the Spring Joint Computer Conf. AFIPS '71 (Spring). – N. Y. : ACM, 1971. – P. 379–385.
18. Hsiao, S.-F. Householder CORDIC algorithms / S.-F. Hsiao, J.-M. Delosme // IEEE Trans. Comput. – 1995. – Vol. 44, № 8. – P. 990–1001.
19. Гантмахер, Ф.П. Теория матриц / Ф.П. Гантмахер; 5-е изд. – М. : Физматлит, 2004. – 560 с.
20. Hsiao, S.-F. Redundant constant-factor implementation of multi-dimensional CORDIC and its application to complex SVD / S.-F. Hsiao, C.-Y. Lau, J.-M. Delosme // IEEE Trans. VLSI Syst. – 2000. – Vol. 25, № 2. – P. 155–166.
21. Петровский, Н.А. Многомерный CORDIC алгоритм кватернионов с «разреженными» итерациями / Н.А. Петровский, М. Парфенюк // Труды 15-й Междунар. конф. «Цифровая обработка сигналов и ее применение» (DSPA'2013). – Т. 2. – М., 2013. – С. 206–210.
22. Петровский, Н.А. 4D-CORDIC арифметика для процессора параунитарного банка фильтров на основе алгебры кватернионов / Н.А. Петровский, М. Парфенюк, А.А. Петровский // Труды 40-й Междунар. науч. конф. «Вопросы оптимизации вычислений» (BOB-XL). – Ялта, 2013. – С. 214–215.
23. Benes, V.E. Mathematical theory of connecting networks and telephone traffic / V.E. Benes. – N. Y. : Academic Press, 1965. – 319 p.
24. Suzuki, T. Generalized block-lifting factorization of M-channel biorthogonal filter banks for lossy-to-lossless image coding / T. Suzuki, M. Ikehara, T. Nguyen // IEEE Trans. Image Process. – 2012. – Vol. 21, № 7. – P. 3220–3228.
25. Wavelet transforms that map integers to integers / A.R. Calderbank [et. al.] // Appl. Comput. Harmon. Anal. – 1998. – Vol. 5, № 3. – P. 332–369.

26. Petrovsky, N. Pipelined embedded processor of quaternionic m-band wavelets for image multiresolution analysis / N. Petrovsky, M. Parfieniuk, A. Petrovsky // 2nd Mediterranean Conf. on Embedded Computing (MECO). – Budva, Montenegro, 2013. – P. 196–199.
27. Оранский, А.М. Аппаратные методы в цифровой вычислительной технике / А.М. Оранский. – Минск : ЕСУ, 1977. – 208 с.
28. Petrovsky, N.A. The CORDIC-inside-lifting architecture for constant-coefficient hardware quaternion multipliers / N.A. Petrovsky, M. Parfieniuk // Proc. Intern. Conf. on Signals and Electronic Systems (ICSSES). – Wrocław, 2012. – P. 1–6.
29. Петровский, Н.А. Рекурсивный процессор умножителя кватернионов со структурной CORDIC-лестничной параметризацией / Н.А. Петровский, А.В. Станкевич // Труды Междунар. науч. конф. «Информационные технологии и системы» (ИТС'2013). – Минск : БГУИР, 2013. – С. 196–197.
30. Meher, P. CORDIC designs for fixed angle of rotation / P. Meher, S. Park // IEEE Trans. VLSI Syst. – 2013. – Vol. 21, № 2. – P. 217–228.

Поступила 05.10.2015

*Белорусский государственный университет
информатики и радиоэлектроники,
Минск, П. Бровки, 6
e-mail: nick@petrovsky.eu,
stankevich@bsuir.by,
palex@bsuir.by*

N.A. Petrovsky, A.V. Stankevich, A.A. Petrovsky

CORDIC TECHNIQUES FOR FIXED ANGLE OF ROTATION IN MULTIPLYING OPERATION OF QUATERNIONS

The article contains a number of solutions for the key element of paraunitary filter banks based on quaternionic algebra (Q-PUBF) – the multiplier of quaternions with usage of CORDIC (Coordinate Rotation Digital Computer) techniques for the fixed angle of rotation where, unlike known solutions, 4D rotation control parameters are represented by nonlinear function of shifts number of input operands of the microrotation operation. Suggested approach of the multiplier designing on a quaternion-constant allows reaching the maximum performance of the multiplier scheme with low use of resources, for example, of FPGA.

НАДЕЖНОСТЬ СИСТЕМ

УДК 004.3

Л.И. Кульбак, В.Б. Алюшкевич, С.А. Золотой

ОЦЕНКА НАДЕЖНОСТИ БОРТОВОЙ АППАРАТУРЫ МАЛЫХ КОСМИЧЕСКИХ АППАРАТОВ В ПРОЦЕССЕ ИХ ПОЛЕТА

Рассматривается методика оценки текущего состояния безотказности бортовой аппаратуры малых космических аппаратов в процессе их полета. Приводятся математическое представление показателя безотказности бортовой аппаратуры и модель структуры малого космического аппарата, а также формулы расчетов показателей безотказной работы блоков структурной схемы надежности бортовой аппаратуры.

Введение

Бортовая аппаратура (БА) космического аппарата (КА) как невосстанавливаемый объект характеризуется одним из свойств надежности – безотказностью в полете КА. Под безотказностью принято понимать способность объекта выполнять требуемые функции в заданных условиях [1, 2], в рассматриваемом случае – в условиях полета КА. Основным показателем безотказности невосстанавливаемых объектов – вероятность безотказной работы $P(t)$ за наработку времени t . В качестве дополнительных показателей используются средняя наработка до отказа $T_{0,с}$ и интенсивность отказов Λ [3].

Следует заметить, что для БА КА вводится показатель, аналогичный показателям долговечности, с названием «срок активного существования КА (САС)» – $T_{САС}$. Его отличительной чертой является то, что в качестве предельного состояния кроме отказа выступают и другие факторы, например ограничения по расходуемым ресурсам. Мерой оценки безотказности БА КА служит единственный показатель – вероятность безотказной работы в течение САС КА $P_{БА}(T_{САС})$.

Малые космические аппараты (МКА) имеют массу не более 500 кг [4]. К ним относятся белорусский (БКА) и российский (Канопус-В) космические аппараты дистанционного зондирования Земли. Несмотря на небольшие размеры МКА, в БА МКА широко применяется избыточность в виде резервирования малой кратности, как правило дублирование. Например, в БА БКА зарезервированы 23 объекта. При этом используются невосстанавливаемые нагруженный и ненагруженный резервы. Текущее значение безотказности МКА зависит от состояния резервных элементов БА. Очевидно, что чем больше резервных элементов БА сохранили работоспособность к текущему моменту времени, тем выше безотказность БА.

1. Математическое представление показателя безотказности БА МКА

Рассмотрим математические определения показателя безотказности БА МКА, используя данные из [3].

Примем следующие допущения:

- отказы элементов БА являются независимыми событиями;
- наработка до отказа элементов комплектующих БА имеет экспоненциальное распределение;
- интенсивность отказов переключающего устройства ненагруженного резерва имеет пренебрежимо малое значение.

Вероятность безотказной работы БА в интервале времени от 0 до $T_{САС}$ определяется по формуле

$$P_{БА}(T_{САС}) = P_{БА}(0; T_{САС}) = \exp\left(-\int_0^{T_{САС}} \lambda(x) dx\right), \quad (1)$$

где $\lambda(x)$ – интенсивность отказов БА;

$P(T_{\text{САС}})$ – вероятность того, что БА проработает безотказно в течение времени $T_{\text{САС}}$, начав работать в момент времени $t = 0$, или вероятность того, что случайное время работы БА до отказа окажется больше $T_{\text{САС}}$.

Вероятность безотказной работы БА в интервале времени от t до $T_{\text{САС}}$ выполняется по формуле

$$P_{\text{БА}}(t; T_{\text{САС}}) = P_{\text{БА}}(0; T_{\text{САС}} - t) = \frac{P_{\text{БА}}(0; T_{\text{САС}})}{P_{\text{БА}}(0; t)} = \frac{P_{\text{БА}}(T_{\text{САС}})}{P_{\text{БА}}(t)}, \quad (2)$$

где $P_{\text{БА}}(t; T_{\text{САС}})$ – вероятность того, что БА проработает безотказно в течение требуемого интервала времени $(T_{\text{САС}} - t)$, начав работать в момент времени t , или условная вероятность того, что случайное время работы БА до отказа окажется больше величины $(T_{\text{САС}} - t)$ при условии, что БА уже проработала безотказно в течение интервала времени t .

Если до момента времени t БА не отказала, то

$$P_{\text{БА}}(t; T_{\text{САС}}) = \exp\left(-\int_t^{T_{\text{САС}}} \lambda(x) dx\right). \quad (3)$$

Для структурной схемы надежности (СН) последовательного вида при произвольном законе наработки до отказа ее блоков вероятность безотказной работы БА определяется следующим образом:

$$P_{\text{БА}}(t; T_{\text{САС}}) = P_{\text{БА}}(0; T_{\text{САС}} - t) = \prod_{i=1}^n P_{\text{Б},i}(t; T_{\text{САС}}) = \frac{P_{\text{БА}}(t; T_{\text{САС}})}{P_{\text{БА}}(t)}, \quad (4)$$

где $P_{\text{Б},i}(t; T_{\text{САС}})$ – вероятность того, что i -й блок СН БА проработает безотказно в течение требуемого интервала времени $(T_{\text{САС}} - t)$, начав работать в момент времени t ; n – количество блоков в составе СН БА.

Для СН последовательного вида при экспоненциальном законе наработки до отказа ее блоков

$$P_{\text{БА}}(t; T_{\text{САС}}) = \exp[-\Lambda_{\text{БА}}(T_{\text{САС}} - t)]. \quad (5)$$

Здесь $\Lambda_{\text{БА}}$ – интенсивность отказов совокупности из n блоков СН БА:

$$\Lambda_{\text{БА}} = \sum_{i=1}^n \Lambda_{\text{Б},i}, \quad (6)$$

где $\Lambda_{\text{Б},i}$ – интенсивность отказов i -го блока СН БА.

2. Модель структуры МКА

Наиболее распространенным и наглядным методом расчета надежности технических систем следует считать метод СН, регламентированный ГОСТ Р 51891–2008 [5], который определяет методы построения модели надежности системы и использование этой модели для вычисления показателей ее надежности.

БА МКА представляет собой невосстанавливаемую в полете систему. Следовательно, для оценки ее надежности можно использовать метод СН – т. е. графического изображения работоспособного состояния системы, которое показывает логическую связь функционирующих компонентов, необходимых для успешной работы системы. СН может быть выполнена на различных уровнях разукрупнения системы (системы в целом, подсистемы, модуля или сборочных единиц) и формируется из блоков (элементов).

По структуре блоки СН БА можно разделить на простые, сложные и сверхсложные (рис. 1). К простым отнесем блоки, которые не имеют в своем составе резервируемых элементов, к сложным – блоки, которые их имеют. Сверхсложные блоки СН представляют собой резервируемые сложные блоки СН.



Рис. 1. Изображения блоков в ССН БА

3. Формулы расчетов показателей безотказности блоков ССН БА

Показателями безотказности блоков ССН БА являются вероятность безотказной работы блока $T_{\text{САС}}$ $P_B(t_{\text{н.САС}})$ и интенсивность отказов блока Λ_B за наработку:

$$t_{\text{н.САС}} = T_{\text{САС}} K_{\text{иЭ}}, \quad (7)$$

где $t_{\text{н.САС}}$ – наработка блока ССН в интервале $T_{\text{САС}}$; $K_{\text{иЭ}}$ – коэффициент интенсивности эксплуатации блока ССН БА в интервале $T_{\text{САС}}$ (доля времени работы блока в полете).

3.1. Простой блок ССН БА

Простой блок ССН БА (см. рис. 1, а) представляет собой один элемент или цепочку элементов, не имеющих резерва. Для такого блока ССН расчет производится по формулам

$$P_B(t_{\text{н.САС}}) = \exp(-\Lambda_B t_{\text{н.САС}}); \quad (8)$$

$$\Lambda_B = \sum_{i=1}^m \lambda_i, \quad (9)$$

где $P_B(t_{\text{н.САС}})$ – вероятность безотказной работы блока ССН БА за наработку в интервале $T_{\text{САС}}$; Λ_B – интенсивность отказов блока ССН БА; m – количество элементов в блоке ССН; λ_i – интенсивность отказов i -го элемента в блоке ССН БА.

3.2. Сложный блок ССН БА с постоянно включенным нагруженным резервом

Сложный блок ССН БА с постоянно включенным нагруженным резервом (см. рис. 1, б) состоит из двух компонентов: основного и резервного. В качестве компонента может использоваться как отдельный элемент, так и цепочка элементов БА. Критерием отказа такого блока является отказ обоих компонентов блока. Поскольку отказ одного компонента блока не сказыва-

ется на работоспособности второго компонента блока, вероятность отказа обоих компонентов блока равна произведению вероятностей отказов компонентов:

$$Q_{Б.Н}(t_{Н.САС}) = Q_{Э.1}(t_{Н.САС})Q_{Э.2}(t_{Н.САС}), \quad (10)$$

где $Q_{Б.Н}(t_{Н.САС})$ – вероятность отказа сложного блока ССН БА с нагруженным резервом за наработку $t_{Н.САС}$; $Q_{Э.1}(t_{Н.САС})$ – вероятность отказа первого компонента блока ССН БА за наработку $t_{Н.САС}$; $Q_{Э.2}(t_{Н.САС})$ – вероятность отказа второго компонента блока ССН БА за наработку $t_{Н.САС}$.

Вероятность безотказной работы блока ССН БА с нагруженным резервом следует вычислять по формуле

$$\begin{aligned} P_{Б.Н}(t_{Н.САС}) &= 1 - Q_{Б.Н}(t_{Н.САС}) = 1 - [(1 - P_{Э.1}(t_{Н.САС}))(1 - P_{Э.2}(t_{Н.САС}))] = \\ &= P_{Э.1}(t_{Н.САС}) + P_{Э.2}(t_{Н.САС}) - P_{Э.1}(t_{Н.САС}) P_{Э.2}(t_{Н.САС}), \end{aligned} \quad (11)$$

где $P_{Б.Н}(t_{Н.САС})$ – вероятность безотказной работы блока ССН БА с нагруженным резервом; $P_{Э.1}(t_{Н.САС})$ – вероятность безотказной работы компонента первого блока ССН; $P_{Э.2}(t_{Н.САС})$ – вероятность безотказной работы компонента второго блока ССН за наработку $t_{Н.САС}$.

Если основной и резервный компоненты равноценны по надежности, формула (11) примет вид

$$P_{Б.Н}(t_{Н.САС}) = 2 P_{Э}(t_{Н.САС}) - P_{Э}^2(t_{Н.САС}), \quad (12)$$

где $P_{Э}(t_{Н.САС})$ – вероятность безотказной работы компонента блока ССН за наработку $t_{Н.САС}$.

В случае экспоненциального распределения наработки до отказа компонентов блока ССН БА формула (12) примет вид

$$P_{Б.Н}(t_{Н.САС}) = 2 \exp(-\lambda_{Э} t_{Н.САС}) - \exp(-2\lambda_{Э} t_{Н.САС}) \approx 1 - (\lambda_{Э} t_{Н.САС})^2. \quad (13)$$

Интенсивность отказов в этом случае вычисляется по формуле [6]

$$\Lambda_{Б.Н}(t_{Н.САС}) = \frac{2\lambda_{Э} \exp(-\lambda_{Э} t_{Н.САС}) [1 - \exp(-\lambda_{Э} t_{Н.САС})]}{1 - [1 - \exp(-\lambda_{Э} t_{Н.САС})]^2} \approx \frac{2\lambda_{Э}^2 t_{Н.САС} (1 - \lambda_{Э} t_{Н.САС})}{1 - (\lambda_{Э} t_{Н.САС})^2} = \frac{2\lambda_{Э}^2 t_{Н.САС}}{1 + \lambda_{Э} t_{Н.САС}}. \quad (14)$$

3.3. Сложный блок ССН БА с резервированием путем замещения ненагруженным резервом

Сложный блок ССН БА с резервированием путем замещения ненагруженным резервом (см. рис. 1, в) состоит также из двух компонентов: основного и резервного. Критерием отказа такого блока является отказ обоих компонентов блока. Рассмотрим вариант, когда интенсивность отказов обоих компонентов равна постоянной величине. Так как отказ одного компонента блока не сказывается на работоспособности второго компонента, вероятности отказа обоих компонентов равны. Процедура работы резервируемого блока ССН такова, что работает один из компонентов блока, резервный компонент включается в работу только после отказа основного. Отказ резервируемого блока наступает после отказа резервного компонента, т. е. интервал работоспособного состояния блока ССН образуется из суммы интервала работоспособности основного и резервного компонентов. Интервалы работоспособности основного и резервного компонентов имеют экспоненциальное распределение.

В работе [7] потоком Эрланга k -го порядка называется поток, у которого интервалы между событиями представляют собой сумму $(k + 1)$ независимых случайных величин, распределенных одинаково по показательному закону с параметром λ , где λ – интенсивность исходного простейшего потока.

Согласно данному определению интервал безотказности блока ССН с ненагруженным резервом относится к категории потока Эрланга первого порядка ($k = 1$), характеристики которого вычисляются по следующим формулам:

– плотность распределения интервалов T в потоке Эрланга первого порядка

$$f_1(t) = \lambda_{\text{Э}}^2 \times t \times \exp(-\lambda_{\text{Э}} \times t), \quad t \geq 0; \quad (15)$$

– функция распределения случайных величин T

$$F_1(t) = \int_0^t f_1(t) dt = 1 - (1 + \lambda_{\text{Э}} t) \times \exp(-\lambda_{\text{Э}} t); \quad (16)$$

– вероятность безотказной работы блока ССН БА с ненагруженным резервом в соответствии с (1) и (16)

$$P_{\text{НН}}(t_{\text{Н.САС}}) = (1 + \lambda_{\text{Э}} t_{\text{Н.САС}}) \times \exp(-\lambda_{\text{Э}} t_{\text{Н.САС}}) \approx 1 - (\lambda_{\text{Э}} t_{\text{Н.САС}})^2; \quad (17)$$

– средняя наработка до отказа блока ССН согласно (14)

$$T_{\text{О.НН}} = \int_0^{\infty} (1 + \lambda_{\text{Э}} t) \times \exp(-\lambda_{\text{Э}} t) dt = \frac{1}{\lambda_{\text{Э}}} + \frac{\lambda_{\text{Э}}}{\lambda_{\text{Э}}^2} = \frac{2}{\lambda_{\text{Э}}}; \quad (18)$$

– интенсивность отказов блока ССН

$$\lambda_{\text{НН}}(t_{\text{Н.САС}}) = \frac{f(t_{\text{Н.САС}})}{P_{\text{НН}}(t_{\text{Н.САС}})} = \frac{\lambda_{\text{Э}}^2 t_{\text{Н.САС}}}{1 + \lambda_{\text{Э}} t_{\text{Н.САС}}} \approx \lambda_{\text{Э}}^2 t_{\text{Н.САС}}. \quad (19)$$

Формулы (13) – (19) справедливы тогда, когда параметры распределения компонентов блока ССН равны. В полете МКА случаются ситуации, когда при отказе резерва в блоке ССН создается резервируемый блок с компонентами, имеющими различные значения параметров. Такие ситуации возникают в сверхсложных блоках ССН. В этом случае необходим другой подход для вывода формул оценки показателей надежности блока ССН.

Известно, что при наличии двух независимых случайных величин x и y с плотностями распределения $f_1(x)$ и $f_2(y)$ случайная величина

$$z = x + y \quad (20)$$

имеет плотность распределения $f(z)$ [8], вычисляемую по формуле

$$f(z) = \int_{-\infty}^{+\infty} f_2(z-x)f_1(x)dx = \int_{-\infty}^{+\infty} f_1(z-y)f_2(y)dy. \quad (21)$$

В связи с тем что значения интервалов работоспособности больше нуля и значения x и y не больше z , формула (21) примет вид

$$f(z) = \int_0^z f_2(z-x)f_1(x)dx = \int_0^z f_1(z-y)f_2(y)dy. \quad (22)$$

Рассмотрим композицию экспоненциальных распределений при различных значениях параметров. Пусть оба компонента блока ССН имеют различные параметры распределения:

$$f(x_1) = \lambda_1 \exp(-\lambda_1 x_1); \quad (23)$$

$$f(x_2) = \lambda_2 \exp(-\lambda_2 x_2). \quad (24)$$

В этом случае согласно (22)

$$\begin{aligned} f(z) &= \int_0^z \lambda_2 \exp[-\lambda_2(z-x_1)] \lambda_1 \exp(-\lambda_1 x_1) dx_1 = \int_0^z \lambda_1 \lambda_2 \exp[-\lambda_2(z-x_1)] \exp(-\lambda_1 x_1) dx_1 = \\ &= \frac{\lambda_1 \lambda_2}{\lambda_2 - \lambda_1} [\exp(-\lambda_1 z) - \exp(-\lambda_2 z)] \approx \lambda_1 \lambda_2 z. \end{aligned} \quad (25)$$

Следовательно,

$$F(t) = \int_0^t \frac{\lambda_1 \lambda_2}{\lambda_2 - \lambda_1} [\exp(-\lambda_1 z) - \exp(-\lambda_2 z)] dz \approx 1 + \frac{1}{\lambda_2 - \lambda_1} [\lambda_1 \exp(-\lambda_2 t) - \lambda_2 \exp(-\lambda_1 t)]; \quad (26)$$

$$P(t) = 1 - F(t) = \frac{1}{\lambda_1 - \lambda_2} [\lambda_1 \exp(-\lambda_2 t) - \lambda_2 \exp(-\lambda_1 t)] \approx 1 - \frac{\lambda_1 \lambda_2 t^2}{2}; \quad (27)$$

$$T_{0.} = \int_0^{\infty} P(t) dt = \int_0^{\infty} \frac{1}{\lambda_1 - \lambda_2} [\lambda_1 \exp(-\lambda_2 t) - \lambda_2 \exp(-\lambda_1 t)] dt = \frac{\lambda_1 + \lambda_2}{\lambda_1 \lambda_2}; \quad (28)$$

$$\lambda(t) = \frac{f(t)}{P(t)} = \lambda_1 \lambda_2 \frac{[\exp(-\lambda_2 t) - \exp(-\lambda_1 t)]}{[\lambda_1 \exp(-\lambda_2 t) - \lambda_2 \exp(-\lambda_1 t)]} \approx \lambda_1 \lambda_2 t. \quad (29)$$

3.4. Сверхсложный блок ССН БА с резервированием путем замещения сложным ненагруженным блоком

Сверхсложный блок ССН БА (см. рис. 1, з) можно рассматривать как сложный блок ССН с резервированием ненагруженным замещением в виде цепочки из простого и сложного блоков с постоянным нагруженным резервом. Критерием отказа такого блока является отказ обеих цепочек блока. В отличие от сложного блока ССН с замещением ненагруженным резервом в этой структуре может создаваться ситуация, когда резервируемые цепочки имеют различные значения интенсивности отказов и вероятности безотказной работы. Такая ситуация создается, когда в резервируемой цепочке откажет один из резервируемых элементов.

В исходном состоянии, когда работоспособны все элементы сверхсложного блока ССН, вероятность безотказной работы сверхсложного блока ССН определяется по формуле

$$P_{\text{нн}}(t_{\text{н.с.а.с}}) = (1 + \lambda_{\text{сб}} t_{\text{н.с.а.с}}) \times \exp(-\lambda_{\text{сб}} t_{\text{н.с.а.с}}) \approx 1 - \frac{(\lambda_{\text{сб}} t_{\text{н.с.а.с}})^2}{2}, \quad (30)$$

где $\lambda_{\text{сб}}$ – интенсивность отказов резервируемой цепочки из структуры сверхсложного блока ССН:

$$\lambda_{\text{сб}} = \lambda_{\text{п}} + \frac{\lambda_{\text{э}}^2 t_{\text{н.с.а.с}}}{1 + \lambda_{\text{э}} t_{\text{н.с.а.с}}}. \quad (31)$$

Здесь $\lambda_{\text{п}}$ – интенсивность отказов простого блока ССН в цепочке элементов; $\lambda_{\text{э}}$ – интенсивность отказов резервируемого элемента в цепочке элементов; $t_{\text{н.с.а.с}}$ – значение наработки сверхсложного блока ССН.

В случае отказа элемента резервируемой части цепочки сложный блок превращается в простой блок, состоящий из цепочки элементов. Интенсивность отказов такого простого блока больше интенсивности отказов сложного блока. Показатели надежности блока ССН вычисляются по формулам (27) – (29). После подстановки значений получим

$$\lambda_1 = \lambda_{\text{сб}}; \quad (32)$$

$$\lambda_2 = \lambda_{\text{п}} + \lambda_{\text{э}}, \quad (33)$$

где $\lambda_{\text{э}}$ – интенсивность отказов цепочки элементов сложного блока ССН, превращенного в простой блок.

В случае отказа нерезервируемого элемента в цепочке сложного блока показатели надежности сверхсложного блока будут равны показателям надежности сложного блока, входящего в состав сверхсложного блока.

К примеру, чтобы рассчитать вероятность безотказной работы сверхсложного блока ССН при интенсивности отказов простого блока цепочки $\lambda_{\text{п}} = 13 \cdot 10^{-8}$ 1/ч, необходимо принять следующие значения: интенсивность отказов резервируемого элемента в цепочке элементов $\lambda_{\text{э}} = 14 \cdot 10^{-8}$ 1/ч и наработка сверхсложного блока ССН $t_{\text{н.с.а.с}} = 46 \cdot 000$ ч.

Интенсивность отказов резервируемого блока определяем по формуле (30). После подстановки в формулу (31) значений λ_{Π} и λ_{Σ} получаем $\lambda_{СБ} = 18,5 \cdot 10^{-8}$ 1/ч.

Значения вероятности безотказной работы сверхсложного блока при условии, что все элементы блока работоспособны, вычисляем по формуле (30). После подстановки в формулу (30) значений $\lambda_{СБ}$ и $t_{НС}$ получаем $P_{БН}(t_{НС}) = 0,999\ 964$.

Вероятность безотказной работы сверхсложного блока при условии, что отказал элемент резервируемой части цепочки сложного блока, определяем по формуле (27). После подстановки в формулу (27) значений $\lambda_1 = \lambda_{СБ}$, $\lambda_2 = \lambda_{\Pi} + \lambda_{\Sigma} = 27 \cdot 10^{-8}$ 1/ч и $t_{НСАС}$ получим $P_{БН}(t_{НС}) = 0,999\ 911$.

4. Оценка безотказности БА МКА в полете

В процессе полета МКА на наземный пункт управления (НПУ) КА поступает телеметрическая информация, которая содержит данные об отказах элементов структуры МКА. В связи с этим можно утверждать, что персонал, участвующий в эксплуатации МКА, в любой момент времени полета t_{Π} обладает информацией о текущей структуре МКА и оставшемся промежутке времени до наступления момента $T_{САС}$.

Совершенно очевидно, что такого вида расчеты эксплуатационный персонал выполнять не может. Представляется целесообразным разработать методику, характеризующую безотказность БА МКА в полете и полезную для выполнения эксплуатационным персоналом МКА. При разработке методики допустимо использовать данные, которые имеются в конструкторском документе «Расчет надежности МКА», и сведения о структуре БА на текущий момент.

Следует иметь в виду, что изменение значения показателя безотказности производится по мере увеличения времени полета и числа отказов элементов в резервируемых объектах БА. Оценку безотказности БА в процессе полета эксплуатационным персоналом целесообразно проводить на момент отказа одного из элементов в резервируемом объекте БА, когда происходит изменение показателя безотказности БА.

Представляется целесообразным в качестве оценки состояния БА в части надежности ввести оценку потенциальной безотказности БА в процессе полета МКА. Совершенно очевидно, что потенциальные возможности БА МКА в части безотказности зависят от количества оставшихся работоспособными резервных элементов, которых вначале достаточно много. Например, в БА БКА зарезервированы 22 объекта. Очевидно, что по мере отказов резервных элементов будет снижаться и уровень безотказности. Сведения о расходовании резервных элементов на борту поступают в наземный комплекс управления (НКУ) по телеметрии.

В связи с изложенным выше возникает задача оценки оставшегося после определенного времени полета МКА потенциала безотказности БА МКА.

Предлагается проводить оценку вероятности безотказной работы БА МКА за наработку, равную сроку активного существования при сохранившемся наборе резервных элементов, и затем сравнивать ее с первоначальным значением.

Для оценки безотказности БА в процессе полета МКА целесообразно использовать структурную схему надежности БА (рис. 2).

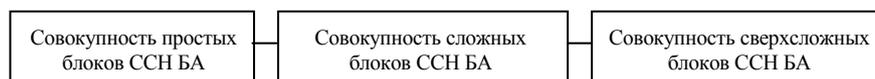


Рис. 2. Структурная схема надежности БА МКА

В соответствии с ССН на рис. 2 исходное значение вероятности безотказной работы БА в интервале САС следует вычислять по формуле

$$P_{БА}(T_{САС}) = P_{С.П}(T_{САС}) P_{С.С}(T_{САС}) P_{С.СС}(T_{САС}), \quad (34)$$

где $P_{БА}(T_{САС})$ – вероятность безотказной работы БА в интервале САС;

$P_{C.П}(T_{CAC})$ – вероятность безотказной работы совокупности простых блоков ССН БА в интервале САС;

$P_{C.C}(T_{CAC})$ – вероятность безотказной работы совокупности сложных блоков ССН БА с простым резервом в интервале САС;

$P_{C.CC}(T_{CAC})$ – вероятность безотказной работы совокупности сверхсложных блоков ССН БА со сложным резервом в интервале САС;

$$P_{C.П}(T_{CAC}) = \prod_{i=1}^{m_1} P_{Пi}(T_{CAC}); \quad (35)$$

$$P_{C.C}(T_{CAC}) = \prod_{i=1}^{m_2} P_{C.i}(T_{CAC}); \quad (36)$$

$$P_{C.CC}(T_{CAC}) = \prod_{i=1}^{m_3} P_{CC.i}(T_{CAC}). \quad (37)$$

Здесь $P_{Пi}(T_{CAC})$, $P_{C.i}(T_{CAC})$, $P_{CC.i}(T_{CAC})$ – вероятности безотказной работы в интервале САС i -го простого, сложного, сверхсложного блоков ССН БА соответственно; m_1 , m_2 , m_3 – количество простых, сложных, сверхсложных блоков в ССН БА соответственно.

При отказе хотя бы одного из совокупности простых блоков ССН БА произойдет отказ БА. При отказе элемента в блоке из совокупности сложных блоков отказ БА не произойдет, произойдет возрастание интенсивности отказов и спад вероятности безотказной работы этого блока, а следовательно, и БА. При отказе в сверхсложном блоке нерезервируемого элемента произойдет то, что происходит при отказе элемента в сложном блоке. При отказе в сверхсложном блоке элемента из резервируемой группы отказ БА не произойдет, более того, сохранится и резерв, но произойдет возрастание интенсивности отказов и спад вероятности безотказности работы блока, а следовательно, и БА.

Новое текущее значение вероятности безотказной работы БА в интервале САС определяется по формуле

$$P_T(T_{CAC}) = \frac{P_{ПР}(T_{CAC}) P_{Э}(T_{CAC})}{P_{О.РГЭ}(T_{CAC})}, \quad (38)$$

где $P_T(T_{CAC})$ – текущее (новое) значение вероятности безотказной работы БА;

$P_{ПР}(T_{CAC})$ – предыдущее текущее значение вероятности безотказной работы БА (сохраненное ранее);

$P_{О.РГЭ}(T_{CAC})$ – вероятность безотказной работы отказавшей группы резервируемых элементов БА;

$P_{Э}(T_{CAC})$ – вероятность безотказной работы элемента, оставшегося работоспособным, из отказавшей группы резервируемых элементов БА.

Новое текущее значение интенсивности отказов БА в интервале САС определяется по формуле

$$\Lambda_T = \Lambda_{ПР} - \Lambda_{РГЭ} + \Lambda_{Э}, \quad (39)$$

где Λ_T – текущее (новое) значение интенсивности отказов БА;

$\Lambda_{ПР}$ – предыдущее текущее значение интенсивности отказов БА (сохраненное ранее);

$\Lambda_{РГЭ}$ – интенсивность отказов отказавшей группы резервируемых элементов БА;

$\Lambda_{Э}$ – интенсивность отказов элемента, оставшегося работоспособным, из отказавшей группы резервируемых элементов БА.

Значения $P_{О.РГЭ}(t_{CAC})$, $P_{Э}(t_{CAC})$, $\Lambda_{РГЭ}$, $\Lambda_{Э}$ берутся из заранее подготовленной таблицы, выполненной в процессе расчета надежности БА.

Оценку потенциальной текущей безотказности БА предлагается проводить по формулам

$$K_{C.B.T} = \frac{P_T(T_{C.A.C.})}{P_H(T_{C.A.C.})}; \quad (40)$$

$$K_{П.И.Т} = \frac{\Lambda_T}{\Lambda_{ПР}}, \quad (41)$$

где $K_{C.B.T}$ – коэффициент снижения вероятности безотказной работы БА по состоянию на текущий момент времени;

$K_{П.И.Т}$ – коэффициент повышения интенсивности отказов БА по состоянию на текущий момент времени;

$P_T(T_{C.A.C.})$ – вероятность безотказной работы БА по состоянию на текущий момент времени;

$P_H(T_{C.A.C.})$ – вероятность безотказной работы БА в исходном состоянии, когда все элементы БА работоспособны.

Например, для МКА с показателями безотказности $P_H(T_{C.A.C.}) = 0,901132$ и $\Lambda_H = 2,542 \cdot 10^{-6}$ 1/ч, модель надежности которого представлена на рис. 2, необходимо определить коэффициенты снижения вероятности безотказной работы БА МКА за САС в пять лет и повышения интенсивности отказов при отказе элементов в сложных блоках ССН БА (табл. 1).

Ниже приведен фрагмент таблицы показателей безотказности сложных блоков ССН МКА, составленной в процессе расчета надежности МКА (табл. 1).

Таблица 1

Показатели безотказности сложных блоков

Дата	Событие	Текущее значение		$K_{C.B.T}$	$K_{П.И.Т}$
		$P_{BA}(t_{C.A.C.})$	$\Lambda_{BA} \cdot 10^{-8}$ 1/ч		
01.01.14	Исходное состояние	0,901 132	254,20	1,000	1,000
01.06.14	Отказал один компьютер БВС БКУ (2.2.4 ССН)	0,892 201	265,07	0,990	1,043
	Отказала резервируемая цепочка элементов в СОС	0,885 278	273,57	0,982	1,076
	Отказала резервируемая цепочка элементов в АСН	0,879 620	280,57	0,976	1,103
	Отказала резервируемая цепочка элементов в ТКС	0,872 794	289,07	0,968	1,137
	Отказал резервируемый узел управления в аппаратуре съемочной системы	0,864 971	295,59	0,960	1,163

Таблица 2

Значения показателей безотказности текущего состояния БА БКА

Дата	Событие	Текущее значение		$K_{C.B.T}$	$K_{П.И.Т}$
		$P_{BA}(t_{C.A.C.})$	$\Lambda_{BA} \cdot 10^{-8}$ 1/ч		
01.01.14	Исходное состояние	0,901 132	254,20	1,000	1,000
01.06.14	Отказал один компьютер БВС БКУ	0,892 201	265,07	0,990	1,043
	Отказала резервируемая цепочка элементов в СОС	0,885 278	273,57	0,982	1,076
	Отказала резервируемая цепочка элементов в АСН	0,879 620	280,57	0,976	1,103
	Отказала резервируемая цепочка элементов в ТКС	0,872 794	289,07	0,968	1,137
	Отказал резервируемый узел управления в аппаратуре съемочной системы	0,864 971	295,59	0,960	1,163

Табл. 2 представляет собой форму документа, который должна вести эксплуатационная организация МКА для фиксации расчетных значений принятых показателей безотказности БА на текущий момент времени.

Заключение

Предложенный вариант оценки текущего состояния безотказности БА МКА легко реализуем в эксплуатирующей организации МКА, но требует предварительной подготовки расчетных значений данных, используемых при оценке текущего состояния безотказности БА МКА еще на этапе разработки эксплуатационной документации организацией – разработчиком БА.

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. Надежность технических систем : справ. / под ред. И.А. Ушакова. – М. : Радио и связь, 1985. – 606 с.
4. Зинченко, О.Н. Малые оптические спутники ДЗЗ / О.Н. Зинченко // Ракурс [Электронный ресурс]. – 2015. – Режим доступа : <http://www.racurs.ru>. – Дата доступа : 07.09.2015.
5. Менеджмент риска. Структурная схема надежности и булевы методы : ГОСТ Р 51891–2008 (ИСО 1161:1984). – М. : Стандартинформ, 2008. – 53 с.
6. Половко, А.М. Основы теории надежности / А.М. Половко, С.В. Гуров. – СПб. : БХВ–Петербург, 2006. – 560 с.
7. Овчаров, Л.А. Прикладные задачи теории массового обслуживания / Л.А. Овчаров. – М. : Машиностроение, 1969. – 324 с.
8. Шор, Я.Б. Статистические методы анализа и контроля качества и надежности / Я.Б. Шор. – М. : Сов. радио, 1962. – 553 с.

Поступила 28.07.2015

*Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: aliushkevich@newman.bas-net.by*

L.I. Kulbak, V.B. Aliushkevich, S.A. Zolotoy

ASSESSMENT OF RELIABILITY OF THE ONBOARD EQUIPMENT OF SMALL SPACECRAFTS DURING THEIR FLIGHT

The technique of an assessment of current state of reliability of the onboard equipment of small spacecrafts during their flight is given. A mathematical representation of the reliability indicator of onboard equipment and the model of structure of a small spacecraft, as well as formulas for calculating the fail-safe functioning indicators of reliability diagram blocks of onboard equipment are adduced.

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. Статья должна иметь индекс УДК (универсальная десятичная классификация).

3. 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.

4. 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 выполнены с хорошим разрешением в масштабе, позволяющем четко различать надписи и обозначения. Подрисовочные подписи с расшифровкой 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.

5. 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 , ∞); символы химических элементов (C, Cl, 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).

6. 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.

7. 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>.

8. 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.

9. Статьи, направляемые на доработку, должны быть возвращены в исправленном виде с ответами на все вопросы.

10. Редакция журнала предоставляет возможность первоочередного опубликования статей, представленных лицами, которые осуществляют послевузовское обучение (аспирантура, докторантура, соискательство) в год завершения обучения.

11. Авторы несут ответственность за направление в редакцию статей, уже опубликованных ранее, или статей, принятых к публикации другими изданиями.

12. 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