

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

ИНФОРМАТИКА

ТОМ 17

2

АПРЕЛЬ-ИЮНЬ
2020

ОТ РЕДАКЦИИ

В журнале «Информатика» публикуются оригинальные и обзорные статьи, описывающие результаты фундаментальных и прикладных исследований специалистов академического и вузовского профиля в области информатики и информационных технологий. Основной целью журнала является публикация наиболее значимых новых результатов в указанной области. Приветствуются статьи, описывающие заключительные результаты научных проектов и диссертационных исследований, открывающие новые направления исследований, которые находятся на стыке информатики и других наук.

Основные разделы журнала: математическое моделирование; обработка сигналов, изображений и речи; прикладные информационные технологии; интеллектуальные системы; космические информационные технологии и геоинформатика; параллельные вычисления; системы, приборы и устройства; распознавание образов; информационная безопасность; автоматизация проектирования.

Журнал «Информатика» включен Высшей аттестационной комиссией Республики Беларусь в список научных изданий для опубликования результатов диссертационных исследований. В декабре 2017 г. журнал включен в базу данных Российского индекса научного цитирования (РИНЦ). С помощью инструментов и сервисов, доступных на платформе eLIBRARY (раздел «Личный кабинет»), можно самостоятельно корректировать список своих публикаций и цитирований в РИНЦ.

Журнал рассчитан на широкий круг специалистов в области информатики и информационных технологий.

Адрес редакции:

ул. Сурганова, 6, к. 305, г. Минск, Беларусь

Тел. +375(017)284 26 22

E-mail: rio@newman.bas-net.by

Сайт журнала: inf.grid.by

THE EDITOR'S NOTE

The journal «Informatics» is a scientific publication in computer sciences and information technologies which reviews the results in basic and applied research of scientists from the universities and academies in the given field. The journal focuses on the most significant and modern papers such as research projects results and PhD/DSc papers in computer sciences, IT and at the boundaries.

The journal covers the following topics: mathematical modeling; processing and recognition of signals, images and speech; applied information technology; intelligent systems; space information technology and GIS technologies; parallel computing; systems, devices and equipment; image recognition; information security; computer aided design.

The journal «Informatics» is in the list of scientific publications recommended by the Higher Attestation Commission of the Republic of Belarus for scientists to publish the results of PhD/DSc research. In December 2017 the journal was included in the database of the Russian Science Citation Index (RISC) and provides the free access to reviewed electronic scientific paper, improving scientific information traffic and also raising quotation of works of the authors who are published in this journal (please use <https://elibrary.ru> or section for authors https://elibrary.ru_author_tools).

The journal is edited for a wide range of specialists in IT and computer sciences.

For further information:

Phone +375 (017) 284 26 22

E-mail: rio@newman.bas-net.by

Office 305, Surganova 6, 220012

Minsk, Belarus

Learn more at: <https://inf.grid.by/jour>

ОБЪЕДИНЕННЫЙ ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ
НАЦИОНАЛЬНОЙ АКАДЕМИИ НАУК БЕЛАРУСИ

ИНФОРМАТИКА

Том 17, № 2, апрель-июнь 2020

Ежеквартальный научный журнал

Издается с января 2004 г.

Учредитель и издатель – Объединенный институт проблем информатики
Национальной академии наук Беларуси

Г л а в н ы й р е д а к т о р

Тузиков Александр Васильевич, д-р физ.-мат. наук, профессор, член-корреспондент
Национальной академии наук Беларуси, генеральный директор Объединенного института
проблем информатики Национальной академии наук Беларуси, Минск, Беларусь

З а м е с т и т е л ь г л а в н о г о р е д а к т о р а

Ковалев Михаил Яковлевич, д-р физ.-мат. наук, профессор, член-корреспондент
Национальной академии наук Беларуси, Объединенный институт проблем информатики
Национальной академии наук Беларуси, Минск, Беларусь

Р е д а к ц и о н н а я к о л л е г и я

Абламейко Сергей Владимирович, д-р техн. наук, профессор, академик Национальной академии наук
Беларуси, Белорусский государственный университет, Минск, Беларусь

Анищенко Владимир Викторович, канд. техн. наук, доцент, ООО «СофтКлуб», Минск, Беларусь

Бибило Петр Николаевич, д-р техн. наук, профессор, Объединенный институт проблем информатики
Национальной академии наук Беларуси, Минск, Беларусь

Бобов Михаил Никитич, д-р техн. наук, профессор, ОАО «АГАТ – системы управления» – управляющая
компания холдинга «Геоинформационные системы управления», Минск, Беларусь

Долгий Александр Борисович, д-р техн. наук, профессор, Высшая инженерная школа Бретани, Нант,
Франция

Дудин Александр Николаевич, д-р физ.-мат. наук, профессор, Белорусский государственный университет,
Минск, Беларусь

Карпов Алексей Анатольевич, д-р техн. наук, доцент, Санкт-Петербургский институт информатики
и автоматизации Российской академии наук, Санкт-Петербург, Россия

Килин Сергей Яковлевич, д-р физ.-мат. наук, профессор, академик Национальной академии наук
Беларуси, Президиум Национальной академии наук Беларуси, Минск, Беларусь

Краснопрошин Виктор Владимирович, д-р техн. наук, профессор, Белорусский государственный
университет, Минск, Беларусь

Крот Александр Михайлович, д-р техн. наук, профессор, Объединенный институт проблем информатики
Национальной академии наук Беларуси, Минск, Беларусь

Кругликов Сергей Владимирович, д-р воен. наук, канд. техн. наук, доцент, Объединенный институт проблем информатики Национальной академии наук Беларуси, Минск, Беларусь

Кундас Семен Петрович, д-р техн. наук, профессор, Белорусский национальный технический университет, Минск, Беларусь

Лиходед Николай Александрович, д-р физ.-мат. наук, профессор, Белорусский государственный университет, Минск, Беларусь

Матус Петр Павлович, д-р физ.-мат. наук, профессор, Институт математики Национальной академии наук Беларуси, Минск, Беларусь

Скляр Валерий Анатольевич, д-р техн. наук, профессор, Университет Авейру, Португалия

Сотсков Юрий Назарович, д-р физ.-мат. наук, профессор, Объединенный институт проблем информатики Национальной академии наук Беларуси, Минск, Беларусь

Стемпковский Александр Леонидович, д-р техн. наук, профессор, академик Российской академии наук, Институт проблем проектирования в микроэлектронике Российской академии наук, Москва, Россия

Харин Юрий Семенович, д-р физ.-мат. наук, профессор, член-корреспондент Национальной академии наук Беларуси, Научно-исследовательский институт прикладных проблем математики и информатики Белорусского государственного университета, Минск, Беларусь

Чернявский Александр Федорович, д-р техн. наук, профессор, академик Национальной академии наук Беларуси, Институт прикладных физических проблем им. А. Н. Севченко Белорусского государственного университета, Минск, Беларусь

Ярмолик Вячеслав Николаевич, д-р техн. наук, профессор, Белорусский государственный университет информатики и радиоэлектроники, Минск, Беларусь

ИНФОРМАТИКА

Том 17, № 2, апрель-июнь 2020

Ответственный за выпуск *Мойсейчик Светлана Сергеевна*

Редактор *Гончаренко Галина Борисовна*

Корректор *Михайлова Анна Антоновна*

Компьютерная верстка *Бутевич Ольга Борисовна*

Сдано в набор 18.05.2020. Подписано в печать 15.06.2020. Формат 60×84 1/8. Бумага офсетная. Гарнитура Таймс. Ризография. Усл. печ. л. 16,0. Уч.-изд. л. 15,7. Тираж 50 экз. Заказ 2.

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, Беларусь.

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

THE UNITED INSTITUTE OF INFORMATICS PROBLEMS
OF THE NATIONAL ACADEMY OF SCIENCES OF BELARUS

INFORMATICS

Vol. 17, no. 2, April-June 2020

Published quarterly

Issued since January 2004

Founder and publisher – the United Institute of Informatics Problems
of the National Academy of Sciences of Belarus

Editor-in-Chief

Alexander V. Tuzikov, Dr. Sci. (Phys.-Math.), Professor, Corresponding Member of the National Academy of Sciences of Belarus, General Director of the United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus

Deputy Editor-in-Chief

Mikhail Y. Kovalyov, Dr. Sci. (Phys.-Math.), Professor, Corresponding Member of the National Academy of Sciences of Belarus, the United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus

Editorial Board

Sergey V. Ablameyko, Dr. Sci. (Eng.), Professor, Academician of the National Academy of Sciences of Belarus, Belarusian State University, Minsk, Belarus

Uladimir V. Anishchanka, Cand. Sci. (Eng.), Associate Professor, SoftClub Ltd., Minsk, Belarus

Petr N. Bibilo, Dr. Sci. (Eng.), Professor, the United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus

Mikhail N. Bobov, Dr. Sci. (Eng.), Professor, Open Joint-Stock Company "AGAT – Control Systems – Managing Company of Geoinformation Control Systems Holding", Minsk, Belarus

Alexandre B. Dolgui, Dr. Sci. (Eng.), Professor, IMT Atlantique, Nantes, France

Alexander N. Dudin, Dr. Sci. (Phys.-Math.), Professor, Belarusian State University, Minsk, Belarus

Alexey A. Karpov, Dr. Sci. (Eng.), Associate Professor, St. Petersburg Institute of Informatics and Automation of the Russian Academy of Sciences, Saint Petersburg, Russia

Sergey Ya. Kilin, Dr. Sci. (Phys.-Math.), Professor, Academician of the National Academy of Sciences of Belarus, Presidium of the National Academy of Sciences of Belarus, Minsk, Belarus

Viktor V. Krasnoproshin, Dr. Sci. (Eng.), Professor, Belarusian State University, Minsk, Belarus

Alexander M. Krot, Dr. Sci. (Eng.), Professor, the United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus

Sergey V. Kruglikov, Dr. Sci. (Milit.), Cand. Sci. (Eng.), Associate Professor, the United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus

Semen P. Kundas, Dr. Sci. (Eng.), Professor, Belarusian National Technical University, Minsk, Belarus

Nikolai A. Likhoded, Dr. Sci. (Phys.-Math.), Professor, Belarusian State University, Minsk, Belarus

Petr P. Matus, Dr. Sci. (Phys.-Math.), Professor, Institute of Mathematics of the National Academy of Sciences of Belarus, Minsk, Belarus

Valery A. Sklyarov, Dr. Sci. (Eng.), Professor, University of Aveiro, Portugal

Yuri N. Sotskov, Dr. Sci. (Phys.-Math.), Professor, the United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus

Alexander L. Stempkovsky, Dr. Sci. (Eng.), Professor, Academician of the Russian Federation Academy of Sciences, the Institute for Design Problems in Microelectronics of the Russian Federation Academy of Sciences, Moscow, Russia

Yuriy S. Kharin, Dr. Sci. (Phys.-Math.), Professor, Corresponding Member of the National Academy of Sciences of Belarus, Research Institute for Applied Problems of Mathematics and Informatics of the Belarusian State University, Minsk, Belarus

Alexander F. Cherniavsky, Dr. Sci. (Eng.), Professor, Academician of the National Academy of Sciences of Belarus, A. N. Sevchenko Institute of Applied Physical Problems of the Belarusian State University, Minsk, Belarus

Vyacheslav N. Yarmolik, Dr. Sci. (Eng.), Professor, Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus

INFORMATICS

Vol. 17, no. 2, April-June 2020

Issue Head *Sviatlana S. Maiseichyk*

Editor *Halina B. Hancharenka*

Corrector *Hanna A. Mikhailava*

Computer Imposition *Volha B. Butsevich*

Sent for press 18.05.2020. Output 15.06.2020. Format 60×84 1/8. Offset paper. Headset Times. Riesography. Printed sheets 16,0. Publisher's signatures 15,7. Circulation 50 copies. Order 2.

State Scientific Institution "The United Institute of Informatics Problems of the National Academy of Sciences of Belarus".

Certificate on the state registration of the publisher, manufacturer, distributor of printing editions

no. 1/274 dated 04.04.2014. License for the press no. 02330/444 dated 18.12.13.

6, Surganov Str., 220012, Minsk, Belarus.

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

СОДЕРЖАНИЕ

ОБРАБОТКА СИГНАЛОВ, ИЗОБРАЖЕНИЙ И РАСПОЗНАВАНИЕ ОБРАЗОВ

- Богуш Р. П., Захарова И. Ю., Абламейко С. В.** Обнаружение объектов на изображениях с большим разрешением на основе их пирамидально-блочной обработки 7
- Жук Р. С., Залесский Б. А., Троцкий Ф. С.** Визуальная навигация автономно летящего БПЛА с целью его возвращения в точку старта 17
- Ма Ц., Цветков В. Ю., Конопелько В. К.** Скелетизация изображений на основе комбинации одно- и двухподытерационных моделей 25
- Вашкевич Г. С., Азаров И. С.** Выделение речевой активности на фоне шумов при помощи компактной сверточной нейронной сети 36

БИОИНФОРМАТИКА

- Хадарович А. Ю., Калиновский А. А., Тузиков А. В.** Предсказание структуры гомодимерных белковых комплексов на основе глубокой нейронной сети 44

АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ

- Ярмолик В. Н., Мрозек И., Ярмолик С. В.** Псевдоисчерпывающее тестирование запоминающих устройств на базе маршевых тестов типа March A 54
- Сапожников В. В., Сапожников Вл. В., Ефанов Д. В.** Схема коррекции сигналов для комбинационных устройств автоматики на основе логического дополнения с контролем вычислений по паритету 71

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

- Сотсков Ю. Н., Егорова Н. Г., Матвейчук Н. М.** Алгоритмы планирования рабочего времени в условиях интервальной неопределенности 86
- Ерофеев В. Т.** Математическая модель проникновения цилиндрических электромагнитных полей с осевой симметрией через плоский экран из пермаллоя 103

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

- Маркевич В. Э., Легкоступ В. В.** Наведение сверхзвукового управляемого объекта в многопозиционной радиолокационной станции воздушного базирования 120

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

CONTENTS

SIGNAL AND IMAGE PROCESSING AND PATTERN RECOGNITION

- Bohush R. P., Zakharava I. Yu., Ablameyko S. V.** Object detection in high resolution images based on multiscale and block processing 7
- Zhuk R. S., Zalesky B. A., Trotski Ph. S.** Solution of problem of returning to the starting point of autonomously flying UAV by visual navigation 17
- Ma J., Tsviatkou V. Yu., Kanapelka V. K.** Image skeletonization based on combination of one- and two-sub-iterations models 25
- Vashkevich R. S., Azarov E. S.** Voice activity detection in noisy conditions using tiny convolutional neural network 36

BIOINFORMATICS

- Hadarovich A. Y., Kalinouski A. A., Tuzikov A. V.** Protein homodimers structure prediction based on deep neural network 44

COMPUTER-AIDED DESIGN

- Yarmolik V. N., Mrozek I., Yarmolik S. V.** Pseudoexhaustive memory testing based on March A type march tests 54
- Sapozhnikov V. V., Sapozhnikov VI. V., Efanov D. V.** Signal correction for combinational automation devices on the basis of Boolean complement with control of calculations by parity 71

MATHEMATICAL MODELING

- Sotskov Yu. N., Egorova N. G., Matsveichuk N. M.** Algorithms for planning working time under interval uncertainty 86
- Erofeenko V. T.** Mathematical model of penetration of cylindrical electromagnetic fields with axial symmetry through the plane screen from permalloy 103

APPLIED INFORMATION TECHNOLOGIES

- Markevich V. E., Legkostup V. V.** Guiding a supersonic controlled object in a multi-position air basing radar 120

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

ОБРАБОТКА СИГНАЛОВ, ИЗОБРАЖЕНИЙ И РАСПОЗНАВАНИЕ ОБРАЗОВ
SIGNAL AND IMAGE PROCESSING AND PATTERN RECOGNITION

УДК 004.9
<https://doi.org/10.37661/1816-0301-2020-17-2-7-16>

Поступила в редакцию 25.02.2020
Received 25.02.2020

Принята к публикации 04.04.2020
Accepted 04.04.2020

Обнаружение объектов на изображениях с большим разрешением на основе их пирамидально-блочной обработки

Р. П. Богуш¹✉, И. Ю. Захарова¹, С. В. Абламейко^{2,3}

¹Полоцкий государственный университет, Новополоцк, Беларусь
✉E-mail: bogushr@mail.ru

²Белорусский государственный университет, Минск, Беларусь

³Объединенный институт проблем информатики
Национальной академии наук Беларуси, Минск, Беларусь

Аннотация. Предлагается алгоритм для обнаружения объектов на изображениях с большим разрешением, основанный на многомасштабном представлении изображения, пирамидально-блочной обработке с перекрытием, применении сверточной нейронной сети для каждого блока и объединении обнаруженных областей. Количество слоев пирамиды определяется размерами изображения и входного слоя используемой сверточной нейронной сети. На всех уровнях, кроме самого верхнего, выполняется блочное разбиение, а применение при этом перекрытия позволяет улучшить правильную классификацию объектов, которые разделяются на фрагменты и расположены в соседних блоках. Решение об объединении таких областей принимается на основе анализа метрики пересечения над объединением для них и принадлежности к одному классу. Представленные результаты тестирования алгоритма подтверждают, что рассмотренный подход позволяет повысить точность обнаружения объектов небольших размеров на изображениях с большим разрешением.

Ключевые слова: сверточная нейронная сеть, блочная обработка, разрешение 4K, обнаружение объектов, многомасштабное представление изображения

Для цитирования. Богуш, Р. П. Обнаружение объектов на изображениях с большим разрешением на основе их пирамидально-блочной обработки / Р. П. Богуш, И. Ю. Захарова, С. В. Абламейко // Информатика. – 2020. – Т. 17, № 2. – С. 7–16. <https://doi.org/10.37661/1816-0301-2020-17-2-7-16>

Object detection in high resolution images based on multiscale and block processing

Rykhard P. Bohush¹✉, Iryna Yu. Zakharava¹, Sergey V. Ablameyko^{2,3}

¹Polotsk State University, Novopolotsk, Belarus
✉E-mail: bogushr@mail.ru

²Belarusian State University, Minsk, Belarus

³The United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus

Abstract. In the paper the algorithm for object detection in high resolution images is proposed. The approach uses multiscale image representation followed by block processing with the overlapping value. For each block the object detection with convolutional neural network was performed. Number of pyramid layers is limited by

the Convolutional Neural Network layer size and input image resolution. Overlapping blocks splitting to improve the classification and detection accuracy is performed on each layer of pyramid except the highest one. Detected areas are merged into one if they have high overlapping value and the same class. Experimental results for the algorithm are presented in the paper.

Keywords: convolutional neural network, block processing, 4K resolution, object detection, multiscale image representation

For citation. Bohush R. P., Zakharava I. Yu., Ablameyko S. V. Object detection in high resolution images based on multiscale and block processing. *Informatics*, 2020, vol. 17, no. 2, pp. 7–16 (in Russian). <https://doi.org/10.37661/1816-0301-2020-17-2-7-16>

Введение. Развитие цифровых видеокамер привело к тому, что на практике все шире используются устройства, позволяющие формировать изображения и видео с большим разрешением формата 4K×2K [1]. Большая сторона изображений при этом может принимать размеры от 3600 до 4500 пикселей для разрешения 4K [2]. На таких изображениях значительно увеличивается качество отображения объектов, в том числе небольших размеров либо удаленных при съемке на значительное расстояние от видеокамеры, и, соответственно, предоставляется возможность автоматического обнаружения объектов, которые при меньшем разрешении не обладают достаточной информативностью для решения этой задачи.

Можно выделить два подхода к обнаружению объектов: на основе сопоставления с эталонным набором признаков и на основе использования сверточных нейронных сетей (СНС). Первый подход предполагает сравнение входного изображения с признаками заранее подготовленных искомым изображений, в качестве которых могут выступать как уровни яркостей каждого пиксела, так и признаки, сформированные по определенным правилам, например гистограммы ориентированных градиентов, признаки Хаара и др. Поскольку выбор оптимального набора информативных признаков для точного описания произвольных изображений является не решенной в полной мере задачей, то при таком подходе требуются разработка и тестирование алгоритмов формирования дескрипторов объектов, ориентированных на заданные изображения. Поэтому в последнее время благодаря стремительному развитию вычислительной мощности компьютерной техники все шире используется другой подход, который основан на применении СНС, предварительно обученных на больших базах данных изображений. Наличие операций свертки с различными типами фильтров и множества слоев в их архитектуре позволяет формировать эффективные наборы признаков, размер которых уменьшается от начала к выходу СНС, для описания объектов, отличающихся нечеткой структурой и множеством вариаций в пределах класса. Однако на первом этапе при использовании СНС выполняется масштабирование входного изображения к размерам входного слоя, что обуславливает некоторое уменьшение информативности объектов, а при небольших их исходных размерах приводит к невозможности обнаружения на изображении. Поэтому актуальной задачей является разработка алгоритмов, позволяющих детектировать объекты малых размеров на изображениях с большим разрешением.

Существующие методы обнаружения объектов на изображениях с большим разрешением. В статье [3] предложен подход для локализации объектов на изображениях с большим разрешением, который основан на блочной обработке с перекрытием и применении СНС для выделения областей объектов в каждом блоке. Основным недостатком этого метода является достаточно высокая вероятность объединения нескольких различных объектов, расположенных на границах обрабатываемых блоков, в одну область и пропуска объекта за счет его фрагментации по блокам.

В работе [4] представлен алгоритм, на вход которого поступают исходное и масштабированное до размера [227×227] изображения. На уменьшенной копии с помощью СНС AlexNet детектируются объекты. Для обнаружения объектов с меньшими размерами исходное изображение делится на непересекающиеся блоки, размеры которых изменялись от 600 до 75 пикселей и обрабатывались с использованием СНС SCN (Spatial Correlation Network). Перекрытие при этом варьировалось от 25 до 50 % относительно размеров блоков. Для алгоритма [4] авторами получено значение полноты обнаружения (recall), составившее 80 % на базе данных

SUN2012 [5]. Известно, что метрика recall не позволяет оценить ложные срабатывания, а применяемая авторами база данных практически не содержит изображения с разрешением 4К.

В статье [6] рассмотрен подход для обнаружения людей, в котором входное изображение разделяется на пересекающиеся блоки большого размера. Если человек обнаружен, то выполняется разбиение на блоки меньшего размера для уточнения ограничивающих рамок обнаруженных областей интереса. Объединение найденных областей основано на вертикально вытянутой (прямоугольной) форме человека. Это означает, что для других классов объектов алгоритм не адаптирован. Оценка точности алгоритма выполнена с использованием базы данных PEVID-UHD [7] и закрытой базы, содержащей несколько большее количество объектов. Для первого набора данных значение средней точности равно 90,7 %, для второго – 75,4 %. При этом использование перекрытия блоков в диапазоне от 20 до 50 пикселей не является достаточным в ряде случаев, кроме того, оно не адаптивно к размеру входного изображения.

В работе [8] предложен алгоритм для обнаружения небольших объектов классов «транспортное средство» и «человек» на изображениях с разрешением 2К и размерами [1920×1080], полученных с малых беспилотных летательных аппаратов для реализации на графических процессорах мобильных устройств. При этом используется блочная обработка исходного изображения СНС PeleeNet со взаимным перекрытием 25 %. Полученные результаты тестирования на базе данных VisDrone2018 свидетельствуют о достижении значения метрики mAP (mean Average Precision) для данного алгоритма 36,67 %. Однако в работе не представлена оценка влияния величины перекрытия блоков на точность алгоритма. Величина пропуска объектов класса «человек» значительно выше по сравнению с классом «транспортное средство». Следовательно, объекты небольших размеров данный алгоритм обнаруживает значительно хуже даже на изображениях с разрешением ниже 4К.

Среди известных моделей СНС, отличающихся архитектурой, точностью и быстродействием, для решения поставленной задачи следует выделить модификации R-CNN (Region-based Convolutional Neural Networks) [9], ResNet (Residual Networks) [10] и YOLO (You Only Look Once) [11]. Первая версия модели R-CNN основана на предварительном выделении регионов на изображении, вычислении признаков с использованием СНС (например, AlexNet) и применении классификатора для идентификации объектов. Данная модель получила развитие в версиях Fast R-CNN [12] и Faster R-CNN [13]. Архитектура Fast R-CNN направлена на уменьшение использования числа регионов на изображении за счет применения слоя ROI, который действует по принципу слоев субдискретизации, позволяя получать области уменьшенного размера [14]. В Faster R-CNN предполагается выполнение двух этапов. На первом из них определяются области на изображении, в которых предположительно могут быть расположены объекты, с применением глубокой полносвязной сети (Region Proposal Network). На втором этапе используется детектор Fast R-CNN, который выполняет поиск объектов в предложенных регионах. Модель Faster R-CNN неустойчива к зашумленным изображениям и требует значительных вычислительных затрат.

Архитектура СНС ResNet использует пропускающие соединения, которые позволяют минимизировать ухудшение качества работы при увеличении количества слоев СНС, если на некотором слое сети достигнут предел точности. Коэффициент ошибок составляет 3,57 % в метрике top 5. В работе [15] представлена модель Inception-ResNet как развитие модели Inception путем ввода замыкающих соединений (shortcut connection), которые при необходимости позволяют пропускать слой и, соответственно, обнулять его влияние на результат работы детектора. Это дает возможность изменять архитектуру сети так, чтобы конечное количество слоев определялось для конкретной задачи в процессе обучения, что позволяет уменьшить коэффициент ошибки до 3,1 % в метрике top 5. Вместе с тем представленные модели характеризуются высокими временными затратами. Поэтому для решения многих задач используются усеченные варианты архитектур с уменьшенным количеством слоев (например, ResNet-34, ResNet-50 и др.), которые, однако, не позволяют достичь точности базовой модели.

Семейство СНС YOLO относится к однопроходным и дает возможность осуществлять локализацию и идентификацию объектов одной моделью. Среди существующих версий данных СНС по результатам тестирования в метрике top 5 наибольшей точностью (93,8 %) отличается YOLO третьей версии (URL: <https://arxiv.org/pdf/1804.02767.pdf>), которая для выделения при-

знаков применяет улучшенную архитектуру Darknet-53, содержащую 53 слоя и использующую 23 замыкающих соединения. Так, в качестве СНС для детектирования объектов в алгоритме их обнаружения на изображениях с разрешением 4K целесообразно применять СНС YOLO v3, поскольку данная версия характеризуется хорошей точностью обнаружения и удовлетворительным временем обработки.

Алгоритм обнаружения объектов на основе многомасштабного представления и блочной обработки сверточной нейронной сетью. Предлагаемый алгоритм требует пирамидального представления исходного изображения в виде набора его копий уменьшающегося масштаба. По мере движения вверх по пирамиде масштаб (размеры и разрешение) оригинала уменьшается. На каждом уровне пирамиды изображения разделяются на блоки, в которых выполняется детектирование с применением СНС. Для обеспечения высокой результативности обнаружения объектов малого размера необходимо перед обработкой минимизировать их разбиение на части, т. е. важно стремиться, чтобы ни на одном из блоков, подаваемых на вход СНС, объект не был фрагментирован. Поэтому используется взаимное перекрытие блоков, и чем выше величина перекрытия, тем больших размеров объект может быть обнаружен без вероятности его разделения на части, что, однако, потребует дополнительных вычислительных затрат. После обработки с применением СНС необходима процедура объединения найденных областей на всех уровнях.

Таким образом, алгоритм обнаружения объектов на изображениях с большим разрешением состоит из следующих шагов:

1. Определение количества уровней пирамиды изображений с учетом того, что размеры верхнего уровня должны быть приближены к размерам входного слоя используемой СНС по формуле

$$P = \lceil \log_2(\max(W, H) / l) \rceil + 1,$$

где W, H – ширина и высота входного изображения; $l \times l$ – размер входного слоя используемой СНС; $\lceil * \rceil$ – ближайшее целое.

Данный подход исключает фрагментацию объектов на минимальном масштабе и позволяет увеличить их правильную классификацию в целом.

2. Разбиение изображения на слое p на блоки с перекрытием. При этом их количество определяется равенством

$$B_p = \left(\frac{W / p - l}{k} + 1 \right) \cdot \left(\frac{H / p - l}{k} + 1 \right) = [B_{W_p}] \cdot [B_{H_p}],$$

где p – номер уровня пирамиды, $p = 1, \dots, P$; k – величина сдвига для блока (в пикселах). При округлении B_{W_p} либо B_{H_p} в сторону уменьшения последний неполный блок объединяется с предыдущим и масштабируется к размеру входного слоя СНС, а при округлении в сторону увеличения дополняется нулями.

Величина перекрытия блоков может быть рассчитана по формуле

$$\alpha = \frac{k}{l} \cdot 100 \% .$$

3. Обнаружение областей-кандидатов, которые могут содержать объект или фрагмент объекта. Для этого каждый блок обрабатывается с использованием СНС, а обнаруженная область-кандидат описывается набором признаков:

$$F = (x_1, y_1, x_2, y_2, E, Cl),$$

где x_1, y_1 – координаты верхнего левого угла найденной области на исходном изображении; x_2, y_2 – координаты нижнего правого угла найденной области на исходном изображении; Cl – класс выделенного объекта; E – уверенность СНС в правильной классификации.

4. Объединение областей-кандидатов.

4.1. Для объединения областей-кандидатов O_i и O_j проводится анализ величины их перекрытия на всех уровнях и блоках, а также принадлежности к одному классу:

$$IoU(O_i, O_j) > 0,5, Cl_i = Cl_j,$$

где

$$IoU(O_i, O_j) = \frac{O_i \cap O_j}{O_i \cup O_j}, \quad i, j \in 1, N | i \neq j.$$

Дескриптор объединенной области $F(O')$ формируется следующим образом:

$$F(O') = \left(\min(x_1^{O_i}, x_1^{O_j}), \min(y_1^{O_i}, y_1^{O_j}), \max(x_2^{O_i}, x_2^{O_j}), \max(y_2^{O_i}, y_2^{O_j}), \max(E^{O_i}, E^{O_j}), Cl(\max(E^{O_i}, E^{O_j})) \right).$$

4.2. При блочной обработке результатов детектирования возможно, что один и тот же объект или его фрагменты будут обнаружены со смещением координат на разных масштабах или в соседних ячейках. Кроме того, признаки фрагмента объекта меньшего размера могут отличаться от признаков исходного объекта и даже быть ближе к дескрипторам другого класса. Поэтому при объединении дополнительно применяется постобработка. Для этого предлагается использовать правило, по которому анализируются и объединяются все области, полученные на предыдущем шаге:

$$IoU(O'_i, O'_j) > 0,2, Cl_i = Cl_j, d_x > t_x, d_y > t_y,$$

где d_x и d_y – значения разности между размерами областей-кандидатов по осям x и y соответственно; t_x и t_y – пороговые уровни.

При уменьшении порогового значения на шаге 4.1 увеличится вероятность ложных объединений объектов одного класса, которые расположены на незначительном расстоянии друг от друга.

Результаты исследования предложенного алгоритма. Для тестирования результативности обнаружения объектов применялась выборка из 780 изображений с разрешением 4К, полученных при различных погодных условиях, а также с разными высотой установки и вертикальным углом наклона видеокамеры. Подготовка аннотированной базы данных изображений выполнена с использованием инструмента labelImg (URL: <https://github.com/tzutalin/labelImg>). Всего на изображениях аннотировано 6049 объектов двух классов («человек» и «транспортное средство») с размерами от $[24 \times 14]$ до $[308 \times 763]$. При этом класс «транспортное средство» является составным, так как в него включены классы «автобус», «машина» и «грузовик».

На рис. 1, а всего размечено 19 объектов, из них 4 принадлежат классу «человек», а 15 относятся к классу «транспортное средство». На рис. 1, б показан пример с тремя объектами, которые расположены в затемненной зоне на близком расстоянии друг от друга.

Для проведения экспериментов предложенный алгоритм был реализован на языке программирования Python с применением фреймворка машинного обучения Pytorch, с помощью которого синтезировались основные элементы архитектуры СНС. Базовые операции над изображениями реализованы средствами библиотеки компьютерного зрения OpenCV. Для повышения скорости обнаружения применена пакетная обработка блоков изображений для каждого уровня пирамиды с использованием технологии CUDA, которая позволяет выполнять параллельную обработку данных на базе ресурсов графического процессора.



Рис. 1. Примеры аннотирования изображений для тестовой базы: а) для двух классов объектов; б) для одного класса объектов

Для оценки качества работы алгоритма вычислялась метрика mAP, предполагающая усреднение значений известной метрики AP (Average Precision) по классам. При этом на первом этапе рассчитывалось количество верных T_p и ложных F_p обнаружений на основе применения метрики объединения над пересечением для найденных объектов и аннотированных на изображениях базы данных с пороговым уровнем $T = 50\%$ для принятия решения. Затем вычислялись точность p и полнота r :

$$p = \frac{T_p}{T_p + F_p}, \quad r = \frac{T_p}{T_p + F_n},$$

где F_n – количество пропусков.

На основе полученных значений рассчитывалась метрика AP для каждого класса для 11 пороговых уровней в диапазоне от 0 до 1 с шагом (0,1) по формуле [16]

$$AP = \frac{1}{11} \cdot \sum_{i=0}^{10} p_{\text{int}}(r_i),$$

где $p_{\text{int}}(r_i) = \max p(\tilde{r}), \tilde{r} \geq r_i$.

На последнем этапе при вычислении метрики mAP полученные на предыдущем шаге значения AP усреднялись для двух классов объектов.

Для того чтобы более оптимально использовать возможности СНС при детектировании и идентификации небольших размеров объектов, необходимо определить пороговую степень уверенности СНС и размер входного слоя, обеспечивающие наибольшую результативность работы. Результаты экспериментов подтвердили, что для обеспечения высокой точности при обнаружении объектов небольших размеров на изображениях с большим разрешением необходимо минимизировать их масштабирование. Так, при размере входного слоя $[608 \times 608]$ значение метрики mAP составляет 21 %, а при размере $[1024 \times 1024]$ (использовался компьютер с центральным процессором Intel Core i5-8600 3,6 ГГц, ОЗУ 16 ГБ, NVIDIA GTX 1060) наибольшее значение для mAP 47 % достигается на тестовом наборе изображений с разрешением 4К, при этом пороговая степень уверенности $T = 50\%$. С учетом этого и в соответствии с шагом 1 предложенного алгоритма изображение с разрешением 4К для обработки необходимо представить в виде пирамиды из трех уровней.

На следующем этапе экспериментов выполнена оценка точности работы алгоритма при различных величинах перекрытия блоков и пороговых уровнях СНС с шагом 5 %. Следует отметить, что для минимального масштаба представления исходного изображения пороговое значение соответствует наиболее результативному для используемой СНС, в данном случае $T = 50\%$. Результаты экспериментов представлены в таблице. Их анализ показывает, что при любых значениях α и T обеспечивается превышение метрики mAP рассмотренного алгоритма по сравнению с mAP используемой СНС. Наибольший выигрыш обеспечивается при $\alpha = 60\%$ и $T = 75\%$, при этом mAP = 75,9 %.

Результаты исследования точности предложенного алгоритма, mAP, %

$T \backslash \alpha$	15	20	25	30	35	40	45	50	55	60	65
65	71,2	71,3	70,3	65,4	73,6	72,3	71,7	71,1	72,8	73,9	72,8
70	71,6	71,8	71,2	66,4	72,9	74,4	72,9	72,3	73,0	74,2	74,9
75	70,2	68,6	69,9	65,4	73,5	73,1	71,1	71,4	74,7	75,9	73,3
80	67,8	68,2	67,4	64,5	73,2	71,1	70,9	70,5	74,0	73,3	72,7
85	64,1	66,8	65,7	64,5	69,6	68,6	68,7	69,3	70,7	70,0	72,2
90	61,3	63,9	63,8	60,5	64,6	64,3	65,5	64,6	64,0	69,9	68,4
95	54,5	56,7	57,2	54,0	56,8	56,7	58,2	58,1	59,5	59,7	61,2

На рис. 2 изображены фрагменты обработанных изображений с разрешением 4K из подготовленной базы данных. С помощью предложенного алгоритма детектировано девять объектов (рис. 2, а), минимальный размер одного из них $[33 \times 13]$, а применение СНС YOLO v3 позволило обнаружить объект размером $[149 \times 54]$ (рис. 2, б).



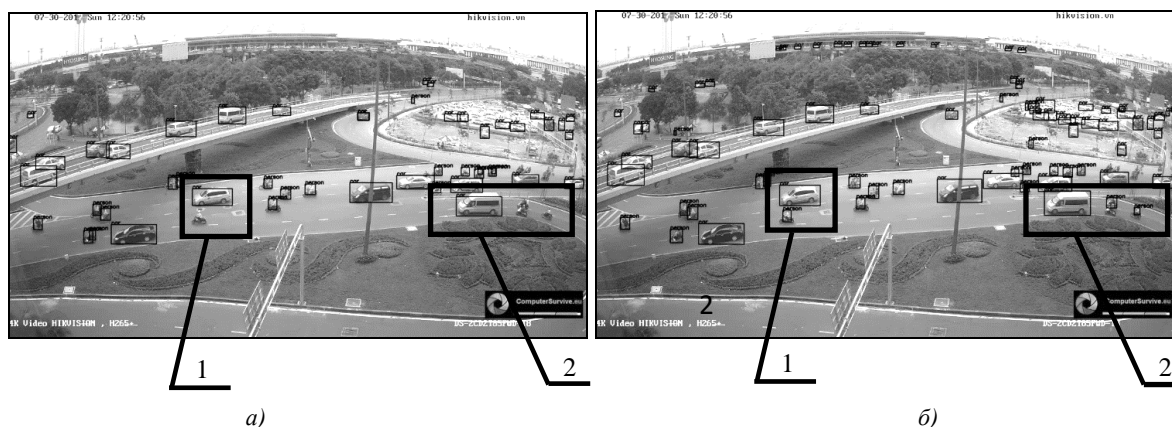
а)



б)

Рис. 2. Примеры обнаружения объектов на изображениях с разрешением 4K: а) с помощью предложенного алгоритма; б) с помощью алгоритма СНС YOLO v3

Тестирование результативности работы предложенного алгоритма также выполнено и на других 4K-изображениях, которые не были включены в аннотированную базу данных. На рис. 3, а показан пример обнаружения объектов на видеокadre (URL: https://www.youtube.com/watch?v=nG_wGUi-Ozc) с разрешением 4K, применением СНС YOLO v3, размером входного слоя $[1024 \times 1024]$ и $T = 50\%$. При этом обнаружено 50 объектов, минимальный размер детектированного объекта составляет $[41 \times 20]$ пикселей.



а)

б)

Рис. 3. Примеры обнаружения объектов на видеокadre с разрешением 4K: а) с помощью алгоритма СНС YOLO v3; б) с помощью предложенного алгоритма

На рис. 3, б представлен результат детектирования объектов предложенным алгоритмом с определенными в результате экспериментов параметрами: размером входного слоя СНС [1024×1024], количеством уровней пирамиды $P=3$, $\alpha=60\%$ и $T=75\%$. При этом обнаружено 80 объектов, их минимальный размер составляет [33×11] пикселей. На рис. 3 видно, что наибольший выигрыш в детектировании достигается именно для объектов небольших размеров, которые на данном изображении размещены в верхней и правой частях изображения.

Анализ результатов детектирования на различных масштабах изображения (рис. 4, а–в) и объединения областей (рис. 4, г) для увеличенного фрагмента 2 из рис. 3, б с тремя объектами подтверждает, что применение пирамидальной обработки позволяет обнаружить объекты различного размера на разных уровнях и повышает точность алгоритма в целом. СНС YOLO v3 не детектировала человека из выделенного фрагмента (см. рис. 3, а).



Рис. 4. Примеры обнаружения объектов на разных уровнях пирамиды для изображения размером [3840×216]: а) на исходном изображении; б) на втором уровне [1920×1080]; в) на третьем уровне [960×540]; г) итоговый результат

На рис. 5 показан пример объединения областей-кандидатов на четвертом шаге алгоритма для фрагмента 1 из рис. 3, б, который свидетельствует об эффективности этого этапа. После третьего шага алгоритма обнаружено шесть областей-кандидатов (рис. 5, а), применение процедуры объединения позволило правильно локализовать человека и улучшить результат для автомобиля (рис. 5, б), а дальнейшая постобработка дала возможность отфильтровать ошибочную локализацию его передней части. Модель СНС YOLO v3 не обнаружила человека в данной области видеокadra (см. рис. 3, а).

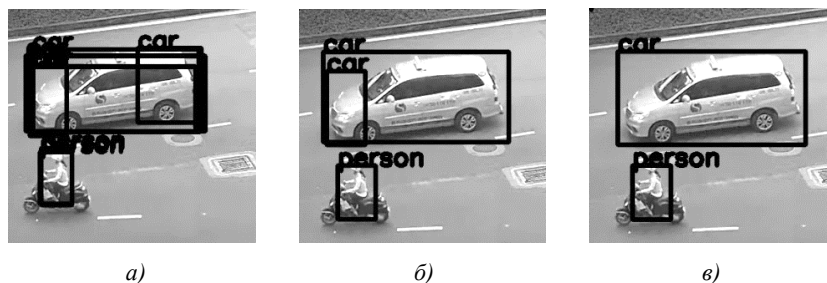


Рис. 5. Примеры объединения областей-кандидатов: а) результат обнаружения областей-кандидатов на всех уровнях пирамиды; б) результат применения объединения (см. шаг 4.1 алгоритма); в) результат постобработки (см. шаг 4.2 алгоритма)

Заключение. В статье предложен алгоритм для обнаружения объектов на изображениях с большим разрешением (4К и выше), включающий: формирование пирамиды изображений, пока размеры верхнего уровня не будут приближены к размерам входного слоя используемой СНС; блочное разбиение с перекрытием для всех полученных слоев; обнаружение объектов с применением СНС в каждом блоке; постобработку для полученных результатов на предыдущем шаге. Для детектирования объектов была использована СНС YOLO v3 с размерами входного слоя [1024×1024], который обеспечивает высокую результативность обнаружения небольших объектов на изображениях с разрешением 4К. Программная реализация разработанного алгоритма выполнена на языке Python с применением библиотеки компьютерного зрения OpenCV, фреймворка машинного обучения Pytorch и технологии CUDA. Для проведения экспериментов по оценке эффективности предложенного алгоритма подготовлена база данных изображений с разрешением 4К с размеченными объектами небольших и средних размеров классов «человек» и «транспортное средство».

Проведенные эксперименты показали, что для используемых данных значение метрики mAP может достигать 75,9%. Соответственно, рассмотренный подход является перспективным при

решении прикладных задач обнаружения объектов малых размеров на изображениях с большим разрешением.

Список литературы

1. Дворкович, А. В. Метрологическое обеспечение видеoinформационных систем / А. В. Дворкович, В. П. Дворкович. – М. : Техносфера, 2015. – 784 с.
2. Goulekas, K. *Visual Effects in a Digital World: a Comprehensive Glossary of over 7,000 Visual Effects Terms* / K. Goulekas. – San Francisco : Morgan Kaufmann, 2001. – 600 p.
3. An effective object detection algorithm for high resolution video by using convolutional neural network / D. Vorobjov [et al.] // *Advances in Neural Networks-ISNN2018. Lecture Notes in Computer Science*. – 2018. – Vol. 10878. – P. 503–510. https://doi.org/10.1007/978-3-319-92537-0_58
4. Yongxi, L. Efficient object detection for high resolution images / L. Yongxi, T. Javidi // *Proc. of 53rd Annual Allerton Conf. on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 30 Sept. – 2 Oct. 2015. – Monticello, 2015. – P. 1091–1098. <https://doi.org/10.1109/ALLERTON.2015.7447130>*
5. Sun database: large-scale scene recognition from abbey to zoo / J. Xiao [et al.] // *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010. – San Francisco, 2010. – P. 3485–3492. <https://doi.org/10.1109/CVPR.2010.5539970>*
6. Ruzicka, V. Fast and accurate object detection in high resolution 4K and 8K video using GPUs / V. Ruzicka, F. Franchetti // *Proc. of 2018 IEEE High Performance Extreme Computing Conf. (HPEC), Waltham, MA, USA, 25–27 Sept. 2018. – Waltham, 2018. – P. 1–7. <https://doi.org/10.1109/HPEC.2018.8547574>*
7. Korshunov, P. UHD video dataset for evaluation of privacy / P. Korshunov, T. Ebrahimi // *Proc. of Sixth Intern. Workshop on Quality of Multimedia Experience (QoMEX), Singapore, 18–20 Sept. 2014. – Singapore, 2014. – P. 232–237. <https://doi.org/10.1109/QoMEX.2014.6982324>*
8. Unel, F. O. The power of tiling for small object detection / F. O. Unel, B. Ozkalayci, C. Çigla // *CVPR Workshops [Electronic resource]. – 2019. – Mode of access: http://openaccess.thecvf.com/content_CVPRW_2019/papers/UAVision/Unel_The_Power_of_Tiling_for_Small_Object_Detection_CVPRW_2019_paper.pdf. – Date of access: 18.01.2020.*
9. Region-based convolutional networks for accurate object detection and segmentation / R. Girshick [et al.] // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. – 2016. – Vol. 38. – P. 142–158. <https://doi.org/10.1109/TPAMI.2015.2437384>
10. Deep residual learning for image recognition / K. He [et al.] // *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA, 27–30 June 2016. – Las Vegas, 2016. – P. 770–778. <https://doi.org/10.1109/CVPR.2016.90>*
11. You only look once: unified, real-time object detection / J. Redmon [et al.] // *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA, 27–30 June 2016. – Las Vegas, 2016. – P. 779–788. <https://doi.org/10.1109/CVPR.2016.91>*
12. Girshick, R. Fast R-CNN / R. Girshick // *Proc. of IEEE Intern. Conf. on Computer Vision (ICCV), Santiago, Chile, 11–18 Dec. 2015. – Santiago, 2015. – P. 1440–1448. <https://doi.org/10.1109/ICCV.2015.169>*
13. Faster R-CNN: towards real-time object detection with region proposal networks / S. Ren [et al.] // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. – 2015. – Vol. 39, no. 6. – P. 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
14. Глубокое обучение для детектирования объектов на изображениях документов / А. А. Крощенко и др. // *Вестник БрГТУ. Физика, математика, информатика*. – 2017. – № 5(107). – С. 2–9.
15. Inception-v4, inception-ResNet and the impact of residual connections on learning / C. Szegedy [et al.] // *Proc. of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), San Francisco, California, USA, 4–9 Febr. 2017. – San Francisco, 2017. – P. 4278–4284.*
16. The pascal Visual Object Classes (VOC) challenge / M. Everingham [et al.] // *Intern. J. of Computer Vision*. – 2010. – Vol. 88. – P. 303–338. <https://doi.org/10.1007/s11263-009-0275-4>

References

1. Dvorkovich A. V., Dvorkovich V. P. *Metrologicheskoe obespechenie videoinformatsionnykh system. Metrological Support of Video Information Systems*, Moscow, Technosfera, 2015, 784 p. (in Russian).
2. Goulekas K. *Visual Effects in a Digital World: a Comprehensive Glossary of over 7,000 Visual Effects Terms*. San Francisco, Morgan Kaufmann, 2001, 600 p.
3. Vorobjov D., Zakharova I., Bohush R., Ablameyko S. An effective object detection algorithm for high resolution video by using convolutional neural network. *Advances in Neural Networks-ISNN2018. Lecture Notes in Computer Science*, 2018, vol. 10878, pp. 503–510. https://doi.org/10.1007/978-3-319-92537-0_58

4. Yongxi L., Javidi T. Efficient object detection for high resolution images. *Proceedings of 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 30 September – 2 October 2015*. Monticello, 2015, pp. 1091–1098. <https://doi.org/10.1109/ALLERTON.2015.7447130>
5. Xiao J., Hays J., Ehinger K., Oliva A., Torralba A. Sun database: large-scale scene recognition from abbey to zoo. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010*. San Francisco, 2010, pp. 3485–3492. <https://doi.org/10.1109/CVPR.2010.5539970>
6. Ruzicka V., Franchetti F. Fast and accurate object detection in high resolution 4K and 8K video using GPUs. *Proceedings of 2018 IEEE High Performance Extreme Computing Conference (HPEC), Waltham, MA, USA, 25–27 September 2018*. Waltham, 2018, pp. 1–7. <https://doi.org/10.1109/HPEC.2018.8547574>
7. Korshunov P., Ebrahimi T. UHD video dataset for evaluation of privacy. *Proceedings of Sixth International Workshop on Quality of Multimedia Experience (QoMEX), Singapore, 18–20 September 2014*. Singapore, 2014, pp. 232–237. <https://doi.org/10.1109/QoMEX.2014.6982324>
8. Unel F. O., Ozkalayci B., Çigla C. The power of tiling for small object detection. *CVPR Workshops, 2019*. Available at: http://openaccess.thecvf.com/content_CVPRW_2019/papers/UAVision/Unel_The_Power_of_Tiling_for_Small_Object_Detection_CVPRW_2019_paper.pdf. (accessed 18.01.2020).
9. Girshick R., Donahue J., Darrell T., Malik J. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016, vol. 38, pp. 142–158. <https://doi.org/10.1109/TPAMI.2015.2437384>
10. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA, 27–30 June 2016*. Las Vegas, 2016, pp. 770–778. <https://doi.org/10.1109/CVPR.2016.90>
11. Redmon J., Divvala S. K., Girshick R. B., Farhadi A. You only look once: unified, real-time object detection. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA, 27–30 June 2016*. Las Vegas, 2016, pp. 779–788. <https://doi.org/10.1109/CVPR.2016.91>
12. Girshick R. Fast R-CNN. *Proceedings of IEEE Intern. Conf. on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015*. Santiago, 2015, pp. 1440–1448. <https://doi.org/10.1109/ICCV.2015.169>
13. Ren S., He K., Girshick R., Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, vol. 39, no. 6, pp. 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
14. Kroshchenko A., Golovko V., Bezobrazov S., Mikhno E., Khatskevich M, ..., Brich A. Glubokoe obuchenie dlia detektirovaniia obiektov na izobrazheniakh dokumentov [Deep training for detecting of objects at images of documents]. *Vestnik Brestskogo gosudarstvennogo tekhnicheskogo universiteta. Fizika, matematika, informatika [Bulletin of the Brest State Technical University. Physics, mathematics, Computer Science]*, 2017, vol. 5 (107), pp. 2–9 (in Russian).
15. Szegedy C., Ioffe S., Vanhoucke V. Inception-v4, inception-ResNet and the impact of residual connections on learning. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), San Francisco, California, USA, 4–9 February 2017*. San Francisco, 2017, pp. 4278–4284.
16. Everingham M., Van Gool L., Williams C., Winn J., Zisserman A. The pascal Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, 2010, vol. 88, pp. 303–338. <https://doi.org/10.1007/s11263-009-0275-4>

Информация об авторах

Богуш Рихард Петрович, кандидат технических наук, доцент, заведующий кафедрой вычислительных систем и сетей, факультет информационных технологий, Полоцкий государственный университет, Полоцк, Беларусь.
E-mail: bogushr@mail.ru

Захарова Ирина Юрьевна, магистр технических наук, аспирант кафедры вычислительных систем и сетей, факультет информационных технологий, Полоцкий государственный университет, Полоцк, Беларусь.
E-mail: ira9992011@yandex.ru

Аблameйко Сергей Владимирович, академик Национальной академии наук Беларуси, доктор технических наук, профессор, профессор механико-математического факультета, Белорусский государственный университет, Минск, Беларусь.
E-mail: ablameyko@bsu.by

Information about the authors

Rykhard P. Bohush, Cand. Sci. (Eng.), Associate Professor, Head of the Department of Computer Systems and Networks, Polotsk State University, Novopolotsk, Belarus.
E-mail: bogushr@mail.ru

Iryna Yu. Zakharava, M. Sci. (Eng.), Postgraduate Student at the Department of Computer Systems and Networks, Polotsk State University, Novopolotsk, Belarus.
E-mail: ira9992011@yandex.ru

Sergey V. Ablameyko, Academician of the National Academy of Sciences of Belarus, Dr. Sci. (Eng.), Professor, Professor of the Faculty of Mechanics and Mathematics, Belarusian State University, Minsk, Belarus.
E-mail: ablameyko@bsu.by

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

УДК 004.021
<https://doi.org/10.37661/1816-0301-2020-17-2-17-24>

Поступила в редакцию 15.01.2020
Received 15.01.2020

Принята к публикации 25.02.2020
Accepted 25.02.2020

Визуальная навигация автономно летящего БПЛА с целью его возвращения в точку старта

Р. С. Жук, Б. А. Залеский[✉], Ф. С. Троицкий

*Объединенный институт проблем информатики
Национальной академии наук Беларуси, Минск, Беларусь*
[✉]E-mail: zalesky@newman.bas-net.by

Аннотация. Рассматривается алгоритм автономной визуальной навигации, предназначенный для возвращения в точку старта беспилотного летательного аппарата (БПЛА), оборудованного одной бортовой видеокамерой и бортовым вычислителем, без использования навигационных сигналов GPS и ГЛОНАСС. Предлагаемый алгоритм схож с широко известными алгоритмами визуальной навигации, такими как одновременная локализация и картографирование (V-SLAM) и визуальная одометрия, однако отличается от них раздельным выполнением процессов картографирования и локализации. Он вычисляет географические координаты признаков, найденных на кадрах, снятых бортовой видеокамерой при полете от точки старта до потери сигналов GPS и ГЛОНАСС. После потери сигнала запускается миссия возвращения и вычисляется лишь положение БПЛА относительно построенной на основе найденных ранее признаков карты, которая используется для возвращения в точку старта. Предложенный подход не требует таких сложных вычислений, как V-SLAM, и не накапливает со временем ошибки в отличие от визуальной одометрии и традиционных методов инерциальной навигации. Алгоритм был реализован и протестирован с помощью квадрокоптера DJI Phantom 3 Pro.

Ключевые слова: автономная визуальная навигация, беспилотные летательные аппараты, визуальная одометрия, возвращение в точку старта, навигационные сигналы

Для цитирования. Жук, Р. С. Визуальная навигация автономно летящего БПЛА с целью его возвращения в точку старта / Р. С. Жук, Б. А. Залеский, Ф. С. Троицкий // Информатика. – 2020. – Т. 17, № 2. – С. 17–24. <https://doi.org/10.37661/1816-0301-2020-17-2-17-24>

Solution of problem of returning to the starting point of autonomously flying UAV by visual navigation

Raman S. Zhuk, Boris A. Zalesky[✉], Philip S. Trotski

*The United Institute of Informatics Problems of the National Academy
of Sciences of Belarus, Minsk, Belarus*
[✉]E-mail: zalesky@newman.bas-net.by

Abstract. An autonomous visual navigation algorithm is considered, designed for “home” return of unmanned aerial vehicle (UAV) equipped with on-board video camera and on-board computer, out of GPS and GLONASS navigation signals. The proposed algorithm is similar to the well-known visual navigation algorithms such as V-SLAM (simultaneous localization and mapping) and visual odometry, however, it differs in separate implementation of mapping and localization processes. It calculates the geographical coordinates of the features on the frames taken by on-board video camera during the flight from the start point until the moment of GPS and GLONASS signals loss. After the loss of the signal the return mission is launched, which provides estimation of the position of UAV relatively the map created by previously found features. Proposed approach does not require such complex calculations as V-SLAM and does not accumulate errors over time, in contrast to visual

odometry and traditional methods of inertial navigation. The algorithm was implemented and tested with use of DJI Phantom 3 Pro quadcopter.

Keywords: autonomous visual navigation, unmanned aerial vehicles, visual odometry, return to the starting point, navigation signals

For citation. Zhuk R. S., Zalesky B. A., Trotski Ph. S. Solution of problem of returning to the starting point of autonomously flying UAV by visual navigation. *Informatics*, 2020, vol. 17, no. 2, pp. 17–24 (in Russian). <https://doi.org/10.37661/1816-0301-2020-17-2-17-24>

Введение. В настоящее время актуальна задача визуальной навигации БПЛА в условиях отсутствия внешних навигационных сигналов GPS или ГЛОНАСС, так как большинство аппаратов оснащается недостаточно точными инерционными датчиками.

Обычно для визуальной навигации используются одна или несколько видеокамер, поэтому в последние несколько десятилетий задача оценки движения камеры в пространстве только по потоку изображений является объектом активных исследований. На сегодняшний день хорошо развиты два подхода к решению этой задачи: V-SLAM [1, 2] и визуальная одометрия [3, 4]. Современные версии упомянутых алгоритмов показывают хорошие результаты и могут работать в реальном времени [5, 6].

Подход V-SLAM основан на идее построения глобальной карты пространственных положений видимых на кадрах видеопотока признаков, согласованной с положениями камеры на протяжении всего времени движения. Он хорошо зарекомендовал себя при навигации в закрытых помещениях и по замкнутым траекториям. Метод визуальной одометрии осуществляет постепенное восстановление траектории (от точки к точке), заботясь только о ее локальной согласованности (о расположении текущей точки траектории относительно предыдущих ее точек в локальной системе координат при неизвестных глобальных координатах), для чего строится лишь локальная карта пространственных положений признаков. Подход V-SLAM гораздо сложнее. Он потенциально более точен, потому что налагает намного больше ограничений на неизвестные положения камеры, но не обязательно более надежен. Например, ошибки при обнаружении точек пересечений траектории могут серьезно повлиять на согласованность всей карты. В случае длинной непересекающейся траектории БПЛА метод V-SLAM не дает преимуществ по сравнению с визуальной одометрией. Очевидным недостатком подхода визуальной одометрии является рост ошибок с течением времени. Самая большая проблема обоих алгоритмов заключается в невозможности оценить масштаб сцены в случае использования лишь одной камеры без дополнительных датчиков.

В последнее время были разработаны несколько оригинальных методов визуально-инерциальной одометрии [7, 8]. В этих методах данные с инерциальных датчиков используются как для уточнения относительных перемещений камеры, так и для определения масштаба сцены. Такие методы сложнее в реализации, чем чисто визуальные, но гораздо точнее и надежнее. Вместе с тем, несмотря на высокую точность, с течением времени они неизбежно накапливают ошибки при полете БПЛА по длинным траекториям.

В настоящей статье представлен метод визуальной навигации, предназначенный для решения задачи возвращения в точку старта БПЛА без использования внешних навигационных сигналов GPS и ГЛОНАСС. Для решения этой задачи строится глобальная карта пространственных положений признаков, видимых на кадрах видеопотока, полученного в то время, когда внешние навигационные сигналы еще присутствуют. После потери сигнала GPS и ГЛОНАСС осуществляется лишь локализация БПЛА относительно построенной карты и возвращение его в точку старта. В результате не производятся сложные вычисления, связанные со взаимным согласованием положений всех признаков с положениями камер. Положение каждого признака вычисляется независимо на основании известных положений камер, в которых он наблюдался. Поскольку глобальная карта построена относительно известных положений камер, дальнейшая визуальная навигация относительно нее осуществляется без накопления ошибок и в известном масштабе.

Предложенный метод был протестирован при помощи квадрокоптера DJI Phantom 3 Pro. Вычислительные эксперименты показали надежность и достаточную точность алгоритма.

Постановка задачи. Рассматривается задача полета БПЛА от точки старта до момента исчезновения используемого навигационного сигнала GPS либо ГЛОНАСС и его возвращения в точку старта в автономном режиме. Во время полета БПЛА от точки старта при наличии навигационного сигнала (до момента его исчезновения) алгоритм строит глобальную карту местности, отображенной бортовой видеочамерой, путем вычисления географических координат особых точек [9], найденных на кадрах. После исчезновения сигнала с помощью этих точек осуществляется навигация БПЛА для его возвращения домой. Для построения глобальной карты местности используются не все кадры, а только так называемые перекрывающиеся ключевые кадры. В алгоритме в качестве ключевого использовался каждый μ -й кадр (с наперед заданным μ). Основное требование при выборе μ заключается в том, что размер в пикселях области, видимой на всех перекрывающихся кадрах, должен быть не меньше заданной части кадра, например четверти, трети и т. д.

Пусть $\Omega = \{I_1, \dots, I_n\}$ – последовательность ключевых кадров, полученных с бортовой камеры БПЛА при полете в условиях присутствия внешних навигационных сигналов GPS или ГЛОНАСС. Для каждого кадра $I_k \in \Omega$ обозначим через T_k и \mathbf{R}_k соответственно вектор перемещения и матрицу поворота, определяющие положение камеры в пространстве относительно неподвижной глобальной системы координат в тот момент, когда был сделан кадр I_k . Точка π на кадре I_k , соответствующая 3D-точке $X = (x, y, z)$ в пространстве с проективными координатами $\tilde{X} = (x, y, z, 1)$, которая является функцией от переменных $\mathbf{R}_k, T_k, \tilde{X}$, определяется согласно выражению [10]

$$\pi(\mathbf{R}_k, T_k, \tilde{X}) = \frac{1}{z} \cdot \mathbf{K} \cdot \mathbf{R}_k \cdot (\mathbf{I}, -T_k) \cdot \tilde{X},$$

где \mathbf{K} – калибровочная матрица бортовой камеры БПЛА, \mathbf{I} – единичная матрица.

Описание алгоритма автономной визуальной навигации. Предложенный алгоритм визуальной навигации состоит из трех больших блоков шагов. На рис. 1 приведена блок-схема алгоритма, на которой первый блок шагов обведен зеленой, второй – синей, а третий – оранжевой рамками.

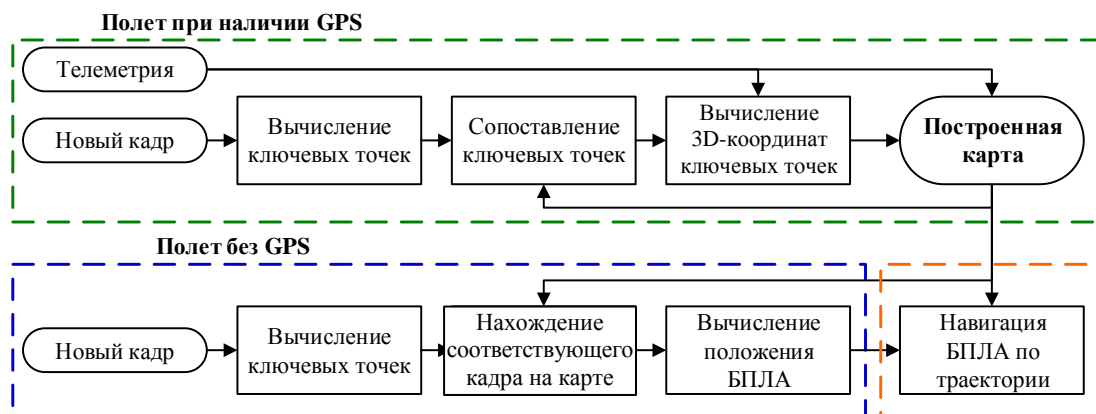


Рис. 1. Блок-схема алгоритма автономной визуальной навигации

Задачей первого блока алгоритма является определение для каждого кадра $I_k \in \Omega$ пространственных координат точек ландшафта $\mathbf{X}^k = \{X_1^k, X_2^k, \dots\}$, соответствующих найденным на нем особым точкам $\mathbf{u}^k = \{u_1^k, u_2^k, \dots\}$.

Для нахождения ключевых точек на кадрах использовались как известные детекторы особых точек SURF, SIFT, AKAZE, так и специально разработанный детектор экстремальных моментов EMD [9]. Проведенные эксперименты показали, что детекторы SURF и EMD обеспечивают нахождение большего количества информативных ключевых точек и поэтому лучше подходят для решения рассматриваемой задачи.

Основным этапом первого блока алгоритма является нахождение множества 3D-точек ландшафта $\Theta = \{X_1, \dots, X_m\}$, видимых на нескольких ключевых кадрах. Вначале на каждом новом кадре I_k ищутся ключевые точки \mathbf{u}^k , которые сопоставляются с ключевыми точками \mathbf{u}^{k-1} с предыдущего шага. Далее для каждой сопоставленной пары ключевых точек $\{u_i^{k-1}, u_j^k\}$ точке u_j^k ставится в соответствие точка X_l , если ранее уже было установлено соответствие между X_l и u_i^{k-1} , либо обоим ключевым точкам ставится в соответствие новая 3D-точка, которая добавляется в множество Θ . В результате для каждой 3D-точки $X_l \in \Theta$ формируется набор ключевых точек $\mathbf{U}^l = \{u_i^s, \dots, u_j^d\}$ из некоторого количества ключевых кадров множества Ω , а также для каждого кадра I_k – набор 3D-точек $\hat{\mathbf{X}}^k = \mathbf{X}^k \cap \Theta$, которым поставлены в соответствие некоторые ключевые точки из \mathbf{u}^k . После этого каждая точка X_l находится из условия

$$\begin{cases} \pi(\mathbf{R}_s, T_s, \tilde{X}_l) - \tilde{u}_i^s = 0, \\ \vdots \\ \pi(\mathbf{R}_d, T_d, \tilde{X}_l) - \tilde{u}_j^d = 0, \end{cases}$$

где \tilde{u}_i^s , \tilde{u}_j^d и \tilde{X}_l – проективные координаты точек u_i^s , u_j^d и X_l соответственно.

Чтобы повысить точность вычисления координат точек множества Θ в более поздней версии алгоритма, была использована усложненная схема сопоставления ключевых точек на кадрах, в которой ключевые точки нового кадра \mathbf{u}^k сопоставлялись с ключевыми точками нескольких предыдущих кадров. В результате был получен надежный метод отслеживания точек ландшафта на ключевых кадрах, которые брались с довольно большим шагом по времени.

Для нахождения координат точек \mathbf{X}^k , соответствующих ключевым точкам каждого кадра $I_k \in \Omega$, сначала оценивались параметры 3D-поверхности первого или второго порядка, аппроксимирующей рельеф, путем приближения точек из $\hat{\mathbf{X}}^k$, после чего находились X_i^k как точки пересечения луча, выходящего из оптического центра камеры и проходящего через u_i^k с найденной 3D-поверхностью.

Второй блок шагов алгоритма автономной визуальной навигации предназначен для нахождения пространственных координат камеры БПЛА во время автономного полета.

Пусть I^t – кадр, полученный при возвращении БПЛА в точку старта. Необходимо вычислить вектор трансляции T^t и матрицу поворота \mathbf{R}^t камеры в момент времени t . Для этого следует найти сначала особые точки \mathbf{u}^t на кадре I^t , а затем полученный при полете от точки старта при наличии навигационных сигналов кадр $I(t) \in \Omega$, между ключевыми точками которого и \mathbf{u}^t удастся найти достаточное количество соответствий. В качестве начального кадра для поиска $I(t)$ берется либо $I(t-1) \in \Omega$ (кадр, обнаруженный на предыдущем шаге), либо послед-

ний кадр Ω , если I^t – первый кадр, полученный при условии отсутствия сигналов GPS или ГЛОНАСС.

Пусть далее $I(t) = I_k$, а $\tilde{\mathbf{u}}^t$ и $\tilde{\mathbf{u}}^k$ – два набора соответствующих друг другу проективных координат особых точек из I^t и I_k , $\tilde{\mathbf{X}}^k$ – набор проективных координат 3D-точек ландшафта, которые соответствуют $\tilde{\mathbf{u}}^k$. Тогда \mathbf{R}^t можно найти при помощи восьмиточечного [11] либо пятиточечного [12] алгоритма, используя только $\tilde{\mathbf{u}}^t$ и $\tilde{\mathbf{u}}^k$, после чего вектор сдвига T^t находится следующим образом:

$$T^t = \operatorname{argmin}_T \sum_i \left\| \tilde{\mathbf{u}}_i^t - (\mathbf{R}^t, T, \tilde{\mathbf{X}}_i^k) \right\|^2.$$

Третий блок алгоритма в каждый момент времени t , когда получена новая оценка T^t положения БПЛА, находит новый вектор D^t направления движения БПЛА, чтобы осуществить его полет к точке старта вдоль сохраненной траектории $\mathbf{T} = \{T_1, \dots, T_n\}$ полета летательного аппарата от точки старта. Эта простая на первый взгляд задача не является тривиальной в случае запутанного маршрута БПЛА (рис. 2).

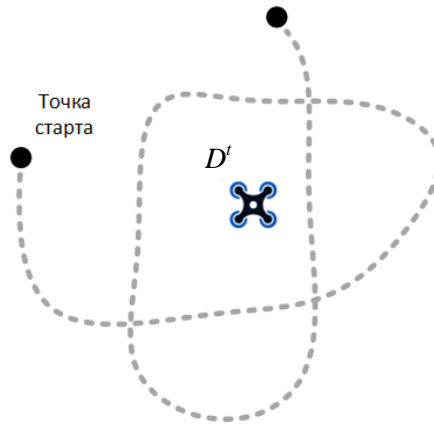


Рис. 2. Пример запутанного маршрута БПЛА

В рассматриваемом алгоритме направление движения на каждом шаге определяется как $D^t = T(t) - T^t$, где $T(t)$ – некоторая специально выбранная точка на траектории \mathbf{T} . Для ее выбора находится T_k – ближайшая точка на траектории \mathbf{T} к позиции T^t , а затем для некоторого наперед заданного расстояния L (в экспериментах были использованы значения от 5 до 20 в зависимости от скорости БПЛА и сложности маршрута) проверяется условие $\|T_k - T^t\| > L$. Если условие выполняется, то $T(t) = T_k$. В противном случае ищется минимальное число l , для которого выполняется условие $\|T_{k-l} - T^t\| > L$, и точка $T(t)$ полагается равной $T(t) = T_{k-l+1}$.

В случае сложной траектории \mathbf{T} может получиться так, что $T(t) = T_i$, а $T(t-1) = T_j$, где $i > j$. Тогда для предотвращения закливания полета необходимо положить $T(t) = T(t-1)$.

Достаточно большое количество экспериментов показало применимость разработанного подхода.

Описание и результаты экспериментов. Для тестирования алгоритма визуальной навигации был использован квадрокоптер DJI Phantom 3 Pro. Так как он не оборудован бортовым компьютером, для проведения испытаний было разработано экспериментальное программное обеспечение, которое позволяло осуществлять управление квадрокоптером с помощью персонального компьютера, а также получать с квадрокоптера телеметрию и видео. Было проведено большое количество экспериментов, в каждом из которых согласно предложенному алгоритму во время полета от точки старта оценивались и сохранялись 3D-координаты точек ландшафта, соответствующие особым точкам на ключевых кадрах, которые брались с определенным шагом.

После достижения БПЛА конечной точки маршрута его координаты оценивались с помощью алгоритма в автономном режиме (без использования данных GPS и ГЛОНАСС), и на основе построенных оценок координат осуществлялось возвращение аппарата в начальную точку. Для повышения стабильности автономного полета БПЛА был использован линейный фильтр Калмана.

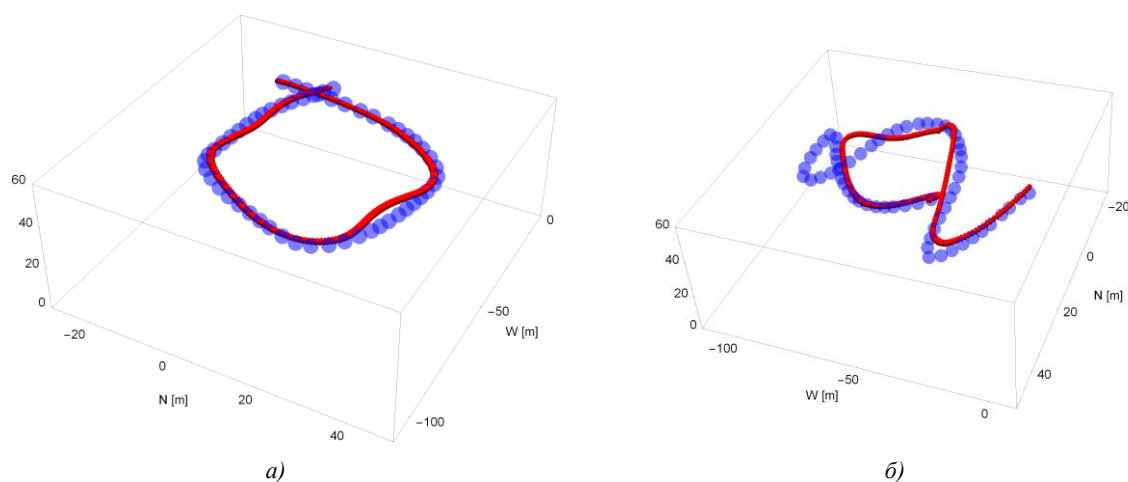


Рис. 3. 3D-реконструкции простых (а) и сложных (б) траекторий экспериментальных полетов БПЛА

На рис. 3 показаны 3D-реконструкции траекторий некоторых из экспериментальных полетов БПЛА, где синим цветом отображена траектория полета от точки старта, а красным – траектория автономного возвращения аппарата в точку старта. На рис. 3, б видно, что алгоритм предотвратил зацикливание траектории на этапе возвращения БПЛА в точку старта при наличии петли в траектории полета от точки старта. Оценки точности вычисления координат алгоритмом во время автономных полетов квадрокоптера приведены в таблице.

Результаты экспериментов, м

Длина маршрута	Высота над уровнем моря	Средняя ошибка с фильтром Калмана	Средняя ошибка без фильтра Калмана	Стандартное отклонение с фильтром Калмана	Стандартное отклонение без фильтра Калмана
200	46	1,43	0,88	0,73	0,45
215	65	2,04	1,15	1,05	0,58
230	46	2,58	1,06	1,47	0,60
261	50	2,11	2,87	1,87	1,76
285	61	0,86	2,29	0,79	2,01
294	46	6,29	4,04	7,78	7,53
300	49	0,17	3,19	0,25	3,06
305	46	2,79	0,94	1,62	0,43
340	65	2,03	1,56	1,89	0,92
481	46	3,66	2,66	5,21	5,01
591	63	0,31	2,47	0,50	2,91
700	49	0,50	2,62	1,82	3,70
1040	61	2,90	4,50	5,34	6,83

Средняя ошибка оценки глобальных координат автономно летящего БПЛА, измеренная по всем экспериментальным маршрутам, составляет 2,07 м. Следует отметить, что величина ошибки не увеличивается со временем, поскольку оценки координат во время автономного полета вычисляются независимо друг от друга.

Заключение. Предложенный алгоритм позволяет вернуть потерявший внешние навигационные сигналы БПЛА в точку старта на основе сохраненных во время полета визуальных признаков. Он отличается простотой и надежностью. Величина погрешности оценки траектории, построенной разработанным алгоритмом, не возрастает в отличие от погрешности оценки траектории алгоритмами инерциальной навигации.

Недостатком предложенного авторами подхода к решению поставленной задачи автономного возвращения в точку старта является необходимость пролета по всему маршруту, проделанному БПЛА до момента потери внешних навигационных сигналов. Этот недостаток можно устранить путем добавления возможности срезать траекторию, для чего необходимо будет некоторое время лететь над неизвестным ландшафтом. Указанную возможность может обеспечить добавление в предложенную систему модулей, реализующих известные алгоритмы визуальной навигации, такие как V-SLAM и визуальная одометрия.

Список использованных источников

1. Durrant-Whyte, H. Simultaneous localization and mapping: part I / H. Durrant-Whyte, T. Bailey // *IEEE Robotics and Automation Magazine*. – 2006. – Vol. 13, no. 2. – P. 99–110.
2. Bailey, T. Simultaneous localization and mapping (SLAM): part II / T. Bailey, H. Durrant-Whyte // *IEEE Robotics and Automation Magazine*. – 2006. – Vol. 13, no. 3. – P. 108–117.
3. Scaramuzza, D. Visual odometry [tutorial]. Part I: The first 30 years and fundamentals / D. Scaramuzza, F. Fraundorfer // *IEEE Robotics and Automation Magazine*. – 2011. – Vol. 18, no. 4. – P. 80–92.
4. Fraundorfer, F. Visual odometry: part II: matching, robustness, optimization, and applications / F. Fraundorfer, D. Scaramuzza // *IEEE Robotics and Automation Magazine*. – 2012. – Vol. 19, no. 2. – P. 78–90.
5. SVO: semidirect visual odometry for monocular and multicamera systems / C. Forster [et al.] // *IEEE Transactions on Robotics*. – 2017. – Vol. 33, no. 2. – P. 249–265.
6. Efficient on-board Stereo SLAM through constrained-covisibility strategies / G. Castro [et al.] // *Robotics and Autonomous Systems*. – 2019. – Vol. 116. – P. 192–205.
7. Qin, T. VINS-Mono: a robust and versatile monocular visual-inertial state estimator / T. Qin, P. Li, S. Shen // *IEEE Transactions on Robotics*. – 2017. – Vol. 34, no. 4. – P. 1004–1020.
8. VIMO: simultaneous visual inertial model-based odometry and force estimation / B. Nisar [et al.] // *IEEE Robotics and Automation Letters*. – 2019. – Vol. 4, no. 3. – P. 2785–2792.
9. Залесский, Б. А. Параллельная версия детектора экстремальных особых точек изображений / Б. А. Залесский, Ф. С. Троцкий // *Информатика*. – 2018. – Т. 15, № 2. – С. 55–63.
10. Форсайт, Д. А. Компьютерное зрение. Современный подход / Д. А. Форсайт, Ж. Понс. – М. : Вильямс, 2004. – С. 76–77.
11. Hartley, R. I. In defense of the eight-point algorithm / R. I. Hartley // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. – 1997. – Vol. 19, no. 6. – P. 580–593.
12. Nister, D. An efficient solution to the five-point relative pose problem / D. Nister // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. – 2004. – Vol. 26, no. 6. – P. 756–770.

References

1. Durrant-Whyte H., Bailey T. Simultaneous localization and mapping: part I. *IEEE Robotics and Automation Magazine*, 2006, vol. 13, no. 2, pp. 99–110.
2. Bailey T., Durrant-Whyte H. Simultaneous localization and mapping (SLAM): part II. *IEEE Robotics and Automation Magazine*, 2006, vol. 13, no. 3, pp. 108–117.
3. Scaramuzza D., Fraundorfer F. Visual odometry [tutorial]. Part I: The first 30 years and fundamentals. *IEEE Robotics and Automation Magazine*, 2011, vol. 18, no. 4, pp. 80–92.
4. Fraundorfer F., Scaramuzza D. Visual odometry: part II: matching, robustness, optimization, and applications. *IEEE Robotics and Automation Magazine*, 2012, vol. 19, no. 2, pp. 78–90.
5. Forster C., Zhang Z., Gassner M., Werlberger M., Scaramuzza D. SVO: semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 2017, vol. 33, no. 2, pp. 249–265.

6. Castro G., Nitsche M., Pire T., Fischer T., De Cristóforis P. Efficient on-board Stereo SLAM through constrained-covisibility strategies. *Robotics and Autonomous Systems*, 2019, vol. 116, pp. 192–205.
7. Qin T., Li P., Shen S. VINS-Mono: a robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 2017, vol. 34, no. 4, pp. 1004–1020.
8. Nisar B., Foehn P., Falanga D., Scaramuzza D. VIMO: simultaneous visual inertial model-based odometry and force estimation. *IEEE Robotics and Automation Letters*, 2019, vol. 4, no. 3, pp. 2785–2792.
9. Zalesky B. A., Trotski Ph. S. Parallelnaja versija detektora jekstremal'nyh osobyh toчек izobrazhenij [Parallel version of detector of extremal key points on images]. *Informatika [Informatics]*, 2018, vol. 15, no. 2, pp. 55–63 (in Russian).
10. Forsyth D. A., Ponce J. *Computer Vision: a Modern Approach*. Prentice Hall, 2003, 693 p.
11. Hartley R. I. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997, vol. 19, no. 6, pp. 580–593.
12. Nister D. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004, vol. 26, no. 6, pp. 756–770.

Информация об авторах

Жук Роман Сергеевич, младший научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси, Минск, Беларусь.

E-mail: ramanzhuck@gmail.com

Залесский Борис Андреевич, доктор физико-математических наук, заведующий лабораторией обработки и распознавания изображений, Объединенный институт проблем информатики Национальной академии наук Беларуси, Минск, Беларусь.

E-mail: zalesky@newman.bas-net.by

Троцкий Филипп Сергеевич, младший научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси, Минск, Беларусь.

E-mail: trotskiphilipp@gmail.com

Information about the authors

Raman S. Zhuk, Junior Researcher, the United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus.

E-mail: ramanzhuck@gmail.com

Boris A. Zalesky, Dr. Sci. (Phys.-Math.), Head of the Laboratory of Image Processing and Recognition, the United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus.

E-mail: zalesky@newman.bas-net.by

Philip S. Trotski, Junior Researcher, the United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus.

E-mail: trotskiphilipp@gmail.com

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

УДК 004.932
<https://doi.org/10.37661/1816-0301-2020-17-2-25-35>

Поступила в редакцию 26.01.2020
Received 26.01.2020

Принята к публикации 24.03.2020
Accepted 24.03.2020

Скелетизация изображений на основе комбинации одно- и двухподытерационных моделей

Ц. Ма, В. Ю. Цветков[✉], В. К. Конопелько

Белорусский государственный университет
информатики и радиоэлектроники, Минск, Беларусь
[✉]E-mail: vtsvet@bsuir.by

Аннотация. Рассматривается задача скелетизации бинарных изображений. Скелетизация дает возможность представить бинарное изображение в виде множества тонких линий, взаимное расположение, размеры и форма которых адекватно описывают размеры, форму и ориентацию в пространстве соответствующих областей изображения. Высокое качество скелетов обеспечивают итерационные параллельные алгоритмы. Они могут реализовываться с использованием одной или нескольких подытераций. На каждой из них происходит удаление избыточных элементов, окрестности которых удовлетворяют определенным условиям. Для многих одноподытерационных алгоритмов характерно нарушение связности и формирование избыточных фрагментов скелета. Наиболее качественные скелеты формирует известный одноподытерационный алгоритм ОПТА (One-Pass Thinning Algorithm), основанный на 18 бинарных масках, который, однако, чувствителен к контурному шуму и имеет высокую вычислительную сложность. Благодаря относительной простоте широко известность получил двухподытерационный алгоритм Zhang – Suen (ZS), основанный на шести логических условиях, но он размывает диагональные линии толщиной 2 пиксела и удаляет области размером 2×2 пиксела. Оба алгоритма не обеспечивают достижение минимальной толщины линий скелета (многие неузловые элементы имеют более двух соседей). Для построения предельно тонких связанных скелетов бинарных изображений с низкой вычислительной сложностью предлагаются математическая модель и алгоритм ОПСА (One-Pass Combination Algorithm) одноподытерационной скелетизации на основе комбинации и упрощения моделей одно- и двухподытерационной скелетизации. Данная модель и алгоритм позволяют повысить скорость скелетизации, восстановить исходное изображение по скелету, снизить избыточность связей элементов скелета.

Ключевые слова: скелетизация изображений, алгоритм ОПТА, алгоритм Zhang – Suen, одноподытерационная скелетизация, двухподытерационная скелетизация

Для цитирования. Ма, Ц. Скелетизация изображений на основе комбинации одно- и двухподытерационных моделей / Ц. Ма, В. Ю. Цветков, В. К. Конопелько // Информатика. – 2020. – Т. 17, № 2. – С. 25–35. <https://doi.org/10.37661/1816-0301-2020-17-2-25-35>

Image skeletonization based on combination of one- and two-sub-iterations models

Jun Ma, Viktor Yu. Tsviatkou[✉], Valery K. Kanapelka

Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus
[✉]E-mail: vtsvet@bsuir.by

Abstract. This paper is focused on the field of the skeletonization of the binary image. Skeletonization makes it possible to represent a binary image in the form of many thin lines, the relative position, sizes and shape of which adequately describe the size, shape and orientation in space of the corresponding image areas. Skeletonization has many variety methods. Iterative parallel algorithms provide high quality skeletons. They can be implemented using one or more sub-iterations. In each iteration, redundant pixels, the neighborhoods of which meet certain conditions,

are removed layer by layer along the contour and finally they leave only the skeleton. Many one-sub-iterations algorithms are characterized by a breakdown in connectivity and the formation of excess skeleton fragments. The highest-quality skeletons are formed by the well-known single-iteration OPTA algorithm, which based on 18 binary masks, but it is sensitive to contour noise and has a high computational complexity. The Zhang and Suen two-iteration algorithm (ZS), which is based on 6 logical conditions, is widely used due to its relative simplicity. But it suffers from the problem of the blurs of the diagonal lines with a thickness of 2 pixels and the lost of the square which size is 2×2 pixels. Besides, both algorithms mentioned above do not achieve the unit pixel thickness of the skeleton lines (many non-node pixels have more than two neighbors). Mathematical model and OPCA (One-Pass Combination Algorithm) algorithm which is based on a combination and simplification of single-iterative OPTA and two-iterative ZS are proposed for constructing extremely thin bound skeletons of binary images with low computational complexity. These model and algorithm also made it possible to accelerate the speed of skeletonization, to enhance recoverability of the original image on the skeleton and to reduce the redundancy of the bonds of the skeleton elements.

Keywords: image skeletonization, algorithm OPTA, algorithm Zhang – Suen, single-iteration skeletonization, double-iteration skeletonization

For citation. Ma J., Tsviatkou V. Yu., Kanapelka V. K. Image skeletonization based on combination of one- and two-sub-iterations models. *Informatics*, 2020, vol. 17, no. 2, pp. 25–35 (in Russian). <https://doi.org/10.37661/1816-0301-2020-17-2-25-35>

Введение. В задачах параметризации объектов изображений часто используется скелетизация (утонышение) – преобразование однородной области, соответствующей объекту, в множество тонких линий, взаимное расположение, размеры и форма которых передают информацию о размере, форме и ориентации в пространстве соответствующей области. Впервые задача утонышения областей бинарных изображений была рассмотрена в работе [1], где для распознавания образов предложен алгоритм на основе оконного усреднения и взвешивания значений пикселей. Затем в статье [2] был представлен алгоритм утонышения единичной области на основе постепенного вложения ее границ. Согласно предложенной в [3] классификации эти работы послужили началом развития двух основных групп алгоритмов скелетизации: неитерационных и итерационных. Неитерационные алгоритмы реализуют скелетизацию за один проход, что обеспечивает высокое быстродействие, но приводит к низкому качеству скелетов. Итерационные алгоритмы постепенно удаляют слои пикселей на границе области до получения тонких линий, что улучшает качество скелетов, но связано с высокой вычислительной сложностью. В свою очередь, итерационные алгоритмы разделяются на последовательные и параллельные. В последовательных алгоритмах результат скелетизации на каждой итерации зависит от порядка выборки пикселей, что приводит к неинвариантности скелета к повороту изображения. В параллельных алгоритмах порядок обработки пикселей на каждой итерации не влияет на результат, что обеспечивает стабильность скелета при повороте изображения и позволяет повысить скорость скелетизации за счет распараллеливания вычислений.

Параллельные алгоритмы разделяются по числу подытераций. В работе [4] предложен один из первых алгоритмов с одной подытерацией, использующий логические условия для определения удаляемых пикселей в окрестности 3×3 ; линий скелета, возникающих из-за незначительных изломов границ областей, и избыточных пикселей на диагональных линиях. Алгоритмы с одной подытерацией на основе логических условий и окрестности 3×3 пиксела предложены также в статьях [5, 6], где эти условия модифицируются для ускорения обработки. Известна также группа одноподытерационных алгоритмов [7–9] на основе масок различных размеров и ориентаций, берущих начало от работ [10, 11]. Алгоритмы данной группы за счет роста вычислительной сложности в некоторых случаях улучшают качество скелетов.

Проблемами многих одноподытерационных алгоритмов являются нарушение связности и формирование избыточных фрагментов скелета. Для их решения в некоторых алгоритмах каждая итерация разбивается на две и более подытерации. Благодаря своей относительной простоте согласно анализу [12] широкую известность получил двухподытерационный алгоритм ZS [13], основанный на шести логических условиях. Как установлено в работе [14], основными недостатками алгоритма ZS являются размытие диагональных линий толщиной 2 пиксела и удаление областей размером 2×2 пиксела. Для устранения этих недостатков и раз-

вития алгоритма ZS предложены следующие его модификации: изменение порогового значения в первом логическом условии [14], использование горизонтальных и вертикальных прямоугольных окон [15, 16] и расширенных окон из 11 пикселей [17–19], расширение алгоритма на полутоновые изображения [20], расширение условий удаления пикселей на второй подытерации для устранения чрезмерной эрозии [21]. Повышение качества скелетов в данных модификациях достигается за счет роста вычислительной сложности. Многие двухподытерационные алгоритмы, как и одноподытерационные, ориентированы на скелетизацию символов сканированных документов.

Целью настоящей работы является формирование связанных скелетов минимальной толщины для объектов произвольной формы на бинарных изображениях с низкой вычислительной сложностью.

Постановка задачи. Для бинарного изображения $I = \|i(y, x)\|_{(y=0, Y-1, x=0, X-1)}$ размером $Y \times X$, пиксели которого имеют значения 1 или 0 в зависимости от принадлежности площадному объекту или фону соответственно, алгоритмы скелетизации формируют матрицу $S = \|s(y, x)\|_{(y=0, Y-1, x=0, X-1)}$ скелетизации, значения элементов которой 1 или 0 указывают на фрагменты скелета или фона соответственно.

В алгоритме ОРТА на каждой итерации окрестность единичного элемента $s_{ОРТА}(y, x)$ матрицы скелетизации $S_{ОРТА}$ проходит две проверки (перед первой итерацией значения пикселей бинарного изображения I переносятся в матрицу $S_{ОРТА}$ скелетизации). Если окрестность элемента $s_{ОРТА}(y, x)$ соответствует маске на рис. 1, а, то $s_{ОРТА}(y, x) \leftarrow 0$ (шаг 1), а если маске на рис. 1, б, то $s_{ОРТА}(y, x) \leftarrow 1$ (шаг 2). Затем полученная матрица $S_{ОРТА}$ скелетизации проходит еще одну проверку. В результате этой проверки удаляются единичные элементы $s_{ОРТА}(y, x)$, окрестность которых совпадает с одной из масок на рис. 1, в (шаг 3). Обобщенная маска для алгоритма ОРТА показана на рис. 1, г, где $p(1) = s_{ОРТА}(y, x)$.

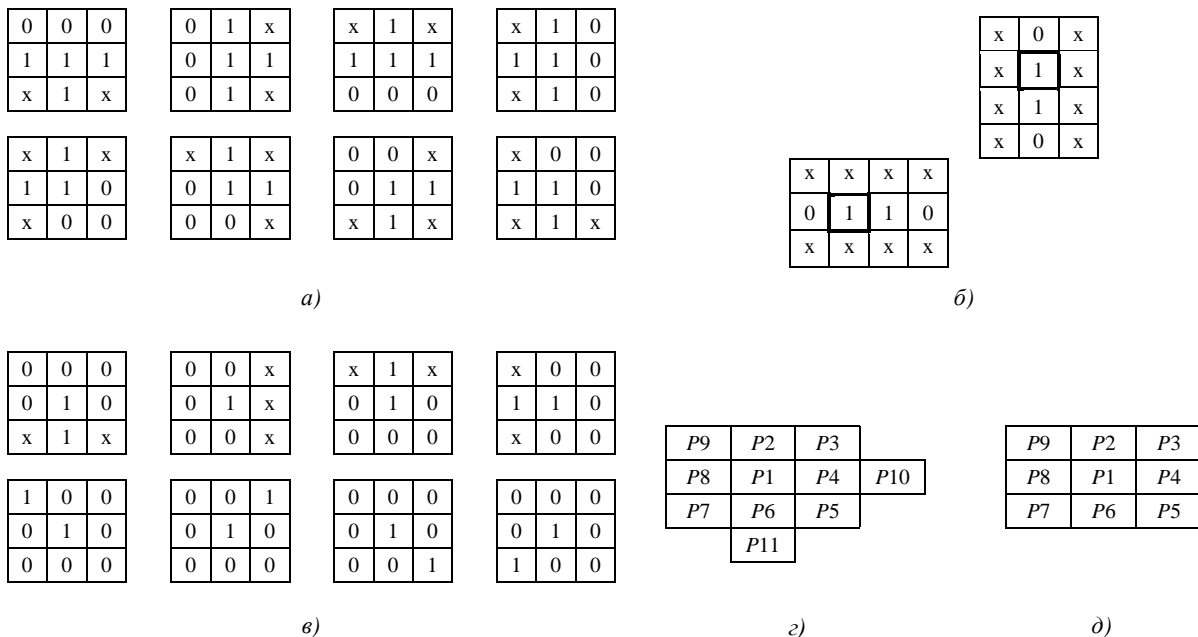


Рис. 1. Бинарные маски алгоритмов ОРТА и ZS: а) на шаге 1 ОРТА (x – безразличное состояние); б) на шаге 2 ОРТА; в) на шаге 3 ОРТА; г) общая по шагам 1–3 ОРТА; д) для алгоритма ZS

В алгоритме ZS каждая итерация состоит из двух подытераций и приводит к обнулению части единичных элементов $s_{zs}(y, x)$ матрицы S_{zs} скелетизации (перед первой итерацией значения пикселей бинарного изображения I переносятся в матрицу S_{zs} скелетизации), смежные элементы в окрестности которых (рис. 1, δ , $p(1) = s_{zs}(y, x)$) удовлетворяют определенным условиям.

На первой подытерации обнуляются элементы $s_{zs}(y, x)$, их смежные элементы удовлетворяют условиям

$$2 \leq \sum_{k=2}^9 p(k) \leq 6; \quad (1)$$

$$A(p(1)) = 1; \quad (2)$$

$$p(2) p(4) p(6) = 0; \quad (3)$$

$$p(4) p(6) p(8) = 0, \quad (4)$$

где $A(p(1))$ – число комбинаций 01, встречающихся среди элементов $p(2)–p(9)$.

На второй подытерации обнуляются элементы $s_{zs}(y, x)$, их смежные элементы удовлетворяют условиям (1), (2) и условиям

$$p(2) p(4) p(8) = 0; \quad (5)$$

$$p(2) p(6) p(8) = 0. \quad (6)$$

Итерации продолжаются до тех пор, пока элементы $s_{zs}(y, x)$ не обнулятся.

В качестве примера на рис. 2 показано бинарное изображение I размером 15×15 пикселей, содержащее несколько объектов (рис. 2, *a*), и бинарные изображения матриц скелетизации S , соответствующие бинарному изображению I и сформированные с помощью одноподытерационного ОРТА [11] (рис. 2, *б*) и двухподытерационного ZS [13] (рис. 2, *в*) алгоритмов скелетизации.

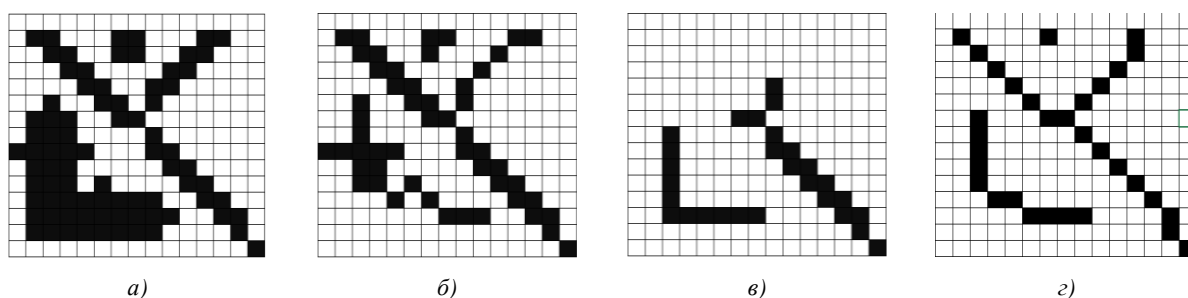


Рис. 2. Бинарное изображение I (*a*) и результаты его скелетизации, полученные с помощью алгоритмов ОРТА (*б*), ZS (*в*), OPCA (*г*)

На рис. 2 видны основные недостатки алгоритмов ОРТА и ZS. В обоих алгоритмах не достигается минимальная толщина линий скелета (многие узловые элементы имеют более двух соседей). Алгоритм ZS теряет некоторые диагональные линии и области 2×2 . Скелеты, формируемые алгоритмом ОРТА, лишены этих недостатков, но подвержены контурному шуму (незначительные искривления контурной линии отражаются на форме скелета).

Имея превосходство в качестве скелетов, алгоритм ОРТА обладает большей вычислительной сложностью в сравнении с алгоритмом ZS: время скелетизации изображения на рис. 2, *a* алгоритмом ОРТА приблизительно в 1,9 раза больше, чем время скелетизации алгоритмом ZS. Это подтверждается теоретической оценкой сложности алгоритмов. Из рис. 1, *a–в*

следует, что в алгоритме ОРТА на обработку каждого единичного элемента $s_{ОРТА}(y, x)$ матрицы $S_{ОРТА}$ на каждой итерации приходится до 50 операций обработки смежных элементов (44 операции на шаге 1 и 6 операций на шаге 2) и еще до 56 операций на каждый единичный элемент $s_{ОРТА}(y, x)$ необходимо для дополнительной проверки на шаге 3. Согласно выражениям (1)–(6) на каждой итерации элементы $p(2)$, $p(4)$, $p(6)$, $p(8)$ выбираются для обработки шесть раз, остальные элементы – три раза. Таким образом, на все единичные элементы $s_{ZS}(y, x)$ матрицы S_{ZS} на каждой итерации приходится 36 операций обработки смежных элементов. Следовательно, вычислительная сложность алгоритма ОРТА приблизительно в 1,4 раза выше вычислительной сложности алгоритма ZS (при большом числе итераций вычислительной сложностью шага 3 алгоритма ОРТА можно пренебречь).

Качество скелетов, сформированных алгоритмом ОРТА, делает его эффективной базой для создания нового алгоритма скелетизации, скорость которого может быть увеличена за счет использования для удаления избыточных элементов скелета вместо бинарных масок некоторых логических условий из алгоритма ZS.

Предлагаемые математическая модель и алгоритм скелетизации. Для построения предельно тонких связанных скелетов (неузловые элементы скелета имеют не более двух соседей) бинарных изображений с низкой вычислительной сложностью предлагается математическая модель ОРСА одноподытерационной скелетизации на основе комбинации и упрощения моделей ОРТА и ZS. Согласно предлагаемой модели на каждой итерации единичный элемент $s_{ОРСА}(y, x)$ матрицы $S_{ОРСА}$ скелетизации обнуляется (перед первой итерацией значения пикселей бинарного изображения I переносятся в матрицу $S_{ОРСА}$ скелетизации), если значения элементов выборки $P = \|p(k)\|_{(k=2,11)}$, формируемой, как показано на рис. 1, $\varepsilon(p(1) = s_{ОРСА}(y, x))$, удовлетворяют условию (1) и условиям

$$\sum_{k=2}^9 |p(k) - p(\text{mod}_8(k-2+1)+2)| = 2; \quad (7)$$

$$\neg((p(4)=1) \wedge (p(8)=0) \wedge (p(10)=0) \vee (p(6)=1) \wedge (p(2)=0) \wedge (p(11)=0)). \quad (8)$$

После выполнения всех итераций, когда не удаляется ни один элемент скелета на основании условий (1), (7) и (8), полученная матрица $S_{ОРСА}$ скелетизации проходит еще одну проверку, в результате которой удаляются единичные элементы $s_{ОРСА}(y, x)$, удовлетворяющие условию

$$(p(k)=0) \wedge (p(\text{mod}_8(k-2+3)+2)=1) \wedge (p(\text{mod}_8(k-2+5)+2)=1) \text{ при } k = \{3, 5, 7, 9\}. \quad (9)$$

Условие (7) эквивалентно условию (2) в модели ZS, условие (8) эквивалентно использованию масок 3×4 и 4×3 в модели ОРТА (см. рис. 1, б). Математическая модель ОРСА отличается от модели ОРТА исключением четырех масок 3×3 , предназначенных для удаления избыточных пикселей на горизонтальных и вертикальных прямых линиях скелета (верхний ряд на рис. 1, а); использованием упрощенного условия (9) вместо четырех масок 3×3 для удаления пикселей в точках изломов линий скелета (нижний ряд на рис. 1, а); исключением восьми масок 3×3 , предназначенных для удаления избыточных концевых пикселей скелета (см. рис. 1, в), и от модели ZS – исключением всех условий удаления пикселей, кроме условий (1) и (2).

Из условий (1) и (7)–(9) следует, что на каждой итерации элементы $p(2)$, $p(4)$, $p(6)$, $p(8)$ выбираются для обработки два раза, элементы $p(3)$, $p(5)$, $p(7)$, $p(9)$, $p(10)$, $p(11)$ – один раз. Таким образом, на каждый единичный элемент $s_{ОРСА}(y, x)$ матрицы $S_{ОРСА}$ на каждой итерации приходится 14 операций обработки смежных элементов. Следовательно, вычислительная сложность алгоритма ОРСА ниже вычислительной сложности алгоритмов ОРТА и ZS

приблизительно в 3,6 и 2,6 раза соответственно (без учета вычислительной сложности дополнительной обработки матриц скелетизации в алгоритмах ОПТА и ОРСА, которой можно пренебречь при большом числе итераций).

Исходя из рассмотренной модели, в работе предлагается алгоритм ОРСА скелетизации на основе комбинации и упрощения алгоритмов ОПТА и ZS. Сущность алгоритма ОРСА состоит в формировании матрицы $S_{ОРСА}$ скелетизации путем переноса в нее значений пикселей обрабатываемого бинарного изображения; итеративной обработке матрицы $S_{ОРСА}$ скелетизации для удаления ее избыточных единичных элементов с использованием условий (1), (7), (8); прекращении итеративной обработки при отсутствии избыточных единичных элементов матрицы $S_{ОРСА}$ скелетизации, для которых выполняются условия (1), (7), (8), и удалении избыточных единичных элементов матрицы $S_{ОРСА}$ скелетизации с помощью условия (9).

Алгоритм ОРСА состоит из следующих шагов:

Вход: бинарное изображение I .

Инициализация матрицы $S_{ОРСА}$ скелетизации: $S_{ОРСА} \leftarrow I$.

Цикл 1 (итеративная обработка матрицы $S_{ОРСА}$).

Инициализация счетчика удаленных элементов: $N_{DE} \leftarrow 0$.

Цикл 2 по (y, x) (удаление избыточных элементов матрицы $S_{ОРСА}$).

Если для $s_{ОРСА}(y, x) = 1$ выполняются условия (1), (7), (8), то $s_{ОРСА}(y, x) \leftarrow 0$;

$N_{DE} \leftarrow N_{DE} + 1$.

Конец цикла 2.

Если $N_{DE} = 0$, то выход из цикла 1.

Конец цикла 1.

Цикл 3 по (y, x) (удаление избыточных элементов матрицы $S_{ОРСА}$).

Если для $s_{ОРСА}(y, x) = 1$ выполняется условие (9), то $s_{ОРСА}(y, x) \leftarrow 0$.

Конец цикла 3.

Выход: матрица $S_{ОРСА}$ скелетизации.

Таким образом, в результате выполнения алгоритма ОРСА формируется матрица $S_{ОРСА}$ скелетизации, единичные элементы которой указывают на линии скелетов, соответствующих бинарным объектам на исходном изображении I .

Оценка эффективности алгоритма скелетизации. Проведено сравнение предложенного алгоритма ОРСА с известными алгоритмами ОПТА [11] и ZS [13] (алгоритмы реализованы на языке программирования C++ и протестированы на компьютере с ОС Windows 8 64-бит, CPU i7 2,6 GHz, RAM 8 GB). При этом использованы предложенные в [22, 23] оценки: толщина скелета ST ($ST \in [0,1]$; чем ближе к единице значение ST , тем скелет тоньше) [22]; скорость скелетизации TS (число удаляемых элементов в секунду; чем больше значение TS , тем скорость утоньшения выше) [23]; средняя ось MA , определяющая возможность восстановления исходного изображения по скелету (чем ближе к единице значение MA , тем восстанавливаемость выше [22]). Данные оценки определяются с помощью следующих выражений:

$$ST = 1 - \frac{\sum_{y=0}^{Y-1} \sum_{x=0}^{X-1} \sum_{k=1}^4 s(y, x) Q(k)}{\sum_{y=0}^{Y-1} \sum_{x=0}^{X-1} s(y, x)}, \quad (10)$$

где $Q(k)$ – бинарная маска (рис. 3), позволяющая выявить избыточные элементы скелета;

$$TS = \frac{\sum_{y=0}^{Y-1} \sum_{x=0}^{X-1} i(y, x) - \sum_{y=0}^{Y-1} \sum_{x=0}^{X-1} s(y, x)}{ET}, \quad (11)$$

где ET – время скелетизации, с;

$$MA = \frac{\sum_{y=0}^{Y-1} \sum_{x=0}^{X-1} i'(y, x)}{\sum_{y=0}^{Y-1} \sum_{x=0}^{X-1} i(y, x)}, \quad (12)$$

где $i'(y, x)$ – пиксели восстановленного (например, с помощью морфологического наращивания) по скелету S бинарного изображения $I' = \left\| i'(y, x) \right\|_{(y=0, Y-1, x=0, X-1)}$.

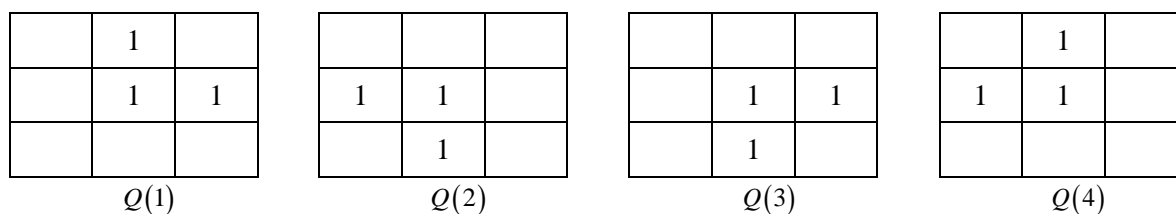


Рис. 3. Бинарные маски

Значения оценок ST , TS , MA для шести тестовых изображений $I(1)$ – $I(6)$ (рис. 4) и алгоритмов ОРТА, ZS и ОРСА приведены в табл. 1. Результаты скелетизации $S(1)$ – $S(6)$ тестовых изображений с помощью алгоритмов ОРТА, ZS и ОРСА показаны на рис. 5–7.

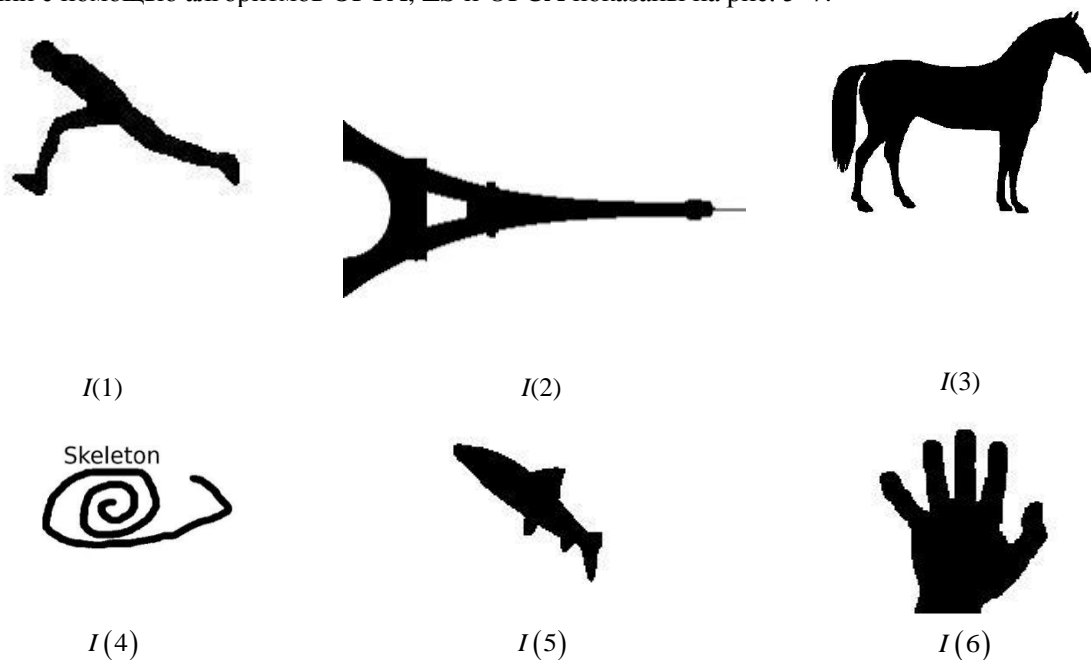


Рис. 4. Тестовые бинарные изображения

В табл. 1 приведены также значения избыточности связей между пикселями скелета SC ($SC \geq 0$). Чем ближе к нулю значение SC , тем меньше избыточных связей и пикселей в скелете.

Неконцевые и узловые пиксели предельно тонкого скелета должны иметь связи только с двумя смежными пикселями. Поэтому для определения избыточности SC предлагается использовать выражение

$$SC = \frac{\sum_{y=1}^{Y-2} \sum_{x=1}^{X-2} \left(s(y, x) \left(\sum_{j=-1}^1 \sum_{i=-1}^1 s(y+j, x+i) - 1 \right) \right) - \left(2 \sum_{y=1}^{Y-2} \sum_{x=1}^{X-2} s(y, x) - N_{EP} \right)}{\sum_{y=1}^{Y-2} \sum_{x=1}^{X-2} s(y, x)}, \quad (13)$$

где N_{EP} – число концевых элементов скелета.

Таблица 1

Оценки результатов скелетизации тестовых изображений $I(1)$ – $I(6)$

Изображение	Алгоритм	ST	TS	MA	SC
$I(1)$	OPTA	0,8631	645	0,8521	0,1237
	ZS	0,7833	854	0,9208	0,1849
	OPCA	0,9877	1318	0,9263	0,0183
$I(2)$	OPTA	0,8965	195	0,9115	0,0952
	ZS	0,8784	462	0,9602	0,1233
	OPCA	0,9993	575	0,9606	0,0032
$I(3)$	OPTA	0,9796	204	0,8941	0,0303
	ZS	0,8350	516	0,9502	0,1500
	OPCA	0,9991	705	0,9544	0,0058
$I(4)$	OPTA	0,9365	793	0,8986	0,0650
	ZS	0,8031	1352	0,9482	0,1700
	OPCA	0,9995	2549	0,9487	0,0028
$I(5)$	OPTA	0,7042	428	0,8540	0,2350
	ZS	0,6566	766	0,9279	0,2756
	OPCA	0,9931	1092	0,9436	0,0172
$I(6)$	OPTA	0,9618	351	0,7204	0,0419
	ZS	0,7275	633	0,9370	0,2220
	OPCA	0,9941	916	0,9493	0,0118

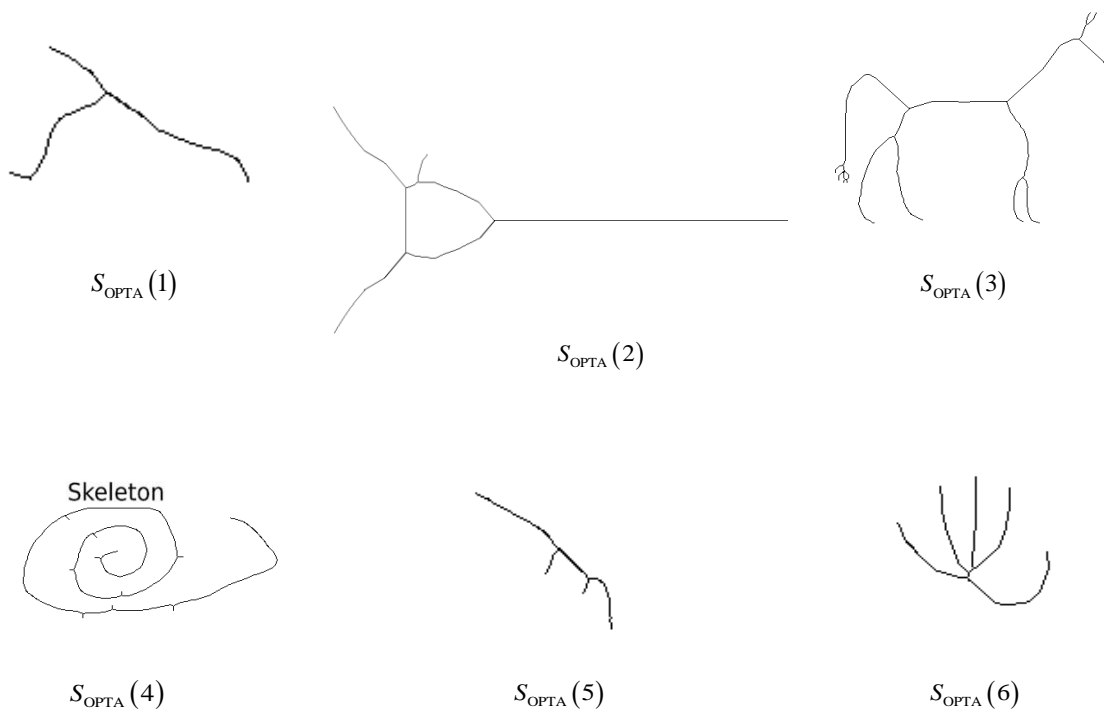


Рис. 5. Результаты скелетизации тестовых изображений с помощью алгоритма OPTA

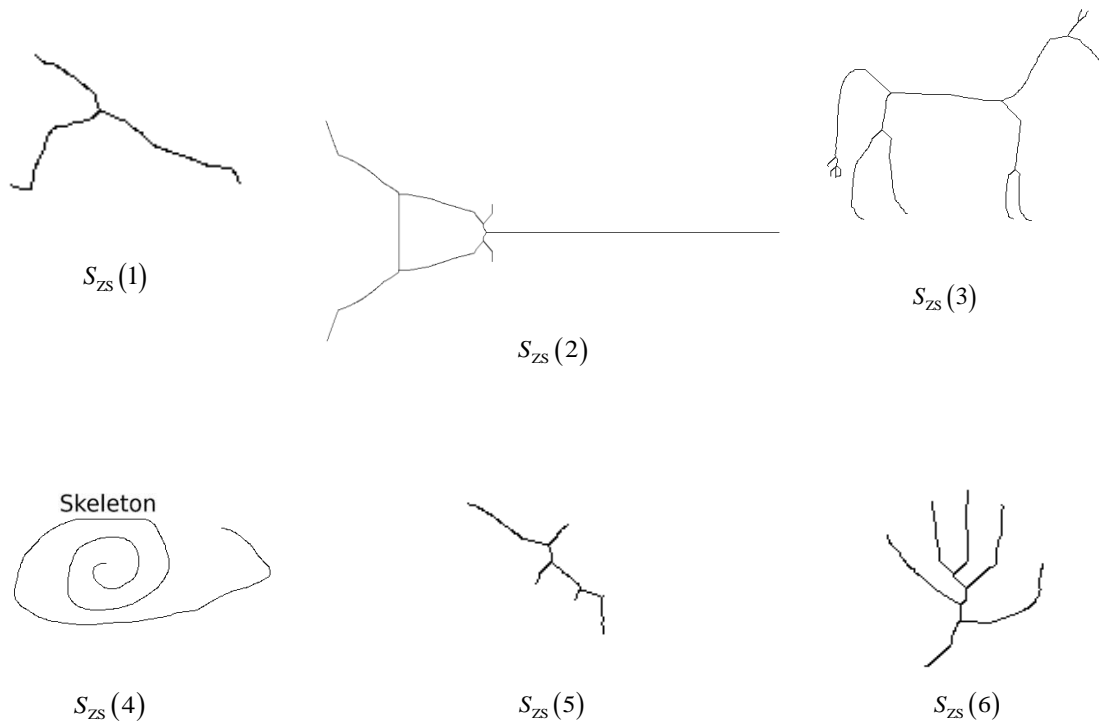


Рис. 6. Результаты скелетизации тестовых изображений с помощью алгоритма ZS

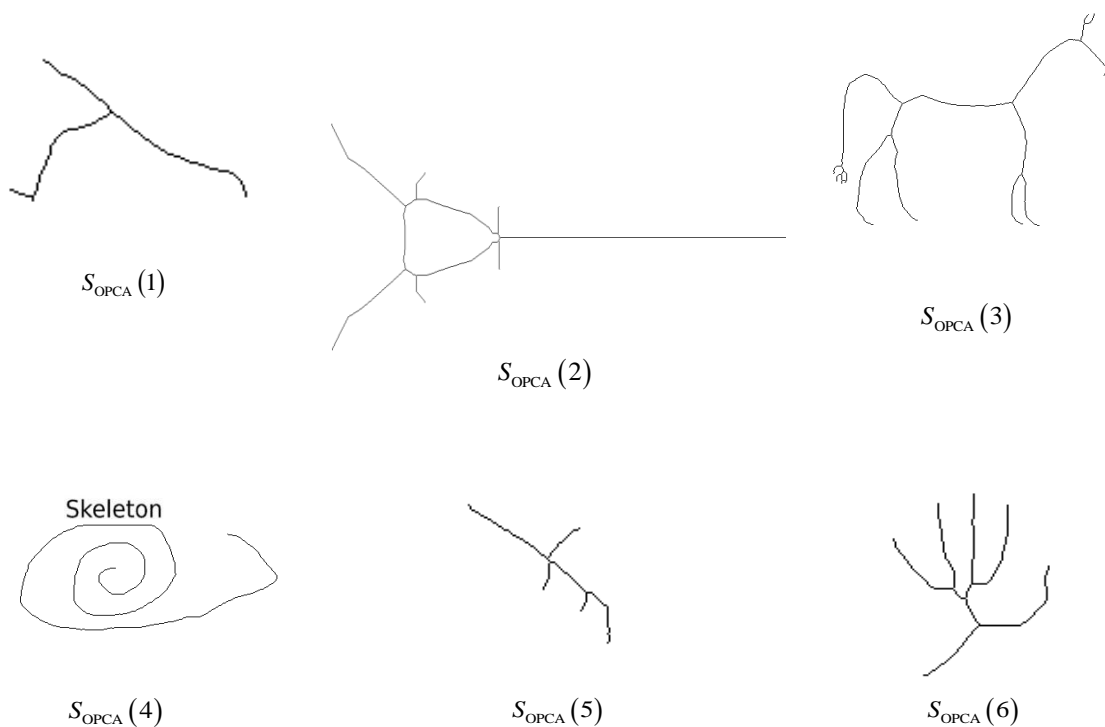


Рис. 7. Результаты скелетизации тестовых изображений с помощью алгоритма OPKA

Все оценки, приведенные в табл. 1, свидетельствуют о более высокой эффективности алгоритма OPKA в сравнении с алгоритмами OPTA и ZS.

В табл. 2 приведены значения оценок результатов скелетизации (10)–(13), усредненные по 100 бинарным изображениям, которые содержатся в работах [13, 14, 23] и тестовой базе Kimia's 99 [24].

Таблица 2

Средние значения оценок результатов скелетизации для 100 изображений, полученные с помощью алгоритмов OPTA, ZS и OPCA

Алгоритм	<i>ST</i>	<i>TS</i>	<i>MA</i>	<i>SC</i>
OPTA	0,9146	305	0,8588	0,0985
ZS	0,7799	640	0,9308	0,1876
OPCA	0,9956	860	0,9460	0,0097

Из табл. 2 следует, что алгоритм OPCA превосходит алгоритмы OPTA и ZS соответственно на 8,1 и 21,6 % по оценке *ST*, в 2,8 и 1,3 раза по оценке *TS*, на 8,7 и 1,5 % по оценке *MA*, в 10,2 и 19,3 раза по оценке *SC*.

Предложенная оценка *SC* избыточности связей между пикселями скелета, так же как и оценка *ST*, характеризует толщину линий скелета, но обеспечивает большую точность. Из табл. 1 следует, что для изображений *I*(3) и *I*(6) алгоритмы OPCA и OPTA практически не отличаются по оценке *ST* (отличия по этим изображениям составляют 2 и 3 % соответственно), однако существенно отличаются по оценке *SC* (в 5,2 и 3,6 раза соответственно).

Заключение. В работе предложены математическая модель и алгоритм OPCA одноподытерационной скелетизации на основе комбинации и упрощения моделей OPTA и ZS. Алгоритм OPCA отличается от OPTA исключением масок, предназначенных для удаления избыточных элементов на горизонтальных и вертикальных прямых линиях скелета; использованием упрощенного условия для удаления пикселей в точках изломов линий скелета; исключением масок, предназначенных для удаления избыточных концевых элементов скелета, а от ZS – исключением всех условий удаления пикселей, кроме двух основных. По сравнению с OPTA алгоритм OPCA обеспечивает уменьшение толщины скелета на 8,1 %, повышение скорости скелетизации в 2,8 раза, повышение восстанавливаемости исходного изображения по скелету на 8,7 %, уменьшение избыточности связей между пикселями скелета в 10,2 раза. По сравнению с ZS алгоритм OPCA обеспечивает уменьшение толщины скелета на 21,6 %, повышение скорости скелетизации в 1,3 раза, повышение восстанавливаемости исходного изображения по скелету на 1,5 %, уменьшение избыточности связей между пикселями скелета в 19,3 раза.

References

1. Dinneen G. P. Programming pattern recognition. *Western Joint Computer Conference, New York, 1–3 March 1955*. New York, 1955, vol. 1, pp. 94–100.
2. Kirsch R. A., Cahn L., Ray C., Urban G. H. Experiments in processing pictorial information with a digital computer. *Eastern Computer Conference, Washington, 9–13 December 1957*. Washington, 1957, vol. 12, pp. 221–229.
3. Lam L., Lee S.-W., Suen C. Y. Thinning methodologies – a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992, vol. 14, no. 9, pp. 869–885.
4. Holt C., Stewart A., Clint C., Perrott R. An improved parallel thinning algorithm. *Communication ACM*, 1987, vol. 30, pp. 156–160.
5. Manzanera A., Bernard T., Preteux F., Longuet B. Ultra-fast skeleton based on an isotropic fully parallel algorithm. *Discrete Geometry for Computer Imagery*, 1999, vol. 1568, pp. 313–324.
6. Bernard T. M., Manzanera A. Improved low complexity fully parallel thinning algorithm. *Proceedings 10th International Conference on Image Analysis and Processing, Venice, Italy, 27–29 September 1999*. Venice, 1999, pp. 215–220.
7. Chen C.-S., Tsai W.-H. A new fast one-pass thinning algorithm and its parallel hardware implementation. *Pattern Recognition Letters*, 1990, vol. 11, pp. 471–477.
8. Wu R.-Y., Tsai W.-H. A new one-pass parallel algorithm for binary images. *Pattern Recognition Letters*, 1992, vol. 13, pp. 715–723.

9. Deng W., Lyengar S. S., Brener N. E. A fast parallel thinning algorithm for the binary image skeletonization. *International Journal of High Performance Computing Applications*, 2000, vol. 14, pp. 65–81.
10. Stefanelli R., Rosenfeld A. Some parallel thinning algorithms for digital pictures. *Journal of the ACM*, 1971, vol. 18, pp. 255–264.
11. Chin R. T., Wan H. K., Stover D. L., Iverson R. D. A one-pass thinning algorithm and its parallel implementation. *Computer Vision, Graphics, and Image Processing*, 1987, vol. 40, pp. 30–40.
12. Harous, S., Elnagar A. Handwritten character-based parallel thinning algorithms: a comparative study. *University of Sharjah Journal of Pure & Applied Sciences*, 2009, vol. 6, no. 1, pp. 81–101.
13. Zhang T. Y., Suen C. Y. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 1984, vol. 27, no. 3, pp. 236–239.
14. Lu H. E., Wang P. S. P. A comment on "a fast parallel algorithm for thinning digital patterns". *Communications of the ACM*, 1986, vol. 29, no. 3, pp. 239–242.
15. Abdulla W. H., Saleh A. O. M., Morad A. H. A preprocessing algorithm for handwritten character recognition. *Pattern Recognition Letters*, 1988, vol. 7, no. 1, pp. 13–18.
16. Sossa J. H. An improved parallel algorithm for thinning digital patterns. *Pattern Recognition Letters*, 1989, vol. 10, pp. 77–80.
17. Guo Z., Hall R. W. Parallel thinning with two-subiteration algorithms. *Communications of the ACM*, 1989, vol. 32, no. 3, pp. 359–373.
18. Guo Z., Hall R. W. Fast fully parallel thinning algorithms. *CVGIP: Image Understanding*, 1992, vol. 55, no. 3, pp. 317–328.
19. Zhang Y. Y., Wang P. P. A parallel thinning algorithm with two-subiteration that generates one-pixel-wide skeletons. *International Conference on Pattern Recognition, Vienna, Austria, 25–29 August 1996*. Vienna, 1996, vol. 4, pp. 457–461.
20. Kundu M. K., Chaudhuri B. B., Majumder D. D. A parallel graytone thinning algorithm (PGTA). *Pattern Recognition Letters*, 1991, vol. 12, no. 8, pp. 491–494.
21. Kwon J.-S., Gi J.-W., Kang E.-K. An enhanced thinning algorithm using parallel processing. *Proceeding of the International Conference on Image Processing, Thessaloniki, Greece, 7–10 October 2001*. Thessaloniki, 2001, pp. 752–755.
22. Jang B. K., Chin R. T. One-pass parallel: analysis, properties, and quantitative evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992, vol. 14, pp. 1129–1140.
23. Boudaoud L. B., Sider A., Tari A. A new thinning algorithm for binary images. *3rd International Conference on Control, Engineering & Information Technology, Tlemcen, Algeria, 25–27 May 2015*. Tlemcen, 2015, pp. 1–6.
24. Sebastian T. B., Klein P. N., Kimia B. B. Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004, vol. 26, no. 5, pp. 550–571.

Информация об авторах

Ма Цзюнь, аспирант кафедры инфокоммуникационных технологий, Белорусский государственный университет информатики и радиоэлектроники, Минск, Беларусь.

E-mail: majun1313@hotmail.com

Цветков Виктор Юрьевич, доктор технических наук, доцент, заведующий кафедрой инфокоммуникационных технологий, Белорусский государственный университет информатики и радиоэлектроники, Минск, Беларусь.

E-mail: vtsvet@bsuir.by

Конопелько Валерий Константинович, доктор технических наук, профессор, профессор кафедры инфокоммуникационных технологий, Белорусский государственный университет информатики и радиоэлектроники, Минск, Беларусь.

E-mail: kafikt@bsuir.by

Information about the authors

Jun Ma, Postgraduate Student of Department of Infocommunication Technologies, Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus.

E-mail: majun1313@hotmail.com

Viktar Yu. Tsviatkou, Dr. Sci (Eng.), Associate Professor, Head of Department of Infocommunication Technologies, Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus.

E-mail: vtsvet@bsuir.by

Valery K. Kanapelka, Dr. Sci (Eng.), Professor, Professor of Department of Infocommunication Technologies, Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus.

E-mail: kafikt@bsuir.by

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

УДК 004.934
<https://doi.org/10.37661/1816-0301-2020-17-2-36-43>

Поступила в редакцию 06.12.2019
Received 06.12.2019

Принята к публикации 08.04.2020
Accepted 08.04.2020

Выделение речевой активности на фоне шумов при помощи компактной сверточной нейронной сети

Г. С. Вашкевич[✉], И. С. Азаров

*Белорусский государственный университет
информатики и радиоэлектроники, Минск, Беларусь
✉E-mail: ryhorv@gmail.com*

Аннотация. Исследуется задача выделения речевой активности из зашумленного звукового сигнала. Предлагается компактная модель сверточной нейронной сети, которая имеет всего 385 параметров. Модель нетребовательна к вычислительным ресурсам, что позволяет использовать ее в рамках концепции Интернета вещей для портативных устройств с низким энергопотреблением. В то же время эта модель обеспечивает высокую точность определения речевой активности на уровне лучших современных аналогов. Указанные полезные свойства достигаются путем применения специального сверточного слоя, учитывающего гармоническую структуру вокализованной речи и устраняющего избыточность модели за счет инвариантности к изменениям частоты основного тона. В рамках экспериментов производительность модели оценивалась в различных шумовых условиях для разных соотношений сигнала и шума. Результаты экспериментов показали, что предложенная модель обеспечивает более высокую точность определения речевой активности по сравнению с моделью, представленной компанией Google в фреймворке WebRTC.

Ключевые слова: детектор речевой активности, гармонический сигнал, сверточная нейронная сеть, частота основного тона, обработка речи

Для цитирования. Вашкевич, Г. С. Выделение речевой активности на фоне шумов при помощи компактной сверточной нейронной сети / Г. С. Вашкевич, И. С. Азаров // Информатика. – 2020. – Т. 17, № 2. – С. 36–43. <https://doi.org/10.37661/1816-0301-2020-17-2-36-43>

Voice activity detection in noisy conditions using tiny convolutional neural network

Ryhor S. Vashkevich[✉], Elias S. Azarov

*Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus
✉E-mail: ryhorv@gmail.com*

Abstract. The paper investigates the problem of voice activity detection from a noisy sound signal. An extremely compact convolutional neural network is proposed. The model has only 385 trainable parameters. Proposed model doesn't require a lot of computational resources that allows to use it as part of the "internet of things" concept for compact low power devices. At the same time the model provides state of the art results in voice activity detection in terms of detection accuracy. The properties of the model are achieved by using a special convolutional layer that considers the harmonic structure of vocal speech. This layer also eliminates redundancy of the model because it has invariance to changes of fundamental frequency. The model performance is evaluated in various noise conditions with different signal-to-noise ratios. The results show that the proposed model provides higher accuracy compared to voice activity detection model from the WebRTC framework by Google.

Keywords: voice activity detector, harmonic signal, convolutional neural network, pitch, speech processing

For citation. Vashkevich R. S., Azarov E. S. Voice activity detection in noisy conditions using tiny convolutional neural network. *Informatics*, 2020, vol. 17, no. 2, pp. 36–43 (in Russian). <https://doi.org/10.37661/1816-0301-2020-17-2-36-43>

Введение. Выделение речевой активности из звукового сигнала является актуальной задачей, которая часто возникает при построении различных систем обработки речи. Разработка детектора речи в реальных условиях эксплуатации дополнительно усложняется из-за наличия в речи посторонних звуков (шума). При высоком уровне зашумления выделить речь становится и вовсе невозможно.

В большинстве научных публикаций предлагаются решения для выделения речевой активности на основе методов машинного обучения, которые обеспечивают высокую точность выделения, но вместе с тем являются вычислительно затратными. С целью сокращения вычислений рассматриваются альтернативные подходы, в том числе и аналитическая модель определения речевой активности на основе анализа формант гармонического сигнала [1]. К недостаткам данного метода можно отнести допущение о том, что речевой сигнал всегда имеет гармоническую структуру. В работе [2] предлагается отказаться от применения технологий машинного обучения путем анализа спектрограммы звукового сигнала, разделенной на две части. Допускается, что информация о речевом сигнале всегда содержится в нижней полосе частот, а в верхней располагается шум. Недостатком такого подхода является чувствительность к низкочастотным шумам.

Решения на основе методов машинного обучения в целом более устойчивы к шумам, поскольку способны учитывать различную природу шумов и большую вариацию человеческого голоса. Один из самых простых примеров применения машинного обучения представлен в работе [3], где используется метод опорных векторов для определения двух классов характеристических признаков речевого сигнала: голоса и шума. Во многих решениях применяются искусственные нейронные сети, которые являются более мощным инструментом для классификации характеристических признаков [4–9]. Высокая вычислительная сложность методов на основе искусственных нейронных сетей связана с большим числом параметров. В частности, в работе [4] при помощи сверточных нейронных сетей предлагается моделировать частотные вариации входного сигнала, а при помощи слоев долгой краткосрочной памяти (long short-term memory, LSTM) сети учитывать его временные вариации. Модель имеет примерно 100 000 параметров. В работах [5, 9] представлены нейронные сети, состоящие только из полносвязных слоев с очень большим числом параметров (модель с 1 млн параметров в [9]), которые требуют большой обучающей выборки и склонны к переобучению. В решениях [5, 8] предлагается сначала удалить шумы из исходного сигнала и оставить только голос. Таким образом, первые несколько слоев моделей выполняют очистку сигнала от шума, а последующие слои используются в качестве классификатора очищенного сигнала.

В публикациях [4, 6, 7] принимается во внимание тот факт, что речевой сигнал является протяженным во времени, и выделяются протяженные во времени признаки речи при помощи рекуррентных нейронных сетей. При этом модель, предложенная в работе [6], имеет достаточно малое количество настраиваемых параметров (около 350), что выгодно выделяет ее на фоне других. Однако рекуррентные нейронные сети сложны в обучении и требуют большой обучающей выборки для надежной работы, поскольку имеют очень большое число возможных состояний.

Во многих задачах обработки речи, в том числе и в задачах детектирования речевой активности, в качестве характеристических признаков используются мел-кепстральные коэффициенты (mel-frequency cepstral coefficients, MFCC). Эти признаки в работах [3, 5, 9] успешно применяются для детектирования речевой активности. В статьях [1, 2] выделяют признаки, основанные на спектрограмме сигнала, принимая во внимание тот факт, что речевой сигнал в большинстве случаев имеет гармоническую структуру.

В последнее время начали набирать популярность end-to-end подходы к построению моделей глубокого обучения. Их суть заключается в том, что на вход глубокой нейросетевой модели подаются данные, которые не прошли предварительную обработку и из которых не извлечены

характеристические признаки. Эти данные сами по себе являются признаками. Например, в работе [4] авторы на вход нейронной сети подают такую аудиозапись, что помогает избежать дополнительных вычислительных затрат на предобработку входных данных. Однако подобные модели тяжело обучать, поскольку для этого требуется огромное количество данных.

В настоящей работе предлагается ультракомпактная модель сверточной нейронной сети для определения речевой активности, которая имеет всего 385 параметров. Предложенная модель нетребовательна к вычислительным ресурсам, что позволяет использовать ее в рамках концепции Интернета вещей для портативных устройств с низким энергопотреблением. Точность определения речевой активности предложенной модели находится на уровне лучших современных аналогов и по результатам экспериментов превосходит наиболее популярный в настоящее время детектор речи, представленный компанией Google в фреймворке WebRTC. Полезные свойства предложенной модели обеспечиваются использованием специального сверточного слоя, учитывающего гармоническую структуру вокализованной речи и устраняющего избыточность за счет инвариантности к изменениям частоты основного тона. В рамках выполненных экспериментов производительность модели оценивалась в различных шумовых условиях с разными соотношениями сигнала и шума. Реализация предложенной модели для выделения речевой активности доступна по адресу <https://github.com/gvashkevich/vad>.

Сверточные нейронные сети для обработки речи. В задачах обработки речи, таких как распознавание, синтез и детектирование, применяют сверточные нейронные сети, состоящие из последовательности сверточных слоев (convolution layers) и слоев объединения (pooling layers). Общая структура одной из таких сетей показана на рис. 1. Входными признаками являются кратковременные амплитудные спектры речевого сигнала.

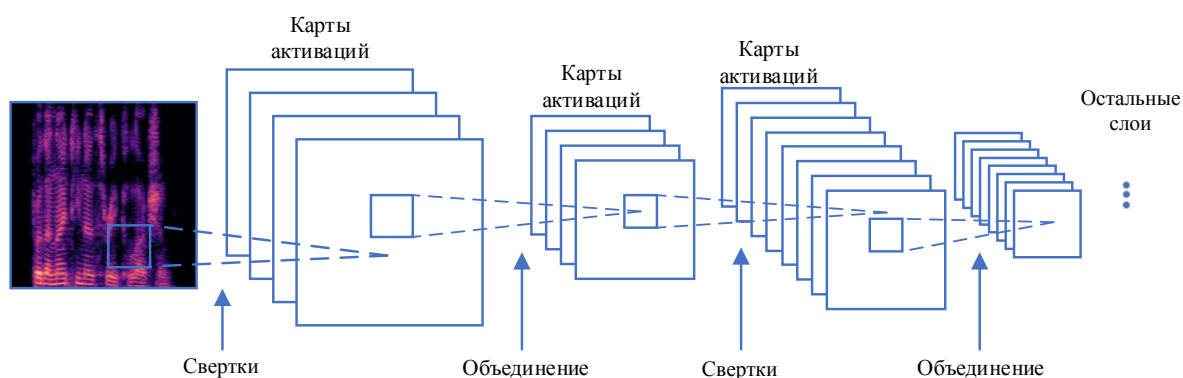


Рис. 1. Сверточная нейронная сеть

Свертка выполняется с фиксированным окном и небольшим шагом. Такой способ обработки звуковых сигналов заимствован из задач компьютерного зрения и обработки изображений, где соседние пиксели равноценны по своему значению. Однако носителем речи служит спектр со специфической структурой. Большая часть речи является вокализованной (от 80 до 95 % общей продолжительности речевого сигнала) и образуется при помощи голосовых связок, создающих периодические колебания. По этой причине спектр речи имеет преимущественно гармоническую структуру и почти вся значимая информация, необходимая для решения задачи, сконцентрирована на частотах, пропорциональных частоте основного тона (F_0) (рис. 2).

Таким образом, весь амплитудный спектр в качестве характеристического вектора является избыточным, причем избыточность можно устранить, выбирая информационно важные компоненты спектра, кратно соответствующие частоте основного тона. Между тем необходимо учитывать, что частота основного тона изменяется во времени в достаточно широких пределах. Считается, что для речевых приложений ее диапазон варьируется от 50 до 450 Гц.

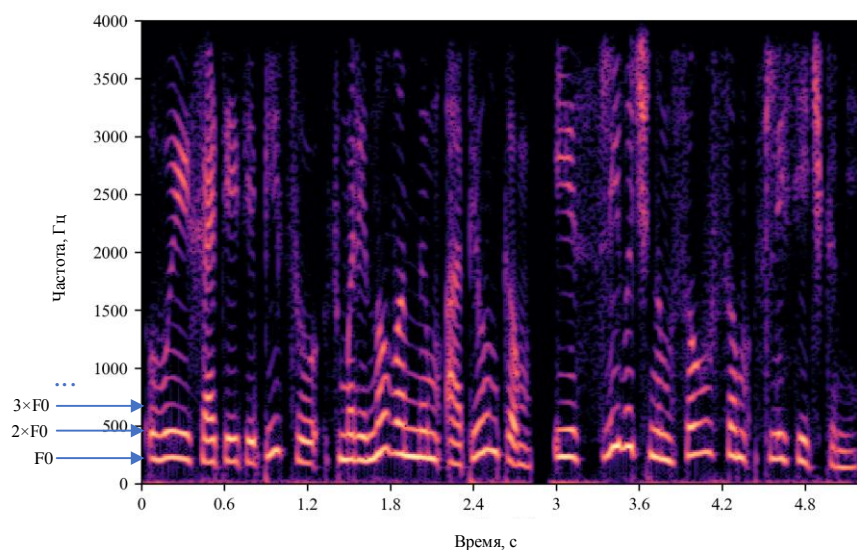


Рис. 2. Амплитудный спектр речевого сигнала

Для устранения избыточности нейронной сети предлагается применять свертки амплитудного спектра, инвариантные к изменению основного тона. Зададим N гипотез о том, что F_0 принимает значения в допустимом диапазоне от $F_{0_{\min}}$ до $F_{0_{\max}}$. Для каждой гипотезы выберем M информационно важных составляющих спектра, соответствующих гармоникам речевого сигнала, частоты которых пропорциональны частоте основного тона. Из выбранных компонент спектра сформируем матрицу признаков размером $N \times M$, где вдоль первой оси будут располагаться гипотезы, а вдоль второй – гармоники, соответствующие гипотезам. Полученную матрицу признаков подадим на вход классической сверточной нейронной сети для классификации речи и шума (рис. 3).

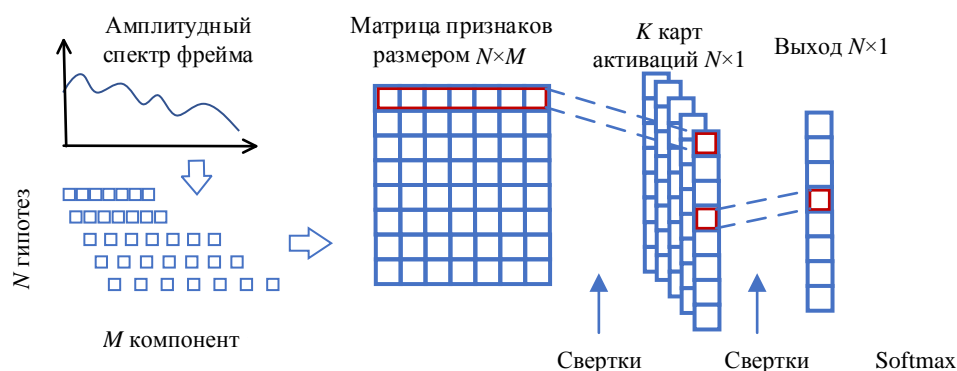


Рис. 3. Предлагаемая модель сверточной нейронной сети для детектирования речевого сигнала

Полученная модель состоит из одного сверточного и одного полносвязного слоя. В свою очередь, сверточный слой состоит из K фильтров с ядром $1 \times M$ и функций активации ReLU. Ядро такого размера позволяет осуществлять операцию свертки только вдоль оси с гармониками исходного сигнала, обрабатывая каждую гипотезу независимо друг от друга. Результат работы сверточного слоя агрегируется при помощи полносвязного слоя с одним нейроном. Для получения распределения вероятностей того, какая из N гипотез содержит в себе гармоническую составляющую, выход второго слоя поступает на функцию активации softmax.

Формирование характеристических признаков. Как было описано в предыдущем разделе, формирование характеристических признаков происходит путем выбора частотных компонент спектрограммы сигнала, кратных заданной частоте основного тона F_0 . Гипотетические

значения $F0_i$ получаются путем равномерного разбиения частотного интервала $F0_{\min} \dots F0_{\max}$ на N значений:

$$F0_i = F0_{\min} + i \cdot \frac{F0_{\max} - F0_{\min}}{N}, \quad i = 0, \dots, N - 1.$$

Далее для каждой гипотезы $F0_i$ из исходной спектрограммы выбираются по M гармоник. Индексы I_i^j всех M гармоник в спектрограмме с частотным разрешением f_r для заданной $F0_i$ вычисляются при помощи выражения

$$I_i^j = \text{round}\left(\frac{1+j}{2} \cdot \frac{F0_i}{f_r} + 1\right), \quad j = 0, \dots, M - 1,$$

где $\text{round}(\ast)$ – операция округления.

Частотное разрешение спектрограммы определяется по формуле

$$f_r = \frac{f_s}{N_{fft}},$$

где f_s – частота дискретизации исходного сигнала, N_{fft} – размер быстрого преобразования Фурье.

Для получения результирующего характеристического вектора признаков X для одного фрейма звукового сигнала s необходимо вычислить дискретное преобразование Фурье, а затем из амплитудного спектра выбрать только те компоненты, которые соответствуют вычисленным индексам I_i^j :

$$S = \log_{10}|FFT(s)|,$$

$$X(i,j) = S(I_i^j), \quad i = 0, \dots, M - 1, \quad j = 0, \dots, M - 1.$$

Таким образом, для каждого входного фрейма формируется матрица признаков X размером $N \times M$, состоящая из N гипотез по M компонент. Общая идея выделения признаков показана на рис. 4, где черным цветом отмечены элементы матрицы признаков X , а красным – элементы матрицы признаков X с частотой $F0$ для каждого кандидата.

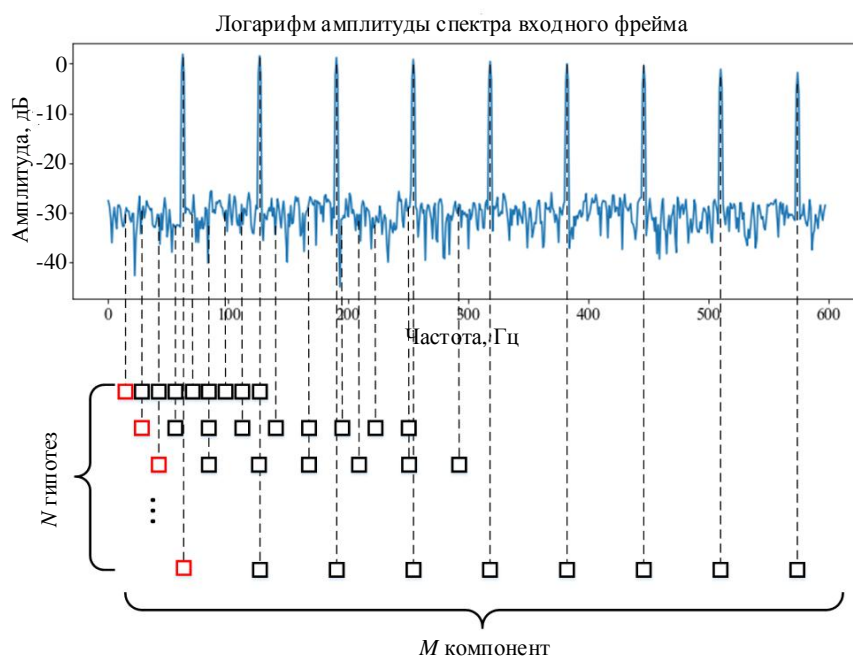


Рис. 4. Процесс формирования матрицы признаков

Задача нейронной сети состоит в определении существования среди предложенных гипотез такой, которая содержит амплитуды гармоник исходного сигнала. Если модель сможет обнаружить эту гипотезу, то текущий входной пример соответствует речевому сигналу, в противном случае данный пример классифицируется как шум.

Экспериментальные исследования. Для тестирования и обучения модели применялся открытый набор данных Musan [10]. Набор состоит из записей трех типов: речи, шума и музыки, однако в экспериментах использовались только записи речи и шума. Обучение производилось на речевых данных с различными шумами и соотношениями сигнала и шума (signal-to-noise ratio, SNR). Для каждой речевой записи случайно выбирались три шумовые записи. Их содержимое по отдельности прибавлялось к речевой записи с заданным коэффициентом k , определяющим значение SNR зашумленного сигнала:

$$\text{SNR} = 20 \cdot \log_{10} \left(\frac{A}{k \cdot A_n} \right),$$

где A и A_n – средние мощности речевого и шумового сигналов соответственно,

$$A = \frac{1}{\text{len}(s)} \sum s^2.$$

Значение коэффициента определялось по формуле

$$k = 10^{\frac{\log_{10}(\frac{A}{A_n}) - \frac{\text{SNR}}{20}}{2}}.$$

Следует учитывать, что значения мощностей A и A_n – квадратичные величины, поэтому зашумленный аддитивным шумом сигнал вычислялся согласно выражению

$$s_{add} = s + \sqrt{k} \cdot s_n.$$

В исследовании значение SNR выбиралось случайным образом в диапазоне от 10 до 20 дБ. Зашумленный входной сигнал разбивался на фреймы длиной 50 мс с шагом 12,5 мс, после чего из полученных фреймов формировались характеристические признаки с диапазоном от 75 до 350 Гц, которые подавались на вход классификатора. Следовательно, каждый фрейм классифицировался независимо от остальных.

Если хотя бы один из выходов сети принимал значение активации больше порогового, то считалось, что входной фрейм относится к речи, в противном случае – к шуму. Использованное пороговое значение составляло 0,15.

В качестве целевых значений для тренировочных данных применялось значение частоты основного тона речевых записей, полученное с помощью алгоритма YAPT [11]. Полученные значения квантовались таким образом, чтобы количество уровней квантования соответствовало числу гипотез. При этом невокализованные участки относились к нулевому уровню.

В ходе экспериментов были эмпирически подобраны оптимальные значения для количественных параметров гармоник $M = 22$, гипотез $N = 100$ и фильтров сверточного слоя $K = 16$.

Обучение сети осуществлялось методом Adam [12] с шагом обучения 0,001. Обучение продолжалось 50 000 итераций с размером минимальной партии в 256 элементов.

Анализ работы модели. Визуализация активаций выходного слоя сети позволяет оценить, какая именно гипотеза во входном векторе признаков представляет собой гармоническую структуру входного сигнала. На рис. 5, б представлены активации выхода нейронной сети для участка аудиозаписи, спектрограмма которого показана на рис. 5, а. Видно, что активации сети имеют высокое значение на тех участках изображения, которые соответствуют вокализованным участкам речи на спектрограмме. При этом изменение номера гипотезы с максимальной активацией повторяет изменение частоты основного тона в исходном сигнале, что обеспечивает возможность применения предложенной модели в качестве оценщика тона.

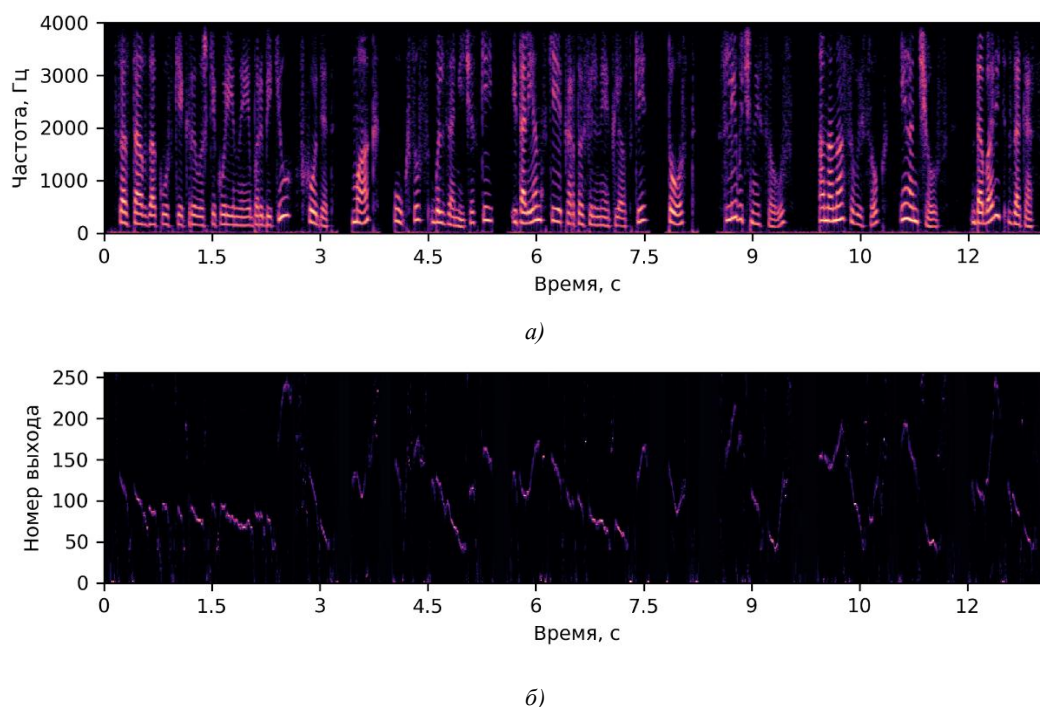


Рис. 5. Визуализация работы сверточной сети

Предложенная модель имеет всего 385 обучаемых коэффициентов, что позволяет ей обучаться на небольшой выборке данных и выделять речевую активность небольшим объемом вычислений. При этом анализ других нейросетевых моделей, описанных в первом разделе, показывает, что они имеют значительно больше обучаемых параметров и требуют значительно больших вычислительных ресурсов как для обучения, так и для применения.

Чтобы получить численные значения, характеризующие качество работы предложенной модели относительно других современных моделей, используется метрика AUC. Предложенная модель сравнивается с моделью, разработанной компанией Google для фреймворка WebRTC (URL: <https://github.com/wiseman/py-webrtcvad>). В настоящее время детектор речевой активности из WebRTC является одной из наиболее популярных открытых моделей для решения этой задачи. Точность предложенной модели (0,8821) превосходит точность модели WebRTC (0,8755) на тестовом наборе данных.

При решении ряда задач, связанных с выделением речевой активности, бывает недостаточно только пофреймовой обработки входного сигнала. Поэтому дальнейшая работа будет посвящена выделению речевой активности на более высоком уровне, где будут учитываться слова или фразы речи. Поскольку предложенная модель способна эффективно выделять речевые признаки, связанные с интонацией речи, данную задачу, предположительно, можно решить, анализируя последовательности обработанных на основе предложенной модели фреймов путем добавления надстройки из дополнительных слоев нейронной сети.

Заключение. Основой предложенной модели выделения речевой активности из зашумленного звукового сигнала на базе компактной сверточной нейронной сети служит специальный сверточный слой, который учитывает гармоническую структуру вокализованных участков речевого сигнала. Добавление такого слоя позволяет значительно сократить количество настраиваемых параметров нейросетевой модели, что обуславливает ее низкую требовательность к объему обучающих данных и вычислительным ресурсам. Данный факт делает предложенную модель идеальной для реализации во встраиваемых системах и мобильных устройствах с низким энергопотреблением.

Качество работы модели подтверждено сравнением с популярным современным решением для выделения речевой активности из фреймворка WebRTC от компании Google: предложенная модель по точности детектирования речи близка к модели WebRTC.

References

1. Yoo I.-C., Lim H., Yook D. Formant-based robust voice activity detection. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 2015, vol. 23, no. 12, pp. 2238–2245. <https://doi.org/10.1109/TASLP.2015.2476762>
2. Pang J. Spectrum energy based voice activity detection. *The 7th IEEE Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, 9–11 January 2017*. Las Vegas, 2017, pp. 1–5. <https://doi.org/10.1109/CCWC.2017.7868454>
3. Kinnunen T., Chernenko E., Tuononen M., Fränti P., Li H. Voice activity detection using MFCC features and support vector machine. *The 12th International Conference on Speech and Computer (SPECOM07), Moscow, Russia, 15–18 October 2007*. Moscow, 2007, vol. 2, pp. 556–561.
4. Zazo R., Sainath T. N., Simko G., Parada C. Feature learning with raw-waveform CLDNNs for voice activity detection. *17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, 8–12 September 2016*. San Francisco, 2016, pp. 3668–3672. <https://doi.org/10.21437/Interspeech.2016-268>
5. Zhang X., Wu J. Denoising deep neural networks based voice activity detection. *International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013*. Vancouver, 2013, pp. 853–857. <https://doi.org/10.1109/ICASSP.2013.6637769>
6. Hughes T., Mierle K. Recurrent neural networks for voice activity detection. *International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013*. Vancouver, 2013, pp. 7378–7382. <https://doi.org/10.1109/ICASSP.2013.6639096>
7. Eyben F., Weninger F., Squartini S., Schuller B. Real-life voice activity detection with LSTM Recurrent Neural Networks and an application to Hollywood movies. *International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013*. Vancouver, 2013, pp. 483–487. <https://doi.org/10.1109/ICASSP.2013.6637694>
8. Wang Q., Du J., Bao X., Wang Z.-R., Dai L.-R., Lee C.-H. A universal VAD based on jointly trained deep neural networks. *16th Annual Conference of the International Speech Communication Association, Dresden, Germany, 6–10 September 2015*. Dresden, 2015, pp. 2282–2286.
9. Ryant N., Liberman M., Yuan J. Speech activity detection on youtube using deep neural networks. *14th Annual Conference of the International Speech Communication Association, Lyon, France, 25–29 August 2013*. Lyon, 2013, pp. 728–731.
10. Snyder D., Chen G., Povey D. *Musan: a Music, Speech, and Noise Corpus*, 2015. Available at: <https://arxiv.org/abs/1510.08484> (accessed 20.10.2019).
11. Kasi K., Zahorian S. A. Yet another algorithm for pitch tracking. *International Conference on Acoustics, Speech, and Signal Processing, Orlando, 13–17 May 2002*. Orlando, 2002, vol. 1, pp. 361–364. <https://doi.org/10.1109/ICASSP.2002.5743729>
12. Kingma D. P., Ba J. *Adam: a Method for Stochastic Optimization*, 2014. Available at: <https://arxiv.org/abs/1412.6980> (accessed 20.10.2019).

Информация об авторах

Вашкевич Григорий Сергеевич, магистр технических наук, аспирант кафедры ЭВС, Белорусский государственный университет информатики и радиоэлектроники, Минск, Беларусь.
E-mail: ryhorv@gmail.com

Азаров Илья Сергеевич, доктор технических наук, доцент, заведующий кафедрой ЭВС, Белорусский государственный университет информатики и радиоэлектроники, Минск, Беларусь.

Information about the authors

Ryhor S. Vashkevich, M. Sci. (Eng.), Postgraduate Student of the Department of EMU, Belarusian State University of Informatics and Radioelectronics Minsk, Belarus.
E-mail: ryhorv@gmail.com

Elias S. Azarov, Dr. Sci. (Eng.), Associate Professor, Head of the Department of EMU, Belarusian State University of Informatics and Radioelectronics Minsk, Belarus.

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

БИОИНФОРМАТИКА
BIOINFORMATICS

УДК 51-76:577.322:539.19:004.94:004.021
<https://doi.org/10.37661/1816-0301-2020-17-2-44-53>

Поступила в редакцию 24.04.2020
Received 24.04.2020

Принята к публикации 07.05.2020
Accepted 07.05.2020

Предсказание структуры гомодимерных белковых комплексов на основе глубокой нейронной сети

А. Ю. Хадарович^{1,2✉}, А. А. Калиновский¹, А. В. Тузиков¹

¹Объединенный институт проблем информатики
Национальной академии наук Беларуси, Минск, Беларусь
✉E-mail: ahadarovich@gmail.com

²Белорусский государственный университет, Минск, Беларусь

Аннотация. Предсказание структур белковых комплексов имеет важные приложения в таких областях, как моделирование биологических процессов и разработка лекарственных средств. Гомодимеры (комплексы, состоящие из двух одинаковых белков) являются наиболее распространенным типом белковых комплексов в природе, но до сих пор нет универсального алгоритма для предсказания их трехмерных структур. Экспериментальные методы для определения структур белковых комплексов требуют значительных затрат времени и ресурсов и имеют свои ограничения. Глубокие нейронные сети позволили предсказать структуры отдельных белков, значительно превзойдя по точности другие алгоритмические подходы. Опираясь на идею этого подхода, в статье разработан алгоритм для моделирования трехмерной структуры гомодимерных комплексов на основе глубокого обучения. Он состоит из двух основных этапов: на первом этапе прогнозируется карта контактов белкового комплекса при помощи глубокой сверточной нейронной сети, а на втором предсказывается трехмерная структура гомодимера на основе полученной карты контактов и процедуры оптимизации. Предложенный подход был протестирован и проверен на наборе белковых гомодимеров из базы данных белков PDB (Protein Data Bank). Разработанная методика может быть использована для оценки моделей белковых гомодимеров в качестве одного из этапов разработки лекарственных соединений.

Ключевые слова: методы глубокого обучения, нейронные сети, моделирование белковых комплексов, гомодимеры, оптимизация

Для цитирования. Хадарович, А. Ю. Предсказание структуры гомодимерных белковых комплексов на основе глубокой нейронной сети / А. Ю. Хадарович, А. А. Калиновский, А. В. Тузиков // Информатика. – 2020. – Т. 17, № 2. – С. 44–53. <https://doi.org/10.37661/1816-0301-2020-17-2-44-53>

Protein homodimers structure prediction based on deep neural network

Anna Y. Hadarovich^{1,2✉}, Alexander A. Kalinouski¹, Alexander V. Tuzikov¹

¹The United Institute of Informatics Problems of the National Academy
of Sciences of Belarus, Minsk, Belarus
✉E-mail: ahadarovich@gmail.com

²Belarusian State University, Minsk, Belarus

Abstract. Structural prediction of protein-protein complexes has important application in such domains as modeling of biological processes and drug design. Homodimers (complexes which consist of two identical proteins) are the most common type of protein complexes in nature but there is still no universal algorithm to

predict their 3D structures. Experimental techniques to identify the structure of protein complex require enormous amount of time and resources, and each method has its own limitations. Recently Deep Neural Networks allowed to predict structures of individual proteins greatly prevailing in accuracy over other algorithmic approaches. Building on the idea of this approach, we developed an algorithm to model the 3D structure of homodimer based on deep learning. It consists of two major steps: at the first step a protein complex contact map is predicted with the deep convolutional neural network, and the second stage is used to predict 3D structure of homodimer based on obtained contact map and optimization procedure. The use of the neural network in combination with optimization procedure based on gradient descent method allowed to predict structures for protein homodimers. The suggested approach was tested and validated on a dataset of protein homodimers from Protein Data Bank (PDB). The developed procedure could be also used for evaluating protein homodimer models as one of the stages in drug compounds developing.

Keywords: deep learning, neural network, modeling of protein complexes, homodimers, optimization

For citation. Nadarovich A. Y., Kalinouski A. A., Tuzikov A. V. Protein homodimers structure prediction based on deep neural network. *Informatics*, 2020, vol. 17, no. 2, pp. 44–53 (in Russian). <https://doi.org/10.37661/1816-0301-2020-17-2-44-53>

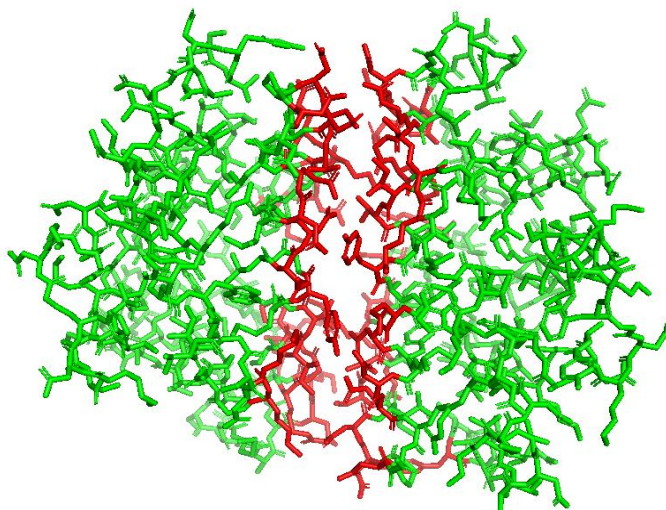
Введение. Большинство процессов в клетке базируется на специфическом взаимодействии белков, для которого необходима определенная пространственная структура. Поэтому биологическая функция белков тесно связана с существованием их в виде трехмерных структур. Даже небольшие изменения этих структур часто ведут к утере или резкому изменению активности белков. Знание пространственной организации белковых молекул не только является ключом к пониманию их функций и механизма работы, но и служит основой для разработки эффективных и безопасных лекарственных средств. Общеизвестно утверждение, что сама аминокислотная последовательность белка (при подходящей температуре и pH воды) определяет ее пространственную структуру [1].

В последние годы набирает популярность подход к машинному обучению, называемый глубоким обучением [2]. В частности, в 2018 г. был предложен алгоритм на основе глубоко обучения для свободного (ab initio) моделирования структуры белка [3, 4], превосходящий по точности предсказания другие ab initio алгоритмы [5]. Подходы, основанные на машинном обучении, и раньше использовались в задаче предсказания структуры белка, однако позволяли с достаточной точностью предсказать не структуру белка, а только ее локальные особенности: карты контактов [6–8], элементы вторичной структуры [9], интерфейсные остатки и т. д. [4, 6, 10]. Успехи методов глубоко обучения обусловлены тем, что количество накопленных данных позволило с достаточной точностью тренировать разработанные модели.

Вступая во взаимодействие, белки образуют белок-белковый комплекс. Задача нахождения трехмерной структуры комплекса, образованного при взаимодействии белков, называется белковым докинг. Так же как и в случае индивидуальных белков, экспериментальные методы могут использоваться в ограниченном количестве случаев и требуют длительного времени, а в массовом порядке и вовсе неприменимы. Поэтому ускоренное развитие получили вычислительные методы, которые позволяют предсказывать трехмерные структуры комплексов быстро, используя в качестве входных данных трехмерные структуры белков, составляющих комплекс. Наибольшее значение при этом имеет область связывания белков, или интерфейс белкового комплекса, поскольку именно в данной области находятся аминокислоты, являющиеся необходимыми для образования комплекса и выполнения соответствующих функций в организме.

Пара множеств аминокислотных остатков $\{(A_1, \dots, A_N), (B_1, \dots, B_M)\}$ называется интерфейсом (или областью связывания), если выполняется следующее условие: для любого остатка A_i из белка A найдется по крайней мере один остаток B_j из белка B , такой, что расстояние между данными остатками не превосходит некоторого порога (величина порога, как правило, варьируется от 6 до 12 Å). На рис. 1 изображены белок-белковый комплекс 1jzk из базы данных PDB [11], состоящий из двух одинаковых белков с выделенным интерфейсом, и аминокислотная последовательность белка, образующего комплекс. Красным цветом выделен интерфейс комплекса, а зеленым цветом – аминокислотные остатки, не входящие в интерфейс. На нижней

панели отображена аминокислотная последовательность белка, образующего комплекс, в которой красным цветом обозначены аминокислоты, принадлежащие интерфейсу, а черным цветом – все остальные аминокислоты.



SVYDAAAQLTADVKKDLRDSWKVIGSDKKGNGVALMTTLFADNQETIGYFKRLGNVSQGAND-
KLRGHSITLMYALQNFIDQLDNPDDLVCVVEKFAVNHITRKISAAEFGKFNPIKVKVLASKNFGDKYANAWAKLVAVVQAAL

Рис. 1. Белок-белковый комплекс 1jzk

В настоящей статье поставлена задача разработки алгоритма предсказания трехмерной структуры комплексов белковых гомодимеров, используя в качестве входных данных трехмерные структуры белков, составляющих комплексы. Данная задача является актуальной, поскольку гомодимеры – это наиболее часто встречающийся тип белковых комплексов [12]. Для ее решения требуется найти взаимное расположение двух белков в пространстве, оптимизирующее выбранную целевую функцию. В качестве критерия оптимизации обычно используют минимизацию энергии связывания белков в комплексе. Это вычислительно сложная оптимизационная задача в шестимерном пространстве параметров, задающих расположение белков в пространстве друг относительно друга. В работе предлагается алгоритм для ее решения, основанный не на выборе конкретной целевой функции, а на машинном обучении. Предложенный алгоритм состоит из следующих этапов:

- формирование набора данных белковых гомодимерных комплексов из базы данных белков, отвечающих заданным требованиям: высокое разрешение экспериментально полученной структуры, отсутствие неточностей в файле описания структуры;
- фильтрация набора данных по площади интерфейса для исключения белковых гомодимеров, которые были ошибочно аннотированы как комплексы;
- построение архитектуры глубокой сверточной нейронной сети для предсказания карты контактов белковых гомодимерных комплексов;
- обучение нейронной сети;
- оценка результатов обучения нейронной сети;
- построение алгоритма предсказания трехмерной структуры гомодимерного комплекса на основе карты интерфейсных контактов и трехмерной структуры входящих в комплекс белков;
- оценка полученных результатов.

Глубокая нейронная сеть для предсказания карты контактов белковых гомодимерных комплексов. Для предсказания контактов (близкого расположения) между атомами двух белковых цепей в составе комплекса была разработана глубокая нейронная полносверточная сеть

с разреженной сверткой и остаточными блоками [13–16]. Архитектура построенной нейронной сети показана на рис. 2.

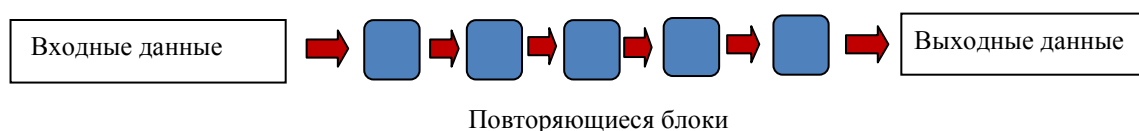


Рис. 2. Схема глубокой нейронной сети для предсказания карты контактов гомодимерного комплекса

Схема повторяющихся блоков, обозначенных на рис. 2 синим цветом, изображена на рис. 3.

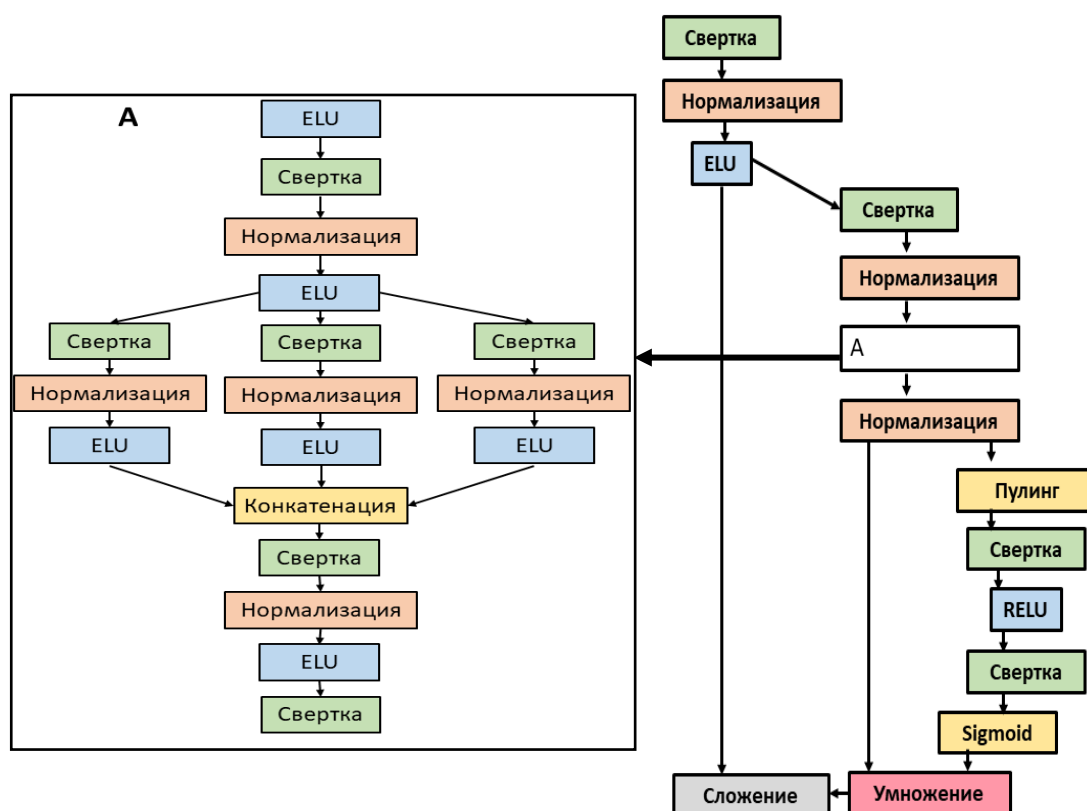


Рис. 3. Схема блока глубокой нейронной сети для предсказания карты контактов гомодимерного комплекса

Блоки ELU, RELU и Sigmoid обозначают выполнение соответствующих функций активации нейронов, блоки умножения и сложения подразумевают поэлементные операции.

На слоях нейронной сети используется активационная функция RELU [1]

$$f(x) = \begin{cases} 0 & \text{при } x < 0; \\ x & \text{при } x \geq 0 \end{cases}$$

и сигмоидная активационная функция, обозначенная на схеме как Sigmoid [2],

$$f(x) = \frac{1}{1 + e^{-x}}.$$

На промежуточных слоях нейронной сети используется активационная функция ELU [3]

$$f(x) = \begin{cases} (e^x - 1) & \text{при } x < 0; \\ x & \text{при } x \geq 0. \end{cases}$$

Преимуществом использования функции ELU является то, что она может принимать отрицательные значения, которые помогают сети изменять веса (значения параметров), выполнять смещения в правильных направлениях, а также производить активации весов нейронной сети при расчете градиента и избегать так называемой dead RELU проблемы, когда большинство компонентов сети не обновляется.

В глубокой нейронной сети выделяют две основные операции: свертку и пулинг. Свертка – это процесс применения фильтра к каждому участку слоя. Данная операция может быть задана формулой [4]

$$x_{i,j}^l = f \left(\sum_m \sum_n x_{i+m,j+n}^{l-1} \cdot w_{m,n}^l + b^l \right),$$

где x^l – выход слоя l в виде матрицы (x – входные данные в случае $l = 0$); $f()$ – функция активации; b^l – коэффициент сдвига слоя l ; $w_{m,n}^l$ – элемент матрицы весовых коэффициентов фильтра для получения слоя l .

Пулинг – это процесс сжатия (уменьшения размеров) входных данных путем некоторой обработки значений блоков данных, который позволяет снизить размерность данных, ускоряя время работы сети. Пулинг «сжимает» карты признаков, полученные на предыдущем сверточном слое, с помощью выбранного преобразования (обычно используется функция выбора максимума значений признаков).

Следует отметить, что некоторым слоям подавались на вход нормализованные данные (на рис. 2 это блоки «Нормализация»), т. е. предварительно обработанные и имеющие нулевое математическое ожидание и единичную дисперсию.

В качестве функции потерь использовалась бинарная кросс-энтропия, вычисляемая по формуле

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)),$$

где y_i – наличие контакта между атомами ($y_i = 1$, если контакт есть, $y_i = 0$ в противном случае); $p(y_i)$ – предсказанная вероятность наличия контакта; N – количество атомов [5].

В качестве входных данных использовались атомы СА (альфа-углерода) и СВ (бета-углерода) аминокислотных остатков. Атом СА в аминокислоте означает первый атом углерода, который крепится к функциональной группе, а второй атом углерода называется атомом СВ. На рис. 4 изображен белок гомодимера с выделенными атомами СА и СВ.

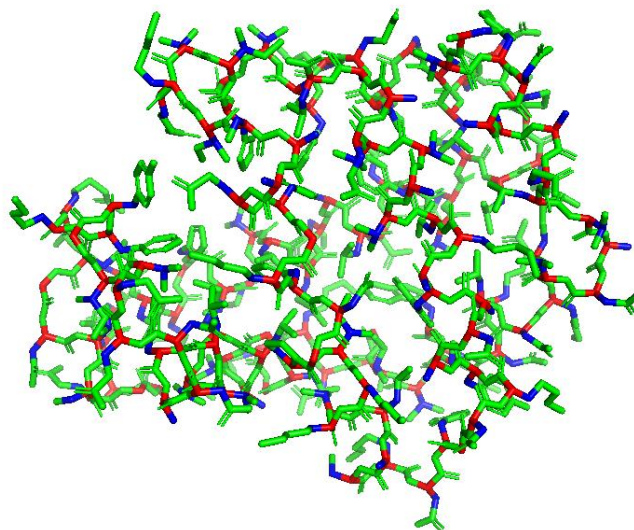


Рис. 4. Белок с выделенными атомами СА (обозначены красным цветом) и СВ (обозначены синим цветом)

Сеть принимает на вход матрицу расстояний между атомами СВ одного белка (матрица расстояний для второго белка будет такой же, поскольку белки одинаковые) и аминокислотную последовательность первой белковой цепи, а в качестве выходных данных выдает бинарную матрицу размером $N \times N$ контактов между атомами СВ двух белков в комплексе, который они составляют (N – количество аминокислотных остатков в одной белковой цепи). Атомы СВ двух белков находятся в контакте, если евклидово расстояние между ними не превосходит 8 \AA .

Для обучения нейронной сети использовались различные параметры, но наибольшую точность показали следующие:

количество эпох (итераций в процессе обучения, включающих предъявление всех примеров из обучающего множества и проверку качества обучения на контрольном множестве) – 500;

количество итераций на одну эпоху – 200;

скорость обучения – 0,0001;

размер входа сети – 128;

оптимизатор – метод Adam [17].

Следует отметить, что авторы статьи предпринимали попытку разработать архитектуру нейронной сети, способную напрямую предсказывать матрицу расстояний между контактами белкового комплекса, однако сделали вывод, что на текущий момент для предсказания расстояний с достаточной точностью не хватает исходных данных.

Оптимизационный алгоритм для предсказания трехмерной структуры белкового гомодимерного комплекса. На вход алгоритма подаются трехмерные структуры двух белков, составляющих гомодимерный комплекс, и предсказанная на предыдущем шаге карта контактов, представляющая собой бинарную матрицу $A = [a(i, j)]$ размером $N \times N$. Элементы матрицы вычисляются по формуле [6]

$$a(i, j) = \begin{cases} 1, & \text{если } d(i, j) < 8 \text{ \AA}, \\ 0 & \text{в противном случае.} \end{cases}$$

Выходом алгоритма является матрица аффинного преобразования T размером 4×4 [7]:

$$T = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

где элементы r_{ij} задают параметры матрицы поворота, а t_{ij} – параметры сдвига, которые применяются к структуре первого белка для получения координат расположения в пространстве второй белковой цепи относительно первой.

Как и в случае стандартного подхода свободного моделирования димерного комплекса, одна белковая структура считается фиксированной в пространстве и ищется расположение другой белковой структуры относительно нее. Это расположение задается шестью параметрами, отвечающими за шесть степеней свободы: три параметра задают поворот и три параметра – сдвиг. При начальной инициализации предполагается, что начальное положение в пространстве второй белковой структуры совпадает с расположением первой, т. е. все шесть параметров равны нулю.

Алгоритм является итерационным, и на каждой итерации выполняются следующие шаги:

- вычисление матрицы расстояний между атомами СВ первой и второй белковой цепи;
- вычисление функции потерь;
- вычисление производной функции потерь;
- преобразование параметров, отвечающих за поворот и сдвиг; минимизация функции потерь в направлении, противоположном градиенту функции;
- вычисление матрицы преобразований для новых координат атомов;
- вычисление новых координат структуры второго белка по полученной матрице.

На последнем этапе шесть параметров преобразуются в матрицу аффинного преобразования, которое затем применяется к первой белковой цепи и задает положение в пространстве второго белка гомодимера. Таким образом, результатом работы алгоритма являются координаты всего гомодимерного белкового комплекса.

Создание и фильтрация набора данных. Набор данных белковых гомодимерных комплексов был получен из базы данных белков PDB [11]. Файлы описания комплексов предоставляются в специальном формате, предусматривающем структурную аннотацию каждой белковой цепи: секция АТОМ файлов с расширением PDB описывает координаты в трехмерном евклидовом пространстве и тип атома для каждой аминокислоты в белковой цепи.

К полученным данным была применена процедура фильтрации: файлы с ошибочной информацией были исключены, отфильтрованы файлы с числом цепочек, отличным от двух, а также удалены комплексы с различным количеством аминокислотных остатков в одном файле (такая ситуация может возникнуть, если во время эксперимента не удалось определить все остатки в белковой цепи). Площадь скрытой поверхности была рассчитана с помощью библиотеки FreeSASA [18] для того, чтобы гарантировать, что выбранные комплексы имеют интерфейсы разумного размера (более 250 \AA^2) [19]. Комплексы с меньшей площадью скрытой поверхности рассматриваются как артефакты эксперимента, а файлы, содержащие такие комплексы, не являются достоверным источником информации о белковом комплексе.

Окончательный набор данных содержал около 10 000 структур. Он был разделен на обучающую и тестовую части в пропорции 4 : 1 соответственно. Процедуры фильтрации реализовывались на языке программирования Python с помощью свободно распространяемой библиотеки для обработки биологических данных BioPython [20].

Результаты работы алгоритма. На рис. 5 изображены графики функции потерь для обучающего и тестового наборов данных. Они демонстрируют скорость обучения модели, а также дают ее качественную оценку: модель обучается с увеличением количества эпох, в то же время не наблюдается переобучения, поскольку значения функции потерь для тестового набора данных незначительно отличаются от значений для обучающего набора.

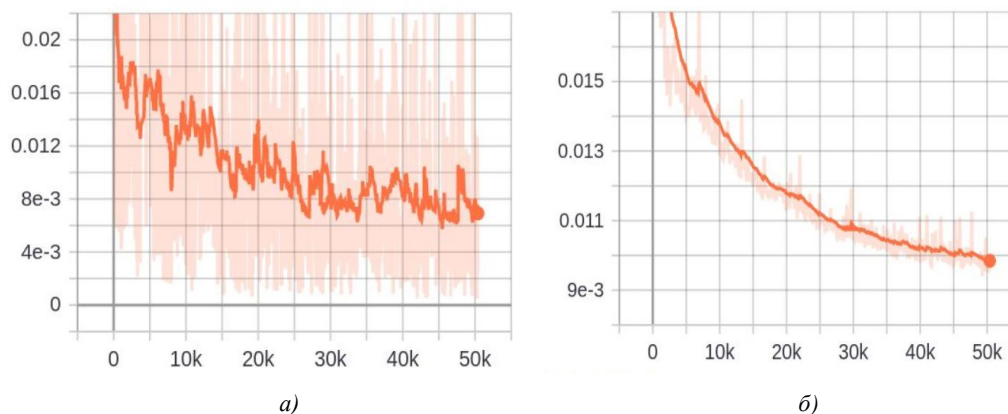


Рис. 5. Значения функции потерь (вертикальная ось) в зависимости от количества итераций (горизонтальная ось) на обучающем (а) и тестовом (б) наборах данных

Пример входных и выходных данных для первого этапа алгоритма представлен на рис. 6. На левой панели показана карта расстояний (евклидово расстояние между атомами СВ) одной белковой цепи в гомодимере, которая использовалась как инвариантное к сдвигу и повороту представление белка для нейронной сети. На средней и правой панелях показаны карты реальных и предсказанных контактов соответственно, где синие пиксели означают отсутствие контакта, а желтые – его наличие. На нижней панели указана цветовая шкала для визуализации расстояний.

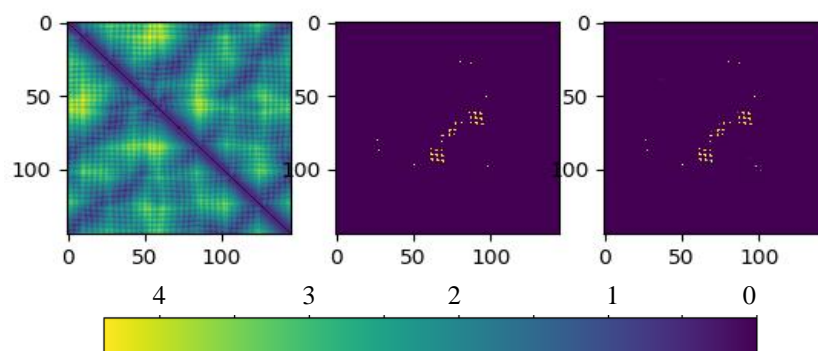


Рис. 6. Пример успешного предсказания карты контактов для гомодимера ljkz

Для более половины протестированных комплексов удалось предсказать большую часть интерфейсных остатков с точностью более 50 %, из них для четверти комплексов – с точностью более 70 %. Предсказанные интерфейсные остатки значительно сокращают процедуру поиска трехмерной структуры гомодимера, поскольку область связывания известна. Это позволяет наложить ограничения на область поиска и существенно сократить количество возможных ориентаций структур белков в комплексе относительно друг друга. Для многих алгоритмов докинга точность предсказания структуры возрастает в несколько раз, если на вход подается информация о предполагаемой интерфейсной области.

Почти для 30 % гомодимеров представленный алгоритм обеспечил предсказания структуры всего гомодимерного комплекса хорошего качества, т. е. среднеквадратическое отклонение между атомами СВ реальной и предсказанной структур составляет менее 10 Å. Пример результата работы оптимизационного алгоритма для предсказания трехмерной структуры белкового гомодимерного комплекса показан на рис. 7. Аффинное преобразование, матрица которого была получена в результате процедуры оптимизации, применялось к первому белку гомодимера (на рисунке изображен зеленым цветом), и полученное расположение второго белка (окрашен в синий цвет) сравнивалось с реальным расположением (изображен голубым цветом). Среднеквадратическое отклонение атомов между суперпозициями голубой и синей белковых цепей равно 4,3 Å.

Был также протестирован другой подход, основанный на том, что на вход нейронной сети подавались карты расстояний между атомами СА. Такой подход дал меньшую точность результатов как при предсказании карты контактов, так и при восстановлении структуры комплекса на втором этапе алгоритма. Это объясняется тем, что атомы СА принадлежат главной цепи белка и расстояние между соответствующими атомами СА двух белков больше, чем в случае с атомами СВ, расположенными на боковых цепях белка. Боковые цепи более гибкие, чем главная цепь, поэтому расстояния между атомами боковой цепи лучше фиксируют изменчивость белковых структур.

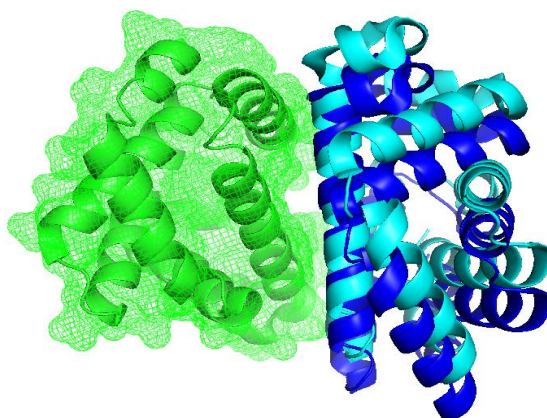


Рис. 7. Пример предсказанной гомодимерной структуры белка, построенной на основе предсказанной карты контактов белкового комплекса

Заключение. Анализ полученных результатов показывает, что разработанный двухэтапный алгоритм позволяет предсказывать трехмерную структуру белковых гомодимеров. Кроме того, разработанная глубокая нейронная сеть дает возможность идентифицировать пары аминокислотных остатков, находящиеся на интерфейсе, и поэтому может быть использована как промежуточный этап при оценивании построенных моделей белковых гомодимеров и фильтрации недопустимых моделей, интерфейсные контакты которых не совпадают с предсказанными.

Разработанный алгоритм восстановления структуры белкового комплекса по предсказанной карте контактов может использоваться не только для гомодимеров, но и для гетеродимеров, состоящих из двух разных белков, поскольку не зависит от количества аминокислотных остатков в одной и другой белковой цепи. Таким образом, результаты исследования свидетельствуют о том, что данный подход может быть обобщен на другие типы белковых комплексов и является перспективным для решения задачи белкового докинга.

References

1. Anfinsen C. B. Principles that govern the folding of protein chains. *Science*, 1973, vol. 181 (4096), pp. 223–230. <https://doi.org/10.1126/science.181.4096.223>
2. Lecun Y., Bengio Y., Hinton G. Deep learning. *Nature*, 2015, vol. 521 (7553), pp. 436–444. <https://doi.org/10.1038/nature14539>
3. Senior A. W., Evans R., Jumper J., Kirkpatrick J., Sifre L., ..., Hassabis D. Improved protein structure prediction using potentials from deep learning. *Nature*, 2020, vol. 577 (7792), pp. 706–710. <https://doi.org/10.1038/s41586-019-1923-7>
4. Billings W. M., Hedelius B., Millicam T., Wingate D., Corte D. D. ProSPR: democratized implementation of alphafold protein distance prediction network. *Biorxiv*, 2019, p. 830273. <https://doi.org/10.1101/830273>
5. Kryshchak A. T., Schwede, Topf M., Fidelis K., Moutl J. Critical assessment of methods of protein structure prediction (CASP) – Round XIII. *Proteins: Structure, Function, and Bioinformatics*, 2019, vol. 87 (12), pp. 1011–1020. <https://doi.org/10.1002/prot.25823>
6. Jones D. T., Kandathil S. M. High precision in protein contact prediction using fully convolutional neural networks and minimal sequence features. *Bioinformatics*, 2018, vol. 34, pp. 3308–3315.
7. Seemayer S., Gruber M., Söding J. CCMpred – fast and precise prediction of protein residue-residue contacts from correlated mutations. *Bioinformatics*, 2014, vol. 30 (21), pp. 3128–3130.
8. Jones D. T., Singh T., Kosciölek T., Tetchner S. MetaPSICOV: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. *Bioinformatics*, 2015, vol. 31 (7), pp. 999–1006.
9. Jiang Q., Jin X., Lee S.-J., Yao S. Protein secondary structure prediction: a survey of the state of the art. *Journal of Molecular Graphics and Modelling*, 2017, vol. 76, pp. 379–402. <https://doi.org/10.1016/j.jm gm.2017.07.015>
10. Skwark M. J., Raimondi D., Michel M., Elofsson A. Improved contact predictions using the recognition of protein like contact patterns. *PLoS Computational Biology*, 2014, vol. 10 (11), p. e1003889. <https://doi.org/10.1371/journal.pcbi.1003889>
11. Berman H. M. The protein data bank: a historical perspective. *Acta Crystallographica Section A: Foundations of Crystallography*, 2008, vol. 64 (1), pp. 88–95. <https://doi.org/10.1107/S0108767307035623>
12. Mou Y., Huang P.-S., Hsu F.-C., Huang S.-J., Mayo S. L. Computational design and experimental verification of a symmetric protein homodimer. *Proceedings of the National Academy of Sciences of the United States of America*, 2015, vol. 112 (34), pp. 10714–10719. <https://doi.org/10.1073/pnas.1505072112>
13. Long J., Shelhamer E., Darrell T. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Boston, Massachusetts, USA, 7–12 June 2015*. Boston, 2015, pp. 3431–3440. <https://doi.org/10.1109/CVPR.2015.7298965>
14. Fu J., Liu J., Tian H., Li Y., Bao Y., ..., Lu H. Dual attention network for scene segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, California, USA, 15–20 June 2019*. Long Beach, 2019, pp. 3141–3149. Available at: <http://arxiv.org/abs/1809.02983> (accessed 27.03.2020).
15. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada, USA, 27–30 June 2016*. Las Vegas, 2016, pp. 770–778. <https://doi.org/10.1109/CVPR.2016.90>
16. Chen L.-C., Papandreou G., Kokkinos I., Murphy K., Yuille A. L. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions*

on *Pattern Analysis and Machine Intelligence*, 2018, vol. 40 (4), pp. 834–848. <https://doi.org/10.1109/TPAMI.2017.2699184>

17. Kingma D. P., Ba J. Adam: a method for stochastic optimization. *3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015*. San Diego, 2015. Available at: <https://arxiv.org/abs/1412.6980> (accessed 27.03.2020).

18. Mitternacht S. FreeSASA: an open source C library for solvent accessible surface area calculations. *F1000Research*, 2016, vol. 5, p. 189. <https://doi.org/10.12688/f1000research.7931.1>

19. Janin J., Bahadur R. P., Chakrabarti P. Protein-protein interaction and quaternary structure. *Quarterly Reviews of Biophysics*, 2008, vol. 41 (2), pp. 133–180. <https://doi.org/10.1017/S0033583508004708>

20. Cock P. J., Antao T., Chang J. T., Chapman B. A., Cox C. J., ..., de Hoon M. J. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 2009, vol. 25 (11), pp. 1422–1423. <https://doi.org/10.1093/bioinformatics/btp163>

Информация об авторах

Хадарович Анна Юрьевна, научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси, Минск, Беларусь.
E-mail: ahadarovich@gmail.com

Калиновский Александр Александрович, научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси, Минск, Беларусь.
E-mail: gakarak@gmail.com

Тузиков Александр Васильевич, член-корреспондент, доктор физико-математических наук, профессор, генеральный директор, Объединенный институт проблем информатики Национальной академии наук Беларуси, Минск, Беларусь.
E-mail: tuzikov@newman.bas-net.by

Information about the authors

Anna Y. Hadarovich, Researcher, the United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus.
E-mail: ahadarovich@gmail.com

Alexander A. Kalinouski, Researcher, the United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus.
E-mail: gakarak@gmail.com

Alexander V. Tuzikov, Corresponding Member, Dr. Sci. (Phys.-Math.), Professor, General Director, the United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus.
E-mail: tuzikov@newman.bas-net.by

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ
COMPUTER-AIDED DESIGN

УДК 004.33.054
<https://doi.org/10.37661/1816-0301-2020-17-2-54-70>

Поступила в редакцию 19.01.2020
Received 19.01.2020

Принята к публикации 25.02.2020
Accepted 25.02.2020

Псевдоисчерпывающее тестирование запоминающих устройств на базе маршевых тестов типа March A

В. Н. Ярмолик¹✉, И. Мрозек², С. В. Ярмолик¹

¹Белорусский государственный университет информатики и радиоэлектроники, Минск, Беларусь
✉E-mail: yarmolik10ru@yahoo.com

²Белостоцкий технический университет, Белосток, Польша

Аннотация. Показывается актуальность тестирования запоминающих устройств современных вычислительных систем. Анализируются методы и алгоритмы реализации тестовых процедур на базе классических маршевых тестов. Выделяются многократные маршевые тесты, позволяющие обнаруживать сложные кодочувствительные неисправности памяти. Для их обнаружения обосновывается необходимое условие, которому должны удовлетворять тестовые процедуры для покрытия сложных неисправностей. Это условие заключается в формировании псевдоисчерпывающего теста для заданного количества произвольных ячеек памяти. Исследуется эффективность однократного и двукратного применения тестов типа MATS++, March C– и March A, а также приводятся ее аналитические оценки для различного количества $k \leq 10$ ячеек памяти, участвующих в неисправности. Обосновывается применимость математической модели комбинаторной задачи собирателя купонов для описания многократного тестирования памяти. Приводятся значения средней, минимальной и максимальной кратности многократных тестов для обеспечения исчерпывающего множества двоичных комбинаций для заданного числа произвольных ячеек памяти. Экспериментально показывается справедливость аналитических оценок и подтверждается высокая эффективность формирования псевдоисчерпывающего покрытия тестами типа March A.

Ключевые слова: тестирование вычислительных систем, встроенное тестирование, многократное тестирование, маршевые тесты памяти, псевдоисчерпывающие тесты

Для цитирования. Ярмолик, В. Н. Псевдоисчерпывающее тестирование запоминающих устройств на базе маршевых тестов типа March A / В. Н. Ярмолик, И. Мрозек, С. В. Ярмолик // Информатика. – 2020. – Т. 17, № 2. – С. 54–70. <https://doi.org/10.37661/1816-0301-2020-17-2-54-70>

Pseudoexhaustive memory testing based on March A type march tests

Vyacheslav N. Yarmolik¹✉, Ireneusz Mrozek², Svetlana V. Yarmolik¹

¹Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus
✉Email: yarmolik10ru@yahoo.com

²Bialystok University of Technology, Bialystok, Poland

Abstract. The relevance of testing of memory devices of modern computing systems is shown. The methods and algorithms for implementing test procedures based on classical March tests are analyzed. Multiple March tests are highlighted to detect complex pattern-sensitive memory faults. To detect them, the necessary condition that test procedures must satisfy to deal complex faults, is substantiated. This condition is in the formation of a pseudo-exhaustive test for a given number of arbitrary memory cells. We study the effectiveness of single and

double application of tests like MATS ++, March C– and March A, and also give its analytical estimates for a different number of $k \leq 10$ memory cells participating in a malfunction. The applicability of the mathematical model of the combinatorial problem of the coupon collector for describing multiple memory testing is substantiated. The values of the average, minimum, and maximum multiplicity of multiple tests are presented to provide an exhaustive set of binary combinations for a given number of arbitrary memory cells. The validity of analytical estimates is experimentally shown and the high efficiency of the formation of a pseudo-exhaustive coverage by tests of the March A type is confirmed.

Keywords: testing of computing systems, embedded testing, multi-run testing, march memory tests, pseudo-exhaustive tests

For citation. Yarmolik V. N., Mrozek I., Yarmolik S. V. Pseudoexhaustive memory testing based on March A type march tests. *Informatics*, 2020, vol. 17, no. 2, pp. 54–70 (in Russian). <https://doi.org/10.37661/1816-0301-2020-17-2-54-70>

Введение. Удельный вес запоминающих устройств в структуре встроенных вычислительных систем достигает 94 % занимаемой системой площади кристалла [1–3]. Поэтому тестирование современных запоминающих устройств представляет собой востребованную и актуальную задачу, для решения которой применяются различные принципы и подходы. Среди них выделяется вероятностное тестирование и его многочисленные разновидности, которые являются эффективной аппроксимацией исчерпывающего и псевдоисчерпывающего тестирования [2–5].

Весьма востребованным методом тестирования запоминающих устройств служит многократное тестирование с изменяемыми начальными условиями на базе классических маршевых тестов [6–9]. Реализация многократного тестирования предполагает выполнение определенных процедур, основанных на изменении либо начального состояния запоминающего устройства, либо его адресной последовательности [6, 10–13]. Применение вероятностных принципов при формировании как адресных последовательностей, так и начальных состояний памяти позволило построить математическую модель тестирования памяти и оценить ее эффективность [2, 5].

Целью настоящей статьи является анализ эффективности многократных маршевых тестов запоминающих устройств с изменяемым начальным состоянием памяти. Объектом исследования служат три специфических множества маршевых тестов, таких как тесты типа MATS++, March C– и March A. Приводятся основные аналитические соотношения эффективности тестов, а также описываются их сложности для формирования псевдоисчерпывающих тестовых наборов, обеспечивающих всевозможные 2^k двоичные комбинации в произвольных k из N запоминающих ячеек памяти.

Однократные маршевые тесты. Маршевые тесты запоминающих устройств формируют в k произвольных ячейках памяти так называемую орбиту, которая представляет собой набор двоичных k -разрядных векторов [4, 11]. Конкретный набор векторов в орбите зависит от трех основных факторов [4]. Прежде всего однозначно определяет структуру орбиты используемый маршевый тест, так как он формулирует правила, по которым генерируются двоичные векторы в ячейках запоминающего устройства. Начальное состояние запоминающего устройства – второй фактор, от которого зависит вид конкретной орбиты. Последовательность формируемых маршевым тестом адресов, представляющая собой порядок обращения к запоминающим ячейкам памяти, является третьим фактором.

Для классических разрушающих маршевых тестов начальным состоянием памяти будет нулевое состояние, формирующееся в результате реализации фазы инициализации $\uparrow\downarrow(w0)$ теста [2, 4, 9]. Символ \uparrow обозначает возрастающую, а символ \downarrow убывающую последовательность адресов, которые отличаются взаимнообратным порядком генерирования, а совместное применение символов $\uparrow\downarrow$ обозначает безразличный порядок адресов [9]. Под фазой маршевого теста понимают набор операций чтения r и записи w нулевого ($r0, w0$) либо единичного ($r1, w1$) значения для каждой ячейки запоминающего устройства. Все операции одной фазы теста выделяются круглыми скобками, перед которыми указывается порядок адресов для данной фазы, а сами фазы в тесте разделяются символом «;» [9]. Последующие фазы теста за фазой инициализации, как правило, используют возрастающий либо убывающий порядок адресов, как это, например, определено для теста MATS ++ $\{\uparrow\downarrow(w0); \uparrow(r0, w1); \downarrow(r1, w0, r0)\}$ и теста March C– $\{\uparrow\downarrow(w0); \uparrow(r0, w1); \uparrow(r1, w0); \downarrow(r0, w1); \downarrow(r1, w0); \uparrow\downarrow(r0)\}$.

В работе [4] исследованы маршевые тесты типа MATS ++ и March C⁻, для которых данные из ячеек памяти считываются и в них записываются инверсные значения. Перед выполнением очередной фазы теста состояние памяти может быть нулевым либо единичным. Для $k > 1$ из N , где N – емкость памяти в битах произвольных ячеек памяти, во время выполнения фаз формируется возрастающая или обратная убывающая последовательность адресов. Поэтому в маршевых тестах типа MATS ++ и March C⁻ существует четыре различные орбиты [4]. Количество орбит и их порядок во время выполнения теста зависят от применяемого маршевого теста. Возможные орбиты для классических маршевых тестов типа MATS ++ и March C⁻, формируемые в произвольных k ячейках памяти, приведены в табл. 1 [4].

Таблица 1

Орбиты, формируемые маршевыми тестами типа MATS ++ и March C ⁻				
Орбита	O_0	O_1	O_2	O_3
Предыдущая фаза теста	(..., w0, ...)	(..., w0, ...)	(..., w1, ...)	(..., w1, ...)
Фаза теста	$\uparrow(\dots, w1, \dots)$	$\downarrow(\dots, w1, \dots)$	$\uparrow(\dots, w0, \dots)$	$\downarrow(\dots, w0, \dots)$
P_0	0 0 0 ... 0 0	0 0 0 ... 0 0	1 1 1 ... 1 1	1 1 1 ... 1 1
P_1	0 0 0 ... 0 1	1 0 0 ... 0 0	1 1 1 ... 1 0	0 1 1 ... 1 1
P_2	0 0 0 ... 1 1	1 1 0 ... 0 0	1 1 1 ... 0 0	0 0 1 ... 1 1
...
P_{k-2}	0 0 1 ... 1 1	1 1 1 ... 0 0	1 1 0 ... 0 0	0 0 0 ... 1 1
P_{k-1}	0 1 1 ... 1 1	1 1 1 ... 1 0	1 0 0 ... 0 0	0 0 0 ... 0 1
P_k	1 1 1 ... 1 1	1 1 1 ... 1 1	0 0 0 ... 0 0	0 0 0 ... 0 0

При описании предыдущей и текущей фаз теста, приведенных в табл. 1, показана последняя операция записи в фазе, после которой могут быть только операции чтения. Отметим, что орбиты O_0 , O_1 , O_2 и O_3 формируются фазами, которые инвертируют содержимое запоминающего устройства. Их применение многообразно и широко представлено в классических маршевых тестах. Например, фаза $\uparrow(r0, w1, r1, w0, r0, w1)$ теста March A и фаза $\uparrow(r0, w1, w0, w1, r1)$ теста March LA, так же как и фаза $\uparrow(r0, w1)$ теста MATS ++, формируют орбиту O_0 . В зависимости от количества фаз и их сочетаний в тесте формируется некоторое подмножество орбит [4]. Маршевый тест MATS ++ генерирует две орбиты, а именно O_0 во время реализации его фазы $\uparrow(r0, w1)$ и O_3 в результате применения фазы $\downarrow(r1, w0, r0)$, которые состоят из одинаковых двоичных наборов, сформированных в обратном порядке. В то же время March C⁻ позволяет генерировать все четыре возможные орбиты $O_i \in \{O_0, O_1, O_2, O_3\}$, что доказывает его более высокую эффективность [4].

Для тестов типа MATS++ и March C⁻ приведем следующее определение [4].

Определение 1. Орбита O_i , $i \in \{0, 1, 2, 3\}$, сформированная в произвольных k ячейках памяти как результат применения одной из фаз маршевого теста типа MATS++ и March C⁻, состоит из k -разрядных двоичных векторов P_0, P_1, \dots, P_k .

Данное определение обобщается для произвольной фазы любого маршевого теста и оценивает сложность орбиты, формируемой фазой маршевого теста. В дальнейшем будем придерживаться обозначения орбит O_i , формируемых тестами типа MATS++ и March C⁻, для которых справедливы следующие два определения [4].

Определение 2. Две орбиты O_j и O_r считаются эквивалентными или равными ($O_j = O_r$), если обе включают одинаковые двоичные векторы независимо от их порядка.

В случае маршевых тестов типа MATS ++ и March C⁻ справедливы равенства $O_0 = O_3$ и $O_1 = O_2$ (см. табл. 1). Для целей сравнения орбит введем меру расстояния $D(O_j, O_r)$ между орбитами O_j и O_r .

Определение 3. Расстоянием $D(O_j, O_r) = D(O_r, O_j)$ между орбитами O_j и O_r является количество неповторяющихся k -разрядных двоичных векторов P_0, P_1, \dots, P_k , входящих в их состав.

Тогда для эквивалентных орбит $O_j = O_r$ расстояние $D(O_j, O_r) = 0$, а для орбит O_j и O_r , которые состоят из неповторяющихся двоичных векторов P_0, P_1, \dots, P_k , входящих в их состав, $D(O_j, O_r) = k + 1$. Анализируя орбиты, представленные в табл. 1, можно показать, что $D(O_0, O_1) = D(O_2, O_3) = D(O_0, O_2) = D(O_1, O_3) = k - 1$, а $D(O_0, O_3) = D(O_1, O_2) = 0$.

Множество разновидностей орбит не ограничивается орбитами O_i , формируемыми тестами типа MATS ++ и March C-. Вторым подмножеством орбит Q_i являются орбиты, формируемые такими маршевыми тестами, как March A, March B, March LA, March LR, March M, March PS и др. [2, 7–9]. В дальнейшем подобные тесты будем называть маршевыми тестами типа March A, которые позволяют формировать орбиты Q_i , $i \in \{0, 1, 2, 3\}$, приведенные в табл. 2.

Таблица 2

Орбиты, формируемые маршевыми тестами типа March A

Орбита	Q_0	Q_1	Q_2	Q_3
Предыдущая фаза теста	(..., w0, ...)	(..., w0, ...)	(..., w1, ...)	(..., w1, ...)
Фаза теста	$\uparrow(\dots, w1, \dots, w0, \dots)$	$\downarrow(\dots, w1, \dots, w0, \dots)$	$\uparrow(\dots, w0, \dots, w1, \dots)$	$\downarrow(\dots, w0, \dots, w1, \dots)$
P_0	0 0 0 ... 0 0	0 0 0 ... 0 0	1 1 1 ... 1 1	1 1 1 ... 1 1
P_1	0 0 0 ... 0 1	1 0 0 ... 0 0	1 1 1 ... 1 0	0 1 1 ... 1 1
P_2	0 0 0 ... 1 0	0 1 0 ... 0 0	1 1 1 ... 0 1	1 0 1 ... 1 1
...
P_{k-2}	0 0 1 ... 0 0	0 0 1 ... 0 0	1 1 0 ... 1 1	1 1 0 ... 1 1
P_{k-1}	0 1 0 ... 0 0	0 0 0 ... 1 0	1 0 1 ... 1 1	1 1 1 ... 0 1
P_k	1 0 0 ... 0 0	0 0 0 ... 0 1	0 1 1 ... 1 1	1 1 1 ... 1 0

Для орбит Q_i , так же как и для орбит O_i , применимы определения 1–3 и следующее утверждение.

Утверждение 1. Расстояние Хэмминга между двумя векторами орбиты Q_i для k произвольных ячеек запоминающего устройства, расположенных по возрастанию их адресов, равняется $HD(P_0, P_r) = 1$ для $r \in \{1, 2, \dots, k\}$ и $HD(P_l, P_r) = 2$ для $l \neq r \in \{1, 2, \dots, k\}$.

Следствием утверждения 1 является отсутствие в пределах орбиты Q_i двух одинаковых двоичных векторов P_j и P_l для $j \neq l \in \{0, 1, 2, \dots, k\}$. Так же как и для O_i , множество орбит Q_i содержит эквивалентные орбиты, которые состоят из одинаковых двоичных векторов. Действительно, из табл. 2 видно, что $Q_0 = Q_1$ и $Q_2 = Q_3$. Соответственно, можно показать, что $D(Q_0, Q_1) = D(Q_2, Q_3) = 0$. Кроме того, $D(Q_0, Q_2) = D(Q_0, Q_3) = D(Q_1, Q_2) = D(Q_1, Q_3) = k + 1$. Отметим, что по сравнению с множеством орбит O_i в случае орбит Q_i достигается большее расстояние между ними, равное $k + 1$. Максимальное количество k -разрядных двоичных наборов для $k > 2$, равное $2k + 2$, может быть получено в результате применения фаз маршевого теста, формирующего, например, орбиты Q_0 и Q_3 .

Дальнейшее увеличение количества двоичных наборов возможно только путем формирования орбит из множества O_i . Каждая из орбит O_i содержит наборы, состоящие из всех единичных 1 1 1 ... 1 1 и всех нулевых 0 0 0 ... 0 0 значений, которые формируются и орбитами Q_0 и Q_3 . Кроме того, орбиты O_0 и O_2 содержат наборы 0 0 0 ... 0 1, 0 1 1 ... 1 1 и 1 0 0 ... 0 0, 1 1 1 ... 1 0, входящие в состав орбит Q_0 и Q_3 соответственно. Поэтому любая орбита O_i формирует только $k + 1 - 4 = k - 3$, $k > 3$, дополнительных k -разрядных двоичных наборов по отношению к наборам, сгенерированным ранее орбитами Q_0 и Q_3 . Таким образом, максимально возможное количество двоичных наборов в k ячейках памяти, формируемых однократным применением маршевого теста, будет определяться как $2k + 2 + 2(k - 3) = 4(k - 1)$. При этом необходимым условием получения такого количества наборов является присутствие в тесте фаз, формирующих соответствующие орбиты. Например, орбиты Q_0 и Q_3 совместно с орбитами O_0 и O_2 позволяют сформировать $4(k - 1)$ двоичных наборов. Одним из маршевых тестов, который формирует максимальное количество двоичных наборов, может быть тест

$$\{\uparrow\downarrow(w0); \uparrow(r0, w1); \downarrow(r1, w0, w1); \uparrow(r1, w0); \downarrow(r0, w1, w0) \uparrow(r0)\}. \quad (1)$$

В дальнейшем данный тест будем называть March O. Сложность данного теста равняется $12N$, и он позволяет формировать $4(k - 1)$ k -разрядных двоичных векторов (табл. 3).

Таблица 3

Орбиты, формируемые маршевым тестом *March O* для $k = 5$

Фаза теста	$\uparrow(r0, w1)$	$\downarrow(r1, w0, w1)$	$\uparrow(r1, w0)$	$\downarrow(r0, w1, w0)$
Орбита	O_0	Q_3	O_2	Q_1
P_0	0 0 0 0 0	1 1 1 1 1	1 1 1 1 1	0 0 0 0 0
P_1	0 0 0 0 1	0 1 1 1 1	1 1 1 1 0	1 0 0 0 0
P_2	0 0 0 1 1	1 0 1 1 1	1 1 1 0 0	0 1 0 0 0
P_3	0 0 1 1 1	1 1 0 1 1	1 1 0 0 0	0 0 1 0 0
P_4	0 1 1 1 1	1 1 1 0 1	1 0 0 0 0	0 0 0 1 0
P_5	1 1 1 1 1	1 1 1 1 0	0 0 0 0 0	0 0 0 0 1

Тест *March O* формирует в произвольных $k = 5$ ячейках запоминающего устройства $4(k - 1) = 4(5 - 1) = 16$ двоичных наборов. Отметим, что тесты типа *MATS++* и *March C-* для того же значения $k = 5$ соответственно формируют $k + 1 = 6$ и $2k = 10$ двоичных наборов. Для обеспечения эффективного обнаружения сложных неисправностей запоминающих устройств аналогичные сочетания орбит двух видов O_i и Q_i используются в классических маршевых тестах (рис. 1).

Для всех приведенных на рис. 1 маршевых тестов показаны сложность их реализации, оцениваемая количеством циклов обращения к запоминающему устройству емкостью N бит, а также фазы теста, которые формируют орбиты вида O_i и Q_i .

March A (15N)							
$\uparrow\downarrow(w0)$	$\uparrow(r0, w1, w0, w1)$	$\uparrow(r1, w0, w1)$	$\downarrow(r1, w0, w1, w0)$	$\downarrow(r0, w1, w0)$			
	O_0	Q_2	O_3	Q_1			
Algorithm B (17N)							
$\uparrow\downarrow(w0)$	$\uparrow(r0, w1, w0, w1)$	$\uparrow(r1, w0, r0, w1)$	$\downarrow(r1, w0, w1, w0)$	$\downarrow(r0, w1, r1, w0)$			
	O_0	Q_2	O_3	Q_1			
March U (13N)							
$\uparrow\downarrow(w0)$	$\uparrow(r0, w1, r1, w0)$	$\uparrow(r0, w1)$	$\downarrow(r1, w0, r0, w1)$	$\downarrow(r1, w0)$			
	Q_0	O_0	Q_3	O_3			
March LR (14N)							
$\uparrow\downarrow(w0)$	$\downarrow(r0, w1)$	$\uparrow(r1, w0, r0, w1)$	$\uparrow(r1, w0)$	$\uparrow(r0, w1, r1, w0)$	$\downarrow(r0)$		
	O_1	Q_2	O_0	Q_0			
March M (16N)							
$\uparrow\downarrow(w0)$	$\uparrow(r0, w1, r1, w0)$	$\uparrow\downarrow(r0)$	$\uparrow(r0, w1)$	$\uparrow\downarrow(r1)$	$\downarrow(r1, w0, r0, w1)$	$\uparrow\downarrow(r1)$	$\downarrow(r1, w0)$
	Q_0		O_0		Q_3		O_3
March PS (23N)							
$\uparrow\downarrow(w0)$	$\uparrow(r0, w1, r1, w0, r0, w1)$	$\uparrow(r1, w0, r0, w1, r1)$	$\uparrow(r1, w0, r0, w1, r1, w0)$	$\uparrow(r0, w1, r1, w0, r0)$			
	O_0	Q_2		O_2			Q_0

Рис. 1. Примеры маршевых тестов

Ранее указывалось, что максимально возможное количество двоичных наборов в k ячейках памяти, формируемых однократным применением маршевого теста, будет определяться как $4(k - 1)$, однако это достигается существенно большей сложностью теста, формирующего нужное сочетание орбит (см. рис. 1).

Двукратные маршевые тесты. Приведенные в предыдущем разделе результаты анализа позволяют сделать следующие выводы. Однократное применение маршевого теста дает возможность получить только одну, две или четыре неэквивалентные орбиты, которые зависят от начального состояния памяти и используемой адресной последовательности. В случае тестов типа *MATS++* применение маршевого теста позволяет генерировать одну орбиту, включающую $k + 1$ различных k -разрядных двоичных векторов. Две орбиты, включающие $2k$ неповторяющихся двоичных векторов, формируются тестами типа *March C-*. Максимальное количе-

ство k -разрядных двоичных векторов, равное $4(k - 1)$, обеспечивается четырьмя орбитами, генерируемыми тестами типа March A.

Для формирования всевозможных 2^k двоичных векторов в $k > 1$ произвольных ячейках памяти маршевый тест типа MATS ++ должен выполняться многократно при различных начальных условиях, обеспечивающих генерирование неэквивалентных орбит в k ячейках. Для $k > 2$ обеспечение 2^k двоичных векторов тестами типа MATS ++ и March C- достигается их применением более одного раза. Неоднократное применение любого маршевого теста, включая тесты типа March A, необходимо для достижения той же цели для $k > 3$. Во всех случаях повторное применение теста осуществляется при использовании различных начальных состояний памяти и (или) различных адресных последовательностей [2, 7, 8]. Подход, основанный на изменении адресных последовательностей, был ранее всесторонне исследован в работах [2, 7, 13], а первоначальные результаты по изменяемым начальным состояниям памяти представлены в [2, 4, 14]. В большей своей части они относились к тестам типа MATS ++. Для этих тестов был доказан и исследован ряд положений, касающихся орбит O_i , которые формируются тестами типа MATS++.

Двукратные маршевые тесты типа MATS++. Для тестов типа MATS++ справедливо следующее утверждение [4].

Утверждение 2. Для произвольных $1 < k \leq N$ ячеек памяти с адресами $\beta, \gamma, \dots, \delta, \varepsilon, \eta$, где $N \geq \beta > \gamma > \dots > \delta > \varepsilon > \eta \geq 0$, запоминающего устройства и различных 2^k их начальных состояний $P_0 = a_\beta a_\gamma \dots a_\delta a_\varepsilon a_\eta$ маршевый тест типа MATS++ при использовании фиксированной адресной последовательности, возрастающей либо убывающей, генерирует 2^k уникальных неэквивалентных орбит O_i .

Данное утверждение гарантирует формирование тестами типа MATS++ множества из $k + 1$ двоичного вектора, причем изменение начального состояния P_0 в указанных ячейках приводит к формированию очередных $k + 1$ векторов, включающих векторы, не совпадающие с ранее сгенерированными двоичными векторами. Для оценки эффективности двукратного применения тестов типа MATS++ в работе [4] приведено следующее утверждение.

Утверждение 3. Для заданного k , $1 < k \leq N$, и постоянного порядка адресов существует $2^k - (k^2 + k)/2 - 1$ орбит O_i , формируемых повторным применением теста типа MATS++ с отличающимися начальными состояниями P_0 и состоящих из двоичных векторов, которые не входят в исходную орбиту O_0 , и $(k^2 + k)/2$ орбит, включающих по два вектора, которые входят в орбиту O_0 .

Таким образом, изменение начального состояния P_0 ячеек памяти приводит к формированию только двух видов орбит. Их точное число, равное $M_1 = (k^2 + k)/2$ и $M_2 = 2^k - (k^2 + k)/2 - 1$, позволяет оценить среднее число M_{ave} двоичных векторов, формируемых в результате двукратного маршевого теста типа MATS ++. Основываясь на общей идее многократного тестирования запоминающих устройств [4], исходные состояния ячеек памяти перед последующим применением маршевого теста формируются как случайные равномерно распределенные значения 0 и 1. Это означает, что для тех же k ячеек памяти новая орбита формируется с равной вероятностью из всех возможных 2^k орбит. Тогда среднее количество новых двоичных векторов, формируемых во время повторного применения теста MATS ++, будет равняться $[M_1 \times (k - 1) + M_2 \times (k + 1)]/2^k$, а значение среднего числа $M_{ave}(\text{MATS ++})$ двоичных векторов, формируемых в результате двукратного маршевого теста MATS ++, определяется соотношением

$$M_{ave}(\text{MATS ++}) = (k + 1) + \frac{M_1 \times (k - 1) + M_2 \times (k + 1)}{2^k} = 2(k + 1) - \frac{(k + 1)^2}{2^k}. \quad (2)$$

Очевидно, что значение M_{ave} существенно увеличится при применении тестов, формирующих более сложные орбиты и большее их число, например тестов типа March C-.

Двукратные маршевые тесты типа March C-. Для оценки эффективности двукратного применения тестов типа March C- отметим, что подобные тесты формируют две неэквивалент-

ные орбиты вида O_i , которыми в зависимости от применяемого теста могут быть O_0 и O_1 , или O_2 и O_3 , либо обе пары орбит вместе. В дальнейшем будем применять обозначения O_{01} и O_{23} для идентификации указанных пар орбит. Данные пары орбит имеют только два повторяющихся двоичных набора P_0 и P_k (см. табл. 1), поэтому их совместное формирование при однократном применении теста обеспечивает в произвольных k ячейках памяти $(k+1) + (k+1) - 2 = 2k$ двоичных наборов. Рассмотрим случай пары орбит O_0 и O_1 (O_{01}), для которых начальное состояние $P_0(O_0)$ первой орбиты O_0 равняется начальному состоянию $P_0(O_1)$ второй орбиты O_1 . Для финальных состояний обеих орбит также выполняется равенство $P_k(O_0) = P_k(O_1)$. Остальные k -разрядные двоичные векторы P_1, P_2, \dots, P_{k-1} , входящие в обе орбиты, связаны соотношением $P_i(O_0) = \overline{P_{k-i}(O_1)}$, $i = \overline{1, k-1}$. Например, для $i = 1$ $P_1(O_0) = 0\ 0\ 0 \dots 0\ 1 = P_{k-1}(O_1) = 1\ 1\ 1 \dots 1\ 0$.

При произвольных k , $2 < k \leq N$, ячейках памяти с адресами $\beta, \gamma, \dots, \delta, \epsilon, \eta$, где $N \geq \beta > \gamma > \dots > \delta > \epsilon > \eta \geq 0$, запоминающего устройства и различных 2^k их начальных состояний $P_0 = a_\beta a_\gamma \dots a_\delta a_\epsilon a_\eta$, где $a_i \in \{0, 1\}$, $i \in \{\beta, \gamma, \dots, \delta, \epsilon, \eta\}$, а $\overline{a_i}$ принимает инверсное значение относительно a_i , структура всего множества неповторяющихся $2k$ двоичных векторов для случая O_{01} определена в табл. 4.

Таблица 4

Пара орбит O_{01} , состоящая из $2k$ двоичных векторов

Вектор	Индексы разрядов вектора					
	k	$k-1$...	3	2	1
P_0	a_β	a_γ	...	a_δ	a_ϵ	a_η
P_1	a_β	a_γ	...	a_δ	a_ϵ	$\overline{a_\eta}$
P_2	a_β	a_γ	...	a_δ	$\overline{a_\epsilon}$	$\overline{a_\eta}$
P_3	a_β	a_γ	...	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$
...
P_{k-1}	a_β	a_γ	...	a_δ	$\overline{a_\epsilon}$	$\overline{a_\eta}$
P_k	$\overline{a_\beta}$	$\overline{a_\gamma}$...	$\overline{a_\delta}$	a_ϵ	a_η
P_{k+1}	$\overline{a_\beta}$	$\overline{a_\gamma}$...	a_δ	a_ϵ	a_η
P_{k+2}	$\overline{a_\beta}$	$\overline{a_\gamma}$...	a_δ	a_ϵ	a_η
...
P_{2k-2}	$\overline{a_\beta}$	$\overline{a_\gamma}$...	$\overline{a_\delta}$	a_ϵ	a_η
P_{2k-1}	$\overline{a_\beta}$	$\overline{a_\gamma}$...	$\overline{a_\delta}$	$\overline{a_\epsilon}$	a_η

Для всех возможных $2^3 = 8$ ($k = 3$) начальных состояний $P_0(O_{01})$ соответствующие восемь орбит, полученные согласно приведенной в табл. 4 процедуре, представлены в табл. 5. Для обозначения конкретной орбиты в скобках указано начальное состояние трех ячеек, входящих в орбиту. Например, обозначение $O_{01}(011)$ определяет пару орбит с начальным состоянием $P_0(O_{01}) = 0\ 1\ 1$.

Таблица 5

Пары орбит для $k = 3$

Вектор	$O_{01}(000)$	$O_{01}(001)$	$O_{01}(010)$	$O_{01}(011)$	$O_{01}(100)$	$O_{01}(101)$	$O_{01}(110)$	$O_{01}(111)$
P_0	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
P_1	0 0 1	0 0 0	0 1 1	0 1 0	1 0 1	1 0 0	1 1 1	1 1 0
P_2	0 1 1	0 1 0	0 0 1	0 0 0	1 1 1	1 1 0	1 0 1	1 0 0
P_3	1 1 1	1 1 0	1 0 1	1 0 0	0 1 1	0 1 0	0 0 1	0 0 0
P_4	1 0 0	1 0 1	1 1 0	1 1 1	0 0 0	0 0 1	0 1 0	0 1 1
P_5	1 1 0	1 1 1	1 0 0	1 0 1	0 1 0	0 1 1	0 0 0	0 0 1

Следует отметить, что произвольные пары орбит O_{01} для $P_0(O_{01}) \neq 0\ 0\ 0 \dots 0$ формируются путем инвертирования столбцов исходной пары орбит $O_{01}(000\dots 0)$, соответствующих ненулевым значениям $P_0(O_{01})$. Например, пара орбит $O_{01}(011)$ формируется путем инвертирования первого и второго столбцов пары орбит $O_{01}(000)$, приведенной в табл. 5.

Из равенства $P_i(O_0) = \overline{P_{k-i}(O_1)}$, $i = 0, k$, справедливого для орбит O_0 и O_1 , следует, что использование инверсного начального состояния при формировании пары орбит O_{01} приводит к генерированию эквивалентного множества k -разрядных двоичных векторов, т. е. $O_{01}(P_0) = O_{01}(\overline{P_0})$. Например, из табл. 5 видно, что $O_{01}(000) = O_{01}(111)$. Таким образом, в результате случайного выбора начального состояния $P_0(O_{01})$ для пары орбит O_{01} при повторном применении теста типа March C– в двух случаях из 2^k возможных будут сгенерированы множества идентичных k -разрядных двоичных векторов. Множества повторяющихся векторов формируются при выборе того же самого начального состояния $P_0(O_{01})$, что и для исходной пары орбит, или его инверсного значения $\overline{P_0(O_{01})}$.

Повторное применение теста типа March C– может увеличить количество двоичных комбинаций, формируемых в произвольных k ячейках только при условии изменения их начального состояния P_0 . Для общего случая пары орбит O_{01} с начальным состоянием $P_0 = a_\beta a_\gamma \dots a_\delta a_\epsilon a_\eta$ новая пара орбит O_{01}^n будет получена в результате инвертирования r ($k > r \geq 1$) разрядов для всех двоичных k -разрядных векторов исходной пары орбит O_{01} . В качестве примера две новые пары орбит O_{01}^{n1} и O_{01}^{n2} как результат инвертирования определенных разрядов в векторах орбиты O_{01} для $k = 5$ показаны в табл. 6. В данном случае использовалось инвертирование соседних разрядов векторов, поэтому отрицания представляют собой единый непрерывный блок.

Пара орбит O_{01}^{n1} является результатом отрицания всего лишь одного разряда с индексом 2 для всех векторов O_{01} , а пара O_{01}^{n2} получена в результате инвертирования блока разрядов векторов с индексами 2, 3 и 4. В результате обе пары орбит O_{01}^{n1} и O_{01}^{n2} имеют шесть новых векторов и включают в себя по четыре вектора из оригинальной пары орбит O_{01} . Действительно, в O_{01}^{n1} используются векторы P_1, P_2 , входящие в O_0 , и инверсные векторы P_8, P_9 из орбиты O_1 , а в O_{01}^{n2} используются соответственно P_1, P_4 и P_6, P_9 (табл. 6).

Таблица 6

Примеры применения одного блока инверсий при формировании O_{01}^n для $k = 5$

Вектор	O_{01}					O_{01}^{n1}					O_{01}^{n2}				
	5	4	3	2	1	5	4	3	2	1	5	4	3	2	1
P_0	a_β	a_γ	a_δ	a_ϵ	a_η	a_β	a_γ	a_δ	a_ϵ	a_η	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$
P_1	a_β	a_γ	a_δ	a_ϵ	$\overline{a_\eta}$	a_β	a_γ	a_δ	a_ϵ	$\overline{a_\eta}$	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$
P_2	a_β	a_γ	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$	a_β	a_γ	a_δ	a_ϵ	$\overline{a_\eta}$	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	$\overline{a_\eta}$
P_3	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	$\overline{a_\eta}$	a_β	$\overline{a_\gamma}$	a_δ	a_ϵ	$\overline{a_\eta}$
P_4	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	$\overline{a_\eta}$	$\overline{a_\beta}$	a_γ	a_δ	a_ϵ	$\overline{a_\eta}$
P_5	$\overline{a_\beta}$	a_γ	a_δ	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	a_δ	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	a_δ	a_ϵ	a_η
P_6	$\overline{a_\beta}$	a_γ	a_δ	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	a_δ	$\overline{a_\epsilon}$	a_η	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\epsilon}$	a_η
P_7	$\overline{a_\beta}$	$\overline{a_\gamma}$	a_δ	a_ϵ	a_η	$\overline{a_\beta}$	$\overline{a_\gamma}$	a_δ	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	$\overline{a_\delta}$	$\overline{a_\epsilon}$	a_η
P_8	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	a_η	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	a_δ	a_ϵ	a_η
P_9	$\overline{a_\beta}$	a_γ	a_δ	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	$\overline{a_\delta}$	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	a_δ	a_ϵ	a_η

Обобщая результаты приведенного анализа, можно сделать следующий вывод. Для случая одного блока последовательных отрицаний, примененных для O_{01} с целью получения новой пары орбит O_{01}^n (см. табл. 4 и 5), результирующая новая пара орбит включает только четыре двоичных вектора исходной пары орбит O_{01} . Остальные векторы отличаются от векторов, вхо-

дящих в O_{01} . Отметим, что для любого k существует k блоков с одним отрицанием, $k - 1$ блок с двумя последовательными отрицаниями, $k - 2$ блока с тремя последовательными отрицаниями и т. д.

Количество пар орбит, полученных в результате инвертирования последовательного блока бит для начального состояния P_0 исходной пары орбит O_{01} , может быть рассчитано следующим образом:

$$M = k + (k - 1) + (k - 2) + \dots + 2 + 1 = (k^2 + k) / 2. \quad (3)$$

Указанные пары орбит O_{01}^n включают в себя четыре вектора из O_{01} .

Применение других наборов отрицаний для получения новой пары орбит, которые включают в себя более одного блока последовательных отрицаний, приводит к получению абсолютно новой пары орбит за исключением одного случая при использовании двух блоков отрицаний. В табл. 7 показаны примеры новых пар орбит O_{01}^{n3} и O_{01}^{n4} как результат применения двух блоков отрицаний.

Таблица 7

Примеры применения более одного блока инверсий при получении O_{01} для $k = 5$

Вектор	O_{01}					O_{01}^{n3}					O_{01}^{n4}				
	5	4	3	2	1	5	4	3	2	1	5	4	3	2	1
P_0	a_β	a_γ	a_δ	a_ϵ	a_η	a_β	$\overline{a_\gamma}$	a_δ	$\overline{a_\epsilon}$	$\overline{a_\eta}$	$\overline{a_\beta}$	$\overline{a_\gamma}$	a_δ	$\overline{a_\epsilon}$	$\overline{a_\eta}$
P_1	a_β	a_γ	a_δ	a_ϵ	a_η	a_β	$\overline{a_\gamma}$	a_δ	a_ϵ	a_η	$\overline{a_\beta}$	$\overline{a_\gamma}$	a_δ	a_ϵ	a_η
P_2	a_β	a_γ	a_δ	a_ϵ	a_η	a_β	$\overline{a_\gamma}$	a_δ	a_ϵ	a_η	$\overline{a_\beta}$	$\overline{a_\gamma}$	a_δ	a_ϵ	a_η
P_3	a_β	a_γ	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	a_η	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	a_η
P_4	a_β	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$	a_β	a_γ	$\overline{a_\delta}$	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	$\overline{a_\delta}$	a_ϵ	a_η
P_5	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$	$\overline{a_\beta}$	a_γ	$\overline{a_\delta}$	a_ϵ	a_η	a_β	a_γ	$\overline{a_\delta}$	a_ϵ	a_η
P_6	$\overline{a_\beta}$	a_γ	a_δ	a_ϵ	a_η	$\overline{a_\beta}$	$\overline{a_\gamma}$	a_δ	$\overline{a_\epsilon}$	$\overline{a_\eta}$	a_β	$\overline{a_\gamma}$	a_δ	$\overline{a_\epsilon}$	$\overline{a_\eta}$
P_7	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$	a_β	a_γ	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$
P_8	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	$\overline{a_\delta}$	a_ϵ	$\overline{a_\eta}$	a_β	a_γ	$\overline{a_\delta}$	$\overline{a_\epsilon}$	$\overline{a_\eta}$
P_9	$\overline{a_\beta}$	$\overline{a_\gamma}$	$\overline{a_\delta}$	a_ϵ	a_η	$\overline{a_\beta}$	a_γ	$\overline{a_\delta}$	a_ϵ	a_η	a_β	a_γ	$\overline{a_\delta}$	a_ϵ	a_η

Новая пара орбит O_{01}^{n3} содержит k -разрядные векторы, непересекающиеся с векторами множества O_{01} , а пара орбит O_{01}^{n4} содержит повторяющиеся векторы P_2, P_3, P_7, P_8 , которые соответствуют векторам P_7, P_8, P_2, P_3 исходной пары орбит O_{01} . Четыре повторяющихся вектора в новой паре орбит обусловлены симметрией исходных орбит O_0 и O_1 , образующих пару O_{01} . Этот эффект возникает в случае двух блоков отрицания, в которые входят первый и k -й разряды векторов орбит. Число таких пар орбит будет вычисляться аналогично соотношению (3) для значения k , равного $k - 2$.

Общее количество новых пар орбит O_{01}^n , включающих в себя четыре вектора из O_{01} , определяется выражением

$$M_1 = M + (k - 2) + (k - 3) + \dots + 2 + 1 = \frac{(k^2 + k)}{2} + \frac{(k - 2)^2 + k - 2}{2} = k^2 - k. \quad (4)$$

Число M_2 новых пар орбит, состоящее из двоичных векторов, которые не участвуют в исходной паре орбит O_{01} , определяется как

$$M_2 = 2^k - (k^2 - k) - 2. \quad (5)$$

Обобщая приведенный анализ, можно сформулировать следующее утверждение, которое справедливо как для пар орбит O_{01} , так и для O_{23} , генерируемых тестами типа March C–.

Утверждение 4. Для заданного k , $2 < k \leq N$, и постоянного порядка адресов существует $2^k - k^2 + k - 2$ пар орбит O_{01} (O_{23}), формируемых тестами типа March C– с другими начальными состояниями P_0 и состоящих из двоичных векторов, которые не входят в исходную пару орбит O_{01} , (O_{23}), а также $k^2 - k$ пар орбит, включающих по четыре вектора, которые входят в две исходные орбиты O_{01} (O_{23}).

Среднее количество новых двоичных векторов, формируемых во время повторного применения теста типа March C–, будет равняться $[M_1 \times (2k - 4) + M_2 \times (2k)] / 2^k$, а значение среднего числа $M_{ave}(\text{March C–})$ двоичных векторов, формируемых в результате применения двукратного маршевого теста March C–, определяется соотношением

$$M_{ave}(\text{March C–}) = 2k + \frac{M_1 \times (2k - 4) + M_2 \times (2k)}{2^k} = 4k - \frac{k^2}{2^{k-2}}. \quad (6)$$

Очевидно, что значение M_{ave} существенно увеличится при применении тестов, формирующих большее число разнообразных орбит.

Двукратные маршевые тесты типа March A. Первоначально отметим, что тесты типа March A формируют орбиты вида Q_i , приведенные в табл. 2. Маршевые тесты, такие как March A, March B, March LA, March LR, March M, March PS и др., как правило, формируют пары орбит $Q_1, Q_2; Q_0, Q_2$ и Q_0, Q_3 (см. рис. 1). Для идентификации подобных пар орбит в дальнейшем будем применять обозначения Q_{12}, Q_{02} и Q_{03} . Как отмечалось ранее, указанные пары орбит формируют максимально возможное количество неповторяющихся k -разрядных двоичных векторов, равное $2k + 2$. Поэтому даже однократное применение теста, генерирующего, например, пару орбит Q_{03} , обеспечивает всевозможные двоичные комбинации в любых $k < 4$ ячейках запоминающего устройства. Действительно, при $k = 3$ орбиты Q_{03} обеспечивают $2 \times k + 2 = 2 \times 3 + 2 = 8$ всевозможных 2^3 трехразрядных двоичных векторов. Не нарушая общности, детально исследуем пару орбит Q_{03} для произвольных k , $3 < k \leq N$, ячеек памяти с адресами $\beta, \gamma, \dots, \delta, \varepsilon, \eta$, где $N \geq \beta > \gamma > \dots > \delta > \varepsilon > \eta \geq 0$, запоминающего устройства (табл. 8).

Таблица 8

Пара орбит Q_{03} , состоящая из $2k+2$ двоичных векторов

Вектор	Индексы разрядов вектора					
	k	$k-1$...	3	2	1
P_0	a_β	a_γ	...	a_δ	a_ε	a_η
P_1	a_β	a_γ	...	a_δ	a_ε	a_η
P_2	a_β	a_γ	...	a_δ	$\overline{a_\varepsilon}$	a_η
P_3	a_β	a_γ	...	$\overline{a_\delta}$	a_ε	a_η
...
P_{k-1}	a_β	a_γ	...	a_δ	a_ε	a_η
P_k	$\overline{a_\beta}$	a_γ	...	a_δ	a_ε	a_η
P_{k+1}	$\overline{a_\beta}$	$\overline{a_\gamma}$...	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$
P_{k+2}	$\overline{a_\beta}$	a_γ	...	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$
P_{k+3}	a_β	a_γ	...	a_δ	a_ε	a_η
...
P_{2k-1}	$\overline{a_\beta}$	$\overline{a_\gamma}$...	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$
P_{2k}	$\overline{a_\beta}$	a_γ	...	$\overline{a_\delta}$	$\overline{a_\varepsilon}$	$\overline{a_\eta}$
P_{2k+1}	a_β	a_γ	...	a_δ	a_ε	a_η

Первоначально отметим особенность структуры Q_{03} , которая заключается в том, что вторая часть Q_{03} (Q_3), состоящая из двоичных наборов $P_{k+1}, P_{k+2}, P_{k+3}, \dots, P_{2k}, P_{2k+1}$, равняется инверсии ее первой части (Q_0), включающей $P_0, P_1, P_2, \dots, P_{k-1}, P_k$. Действительно, $P_{k+1} = \overline{P_0}$ и $P_{k+1+j} = \overline{P_{k+1-j}}$, $j = 1, 2, \dots, k$. Вторая особенность Q_{03} обусловлена инверсным соотношением двух частей Q_0 и Q_3 пары орбит Q_{03} . Она состоит в том, что вес $w(\overline{P_i})$ инверсного k -разрядного вектора $\overline{P_i}$ определяется весом $w(P_i)$ произвольного исходного двоичного вектора P_i как $w(\overline{P_i}) = k - w(P_i)$ и, наоборот, как $w(P_i) = k - w(\overline{P_i})$, где вес $w(P_i)$ двоичного вектора P_i определяется количеством его единичных компонент. Из табл. 8 видно, что $w(P_{k+1}) = w(\overline{P_0}) = k - w(P_0)$, а $w(P_{k+1+j}) = w(\overline{P_{k+1-j}}) = k - w(P_{k+1-j})$.

Для нулевого начального состояния $P_0 = 0\ 0\ 0 \dots 0$ двоичные k -разрядные векторы пары орбит Q_{03} имеют веса $w(P_0) = 0$, $w(P_1) = w(P_2) = \dots = w(P_k) = 1$, $w(P_{k+1}) = k$ и $w(P_{k+2}) = w(P_{k+3}) = \dots = w(P_{2k+1}) = k - 1$. Примеры пар орбит Q_{03} для $k = 5$ и различных начальных состояний P_0 приведены в табл. 9.

Таблица 9

Восемь пар орбит для $k = 5$

Вектор	$Q_{03}(00000)$	$Q_{03}(00010)$	$Q_{03}(01010)$	$Q_{03}(11010)$	$Q_{03}(11110)$
P_0	00000	00010	01010	11010	11110
P_1	00001	00011	01011	11011	11111
P_2	00010	00000	01000	11000	11100
P_3	00100	00110	01110	11110	11010
P_4	01000	01010	00010	10010	10110
P_5	10000	10010	11010	01010	01110
P_6	11111	11101	10101	00101	00001
P_7	01111	01101	00101	10101	10001
P_8	10111	10101	11101	01101	01001
P_9	11011	11001	10001	00001	00101
P_{10}	11101	11111	10111	00111	00011
P_{11}	11110	11100	10100	00100	00000

Анализ пар орбит Q_{03} , представленных в табл. 8 и 9, показывает, что независимо от начального состояния P_0 для любого $k < 4$ в произвольных k ячейках запоминающего устройства однократные тесты типа March A, генерирующие Q_{03} , обеспечивают всевозможные k -разрядные двоичные векторы. Для $k \geq 4$ необходимо повторное применение указанных тестов. Первоначально рассмотрим двукратное применение подобных тестов для $k > 4$.

Учитывая то, что новая пара орбит Q_{03} формируется с равной вероятностью для всех возможных 2^k начальных состояний орбит P_0 , докажем следующее утверждение.

Утверждение 5. Для заданного k , $4 < k < N$, и постоянного порядка адресов существует $2^k - k^2 - k - 2$ пар орбит Q_{03} , формируемых тестами типа March A с другими начальными состояниями P_0 и состоящих из двоичных векторов, которые не входят в исходную пару орбит Q_{03} , а также $k^2 + k$ пар орбит Q_{03} , включающих по четыре вектора, которые входят в две исходные орбиты Q_0 и Q_3 .

Доказательство. Для заданного значения $k > 4$ существует 2^k пар орбит Q_{03} , которые представлены в табл. 8 и 9 и определяются начальным состоянием P_0 . Не нарушая общности рассуждений, предположим, что $P_0 = 0\ 0\ 0 \dots 0$ и, соответственно, векторы, входящие в Q_{03} , имеют значения весов $w(P_i) \in \{0, 1, k - 1, k\}$. Необходимо отметить, что Q_{03} включает всевозможные k -разрядные векторы с указанными весами 0, 1, $k - 1$ и k , т. е. по одному вектору с весами 0 и k и по k векторов с весами 1 и $k - 1$.

Новые начальные состояния $P_0 \neq 0\ 0\ 0 \dots 0$ в зависимости от количества единичных значений в векторе P_0 будут определять инверсию соответствующих разрядов пары орбит Q_{03} (см. табл. 9). Рассмотрим случай нового начального состояния P_0 , полученного в результате применения операции инверсии только одного бита первоначального состояния P_0 исходной

пары орбит Q_{03} . В этом случае новая пара орбит будет содержать четыре k -разрядных вектора, ранее использованных в исходной паре орбит Q_{03} , а именно векторы $P_0 = 0\ 0\ 0\ \dots\ 0$ и $P_{k+1} = 1\ 1\ 1\ \dots\ 1$, полученные в результате инвертирования одного бита в соответствующем векторе с весом 1 и его инверсии с весом $k - 1$. Кроме того, результат инвертирования одного бита в векторах $P_0 = 0\ 0\ 0\ \dots\ 0$ и $P_{k+1} = 1\ 1\ 1\ \dots\ 1$ приведет к повторению в новой орбите одного из векторов с весом 1 и весом $k - 1$ исходной пары орбит. Пример получения новой пары орбит $Q_{03}(00010)$ для $k = 5$ как результат отрицания второго бита начального состояния приведен в табл. 9. В этой же таблице дан пример получения $Q_{03}(01010)$ как результат применения двух отрицаний для формирования нового начального состояния P_0 . Отметим, что использование двух отрицаний также приводит к формированию новой орбиты, в которой всегда будут повторяться четыре вектора, используемых в исходной орбите. Только два вектора из полного множества векторов веса 1 в результате применения двух отрицаний для двух произвольных разрядов будут преобразованы в векторы с весом 1, а остальные векторы будут иметь вес 3. Аналогичное утверждение справедливо и для векторов с весом $k - 1$. Только два из них в результате инвертирования двух бит будут иметь вес $k - 1$, а остальные преобразуются в векторы с весом $k - 3$.

Учитывая инверсную структуру ($Q_0 = \overline{Q_3}$) пары орбит Q_{03} , применение $k - 1$ отрицания для получения новой пары орбит эквивалентно применению одного отрицания, а использование $k - 2$ отрицаний повторяет процедуру применения двух отрицаний. Во всех четырех случаях новые начальные состояния, полученные в результате применения 1, 2, $k - 2$ и $k - 1$ отрицаний, приводят к получению новой орбиты, содержащей четыре k -разрядных вектора, ранее использованных в исходной паре орбит Q_{03} . Тогда общее количество новых пар орбит Q_{03}^n , включающих в себя четыре вектора из Q_{03} , для $k > 4$ определяется выражением

$$M_1 = 2 \binom{k}{1} + 2 \binom{k}{2} = 2k + 2 \frac{k(k-1)}{2} = k^2 + k. \quad (7)$$

Для случая, когда количество отрицаний равняется трем либо больше трех, но меньше $k - 2$, все двоичные векторы пары орбит Q_{03} преобразуются в векторы, веса которых будут отличаться от $w(P_i) \in \{0, 1, k - 1, k\}$. Действительно, применение трех отрицаний для $k > 4$ приводит к тому, что вектор с весом 0 преобразуется в вектор с весом $w = 3$, а вектор с весом 1 преобразуется в вектор, для которого $w = 2$ либо $w = 4$. Кроме того, вектор, имеющий вес k , преобразуется в вектор с весом $k - 3$, а векторы, вес которых равен $k - 1$, преобразуются в векторы с весами $k - 2$ либо $k - 4$. Таким образом, для $k \geq 5$ использование трех отрицаний при формировании нового начального состояния приводит к формированию пары орбит Q_{03} , состоящей из двоичных k -разрядных векторов, которые не входят в исходную пару орбит Q_{03} . Аналогичным образом можно показать, что применение большего числа отрицаний, не превышающего $k - 3$ для заданного k , также приводит к формированию пары орбит Q_{03} , двоичные векторы которых не пересекаются с векторами исходной орбиты. Число M_2 новых пар орбит, состоящих из двоичных векторов, которые не участвуют в исходной паре орбит O_{01} , определяется равенством

$$M_2 = 2^k - (k^2 + k) - 2, \quad (8)$$

что и требовалось доказать.

Утверждение 5 справедливо для произвольного значения $k > 4$. В случае $k = 4$ новые начальные состояния P_0 могут быть получены путем применения одного, двух и трех отрицаний. Для одного и трех отрицаний новые орбиты так же, как и для $k > 4$, будут содержать по четыре k -разрядных вектора, а для двух отрицаний – по восемь векторов, которые использовались в исходной паре орбит. Это объясняется тем, что только для $k = 4$ два отрицания соответствуют $k - 2$ отрицаниям.

Среднее количество новых k -разрядных двоичных векторов, формируемых парой орбит Q_{03} во время повторного применения теста типа March A, будет равняться

$[M_1 \times (2k-2) + M_2 \times (2k+2)] / 2^k$ для $k > 4$. При $k = 4$ это количество равняется $[8 \times 4 + 6 \times 8] / 2^4 = 5$. Среднее число $M_{ave}(Q_{03})$ двоичных векторов, формируемых в результате применения двукратного маршевого теста типа March A за счет пары орбит Q_{03} , определяется соотношением

$$M_{ave}(Q_{03}) = 2 \cdot (4+1) + 5 = 15 \quad \text{для } k = 4,$$

$$M_{ave}(Q_{03}) = 2 \cdot (k+1) + \frac{M_1 \cdot (2k-2) + M_2 \cdot (2k+2)}{2^k} = 4(k+1) - \frac{2k^2 + 3k + 2}{2^k} \quad \text{для } k > 4. \quad (9)$$

Отметим, что соотношения (9) позволяют получить лишь нижнюю оценку $M_{ave}(\text{March A})$, так как тесты типа March A формируют не только пары орбит Q_j , но и пары орбит O_i (см. рис. 1).

С целью оценки эффективности двукратных маршевых тестов типа March A были получены экспериментальные результаты, показывающие среднее количество двоичных векторов, формируемых в k произвольных ячейках памяти. Для каждого значения k в табл. 10 приведены усредненные значения, полученные в результате 100 000 экспериментов, заключающихся в случайном формировании начального состояния памяти. Для каждого эксперимента определялось количество k -разрядных векторов, формируемых в k ячейках памяти, адреса которых выбирались случайным образом.

Таблица 10

Экспериментальные значения $M_{ave}(Test)$

k	2	3	4	5	6	7	8	9	10
2^k	4	8	16	32	64	128	256	512	1024
$M_{ave}(\text{MATS}++)$	3,75	6,00	8,43	10,87	13,21	15,51	17,69	19,81	21,91
$M_{ave}(\text{March } C-)$	4	7,55	12,10	17,03	21,93	26,63	31,13	35,48	39,53
$M_{ave}(\text{March A})$	4	8	14,48	21,98	29,64	37,05	44,11	50,86	57,34

В табл. 10 представлены также аналогичные экспериментальные результаты для тестов типа MATS++ и March C–, которые целиком соответствуют теоретическим результатам, полученным согласно соотношениям (2) и (6). Из таблицы видно, что, например, двукратный тест MATS++ для случая $k = 4$ в среднем формирует 8,43 четырехразрядного вектора в произвольных четырех ячейках памяти. В то же время тест типа March C– формирует уже 12,1 вектора, а тесты из множества March A формируют 14,48 двоичных наборов. Очевидное преимущество тестов типа March A достигается за счет их большей сложности в сравнении со сложностью тестов других типов, которая в случае теста March A равняется $15N$.

Эффективность маршевых тестов типа MATS ++, March C– и March A. Эффективность однократного применения маршевых тестов типа MATS ++ ($6N$), March C– ($10N$), March A ($15N$) и March O ($12N$), для которых в скобках показана их сложность, оценим средним количеством $C_1(Test)$ двоичных векторов, формируемых в произвольных k из N ячеек памяти за N циклов обращения к памяти. Соответствующие характеристики как результат деления числа двоичных векторов на коэффициент сложности теста для указанных разновидностей тестов примут вид

$$C_1(\text{MATS } ++) = (k+1)/6, \quad C_1(\text{March } C-) = 2k/10,$$

$$C_1(\text{March A}) = 4(k-1)/15, \quad C_1(\text{March O}) = 4(k-1)/12. \quad (10)$$

Отметим, что все указанные разновидности тестов типа MATS ++, March C– и March A при однократном их применении гарантируют формирование в каждой ячейке памяти ($k = 1$) нулевого и единичного значения. Тесты типа March C– и March A, кроме того, обеспечивают всевозможные комбинации для двух произвольных ячеек памяти ($k = 2$), а тесты типа March A – и для трех ячеек ($k = 3$). Значения численных характеристик эффективности однократных тестов, представленных в формуле (10) для различных значений k , приведены в табл. 11.

Таблица 11

Эффективность $C_1(Test)$ однократного применения тестов

k	2	3	4	5	6	7	8	9	10
$C_1(MATS ++)$	0,50	0,67	0,83	1,00	1,17	1,33	1,50	1,67	1,83
$C_1(March C-)$	–	0,60	0,80	1,00	1,20	1,40	1,60	1,80	2,00
$C_1(March A)$	–	–	0,80	1,07	1,33	1,60	1,87	2,13	2,40
$C_1(March O)$	–	–	1,00	1,33	1,66	2,00	2,33	2,66	3,00

Анализ характеристики $C_1(Test)$ показывает наибольшую эффективность тестов типа March A и наименьшую эффективность тестов MATS ++, особенно с ростом значения величины k . В то же время для $k = 4$ наибольшую эффективность показывают тесты March O и MATS ++, а тесты March C- и March A имеют одинаковую эффективность. Для $k = 5$ эффективность однократных тестов MATS ++ и March C- одинакова, однако уступает эффективности тестов March A и March O.

При оценке эффективности двукратных тестов $C_2(Test)$ используем полученные аналитические выражения (2) и (6) для среднего количества $M_{ave}(Test)$ в случае тестов MATS ++ и March C-, а также экспериментальные оценки, приведенные в табл. 10, для тестов March A.

Таблица 12

Эффективность $C_2(Test)$ двукратного применения тестов

k	2	3	4	5	6	7	8	9	10
$C_2(MATS ++)$	0,31	0,50	0,70	0,91	1,10	1,29	1,47	1,65	1,82
$C_2(March C-)$	–	0,33	0,60	0,84	1,09	1,32	1,55	1,76	1,98
$C_2(March A)$	–	–	0,48	0,73	0,99	1,24	1,47	1,69	1,91
$C_2(March O)$	–	–	0,60	0,92	1,24	1,54	1,84	2,12	2,39

Характеристика $C_2(Test)$, так же как и $C_1(Test)$, является сравнительной оценкой эффективности маршевых тестов, которая показывает среднее количество k -разрядных двоичных векторов в произвольных k ячейках памяти, формируемых конкретным маршевым тестом за N циклов обращения к памяти. Значения данной характеристики показывают несомненную эффективность тестов типа March A, особенно для небольших значений k . Высокая эффективность теста March O объясняется его ориентацией на формирование всевозможных k -разрядных двоичных комбинаций в произвольных k ячейках запоминающего устройства, что является необходимым условием обнаружения сложных неисправностей памяти [4].

Как было показано ранее в работе [4], формирование всевозможных двоичных комбинаций в произвольных k ячейках памяти достигается многократным применением маршевых тестов, среди которых обоснованно выделяются тесты типа March C-. Однако результаты, представленные в предыдущих разделах, показывают большую эффективность тестов типа March A, которые позволяют генерировать большее число орбит и, соответственно, двоичных k -разрядных векторов. Для подтверждения данного утверждения в табл. 13 представлены экспериментальные результаты, показывающие сравнительную эффективность многократных тестов March C- и March A для случая $k = 8$.

Таблица 13

Количество восьмиразрядных векторов (%) по отношению к их максимальному числу 2^8 , %

Количество итераций теста	$M_{ave}(March C-)$			$M_{ave}(March A)$		
	min	max	ave	min	max	ave
5	15,63	31,25	27,58	22,27	44,92	37,55
10	28,13	59,38	47,53	39,45	73,83	60,92
20	46,88	88,28	72,40	57,81	95,70	84,58
30	57,81	99,22	85,36	68,36	100,00	93,79
40	66,41	100,00	92,21	76,17	100,00	97,47
50	72,66	100,00	95,77	82,42	100,00	98,91
60	78,13	100,00	97,65	87,50	100,00	99,50
70	81,25	100,00	98,67	90,63	100,00	99,77
80	83,59	100,00	99,22	92,19	100,00	99,89
90	85,16	100,00	99,54	95,31	100,00	99,94
100	85,16	100,00	99,71	95,31	100,00	99,97

Графическая иллюстрация результатов экспериментов (рис. 2) показывает несомненное преимущество тестов типа March A по сравнению с другими классическими маршевыми тестами для обеспечения всевозможных двоичных комбинаций в произвольных восьми ячейках запоминающего устройства.

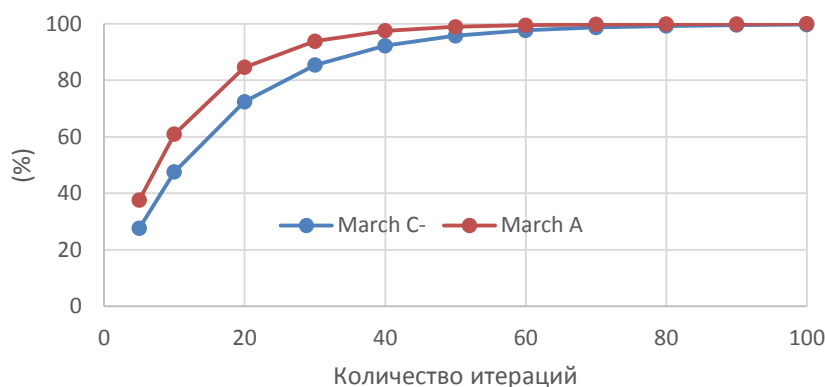


Рис. 2. Сравнительная эффективность тестов March C– и March A

На рис. 2 видно, что тест March A всегда превосходит тест March C–, так как для достижения той же полноты покрытия для теста March A требуется существенно меньше его итераций. Например, десятикратный тест March A обеспечивает в произвольных восьми ячейках памяти в среднем 60,92 % от 256 двоичных векторов, в то же время десятикратный тест March C– — только 47,53 % (см. табл. 13). Для произвольного количества k ячеек памяти высокая эффективность тестов типа March A подтверждается экспериментальными данными, представленными в табл. 14.

Таблица 14

Экспериментальные значения для $M_{ave}(\text{March A})$

k	3	4	5	6	7	8	9	10
2^k	8	16	32	64	128	256	512	1024
$M_{ave}(\text{March A})$	1	2,9	7,3	15,15	30,81	63,81	126,35	248
$M_{min}(\text{March A})$	1	2	4	6	12	28	60	131
$M_{max}(\text{March A})$	1	26	40	62	118	205	340	544
$M_{ave}(\text{March C-})$	2,33	4,42	9,34	19,15	40,55	82,62	169,32	339,48

Для сопоставительного анализа в последней строке табл. 14 представлены полученные ранее в работе [4] экспериментальные оценки $M_{ave}(\text{March C-})$, которые для всех k заметно уступают $M_{ave}(\text{March A})$. Например, для обеспечения всевозможных двоичных комбинаций путем многократного применения теста March A в произвольных 10 ячейках памяти в среднем необходимо применить данный тест 248 раз. Тот же результат может быть достигнут тестом March C–, средняя кратность которого принимает значение, равное 339,48.

Заключение. В работе проведен анализ маршевых тестов, применяемых для тестирования современных запоминающих устройств. Выделены три основных подмножества подобных тестов, а именно тесты типа MATS ++, March C– и March A. Исследована эффективность применения этих разновидностей тестов для реализации псевдоисчерпывающего тестирования запоминающих устройств. Получены аналитические соотношения, позволяющие оценить эффективность маршевых тестов для целей генерирования всевозможных k -разрядных двоичных комбинаций в k произвольных ячейках памяти. Полученные аналитические оценки эффективности реализации псевдоисчерпывающего тестирования памяти подтверждены экспериментальными результатами, показывающими преимущества тестов типа March A. При-

менение подобных тестов для целей реализации псевдоисчерпывающего тестирования требует существенно меньших временных затрат по сравнению с тестами типа MATS ++ и March C–.

Список использованных источников

1. Wang, L.-T. VLSI Test Principles and Architectures: Design for Testability / L.-T. Wang, C.-W. Wu, X. Wen. – Elsevier, 2006. – 808 p.
2. Ярмолик, В. Н. Контроль и диагностика вычислительных систем / В. Н. Ярмолик. – Минск : Бестпринт, 2019. – 387 с.
3. Иванюк, А. А. Проектирование встраиваемых цифровых устройств и систем / А. А. Иванюк. – Минск : Бестпринт, 2012. – 338 с.
4. Ярмолик, В. Н. Псевдоисчерпывающее тестирование запоминающих устройств на базе многократных маршевых тестов / В. Н. Ярмолик, И. Мрозек, В. А. Леванцевич // Информатика. – 2018. – № 1(15). – С. 110–121.
5. Sharma, A. K. Semiconductor Memories: Technology, Testing, and Reliability / A. K. Sharma. – London : John Wiley & Sons, 2002. – 480 p.
6. Niggemeyer, D. Integration of non-classical faults in standard march tests / D. Niggemeyer, M. Redeker, J. Otterstedt // Records of the IEEE Intern. Workshop on Memory Technology, Design and Testing. – San Jose, 1998. – P. 91–98.
7. Неразрушающее тестирование запоминающих устройств / В. Н. Ярмолик [и др.]. – Минск : Бестпринт, 2005. – 230 с.
8. Mrozek, I. Multi-run Memory Tests for Pattern Sensitive Faults / I. Mrozek. – Cham : Springer International Publishing AG, 2019. – 135 p.
9. Goor, A. J. Testing Semiconductor Memories, Theory and Practice / A. J. Goor. – Chichester : John Wiley & Sons, 1991. – 536 p.
10. Yarmolik, S. V. Address sequences and backgrounds with different hamming distance for multiple run march tests / S. V. Yarmolik // IEEE Intern. J. of Applied Mathematics and Computer Science. – 2008. – Vol. 18, no. 3. – P. 329–339.
11. Sokol, B. Address sequence for march tests to detect pattern sensitive faults / B. Sokol, S. V. Yarmolik // Proc. of 3rd IEEE Intern. Workshop on Electronic Design Test and Applications (DELTA'06). – Kuala Lumpur, Malaysia, 2006. – P. 354–357.
12. Sokol, B. Impact of the address changing on the detection of pattern sensitive faults / B. Sokol, I. Mrozek, V. N. Yarmolik // Information Processing and Security Systems. – London : Springer Science + Business Media, Inc., 2005. – P. 217–226.
13. Ярмолик, В. Н. Адресные последовательности для многократного тестирования ОЗУ / В. Н. Ярмолик, С. В. Ярмолик // Информатика. – 2014. – № 3(39). – С. 92–103.
14. Mrozek, I. Antirandom test vectors for BIST in Hardware / Software systems / I. Mrozek, V. N. Yarmolik // Fundamenta Informaticae. – 2012. – No. 119. – P. 1–23.
15. Mrozek, I. Iterative antirandom testing / I. Mrozek, V. N. Yarmolik // J. of Electronic Testing: Theory and Applications (JETTA). – 2012. – Vol. 9, no. 3. – P. 251–266.

References

1. Wang L.-T., Wu C.-W., Wen X. *VLSI Test Principles and Architectures: Design for Testability*. Elsevier, 2006, 808 p.
2. Yarmolik V. N. Kontrol' i diagnostika vuchislitel'nyh sistem. *Monitoring and Diagnostics of Computer Systems*. Minsk, Bestprint, 2019, 387 p. (in Russian).
3. Ivaniuk A. A. Proektirovanie vstraivaemuh cifrovuh ustroystv i sistem. *Designing Embedded Digital Devices and Systems*. Minsk, Bestprint, 2012, 338 p. (in Russian).
4. Yarmolik V. N., Mrozek I., Levancevich V. A. Pseudoisчерpuyayuschee testirovanie zapominayuschih ustroystv na baze mnogokratnych marshevyh testov [Pseudo-exhaustive storage testing based on multiple march tests]. *Informatika [Informatics]*, 2018, no. 1(15), pp. 110–121 (in Russian).
5. Sharma A. K. *Semiconductor Memories: Technology, Testing, and Reliability*. London, John Wiley & Sons, 2002, 480 p.
6. Niggemeyer D., Redeker M., Otterstedt J. Integration of non-classical faults in standard march tests. *Records of the IEEE International Workshop on Memory Technology, Design and Testing*, San Jose, 1998, pp. 91–98.

7. Yarmolik V. N., Murashko I. A., Kummert A., Ivaniuk A. A. Nerazrushayushee testirovanie zapominayuschih ustroystv. *Non-Destructive Storage Testing*. Minsk, Bestprint, 2005, 230 p. (in Russian).
8. Mrozek I. *Multi-run Memory Tests for Pattern Sensitive Faults*. Cham, Springer International Publishing AG, 2019, 135 p.
9. Goor A. J. *Testing Semiconductor Memories, Theory and Practice*. Chichester, John Wiley & Sons, 1991, 536 p.
10. Yarmolik S. V. Address sequences and backgrounds with different hamming distance for multiple run march tests. *IEEE International Journal of Applied Mathematics and Computer Science*, 2008, vol. 18, no. 3, pp. 329–339.
11. Sokol B., Yarmolik S. V. Address sequence for march tests to detect pattern sensitive faults. *Proceedings of 3rd IEEE International Workshop on Electronic Design Test and Applications (DELTA'06)*. Kuala Lumpur, Malaysia, 2006, pp. 354–357.
12. Sokol B., Mrozek I., Yarmolik V. N. Impact of the address changing on the detection of pattern sensitive faults. *Information Processing and Security Systems*. London, Springer Science + Business Media, Inc., 2005, pp. 217–226.
13. Yarmolik V. N., Yarmolik S. V. Adresnye posledovatel'nosti dlya mnogokratnogo testirovaniya OZU [Address sequences for repeated testing of RAM]. *Informatika [Informatics]*, 2014, no. 3(39), pp. 92–103 (in Russian).
14. Mrozek I., Yarmolik V. N. Antirandom test vectors for BIST in Hardware / Software systems. *Fundamenta Informaticae*, 2012, no. 119, pp. 1–23.
15. Mrozek I., Yarmolik V. N. Iterative antirandom testing. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 2012, vol. 9, no. 3, pp. 251–266.

Информация об авторах

Ярмолик Вячеслав Николаевич, доктор технических наук, профессор, Белорусский государственный университет радиоэлектроники и информатики, Минск, Беларусь.
E-mail: yarmolik10ru@yahoo.com

Мрозек Ирэнэуш, доктор, адъюнкт, Белостокский технический университет, Белосток, Польша.
E-mail: i.mrozek@pb.edu.pl

Ярмолик Светлана Вячеславовна, кандидат технических наук, Белорусский государственный университет радиоэлектроники и информатики, Минск, Беларусь.
E-mail: syarmolik@gmail.com

Information about the authors

Vyacheslav N. Yarmolik, Dr. Sci. (Eng.), Professor, Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus.
E-mail: yarmolik10ru@yahoo.com

Ireneusz Mrozek, Dr., Lecture, Bialystok University of Technology, Bialystok, Poland.
E-mail: i.mrozek@pb.edu.pl

Svetlana V. Yarmolik, Cand. Sci. (Eng.), Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus.
E-mail: syarmolik@gmail.com

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

УДК 004.052.32+681.518.5
<https://doi.org/10.37661/1816-0301-2020-17-2-71-85>

Поступила в редакцию 30.10.2019
Received 30.10.2019

Принята к публикации 03.02.2020
Accepted 03.02.2020

Схема коррекции сигналов для комбинационных устройств автоматики на основе логического дополнения с контролем вычислений по паритету

В. В. Сапожников¹, Вл. В. Сапожников¹, Д. В. Ефанов²✉

¹Петербургский государственный университет путей сообщения
Императора Александра I, Санкт-Петербург, Россия

²ООО «ЛокоТех-Сигнал», Российский университет транспорта, Москва, Россия

✉E-mail: TrES-4b@yandex.ru

Аннотация. Предложена более простая структура системы с коррекцией ошибок в вычислениях, чем известные структуры, основанные на дублировании и троировании блоков с мажоритарным принципом выбора значений сигналов. Новую отказоустойчивую структуру целесообразно использовать для устройств автоматики с комбинационной логикой. При синтезе отказоустойчивой структуры применяется метод паритета для установления факта возникновения неисправности в контролируемом объекте и метод логического дополнения для определения неправильно вычисленных выходных функций и формирования сигналов для их коррекции. Приведена структурная схема системы с коррекцией ошибок и дано ее описание. Представлен алгоритм синтеза контрольного оборудования с минимизацией сложности его технической реализации. Результаты экспериментов с контрольными комбинационными схемами подтверждают высокую эффективность применения предложенной структуры системы с коррекцией ошибок.

Ключевые слова: комбинационные устройства автоматики, системы с обнаружением неисправностей, системы с коррекцией ошибок в вычислениях, отказоустойчивые системы, дублирование, троирование, мажоритарный принцип контроля, метод логического дополнения, код паритета

Для цитирования. Сапожников, В. В. Схема коррекции сигналов для комбинационных устройств автоматики на основе логического дополнения с контролем вычислений по паритету / В. В. Сапожников, Вл. В. Сапожников, Д. В. Ефанов // Информатика. – 2020. – Т. 17, № 2. – С. 71–85. <https://doi.org/10.37661/1816-0301-2020-17-2-71-85>

Signal correction for combinational automation devices on the basis of Boolean complement with control of calculations by parity

Valery V. Sapozhnikov¹, Vladimir V. Sapozhnikov¹, Dmitry V. Efanov²✉

¹Emperor Alexander I St. Petersburg State Transport University, Saint Petersburg, Russia

²"LocoTech-Signal" LLC, Russian University of Transport, Moscow, Russia

✉E-mail: TrES-4b@yandex.ru

Abstract. Simpler than known structure of the system with error correction in calculations is proposed based on duplication and triplication of blocks with majority principle of choosing the values of signals. It is advisable to use the new fault-tolerant structure for automation devices with combinational logic. In fault-tolerant structure synthesis, the parity method is used to establish the fact of a fault in the main logic unit and the logical complement method is used to determine incorrectly calculated output functions and to generate signals for their correction. The method also allows to adjust the values of incorrectly calculated functions. Structural diagram

and description of error correction system are given. The synthesis algorithm of control equipment is described with minimization of the technical implementation complexity. The experiment results with control combinational circuits are given, confirming the high efficiency of proposed system structure with error correction.

Keywords: combinational automation devices, fault-detection systems, systems with error correction in calculations, fault-tolerant systems, failure redundancy, triplication, majority control principle, Boolean complement method, parity code

For citation. Sapozhnikov V. V., Sapozhnikov V. V., Efanov D. V. Signal correction for combinational automation devices on the basis of Boolean complement with control of calculations by parity. *Informatics*, 2020, vol. 17, no. 2, pp. 71–85 (in Russian). <https://doi.org/10.37661/1816-0301-2020-17-2-71-85>

Введение. При построении систем автоматического управления в виде дискретных устройств для повышения их надежности используются структуры, в которых либо обеспечивается контроль правильности функционирования системы (правильности выполнения заданных алгоритмов и вычислений), либо достигается свойство нечувствительности к отказам внутренних элементов системы, при возникновении которых сохраняется правильность функционирования [1–5]. Системы с контролем в процессе функционирования называют также системами с обнаружением неисправностей, а системы, нечувствительные к отказам, – отказоустойчивыми системами, или системами с коррекцией неверно вычисленных сигналов. Для систем первого и второго вида указывается множество элементов, относительно отказов которых выполняются обозначенные свойства систем.

При построении обоих видов систем широко применяются методы избыточного кодирования информации, передаваемой как между блоками внутри систем управления, так и между системой управления и внешними объектами управления.

Примером системы с обнаружением неисправностей может служить часто применяемая структура дублирования одинаковых комплектов устройства [6, 7]. При дублировании используются свойства кодов с повторением, а именно возможность обнаружения ими любых видов искажений в информационных разрядах кодовых слов [8]. Структурная схема системы дублирования включает в себя блоки основной и контрольной логики, а также самопроверяемый компаратор [9]. Блок контрольной логики – это, как правило, копия блока основной логики либо устройство, синтезированное по иным принципам, но реализующим те же функции. Компаратор же является устройством сравнения значений на одноименных выходах блоков основной и контрольной логики. В практических реализациях структурная избыточность системы дублирования оценивается величиной, более чем втрое превышающей сложность реализации блока основной логики [10].

Отказоустойчивые системы имеют гораздо большую сложность технической реализации, чем системы с обнаружением неисправностей. На практике используются структуры с тройной избыточностью, в которых устанавливаются три копии исходного объекта, а выходные сигналы формируются при помощи мажоритарных схем коррекции [11–17]. Применяются также системы, основанные на принципе четырехкратной избыточности: в таких системах один комплект, реализованный по принципу дублирования, находится под нагрузкой, а второй находится в резерве, на который переключается нагрузка при зафиксированном отказе первого [18]. Известны и другие модификации отказоустойчивых структур дискретных систем.

В настоящей статье предлагается новая отказоустойчивая структура дискретных устройств, позволяющая получать системы с уменьшенной структурной избыточностью по сравнению с традиционными структурами, основанными на многократном резервировании исходных объектов.

Известные отказоустойчивые структуры. Распространенной является отказоустойчивая структура системы с коррекцией ошибок по мажоритарному принципу (*TMR*-системы, или majority correction system). Для достижения свойства отказоустойчивости относительно неисправностей в данной структуре используются три идентичных устройства $F(x)$, вычисляющих на своих выходах одинаковые логические сигналы на одинаковых входных воздействиях. Для сравнения вычисленных значений сигналов на одноименных выходах различных устройств используются модули сравнения, функционирующие по принципу голосования «два из трех» (мажоритарные элементы). Часто мажоритарные элементы реализуются в виде устройств, со-

храняющих работоспособность при одиночном отказе любого внутреннего компонента [2, 18]. Тройная избыточность структуры необходима для наделения ее свойством нечувствительности к проявлениям неисправностей в исходном устройстве в виде искажений сигналов на его выходах. Использование такой структуры на практике приводит к существенному увеличению сложности технической реализации конечного устройства. Кроме того, приведенная структура имеет недостаток, связанный с невозможностью идентификации неверно функционирующего устройства без введения дополнительного контроля каждого блока.

При разработке структуры с мажоритарным принципом коррекции могут иметь место несколько вариантов:

- исходное устройство $F(x)$ дается разработчику в неизменяемом виде, а его задача состоит в разработке отказоустойчивой системы, при этом разработчик может пользоваться точными копиями данного устройства;
- имеется возможность оптимизации двух дополнительных копий устройства $F(x)$ в процессе разработки отказоустойчивой системы;
- возможна оптимизация самого устройства $F(x)$ и его копий.

Очевидно, что из трех вариантов наименьшую структурную избыточность будет давать последний. Для типовых отказоустойчивых структур, таким образом, имеет смысл введение понятия *структуры с минимальной избыточностью*, которая будет для выбранного варианта внесения избыточности давать наименьшую сложность технической реализации.

Для создания отказоустойчивой структуры может быть применена структурная схема, включающая в себя вместо трех копий исходного объекта только две (рис. 1). При этом одна копия (или же оба комплекта устройств $F(x)$) снабжается схемой контроля по какому-либо диагностическому признаку. Известны варианты реализации подобных схем при нахождении обоих комплектов под нагрузкой и нахождении одного комплекта под нагрузкой, а второго – в горячем резерве [2]. Для случая отказоустойчивых комбинационных схем целесообразно использовать вариант с постоянно нагруженными комплектами $F(x)$ и контролем вычислений одного из них.

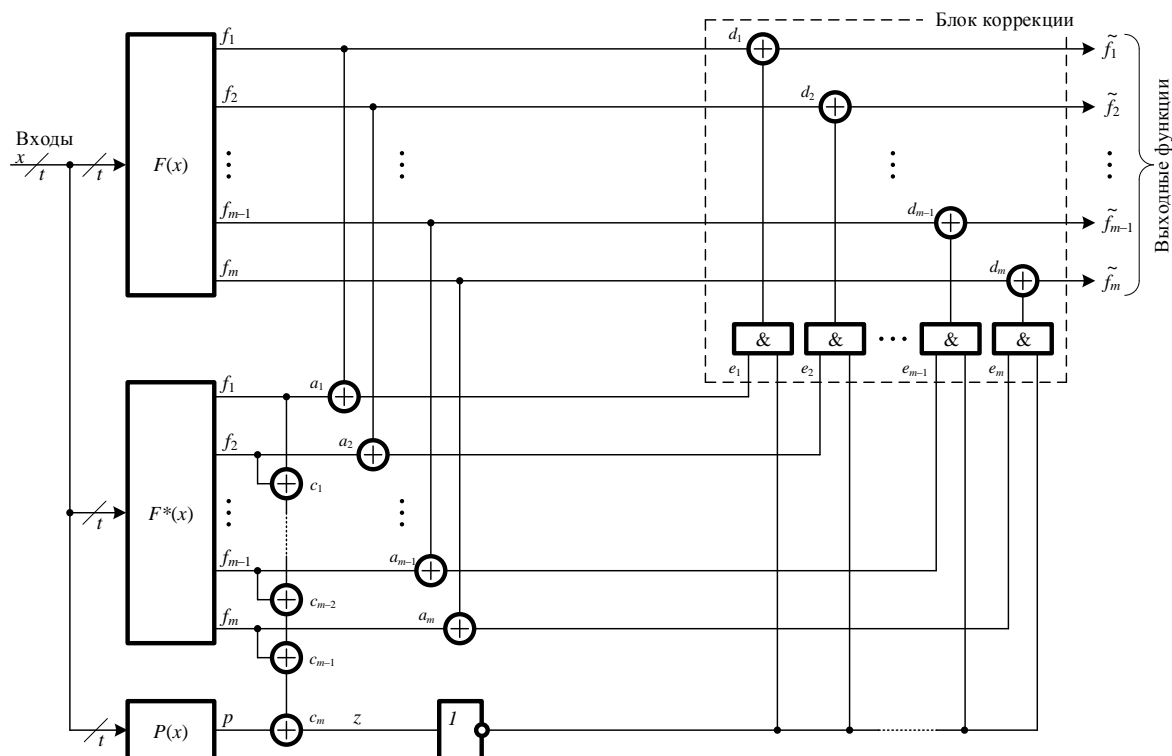


Рис. 1. Отказоустойчивая структура «дублирование с коррекцией»

Например, на рис. 1 показана структура системы дублирования с коррекцией и контролем вычислений по паритету (duplication correction system with parity checking). Более кратко будем называть ее структурой «дублирование с коррекцией».

Отметим, что если отходить от мажоритарного принципа реализации отказоустойчивой структуры, то требуется решить следующие три задачи. Первая и основная задача состоит в необходимости восстановления правильных значений искаженных выходных функций основного устройства $F(x)$. В структуру «дублирование с коррекцией» для решения этой задачи вводится блок коррекции (БК), включающий в себя m элементов сложения по модулю два (XOR) $d_1, d_2, \dots, d_{m-1}, d_m$, выходы которых соединяются с выходами всего устройства, а входы – с выходами основного блока $F(x)$ и выходами элементов логического умножения (AND). Последние непосредственно формируют логические сигналы, восстанавливающие правильные значения искаженных значений сигналов на выходах основного блока. Ошибки в вычислениях, возникающие в результате неисправностей элементов БК, не исправляются, это касается и выходных элементов в мажоритарных структурах.

Вторая задача заключается в необходимости определения тех выходов основного блока $F(x)$, на которых в результате неисправностей элементов в его структуре реализуются ошибочные функции. С этой целью в структуру (см. рис. 1) вводится второй блок $F^*(x)$ как копия основного блока и элементы сложения по модулю два $a_1, a_2, \dots, a_{m-1}, a_m$, на которых сравниваются сигналы с одноименных выходов блоков $F(x)$ и $F^*(x)$. Активные сигналы, возникающие на выходах этих элементов, свидетельствуют о наличии неисправностей в одном из блоков.

Коррекция логических сигналов в БК должна осуществляться только в случае, когда неисправен основной блок $F(x)$. Фиксация факта отказа этого блока является третьей задачей, которая в структуре на рис. 1 решается путем контроля исправности блока $F^*(x)$ при помощи схемы паритета (блок $P(x)$ и элементы сложения по модулю два $c_1, c_2, \dots, c_{m-1}, c_m$).

Выходная функция f_i корректируется только в том случае, когда функция $z = 0$ (блок $F^*(x)$ исправен) и на выходе элемента сложения по модулю два a_i присутствует сигнал 1.

Представленное на рис. 1 обустройство схемы коррекции во многих случаях позволяет синтезировать более простые отказоустойчивые дискретные устройства, чем по структурной схеме с мажоритарной коррекцией. Следует отметить, что при построении структуры, приведенной на рис. 1, как и в схеме с тройной избыточностью, возможны варианты с оптимизацией второго устройства $F(x)$, а также обоих устройств, что еще больше уменьшит сложность технической реализации структуры «дублирование с коррекцией». В последнем случае будет получена структура дублирования с минимальной избыточностью.

Поставим следующую задачу: покажем принципиальную возможность построения такой отказоустойчивой структуры, которая в своем составе имеет только одно исходное устройство $F(x)$, что позволяет для определенных схем уменьшить сложность технической реализации по сравнению со структурами дублирования и с мажоритарной коррекцией сигналов, имеющих минимальную избыточность.

Новая отказоустойчивая структура. Существенным недостатком мажоритарной отказоустойчивой структуры и структуры «дублирование с коррекцией» является то, что при их построении возможно получение только одного варианта структуры (если не учитывать варианты, связанные с оптимизацией всех или только некоторых блоков структур). При этом никак не учитываются конкретные особенности блока $F(x)$, связанные с определенными характеристиками булевых функций $f_1, f_2, \dots, f_{m-1}, f_m$. Как показано в работах [19–21] на примере построения схем с контролем в процессе функционирования, возможность учета указанных особенностей появляется при использовании в процессе построения дискретных устройств принципа логического дополнения. В этом случае при применении одной и той же типовой структуры можно реализовать несколько вариантов с различной сложностью, из которых может быть выбран вариант с минимальной сложностью.

На рис. 2 показана структура системы коррекции на основе логического дополнения с контролем вычислений по паритету (parity correction system based on Boolean complement). Основной блок $F(x)$, вычисляющий систему булевых функций $f_1, f_2, \dots, f_{m-1}, f_m$ от t входных переменных, является исходным комбинационным устройством автоматики. Именно для него

строится отказоустойчивая структура. Восстановление искаженных значений функций $f_1, f_2, \dots, f_{m-1}, f_m$, как и в структуре на рис. 1, осуществляется при помощи аналогичного блока коррекции. Задача установления факта возникновения неисправности в блоке $F(x)$ решается при помощи схемы паритета (блок $P(x)$ и элементы сложения по модулю два $c_1, c_2, \dots, c_{m-1}, c_m$), которая в отличие от схемы на рис. 1 контролирует значения сигналов непосредственно на выходах основного блока.

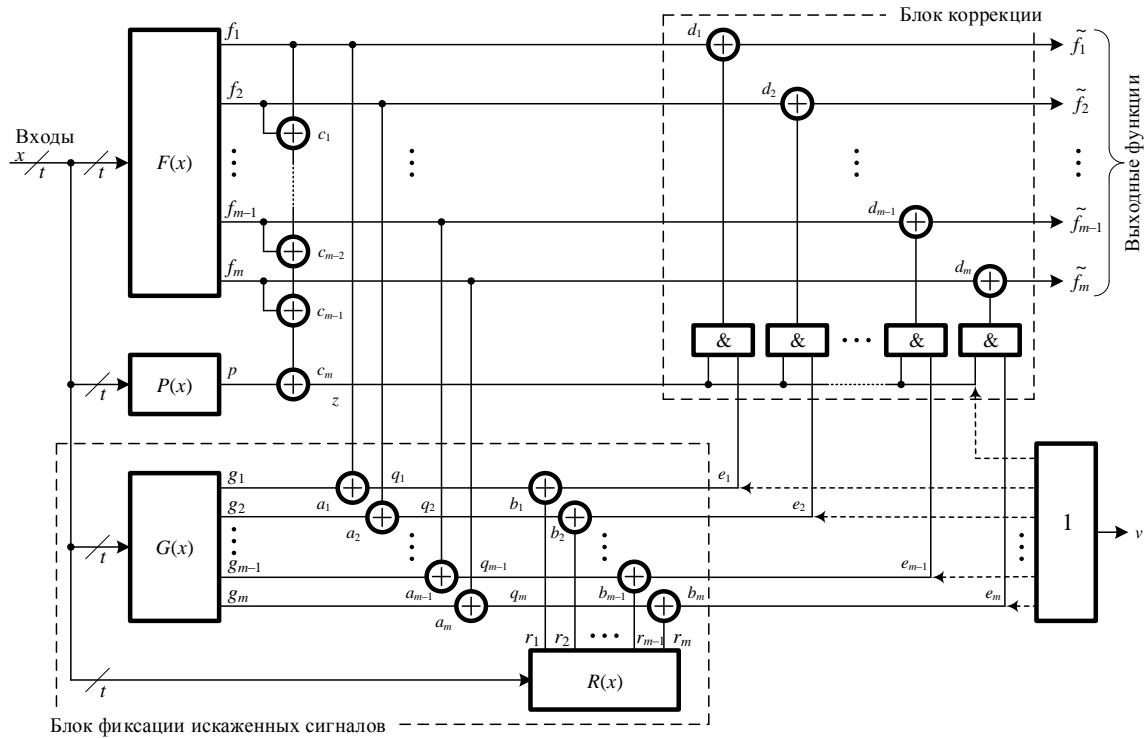


Рис. 2. Отказоустойчивая структура на основе логического дополнения с контролем вычислений по паритету

Принципиально по-новому решается задача выявления тех выходов блока $F(x)$, на которых имеет место искажение правильного сигнала. Для этого в структуре используется блок фиксации искаженных сигналов (ФИС). В него входит блок логического дополнения $G(x)$, на выходах которого вычисляются специальные функции дополнения $g_1, g_2, \dots, g_{m-1}, g_m$. Последние сравниваются с одноименными функциями выходов блока $F(x)$ при помощи элементов сложения по модулю два $a_1, a_2, \dots, a_{m-1}, a_m$, на выходах которых вычисляются функции сравнения $q_1, q_2, \dots, q_{m-1}, q_m$. Блок ФИС включает в себя также блок коррекции функций сравнения $R(x)$, вычисляющий функции $r_1, r_2, \dots, r_{m-1}, r_m$, которые выбираются с таким расчетом, чтобы на выходе блока ФИС в исправной структуре были реализованы функции

$$\begin{aligned} e_1 &= f_1 \oplus g_1 \oplus r_1 = 0, \\ e_2 &= f_2 \oplus g_2 \oplus r_2 = 0, \\ &\dots \\ e_m &= f_m \oplus g_m \oplus r_m = 0. \end{aligned} \quad (1)$$

Если в результате неисправности в блоке $F(x)$ искажается значение функции f_i , то, как следует из (1), $e_i = 1$ и активизируются вход и выход соответствующего элемента логического умножения (при этом на выходе схемы паритета сигнал $z = 1$). Осуществляется исправление неправильного значения функции f_i .

При синтезе блока ФИС необходимо определить функции $g_1, g_2, \dots, g_{m-1}, g_m$ и $r_1, r_2, \dots, r_{m-1}, r_m$. Из выражения (1) следует, что $g_i = f_i \oplus r_i$. Каждая функция r_i может быть выбрана произвольным образом независимо от того, какими являются функции f_i и g_i . От выбранного варианта функции r_i зависит сложность реализации функции g_i , при этом никакой зависимости между функциями g_i и r_i с различными индексами i не существует. В связи с этим задача получения структуры блока ФИС с наименьшей сложностью может быть решена следующим образом. Для каждой пары функций (g_i, r_i) путем подбора функции r_i находится вариант с наименьшей сложностью обеих функций. Затем блоки $G(x)$ и $R(x)$ описываются наборами соответствующих функций и осуществляется оптимизация структуры каждого блока в отдельности (возможна и совместная оптимизация обоих блоков).

Очевидно, что при реализации отказоустойчивой структуры на рис. 2 для любого блока $F(x)$ существует много вариантов набора функций $\{r_1, r_2, \dots, r_{m-1}, r_m\}$, при которых свойство отказоустойчивости достигается. Поэтому возможен выбор варианта, при котором обеспечивается минимальная сложность блока ФИС.

Необходимо отметить особый вариант набора функций $\{r_1, r_2, \dots, r_{m-1}, r_m\}$, в котором каждая функция приравнивается к какой-либо (любой) входной переменной $x_1, x_2, \dots, x_{t-1}, x_t$. В этом случае реализация блока $R(x)$ не требует аппаратных затрат.

Неисправности элементов схемы паритета, при которых устанавливается значение функции $z = 1$, не приводят к искажению правильных значений функций $f_1, f_2, \dots, f_{m-1}, f_m$, так как в это время блок ФИС исправен и все функции $e_i = 0$. И наоборот, неисправности элементов блока ФИС, которые вызывают появление значений функций $e_1 = 1$, также не приводят к изменению указанных функций, так как в этом случае сигнал $z = 0$. Элементы БК следует реализовать в виде высоконадежных устройств с резервированием внутренних компонентов. Отметим, что структура на рис. 2 может быть дополнена элементом, на выходе которого реализуется функция $v = z \vee e_1 \vee e_2 \vee \dots \vee e_m$ и формируется сигнал о наличии неисправности в схеме паритета и блока ФИС при исправном основном блоке $F(x)$.

Корректирующая способность структуры на рис. 1 ограничивается тем, что при помощи паритета обнаруживаются только те неисправности в блоке $F(x)$, которые искажают значения сигналов на нечетном числе выходов блока. Для улучшения корректирующей способности необходимо повышать обнаруживающую способность схемы паритета либо за счет изменения внутренней структуры блока $F(x)$ [22] (если это возможно), либо путем организации контроля для отдельных групп независимых выходов блока [23].

Построение блока фиксации искаженных сигналов. При построении блока ФИС необходимо провести взаимосвязанный синтез блоков $G(x)$ и $R(x)$. Как отмечено выше, возможно большое число вариантов проведения этого синтеза, каждый из которых связан с выбором конкретных функций в наборе $\{r_1, r_2, \dots, r_{m-1}, r_m\}$. Далее рассмотрим случай, в котором этот выбор ограничен возможностью приравнивания каждой функции r_i какой-либо входной переменной. При этом задача построения блока ФИС сводится к задаче синтеза блока логического дополнения $G(x)$.

Алгоритм синтеза блока $G(x)$:

1. Для каждой функции f_i ($i \in \{1, 2, \dots, m\}$) осуществляем поиск переменной x_j , которая позволяет получить наиболее простое с точки зрения технической реализации выражение для функции $g_i = f_i \oplus x_j$.
2. Полагаем $i = 1$.
3. Полагаем $j = 1$.
4. Выбираем функцию f_i и вычисляем для нее вариант функции дополнения g_i с коррекцией по переменной x_j на всех входных наборах $\langle x_1 x_{t-1} \dots x_2 x_1 \rangle$.
5. Минимизируем полученную функцию g_i и записываем ее в память.
6. Полагаем $j := j + 1$.
7. Проверяем условие достижения конечной входной переменной $x_t : j > t$? Если нет, то возвращаемся к п. 4 данного алгоритма, в противном случае переходим к п. 8.
8. Полагаем $i := i + 1$.

9. Проверяем условие достижения конечной функции $f_m : i > m$? Если нет, то возвращаемся к п. 3 данного алгоритма, в противном случае переходим к п. 10.

10. Для каждой функции g_i определяем ту переменную $(x_j)_{\min}^i$, коррекция по которой дает наименьшую сложность реализации в выбранном элементном базисе.

11. Синтезируем блок реализации системы функций $g_i (i \in \{1, 2, \dots, m\})$.

12. Структурную схему, полученную на предыдущем этапе, оптимизируем, что и дает конечную структурную схему блока $G(x)$.

Рассмотрим работу данного алгоритма на примере синтеза отказоустойчивой структуры для произвольного комбинационного устройства автоматики.

Пример синтеза отказоустойчивой структуры. Построим схему контроля и коррекции для комбинационного устройства, реализующего систему, которая состоит из шести булевых функций от четырех переменных. Работа данного устройства описывается таблицей истинности (табл. 1).

Будем рассматривать задачу синтеза устройства $F(x)$ на логических элементах. При этом на структуру устройства никакие ограничения не накладываются.

Неисправности во внутренней структуре устройства $F(x)$ вызывают возникновение ошибок на внутренних линиях схемы, а затем часть из них транслируется на выходы самого устройства $f_1 \div f_6$, вызывая в информационном векторе $\langle f_1 f_2 f_3 f_4 f_5 f_6 \rangle$ ошибку определенной кратности. Синтезируемая схема контроля и коррекции предназначена для маскировки этой ошибки. Отсюда следует, что на множество неисправностей, допустимых в устройстве $F(x)$, никаких ограничений не накладывается. Таким образом, подсхема контроля при неисправности, которая проявилась хотя бы на одном из выходов устройства $F(x)$, должна зафиксировать наличие ошибки, а подсхема коррекции – использовать сигналы от подсхемы контроля и исправить неверные сигналы на выходах $f_1 \div f_6$.

Пользуясь предложенным алгоритмом, вычислим функции дополнения. Эти функции представлены в табл. 1, там же приведена функция паритета.

Таблица 1

Описание функций, реализуемых на выходах схемы коррекции

$x_1 x_2 x_3 x_4$	$f_1 f_2 f_3 f_4 f_5 f_6$	p	$g_1^j = f_1 \oplus x_j$	$g_2^j = f_2 \oplus x_j$	$g_3^j = f_3 \oplus x_j$	$g_4^j = f_4 \oplus x_j$	$g_5^j = f_5 \oplus x_j$	$g_6^j = f_6 \oplus x_j$
			$g_1^1 g_1^2 g_1^3 g_1^4$	$g_2^1 g_2^2 g_2^3 g_2^4$	$g_3^1 g_3^2 g_3^3 g_3^4$	$g_4^1 g_4^2 g_4^3 g_4^4$	$g_5^1 g_5^2 g_5^3 g_5^4$	$g_6^1 g_6^2 g_6^3 g_6^4$
0000	010010	0	0000	1111	0000	0000	1111	0000
0001	101101	0	1110	0001	1110	1110	0001	1110
0010	101001	1	1101	0010	1101	0010	0010	1101
0011	010110	1	0011	1100	0011	1100	1100	0011
0100	101001	1	1011	0100	1011	0100	0100	1011
0101	000010	1	0101	0101	0101	0101	1010	0101
0110	010011	1	0110	1001	0110	0110	1001	1001
0111	001100	0	0111	0111	1000	1000	0111	0111
1000	101000	0	0111	1000	0111	1000	1000	1000
1001	111110	1	0110	0110	0110	0110	0110	1001
1010	110111	1	0101	0101	1010	0101	0101	0101
1011	000010	1	1011	1011	1011	1011	0100	1011
1100	010011	1	1100	0011	1100	1100	0011	0011
1101	100100	0	0010	1101	1101	0010	1101	1101
1110	100100	0	0001	1110	1110	0001	1110	1110
1111	011010	1	1111	0000	0000	1111	0000	1111

Для оценки сложности реализации функций, вычисляемых на выходах блоков схемы коррекции, будем использовать показатель L – число внутренних входов логических элементов схемы без учета инверсий на входах [24].

Определим сложность реализации устройства $F(x)$, функции которого, полученные путем минимизации каждой из функций, имеют вид

$$\begin{aligned} f_1 &= \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3} \vee \overline{x_1 x_3 x_4} \vee \overline{x_1 x_3 x_4} \vee \overline{x_2 x_3 x_4} \vee \overline{x_2 x_3 x_4}; \\ f_2 &= \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4}; \\ f_3 &= \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3} \vee \overline{x_2 x_3 x_4}; \\ f_4 &= \overline{x_1 x_2 x_4} \vee \overline{x_1 x_3 x_4} \vee \overline{x_1 x_3 x_4} \vee \overline{x_1 x_3 x_4}; \\ f_5 &= \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_4} \vee \overline{x_1 x_2 x_3} \vee \overline{x_1 x_3 x_4}; \\ f_6 &= \overline{x_1 x_2 x_3 x_4} \vee \overline{x_2 x_3 x_4} \vee \overline{x_1 x_3 x_4} \vee \overline{x_2 x_3 x_4}. \end{aligned}$$

Сложность реализации устройства $F(x)$ определяется выражением $L_{F(x)} = 152$.

Из табл. 1 следует, что функция, реализуемая на выходе блока $P(x)$, имеет вид

$$p = \overline{x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3} \vee \overline{x_1 x_2 x_4} \vee \overline{x_1 x_2 x_4} \vee \overline{x_1 x_3 x_4} \vee \overline{x_2 x_3}.$$

Сложность реализации такой функции $L_{P(x)} = 22$.

Определим функции логического дополнения. Вычислим все варианты функций $g_1^j = f_1 \oplus x_j$, $g_2^j = f_2 \oplus x_j$, $g_3^j = f_3 \oplus x_j$, $g_4^j = f_4 \oplus x_j$, $g_5^j = f_5 \oplus x_j$ и $g_6^j = f_6 \oplus x_j$. Среди каждой группы одноименных функций выберем функцию с наименьшей сложностью реализации (см. табл. 1).

Приведем наиболее простые функции для реализации на выходе блока $G(x)$:

$$\begin{aligned} g_1^4 &= \overline{x_1 x_2 x_3} \vee \overline{x_1 x_2 x_4} \vee \overline{x_1 x_3} \vee \overline{x_2 x_3} \vee \overline{x_3 x_4}; \\ g_2^1 &= \overline{x_2 x_3 x_4} \vee \overline{x_2 x_3 x_4} \vee \overline{x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4}; \\ g_3^3 &= \overline{x_1 x_2 x_4} \vee \overline{x_2 x_3 x_4} \vee \overline{x_2 x_4} \vee \overline{x_1 x_2}; \\ g_4^4 &= \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_3}; \\ g_5^1 &= \overline{x_2 x_3 x_4} \vee \overline{x_2 x_3 x_4} \vee \overline{x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4}; \\ g_6^2 &= \overline{x_1 x_3 x_4} \vee \overline{x_2 x_3 x_4} \vee \overline{x_1 x_3 x_4} \vee \overline{x_2 x_4}. \end{aligned}$$

Использование полученных выше функций позволяет синтезировать блок дополнения со сложностью реализации $L_{G(x)} = 88$.

Учитывая сложности реализации элементов сложения по модулю два и логического умножения, входящих в схему паритета и блока ФИС ($L_{XOR} = 6$ и $L_{AND} = 2$), получим значение сложности технической реализации отказоустойчивой структуры на основе логического дополнения:

$$L_{PC} = L_{F(x)} + L_{G(x)} + L_{P(x)} + 4 \cdot 6L_{XOR} + 6L_{AND} = 152 + 88 + 22 + 24 \cdot 6 + 6 \cdot 2 = 418.$$

Сложность полученной структуры в 3,01 раза превышает сложность реализации самого блока $F(x)$.

Сравним полученный результат со сложностью традиционной отказоустойчивой структуры «дублирование с коррекцией» и структурой на основе использования мажоритарного принципа «два из трех».

Сложность структуры «дублирование с коррекцией» определяется величиной

$$L_D = 2L_{F(x)} + L_{P(x)} + 3 \cdot 6 \cdot L_{XOR} + 6L_{AND} + L_{INV} = 2 \cdot 152 + 22 + 18 \cdot 6 + 6 \cdot 2 + 1 = 447.$$

Найдем отношение величин L_{PC} и L_D :

$$\delta = \frac{L_{PC}}{L_D} \cdot 100 \% = \frac{418}{447} \cdot 100 \% \approx 93,51 \%$$

Из данного результата следует, что система, полученная на основе логического дополнения, для рассматриваемого комбинационного устройства является более простой, чем структура «дублирование с коррекцией».

Сравним величину L_{PC} с величиной сложности реализации схемы коррекции по принципу «два из трех» L_{MC} (в ней использованы мажоритарные элементы, сложность которых оценивается величиной $L_{\geq 2} = 9$):

$$L_{MC} = 3L_{F(x)} + 5L_{\geq 2} = 3 \cdot 152 + 6 \cdot 9 = 510.$$

Найдем отношение величин L_{PC} и L_{MC} :

$$\mu = \frac{L_{PC}}{L_{MC}} \cdot 100 \% = \frac{418}{510} \cdot 100 \% \approx 81,96 \%$$

Полученный результат свидетельствует о том, что выигрыш в сложности технической реализации структуры коррекции на основе логического дополнения перед структурой мажоритарной коррекции составляет практически 20 %.

Следует отметить, что структурные схемы блоков $F(x)$ и $G(x)$ могут быть оптимизированы за счет использования общих подсхем для реализации различных функций. Анализ уравнений, описывающих блоки $F(x)$ и $G(x)$, позволяет получить следующие более сжатые их описания (выделены только совпадающие конъюнкции):

– для функций блока $F(x)$

$$f_1 = (\overline{x_1 x_2 x_3 x_4} = a) \vee x_1 \overline{x_2 x_3} \vee x_1 \overline{x_3 x_4} \vee (x_1 x_3 \overline{x_4} = b) \vee \overline{x_2 x_3 x_4} \vee (\overline{x_2 x_3 x_4} = c);$$

$$f_2 = (\overline{x_1 x_2 x_3 x_4} = d) \vee (\overline{x_1 x_2 x_3 x_4} = e) \vee (\overline{x_1 x_2 x_3 x_4} = g) \vee \\ \vee (x_1 x_2 \overline{x_3 x_4} = h) \vee x_1 \overline{x_2 x_3 x_4} \vee x_1 x_2 \overline{x_3 x_4} \vee x_1 \overline{x_2 x_3 x_4};$$

$$f_3 = a \vee x_1 \overline{x_2 x_3 x_4} \vee x_2 \overline{x_3 x_4} \vee x_1 \overline{x_2 x_3} \vee x_2 \overline{x_3 x_4};$$

$$f_4 = \overline{x_1 x_2 x_4} \vee \overline{x_1 x_3 x_4} \vee \overline{x_1 x_3 x_4} \vee b;$$

$$f_5 = d \vee e \vee \overline{x_1 x_2 x_3 x_4} \vee g \vee h \vee x_1 \overline{x_2 x_4} \vee x_1 \overline{x_2 x_3} \vee x_1 x_3 x_4;$$

$$f_6 = \overline{x_1 x_2 x_3 x_4} \vee \overline{x_2 x_3 x_4} \vee \overline{x_1 x_3 x_4} \vee c;$$

– для функций блока $G(x)$

$$g_1^4 = \overline{x_1 x_2 x_3} \vee x_1 \overline{x_2 x_4} \vee (x_1 x_3 = i) \vee \overline{x_2 x_3} \vee x_3 x_4;$$

$$g_2^1 = (\overline{x_2 x_3 x_4} = j) \vee \overline{x_2 x_3 x_4} \vee (x_2 x_3 \overline{x_4} = k) \vee x_1 \overline{x_2 x_3 x_4};$$

$$g_3^3 = x_1 x_2 \bar{x}_4 \vee k \vee \bar{x}_2 x_4 \vee x_1 \bar{x}_2;$$

$$g_4^4 = \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \vee i;$$

$$g_5^1 = j \vee x_2 \bar{x}_3 x_4 \vee k \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4;$$

$$g_6^2 = \bar{x}_1 \bar{x}_3 \bar{x}_4 \vee \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_3 \bar{x}_4 \vee x_2 x_4.$$

Сложности реализации блоков $F(x)$ и $G(x)$, таким образом, уменьшаются до величин $L_{F(x)} = 133$ и $L_{G(x)} = 83$. Это вызывает и уменьшение сложности полных схем коррекции до величин $L_{PC} = 394$ и $L_D = 409$. По-прежнему наилучший результат дает предложенная авторами схема коррекции. В табл. 2 приведены рассчитанные показатели сложности технической реализации отказоустойчивых структур всех трех видов и реализаций. Видно, что применение предложенной отказоустойчивой структуры позволяет получать более простые устройства, чем при использовании известных структур.

Таблица 2

Показатели сложности технической реализации отказоустойчивых структур при различных способах реализации блоков основной и дополнительной логики

Способ реализации	L_{MC}	L_D	L_{PC}	$\mu, \%$	$\delta, \%$
Без оптимизации структур компонентов	510	447	418	81,96	93,51
С использованием оригинального блока $F(x)$ и оптимизированными остальными блоками	472	428	413	87,5	96,5
С оптимизацией структур всех компонентов	453	409	394	86,98	96,33

Процедура поиска наилучшего решения по оптимизации структур блоков $F(x)$ и $G(x)$ достаточно затруднена при работе вручную. Поэтому целесообразно прибегнуть к использованию известных средств оптимизации, например SIS Berkeley (University of California) [25, 26].

Эксперименты с большим числом комбинационных устройств показывают, что предложенная отказоустойчивая структура обладает преимуществом перед известными. В ходе исследований особенностей предложенной отказоустойчивой структуры были проведены экспериментальные исследования с комбинационными устройствами из специального набора контрольных схем (URL: <http://ddd.fit.cvut.cz/prj/Benchmarks/>). Эксперименты заключались в построении для контрольных схем предложенной отказоустойчивой структуры, а также в анализе корректирующей способности. Для каждой контрольной схемы строилась предложенная структура, а также вычислялась занимаемая площадь (area overhead), оцениваемая по числу букв в формульной записи (literals). В ходе оценки корректирующей способности схем коррекции моделировалась работа контрольных комбинационных схем на 1000 псевдослучайных входных комбинациях при внесении последовательно всех одиночных константных неисправностей на выходах их внутренних логических элементов. Вероятность коррекции ошибки p_c определялась по формуле

$$p_c = \frac{\sum_{i=1}^N \frac{n_i}{1000}}{N}, \quad (2)$$

где N – число всех неисправностей для данной структуры (в эксперименте рассматривалась модель одиночных константных неисправностей);

n_i – число входных векторов, на которых присутствует правильный выходной вектор.

В ходе эксперимента для новой структуры получены следующие показатели: средняя площадь блока вычисления функции паритета составила 28,6 % от средней площади исходной схемы, блока контрольной логики – 38,9 %, дополнительных элементов сложения по модулю два и логического умножения, входящих в схему паритета и блока ФИС, – 81,5 %. В среднем вся схема коррекции занимает площадь, равную 249 % от площади исходной схемы. Полученные результаты соизмеримы с результатами оценки сложности технической реализации систем с обнаружением неисправностей на основе дублирования (в среднем 300–400 % от площади исходного устройства) [18].

Для 15 из 22 схем получены значения величин $\delta < 100$ %, причем для ряда схем этот показатель не превышает 80 %. Гораздо больший эффект получен при сравнении со схемой коррекции на основе мажоритарного принципа: для 17 из 22 схем получены значения $\mu < 100$ %, а для некоторых из них – менее 70 %. Средние значения относительных показателей $\delta = 92,281$ % и $\mu = 86,916$ %, что говорит о преимуществе предложенной структуры перед известными.

Вероятности коррекции ошибок для структур каждого из рассматриваемых комбинационных устройств были получены большими, чем 0,9, а для ряда схем – свыше $p_c = 0,98$. Среднее значение анализируемой величины $p_c = 0,95421$, что свидетельствует о высокой эффективности самих схем коррекции для представленных контрольных комбинационных устройств. Следует также отметить, что при проведении экспериментов специально не выделялись группы независимых выходов и не модифицировались структуры исходных комбинационных устройств. Таким образом, тестирование контрольных схем осуществлялось без привязки к особенностям их структуры и, соответственно, не были исключены ошибки с четными кратностями (напомним, что они не обнаруживаются при контроле по паритету).

Заключение. Способность рассмотренной структуры исправлять последствия отказов внутренних элементов характеризуется следующими положениями:

1. В структуре выделены отдельные блоки: $F(x)$, схема паритета, ФИС и блок коррекции.
2. Элементы блока коррекции должны быть построены в виде резервированных структур, устойчивых к одиночным отказам внутренних деталей.
3. Допускается возникновение неисправностей только в одном из блоков ($F(x)$, схема паритета и ФИС).
4. В основном блоке $F(x)$ допускается любая комбинация одиночных неисправностей, возникновение которых приводит к искажениям нечетного числа функций из множества $\{f_1, f_2, \dots, f_{m-1}, f_m\}$.
5. Внутри схемы коррекции и блока ФИС допускается любая комбинация одиночных неисправностей.
6. Искажения значений выходных сигналов устройства вызывают отказы только элементов блока $F(x)$. Отказы элементов схемы паритета и блока ФИС не приводят к искажению правильных выходных сигналов при исправном блоке $F(x)$.

В сравнении с традиционными структурами «дублирование с коррекцией» и на основе мажоритарного принципа коррекции предложенная структура оказывается более предпочтительной по показателям сложности технической реализации. При этом в общем при использовании новой схемы коррекции сигналов можно добиться эффекта сокращения сложности технической реализации до 25–30 % по сравнению со структурой «дублирование с коррекцией» и до 40–45 % по сравнению со структурой на основе мажоритарного принципа коррекции.

Кроме того, эксперимент показал, что использование предложенной в статье схемы коррекции сигналов позволяет достигать высоких показателей корректирующей способности даже без анализа их структурных схем, модификации схем в схемы с независимыми выходами и без выделения групп независимых выходов. В эксперименте для 22 комбинационных устройств получено среднее значение вероятности коррекции $p_c = 0,95421$.

Следует отметить две особенности предложенной структуры. Первая заключается в том, что по сравнению со структурой «дублирование с коррекцией» в новой структуре за счет двух каскадов сумматоров по модулю два $a_1, a_2, \dots, a_{m-1}, a_m$ и $b_1, b_2, \dots, b_{m-1}, b_m$ увеличено время прохождения сигналов и, таким образом, снижено быстродействие структуры. Тем не менее задержка незначительна и путь прохождения сигнала увеличивается незначительно. Увеличение времени срабатывания можно считать некоторой ценой за снижение аппаратных затрат на реализа-

цию предложенной структуры. Вторая особенность связана с процедурами поиска значений функций логического дополнения $g_1, g_2, \dots, g_{m-1}, g_m$ и выбора функций коррекции $r_1, r_2, \dots, r_{m-1}, r_m$. Требуется анализ $Q = n_x n_f$ (n_x и n_f – число входов и число выходов блока $F(x)$ соответственно) различных способов вычисления функций логического дополнения на 2^{n_x} входных комбинациях. Число Q существенно зависит от числа входов и выходов устройства $F(x)$, что определяет ограничения на применение предложенного авторами алгоритма получения функций дополнения и выбора функций коррекции. Для современной вычислительной техники полный алгоритм вычисления реализуется за разумное время при числе входных переменных, не превышающих значений 30–35 (это связано и с возможностями автоматической минимизации функций алгебры логики). При большем числе входов можно использовать упрощенный алгоритм вычисления функций дополнения, рассмотренный в настоящей статье. Как показали описанные в статье эксперименты с контрольными комбинационными схемами, это позволяет получать достаточно хорошие результаты.

Использование логического дополнения при синтезе отказоустойчивых структур открывает новые пути реализации схем коррекции с уменьшенной избыточностью.

Список использованных источников

1. Согомонян, Е. С. Самопроверяемые устройства и отказоустойчивые системы / Е. С. Согомонян, Е. В. Слабаков. – М. : Радио и связь, 1989. – 208 с.
2. Методы построения безопасных микроэлектронных систем железнодорожной автоматики / В. В. Сапожников [и др.] ; под ред. Вл. В. Сапожникова. – М. : Транспорт, 1995. – 272 с.
3. Харченко, В. С. Модели и свойства многоальтернативных отказоустойчивых систем / В. С. Харченко // Автоматика и телемеханика. – 1992. – № 12. – С. 140–147.
4. Скляр, В. В. Отказоустойчивые компьютерные системы управления с версионно-пороговой адаптацией: способы адаптации, оценка надежности, выбор архитектур / В. В. Скляр, В. С. Харченко // Автоматика и телемеханика. – 2002. – № 6. – С. 131–145.
5. Гаврилов, М. А. Надежность дискретных систем / М. А. Гаврилов, В. М. Остиану, А. И. Потехин // Итоги науки и техники. Сер. «Теория вероятностей. Математическая статистика. Теоретическая кибернетика». – 1969, 1970. – С. 7–104.
6. Goessel, M. Error Detection Circuits / M. Goessel, S. Graf. – London : McGraw-Hill, 1994. – 261 p.
7. Fujiwara, E. Code Design for Dependable Systems: Theory and Practical Applications / E. Fujiwara. – John Wiley & Sons, 2006. – 720 p.
8. Кодирование информации (двоичные коды) / Н. Т. Березюк [и др.] ; под ред. Н. Т. Березюка. – Харьков : Вища школа, 1978. – 252 с.
9. Nikolos, D. Self-testing embedded two-rail checkers / D. Nikolos // J. of Electronic Testing: Theory and Applications. – 1998. – Vol. 12, no. 1–2. – P. 69–79.
10. Новые структуры систем функционального контроля логических схем / В. В. Сапожников [и др.] // Автоматика и телемеханика. – 2017. – № 2. – С. 127–143.
11. Hamamatsu, M. Finding the optimal configuration of a cascading TMR system / M. Hamamatsu, T. Tsuchiya, T. Kikuno // 14th IEEE Pacific Rim Intern. Symp. on Dependable Computing, 15–17 Dec. 2008, Taipei, Taiwan. – Taipei, 2008. – P. 329–350. <https://doi.org/10.1109/PRDC.2008.12>
12. Matsumoto, K. Evaluating the fault tolerance of stateful TMR / K. Matsumoto, M. Uehara, H. Mori // 13th Intern. Conf. on Network-Based Information Systems, 14–16 Sept. 2010, Takayama, Japan. – Takayama, 2010. – P. 332–336. <https://doi.org/10.1109/NBiS.2010.86>
13. Останин, С. А. Синтез отказоустойчивых комбинационных схем / С. А. Останин // Прикладная дискретная математика (приложение № 1). – 2009. – № 1. – С. 71–72.
14. Методы синтеза сбоеустойчивых комбинационных КМОП-схем, обеспечивающих автоматическое исправление ошибок / А. Л. Стемпковский [и др.] // Известия ЮФУ. Технические науки. – 2017. – № 7(192). – С. 197–210. <https://doi.org/10.23683/2311-3103-2017-7-197-210>
15. Enhanced duplication method with TMR-like masking abilities / J. Borecký [et al.] // Euromicro Conf. on Digital System Design (DSD), 31 Aug. – 2 Sept. 2016, Limassol, Cyprus. – Limassol, 2016. – P. 690–693. <https://doi.org/10.1109/DSD.2016.91>
16. Krcma, M. Triple modular redundancy used in field programmable neural networks / M. Krcma, Z. Kotasek, J. Lojda // Proc. of 15th IEEE East-West Design & Test Symposium (EWDTS 2017), 29 Sept. – 2 Oct. 2017, Novi Sad, Serbia. – Novi Sad, 2017. – P. 372–377. <https://doi.org/10.1109/EWDTS.2017.8110128>

17. Sogomonyan, E. S. Self-Correction Fault-Tolerant Systems / E. S. Sogomonyan. – Preprint, Oct. 2018. – 30 p.
18. Сапожников, В. В. Основы теории надежности и технической диагностики / В. В. Сапожников, Вл. В. Сапожников, Д. В. Ефанов. – СПб. : Лань, 2019. – 588 с.
19. Организация функционального контроля комбинационных схем методом логического дополнения / В. В. Сапожников // Электронное моделирование. – 2002. – Т. 24, № 6. – С. 51–66.
20. Логическое дополнение – новый метод контроля комбинационных схем / М. Гессель [и др.] // Автоматика и телемеханика. – 2003. – № 1. – С. 167–176.
21. New Methods of Concurrent Checking / M. Goessel [et al.]. – 1st ed. – Dordrecht : Springer Science + Business Media B.V., 2008. – 184 p.
22. Self-checking combinational circuits with unidirectionally independent outputs / A. Morosow [et al.] // VLSI Design. – 1998. – Vol. 5, iss. 4. – P. 333–345. <https://doi.org/10.1155/1998/20389>
23. Ефанов, Д. В. Синтез самопроверяемых комбинационных устройств на основе выделения специальных групп выходов / Д. В. Ефанов, В. В. Сапожников, Вл. В. Сапожников // Автоматика и телемеханика. – 2018. – № 9. – С. 79–94.
24. Сапожников, В. В. Самопроверяемые дискретные устройства / В. В. Сапожников, Вл. В. Сапожников. – СПб. : Энергоатомиздат, 1992. – 224 с.
25. SIS: a system for sequential circuit synthesis / E. M. Sentovich [et al.]. – University of California, Berkeley, 1992. – 45 p.
26. Sequential circuit design using synthesis and optimization / E. M. Sentovich [et al.] // Proc. IEEE Intern. Conf. on Computer Design: VLSI in Computers & Processors, 11–14 Oct. 1992, Cambridge, MA, USA. – Cambridge, 1992. – P. 328–333. <https://doi.org/10.1109/ICCD.1992.276282>

References

1. Sogomonyan E. S., Slabakov E. V. Samoproveryaemye ustrojstva i otkazoustojchivye sistemy. *Self-checking Devices and Fault-tolerance Systems*. Moscow, Radio i svyaz', 1989, 208 p. (in Russian).
2. Sapozhnikov V. V., Sapozhnikov Vl. V., Hristov H. A., Gavzov D. V., Sapozhnikov Vl. V. (ed.). Metody postroeniya bezopasnyh mikroelektronnyh sistem zheleznodorozhnoj avtomatiki. *Methods of Synthesis Safety Microelectronic Railway Automation Systems*. Moscow, Transport, 1995, 272 p. (in Russian).
3. Kharchenko V. S. Modeli i svoystva mnogoal'ternativnyh otkazoustojchivyh system [Models and properties of multialternative fault-tolerant systems]. *Avtomatika i telemekhanika [Automation and Remote Control]*, 1992, no. 12, pp. 140–147 (in Russian).
4. Sklyar V. V., Kharchenko V. S. Otkazoustojchivye komp'yuternye sistemy upravleniya s versionno-porogovoj adaptaciej: sposoby adaptacii, ocenka nadezhnosti, vybor arhitektur [Fault-tolerant computer-aided control systems with multiversion-threshold adaptation: adaptation methods, reliability estimation, and choice of an architecture]. *Avtomatika i telemekhanika [Automation and Remote Control]*, 2002, no. 6, pp. 131–145 (in Russian).
5. Gavrilov M. A., Ostianu V. M., Potekhin A. I. Nadezhnost' diskretnykh sistem [Reliability of discrete systems]. Itogi nauki i tekhniki. Seriya "Teoriya veroyatnostej. Matematicheskaya statistika. Teoreticheskaya kibernetika" [Results of Science and Technology. Series "Probability Theory. Math statistics. Theoretical Cybernetics"], 1969, 1970, pp. 7–104 (in Russian).
6. Goessel M., Graf S. *Error Detection Circuits*. London, McGraw-Hill, 1994, 261 p.
7. Fujiwara E. *Code Design for Dependable Systems: Theory and Practical Applications*. John Wiley & Sons, 2006, 720 p.
8. Berezyuk N. T., Andrushchenko A. G., Moshchickij S. S., Glushkov V. I., Benesha M. M., Gavrilov V. A., N. T. Berezyuk (ed.). Kodirovanie informacii (dvoichnye kody). *Information Coding (Binary Codes)*. Kharkiv, Vishcha shkola, 1978, 252 p. (in Russian).
9. Nikolos D. Self-testing embedded two-rail checkers. *Journal of Electronic Testing: Theory and Applications*, 1998, vol. 12, no. 1–2, pp. 69–79.
10. Sapozhnikov V. V., Sapozhnikov Vl. V., Efanov D. V., Dmitriev V. V. Novye struktury sistem funkcional'nogo kontrolya logicheskikh skhem [New structures of the concurrent error detection systems for logic circuits]. *Avtomatika i telemekhanika [Automation and Remote Control]*, 2017, no. 2, pp. 127–143 (in Russian).
11. Hamamatsu M., Tsuchiya T., Kikuno T. Finding the optimal configuration of a cascading TMR system. *14th IEEE Pacific Rim International Symposium on Dependable Computing, 15–17 December 2008, Taipei, Taiwan*. Taipei, 2008, pp. 329–350. <https://doi.org/10.1109/PRDC.2008.12>

12. Matsumoto K., Uehara M., Mori H. Evaluating the fault tolerance of stateful TMR. *13th International Conference on Network-Based Information Systems, 14–16 September 2010, Takayama, Japan*. Takayama, 2010, pp. 332–336. <https://doi.org/10.1109/NBiS.2010.86>
13. Ostanin S. A. Sintez otkazoustojchivyyh kombinacionnyh skhem [Synthesis of fail-safe combinational circuits]. *Prikladnaya diskretnaya matematika (prilozhenie № 1) [Applied Discrete Mathematics (Appendix No. 1)]*, 2009, no. 1, pp. 71–72 (in Russian).
14. Stempkovskij A. L., Tel'puhov D. V., Zhukova T. D., Gurov S. I., Solov'ev R. A. Metody sinteza sboeustojchivyyh kombinacionnyh KMOP-skhem, obespechivayushchih avtomaticheskoe ispravlenie oshibok [Synthesis methods for fault-tolerant combinational CMOS circuits providing automatic error correction]. *Izvestiya Juzhnogo federal'nogo universiteta. Tekhnicheskie nauki [Proceedings of the Southern Federal University. Technical Science]*, 2017, no. 7(192). pp. 197–210. <https://doi.org/10.23683/2311-3103-2017-7-197-210> (in Russian).
15. Borecký J., Kohlík M., Vít P., Kubátová H. Enhanced duplication method with TMR-like masking abilities. *Euromicro Conference on Digital System Design (DSD), 31 August – 2 September 2016, Limassol, Cyprus*. Limassol, 2016, pp. 690–693. <https://doi.org/10.1109/DSD.2016.91>
16. Krcma M., Kotasek Z., Lojda J. Triple modular redundancy used in field programmable neural networks. *Proceedings of 15th IEEE East-West Design & Test Symposium (EWDTS 2017), 29 September – 2 October 2017, Novi Sad, Serbia*. Novi Sad, 2017, pp. 372–377. <https://doi.org/10.1109/EWDTS.2017.8110128>
17. Sogomonyan E. S. *Self-Correction Fault-Tolerant Systems*. Preprint, October 2018, 30 p.
18. Sapozhnikov V. V., Sapozhnikov V. I., Efanov D. V. Osnovy teorii nadezhnosti i tekhnicheskoy diagnostiki. *Fundamentals of the Theory of Reliability and Technical Diagnostics*. Saint Petersburg, Lan, 2019, 588 p. (in Russian).
19. Sapozhnikov V. V., Sapozhnikov V. I., Dmitriev A. V., Morozov A. V., Gessel' M. Organizaciya funkcional'nogo kontrolya kombinacionnyh skhem metodom logicheskogo dopolneniya [The organization of functional control of combinational circuits by the method of complement]. *Electronnoe modelirovanie [Electronic Modeling]*, 2002, vol. 24, no. 6, pp. 51–66 (in Russian).
20. Gessel' M., Morozov A. V., Sapozhnikov V. V., Sapozhnikov V. I. Logicheskoe dopolnenie – novyj metod kontrolya kombinacionnyh skhem [Logic complement, a new method of checking the combinational circuits]. *Avtomatika i telemekhanika [Automation and Remote Control]*, 2003, no. 1, pp. 167–176 (in Russian).
21. Gessel' M., Ocheretny V., Sogomonyan E., Marienfeld D. *New Methods of Concurrent Checking*. Dordrecht, Springer Science + Business Media B.V., 2008, 184 p.
22. Morosow A., Saposhnikov V. V., Saposhnikov V. I., Goessel M. Self-checking combinational circuits with unidirectionally independent outputs. *VLSI Design*, 1998, vol. 5, iss. 4, pp. 333–345. <https://doi.org/10.1155/1998/20389>
23. Efanov D. V., Sapozhnikov V. V., Sapozhnikov V. I. Sintez samoproveryaemyh kombinacionnyh ustrojstv na osnove vydeleniya special'nyh grupp vyhodov [Synthesis of self-checking combination devices based on allocating special groups of outputs]. *Avtomatika i telemekhanika [Automation and Remote Control]*, 2018, no. 9, pp. 79–94 (in Russian).
24. Sapozhnikov V. V., Sapozhnikov V. I. Samoproveryaemye diskretnye ustrojstva. *Self-Checking Discrete Devices*. Saint Petersburg, Energoatomizdat, 1992, 224 p. (in Russian).
25. Sentovich E. M., Singh K. J., Lavagno L., Moon C., Murgai R., ..., Sangiovanni-Vincentelli A. *SIS: A System for Sequential Circuit Synthesis*. University of California, Berkeley, 1992, 45 p.
26. Sentovich E. M., Singh K. J., Moon C., Savoj H., Brayton R. K., Sangiovanni-Vincentelli A. Sequential circuit design using synthesis and optimization. *Proceedings IEEE International Conference on Computer Design: VLSI in Computers & Processors, 11–14 October 1992, Cambridge, MA, USA*. Cambridge, 1992, pp. 328–333. <https://doi.org/10.1109/ICCD.1992.276282>

Информация об авторах

Сапожников Валерий Владимирович, доктор технических наук, профессор, профессор кафедры «Автоматика и телемеханика на железных дорогах», Петербургский государственный университет путей сообщения Императора Александра I, Санкт-Петербург, Россия.
E-mail: port.at.pgups@gmail.com

Information about the authors

Valery V. Sapozhnikov, Dr. Sci. (Eng.), Professor, Professor of "Automation and Remote Control on Railways" Department, Emperor Alexander I St. Petersburg State Transport University, Saint Petersburg, Russia.
E-mail: port.at.pgups@gmail.com

Сапожников Владимир Владимирович, доктор технических наук, профессор, профессор кафедры «Автоматика и телемеханика на железных дорогах», Петербургский государственный университет путей сообщения Императора Александра I, Санкт-Петербург, Россия.
E-mail: at.pgups@gmail.com

Ефанов Дмитрий Викторович, доктор технических наук, доцент, руководитель направления комплексных систем управления, ООО «ЛокоТех-Сигнал»; профессор кафедры «Автоматика, телемеханика и связь на железнодорожном транспорте», Российский университет транспорта, Москва, Россия.
E-mail: TrES-4b@yandex.ru

Vladimir V. Sapozhnikov, Dr. Sci. (Eng.), Professor, Professor of "Automation and Remote Control on Railways" Department, Emperor Alexander I St. Petersburg State Transport University, Saint Petersburg, Russia.
E-mail: at.pgups@gmail.com

Dmitry V. Efanov, Dr. Sci. (Eng.), Associate Professor, Head of Integrated Control Systems Division, "LocoTech-Signal" LLC; Professor of "Automation, Remote Control and Communication on Railway Transport" Department, Russian University of Transport, Moscow, Russia.
E-mail: TrES-4b@yandex.ru

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ
MATHEMATICAL MODELING

УДК 681.32
<https://doi.org/10.37661/1816-0301-2020-17-2-86-102>

Поступила в редакцию 23.09.2019
Received 23.09.2019

Принята к публикации 10.03.2020
Accepted 10.03.2020

**Алгоритмы планирования рабочего времени
в условиях интервальной неопределенности**

Ю. Н. Сотсков^{1✉}, Н. Г. Егорова¹, Н. М. Матвейчук²

¹*Объединенный институт проблем информатики
Национальной академии наук Беларуси, Минск, Беларусь*
✉E-mail: sotskov48@mail.ru

²*Белорусский государственный аграрный технический университет,
Минск, Беларусь*

Аннотация. Тайм-менеджмент используется для планирования доступного времени в соответствии с персональными целями и предпочтениями работника. Критерий эффективности тайм-менеджмента состоит в достижении поставленных целей за минимально возможное время. Предлагается краткий обзор публикаций по тайм-менеджменту. Показывается, как теория расписаний может использоваться в оптимальном планировании для тайм-менеджмента. Рассматривается задача построения оптимального расписания выполнения работником запланированных работ при условии, что при составлении расписания известны только нижняя и верхняя границы возможной длительности выполнения каждой из запланированных работ.

Разработаны программы для построения перестановки выполнения запланированных работ на основе построенных перестановок с наибольшим относительным полупериметром параллелепипеда оптимальности. Проведенный на компьютере вычислительный эксперимент показал эффективность предложенных алгоритмов при составлении расписаний для тайм-менеджмента.

Ключевые слова: теория расписаний, тайм-менеджмент, один прибор, среднее время обслуживания требований, неопределенные длительности обслуживания требований

Для цитирования. Сотсков, Ю. Н. Алгоритмы планирования рабочего времени в условиях интервальной неопределенности / Ю. Н. Сотсков, Н. Г. Егорова, Н. М. Матвейчук // Информатика. – 2020. – Т. 17, № 2. – С. 86–102. <https://doi.org/10.37661/1816-0301-2020-17-2-86-102>

Algorithms for planning working time under interval uncertainty

Yuri N. Sotskov^{1✉}, Natalja G. Egorova¹, Natalja M. Matsveichuk²

¹*The United Institute of Informatics Problems of the National Academy
of Sciences of Belarus, Minsk, Belarus*
✉E-mail: sotskov48@mail.ru

²*Belarusian State Agrarian Technical University, Minsk, Belarus*

Abstract. Time-management refers to planning the available time in correspondence with personal goals and preferences of the employee. The criterion for efficient time-management lies in achieving definite goals in minimum possible time. This paper presents a short review of the literature on time-management. It is shown how scheduling theory may be used for optimal planning in time-management. The problem of minimizing the total (average) weighted completion time of the planned jobs by an employee is considered provided that only

lower and upper bounds of the possible processing time of each job are known before scheduling. Software for constructing a permutation of the given jobs with the largest relative semi-perimeter of the optimality parallelepiped has been developed. Computational experiment on the computer showed the effectiveness of the developed algorithms for constructing schedules for time-management.

Keywords: scheduling theory, time-management, single machine, average job completion time, uncertain job processing time periods

For citation. Sotskov Yu. N., Egorova N. G., Matsveichuk N. M. Algorithms for planning working time under interval uncertainty. *Informatics*, 2020, vol. 17, no. 2, pp. 86–102 (in Russian). <https://doi.org/10.37661/1816-0301-2020-17-2-86-102>

Введение. Методики тайм-менеджмента основаны на психологических и социологических исследованиях, целью которых является эффективное планирование деятельности человека. Тайм-менеджмент представляет собой технологию организации целенаправленной деятельности в течение определенного периода времени. Совокупность правил, рекомендаций и утверждений позволяет более эффективно использовать время для повышения производительности труда, уменьшения возникающих перегрузок и усталости, устранения нервозности и стрессов. Использование тайм-менеджмента может позволить работнику экономить до 50 % времени на выполнение запланированных работ, потратив при этом до 10 % времени на планирование рабочего дня, недели, месяца [1, 2]. Это принесет пользу и повысит прибыль всей компании, поскольку «эффективная компания начинается с эффективных сотрудников» [3].

Эффективность использования технологии тайм-менеджмента оценивается по различным критериям. В качестве одного из таких критериев можно использовать максимизацию условного дохода, который связан со скорейшим завершением всех запланированных работ.

В настоящей статье рассматривается задача построения оптимальных расписаний выполнения запланированных работ одним исполнителем. Критерием оптимальности расписания является минимизация суммарного (или среднего) взвешенного времени завершения множества работ, запланированных исполнителем на определенный период времени. Предполагается, что на этапе составления расписания для каждой запланированной работы известны нижняя и верхняя границы возможной продолжительности работы.

Перестановка $\pi_e = (\pi_{e_1}, \pi_{e_2}, \dots, \pi_{e_p})$ выполнения работ в заданном интервале планирования определяется как конкатенация перестановок π_{e_k} выполнения работ, запланированных на k -й день. Перестановка π_{e_k} выполнения работ, запланированных на k -й день, может включать только те работы, которые поступили для выполнения на момент составления дневного расписания. Иными словами, перестановка π_{e_k} включает работы, поступившие в k -й день для последующего выполнения, а также работы, поступившие и не выполненные в предыдущие дни интервала планирования. В течение k -го дня работы выполняются исполнителем в соответствии с перестановкой π_k до тех пор, пока начало s_{k_r} выполнения очередной работы J_{k_r} согласно перестановке $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_d})$ не выйдет за пределы рабочего времени k -го дня.

В качестве приближенного решения задачи построения оптимального расписания на k -й день используется перестановка π_k выполнения запланированных работ с наибольшим относительным полупериметром параллелепипеда оптимальности. Перестановка π_k строится на основе блочной структуры распределения работ. В статье приводится ряд утверждений, которые используются для разработки полиномиальных алгоритмов и компьютерных программ для нахождения перестановки π_{e_k} , а также представлены результаты вычислительного эксперимента, проведенного на персональном компьютере для построения тестовых расписаний.

Обзор литературы и мотивация исследований. Большинство людей, хотя и стремятся повысить эффективность своей деятельности, все же недостаточно внимания уделяют планированию рабочего времени, что приводит к снижению эффективности работы и, как следствие, к уменьшению доходов [3]. В последние три десятилетия тайм-менеджмент стал популярным направлением учебных курсов, пособий, тренингов и семинаров. Тайм-менеджменту посвяще-

но множество статей и книг, предлагающих практические рекомендации по планированию работ и оценке эффективности использования рабочего и личного времени.

В статье [3] отмечается, что некоторые люди стремятся выполнять несколько работ и решать несколько сложных задач на работе или дома одновременно. При этом каждый человек может сомневаться в том, что у него будет достаточно времени для завершения начатых и запланированных работ. В результате переживаний у людей возникают проблемы со сном, поскольку и ночью их не покидают мысли о незавершенных и невыполненных делах. После бессонной ночи человек просыпается с чувством вины, начинает очередной день так же, как и предыдущий, с выполнения работ, которые являются малозначимыми и несущественными. Такой порочный круг продолжается день за днем, месяц за месяцем.

В работах [3, 4] описывается ситуация, возникающая из-за привычки человека откладывать на потом важные и срочные дела, которые могут быть неприятными и непривычными, либо из-за его стремления выполнять идеально малозначимые и бесполезные работы. Вследствие неорганизованности и потери контроля над своим временем у человека накапливается усталость, что приводит к низкой производительности и неэффективности усилий, потраченных на малозначимые работы. В итоге беспокойство, тревога и стресс становятся постоянными спутниками человека. В тайм-менеджменте даются рекомендации, как научиться «управлять своим временем, вместо того чтобы позволить времени управлять собой» [4].

В статье [2] приведен список вопросов, на которые предлагается ответить тем, кто сомневается в полезности тайм-менеджмента для работы. Тратите ли вы много времени на совещания? Часто ли вашу работу прерывают телефонные звонки или неожиданные посетители? Работаете ли вы дольше ваших подчиненных? Часто ли вы берете работу на дом? Загружен ли ваш рабочий стол бумагами, ожидающими решений? Чувствуете ли вы, что вам приходится вникать в каждую деталь делегированной подчиненному работы потому, что он не может выполнить поручения правильно и в установленные сроки? Если бы вы завершали работу за семь часов вместо восьми, как обычно, смогли бы вы эффективно использовать освободившееся время? Если хотя бы на несколько из перечисленных вопросов получены положительные ответы, то рабочее время используется вами неэффективно.

В публикациях [1, 5–11] описаны результаты исследований по оценке влияния применения тайм-менеджмента на эффективность работников, успеваемость учеников старших классов и студентов колледжей и университетов. Была обнаружена корреляция между использованием тайм-менеджмента и исполнением респондентов, а также между использованием тайм-менеджмента и академической успеваемостью учеников и студентов. Так, в работах [6–8] отмечается, что успеваемость студентов главным образом обусловлена их умением правильно распоряжаться доступным временем. Недостаточно эффективное планирование времени уменьшает объем и снижает качество академических достижений, приводит к недостаточности времени для личных и социальных мероприятий, снижает удовлетворенность полученными результатами в образовании, препятствует развитию таких черт личности, как критическое мышление и способность принимать обоснованные решения. Между тем в статье [6] утверждается, что с повышением доли дистанционного образования значение тайм-менеджмента для эффективной учебы студентов будет возрастать.

В статье [9] отмечено, что показатель творческих способностей человека зависит от использования тайм-менеджмента, при этом познавательные способности человека не оказывают существенного влияния на эффективность его применения. Таким образом, тайм-менеджмент представляет собой технологию, которую практически любой индивидуум может использовать для повышения качества своих достижений и эффективности работы.

В статье [12] исследовалась устойчивость связи между применением технологии тайм-менеджмента и основными видами памяти (перспективной и ретроспективной). На основе полученных данных был сделан вывод о том, что технология тайм-менеджмента помогает людям даже с плохой памятью успешно выполнять поставленные перед ними задачи. Замечено также, что использование тайм-менеджмента со временем позволяет все более точно оценивать время, которое фактически требуется для выполнения той или иной работы.

В работах [10, 11, 13] обращается внимание на то, как по-разному люди следуют одним и тем же рекомендациям тайм-менеджмента. В [10] отмечено, что для успешного применения

тайм-менеджмента недостаточно формально ознакомиться с этой методикой, необходимо освоить ее применение на практике. В [11] показано, что разработка консультантом индивидуальных ежедневных планов с учетом личных особенностей человека является наиболее эффективной для повышения его производительности. Некоторые публикации по тайм-менеджменту могут вызвать у читателя беспокойство, неудовлетворенность собой и даже чувство вины, поскольку следовать на практике рекомендациям тайм-менеджмента, оказывается, достаточно сложно, что может привести человека к неуверенности в себе. Стресс, чувство вины и потеря контроля над своим временем могут повысить тревожность человека. В таком состоянии человеку трудно взяться за выполнение новых и важных для него работ. Это согласуется с замечаниями, приведенными в статьях [4, 11], о том, что самостоятельное составление человеком расписания выполнения запланированных работ может повлечь сомнения в правильности (оптимальности) построенного расписания. В результате может возникнуть дополнительная причина для беспокойства. В этой связи следует обратить внимание на публикации [2, 4], авторы которых предостерегают от механического, непродуманного использования рекомендаций тайм-менеджмента неподготовленными пользователями.

В статье [14] отмечается, что общее структурирование и упорядочение рабочего и личного времени на основе технологии тайм-менеджмента, как правило, уменьшает чувство беспокойства человека по поводу выполнения запланированных работ. Обнаружена отрицательная корреляция между уровнем тревожности респондентов и применением тайм-менеджмента. Ключевым компонентом, позволяющим повысить уверенность при организации рабочего и личного времени, является работа с календарем. Большинство людей плохо используют как бумажные, так и электронные календари. Описанный в работе [15] эксперимент показал, что менее 15 % респондентов записывали в календаре три или более работы, запланированные на день. Установлено, что успешное применение тайм-менеджмента повышает мотивацию человека использовать эту технологию в дальнейшем.

Тайм-менеджмент можно применять как средство эффективного распределения времени с целью своевременного выполнения запланированных работ [4, 10]. Тайм-менеджмент включает: решение о том, какие работы следует выбрать для выполнения (правило 1); расстановку приоритетов работ как по важности, так и по срочности и последующее упорядочение выбранных работ в соответствии с приоритетами (правило 2); концентрацию внимания на наиболее важных работах и выполнение их в периоды наибольшей активности человека (правило 3); эффективное управление возможными обстоятельствами, отвлекающими от выполнения начатой работы (правило 4).

Относительно правила 4 отмечается, что следует избегать прерываний выполнения начатой работы (незапланированных совещаний, телефонных разговоров, приема посетителей), поскольку помимо прямых потерь времени возникает необходимость тратить время на повторную подготовку к выполнению прерванной работы [2, 3].

Рассмотрим правила 1 и 2. Для рабочего времени общий список планируемых к выполнению работ составляется достаточно просто (в отличие от процесса определения общих жизненных целей человека на продолжительный период времени). Поэтому при составлении расписания на небольшой интервал (день, неделю и даже месяц) вначале следует рассмотреть весь список предполагаемых работ и выбрать из этого списка наиболее важные работы, которые планируется выполнить именно в данный период времени. Далее регулярно (еженедельно, а по возможности ежедневно) следует пересматривать список запланированных работ, поскольку приоритеты работ могут с течением времени меняться [4].

Во многих публикациях по тайм-менеджменту указывается на необходимость упорядочения выбранных работ. В статье [4] обращается внимание на то, что упорядочение работы особенно полезно в случаях, когда имеется эффективный алгоритм оптимального упорядочения или у пользователя есть возможность делегировать этот процесс. В противном случае упорядочение выбранных работ само по себе может стать для исполнителя непростой работой, требующей и дополнительного времени, и определенных навыков. Как правило, расстановку приоритетов для выполнения работ, запланированных на определенный интервал, и определение порядка невыполненных работ приходится выполнять многократно в течение интервала планирования. Поэтому так важно иметь быстро реализуемое средство для определения оптимального порядка

для выполнения запланированных работ с целью оперативного и эффективного управления своим временем.

Приведенный обзор публикаций [1–15] позволяет сделать вывод о том, что тайм-менеджмент представляет собой средство, помогающее пользователю решать различные вопросы, связанные с оптимальным распределением своего времени, с целью выполнения большего числа наиболее полезных работ в оптимальные для этих работ сроки. Для более эффективного применения технологии тайм-менеджмента желательно автоматизировать процесс составления расписаний выполнения запланированных работ для того, чтобы пользователь мог достаточно оперативно получать рекомендации по порядку выполнения текущих и новых для него работ. Заканчивая очередную работу, пользователь должен иметь возможность быстро определить, какую именно работу из множества запланированных, но еще не выполненных работ следует начинать в данный момент времени.

Далее описываются алгоритмы, предназначенные для составления близких к оптимальным расписаний при выполнении множества запланированных работ. Эффективность алгоритмов и разработанных программ продемонстрирована на случайно сгенерированных примерах.

Постановка задачи оптимального упорядочения. Из публикаций [1–15] следует важность разработки алгоритмов для решения задачи составления оптимальных расписаний выполнения работ одним исполнителем. Такая задача составления расписаний имеет ряд особенностей, первая из которых состоит в неопределенности длительностей выполнения запланированных работ на момент составления расписания их выполнения. Действительно, для человека в отличие от станка, робота или машины довольно сложно заранее определить точное время, которое фактически потребуется для выполнения той или иной работы. Между тем можно заранее определить вполне реалистичную оценку промежутка времени, который будет заведомо содержать длительность выполнения работы исполнителем, т. е. можно заранее определить верхние и нижние границы возможной длительности выполнения каждой работы из общего списка запланированных работ.

Вторая особенность определяется возможностью для исполнителя выбирать из списка предлагаемых работ множество тех работ, выполнение которых следует запланировать в течение определенного интервала времени. Согласно одному из принципов Парето 20 % выполняемых исполнителем работ приносят ему 80 % прибыли, поэтому вполне оправданным представляется подход, позволяющий исполнителю перед составлением расписания на определенный период времени выбирать, какие именно работы из списка возможных работ следует выполнить в течение заданного интервала планирования. Вместе с тем работы могут иметь различную важность для исполнителя. Важность работы можно характеризовать положительным числом, которое будем называть весом. Вес работы может быть связан с условным доходом, который получит исполнитель в результате ее выполнения. Целью планирования работ для исполнителя будем считать максимизацию его условного дохода от выполнения запланированных работ. Достаточно правдоподобным выглядит предположение о том, что доход человека связан с моментами завершения запланированных работ (с учетом их весов), причем доход от выполненной исполнителем конкретной работы тем больше, чем раньше он выполнит эту работу.

Таким образом, получаем задачу, в которой необходимо составить расписание выполнения одним исполнителем множества выбранных работ с неопределенными длительностями, которое является оптимальным в смысле максимизации дохода от реализации расписания выполнения всех запланированных работ. Такие задачи рассматриваются в рамках теории расписаний, что позволяет использовать принятую в теории расписаний терминологию для описания алгоритмов и полученных результатов.

Следует отметить, что классические алгоритмы теории расписаний не учитывают особенности, которые отличают множество работ, выполняемых человеком.

Рассмотрим задачу составления расписаний для одного обслуживающего прибора с неопределенными числовыми параметрами и критерием минимизации суммы взвешенных моментов завершения выполнения всех запланированных работ. В соответствии с трехпозиционной формой $\alpha/\beta/\gamma$ классификации задач теории расписаний [16–19] такую задачу будем обозначать $1|p_i^L \leq p_i \leq p_i^U|\sum w_i C_i$. В этой форме обозначения позиция α характеризует число обслужива-

ющих приборов (в рассматриваемом случае это один исполнитель) и тип системы обслуживания. Позиция β определяет характеристики обслуживаемых требований (т. е. работ одного исполнителя). Позиция γ определяет одну или несколько целевых функций, задающих оптимальность искомого расписания выполнения всех запланированных работ. Отметим, что в большинстве задач теории расписаний рассматривается критерий минимизации значения целевой функции на множестве допустимых расписаний.

В задаче $1 \mid p_i^L \leq p_i \leq p_i^U \mid \sum w_i C_i$ требуется построить оптимальное расписание выполнения одним исполнителем множества работ $G = \{J_1, J_2, \dots, J_n\}$ с заданными весами $w_i > 0$, $J_i \in G$.

В качестве критерия оптимальности используется минимизация суммы $\sum_{i=1}^n w_i C_i$ взвешенных моментов C_i завершения выполнения всех работ $J_i \in G$. Поскольку число работ n зафиксировано на момент составления расписания, то минимизация суммы $\sum_{i=1}^n w_i C_i$ взвешенных моментов C_i завершения выполнения всех работ $J_i \in G$ эквивалентна минимизации среднего взвешенного времени $\sum_{i=1}^n w_i C_i / n$ завершения выполнения каждой работы $J_i \in G$.

Покажем, что задачу минимизации целевой функции $\sum_{i=1}^n w_i C_i$ можно интерпретировать как задачу максимизации суммарной прибыли исполнителя. Пусть выполнение работы $J_i \in G$ приносит исполнителю прибыль, оцениваемую величиной $H - C_i$, где H – достаточно большое число. Чем раньше закончена работа, тем больше прибыль исполнителя. Задача минимизации величины $\sum_{i=1}^n w_i C_i$ равносильна задаче максимизации суммарной прибыли $\sum_{i=1}^n (H - w_i C_i) = nH - \sum_{i=1}^n w_i C_i$ исполнителя, поскольку произведение nH не зависит от порядка выполнения работ множества $G = \{J_1, J_2, \dots, J_n\}$.

В неопределенной задаче $1 \mid p_i^L \leq p_i \leq p_i^U \mid \sum w_i C_i$ предполагается, что на момент построения расписания известны только нижняя $p_i^L > 0$ и верхняя $p_i^U \geq p_i^L$ границы возможной длительности выполнения работы $J_i \in G$. Работа $J_i \in G$ должна быть выполнена без прерываний процесса выполнения за время $p_i \in [p_i^L, p_i^U]$, которое становится известным в момент завершения выполнения этой работы.

Если границы p_i^L и p_i^U для каждой работы $J_i \in G$ совпадают, то неопределенная задача $1 \mid p_i^L \leq p_i \leq p_i^U \mid \sum w_i C_i$ превращается в детерминированную задачу $1 \parallel \sum w_i C_i$, которая решается за время $O(n \log n)$ [16].

Пусть $T = \{p : p \in R_+^n, p_i^L \leq p_i \leq p_i^U, i \in \{1, 2, \dots, n\}\}$ обозначает множество всех векторов $p = (p_1, p_2, \dots, p_n)$ возможных длительностей выполнения работ из заданного множества $G = \{J_1, J_2, \dots, J_n\}$, а R_+^n – множество векторов размерности n с неотрицательными действительными компонентами. Вектор $p \in T$ длительностей работ принято называть сценарием.

Для неопределенной задачи $1 \mid p_i^L \leq p_i \leq p_i^U \mid \sum w_i C_i$, вообще говоря, не существует такой перестановки из множества $S = \{\pi_1, \pi_2, \dots, \pi_n\}$ всех $n!$ перестановок выполнения работ множества G , которая оставалась бы оптимальной при всех сценариях из множества T .

Для оценки устойчивости оптимальной для сценария $p = (p_1, p_2, \dots, p_n)$ перестановки $\pi_k \in S$ выполнения работ множества G к вариациям T длительностей выполнения этих работ предлагается использовать относительный полупериметр параллелепипеда оптимальности

перестановки $\pi_k \in S$. В статьях [17, 18] были исследованы свойства параллелепипеда оптимальности.

Блоки работ и параллелепипед оптимальности для перестановки $\pi_k \in S$ их выполнения. Детерминированную задачу $1 | \sum w_i C_i$ с фиксированным сценарием $p \in T$ будем обозначать $1 | p | \sum w_i C_i$. Пусть $M = (k_1, k_2, \dots, k_{i_{M1}})$, $k_1 < k_2 < \dots < k_{i_{M1}}$, обозначает упорядоченное подмножество множества $\{1, 2, \dots, n\}$.

Определение 1. Максимальный (по включению) параллелепипед

$$OB(\pi_k, T) = [l_{k_1}^*, u_{k_1}^*] \times [l_{k_2}^*, u_{k_2}^*] \times \dots \times [l_{k_{i_{M1}}}^*, u_{k_{i_{M1}}}^*] \subseteq T$$

называют *параллелепипедом оптимальности* для перестановки $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$ выполнения работ $G = \{J_1, J_2, \dots, J_n\}$, если для любого сценария $p = (p_1, p_2, \dots, p_n) \in T$, при котором перестановка π_k является оптимальной для задачи $1 | p | \sum w_i C_i$, эта перестановка остается оптимальной и для задачи $1 | p' | \sum w_i C_i$ со сценарием

$$p' \in [p_1, p_1] \times [p_2, p_2] \times \dots \times [p_{i_g-1}, p_{i_g-1}] \times [l_{i_g}^*, u_{i_g}^*] \times [p_{i_g+1}, p_{i_g+1}] \times \dots \times [p_n, p_n]$$

для любого индекса $i_g \in M = (k_1, k_2, \dots, k_{i_{M1}})$. Если не существует сценария $p \in T$, при котором перестановка π_k является оптимальной для задачи $1 | p | \sum w_i C_i$, то полагаем $OB(\pi_k, T) = \emptyset$.

Вариация p' длительности p_{i_g} выполнения работы $J_{i_g} \in G$ в пределах указанного в определении 1 отрезка $[l_{i_g}^*, u_{i_g}^*]$ не нарушает оптимальности перестановки $\pi_k \in S$, если имеет место включение $p_{i_g} \in [l_{i_g}^*, u_{i_g}^*]$. Непустой отрезок $[l_{i_g}^*, u_{i_g}^*]$, $l_{i_g}^* < u_{i_g}^*$, будем называть *отрезком оптимальности* работы J_{i_g} в перестановке $\pi_k \in S$.

В качестве приближенного решения задачи $1 | p_i^L \leq p_i \leq p_i^U | \sum w_i C_i$ будем использовать перестановку π_k с максимальным значением относительного полупериметра $PerOB(\pi_k)$, равно сумме величин $(u_{i_g}^* - l_{i_g}^*) / (p_{i_g}^U - p_{i_g}^L)$ для всех работ G . Перестановку $\pi_k \in S$ с максимальным значением $PerOB(\pi_k)$ будем называть *эффективной* для задачи $1 | p_i^L \leq p_i \leq p_i^U | \sum w_i C_i$.

Определение 2. Блоком называют максимальное подмножество $B_r = \{J_{r_1}, J_{r_2}, \dots, J_{r_{|B_r|}}\} \subseteq G$ множества заданных работ $G = \{J_1, J_2, \dots, J_n\}$, для которых выполняется соотношение

$$\min_{J_{r_i} \in B_r} \left\{ \frac{p_{r_i}^U}{w_{r_i}} \right\} \geq \max_{J_{r_i} \in B_r} \left\{ \frac{p_{r_i}^L}{w_{r_i}} \right\}. \text{ Отрезок } [b_r^L, b_r^U], \text{ для которого выполняются равенства } b_r^L = \\ = \max_{J_{r_i} \in B_r} \left\{ \frac{p_{r_i}^L}{w_{r_i}} \right\} \text{ и } b_r^U = \min_{J_{r_i} \in B_r} \left\{ \frac{p_{r_i}^U}{w_{r_i}} \right\}, \text{ называют ядром блока } B_r.$$

Работы, принадлежащие только одному блоку, будем называть *фиксированными* (в блоках), а принадлежащие нескольким блокам – *нефиксированными*. Блоки, содержащие только нефиксированные работы, будем называть *виртуальными*.

Алгоритмы приближенного решения задачи $1 | p_i^L \leq p_i \leq p_i^U | \sum w_i C_i$. В статье [17] разработан алгоритм сложности $O(n \log n)$ нахождения всех блоков B для задачи $1 | p_i^L \leq p_i \leq p_i^U | \sum w_i C_i$. Будем полагать, что все блоки $\{B_1, B_2, \dots, B_m\} = B$ пронумерованы в порядке возрастания как левых, так и правых границ их ядер. В статье [17] доказана следующая теорема.

Теорема 1. Если существуют два смежных блока B_r и B_{r+1} без общих работ, $B_r \cap B_{r+1} = \emptyset$, то задачу $1 | p_i^L \leq p_i \leq p_i^U | \sum w_i C_i$ можно декомпозировать на две подзадачи с множествами работ P_1 и P_2 , $G = P_1 \cup P_2$, где P_1 – множество работ из блоков B_1, B_2, \dots, B_r , а P_2 – множество работ из блоков $B_{r+1}, B_{r+2}, \dots, B_m$. Полученные параллелепипеды оптимальности $OB(\pi_k(1), T_1)$ и $OB(\pi_k(2), T_2)$, $T = T_1 \cup T_2$, для таких задач определяют параллелепипед оптимальности $OB(\pi_k, T) = OB(\pi_k(1), T_1) \times OB(\pi_k(2), T_2)$ для задачи $1 | p_i^L \leq p_i \leq p_i^U | \sum w_i C_i$.

Поскольку декомпозиция задачи на подзадачи возможна при наличии смежных блоков без общих работ, то справедливо неравенство $\min_{J_{r_i} \in B_{r+1}} \left\{ \frac{p_{r_i}^L}{w_{r_i}} \right\} \geq \max_{J_{r_i} \in B_r} \left\{ \frac{p_{r_i}^U}{w_{r_i}} \right\}$ и для декомпозиции задачи на подзадачи достаточно в перестановке $\pi_l = (J_{l_1}, J_{l_2}, \dots, J_{l_n})$, в которой работы упорядочены в порядке неубывания величины p_i^L/w_i , найти работы J_{l_k} , удовлетворяющие условию $\frac{p_{l_k}^L}{w_{l_k}} \geq \max_{1 \leq j \leq k-1} \left\{ \frac{p_{l_j}^U}{w_{l_j}} \right\}$. В случае существования такой работы J_{l_k} задачу $1 | p_i^L \leq p_i \leq p_i^U | \sum w_i C_i$

можно декомпозировать на две подзадачи с множествами работ $P_1 = \{J_{l_1}, J_{l_2}, \dots, J_{l_{k-1}}\}$ и

$P_2 = \{J_{l_k}, J_{l_{k+1}}, \dots, J_{l_n}\}$, так как $\frac{p_{l_k}^L}{w_{l_k}} = \min_{J_{r_i} \in B_{r+1}} \left\{ \frac{p_{r_i}^L}{w_{r_i}} \right\} \geq \max_{J_{r_i} \in B_r} \left\{ \frac{p_{r_i}^U}{w_{r_i}} \right\} = \max_{0 < j < k} \left\{ \frac{p_{l_j}^U}{w_{l_j}} \right\}$. Поскольку

$\max_{1 \leq j \leq k-1} \left\{ \frac{p_{l_j}^U}{w_{l_j}} \right\} = \max \left\{ \max_{1 \leq j \leq k-2} \left\{ \frac{p_{l_j}^U}{w_{l_j}} \right\}, \frac{p_{l_{k-1}}^U}{w_{l_{k-1}}} \right\}$, то для декомпозиции задачи $1 | p_i^L \leq p_i \leq p_i^U | \sum w_i C_i$,

в которой работы упорядочены в порядке неубывания величины p_i^L/w_i , достаточно выполнить $O(n)$ элементарных операций.

В статье [17] разработан алгоритм сложности $O(n \log n)$ декомпозиции задачи $1 | p_i^L \leq p_i \leq p_i^U | \sum w_i C_i$ на подзадачи и установлено, что если задача $1 | p_i^L \leq p_i \leq p_i^U | \sum w_i C_i$ содержит только один блок, то параллелепипед $OB(\pi_k, T)$ с максимальным значением величины $PerOB(\pi_k)$ определяется первой, второй, предпоследней и последней работами в перестановке $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_n}) \in S$, которые можно выбрать на основе соотношений

$$\left(X_1 - \frac{p_{k_1}^L}{w_{k_1}} \right) / \left(\frac{p_{k_1}^U}{w_{k_1}} - \frac{p_{k_1}^L}{w_{k_1}} \right) = \max_{G_i^*} \left\{ \left(X_1 - \frac{p_i^L}{w_i} \right) / \left(\frac{p_i^U}{w_i} - \frac{p_i^L}{w_i} \right) \right\}, \quad \frac{p_{k_2}^L}{w_{k_2}} = \max_{G_i^*} \left\{ \frac{p_i^L}{w_i} \right\},$$

$$\frac{p_{k_{n-1}}^U}{w_{k_{n-1}}} = \min_{G_i^*} \left\{ \frac{p_i^U}{w_i} \right\}, \quad \left(\frac{p_{k_n}^U}{w_{k_n}} - X_2 \right) / \left(\frac{p_{k_n}^U}{w_{k_n}} - \frac{p_{k_n}^L}{w_{k_n}} \right) = \max_{G_i^*} \left\{ \left(\frac{p_i^U}{w_i} - X_2 \right) / \left(\frac{p_i^U}{w_i} - \frac{p_i^L}{w_i} \right) \right\},$$

где G_i^* обозначает множество работ, не выбранных в позиции $q \in \{1, 2, n-1, n\}$, $X_1 = \max_{G_i^*} \left\{ \frac{p_i^L}{w_i} \right\}$,

$$X_2 = \min_{G_i^*} \left\{ \frac{p_i^U}{w_i} \right\}.$$

В статье [18] доказана следующая теорема.

Теорема 2. Если $B_1 = G$, то перестановку $\pi_k \in S$ с максимальным значением $PerOB(\pi_k)$ для задачи $1 | p_i^L \leq p_i \leq p_i^U | \sum w_i C_i$ можно построить за время $O(n)$.

Если каждая нефиксированная работа из множества G назначена в один из блоков подмножества блоков, которым она принадлежит, то полученная на основе теоремы 2 перестановка π_k с максимальным значением $PerOB(\pi_k)$ может быть построена за время $O(n \log n)$.

При построении эффективной перестановки для задачи $1 | p_i^L \leq p_i \leq p_i^U | \sum w_i C_i$, если условия теорем 1 и 2 не выполняются, используется алгоритм динамического программирования (ДМ).

Алгоритм ДМ строит дерево решений высотой $m = |B|$, в котором ранг вершины соответствует номеру анализируемого блока работ. На каждой итерации t алгоритма ДМ для каждой висячей вершины $\pi^j(B_r, G^j) \in V$, выбранной из множества построенных вершин (j – номер вершины в дереве решений, B_r – номер блока, G^j – множество нефиксированных работ, принадлежащих блоку B_r в вершине $\pi^j(B_r, G^j) \in V$), строим множество всех потомков следующим образом. Пусть B_{r+1}^t обозначает множество всех работ, фиксированных в блоке B_{r+1} , а также нефиксированных работ блока B_{r+1} , не распределенных в свои блоки на предыдущих итерациях алгоритма ДМ. Рассмотрим множество $G[B_{r+1}^t] \subseteq B_{r+1}^t$, состоящее из нефиксированных работ, принадлежащих блоку B_{r+2} . Поскольку работы множества $G[B_{r+1}^t]$ на последующих итерациях могут быть распределены как в блок B_{r+1} , так и в один из последующих блоков, то для этих работ строится множество всех подмножеств $P(G[B_{r+1}^t]) = \{\emptyset, \dots, G[B_{r+1}^t]\}$, которые порождают $2^{|G[B_{r+1}^t]|}$ вершин дерева решений (потомков вершины $\pi^j(B_r, G^j) \in V$).

Пусть G_i , $|G_i| = m_i$, обозначает множество нефиксированных работ, распределенных в блок B_{r+1} в вершине $\pi^i(B_{r+1}, G^i)$, $G^i \in P(G[B_{r+1}^t])$, дерева решений, а также фиксированных работ в соответствующем блоке B_{r+1} . В каждой построенной вершине $\pi^i(B_{r+1}, G^i)$, $G^i \in P(G[B_{r+1}^t])$, дерева решений упорядочим работы множества G_i и вычислим значение штрафа $F(\pi^i)$. Для множества упорядоченных работ $G_i' = (J_{i_1}, J_{i_2}, \dots, J_{i_{m_i-1}}, J_{i_{m_i}})$ в вершине $\pi^i(B_{r+1}, G^i)$ штраф $F(\pi^i)$ вычисляется по формулам

$$F(\pi^i) = F(\pi^j) + \Delta, \quad F(\emptyset) = 0,$$

$$\Delta = \sum_{k=1}^{m_i} \left(1 - \frac{u_{i_k}^* - l_{i_k}^*}{p_{i_k}^U - p_{i_k}^L} \right) + \max \left\{ 0, \left(\frac{p_{j_{m_j}}^U}{w_{j_{m_j}}} - \max \left\{ \frac{p_{i_1}^L}{w_{i_1}}, \frac{p_{j_{m_j}}^L}{w_{j_{m_j}}}, \frac{p_{j_{m_j-1}}^U}{w_{j_{m_j-1}}} \right\} \right) / (p_{j_{m_j}}^U - p_{j_{m_j}}^L) \right\}.$$

В статье [17] доказана следующая теорема.

Теорема 3. Если работа J_i принадлежит хотя бы двум блокам, $J_i \in B_r$, $J_i \in B_{r+1}$, и в эффективной перестановке π_k работа J_i следует за всеми работами (предшествует хотя бы одной работе) J_j , $J_j \in B_r$, $J_j \notin B_{r+1}$, то эффективную перестановку π_k можно получить в результате включения работы J_i в блок B_{r+s} , $s \geq 1$ (в блок B_{r-s} , $s \geq 0$, соответственно).

Из теоремы 3 следует, что при упорядочении работ внутри блока B_{r+1} в вершине $\pi^i(B_{r+1}, G^i)$ дерева решений достаточно рассмотреть случаи, в которых последняя работа $J_{i_{m_i}}$ блока B_{r+1} удовлетворяет условиям $J_{i_{m_i}} \in B_{r+1}$ и $J_{i_{m_i}} \notin B_{r+2}$. При этом упорядочение работ производится аналогично решению задачи $1 | p_i^L \leq p_i \leq p_i^U | \sum w_i C_i$ с одним блоком $B_1 = G$, которое описано в статье [17] и имеет сложность $O(|B_r|)$. Для этого достаточно выбрать работы в первую, вторую, предпоследнюю ($m_i - 1$) и последнюю (m_i) позиции перестановки работ из блока B_{r+1} на основе соотношений

$$\frac{X_1 - \frac{\tilde{p}_i^L}{w_i}}{\frac{p_i^U}{w_i} - \frac{p_i^L}{w_i}} = \max_{G_i^*} \left\{ \begin{array}{l} \max \left\{ 0, \left(X_1 - \max \left\{ \frac{p_{j_m}^U}{w_{j_m}}, \frac{p_i^L}{w_i} \right\} \right) / \left(\frac{p_i^U}{w_i} - \frac{p_i^L}{w_i} \right) \right\} - \\ - \max \left\{ 0, \left(\frac{p_{j_m}^U}{w_{j_m}} - \max \left\{ \frac{p_i^L}{w_i}, \frac{p_{j_m}^L}{w_{j_m}}, \frac{p_{j_{m-1}}^U}{w_{j_{m-1}}} \right\} \right) / \left(\frac{p_{j_m}^U}{w_{j_m}} - \frac{p_{j_m}^L}{w_{j_m}} \right) \right\} \end{array} \right\},$$

$$\frac{p_{i_2}^L}{w_{i_2}} = \max_{G_i^*} \left\{ \frac{p_i^L}{w_i} \right\}, \quad \frac{p_{i_{m-1}}^U}{w_{i_{m-1}}} = \min_{G_i^*} \left\{ \frac{p_i^U}{w_i} \right\},$$

$$\left(\frac{p_{i_{m_i}}^U}{w_{i_{m_i}}} - X_2 \right) / \left(\frac{p_{i_{m_i}}^U}{w_{i_{m_i}}} - \frac{p_{i_{m_i}}^L}{w_{i_{m_i}}} \right) = \max_{G_i^*, J_{i_{m_i}} \in B_{r+1}, J_{i_{m_i}} \notin B_{r+2}} \left\{ \left(\frac{p_i^U}{w_i} - X_2 \right) / \left(\frac{p_i^U}{w_i} - \frac{p_i^L}{w_i} \right) \right\},$$

где G_i^* обозначает множество работ, не выбранных в позиции $q \in \{1, 2, m_i - 1, m_i\}$,

$$X_1 = \max_{G_i^*} \left\{ \frac{p_i^L}{w_i} \right\}, \quad X_2 = \min_{G_i^*} \left\{ \frac{p_i^U}{w_i} \right\}.$$

Вершины $\pi^j(B_r, G^j)$ дерева решений с одинаковыми номерами блока B_r и наборами фиксированных работ в блоке B_r и предыдущих блоках, для которых выполняется неравенство

$$\frac{p_{j_{m_j}}^U}{w_{j_{m_j}}} \leq \min_{J_i \in B_{r+1}} \left\{ \frac{p_i^L}{w_i} \right\},$$

определяют одно и то же состояние в пространстве решений задачи

$$1 | p_i^L \leq p_i \leq p_i^U | \sum w_i C_i.$$

Из множества висячих вершин построенного дерева решений для дальнейшего ветвления выбирается вершина $\pi^i(B_{r+1}, G^i)$ с наименьшим штрафом $F(\pi^i)$. Если таких вершин окажется несколько, то выбирается висячая вершина $\pi^i(B_{r+1}, G^i)$ с наименьшим штрафом $F(\pi^i)$ из построенного дерева решений с наибольшим номером $r+1$ блока B_{r+1} .

Определим условия окончания работы алгоритма ДМ: должна быть построена перестановка, содержащая все работы множества G ; для построенной перестановки работ множества G должен быть получен минимальный штраф $F(\pi^i)$ среди всех штрафов для нерассмотренных (висячих) вершин построенного дерева решений. Если эти условия выполнены, то следует восстановить эффективную перестановку обратным ходом по построенному дереву решений.

План вычислительного эксперимента. Был проведен вычислительный эксперимент на персональном компьютере по оценке эффективности разработанных алгоритмов для исполнителя, использующего технологию тайм-менеджмента в течение месяца. В эксперименте оценивался 30-дневный период составления расписаний на каждый день. Предполагалось, что каждый день поступает 10 новых работ для последующего выполнения исполнителем.

Для 10 работ из множества $G_1 = \{J_1, J_2, \dots, J_{10}\}$, поступивших для выполнения в первый день, строилась эффективная перестановка $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_{10}})$ выполнения работ из этого множества. Предполагалось, что для выполнения запланированных на один день работ исполнитель располагал не более чем 800 единицами времени в течение каждого дня. Все расчеты проводились в целых числах указанных единиц времени.

В течение первого дня работы из множества $G_1 = \{J_1, J_2, \dots, J_{10}\}$ выполнялись исполнителем в соответствии с расписанием (перестановкой π_k для 10 поступивших для выполнения работ) до тех пор, пока начало s_{k_r} выполнения очередной работы J_{k_r} в перестановке $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_{10}})$ не выходило за пределы рабочего времени первого дня, т. е. пока выполнялось неравенство $s_{k_r} < 800$.

Все работы подмножества $G(1)$ множества $G_1 = \{J_1, J_2, \dots, J_{10}\}$, которые не были выполнены в первый день, добавлялись к множеству 10 работ $G_2 = \{J_{11}, J_{12}, \dots, J_{20}\}$, поступивших для выполнения во второй день. К началу второго дня строилась эффективная перестановка $\pi_h = (J_{h_1}, J_{h_2}, \dots, J_{h_{10+G(1)}})$ выполнения всех работ из множества $G_2 \cup G(1)$. В течение второго дня работы множества $G_2 \cup G(1)$ выполнялись исполнителем до тех пор, пока начало s_{h_u} выполнения очередной работы J_{h_u} в перестановке $\pi_h = (J_{h_1}, J_{h_2}, \dots, J_{h_{10+G(1)}})$ не выходило за пределы рабочего времени второго дня, т. е. $s_{h_u} < 800$.

Все работы подмножества $G(2)$ множества работ $G_2 \cup G(1)$, не выполненные во второй день, добавлялись к множеству 10 работ $G_3 = \{J_{21}, J_{22}, \dots, J_{30}\}$, поступивших для выполнения в третий день. К началу третьего дня строилась эффективная перестановка $\pi_t = (J_{t_1}, J_{t_2}, \dots, J_{t_{10+G(2)}})$ выполнения работ из множества $G_3 \cup G(2)$. В течение третьего дня работы множества $G_3 \cup G(2)$ выполнялись исполнителем до тех пор, пока начало s_{t_v} выполнения очередной работы J_{t_v} в перестановке $\pi_t = (J_{t_1}, J_{t_2}, \dots, J_{t_{10+G(2)}})$ не выходило за пределы рабочего времени третьего дня, т. е. $s_{t_v} < 800$.

Описанный выше процесс выполнения работ исполнителем продолжался аналогично вплоть до 30-го дня включительно. Начиная с 31-го дня новые работы для исполнителя не поступали. В последующие после 30-го числа дни построение эффективных перестановок для выполнения исполнителем производилось только для подмножества ранее поступивших для выполнения работ, которые не были выполнены исполнителем в течение предыдущих 30 дней. Такое изменение регламента поступления новых работ для исполнителя в 31-й и, возможно, в последующие дни было обусловлено необходимостью сравнения эффективности разработанных алгоритмов с другими известными алгоритмами для того, чтобы данное сравнение значений целевой функции $\sum w_i C_i / n$ выполнялось на одном и том же множестве и при одном и том же значении n выполненных исполнителем работ. Отметим, что и при реальном планировании работ можно предполагать, что все запланированные для исполнителя работы должны быть в конечном итоге полностью выполнены. К началу последнего дня интервала планирования строилась эффективная перестановка $\pi_p = (J_{p_1}, J_{p_2}, \dots, J_{p_q})$.

В проведенном на персональном компьютере вычислительном эксперименте из частичных перестановок $\pi_k = (J_{k_1}, J_{k_2}, \dots, J_{k_r}, \dots, J_{k_{10}})$, $\pi_h = (J_{h_1}, J_{h_2}, \dots, J_{h_u}, \dots, J_{h_{10+G(1)}})$, $\pi_t = (J_{t_1}, J_{t_2}, \dots, J_{t_v}, \dots, J_{t_{10+G(2)}})$, ..., $\pi_p = (J_{p_1}, J_{p_2}, \dots, J_{p_q})$ выполнения работ в каждый день из интервала планирования строилась перестановка $\pi_e = (J_{k_1}, J_{k_2}, \dots, J_{k_r}, J_{h_1}, J_{h_2}, \dots, J_{h_u}, J_{t_1}, J_{t_2}, \dots, J_{t_v}, \dots, J_{p_1}, J_{p_2}, \dots, J_{p_q})$ выполнения исполнителем всех работ в течение всех дней рассматриваемого интервала планирования. Аналогично из частичных перестановок выполнения работ в течение дня строились перестановки π_{opt} и π_{mid-p} .

Перестановка π_{opt} строилась из частичных перестановок работ, которые являются оптимальными для фактических длительностей $p^* = (p_1^*, p_2^*, \dots, p_n^*)$ выполнения работ. Напомним, что все фактические длительности выполнения работ становятся известными только после завершения выполнения всех работ множества G .

Перестановка π_{mid-p} строилась из частичных перестановок работ, оптимальных для соответствующей детерминированной задачи $1|p|\sum w_i C_i$, с фиксированными средними значениями $p_i = 1/2(p_i^U + p_i^L)$ заданных интервалов длительностей выполнения работ.

Цель вычислительного эксперимента состояла в определении отклонения $\gamma = [(\Phi - \Phi^*) / \Phi^*] \cdot 100\%$ значения целевой функции $\Phi = \sum_{i=1}^n w_i C_i^0$, полученного для перестановки π_e выполнения исполнителем запланированных работ, от значения целевой функции $\Phi^* = \sum_{i=1}^n w_i C_i^*$, полученного для перестановки π_{opt} . Кроме того, определялось отклонение $\gamma_{mid-p} = [(\Phi^{mid-p} - \Phi^*) / \Phi^*] \cdot 100\%$ значения целевой функции Φ^{mid-p} , полученного для перестановки π_{mid-p} , от значения целевой функции Φ^* , полученного для перестановки π_{opt} . По значениям Φ и Φ^{mid-p} оценивалось соотношение Φ^{mid-p} / Φ , показывающее преимущества разработанных алгоритмов по сравнению с известными алгоритмами [19].

Вторая часть эксперимента проводилась по тому же плану, что и первая часть, с той лишь разницей, что ежедневно поступало для выполнения 20 новых работ вместо 10 (как в первой части эксперимента), причем с меньшими отрезками возможных длительностей выполнения.

При проведении вычислительного эксперимента были выделены случаи, когда предложенные в настоящей статье алгоритмы оказываются более эффективными по сравнению с известными алгоритмами [18, 19], разработанными для приближенного решения задачи $1 | p_i^L \leq p_i \leq p_i^U | \sum w_i C_i$.

Результаты экспериментальных исследований. В вычислительном эксперименте были сгенерированы и решены серии примеров с единственным блоком работ, а также серии примеров, содержащих случайное число k нефиксированных работ, $(n-k)$ фиксированных работ и случайное количество блоков работ. Веса работ выбирались равновероятно из диапазона [1, 10]. Фиксированные работы распределялись примерно поровну между m блоками работ.

Отрезки допустимых длительностей выполнения работ $[p_i^L, p_i^U]$ определялись из сгенерированных отрезков $[\bar{p}_i^L, \bar{p}_i^U] = [p_i^L / w_i, p_i^U / w_i]$ как $[p_i^L, p_i^U] = [w_i \bar{p}_i^L, w_i \bar{p}_i^U]$.

Генерация отрезков $[\bar{p}_i^L, \bar{p}_i^U]$ для фиксированных работ, содержащихся в блоке B_j , проводилась следующим образом: по заданному ядру блока $[b_j^L, b_j^U]$ генерировались отрезки $[\bar{p}_i^L, \bar{p}_i^U] \supset [b_j^L, b_j^U]$, $\bar{p}_i^L \in [b_j^L - D, b_j^L]$, $\bar{p}_i^U \in [b_j^U, b_j^U + D]$, $D \in \{2, 3, 5, 7, 10, 15\}$.

Генерация отрезков $[\bar{p}_i^L, \bar{p}_i^U]$ для нефиксированных работ проводилась аналогично. Для заданного отрезка $[b_i^L, b_i^U] \supset [b_m^L, b_m^U]$ генерировались отрезки $[\bar{p}_i^L, \bar{p}_i^U] \supset [\bar{b}_i^L, \bar{b}_i^U]$, $\bar{p}_i^L \in [\bar{b}_i^L - D, \bar{b}_i^L]$, $\bar{p}_i^U \in [\bar{b}_i^U, \bar{b}_i^U + D]$, $D \in \{1, 2, 3, 4, 5, 10\}$.

При генерации отрезков $[\bar{p}_i^L, \bar{p}_i^U]$ для примеров с единственным блоком работ для заданных отрезков $[\bar{b}_z^L, \bar{b}_z^U] \supset [b_z^L, b_z^U]$ генерировались отрезки $[\bar{p}_i^L, \bar{p}_i^U] \supset [\bar{b}_z^L, \bar{b}_z^U]$, $\bar{p}_i^L \in [\bar{b}_z^L - D, \bar{b}_z^L]$, $\bar{p}_i^U \in [\bar{b}_z^U, \bar{b}_z^U + D]$, $D \in \{1, 2, 3, 4, 5, 10\}$.

Для генерации фактического сценария использовались законы распределения случайной величины, а именно равномерное распределение, гамма-распределение с параметрами $\alpha = 9$ и $\beta = 2$, гамма-распределение с параметрами $\alpha = 4$ и $\beta = 2$. Закон распределения вероятностей случайной длительности работы выбирался случайным образом из множества трех указанных распределений. Для каждой серии сгенерированных задач вычислялось среднее значение относительных погрешностей γ и γ_{mid-p} .

Каждая серия вычислительного эксперимента состояла из 10 задач. В табл. 1 представлены результаты вычислений первой части, а в табл. 2 – второй части эксперимента. Серии задач в вычислительном эксперименте отличаются одна от другой количеством фиксированных работ и блоков.

Таблица 1

Результаты построения эффективных и фактически оптимальных перестановок
для случайно сгенерированных задач в первой части эксперимента

Число блоков / <i>B</i> /	Количество нефиксированных работ	γ	γ_{mid-p}	Φ^{mid-p} / Φ	Уменьшение относительной погрешности, %
1	–	0,177 112	0,189 004 7	1,000 119	6,292 430 8
2	1	0,088 176	0,108 613	1,000 204	18,816 080
2	2	0,111 966	0,143 524	1,000 316	21,988 040
2	3	0,114 696	0,143 250	1,000 285	19,932 880
2	4	0,186 075	0,204 768	1,000 187	9,129 282
2	5	0,320 890	0,351 293	1,000 303	8,654 531
2	10	0,697 202	0,947 337	1,002 488	26,404 030
3	1	0,095 627	0,151 388	1,000 557	36,832 750
3	2	0,057 943	0,062 483	1,000 046	7,264 948
3	3	0,084 613	0,101 448	1,000 168	16,595 360
3	4	0,112 494	0,195 184	1,000 827	42,365 380
3	5	0,158 733	0,279 461	1,001 217	43,200 360
4	1	0,188 805	0,269 446	1,000 805	29,928 390
4	2	0,288 984	0,473 585	1,001 847	38,979 460
4	3	0,180 573	0,337 540	1,001 573	46,503 200
4	4	0,298 377	0,445 823	1,001 468	33,072 780
4	5	0,257 539	0,384 898	1,001 278	33,088 990
8	1	0,116 846	0,138 202	1,000 213	15,452 890
Среднее значение	–	0,196 481	0,273 736	1,000 772	25,250 100

Таблица 2

Результаты построения эффективных и фактически оптимальных перестановок
для случайно сгенерированных задач во второй части эксперимента

Число блоков / <i>B</i> /	Количество нефиксированных работ	γ	γ_{mid-p}	Φ^{mid-p} / Φ	Уменьшение относительной погрешности, %
1	–	0,423 003	0,461 168	1,000 380	8,275 659 7
2	1	0,087 304	0,103 415	1,000 164	15,578 700
2	2	0,132 425	0,160 933	1,000 285	17,714 160
2	3	0,192 217	0,279 213	1,000 873	31,157 531
2	4	0,428 103	0,444 762	1,000 166	3,745 740
2	5	0,434 013	0,483 155	1,000 492	10,171 060
2	10	0,740 477	0,885 048	1,001 435	16,334 810
3	1	0,058 425	0,087 667	1,000 295	33,355 640
3	2	0,137 929	0,174 655	1,000 371	21,028 180
3	3	0,195 673	0,201 804	1,000 062	3,038 112
3	4	0,432 008	0,549 198	1,001 178	21,338 400
3	5	0,476 762	0,549 237	1,000 729	13,195 710
4	1	0,091 829	0,118 126	1,000 265	22,261 680
4	2	0,087 443	0,131 205	1,000 438	33,354 240
4	3	0,113 506	0,138 308	1,000 248	17,932 620
4	4	0,130 495	0,177 014	1,000 466	26,280 004
4	5	0,152 188	0,169 121	1,000 169	10,012 310
8	1	0,248 513	0,264 660	1,000 162	6,101 085
Среднее значение	–	0,253 462	0,298 816	1,000 454	17,270 870

В перестановках, полученных с использованием частичных перестановок с максимальным значением величины $PerOB(\pi_k)$ для многогранника оптимальности, значение погрешности целевой функции в среднем было на 21 % меньше, чем значение погрешности в перестановках, полученных с использованием частичных перестановок, построенных по известным алгоритмам [18, 19].

Заключение. В работе исследована задача построения оптимальных расписаний выполнения запланированных на месяц работ одним исполнителем, оценивался 30-дневный период составления дневных расписаний. Предполагалось, что каждый день поступает 10 или 20 новых работ для последующего выполнения исполнителем. При планировании работ на один день решалась задача $1 | p_i^L \leq p_i \leq p_i^U | \sum w_i C_i$ составления расписания выполнения одним исполнителем множества запланированных работ с неопределенными длительностями их выполнения с целью минимизации суммы взвешенных моментов завершения выполнения работ.

Приведены алгоритмы построения эффективной перестановки π_k множества работ с максимальным значением относительного полупериметра $PerOB(\pi_k)$, которую предлагается использовать в качестве приближенного решения задачи $1 | p_i^L \leq p_i \leq p_i^U | \sum w_i C_i$, а также алгоритм построения перестановки выполнения всех запланированных работ на основе эффективных перестановок для множества поступивших работ.

Для выбора наиболее важных для пользователя работ и составления оптимального плана (расписания) их выполнения использовался персональный компьютер. Проведенный вычислительный эксперимент на случайно сгенерированных тестовых задачах показал, что применение эффективной перестановки, построенной разработанными программами, обеспечивает погрешность, не превышающую 0,75 % от фактически оптимальной перестановки выполнения работ, полученной при известных длительностях всех запланированных работ. (Фактически оптимальная перестановка может быть построена только после завершения выполнения всех запланированных работ, когда становятся известными длительности выполнения всех запланированных работ.) Значение достигаемой погрешности было в среднем на 21 % меньше по сравнению с погрешностью, полученной при использовании известных алгоритмов приближенного решения задачи $1 | p_i^L \leq p_i \leq p_i^U | \sum w_i C_i$.

Предложенные алгоритмы могут применяться и для решения задач составления производственных расписаний для одного обслуживающего прибора, а также в случае, когда среди множества имеющихся приборов только один прибор является «узким местом» производственного процесса.

Отметим, что при составлении расписания для человека можно не ограничиваться одним критерием оптимальности выполнения запланированных работ. Важных для исполнителя критериев оптимальности может быть несколько в зависимости от его особенностей и специфики самих работ. Поэтому в дальнейшем предполагается применить описанный в настоящей статье подход для случая нескольких критериев оптимальности, упорядочивая их по важности для конкретного исполнителя.

В статье [20] приведены следующие критерии, которые могут использоваться при составлении расписаний для тайм-менеджмента:

- максимизация числа работ, выполненных в течение рабочего дня;
- минимизация общего времени завершения всех работ $C_{\max} = \max\{C_i : J_i \in G\}$;
- минимизация взвешенного числа запаздывающих работ $J_i \in G$ относительно заданных директивных сроков D_i (минимизация значения функции $\sum w_i U_i$, где $U_i = 1$, если завершение работы J_i запаздало, $C_i > D_i$, и $U_i = 0$, если работа J_i выполнена в срок, $C_i \leq D_i$);
- минимизация взвешенного суммарного запаздывания работ $J_i \in G$ относительно директивных сроков D_i , т. е. минимизация значения функции $\sum w_i \max\{0, C_i - D_i\}$;
- минимизация максимального временного смещения моментов C_i завершения работ $J_i \in G$ относительно заданных директивных сроков D_i , т. е. минимизация функции $\max\{C_i - D_i : J_i \in G\}$;

- минимизация числа работ, начатых в течение рабочего дня, но не завершенных к концу рабочего дня;
- минимизация суммы времен переналадок (переходов, переездов и подготовки рабочего места), необходимых для выполнения очередной работы после завершения предыдущей;
- обеспечение чередования в расписании легких и трудных работ из множества G ;
- предпочтение к выполнению сложных работ утром в начале рабочего дня или, наоборот, в конце рабочего дня;
- предпочтение к выполнению наибольшего количества последовательных работ на одном рабочем месте исполнителя;
- минимизация числа прерываний работ (минимизация числа переключений с одной работы на другую, если из-за этого возникают дополнительные потери времени или другие издержки);
- предпочтение к расписаниям с повторяющимися перестановками одних и тех же (или близких по содержанию) работ в течение длительного периода времени (периодичность расписаний).

Список использованных источников

1. The relationship between time management and job performance in event management / N. L. Ahmad [et al.] // *Procedia – Social and Behavioral Sciences*. – 2012. – Vol. 65. – P. 937–941.
2. Eilon, S. Time-management / S. Eilon // *OMEGA Intern. J. of Management Sciences*. – 1993. – Vol. 21, no. 3. – P. 255–259.
3. Sherwood, B. J. Personal time-management allows you to work smarter with less effort / B. J. Sherwood // *Sherwood on Management*. – 2005. – July/August. – P. 44–45.
4. Jackson, V. P. Time management: a realistic approach / V. P. Jackson // *J. of the American College of Radiology*. – 2009. – Vol. 6, no.6. – P. 434–436.
5. Xu, J. Predicting homework time management at the secondary school level: a multilevel analysis / J. Xu // *Learning and Individual Differences*. – 2010. – Vol. 20. – P. 34–39.
6. MacCann, C. Strategies for success in education: time management is more important for part-time than full-time community college students / C. MacCann, G. J. Fogarty, R. D. Roberts // *Learning and Individual Differences*. – 2012. – Vol. 22. – P. 618–623.
7. Assessing time-management skills in terms of age, gender and anxiety levels: a study of nursing and midwifery students in Turkey / H. Kaya [et al.] // *Nurse Education in Practice*. – 2012. – Vol. 12. – P. 284–288.
8. The assessment of time management in middle-school students / O. L. Liu [et al.] // *Personality and Individual Differences*. – 2009. – Vol. 47. – P. 174–179.
9. Zampetakis, L. A. On the relationship between individual creativity and time management / L. A. Zampetakis, N. Bouranta, V. S. Moustakis // *Thinking Skills and Creativity*. – 2010. – Vol. 5. – P. 23–32.
10. Woolfolk, A. E. Time management: an experimental investigation / A. E. Woolfolk, R. L. Woolfolk // *J. of School Psychology*. – 1986. – Vol. 24. – P. 267–275.
11. Indreica, E.-S. Effects of learning styles and time management on academic achievement / E.-S. Indreica, A.-M. Cazan, C. Truta // *Procedia – Social and Behavioral Sciences*. – 2011. – Vol. 30. – P. 1096–1102.
12. Macan, T. Will you remember to read this article later then you have time? The relationship between prospective memory and time management / T. Macan, J.M. Gibson, J. Cunningham // *Personality and Individual Differences*. – 2010. – Vol. 48. – P. 725–730.
13. Ho, B. Time management of final undergraduate English projects: supervisees' and the supervisor's coping strategies / B. Ho // *System*. – 2003. – Vol. 31. – P. 231–245.
14. Kelly, W. E. No time to worry: the relationship between worry, time structure, and time management / W. E. Kelly // *Personality and Individual Differences*. – 2003. – Vol. 35. – P. 1119–1126.
15. Hund, P. M. Representation of time in digital calendars: an argument for a unified, continuous a multi-granular calendar view / P. M. Hund, J. Dowell, K. Mueller // *Intern. J. of Human-Computer Studies*. – 2014. – Vol. 72. – P. 1–11.
16. Smith, W. E. Various optimizers for single-stage production / W. E. Smith // *Naval Research Logistics Quarterly*. – 1956. – Vol. 3, no. 1. – P. 59–66.
17. The optimality box in uncertain data for minimizing the sum of the weighted job completion times / T.-C. Lai [et al.] // *Intern. J. of Production Research*. – 2018. – Vol. 56, no. 19. – P. 6336–6362.
18. Sotskov, Yu. N. Single machine scheduling problem with interval processing times and total completion time objective / Yu. N. Sotskov, N. G. Egorova // *Algorithms*. – 2018. – Vol. 75. – P. 21–40.

19. Allahverdi, A. Single machine scheduling problem with interval processing times to minimize mean weighted completion times / A. Allahverdi, H. Aydilek, A. Aydilek // *Computers & Operations Research*. – 2014. – Vol. 51. – P. 200–207.

20. Модели и комплекс программ для планирования рабочего времени / Ю. Н. Сотсков [и др.] // *Информатика*. – 2007. – № 4. – С. 23–36.

References

1. Ahmad N. L., Yusuf A. N. M., Shobri N. D. M., Wahab S. The relationship between time management and job performance in event management. *Procedia – Social and Behavioral Sciences*, 2012, vol. 65, pp. 937–941.

2. Eilon S. Time-management. *OMEGA International Journal of Management Sciences*, 1993, vol. 21, no 3, pp. 255–259.

3. Sherwood B. J. Personal time-management allows you to work smarter with less effort. *Sherwood on Management*, 2005, July/August, pp. 44–45.

4. Jackson V. P. Time management: a realistic approach. *Journal of the American College of Radiology*, 2009, vol. 6, no 6, pp. 434–436.

5. Xu J. Predicting homework time management at the secondary school level: a multilevel analysis. *Learning and Individual Differences*, 2010, vol. 20, pp. 34–39.

6. MacCann C., Fogarty G. J., Roberts R. D. Strategies for success in education: time management is more important for part-time than full-time community college students. *Learning and Individual Differences*, 2012, vol. 22, pp. 618–623.

7. Kaya H., Kaya N., Palloş A. Ö., Küçük L. Assessing time-management skills in terms of age, gender and anxiety levels: a study of nursing and midwifery students in Turkey. *Nurse Education in Practice*, 2012, vol. 12, pp. 284–288.

8. Liu O. L., Rijmen F., MacCann C., Roberts R. The assessment of time management in middle-school students. *Personality and Individual Differences*, 2009, vol. 47, pp. 174–179.

9. Zampetakis L. A., Bouranta N., Moustakis V. S. On the relationship between individual creativity and time management. *Thinking Skills and Creativity*, 2010, vol. 5, pp. 23–32.

10. Woolfolk A. E., Woolfolk R. L. Time management: an experimental investigation. *Journal of School Psychology*, 1986, vol. 24, pp. 267–275.

11. Indreica E.-S., Cazan A.-M., Truta C. Effects of learning styles and time management on academic achievement. *Procedia – Social and Behavioral Sciences*, 2011, vol. 30, pp. 1096–1102.

12. Macan T., Gibson J. M., Cunningham J. Will you remember to read this article later then you have time? The relationship between prospective memory and time management. *Personality and Individual Differences*, 2010, vol. 48, pp. 725–730.

13. Ho B. Time management of final undergraduate English projects: supervisees' and the supervisor's coping strategies. *System*, 2003, vol. 31, pp. 231–245.

14. Kelly W. E. No time to worry: the relationship between worry, time structure, and time management. *Personality and Individual Differences*, 2003, vol. 35, pp. 1119–1126.

15. Hund P. M., Dowell J., Mueller K. Representation of time in digital calendars: an argument for a unified, continuous a multi-granular calendar view. *International Journal of Human-Computer Studies*, 2014, vol. 72, pp. 1–11.

16. Smith W. E. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 1956, vol. 3, no 1, pp. 59–66.

17. Lai T.-C., Sotskov Yu. N., Egorova N. G., Werner F. The optimality box in uncertain data for minimizing the sum of the weighted job completion times. *International Journal of Production Research*, 2018, vol. 56, no 19, pp. 6336–6362.

18. Sotskov Yu. N., Egorova N. G. Single machine scheduling problem with interval processing times and total completion time objective. *Algorithms*, 2018, vol. 75, pp. 21–40.

19. Allahverdi A., Aydilek H., Aydilek A. Single machine scheduling problem with interval processing times to minimize mean weighted completion times. *Computers & Operations Research*, 2014, vol. 51, pp. 200–207.

20. Sotskov Yu. N., Egorova N. G., Matsveichuck N. M., Petrova E. A. Modeli i complex program dlya planirovaniya rabocheho vremeni [Models and program package for optimal scheduling the working time]. *Informatika [Informatics]*, 2007, no. 4, pp. 23–36 (in Russian).

Информация об авторах

Сотсков Юрий Назарович, доктор физико-математических наук, профессор, главный научный сотрудник лаборатории математической кибернетики, Объединенный институт проблем информатики Национальной академии наук Беларуси, Минск, Беларусь.
E-mail: sotskov48@mail.ru

Егорова Наталья Геннадьевна, кандидат технических наук, старший научный сотрудник лаборатории математической кибернетики, Объединенный институт проблем информатики Национальной академии наук Беларуси, Минск, Беларусь.
E-mail: NataMog@yandex.by

Матвейчук Наталья Михайловна, кандидат физико-математических наук, доцент кафедры автоматизированных систем управления производством, Белорусский государственный аграрный технический университет, Минск, Беларусь.
E-mail: matsveichuk@tut.by

Information about the authors

Yuri N. Sotskov, Dr. Sci. (Phys.-Math.), Professor, Principal Researcher of the Laboratory of Mathematical Cybernetics, the United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus.
E-mail: sotskov48@mail.ru

Natalja G. Egorova, Cand. Sci. (Eng.), Senior Researcher of the Laboratory of Mathematical Cybernetics, the United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus.
E-mail: NataMog@yandex.by

Natalja M. Matsveichuk, Cand. Sci. (Phys.-Math.), Associate Professor of the Department of Automated Production Management Systems, Belarusian State Agrarian Technical University, Minsk, Belarus.
E-mail: matsveichuk@tut.by

ISSN 1816-0301 (Print)
ISSN 2617-6963 (Online)

УДК 517.958:537.8
<https://doi.org/10.37661/1816-0301-2020-17-2-103-119>

Поступила в редакцию 31.12.2019
Received 31.12.2019

Принята к публикации 16.03.2020
Accepted 16.03.2020

Математическая модель проникновения цилиндрических электромагнитных полей с осевой симметрией через плоский экран из пермаллоя

В. Т. Ерофеенко

*Учреждение БГУ «НИИ прикладных проблем математики и информатики»,
Минск, Беларусь
E-mail: bsu_erofeenko@tut.by*

Аннотация. Разработана методика решения краевой задачи проникновения монохроматических электромагнитных полей с осевой симметрией через плоский однослойный экран, выполненный из пермаллоя. Постановка краевой задачи экранирования основывается на использовании системы уравнений Максвелла и дополнительного нелинейного дифференциального уравнения для поля намагниченности, характеризующего пермаллой. Применяются классические граничные условия непрерывности тангенциальных составляющих полей и дополнительные дифференциальные граничные условия для поля намагниченности на лицевых поверхностях экрана. Для упрощения решения задачи в результате исключения величин второго порядка малости, входящих в нелинейное уравнение, нелинейная задача преобразована в линейную. Используются корни (волновые числа) дисперсионного алгебраического уравнения четвертого порядка, которые характеризуют электромагнитные поля с осевой симметрией в слое из пермаллоя. Построены последовательности четырех прямых и четырех обратных электромагнитных волн с осевой симметрией, распространяющихся в противоположных направлениях в слое пермаллоя. Получены двухсторонние граничные условия, связывающие электромагнитные поля по обе стороны экрана. Аналитически вычислены амплитуды отраженного и прошедшего через экран электромагнитных полей.

Ключевые слова: математические модели, двухсторонние граничные условия, краевая задача, задача экранирования, поле намагниченности, дисперсионное уравнение, электромагнитные волны с осевой симметрией, пермаллой, аналитическое моделирование, экран

Благодарности. Работа выполнена в рамках задания 1.1.22 Государственной программы научных исследований «Информатика, космос и безопасность» на 2019–2020 гг.

Для цитирования. Ерофеенко, В. Т. Математическая модель проникновения цилиндрических электромагнитных полей с осевой симметрией через плоский экран из пермаллоя / В. Т. Ерофеенко // Информатика. – 2020. – Т. 17, № 2. – С. 103–119. <https://doi.org/10.37661/1816-0301-2020-17-2-103-119>

Mathematical model of penetration of cylindrical electromagnetic fields with axial symmetry through the plane screen from permalloy

Viktor T. Erofeenko

*Establishment of the Belarusian State University "Research Institute for Applied Problems
of Mathematics and Informatics", Minsk, Belarus
E-mail: bsu_erofeenko@tut.by*

Abstract. The method for solving the boundary-value problem of penetration of monochromatic electromagnetic fields with axial symmetry through the plane screen made from the permalloy is developed. The boundary-value problem is based on the use of differential Maxwell equations and complementary nonlinear differential equation for the field of magnetization, characterizing the permalloy. Classical boundary conditions of continuity

of the tangential components of the fields and complementary boundary conditions for the field of magnetization on the face surfaces of the shield are used. For solution simplification of the boundary-value problem as a result of exclusion value the entering in nonlinear equation second-order infinitesimal, nonlinear task is transformed into linear task. Roots (wave numbers) of a dispersion algebraic equations of four order, which characterizing electromagnetic fields with axial symmetry in layer made from the permalloy, is constructed. The sequences of four forward and four backward counter-propagating electromagnetic waves with axial symmetry in the layer of permalloy is formed. Two-sided boundary conditions connecting electromagnetic fields with axial symmetry on both sides of the screen is constructed. The amplitudes of reflected and transient through the shield electromagnetic fields are calculated.

Keywords: mathematical models, two-sided boundary conditions, boundary-value problem, shielding task, field of magnetization, dispersing equation, electromagnetic waves with axial symmetry, permalloy, analytical modeling, screen

Acknowledgements. The work was performed as part of assignment 1.1.22 of the State program of scientific research "Informatics, space and security" for 2019–2020.

For citation. Erofeenko V. T. Mathematical model of penetration of cylindrical electromagnetic fields with axial symmetry through the plane screen from permalloy. *Informatics*, 2020, vol. 17, no. 2, pp. 103–119 (in Russian). <https://doi.org/10.37661/1816-0301-2020-17-2-103-119>

Введение. Разработка математических методов моделирования распространения излучений электромагнитных волн в композитных материалах является актуальным направлением исследований в математической физике [1]. Анализ таких материалов требует применения специальных математических моделей, адекватно описывающих их электрические и магнитные свойства [2]. Для конструирования электронных технических устройств используются экраны и пленки из композитов. В настоящее время актуально исследование экранирующих свойств пленок из пермаллоя [3]. Материал из пермаллоя обладает свойством намагниченности, которое описывается дополнительным нелинейным дифференциальным уравнением второго порядка для поля намагниченности. С целью упрощения модели нелинейное уравнение намагниченности преобразуется в линейное уравнение [3, 4]. Чтобы обеспечить единственность решения краевой задачи для пермаллоевых материалов, используются дополнительные граничные условия на поверхностях экрана. Для моделирования электродинамического контакта двух материалов на поверхностях раздела сред в задачах экранирования разрабатываются соответствующие граничные условия [5, 6].

В статье предложена методика решения краевой задачи для системы уравнений Максвелла и параболического дифференциального уравнения с частными производными второго порядка для поля намагниченности, в которой описывается экранирование монохроматических электромагнитных полей с круговой симметрией плоским однослойным экраном из пермаллоя. В качестве первичного поля с круговой симметрией, воздействующего на экран, выбрана комбинация базисных TE - и TH -поляризованных полей в цилиндрических координатах с осевой симметрией, распространяющихся ортогонально к экрану. Дополнительно разработаны двухсторонние граничные условия, связывающие поля с круговой симметрией по обе стороны однослойного экрана, которые позволяют построить аналитическое решение исходной задачи экранирования, не используя поля в слое экрана. Двухсторонние граничные условия применяются для моделирования пленочных экранов: однослойных [4, 7, 8] и многослойных [2, 9], со сфероидами частицами [10], в нестационарной электродинамике [11], с упругими свойствами материалов [12]. Разработанная в настоящей статье методика решения краевой задачи экранирования для экрана из пермаллоя аналогична методике, предложенной в статье [13] для биизотропного экрана.

Система электродинамических уравнений. Для моделирования электромагнитного поля \vec{E} , \vec{H} и поля намагниченности \vec{M} , распространяющихся в слое из пермаллоя, используется система дифференциальных уравнений

$$\operatorname{rot} \vec{E} = -\mu_0 \frac{\partial}{\partial t} (\vec{H} + \vec{M}); \quad (1)$$

$$\operatorname{rot} \vec{H} = \delta \vec{E}; \quad (2)$$

$$\frac{1}{\dot{\gamma}} \frac{\partial \vec{M}}{\partial t} = \vec{M} \times (\vec{H} + a \Delta \vec{M} - g \dot{\vec{M}} \times \vec{H}), \quad (3)$$

где постоянные имеют физические размерности $[\dot{\gamma}] = \text{м} / \text{А} \cdot \text{с}$, $[a] = \text{м}^2$, $[g] = \text{м} / \text{А}$, $[\delta] = \text{См} / \text{м}$, $\mu_0 = 4\pi \cdot 10^{-7} \text{ Гн} / \text{м}$; \times – векторное произведение.

Построим решения уравнений (1)–(3) вида [3]

$$\vec{H} = \vec{H}_0 + \vec{H}e^{-i\omega t}, \quad \vec{M} = \vec{M}_0 + \vec{M}e^{-i\omega t}, \quad \vec{E} = \vec{E}e^{-i\omega t}, \quad (4)$$

где $\vec{H}_0 = \dot{H}_0 \vec{e}_z$, $\vec{M}_0 = \dot{M}_0 \vec{e}_z$; $\dot{H}_0, \dot{M}_0 = \text{const}$, $[\dot{H}_0] = [\dot{M}_0] = \text{А} / \text{м}$, $|\vec{H}| \ll \dot{H}_0$, $|\vec{M}| \ll \dot{M}_0$, $\omega = 2\pi f$ – круговая частота поля.

Подставим выражения (4) в (1)–(3) и преобразуем нелинейное уравнение (3) в линейное, пренебрегая величинами второго порядка малости: $\vec{M} \times \vec{H} \approx 0$, $\vec{M} \times (\vec{M} \times \vec{e}_z) \approx 0$.

В результате получим уравнения, которые будем использовать для построения монохроматических полей с осевой симметрией:

$$\operatorname{rot} \vec{E} = i\omega\mu_0 (\vec{H} + \vec{M}); \quad (5)$$

$$\operatorname{rot} \vec{H} = \delta \vec{E}, \quad (6)$$

$$i \frac{\Omega}{\dot{\gamma}} \vec{M} = \dot{H}_0 \vec{e}_z \times \vec{M} - \dot{M}_0 \vec{e}_z \times \vec{H} - a \dot{M}_0 \vec{e}_z \times \Delta \vec{M} - \\ - g \dot{M}_0^2 (\vec{e}_z \times \vec{H}) \times \vec{e}_z + g \dot{H}_0 \dot{M}_0 (\vec{e}_z \times \vec{M}) \times \vec{e}_z,$$

где \vec{E}, \vec{H} – монохроматическое электромагнитное поле, \vec{M} – монохроматическое поле намагниченности.

Исключим из уравнения (5) вектор \vec{E} с помощью (6) и применим оператор div к уравнению (5). Получим систему линейных уравнений для \vec{M} и \vec{H} :

$$\operatorname{grad} \operatorname{div} \vec{H} - \Delta \vec{H} = i\sigma k_0^2 (\vec{H} + \vec{M}); \quad (7)$$

$$\operatorname{div} \vec{H} = -\operatorname{div} \vec{M}; \quad (8)$$

$$i\Omega \vec{M} = \eta \vec{e}_z \times \vec{M} - \vec{e}_z \times \vec{H} - \frac{a}{k_0^2} \vec{e}_z \times \Delta \vec{M} - g (\vec{e}_z \times \vec{H}) \times \vec{e}_z + \eta g (\vec{e}_z \times \vec{M}) \times \vec{e}_z, \quad (9)$$

где σ, η, a, g – безразмерные постоянные, Ω – безразмерная частота, $\sigma = \mu_0 \delta c^2 / \omega$, $\eta = \dot{H}_0 / \dot{M}_0$,

$a = a k_0^2$, $g = g \dot{M}_0$, $\Omega = \frac{\omega}{M_0 \dot{\gamma}}$, $k_0 = \frac{\omega}{c}$, c – скорость света в вакууме.

Пучки цилиндрических электромагнитных волн и полей намагниченности. Построим систему пучков электромагнитных полей с круговой симметрией вида $\Phi_m = \exp(im\varphi)$, $m = 0, 1, 2, \dots$, и временной зависимостью $\exp(-i\omega t)$, $\omega = 2\pi f$, f – частота поля, которые удовлетворяют уравнениям (7)–(9). Для построения воспользуемся базисными цилиндрическими полями [2, с. 139], представив пучки волн в виде разложений

$$\vec{E} = \vec{E}^{(+)}(\vec{\rho}; \lambda) = e_1 \vec{M}_m^{(+)}(\vec{\rho}; \lambda, k) + e_2 \vec{M}_m^{(+2)}(\vec{\rho}; \lambda, k),$$

$$\vec{H} = \vec{H}^{[+]}(\vec{\rho}; \lambda) = h_1 \vec{M}_m^{(+1)}(\vec{\rho}; \lambda, k) + h_2 \vec{M}_m^{(+2)}(\vec{\rho}; \lambda, k) + m_2 \frac{\lambda}{k} J_m(\lambda \rho) \Phi_m F^{(+1)} \vec{e}_z, \quad (10)$$

$$\vec{M} = \vec{M}^{[+]}(\vec{\rho}; \lambda) = m_1 \vec{M}_m^{(+1)}(\vec{\rho}; \lambda, k) + m_2 \vec{M}_m^{(+2)}(\vec{\rho}; \lambda, k) - m_2 \frac{\lambda}{k} J_m(\lambda \rho) \Phi_m F^{(+1)} \vec{e}_z;$$

$$\vec{E} = \vec{E}^{[-]}(\vec{\rho}; \lambda) = e_1 \vec{M}_m^{(-1)}(\vec{\rho}; \lambda, k) - e_2 \vec{M}_m^{(-2)}(\vec{\rho}; \lambda, k),$$

$$\vec{H} = \vec{H}^{[-]}(\vec{\rho}; \lambda) = h_1 \vec{M}_m^{(-1)}(\vec{\rho}; \lambda, k) - h_2 \vec{M}_m^{(-2)}(\vec{\rho}; \lambda, k) - m_2 \frac{\lambda}{k} J_m(\lambda \rho) \Phi_m F^{(-1)} \vec{e}_z, \quad (11)$$

$$\vec{M} = \vec{M}^{[-]}(\vec{\rho}; \lambda) = m_1 \vec{M}_m^{(-1)}(\vec{\rho}; \lambda, k) - m_2 \vec{M}_m^{(-2)}(\vec{\rho}; \lambda, k) + m_2 \frac{\lambda}{k} J_m(\lambda \rho) \Phi_m F^{(-1)} \vec{e}_z,$$

где e_j, h_j, m_j ($j=1, 2$), k, ν – величины, подлежащие определению; λ – произвольный параметр, $\vec{\rho} = (\rho, \varphi, z)$ – цилиндрические координаты;

$$\vec{M}_m^{(\pm 1)} = \vec{M}_m^{(\pm 1)}(\vec{\rho}; \lambda, k) = \vec{V}_m^{(1)}(\lambda \rho) \Phi_m e^{\pm \nu z}, \quad m = 0, \pm 1, \dots, \quad (12)$$

$$\vec{M}_m^{(\pm 2)} = \vec{M}_m^{(\pm 2)}(\vec{\rho}; \lambda, k) = \frac{1}{k} \left(\pm \nu \vec{V}_m^{(2)}(\lambda \rho) + \lambda J_m(\lambda \rho) \vec{e}_z \right) \Phi_m e^{\pm \nu z},$$

$$\nu = \sqrt{\lambda^2 - k^2}, \quad -\frac{\pi}{2} \leq \arg \nu < \frac{\pi}{2}, \quad F^{(\pm)} = e^{\pm \nu z}, \quad \Phi_m = e^{im\varphi};$$

$$\vec{V}_m^{(1)}(\lambda \rho) = \frac{im}{\lambda \rho} J_m(\lambda \rho) \vec{e}_\rho - J'_m(\lambda \rho) \vec{e}_\varphi, \quad \vec{V}_m^{(2)}(\lambda \rho) = J'_m(\lambda \rho) \vec{e}_\rho + \frac{im}{\lambda \rho} J_m(\lambda \rho) \vec{e}_\varphi, \quad (13)$$

где $0 \leq \lambda < \infty$; $\vec{e}_\rho, \vec{e}_\varphi, \vec{e}_z$ – орты цилиндрической системы координат, $J_m(\cdot)$ – функции Бесселя, $c = 1/\sqrt{\varepsilon_0 \mu_0}$.

Теорема. Система электромагнитных волн и волн намагниченности с осевой симметрией $\vec{E}^{[\pm s]}(\vec{\rho}; \lambda), \vec{H}^{[\pm s]}(\vec{\rho}; \lambda), \vec{M}^{[\pm s]}(\vec{\rho}; \lambda)$ ($s = 1, 2, 3, 4$), распространяющихся в среде из пермаллоя, задается формулами (10), (11), для которых амплитуды определяются аналитически в виде

$$m_1 = m_1^{(s)} = -K_{12}^{(s)} = -g_0 \frac{\bar{\nu}^{(s)}}{K_s} (g\Omega_k^{(s)} + \Omega_0), \quad (14)$$

$$m_2 = m_2^{(s)} = K_{11}^{(s)} = L_1^{(s)} + g_0 (g\Omega_0 - \Omega_k^{(s)});$$

$$h_1 = h_1^{(s)} = L_1^{(s)} m_1^{(s)}, \quad h_2 = h_2^{(s)} = L_2^{(s)} m_2^{(s)}; \quad (15)$$

$$e_1 = e_1^{(s)} = \dot{Z} (K_s L_2^{(s)} + \frac{\bar{\lambda}^2}{K_s}) m_2^{(s)}, \quad e_2 = e_2^{(s)} = \dot{Z} K_s L_1^{(s)} m_1^{(s)}, \quad (16)$$

где

$$k = k^{(s)} = k_0 K_s, \quad \bar{\nu} = \frac{\nu}{k_0} = \bar{\nu}^{(s)} = \sqrt{\bar{\lambda}^2 - K_s^2}, \quad -\frac{\pi}{2} \leq \arg \bar{\nu}^{(s)} < \frac{\pi}{2}, \quad L_1 = L_1^{(s)} = \frac{i\sigma}{K_s^2 - i\sigma}, \quad (17)$$

$$L_2 = L_2^{(s)} = \frac{i\sigma - \bar{\lambda}^2}{K_s^2 - i\sigma}, \quad 0 \leq \bar{\lambda} = \frac{\lambda}{k_0} < \infty, \quad \dot{Z} = \frac{k_0}{\dot{\sigma}}, \quad [\dot{Z}] = \text{Om}, \quad F^{(\pm)} = F_s^{(\pm)}(z) = e^{\pm \nu^{(s)} z}.$$

Волновые числа K_s определяются из дисперсионного уравнения

$$a_8 K^8 + a_6 K^6 + a_4 K^4 + a_2 K^2 + a_0 = 0, \quad (18)$$

где

$$\begin{aligned} a_8 &= a^2, a_6 = 2(\eta a - i\sigma a^2), a_4 = \Omega_0^2 + \eta^2 - a l_2 - a^2 \sigma^2 - 4i\sigma \eta a, \Omega_0 = i\Omega - g\eta, \\ a_2 &= (i\sigma a + g\Omega_0 - \eta)l_2 - 2\eta a \sigma^2 - 2i\sigma(\Omega_0^2 + \eta^2), \\ a_0 &= l_1(g^2 + 1) - i\sigma(g\Omega_0 - \eta)l_2 - \sigma^2(\Omega_0^2 + \eta^2), \quad l_1 = -\sigma(\sigma + i\bar{\lambda}^2), \quad l_2 = 2i\sigma - \bar{\lambda}^2. \end{aligned} \quad (19)$$

Восемь корней алгебраического уравнения (18) задаются формулами $K = \pm K_s$, $s = 1, 2, 3, 4$,

$$-\frac{\pi}{2} < \arg K_s \leq \frac{\pi}{2}.$$

Доказательство. Доказательство теоремы осуществим на примере полей (10). Заметим, что уравнение (8) для полей (10) выполнено автоматически. Действительно, применяя формулы [2, с. 131, 291], получим равенства

$$\begin{aligned} \operatorname{div} \vec{M}_m^{(+1)}(\vec{\rho}; \lambda, k) = 0, \operatorname{div} \vec{M}_m^{(+2)}(\vec{\rho}; \lambda, k) = 0, \operatorname{div}(J_m \Phi_m F \vec{e}_z) = \nu J_m \Phi_m F, F = e^{\nu z}, \\ \operatorname{div} \vec{H} = m_2(\lambda) \frac{\lambda}{k} \nu J_m(\lambda \rho) \Phi_m F, \operatorname{div} \vec{M} = -m_2(\lambda) \frac{\lambda}{k} \nu J_m(\lambda \rho) \Phi_m F. \end{aligned} \quad (20)$$

В результате тождество для уравнения (8) выполнено.

Удовлетворим уравнению (6), подставляя в него поля (10). Учитывая выражения для базисных векторов (13) и формулы

$$\operatorname{rot} \vec{M}_m^{(+1)} = k \vec{M}_m^{(+2)}, \operatorname{rot} \vec{M}_m^{(+2)} = k \vec{M}_m^{(+1)}, \quad (21)$$

$$\operatorname{rot}(J_m(\lambda \rho) \Phi_m F \vec{e}_z) = \left(\frac{im}{\rho} J_m(\lambda \rho) \vec{e}_\rho - \lambda J'_m(\lambda \rho) \vec{e}_\varphi \right) \Phi_m F = \lambda \vec{V}_m^{(1)} \Phi_m F,$$

получим равенство

$$\operatorname{rot} \vec{H} = h_1 k \vec{M}_m^{(+2)} + h_2 k \vec{M}_m^{(+1)} + m_2 \frac{\lambda^2}{k} \vec{V}_m^{(1)} \Phi_m F = \dot{\sigma} (e_1 \vec{M}_m^{(+1)} + e_2 \vec{M}_m^{(+2)}).$$

Учитывая разложения (12) и приравнявая коэффициенты при базисных векторах $\vec{V}_m^{(1)}$, $\vec{V}_m^{(2)}$, \vec{e}_z , придем к системе алгебраических уравнений

$$e_1 = \frac{1}{\dot{\sigma}} (k h_2 + \frac{\lambda^2}{k} m_2), e_2 = \frac{k}{\dot{\sigma}} h_1. \quad (22)$$

Удовлетворим уравнению (7), подставляя в него поля (10). Воспользуемся классическим тождеством [2, с. 291] и формулами (21):

$$\begin{aligned} \operatorname{grad} \operatorname{div} \vec{H} - \Delta \vec{H} = \operatorname{rot} \operatorname{rot} \vec{H} = k^2 (h_1 \vec{M}_m^{(+1)} + h_2 \vec{M}_m^{(+2)}) + m_2(\lambda) \frac{\lambda}{k} \operatorname{rot} \operatorname{rot}(J_m(\lambda \rho) \Phi_m F \vec{e}_z) = \\ = k^2 (h_1 \vec{M}_m^{(+1)} + h_2 \vec{M}_m^{(+2)}) + m_2 \frac{\lambda}{k} [\nu \lambda J'_m(\lambda \rho) \vec{e}_\rho + \frac{im\nu}{\rho} J_m(\lambda \rho) \vec{e}_\varphi + \\ + (\frac{m^2}{\rho^2} J_m(\lambda \rho) - \frac{1}{\rho} \frac{\partial}{\partial \rho} (\rho \frac{\partial J_m(\lambda \rho)}{\partial \rho})) \vec{e}_z] \Phi_m F. \end{aligned}$$

Учтем дифференциальное уравнение для функции Бесселя

$$\frac{1}{x} \frac{\partial}{\partial x} \left(x \frac{\partial J_m(x)}{\partial x} \right) + \left(1 - \frac{m^2}{x^2} \right) J_m(x) = 0$$

и с помощью представлений (12), (13) получим аналитическую формулу

$$\begin{aligned} \text{graddiv } \vec{H} - \Delta \vec{H} = \text{rotrot } \vec{H} &= [k^2 (h_1 \vec{V}_m^{(1)} + h_2 \frac{1}{k} (v \vec{V}_m^{(2)} + \lambda J_m(\lambda \rho) \vec{e}_z)) + \\ &+ m_2 \frac{\lambda^2}{k} (v \vec{V}_m^{(2)} + \lambda J_m(\lambda \rho) \vec{e}_z)] \Phi_m F. \end{aligned} \quad (23)$$

С учетом формулы (23) уравнение (7) преобразуем к виду

$$\begin{aligned} \text{graddiv } \vec{H} - \Delta \vec{H} &= [k^2 (h_1 \vec{V}_m^{(1)} + h_2 \frac{1}{k} (v \vec{V}_m^{(2)} + \lambda J_m \vec{e}_z)) + \\ &+ m_2 \frac{\lambda^2}{k} (v \vec{V}_m^{(2)} + \lambda J_m \vec{e}_z)] \Phi_m F = i \sigma k_0^2 [(h_1 + m_1) \vec{V}_m^{(1)} + (h_2 + m_2) \frac{1}{k} (v \vec{V}_m^{(2)} + \lambda J_m \vec{e}_z)] \Phi_m F. \end{aligned}$$

Приравняем коэффициенты в этом тождестве при базисных векторах $\vec{V}_m^{(1)}$, $\vec{V}_m^{(2)}$, \vec{e}_z и придем к системе алгебраических уравнений

$$k^2 h_1 = i \sigma k_0^2 (h_1 + m_1), \quad k^2 h_2 + \lambda^2 m_2 = i \sigma k_0^2 (h_2 + m_2)$$

или

$$h_1 = L_1 m_1, \quad h_2 = L_2 m_2, \quad (24)$$

где

$$L_1 = \frac{i \sigma}{K^2 - i \sigma}, \quad L_2 = \frac{i \sigma - \bar{\lambda}^2}{K^2 - i \sigma}, \quad K = \frac{k}{k_0}, \quad \bar{\lambda} = \frac{\lambda}{k_0}.$$

Запишем систему уравнений (24) в матричном виде

$$\hat{h} = \hat{L} \vec{m}, \quad \vec{h} = \begin{pmatrix} h_1 \\ h_2 \end{pmatrix}, \quad \vec{m} = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix}, \quad \hat{L} = \begin{pmatrix} L_1 & 0 \\ 0 & L_2 \end{pmatrix}. \quad (25)$$

Уравнение (9) также преобразуем к матричному виду. Воспользуемся формулой (23), учитывая одинаковое аналитическое представление векторов \vec{H} и \vec{M} :

$$\begin{aligned} \text{graddiv } \vec{M} - \Delta \vec{M} = \text{rotrot } \vec{M} &= [k^2 (m_1 \vec{V}_m^{(1)} + m_2 \frac{1}{k} (v \vec{V}_m^{(2)} + \lambda J_m(\lambda \rho) \vec{e}_z)) - \\ &- m_2 \frac{\lambda^2}{k} (v \vec{V}_m^{(2)} + \lambda J_m(\lambda \rho) \vec{e}_z)] \Phi_m F. \end{aligned} \quad (26)$$

Определим действие оператора graddiv на поле намагничности \vec{M} . Из формул (20) следует выражение

$$\text{graddiv } \vec{M} = -m_2 \frac{\lambda v}{k} \text{grad}(J_m(\lambda \rho) \Phi_m F) = -m_2 \frac{\lambda v}{k} (\lambda \vec{V}_m^{(2)} + v J_m(\lambda \rho) \vec{e}_z) \Phi_m F. \quad (27)$$

Вычислим действие оператора Δ на поле \vec{M} . Учитывая формулы (26), (27), получим выражение

$$\begin{aligned} \Delta \vec{M} = \text{graddiv } \vec{M} - \text{rotrot } \vec{M} = & -m_2 \frac{\lambda v}{k} (\lambda \vec{V}_m^{(2)} + v J_m(\lambda \rho) \vec{e}_z) \Phi_m F - \\ & - [k^2 (m_1 \vec{V}_m^{(1)} + m_2 \frac{1}{k} (v \vec{V}_m^{(2)} + \lambda J_m(\lambda \rho) \vec{e}_z)) - m_2 \frac{\lambda^2}{k} (v \vec{V}_m^{(2)} + \lambda J_m(\lambda \rho) \vec{e}_z)] \Phi_m F. \end{aligned} \quad (28)$$

Применяя правила векторных произведений базисных векторов $\vec{e}_\rho, \vec{e}_\phi, \vec{e}_z$, получим формулы

$$\begin{aligned} \vec{e}_z \times \vec{V}_m^{(1)} = \vec{V}_m^{(2)}, \quad \vec{e}_z \times \vec{V}_m^{(2)} = -\vec{V}_m^{(1)}, \\ \vec{e}_z \times \vec{M}_m^{(+1)} = \vec{V}_m^{(2)} \Phi_m F, \quad \vec{e}_z \times \vec{M}_m^{(+2)} = -\frac{v}{k} \vec{V}_m^{(1)} \Phi_m F. \end{aligned} \quad (29)$$

Из равенства (28) с учетом формул (29) следует соотношение

$$\vec{e}_z \times \Delta \vec{M} = k(m_2 v \vec{V}_m^{(1)} - m_1 k \vec{V}_m^{(2)}) \Phi_m F. \quad (30)$$

Удовлетворим уравнению (9), подставляя поля (10), которые предварительно разложим на составляющие (12). Учитывая формулы (29), (30), преобразуем уравнение (9) в тождество

$$\begin{aligned} i\Omega(m_1 \vec{V}_m^{(1)} + m_2 \frac{v}{k} \vec{V}_m^{(2)}) = \eta(m_1 \vec{V}_m^{(2)} - m_2 \frac{v}{k} \vec{V}_m^{(1)}) - h_1 \vec{V}_m^{(2)} + h_2 \frac{v}{k} \vec{V}_m^{(1)} - \\ - \frac{ak}{k_0^2} (m_2 v \vec{V}_m^{(1)} - m_1 k \vec{V}_m^{(2)}) - g(h_1 \vec{V}_m^{(1)} + h_2 \frac{v}{k} \vec{V}_m^{(2)}) + g \eta(m_1 \vec{V}_m^{(1)} + m_2 \frac{v}{k} \vec{V}_m^{(2)}). \end{aligned}$$

Приравняем коэффициенты при базисных векторах $\vec{V}_m^{(1)}, \vec{V}_m^{(2)}$ и получим систему алгебраических уравнений

$$\Omega_0 m_1 + \frac{\bar{v}}{K} \Omega_k m_2 = -g h_1 + \frac{\bar{v}}{K} h_2, \quad -\Omega_k m_1 + \frac{\bar{v}}{K} \Omega_0 m_2 = -h_1 - g \frac{\bar{v}}{K} h_2,$$

где $\Omega_0 = i\Omega - g\eta$, $\Omega_k = \eta + aK^2$, величина $K = k/k_0$ подлежит определению.

Предыдущие уравнения запишем в матричном виде

$$\hat{\Omega} \vec{m} = -\hat{V} \hat{h}$$

или в виде

$$\hat{h} = \hat{G} \vec{m}, \quad (31)$$

где

$$\begin{aligned} \hat{\Omega} = \begin{pmatrix} \Omega_0, & \frac{\bar{v}}{K} \Omega_k \\ -\Omega_k, & \frac{\bar{v}}{K} \Omega_0 \end{pmatrix}, \quad \hat{V} = \begin{pmatrix} g, & -\frac{\bar{v}}{K} \\ 1, & g \frac{\bar{v}}{K} \end{pmatrix}, \\ \hat{G} = -\hat{V}^{-1} \hat{\Omega} = -g_0 \begin{pmatrix} g\Omega_0 - \Omega_k, & \frac{\bar{v}}{K} (g\Omega_k + \Omega_0) \\ -\frac{K}{\bar{v}} (g\Omega_k + \Omega_0), & g\Omega_0 - \Omega_k \end{pmatrix}, \quad \hat{V}^{-1} = g_0 \frac{K}{\bar{v}} \begin{pmatrix} g \frac{\bar{v}}{K}, & \frac{\bar{v}}{K} \\ -1, & g \end{pmatrix} - \end{aligned}$$

обратная матрица, $g_0 = \frac{1}{g^2 + 1}$, $\bar{v} = \sqrt{\lambda^2 - K^2}$, $-\frac{\pi}{2} \leq \arg \bar{v} < \frac{\pi}{2}$, $K = \frac{k}{k_0}$, $\bar{\lambda} = \frac{\lambda}{k_0}$.

Сравним матричные равенства (25), (31):

$$\hat{K}\bar{m} = 0, \quad (32)$$

$$\hat{K} = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} = \hat{L} - \hat{G} = \begin{pmatrix} L_1 + g_0(g\Omega_0 - \Omega_k), & g_0 \frac{\bar{v}}{K}(g\Omega_k + \Omega_0) \\ -g_0 \frac{K}{\bar{v}}(g\Omega_k + \Omega_0), & L_2 + g_0(g\Omega_0 - \Omega_k) \end{pmatrix}.$$

Нетривиальное решение уравнения (32) существует, когда определитель матрицы \hat{K} равен нулю: $|\hat{K}| = K_{11}K_{22} - K_{12}K_{21} = 0$. Из уравнения (32) определяются амплитуды вектора \bar{M} (10): $m_1 = -K_{12}$, $m_2 = K_{11}$.

Вычислим определитель $|\hat{K}|$, используя зависимости от параметра K величин L_j , Ω_0 и Ω_k :

$$\begin{aligned} |\hat{K}| &= L_1L_2 + (L_1 + L_2)g_0(g\Omega_0 - \Omega_k) + g_0^2(g\Omega_0 - \Omega_k)^2 + g_0^2(g\Omega_k + \Omega_0)^2 = \\ &= \frac{l_1}{(K^2 - i\sigma)^2} + \frac{l_2}{(K^2 - i\sigma)} g_0(g\Omega_0 - \Omega_k) + g_0(\Omega_0^2 + \Omega_k^2) = 0. \end{aligned} \quad (33)$$

Умножим уравнение (33) на $(K^2 - i\sigma)^2$ и получим дисперсионное уравнение (18) для параметра K с коэффициентами (19).

Определим корни K_s , $s = 1, 2, 3, 4$, уравнения (18) и подставим в формулы для величин (17). Вычислим амплитуды (14)–(16) для четырех прямых волн $\vec{E}^{[+s]}(\vec{\rho}; \lambda)$, $\vec{H}^{[+s]}(\vec{\rho}; \lambda)$, $\vec{M}^{[+s]}(\vec{\rho}; \lambda)$ (10) и четырех обратных волн $\vec{E}^{[-s]}(\vec{\rho}; \lambda)$, $\vec{H}^{[-s]}(\vec{\rho}; \lambda)$, $\vec{M}^{[-s]}(\vec{\rho}; \lambda)$ (11). Из уравнения (32) найдем амплитуды вектора \vec{M} : $m_1 = -K_{12}^{(s)}$, $m_2 = K_{11}^{(s)}$. Далее, учитывая формулы (24), (22), определим амплитуды $h_j^{(s)}$, $e_j^{(s)}$ полей \vec{H} , \vec{E} . В результате получим требуемые формулы (14)–(16). ■

Постановка краевой задачи экранирования. В пространстве E^3 с электрической и магнитной постоянными ϵ_0 , μ_0 расположен плоский экран $D(0 < z < \Delta)$ толщиной Δ , наполненный пермаллоем. Из полупространства $D_1(z < 0)$ на слой D воздействует первичное электромагнитное поле $\vec{E}_0(\vec{\rho}; \lambda)$, $\vec{H}_0(\vec{\rho}; \lambda)$ с круговой симметрией вида $\Phi_m = \exp(im\varphi)$, $m = 0, 1, 2, \dots$, и временной зависимостью $\exp(-i\omega t)$, $\omega = 2\pi f$ [2, с. 131]:

$$\begin{aligned} \vec{E}_0 &= \vec{E}_0(\vec{\rho}; \lambda) = A(\lambda)\vec{M}_m^{(-1)}(\vec{\rho}; \lambda, k_0) + B(\lambda)\vec{M}_m^{(-2)}(\vec{\rho}; \lambda, k_0), \\ \vec{H}_0 &= \vec{H}_0(\vec{\rho}; \lambda) = h_0((A(\lambda)\vec{M}_m^{(-2)}(\vec{\rho}; \lambda, k_0) + B(\lambda)\vec{M}_m^{(-1)}(\vec{\rho}; \lambda, k_0)), \end{aligned} \quad (34)$$

где $A(\lambda)$, $B(\lambda)$ – заданные амплитуды, $h_0 = \frac{1}{iZ_0}$, $Z_0 = \sqrt{\frac{\mu_0}{\epsilon_0}} = 377 \frac{\text{кг} \cdot \text{м}^2}{\text{с}^3 \cdot \text{А}^2}$, $\vec{\rho} = (\rho, \varphi, z)$ – цилиндрические координаты,

$$\vec{M}_m^{(\pm 1)}(\vec{\rho}; \lambda, k_0) = \vec{V}_m^{(1)}(\lambda \rho) \Phi_m e^{\pm \nu_0 z}, \quad (35)$$

$$\vec{M}_m^{(\pm 2)}(\vec{\rho}; \lambda, k_0) = \frac{1}{k_0} \left(\pm \nu_0 \vec{V}_m^{(2)}(\lambda \rho) + \lambda J_m(\lambda \rho) \vec{e}_z \right) \Phi_m e^{\pm \nu_0 z},$$

$$\nu_0 = \sqrt{\lambda^2 - k_0^2}, \quad -\frac{\pi}{2} \leq \arg \nu_0 < \frac{\pi}{2}, \quad F_0^{(\pm)}(z) = e^{\pm \nu_0 z}.$$

Краевая задача. Для заданного поля (34) требуется определить поля $\vec{E}'_1, \vec{H}'_1; \vec{E}_2, \vec{H}_2$ соответственно в областях D_1, D_2 и поля $\vec{E}, \vec{H}, \vec{M}$ в области D , которые удовлетворяют:

– уравнениям Максвелла

$$\text{rot } \vec{E}'_1 = i\omega \mu_0 \vec{H}'_1, \quad \text{rot } \vec{H}'_1 = -i\omega \varepsilon_0 \vec{E}'_1, \quad z < 0, \quad (36)$$

$$\text{rot } \vec{E}_2 = i\omega \mu_0 \vec{H}_2, \quad \text{rot } \vec{H}_2 = -i\omega \varepsilon_0 \vec{E}_2, \quad z > \Delta;$$

$$\text{rot } \vec{E} = i\omega \mu_0 (\vec{H} + \vec{M}), \quad \text{rot } \vec{H} = \dot{\sigma} \vec{E}, \quad 0 < z < \Delta; \quad (37)$$

$$(i\Omega - \eta g) \vec{M} = \eta [\vec{n}, \vec{M}] - [\vec{n}, \vec{H}] - \frac{a}{k_0^2} [\vec{n}, \Delta \vec{M}] - g \vec{H}_\tau, \quad (38)$$

где $\vec{n} = \vec{e}_z$, $\vec{H}_\tau = (\vec{n} \times \vec{H}) \times \vec{n} = [[\vec{n}, \vec{H}], \vec{n}]$;

– граничным условиям сопряжения на плоскости $\Gamma_1 (z=0)$

$$(\vec{E}_\tau - \vec{E}_{1\tau})|_{z=0} = 0, \quad (\vec{H}_\tau - \vec{H}_{1\tau})|_{z=0} = 0; \quad (39)$$

$$\left(\frac{d\vec{M}}{k_0 dz} + p\vec{M} \right) \Big|_{z=0} = 0; \quad (40)$$

– граничным условиям сопряжения на плоскости $\Gamma_2 (z=\Delta)$

$$(E_\tau - E_{2\tau})|_{z=\Delta} = 0, \quad (H_\tau - H_{2\tau})|_{z=\Delta} = 0; \quad (41)$$

$$\left(\frac{d\vec{M}}{k_0 dz} - q\vec{M} \right) \Big|_{z=\Delta} = 0 \quad (42)$$

и условиям излучения на бесконечности.

Решение краевой задачи. Решение задачи (36)–(42) представим в виде суперпозиции базисных полей (35) в областях D_1, D_2 и базисных полей (10), (11) в слое D для волновых чисел $k = k^{(s)}$:

$$\vec{E}'_1 = x_1 \vec{M}_m^{(+1)}(\vec{\rho}; \lambda, k_0) + x_2 \vec{M}_m^{(+2)}(\vec{\rho}; \lambda, k_0), \quad (43)$$

$$\vec{H}'_1 = h_0 \left(x_1 \vec{M}_m^{(+2)}(\vec{\rho}; \lambda, k_0) + x_2 \vec{M}_m^{(+1)}(\vec{\rho}; \lambda, k_0) \right), \quad z < 0;$$

$$\vec{E}_2 = y_1 \vec{M}_m^{(-1)}(\vec{\rho}; \lambda, k_0) + y_2 \vec{M}_m^{(-2)}(\vec{\rho}; \lambda, k_0), \quad (44)$$

$$\vec{H}_2 = h_0 \left(y_1 \vec{M}_m^{(-2)}(\vec{\rho}; \lambda, k_0) + y_2 \vec{M}_m^{(-1)}(\vec{\rho}; \lambda, k_0) \right), \quad z > \Delta;$$

$$\vec{E} = \sum_{s=1}^4 \left(z_s \vec{E}^{[+s]}(\vec{\rho}; \lambda) + z_{-s} \vec{E}^{[-s]}(\vec{\rho}; \lambda) \right), \quad \vec{H} = \sum_{s=1}^4 \left(z_s \vec{H}^{[+s]}(\vec{\rho}; \lambda) + z_{-s} \vec{H}^{[-s]}(\vec{\rho}; \lambda) \right), \quad (45)$$

$$\vec{M} = \sum_{s=1}^4 \left(z_s \vec{M}^{[+s]}(\vec{\rho}; \lambda) + z_{-s} \vec{M}^{[-s]}(\vec{\rho}; \lambda) \right), \quad 0 < z < \Delta,$$

где поля (43)–(45) удовлетворяют уравнениям (36)–(38); коэффициенты x_j, y_j, z_s, z_{-s} ($j=1, 2, s=1, 2, 3, 4$) подлежат определению из граничных условий (39)–(42).

Удовлетворим граничному условию (40), подставляя (45) в (40) с учетом формул (10), (11) при $k = k^{(s)}$:

$$\begin{aligned} \left. \left(\frac{d\vec{M}}{k_0 dz} + p\vec{M} \right) \right|_{z=0} &= \sum_{s=1}^4 \left[z_s \left(\frac{d\vec{M}^{[+s]}}{k_0 dz} + p\vec{M}^{[+s]} \right) + z_{-s} \left(\frac{d\vec{M}^{[-s]}}{k_0 dz} + p\vec{M}^{[-s]} \right) \right] \Big|_{z=0} = \\ &= \sum_{s=1}^4 \left[z_s \left(\frac{d}{k_0 dz} \left(m_1^{(s)} \vec{M}_m^{(+1)} + m_2^{(s)} \vec{M}_m^{(+2)} - m_2^{(s)} \frac{\lambda}{k^{(s)}} J_m \Phi_m \vec{e}_z \right) + \right. \right. \\ &\quad \left. \left. + p \left(m_1^{(s)} \vec{M}_m^{(+1)} + m_2^{(s)} \vec{M}_m^{(+2)} - m_2^{(s)} \frac{\lambda}{k^{(s)}} J_m \Phi_m \vec{e}_z \right) \right) + \right. \\ &\quad \left. + z_{-s} \left(\frac{d}{k_0 dz} \left(m_1^{(s)} \vec{M}_m^{(-1)} - m_2^{(s)} \vec{M}_m^{(-2)} + m_2^{(s)} \frac{\lambda}{k^{(s)}} J_m \Phi_m \right) + \right. \right. \\ &\quad \left. \left. + p \left(m_1^{(s)} \vec{M}_m^{(-1)} - m_2^{(s)} \vec{M}_m^{(-2)} + m_2^{(s)} \frac{\lambda}{k^{(s)}} J_m \Phi_m \vec{e}_z \right) \right) \right] \Big|_{z=0} = 0. \end{aligned}$$

Учитывая формулы (12) для базисных полей при $k = k^{(s)}$, запишем уравнение

$$\begin{aligned} \left. \left(\frac{d\vec{M}}{k_0 dz} + p\vec{M} \right) \right|_{z=0} &= \sum_{s=1}^4 \left[z_s \left(\frac{v^{(s)}}{k_0} \left(m_1^{(s)} \vec{V}_m^{(1)} + m_2^{(s)} \frac{v^{(s)}}{k^{(s)}} \vec{V}_m^{(2)} \right) + p \left(m_1^{(s)} \vec{V}_m^{(1)} + m_2^{(s)} \frac{v^{(s)}}{k^{(s)}} \vec{V}_m^{(2)} \right) \right) + \right. \\ &\quad \left. + z_{-s} \left(\frac{v^{(s)}}{k_0} \left(m_1^{(s)} \vec{V}_m^{(1)} + m_2^{(s)} \frac{v^{(s)}}{k^{(s)}} \vec{V}_m^{(2)} \right) + p \left(m_1^{(s)} \vec{V}_m^{(1)} + m_2^{(s)} \frac{v^{(s)}}{k^{(s)}} \vec{V}_m^{(2)} \right) \right) \right] \Phi_m = 0. \end{aligned}$$

Приравнивая коэффициенты в данном уравнении при базисных векторах $\vec{V}_m^{(1)}, \vec{V}_m^{(2)}$, придем к соотношениям

$$\sum_{s=1}^4 \left[z_s (\bar{v}^{(s)} + p) - z_{-s} (\bar{v}^{(s)} - p) \right] m_1^{(s)} = 0, \quad \sum_{s=1}^4 \left[z_s (\bar{v}^{(s)} + p) - z_{-s} (\bar{v}^{(s)} - p) \right] \frac{\bar{v}^{(s)}}{K_s} m_2^{(s)} = 0. \quad (46)$$

Аналогично из граничного условия (42) получим соотношения

$$\begin{aligned} \sum_{s=1}^4 \left[z_s (\bar{v}^{(s)} - q) F_s^{(+)} - z_{-s} (\bar{v}^{(s)} + q) F_s^{(-)} \right] m_1^{(s)} &= 0, \\ \sum_{s=1}^4 \left[z_s (\bar{v}^{(s)} - q) F_s^{(+)} - z_{-s} (\bar{v}^{(s)} - q) F_s^{(-)} \right] \frac{\bar{v}^{(s)}}{K_s} m_2^{(s)} &= 0. \end{aligned} \quad (47)$$

Запишем касательные составляющие первичного поля (34) и отраженного поля (43) на плоскости $\Gamma_1(z=0)$ в базисе $\vec{V}_m^{(1)}, \vec{V}_m^{(2)}$:

$$\begin{aligned}\vec{E}_{0\tau}(\lambda)\Big|_{z=0} &= (E_{0v_1}(\lambda)\vec{V}_m^{(1)} + E_{0v_2}(\lambda)\vec{V}_m^{(2)})\Phi_m, \quad \vec{H}_{0\tau}(\lambda)\Big|_{z=0} = (H_{0v_1}(\lambda)\vec{V}_m^{(1)} + H_{0v_2}(\lambda)\vec{V}_m^{(2)})\Phi_m; \\ \vec{E}'_{1\tau}(\lambda)\Big|_{z=0} &= (E'_{1v_1}(\lambda)\vec{V}_m^{(1)} + E'_{1v_2}(\lambda)\vec{V}_m^{(2)})\Phi_m, \quad \vec{H}'_{1\tau}(\lambda)\Big|_{z=0} = (H'_{1v_1}(\lambda)\vec{V}_m^{(1)} + H'_{1v_2}(\lambda)\vec{V}_m^{(2)})\Phi_m,\end{aligned}$$

где

$$\begin{aligned}E_{0v_1}(\lambda) &= A(\lambda), \quad E_{0v_2} = -\bar{v}_0 B(\lambda), \quad H_{0v_1} = h_0 B(\lambda), \quad H_{0v_2} = -h_0 \bar{v}_0 A(\lambda); \\ E'_{1v_1}(\lambda) &= x_1(\lambda), \quad E'_{1v_2} = \bar{v}_0 x_2(\lambda), \quad H'_{1v_1} = h_0 x_2(\lambda), \quad H'_{1v_2} = h_0 \bar{v}_0 x_1(\lambda).\end{aligned}$$

В результате касательные составляющие суммарного поля $\vec{E}_1 = \vec{E}_0 + \vec{E}'_1$, $\vec{H}_1 = \vec{H}_0 + \vec{H}'_1$ в полу-пространстве D_1 на плоскости Γ_1 определяются формулами

$$\vec{E}_{1\tau}(\lambda)\Big|_{z=0} = (E_{1v_1}(\lambda)\vec{V}_m^{(1)} + E_{1v_2}(\lambda)\vec{V}_m^{(2)})\Phi_m, \quad \vec{H}_{1\tau}(\lambda)\Big|_{z=0} = (H_{1v_1}(\lambda)\vec{V}_m^{(1)} + H_{1v_2}(\lambda)\vec{V}_m^{(2)})\Phi_m, \quad (48)$$

где

$$\begin{aligned}E_{1v_1}(\lambda) &= x_1(\lambda) + A(\lambda), \quad E_{1v_2}(\lambda) = \bar{v}_0(x_2(\lambda) - B(\lambda)), \\ H_{1v_1} &= h_0(x_2(\lambda) + B(\lambda)), \quad H_{1v_2}(\lambda) = h_0 \bar{v}_0(x_1(\lambda) - A(\lambda)), \quad \bar{v}_0 = \frac{v_0}{k_0}.\end{aligned}$$

Аналогично для касательных составляющих полей \vec{E}_2, \vec{H}_2 (44) на плоскости $\Gamma_1(z=\Delta)$ получим представление

$$\vec{E}_{2\tau}(\lambda)\Big|_{z=\Delta} = (E_{2v_1}(\lambda)\vec{V}_m^{(1)} + E_{2v_2}(\lambda)\vec{V}_m^{(2)})\Phi_m, \quad \vec{H}_{2\tau}(\lambda)\Big|_{z=\Delta} = (H_{2v_1}(\lambda)\vec{V}_m^{(1)} + H_{2v_2}(\lambda)\vec{V}_m^{(2)})\Phi_m, \quad (49)$$

где

$$\begin{aligned}E_{2v_1}(\lambda) &= y_1(\lambda)F_0^{(-)}, \quad E_{2v_2}(\lambda) = -\bar{v}_0 y_2(\lambda)F_0^{(-)}, \quad H_{2v_1}(\lambda) = h_0 y_2(\lambda)F_0^{(-)}, \\ H_{2v_2}(\lambda) &= -h_0 \bar{v}_0 y_1(\lambda)F_0^{(-)}, \quad F_0^{(\pm)} = e^{\pm v_0 \Delta}.\end{aligned}$$

Удовлетворим граничным условиям (39), (41). Запишем касательные составляющие полей (45) в слое D на плоскости $z = \text{const}$ в базисе $\vec{V}_m^{(1)}, \vec{V}_m^{(2)}$, учитывая представление полей (10), (11) при $k = k^{(s)}$:

$$\begin{aligned}\vec{E}_\tau(\lambda)\Big|_z &= (E_{v_1 z} \vec{V}_m^{(1)} + E_{v_2 z} \vec{V}_m^{(2)})\Phi_m = \sum_{s=1}^4 \left(z_s \vec{E}^{[+s]} + z_{-s} \vec{E}^{[-s]} \right)\Big|_z = \\ &= \sum_{s=1}^4 \left[z_s \left(e_1^{(s)} \vec{M}_{m\tau}^{(+1)} + e_2^{(s)} \vec{M}_{m\tau}^{(+2)} \right) + z_{-s} \left(e_1^{(s)} \vec{M}_{m\tau}^{(-1)} - e_2^{(s)} \vec{M}_{m\tau}^{(-2)} \right) \right]\Big|_z = \\ &= \sum_{s=1}^4 \left[z_s \left(e_1^{(s)} \vec{V}_m^{(1)} + e_2^{(s)} \frac{v^{(s)}}{k^{(s)}} \vec{V}_m^{(2)} \right) F_s^{(+)}(z) + z_{-s} \left(e_1^{(s)} \vec{V}_m^{(1)} + e_2^{(s)} \frac{v^{(s)}}{k^{(s)}} \vec{V}_m^{(2)} \right) F_s^{(-)}(z) \right] \Phi_m, \quad (50) \\ F_s^{(\pm)}(z) &= e^{\pm v^{(s)} z}.\end{aligned}$$

Из равенств (50) следуют формулы

$$\begin{aligned}E_{v_1 z} &= \sum_{s=1}^4 \left(z_s F_s^{(+)}(z) + z_{-s} F_s^{(-)}(z) \right) e_1^{(s)}, \\ E_{v_2 z} &= \sum_{s=1}^4 \left(z_s F_s^{(+)}(z) + z_{-s} F_s^{(-)}(z) \right) \frac{v^{(s)}}{k^{(s)}} e_2^{(s)}.\end{aligned}$$

Для магнитного поля получим представление

$$\vec{H}_\tau(\lambda)|_z = (H_{v_1z} \vec{V}_m^{(1)} + H_{v_2z}(\lambda) \vec{V}_m^{(2)}) \Phi_m, \quad (51)$$

где

$$H_{v_1z} = \sum_{s=1}^4 (z_s F_s^{(+)}(z) + z_{-s} F_s^{(-)}(z)) h_1^{(s)},$$

$$H_{v_2z} = \sum_{s=1}^4 (z_s F_s^{(+)}(z) + z_{-s} F_s^{(-)}(z)) \frac{v^{(s)}}{k^{(s)}} h_2^{(s)}.$$

Касательные составляющие (50), (51) на плоскостях Γ_1 и Γ_2 имеют вид

$$\vec{E}_\tau|_{z=0} = (E_{v_10} \vec{V}_m^{(1)} + E_{v_20} \vec{V}_m^{(2)}) \Phi_m, \quad \vec{H}_\tau|_{z=0} = (H_{v_10} \vec{V}_m^{(1)} + H_{v_20} \vec{V}_m^{(2)}) \Phi_m,$$

$$\vec{E}_\tau|_{z=\Delta} = (E_{v_1\Delta} \vec{V}_m^{(1)} + E_{v_2\Delta} \vec{V}_m^{(2)}) \Phi_m, \quad \vec{H}_\tau|_{z=\Delta} = (H_{v_1\Delta} \vec{V}_m^{(1)} + H_{v_2\Delta} \vec{V}_m^{(2)}) \Phi_m, \quad (52)$$

где

$$E_{v_10} = \sum_{s=1}^4 (z_s + z_{-s}) e_1^{(s)}, \quad E_{v_20} = \sum_{s=1}^4 (z_s + z_{-s}) \frac{v^{(s)}}{k^{(s)}} e_2^{(s)},$$

$$H_{v_10} = \sum_{s=1}^4 (z_s + z_{-s}) h_1^{(s)}, \quad H_{v_20} = \sum_{s=1}^4 (z_s + z_{-s}) \frac{v^{(s)}}{k^{(s)}} h_2^{(s)},$$

$$E_{v_1\Delta} = \sum_{s=1}^4 (z_s F^{(+s)} + z_{-s} F^{(-s)}) e_1^{(s)}, \quad E_{v_2\Delta} = \sum_{s=1}^4 (z_s F^{(+s)} + z_{-s} F^{(-s)}) \frac{v^{(s)}}{k^{(s)}} e_2^{(s)},$$

$$H_{v_1\Delta} = \sum_{s=1}^4 (z_s F^{(+s)} + z_{-s} F^{(-s)}) h_1^{(s)},$$

$$H_{v_2\Delta} = \sum_{s=1}^4 (z_s F^{(+s)} + z_{-s} F^{(-s)}) \frac{v^{(s)}}{k^{(s)}} h_2^{(s)},$$

$$F^{(\pm s)} = F_s^{(\pm)}(\Delta) = e^{\pm v^{(s)} z}.$$

Введем вспомогательные векторы, образованные из компонент полей (48), (49):

$$\vec{U}_1 = \begin{pmatrix} E_{1v_1} \\ H_{1v_2} \\ E_{1v_2} \\ H_{1v_1} \end{pmatrix} = \begin{pmatrix} x_1 + A \\ h_0 \bar{v}_0 (x_1 - A) \\ \bar{v}_0 (x_2 - B) \\ h_0 (x_2 + B) \end{pmatrix}, \quad \vec{U}_2 = \begin{pmatrix} E_{2v_1} \\ H_{2v_2} \\ E_{2v_2} \\ H_{2v_1} \end{pmatrix} = \begin{pmatrix} y_1 F_0^{(-)} \\ -h_0 \bar{v}_0 y_1 F_0^{(-)} \\ -\bar{v}_0 y_2 F_0^{(-)} \\ h_0 y_2 F_0^{(-)} \end{pmatrix}. \quad (53)$$

Введем также вспомогательные векторы из компонент полей (52):

$$\vec{W}_1 = (E_{v_10}, H_{v_20}, E_{v_20}, H_{v_10})_\Gamma, \quad \vec{W}_2 = (E_{v_1\Delta}, H_{v_2\Delta}, E_{v_2\Delta}, H_{v_1\Delta})_\Gamma,$$

$$\vec{z}_+ = (z_1, z_2, z_3, z_4)^\Gamma, \quad \vec{z}_- = (z_{-1}, z_{-2}, z_{-3}, z_{-4})^\Gamma$$

и матрицы

$$\hat{P}_j = \{P_{ls}^{(j)}\}, \hat{M}_j = \{M_{ls}^{(j)}\}, \quad j=1, 2; \quad l, s=1, 2, 3, 4;$$

$$P_{1s}^{(1)} = e_1^{(s)}, P_{2s}^{(1)} = \frac{v^{(s)}}{k^{(s)}} h_2^{(s)}, P_{3s}^{(1)} = \frac{v^{(s)}}{k^{(s)}} e_2^{(s)}, P_{4s}^{(1)} = h_1^{(s)};$$

$$M_{1s}^{(1)} = e_1^{(s)}, M_{2s}^{(1)} = \frac{v^{(s)}}{k^{(s)}} h_2^{(s)}, M_{3s}^{(1)} = \frac{v^{(s)}}{k^{(s)}} e_2^{(s)}, M_{4s}^{(1)} = h_1^{(s)}; \quad (54)$$

$$P_{1s}^{(2)} = F^{(+s)} e_1^{(s)}, P_{2s}^{(2)} = F^{(+s)} \frac{v^{(s)}}{k^{(s)}} h_2^{(s)}, P_{3s}^{(2)} = F^{(+s)} \frac{v^{(s)}}{k^{(s)}} e_2^{(s)}, P_{4s}^{(2)} = F^{(+s)} h_1^{(s)};$$

$$M_{1s}^{(2)} = F^{(-s)} e_1^{(s)}, M_{2s}^{(2)} = F^{(-s)} \frac{v^{(s)}}{k^{(s)}} h_2^{(s)}, M_{3s}^{(2)} = F^{(-s)} \frac{v^{(s)}}{k^{(s)}} e_2^{(s)}, M_{4s}^{(2)} = F^{(-s)} h_1^{(s)}.$$

Используя компоненты касательных векторов (52), векторы \vec{z}_+ , \vec{z}_- и матрицы (54), запишем векторы \vec{W}_j в матричном виде. Получим формулу

$$\vec{W}_j = \hat{P}_j \vec{z}_+ + \hat{M}_j \vec{z}_-. \quad (55)$$

Скалярные соотношения (46), (47) запишем в матричном виде

$$\hat{Q} \vec{z}_+ = \hat{L} \vec{z}_-, \quad (56)$$

где матрицы $\hat{Q} = \{Q_{ls}\}$, $\hat{L} = \{L_{ls}\}$, $l, s=1, 2, 3, 4$;

$$Q_{1s} = (\bar{v}^{(s)} + p) m_1^{(s)}, Q_{2s} = (\bar{v}^{(s)} + p) \frac{\bar{v}^{(s)}}{K_s} m_2^{(s)},$$

$$Q_{3s} = (\bar{v}^{(s)} - q) F_s^{(+)} m_1^{(s)}, Q_{4s} = (\bar{v}^{(s)} - q) F_s^{(+)} \frac{\bar{v}^{(s)}}{K_s} m_2^{(s)};$$

$$L_{1s} = (\bar{v}^{(s)} - p) m_1^{(s)}, L_{2s} = (\bar{v}^{(s)} - p) \frac{\bar{v}^{(s)}}{K_s} m_2^{(s)},$$

$$L_{3s} = (\bar{v}^{(s)} + q) F_s^{(-)} m_1^{(s)}, L_{4s} = (\bar{v}^{(s)} + q) F_s^{(-)} \frac{\bar{v}^{(s)}}{K_s} m_2^{(s)}.$$

Удовлетворим граничным условиям (39), (41), сравнивая касательные составляющие (53), (55) полей на плоскостях Γ_1 и Γ_2 :

$$\vec{U}_1 = \hat{P}_1 \vec{z}_+ + \hat{M}_1 \vec{z}_-, \quad \vec{U}_2 = \hat{P}_2 \vec{z}_+ + \hat{M}_2 \vec{z}_-. \quad (57)$$

Из уравнений (57) с помощью уравнения (56) исключим вектор \vec{z}_- :

$$\vec{U}_1 = (\hat{P}_1 + \hat{M}_1 \hat{L}^{-1} \hat{Q}) \vec{z}_+, \quad \vec{U}_2 = (\hat{P}_2 + \hat{M}_2 \hat{L}^{-1} \hat{Q}) \vec{z}_+. \quad (58)$$

Из первого уравнения (58), используя второе уравнение (58), исключим вектор \vec{z}_+ :

$$\begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \\ C_{41} & C_{42} & C_{43} & C_{44} \end{pmatrix} \begin{pmatrix} E_{2v_1} \\ H_{2v_2} \\ E_{2v_2} \\ H_{2v_1} \end{pmatrix} = \begin{pmatrix} E_{1v_1} \\ H_{1v_2} \\ E_{1v_2} \\ H_{1v_1} \end{pmatrix}, \quad (59)$$

где $\hat{C} = \{C_{ls}\} = (\hat{P}_1 + \hat{M}_1 \hat{L}^{-1} \hat{Q}) (\hat{P}_2 + \hat{M}_2 \hat{L}^{-1} \hat{Q})^{-1}$.

Подставим векторы \vec{U}_j (53) в (59) и получим систему алгебраических уравнений для определения амплитуд x_j, y_j :

$$\begin{aligned} C_{11}y_1 - C_{12}h_0\bar{v}_0y_1 - C_{13}\bar{v}_0y_2 + C_{14}h_0y_2 &= (x_1 + A)/F_0^{(-)}, \\ C_{21}y_1 - C_{22}h_0\bar{v}_0y_1 - C_{23}\bar{v}_0y_2 + C_{24}h_0y_2 &= h_0\bar{v}_0(x_1 - A)/F_0^{(-)}, \\ C_{31}y_1 - C_{32}h_0\bar{v}_0y_1 - C_{33}\bar{v}_0y_2 + C_{34}h_0y_2 &= \bar{v}_0(x_2 - B)/F_0^{(-)}, \\ C_{41}y_1 - C_{42}h_0\bar{v}_0y_1 - C_{43}\bar{v}_0y_2 + C_{44}h_0y_2 &= h_0(x_2 + B)/F_0^{(-)}. \end{aligned} \quad (60)$$

Комбинируя уравнения (60), исключим величины x_1, x_2 . Получим систему алгебраических уравнений для определения амплитуд y_1, y_2 :

$$Q_{11}y_1 + Q_{12}y_2 = 2A, \quad Q_{21}y_1 + Q_{22}y_2 = 2B, \quad (61)$$

где

$$\begin{aligned} Q_{11} &= C_{11} - h_0\bar{v}_0C_{12} - \frac{1}{h_0\bar{v}_0}C_{21} + C_{22}, \quad Q_{12} = -\bar{v}_0C_{13} + h_0C_{14} + \frac{1}{h_0}C_{23} - \frac{1}{\bar{v}_0}C_{24}, \\ Q_{21} &= -\frac{1}{\bar{v}_0}C_{31} + h_0C_{32} + \frac{1}{h_0}C_{41} - \bar{v}_0C_{42}, \quad Q_{22} = C_{33} - \frac{h_0}{\bar{v}_0}C_{34} - \frac{\bar{v}_0}{h_0}C_{43} + C_{44}. \end{aligned}$$

Разрешим систему уравнений (61):

$$y_1 = \frac{2}{d}(Q_{22}A - Q_{12}B), \quad y_2 = \frac{2}{d}(Q_{11}B - Q_{21}A), \quad d = Q_{11}Q_{22} - Q_{12}Q_{21}.$$

Амплитуды x_1, x_2 определим из первого и четвертого уравнений (60):

$$\begin{aligned} x_1 &= -A + \frac{2}{d}F_0^{(-)} \left[((C_{11} - h_0\bar{v}_0C_{12})Q_{22} - (h_0C_{14} - \bar{v}_0C_{13})Q_{21})A + \right. \\ &\quad \left. + ((h_0C_{14} - \bar{v}_0C_{13})Q_{11} - (C_{11} - h_0\bar{v}_0C_{12})Q_{12})B \right], \\ x_2 &= -B + \frac{2}{h_0d}F_0^{(-)} \left[((C_{41} - h_0\bar{v}_0C_{42})Q_{22} - (h_0C_{44} - \bar{v}_0C_{43})Q_{21})A + \right. \\ &\quad \left. + ((h_0C_{44} - \bar{v}_0C_{43})Q_{11} - (C_{41} - h_0\bar{v}_0C_{42})Q_{12})B \right]. \end{aligned}$$

Амплитуды отраженного и прошедшего через экран полей определены. Амплитуды полей (45) в слое экрана не вычисляем, так как практический интерес представляют отраженные и прошедшие через экран поля. ■

Следствие. На поверхностях плоского экрана из пермаллоя при воздействии первичного поля с осевой симметрией (34) выполнены двухсторонние нелокальные граничные условия, связывающие электромагнитные поля по обе стороны экрана на плоскостях Γ_1 и Γ_2 :

$$\vec{U}_1(M_1) = \hat{C}\vec{U}_2(M_2),$$

где

$$M_1(x, y, 0) \in \Gamma_1, M_2(x, y, \Delta) \in \Gamma_2, \vec{U}_j = (E_{jv_1}, H_{jv_2}, E_{jv_2}, H_{jv_1})^T, \\ \hat{C} = \hat{C}(m, \lambda, \omega; \dot{a}, \dot{g}, \dot{\gamma}, \dot{\sigma}, \Delta; H_0, M_0) = (\hat{P}_1 + \hat{M}_1 \hat{L}^{-1} \hat{Q}) (\hat{P}_2 + \hat{M}_2 \hat{L}^{-1} \hat{Q})^{-1},$$

компоненты матриц \hat{P}_j, \hat{M}_j задаются выражениями (54), матрицы \hat{L}, \hat{Q} определены в соотношении (56).

Заключение. В статье разработана методика моделирования процессов проникновения последовательностей монохроматических электромагнитных полей с осевой симметрией, зависящих от действительного параметра λ , через пленочный плоский однослойный экран, выполненный из пермаллоя. Аналитически решена краевая задача с дополнительными дифференциальными граничными условиями для поля намагниченности на поверхностях экрана, обеспечивающими единственность решения задачи, и вычислены амплитуды отраженного и прошедшего через экран электромагнитных полей. Новизна работы состоит в том, что метод решения краевой задачи с использованием дополнительных граничных условий позволил получить алгебраическое дисперсионное уравнение четвертой степени с коэффициентами, зависящими от действительного параметра λ и дискретного параметра m . С помощью четырех комплексных корней дисперсионного уравнения, зависящих от параметров λ и m , определены четыре независимых электромагнитных поля с осевой симметрией, распространяющихся в слое пермаллоя в прямом направлении, и четыре поля – в обратном направлении. Построены двухсторонние граничные условия, связывающие поля по обе стороны экрана. Результаты работы могут быть использованы при практическом создании экранов с намагниченностью для защиты от воздействия внешних симметричных пучков электромагнитных волн.

Список использованных источников

1. Виноградов, А. П. Электродинамика композитных материалов. – М. : Эдиториал УРСС, 2001. – 206 с.
2. Ерофеенко, В. Т. Аналитическое моделирование в электродинамике / В. Т. Ерофеенко, И. С. Козловская. – Минск : БГУ, 2010. – 304 с.
3. Закономерности проникновения электромагнитных волн через металлические магнитные пленки / А. Б. Ринкевич [и др.] // Журнал технической физики. – 2009. – Т. 79, вып. 9. – С. 96–106.
4. Ерофеенко, В. Т. Математическая модель экранирования монохроматических электромагнитных полей плоскими экранами из пермаллоя / В. Т. Ерофеенко // Информатика. – 2019. – Т. 16, № 2. – С. 7–18.
5. Шевченко, В. В. О прохождении плоских волн через границу двух поглощающих сред / В. В. Шевченко // Радиотехника и электроника. – 2004. – Т. 49, № 9. – С. 1048–1053.
6. Ерофеенко, В. Т. Моделирование электродинамического контакта двух материалов при воздействии электромагнитных волн / В. Т. Ерофеенко // Радиотехника и электроника. – 2012. – Т. 57, № 3. – С. 314–319.
7. Ерофеенко, В. Т. Модели граничных условий на композиционных экранах для электромагнитных полей с осевой симметрией / В. Т. Ерофеенко // Весці Нац. акад. навук Беларусі. Сер. фіз.-мат. навук. – 2010. – № 2. – С. 41–45.
8. Ерофеенко, В. Т. Преобразование пучков электромагнитных волн при прохождении через экран из кирального метаматериала / В. Т. Ерофеенко, В. Ф. Бондаренко // Информатика. – 2013. – № 1(37). – С. 5–17.
9. Бондаренко, В. Ф. Экранирование импульсных электромагнитных полей многослойными плоскопараллельными экранами с чередующимися магнитными и немагнитными слоями / В. Ф. Бондаренко, В. Т. Ерофеенко // Физические основы приборостроения. – 2017. – Т. 6, № 2. – С. 53–66.
10. Ерофеенко, В. Т. Решение задачи экранирования круговых пучков электромагнитных волн плоским магнитодиэлектрическим экраном / В. Т. Ерофеенко, А. И. Урбанович // Труды XXIX Междунар. конф.

«Радиационная физика твердого тела», Севастополь, 08–13 июля 2019 г. ; под ред. Г. Г. Бондаренко. – М. : ФГБНУ «НИИПМТ», 2019. – С. 352–362.

11. Ерофеенко, В. Т. Двухсторонние граничные условия нестационарной электродинамики на тонких проводящих оболочках / В. Т. Ерофеенко, Е. П. Красковская // Радиотехника и электроника. – 2001. – Т. 46, № 11. – С. 1293–1298.

12. Ерофеенко, В. Т. Моделирование двухсторонних граничных условий для акустических волн на упругом экране / В. Т. Ерофеенко // Весці Нац. акад. навук Беларусі. Сер. фіз.-мат. навук. – 2010. – № 4. – С. 76–84.

13. Ерофеенко, В. Т. Моделирование поверхностных электромагнитных волн с осевой симметрией на биизотропном однослойном плоском экране / В. Т. Ерофеенко // Информатика. – 2019. – Т. 16, № 4. – С. 63–76.

References

1. Vinogradov A. P. Jelektrodinamika kompozitnyh materialov. *Electrodynamics of Composite Materials*. Moscow, Editorial URSS, 2001, 206 p. (in Russian).

2. Erofeenko V. T., Kozlovskaja I. S. Analiticheskoe modelirovanie v jelektrodinamike. *Analytical Modeling in Electrodynamicity*. Minsk, Belorusskij gosudarstvennyj universitet, 2010, 304 p. (in Russian).

3. Rinkevich A. B., Perov D. V., Vas'kovskiy V. O., Lepalovskiy V. N. Zakonomernosti proniknovenija jelektromagnitnyh voln cherez metallicheskie magnitnye plenki [Regularity of a penetration electromagnetic waves across the metallic magnetic films]. *Zhurnal tehnicheckoj fiziki [Technical Physics]*, 2009, vol. 79, no. 9, pp. 96–106 (in Russian).

4. Erofeenko V. T. Matematicheskaja model' jekranirovaniya monohromaticheskikh jelektromagnitnyh polej ploskimi jekranami iz permalloya [Mathematical model of shielding monochromatic electromagnetic fields by means of plane screens made of permalloy]. *Informatika [Informatics]*, 2019, vol. 16, no. 2, pp. 7–18 (in Russian).

5. Schevchenko V. V. O prohozhdenii ploskih voln cherez granicu dvuh pogloshhajushchih sred [About pass of a plane waves through a boundary of the two absorbing medium]. *Radiotekhnika i jelektronika [Journal of Communications Technology and Electronics]*, 2004, vol. 49, no. 9, pp. 1048–1053 (in Russian).

6. Erofeenko V. T. Modelirovanie jelektrodynamicheckogo kontakta dvuh materialov pri vozdeystvii elektromagnitnyh voln [Modeling of the electrodynamic contact of two materials at the action of electromagnetic waves]. *Radiotekhnika i jelektronika [Journal of Communications Technology and Electronics]*, 2012, vol. 57, no. 3, pp. 314–319 (in Russian).

7. Erofeenko V. T. Modeli granichnyh uslovij na kompozicionnyh ekranah dlja elektromagnitnyh polej s osevoj simmetriej [Models for boundary conditions on the composite screens forelectromagnetic fields with axial symmetry]. *Vestsi Natsyanalnai akademii navuk Belarusi. Seryia fizika-matjematychnyh navuk [Proceedings of the National Academy of Sciences of Belarus. Physics and Mathematical Series]*, 2010, no. 2, pp. 41–45 (in Russian).

8. Erofeenko V. T., Bondarenko V. F. Preobrazovanie puchkov elektromagnitnyh voln pri prohozhdenii cherez ekran iz kiral'nogo metamateriala [Transformation of beams of electromagnetic waves passing through a chiral metamaterialscreen]. *Informatika [Informatics]*, 2013, no. 1(37), pp. 5–17 (in Russian).

9. Bondarenko V. F., Erofeenko V. T. Ekranirovanie impul'snyh jelektromagnitnyh polej mnogoslajnymi ploskoparallel'nymi jekranami s cheredujushhimisja magnitnymi i nemagnitnymi slojami [The shielding of impulse electromagnetic fields by multilayer plane-parallel screens with alternating magnetic and non-magnetic layers]. *Fizicheskie osnovy priborostroenija [Physikal Base of the Apparatus Construction]*, 2017, vol. 6, no. 2, pp. 53–66 (in Russian).

10. Erofeenko V. T., Urbanovich A. I. Reshenie zadachi ekranirovaniya krugovyh puchkov elektromagnitnyh voln ploskim magnitodielektricheskim jekranom [Solution of the problems of shielding of the beams of electromagnetic waves by means of plane magnetodielectrical screen]. *Trudy XXIX Mezhdunarodnoj konferencii "Radiacionnaja fizika tvjordogo tela", Sevastopol', 08–13 ijulja 2019 g. [Works of the XXIX International Conference "Radiative Physics of the Rigid Body", Sevastopol, 08–13 July 2019]*, Moskva, Federal'noe gosudarstvennoe bjuzhethnoe nauchnoe uchrezhdenie "Nauchno-issledovatel'skij institut perspektivnyh materialov i tehnologij", 2019, pp. 352–362 (in Russian).

11. Erofeenko V. T., Kraskovskaja E. P. Dvuhstoronnie granichnye uslovija nestacionarnoj jelektrodinamiki na tonkih provodjashchih obolochkah [The two-sided boundary conditions of nonstationary electrodynamicity on the thin conductivity hulls]. *Radiotekhnika i jelektronika [Journal of Communications Technology and Electronics]*, 2001, vol. 46, no. 11, pp. 1293–1298 (in Russian).

12. Erofeenko V. T. Modelirovanie dvuhstoronnih granichnyh uslovij dlja akusticheskikh voln na uprugom jekrane [Modeling of the two-sided boundary conditions for acoustic waves on a elastic screen]. *Vesti Natsyianalnai akademii navuk Belarusi. Seryia fizika-matjematychnyh navuk [Proceedings of the National Academy of Sciences of Belarus. Physics and Mathematical Series]*, 2010, no. 4, pp. 76–84 (in Russian).

13. Erofeenko V. T. Modelirovanie poverchnostnyh elektrtomagnitnyh voln s osevoj simmetrieoj na biizotropnom odnoslojnom ploskom ekrane [Modeling of surface electromagnetic waves with axial symmetry on a bi-isotropic one-layer plane screen]. *Informatika [Informatics]*, 2019, vol. 16, no. 4, pp. 63–76 (in Russian).

Информация об авторе

Ерофеенко Виктор Тихонович, доктор физико-математических наук, профессор, главный научный сотрудник научно-исследовательской лаборатории математических методов защиты информации, Учреждение БГУ «НИИ прикладных проблем математики и информатики».
E-mail: bsu_erofeenko@tut.by

Information about the author

Viktor T. Erofeenko, Dr. Sci. (Phys.-Math.), Professor, Chief Researcher Associate of the Research Laboratory of Mathematical Methods of Information Security, Establishment of the Belarusian State University "Research Institute for Applied Problems of Mathematics and Informatics", Minsk, Belarus.
E-mail: bsu_erofeenko@tut.by

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
APPLIED INFORMATION TECHNOLOGIESУДК 621.382.3
<https://doi.org/10.37661/1816-0301-2020-17-2-120-138>Поступила в редакцию 09.03.2020
Received 09.03.2020Принята к публикации 16.03.2020
Accepted 16.03.2020**Наведение сверхзвукового управляемого объекта
в многопозиционной радиолокационной станции
воздушного базирования****В. Э. Маркевич[✉], В. В. Легкоступ**

ОАО «АЛЕВКУРП», Минск, Беларусь

[✉]E-mail: mark.vit@tut.by

Аннотация. Исследована возможность наведения управляемого сверхзвукового авиационного беспилотного перехватчика на движущуюся цель в автономной многопозиционной радиолокационной станции воздушного базирования. Для получения координатной информации выбран алгоритм разностно-дальномерных и разностных дальностно-доплеровских пространственных измерений при использовании ограниченного числа передающих позиций, не обеспечивающих однозначное определение координат и скорости объекта с высокой точностью. Предложены различные подходы для устранения априорной неопределенности оцениваемых координат в условиях ограниченного набора измерительных радиотехнических позиций.

Рассмотрен аналитический метод конструирования устройства оптимального управления состоянием нелинейного многомерного и многосвязного динамического объекта с учетом ограничений, позволяющий получить конечные расчетные соотношения в замкнутой алгебраической форме. Выполнен синтез устройства управления при использовании первичных измерений в декартовой и сферической системах координат, получены несколько разновидностей комбинированного метода наведения управляемого объекта в мгновенную точку встречи с целью. Приведенные алгоритмы наведения являются обобщением широко используемого на практике метода пропорциональной навигации и его различных модификаций. Отличительной особенностью синтезированных алгоритмов является естественный учет нелинейной, многомерной и многосвязной структуры объекта управления, а также параметров нестационарных возмущающих воздействий (ускорения силы тяжести, проекций продольного ускорения объекта и ускорения маневра цели), действующих в процессе наведения.

Получены унифицированные кинематические дифференциальные уравнения, описывающие динамику движения объекта управления, которые могут быть использованы при синтезе регулятора, функционирующего по задаваемым переключаемым или плавно сопрягаемым нестационарным критериям оптимальности, именуемым целевыми функциями или целевыми интегральными многообразиями. Разработанные алгоритмы могут быть использованы при проектировании автономных систем самонаведения и телеуправления, реализуемых аппаратно-программным способом как на борту беспилотного авиационного или артиллерийского перехватчика, так и в составе автономных многопозиционных радиолокационных станций воздушного базирования.

Ключевые слова: сверхзвуковой беспилотный летательный аппарат, аналитическое конструирование агрегированных регуляторов, комбинированный метод пропорциональной навигации, многопозиционная радиолокационная станция воздушного базирования, разностно-дальномерный метод измерений

Для цитирования. Маркевич, В. Э. Наведение сверхзвукового управляемого объекта в многопозиционной радиолокационной станции воздушного базирования / В. Э. Маркевич, В. В. Легкоступ // Информатика. – 2020. – Т. 17, № 2. – С. 120–138. <https://doi.org/10.37661/1816-0301-2020-17-2-120-138>

Guiding a supersonic controlled object in a multi-position air basing radar

Vitaliy E. Markevich[✉], Victor V. Legkostup

OJSC "ALEVKURP", Minsk, Belarus

[✉]E-mail: mark.vit@tut.by

Abstract. The possibility of control of a guided supersonic aviation unmanned interceptor on a moving target in an autonomous airborne multi-position radar station has been investigated. To obtain the coordinate information, the algorithm of difference-range finding and difference range-Doppler spatial measurements is selected using a limited number of transmitting positions that do not provide an unambiguous determination of the coordinates and speed of the object with high accuracy. The paper proposes various approaches to eliminate a priori uncertainty regarding the estimated coordinates in a limited set of measuring radio engineering positions.

An analytical method is considered for constructing an optimal state control device for a nonlinear multidimensional and multiply connected dynamic object, taking into account the constraints, which allows one to obtain finite computational relations in a closed algebraic form. The control device was synthesized using measurements in the Cartesian and spherical coordinate systems, several varieties of the combined method of pointing a controlled object to an instant meeting point were obtained. The given guidance algorithms are a generalization of the method of proportional navigation widely used in practice and its modifications. A distinctive feature of the synthesized algorithms is the natural accounting for the nonlinear, multidimensional, and multiply connected structure of the control object, as well as the parameters of unsteady perturbations (acceleration of gravity, projections of the longitudinal acceleration of the object and acceleration of the target's maneuver) that act in the guidance process.

The unified kinematic differential equations that describe the dynamics of the control object, are obtained, and they can be used to synthesize a controller that operates according to switched or smoothly matched non-stationary optimality criteria (target functions, target integral manifolds). The developed algorithms can be used in the design of autonomous homing and telecontrol systems, implemented in hardware and software both on board an unmanned aerial or artillery interceptor, and as a part of autonomous multiposition airborne radar stations.

Keywords: supersonic unmanned aerial vehicle, analytical design of aggregated regulators, combined method of proportional navigation, multi-position air-based radar, differential-range measurement method

For citation. Markevich V. E., Legkostup V. V. Guiding a supersonic controlled object in a multi-position air basing radar. *Informatics*, 2020, vol. 17, no. 2, pp. 120–138 (in Russian). <https://doi.org/10.37661/1816-0301-2020-17-2-120-138>

Введение. В настоящее время применяются следующие способы управления и коррекции полета сверхзвуковых беспилотных летательных аппаратов (БПЛА) и различных объектов управления (ОУ) наземного и воздушного базирования, запускаемых с борта летательного аппарата или с наземной пусковой установки:

- измерение собственных координат движения объекта и координат цели с использованием наземной радиотехнической позиции (станции наведения и передачи команд);
- измерение координат собственного движения ОУ с использованием бесплатформенных инерциальных навигационных систем (БИНС);
- измерение координат относительного движения цели и объекта на борту ОУ, например разведывательно-ударного БПЛА или управляемого аэродинамического средства доставки специального назначения с наземным или воздушным стартом;
- комбинация перечисленных способов измерения в интересах повышения точности и помехоустойчивости системы управления в целом.

В некоторых случаях для обозначения подобных ОУ используют, например, название «управляемый авиационный (артиллерийский) беспилотный перехватчик» (УАБП). При этом термин «авиационный (артиллерийский)» характеризует способ старта (воздушный или наземный), а слово «перехватчик» служит для обозначения функционального (целевого) назначения ОУ.

Для наведения ОУ на наземные неподвижные или малоподвижные точечные объекты используется внешний подсвет (как правило, лазерный), реализуемый с помощью передатчика, устанавливаемого на борту пилотируемого носителя, с борта которого осуществляется пуск.

Для наземных пусковых установок используются станции лазерного целеуказания (подсвета) и передачи данных, транспортируемых на колесном шасси (полустационарные) или ручную (носимые).

Наиболее существенными недостатками ОУ с лазерным или радиолокационным наведением являются их высокая стоимость, низкая мобильность, а также необходимость постоянного лазерного или радиолокационного подсвета, осуществляемого в ряде случаев с привлечением оператора наведения или группы операторов, что закономерно повышает огневую уязвимость аппаратуры управления с противодействующей стороны. Одним из перспективных вариантов дальнейшего совершенствования и повышения эффективности применения УАБП является использование нескольких автономных БПЛА с размещенными на борту приемо-передающими станциями, образующими в совокупности распределенную в пространстве радиолокационную станцию (РЛС), или многопозиционную РЛС (МП РЛС). В зависимости от используемого способа измерения координат различают активные, полуактивные и пассивные МП РЛС. В настоящей статье основное внимание уделено полуактивным МП РЛС, реализующим гиперболический способ оценивания координат и скорости движения объекта в пространстве.

Преимущества применения разностно-дальномерных (РД) и разностных дальностно-доплеровских (РДД) РЛС с реализацией обработки как на борту одной или нескольких позиций, так и на борту ОУ определяются следующими факторами:

- необходимостью снижения стоимости разработки, серийного производства и эксплуатации (применения) ОУ;
- исключением из процесса непосредственного измерения координат и выработки команд управления (наведения) оператора или группы операторов;
- организацией безопасной работы наземных операторов, участвующих в обеспечении процесса коррекции траектории полета и наведения ОУ на выбранный целевой объект (цель).

Существующие в настоящее время возможности аппаратной и программной реализации алгоритмов управления ракетами позволяют отказаться от превалировавшего до недавнего времени подхода, основанного на максимально возможном упрощении алгоритмов обработки, обеспечивающих допустимое качество функционирования. Нынешний уровень, имеющиеся возможности и ближайшие перспективы развития средств обработки радиолокационной и оптической информации позволяют реализовывать не только упрощенные алгоритмы с неполным учетом доступной априори информации, но и оптимальные алгоритмы обработки сигнальных и траекторных признаков, позволяющие значительно повысить эффективность применения как бортовых, так и наземных систем наведения.

Принципы оценивания координат объекта в РДД МП РЛС воздушного базирования.

Для разработки алгоритмов оценивания координат и синтеза оптимального управления воспользуемся подходами и методиками, представленными в работах [1–5]. Для измерения координат и управления объектом введем в рассмотрение несколько систем координат, связанных как с неподвижными позициями, так и подвижными объектами:

- базовую (целевую) систему координат (СК), центр которой совпадает с известным положением целевой точки, продольная ось направлена на географический север, вертикальная ось совпадает с местной вертикалью, а поперечная ось дополняет систему до правой;
- измерительную систему координат наземной РЛС (ИСК НРЛС), которая совмещена с местоположением наземной РЛС наведения и подсвета, продольная ось совпадает с линией визирования ОУ с ориентацией единичных направляющих векторов, дополняющих систему до правой;
- измерительную систему координат МП РЛС (ИСК МП РЛС), центр которой находится в центре масс треугольника, координаты вершины которого совпадают с координатами БПЛА-ретрансляторов, продольная ось ортогональна плоскости указанного треугольника и ориентирована в сторону цели, положение остальных осей определяется в соответствии с приведенным ниже правилом;
- скоростную систему координат (СкСК), которая связана с вектором скорости движения объекта относительно воздушной среды, ее центр совмещен с кинематическим центром масс, продольная ось совпадает с вектором скорости, вертикальная ось перпендикулярна вектору скорости и лежит в базовой плоскости ОУ, поперечная (боковая) ось дополняет систему координат до правой.

Положение осей ИСК МП РЛС задается координатами направляющих векторов продольной и поперечных осей, которые определяются исходя из следующей постановки задачи. Дан единичный вектор нормали к плоскости системы из трех БПЛА $\mathbf{n}_x(x, y, z)$. Необходимо получить два единичных вектора $\mathbf{n}_y(x, y, z)$ и $\mathbf{n}_z(x, y, z)$, удовлетворяющих условию взаимной ортогональности и образующих в совокупности правую тройку векторов

$$\mathbf{n}_z = (\mathbf{n}_x \times \mathbf{n}_y), \quad \mathbf{n}_x = (\mathbf{n}_y \times \mathbf{n}_z), \quad \mathbf{n}_y = (\mathbf{n}_z \times \mathbf{n}_x). \quad (1)$$

Это детерминированная задача, хоть и имеющая множество решений.

Последовательность выбора требуемого измерительного базиса заключается в следующем:

1. Рассчитывается центр масс системы из трех БПЛА (центр масс плоскости пространственного треугольника):

$$\mathbf{c}_0 = \frac{1}{3} \sum_{j=1}^3 \mathbf{c}_j, \quad \Delta \mathbf{r}_j = \mathbf{c}_j - \mathbf{c}_0, \quad \mathbf{n}_x = \frac{[\Delta \mathbf{r}_j \times \Delta \mathbf{r}_n]}{||[\Delta \mathbf{r}_j \times \Delta \mathbf{r}_n]||}, \quad j \in (1, 3), \quad n \in (1, 3), \quad j \neq n. \quad (2)$$

2. Выбирается произвольный вектор $\mathbf{e}(x, y, z)$, неколлинеарный с $\mathbf{n}_x(x, y, z)$, например вектор (URL: <https://nabbla1.livejournal.com/96041.html>)

$$\mathbf{e} = \text{if} \left(|x| > \frac{1}{\sqrt{3}}, (0 \ 1 \ 0)^T, (1 \ 0 \ 0)^T \right). \quad (3)$$

3. Определяются направляющие векторы оставшихся осей ИСК:

$$\mathbf{n}_y = \frac{(\mathbf{e} \times \mathbf{n}_x)}{||(\mathbf{e} \times \mathbf{n}_x)||}, \quad \mathbf{n}_z = \frac{(\mathbf{n}_x \times \mathbf{n}_y)}{||(\mathbf{n}_x \times \mathbf{n}_y)||}. \quad (4)$$

Координаты ОУ в дальнейшем будут определяться в ИСК, центр и направление осей которой будут динамически изменяться относительно неподвижной базовой (целевой) СК из-за абсолютного и взаимного перемещения в пространстве измерительного треугольника, образованного тремя барражирующими БПЛА.

Возможен вариант привязки центра ИСК к центру окружности, описанной вокруг измерительного треугольника. В этом случае координаты центра подвижной ИСК \mathbf{c}_0 находятся из системы векторных уравнений

$$(\mathbf{c}_j - \mathbf{c}_n) \times (0, 5(\mathbf{c}_j - \mathbf{c}_n) - \mathbf{c}_0) = 0, \quad j \in (1, 3), \quad n \in (1, 3), \quad j \neq n. \quad (5)$$

Математическую модель канала наблюдения можно представить в виде уравнений связи измерений и наблюдений для РДД МП РЛС, записанных в векторно-матричной форме:

$$\begin{aligned} \Delta \rho_{jn} &= |\mathbf{r}_0 - \Delta \mathbf{r}_j| - |\mathbf{r}_0 - \Delta \mathbf{r}_n|, \\ \Delta v_{jn} &= \frac{(\mathbf{r}_0 - \Delta \mathbf{r}_j)^T (\mathbf{v}_0 - \Delta \mathbf{v}_j)}{|\mathbf{r}_0 - \Delta \mathbf{r}_j|} - \frac{(\mathbf{r}_0 - \Delta \mathbf{r}_n)^T (\mathbf{v}_0 - \Delta \mathbf{v}_n)}{|\mathbf{r}_0 - \Delta \mathbf{r}_n|}, \\ \Delta \mathbf{r}_j &= \mathbf{c}_j - \mathbf{c}_0, \quad \mathbf{c}_0 = \frac{1}{3} \sum_{j=1}^3 \mathbf{c}_j, \quad \Delta \mathbf{v}_j = \mathbf{s}_j - \mathbf{s}_0, \quad \mathbf{s}_0 = \frac{1}{3} \sum_{j=1}^3 \mathbf{s}_j, \quad j \in (1, 3), \end{aligned} \quad (6)$$

где $\mathbf{r}_0, \mathbf{v}_0$ – искомый вектор координат и скоростей ОУ; $\mathbf{c}_0, \mathbf{c}_j$ – вектор координат центра масс (ЦМ) системы из трех БПЛА и координаты j -го БПЛА в базовой (неподвижной) СК соответ-

ственно; $\mathbf{s}_0, \mathbf{s}_j$ – скорости перемещения ЦМ системы из трех БПЛА и j -го БПЛА соответственно; $\Delta \mathbf{r}_j$ – радиус-вектор, характеризующий направление на ОУ относительно j -го БПЛА; $\Delta \mathbf{v}_j$ – вектор, характеризующий компоненты скорости ОУ относительно j -го БПЛА.

Приведенные соотношения определяют связь между измеряемыми и оцениваемыми (наблюдаемыми) переменными в СК, связанной с центром масс системы из трех БПЛА. В качестве наблюдаемых переменных могут применяться относительные сферические либо декартовы координаты. При использовании декартовых координат (x, y, z) и их производных (v_x, v_y, v_z) либо сферических координат $(\varepsilon, \beta, \rho)$ и их производных $(\dot{\varepsilon}, \dot{\beta}, \dot{\rho})$ система будет неполной (неопределенной) как для РД-, так и для РДД-измерений. Это обусловлено тем, что число уравнений измерений меньше числа оцениваемых координат (два уравнения и три координаты для РД-метода (РДМ), четыре уравнения и шесть координат для РДД-метода (РДДМ)). Задача решается при рассмотрении одной из искоемых координат, как правило, радиальной дальности (скорости), или продольной координаты x (скорости v_x) в качестве аппроксимируемого нестационарного параметра с известным законом изменения во времени. Альтернативным вариантом является использование информации о текущих координатах и скоростях ОУ от БИНС. Исследование характеристик точности оценок, получаемых от БИНС, и их влияния на ошибки наведения ОУ и вероятности поражения цели является отдельной теоретической задачей.

Таким образом, обеспечить вычислительную определенность системы между числом переменных и числом уравнений измерений можно следующими способами:

- 1) гладкой или кусочно-ломаной экстраполяцией (аппроксимацией) неизвестной координаты (производной) в зависимости от исходных данных для стрельбы и времени полета ОУ;
- 2) совместной обработкой разностных измерений, полученных в различных точках траектории полета ОУ с минимальной кривизной;
- 3) оцениванием одной из координат и ее производной с помощью измерителей другой физической природы (например, при использовании БИНС, бортовых оптических или радиотехнических датчиков ориентации и т. д.).

Первый и второй варианты отличаются относительной простотой реализации, однако не всегда обеспечивают требуемую точность оценивания, в том числе для наведения на неподвижную цель. Третий вариант комплексирования позволяет получить высокую точность измерений, но требует при этом наличия на борту ОУ независимых инерциальных либо активно-пассивных радиотехнических или оптических измерителей.

Компромиссным вариантом является использование БИНС, работающей совместно с РДД-измерителями на конечном участке наведения, когда дальность до наземной или воздушной цели не превышает 5–6 км. С одной стороны, это позволяет снизить вклад навигационных погрешностей из-за неидеальных характеристик датчиков угловых скоростей (ДУС) и датчиков линейных ускорений (ДЛУ) при кратковременных интервалах комплексирования, а с другой – снизить стоимость бортового навигационного оборудования в составе системы ориентирования ОУ. При этом момент начала интегрирования координат, а также оценку дальности можно определять по результатам сравнения оценок доплеровской частоты между БПЛА и ОУ для каждой из подвижных позиций. Данная возможность основана на том, что в момент полета ОУ через плоскость, образованную тремя подвижными БПЛА, доплеровская частота сигнала по линии БПЛА – ОУ изменит знак. В момент прохождения объекта через плоскость измерений пространственная навигационная задача трансформируется в двухмерную, что позволяет оценить координаты ОУ на плоскости, образованной треугольником, в вершинах которого находятся позиции РЛС. По известным координатам БПЛА с применением формул пересчета определяются координаты ОУ в пространстве. Следует отметить, что подобный вариант реализуем в случае, когда скомпенсированы уходы частоты генераторов бортовых передатчиков подвижных позиций и местного гетеродина приемника ОУ. В настоящей работе вопрос обеспечения стабильности частоты передающего и приемного трактов не рассматривается.

Информационная подсистема сопровождения (оценивания координат) и наведения входит в состав системы управления ближнего или среднего радиуса действия, состоящей из следующих элементов:

- наземной станции сопровождения и коррекции траектории ОУ на инерциальном участке наведения, сопрягаемой с пусковыми установками или артиллерийскими орудиями;
- группы приемо-передающих позиций воздушного базирования в районе цели, состоящей из трех БПЛА и предназначенной для обеспечения точного наведения на конечном участке траектории;
- подсистемы оценивания координат цели с использованием отдельного БПЛА с бортовой оптико-локационной станцией.

Вопросы технической реализации системы измерения координат цели требуют отдельного и всестороннего исследования. Поэтому в дальнейшем ограничимся рассмотрением вопросов, связанных исключительно с сопровождением и наведением ОУ. Использование РД- или РДД-метода измерения координат в МП РЛС предполагает реализацию:

- измерительного приемника и устройства управления, работающих по сигналам передающих устройств БПЛА, на борту наводимого ОУ;
- измерительного приемника и устройства управления, работающих по сигналам передающего устройства ОУ, на борту одного или нескольких БПЛА;

Комбинация перечисленных способов – первичное измерение разностных координат – выполняется на борту БПЛА, измеренные координаты передаются на ОУ, формирование команд управления происходит на борту ОУ.

Для приведенных вариантов реализации характерно наличие задержки между моментами измерения и формирования команд управления на период обновления информации, что при высоких скоростях ОУ приводит к росту ошибки наведения (промаха). Кроме того, заметно возрастает сложность аппаратуры линии связи и управления при незначительном приросте эффективности работы. Поэтому в рамках настоящей работы ограничимся рассмотрением варианта реализации устройства оценивания и наведения на борту ОУ. В дальнейшем будем полагать, что собственные координаты и проекции скоростей перемещения БПЛА с приемлемой для наведения точностью измеряются с помощью бортовых БИНС, корректируемых согласно данным взаимных суммарно-разностных измерений по линии БПЛА – БПЛА или данным наземной РЛС управления и передачи команд. Для оценивания координат ОУ на основе уравнений (6) могут быть использованы методы:

- поиска экстремума векторной функции векторной переменной, основанные на минимизации целевой функции (ЦФ) [1];
- статистической теории оптимальной многомерной нелинейной фильтрации в гауссовском приближении оцениваемых процессов [2];
- оценивания фазовых координат и идентификации неизвестных параметров в рамках синергетической теории управления [3, 4].

Непрерывное оценивание координат и параметров движения объекта. Координаты ОУ могут определяться в виде рекуррентной процедуры поиска вектора координат (сферических углов и их производных) в заданном диапазоне радиальных дальностей ρ_0 и скоростей v_0 ОУ, доставляющих минимум функционалу

$$J(\mathbf{x}) = \left(\|\Delta \rho \ \Delta v\| - \|\mathbf{f}_\rho(\mathbf{x}) \ \mathbf{f}_v(\mathbf{x})\| \right)^T \mathbf{R}^{-1} \left(\|\Delta \rho \ \Delta v\| - \|\mathbf{f}_\rho(\mathbf{x}) \ \mathbf{f}_v(\mathbf{x})\| \right), \quad (7)$$

$$\mathbf{x} = \min_{\rho \in \mathbf{R}, v \in \mathbf{U}} \mathbf{J}(\mathbf{x}), \quad \mathbf{J}(\mathbf{x}) = (J_1(\mathbf{x}) \ J_2(\mathbf{x}) \ \dots \ J_k(\mathbf{x}))^T,$$

где $\|\Delta \rho \ \Delta v\| = \|\Delta \rho_1 \ \Delta \rho_2 \ \Delta v_1 \ \Delta v_2\|$ – объединенный вектор разности радиальных дальностей и скоростей для набора из трех БПЛА; $\mathbf{f}_\rho(\mathbf{x})$, $\mathbf{f}_v(\mathbf{x})$ – векторные функции, определяющие зависимость первичных измерений от искомым сферических координат; \mathbf{R} – ковариационная матрица ошибок первичных измерений; $\rho \in \mathbf{R}$, $v \in \mathbf{U}$ – диапазон изменения априорных оценок радиальной дальности и скорости ОУ в ИСК в окрестности истинных значений.

Измерение координат с привлечением методов теории нелинейной оптимальной фильтрации параметров [2] выполняется на основе кинематических уравнений (уравнений состояний динамической системы) для сосредоточенного канала наблюдения:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{C}(\mathbf{x})\mathbf{u} + \mathbf{G}(\mathbf{x})\xi, \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}) + \mu,\end{aligned}\quad (8)$$

где $\mathbf{x}(t)$ – вектор фазовых координат системы дифференциальных уравнений (СДУ); \mathbf{u} – вектор управления состоянием динамической системы; $\mathbf{C}(\mathbf{x})$ – матрица коэффициентов управления, определяемых структурой уравнений состояний; \mathbf{y} – вектор первичных параметров $\mathbf{h}(t, \mathbf{x})$, наблюдаемый на фоне нормального белого шума $\mu(t)$ с ковариационной матрицей шумов наблюдения $\langle \mu(t_0)\mu(t_0 - \tau)^T \rangle = \mathbf{R}_\mu \delta(\tau)$; $\xi(t)$ – некоррелированные гауссовские белые шумы состояний с ковариационной матрицей $\langle \xi(t_0)^T \xi(t_0 - \tau) \rangle = \mathbf{Q}_\xi \delta(\tau)$; \mathbf{Q}_ξ – матрица спектральных плотностей шумов состояний размерности $N \times N$, N – размерность системы уравнений динамики ОУ.

Уравнения оптимального управления получены в [2]:

$$\begin{aligned}\hat{\mathbf{u}} &= \mathbf{C}^+ \mathbf{\Gamma} \mathbf{A} (\mathbf{x}_0 - \hat{\mathbf{x}}), \\ \dot{\mathbf{\Gamma}} &= \mathbf{f}_x(\hat{\mathbf{x}}) \mathbf{\Gamma} + \mathbf{\Gamma} \mathbf{f}_x(\hat{\mathbf{x}})^T - \mathbf{\Gamma} \mathbf{f}_x(\hat{\mathbf{x}}) \mathbf{\Gamma} + \mathbf{C} \mathbf{B} \mathbf{C}^T, \\ \mathbf{C}^+ &= (\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T,\end{aligned}\quad (9)$$

где $\mathbf{\Gamma}$ – матрица управления; \mathbf{x}_0 – заданная кинематическая координата системы; \mathbf{C}^+ – псевдообратная матрица (матрица Мура – Пенроуза) для исходной матрицы \mathbf{C} ; \mathbf{A} , \mathbf{B} – симметрические положительно определенные матрицы штрафов на отклонение оценок фазовых координат от заданного значения и на величину сигнала управления.

Уравнения квазиоптимального оценивания, синтезированные методом инвариантного погружения, имеют вид [2]

$$\dot{\mathbf{P}} = \mathbf{f}_x(\hat{\mathbf{x}}) \mathbf{P} + \mathbf{P} \mathbf{f}_x(\hat{\mathbf{x}})^T + \mathbf{P} \frac{\partial}{\partial \hat{\mathbf{x}}} (\mathbf{H}_0(\hat{\mathbf{x}}) (\mathbf{y} - \mathbf{h}(\hat{\mathbf{x}}))) \mathbf{P} + \mathbf{G} \mathbf{Q} \mathbf{G}^T; \quad (10)$$

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{f}(\hat{\mathbf{x}}) + \mathbf{P} \mathbf{h}_x(\hat{\mathbf{x}}) \mathbf{R}^{-1} (\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}})); \quad (11)$$

$$\mathbf{f}_x(\hat{\mathbf{x}}) = \frac{\partial \mathbf{f}(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}}, \quad \mathbf{h}_x(\hat{\mathbf{x}}) = \frac{\partial \mathbf{h}(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}}, \quad (12)$$

$$\mathbf{H}_0(\hat{\mathbf{x}}) = \mathbf{h}_x(\hat{\mathbf{x}})^T \mathbf{R}^{-1},$$

где $\mathbf{h}_x(\hat{\mathbf{x}}) = \nabla \mathbf{h}_x(\mathbf{x}) = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}$ – градиент вектора первичных измерений по оцениваемым параметрам; \mathbf{G} – матрица коэффициентов усиления шумов состояния, определяемая исходя из структуры устройства оценивания.

Из выражения (10) для оценки матрицы апостериорной дисперсии \mathbf{P} при некоррелированных шумах наблюдений получим формулу

$$\begin{aligned}\mathbf{T}(\hat{\mathbf{x}}) &= \frac{\partial}{\partial \hat{\mathbf{x}}} (\mathbf{h}_x(\hat{\mathbf{x}})^T \mathbf{R}^{-1} (\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}))) = -\mathbf{h}_x(\hat{\mathbf{x}})^T \mathbf{R}^{-1} \mathbf{h}_x(\hat{\mathbf{x}}) + \sum_{p=0}^{P-1} \mathbf{h}_{pxx}(\hat{\mathbf{x}}) \frac{\mathbf{z}_p - \mathbf{h}_p(\hat{\mathbf{x}})}{R_{pp}}, \\ \mathbf{h}_{pxx}(\hat{\mathbf{x}}) &= \left[\frac{\partial^2 \mathbf{h}_p(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}_j \partial \hat{\mathbf{x}}_k} \right], \quad p \in (0, P-1),\end{aligned}\quad (13)$$

где $p \in (1, P)$ – число каналов наблюдений (первичных измерений); R_{jk} – элемент ковариационной матрицы шумов наблюдения.

Оценивание (измерение координат) с применением методов синергетической теории управления является сравнительно новым и мало распространенным подходом, что обуславливает необходимость его отдельного и детального рассмотрения. Поэтому в данной работе для моделирования устройства оценивания ограничимся использованием приведенных выше классических алгоритмов нелинейной фильтрации в гауссовском приближении.

Математическая модель кинематики движения объекта управления. Для выбора модели при описании кинематики движения ОУ необходимо определить совокупность параметров движения (углов и линейных координат, угловых и линейных скоростей), которые доступны для оценивания (наблюдения) и для которых существует дифференциальная или алгебраическая связь с вектором управляемых параметров, например, нормального ускорения ОУ.

Предположим, что управление объектом будет реализовано в двух вариантах: в декартовой СК (ДСК) и сферической СК (СфСК). Управлять будем угловой скоростью вращения (УСВ) и углом ориентации линии визирования ОУ – цель (при реализации управления в СфСК) либо рассогласованием ОУ – цель (при реализации управления в ДСК). Уравнения кинематики с учетом ограничений, накладываемых на нормальные ускорения, примут следующий вид:

$$\begin{aligned}
 \dot{D}_2 &= V_2, \\
 \dot{V}_2 &= W_{x2} + D_2 (\omega_{y2}^2 + \omega_{z2}^2), \\
 \dot{\theta}_2 &= \omega_{z2}, \\
 \dot{\phi}_2 &= \frac{\omega_{y2}}{\cos(\theta_2)}, \\
 \dot{\omega}_{z2} &= \frac{W_0 \operatorname{th}\left(\frac{W_{y2}}{W_0}\right) - 2V_2 \omega_{z2} + W_{yc}}{D_2} - \omega_{y2}^2 \operatorname{tg}(\theta_2), \\
 \dot{\omega}_{y2} &= \frac{-W_0 \operatorname{th}\left(\frac{W_{z2}}{W_0}\right) - 2V_2 \omega_{y2} + W_{zc}}{D_2} + \omega_{y2} \omega_{z2} \operatorname{tg}(\theta_2), \\
 \dot{W}_{y2} &= \alpha_y (U_y - W_{y2}), \\
 \dot{W}_{z2} &= \alpha_z (U_z - W_{z2}),
 \end{aligned} \tag{14}$$

где D_2, V_2 – радиальная дальность и скорость цели относительно ОУ; θ_2, ϕ_2 – углы наклона линии визирования в вертикальной и горизонтальной плоскостях; $\omega_2 = (\omega_{x2} \ \omega_{y2} \ \omega_{z2})^T$ – вектор УСВ линии визирования ОУ – цель; α_y, α_z – нестационарные параметры, характеризующие инерционность объекта по каналам управления и определяемые полосой пропускания контура стабилизации ракеты; U_y, U_z – команда управления; W_0 – величина максимальных располагаемых нормальных ускорений ОУ в выбранной СК; W_{yc}, W_{zc} – ускорения для компенсации влияния силы тяжести, продольного ускорения ОУ и ускорения маневра цели в соответствующие системе координат; $\operatorname{th}(x)$ – функция гиперболического тангенса, реализующая плавное ограничение.

Запишем уравнения кинематики движения ОУ в СфСК:

$$\begin{aligned}
 \dot{\rho}_0 &= V_0, \\
 \dot{V}_j &= W_{x0} + \rho_0 (\omega_{y0}^2 + \omega_{z0}^2), \\
 \dot{\varepsilon}_0 &= \omega_{z0}, \\
 \dot{\beta}_0 &= \frac{\omega_{y0}}{\cos(\varepsilon_0)},
 \end{aligned} \tag{15}$$

$$\begin{aligned}\dot{\omega}_{z0} &= \frac{W_0 \operatorname{th}\left(\frac{W_{y0}}{W_0}\right) - 2V_0\omega_{z0} + W_{yc}}{\rho_0} - \omega_{y0}^2 \operatorname{tg}(\varepsilon_0), \\ \dot{\omega}_{y0} &= \frac{-W_0 \operatorname{th}\left(\frac{W_{z0}}{W_0}\right) - 2V_0\omega_{y0} + W_{zc}}{\rho_0} + \omega_{y0}\omega_{z0} \operatorname{tg}(\varepsilon_0), \\ \dot{W}_{y0} &= \alpha_y (U_y - W_{y0}), \\ \dot{W}_{z0} &= \alpha_z (U_z - W_{z0}),\end{aligned}\quad (15)$$

где D_0, V_0 – радиальная дальность и скорость цели в ИСК; ε_0, β_0 – углы вертикального и горизонтального пеленгов ОУ; $\omega_0 = (\omega_{x0} \ \omega_{y0} \ \omega_{z0})^T$ – вектор УСВ линии дальности ОУ.

Система уравнений (15) позволяет выполнять синтез комбинированных систем управления. Например, на начальном этапе используется метод телеуправления (полное или частичное спрямление, параллельное сближение), а на конечном – метод самонаведения (управление ориентацией вектора скорости, метод переменного упреждения, метод пропорциональной навигации (МПН), наведение в мгновенную точку встречи и т. д.). Формирование конкретной структуры алгоритма управления методов осуществляется на этапе синтеза закона наведения.

В зависимости от используемых уравнений кинематики движения (уравнений состояний), записанных для декартовой либо сферической СК, матрица связи измерений и наблюдений будет иметь различный вид. Так, для уравнений измерений (6) матрица связи наблюдений и измерений (матрица Якоби от вектор-функции первичных измерений) запишется следующим образом:

$$\begin{pmatrix} \frac{(\mathbf{r}_0 - \Delta \mathbf{r}_1)^T \frac{\partial \mathbf{r}_0}{\partial \alpha_0}}{|\mathbf{r}_0 - \Delta \mathbf{r}_1|} - \frac{(\mathbf{r}_0 - \Delta \mathbf{r}_2)^T \frac{\partial \mathbf{r}_0}{\partial \alpha_0}}{|\mathbf{r}_0 - \Delta \mathbf{r}_2|} & \dots & \frac{(\mathbf{r}_0 - \Delta \mathbf{r}_1)^T \frac{\partial \mathbf{r}_0}{\partial \alpha_2}}{|\mathbf{r}_0 - \Delta \mathbf{r}_1|} - \frac{(\mathbf{r}_0 - \Delta \mathbf{r}_2)^T \frac{\partial \mathbf{r}_0}{\partial \alpha_2}}{|\mathbf{r}_0 - \Delta \mathbf{r}_2|} \\ \dots & \frac{(\mathbf{r}_0 - \Delta \mathbf{r}_1)^T \frac{\partial \mathbf{r}_0}{\partial \alpha_1}}{|\mathbf{r}_0 - \Delta \mathbf{r}_1|} - \frac{(\mathbf{r}_0 - \Delta \mathbf{r}_3)^T \frac{\partial \mathbf{r}_0}{\partial \alpha_1}}{|\mathbf{r}_0 - \Delta \mathbf{r}_3|} & \dots \\ \frac{(\mathbf{r}_0 - \Delta \mathbf{r}_1)^T \frac{\partial \mathbf{r}_0}{\partial \alpha_0}}{|\mathbf{r}_0 - \Delta \mathbf{r}_1|} - \frac{(\mathbf{r}_0 - \Delta \mathbf{r}_4)^T \frac{\partial \mathbf{r}_0}{\partial \alpha_0}}{|\mathbf{r}_0 - \Delta \mathbf{r}_4|} & \dots & \frac{(\mathbf{r}_0 - \Delta \mathbf{r}_1)^T \frac{\partial \mathbf{r}_0}{\partial \alpha_2}}{|\mathbf{r}_0 - \Delta \mathbf{r}_1|} - \frac{(\mathbf{r}_0 - \Delta \mathbf{r}_4)^T \frac{\partial \mathbf{r}_0}{\partial \alpha_2}}{|\mathbf{r}_0 - \Delta \mathbf{r}_4|} \end{pmatrix}. \quad (16)$$

При этом элементы матрицы связи наблюдений и состояний будут определяться выражениями

$$\begin{aligned}\mathbf{r}_0 &= (x \ y \ z)^T = (\rho \cos(\varepsilon) \cos(\beta) \ \rho \sin(\varepsilon) \ - \ \rho \cos(\varepsilon) \sin(\beta))^T, \\ \mathbf{a} = (x \ y \ z)^T, \quad \frac{\partial \mathbf{r}_0}{\partial \mathbf{a}} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{a} = (\varepsilon \ \beta \ \rho)^T, \quad \frac{\partial \mathbf{r}_0}{\partial \mathbf{a}} = \begin{pmatrix} \frac{\partial \mathbf{r}_{0x}}{\partial \varepsilon} & \frac{\partial \mathbf{r}_{0x}}{\partial \beta} & \frac{\partial \mathbf{r}_{0x}}{\partial \rho} \\ \frac{\partial \mathbf{r}_{0y}}{\partial \varepsilon} & \frac{\partial \mathbf{r}_{0y}}{\partial \beta} & \frac{\partial \mathbf{r}_{0y}}{\partial \rho} \\ \frac{\partial \mathbf{r}_{0z}}{\partial \varepsilon} & \frac{\partial \mathbf{r}_{0z}}{\partial \beta} & \frac{\partial \mathbf{r}_{0z}}{\partial \rho} \end{pmatrix},\end{aligned}\quad (17)$$

где $\mathbf{a} = (x \ y \ z)^T$ – вектор состояний при использовании разностно-дальномерных измерений и ДСК; $\mathbf{a} = (\varepsilon \ \beta \ \rho)^T$ – вектор состояний при использовании разностно-дальномерных измерений и СфСК.

Синтез алгоритма управления нелинейным динамическим объектом. Сформулируем задачу синтеза алгоритма наведения следующим образом. Необходимо найти аналитический

закон формирования команд управления, который обеспечивает наведение ОУ по заданной траектории (с желаемыми угловыми координатами и скоростями), программной или динамически изменяющейся в зависимости от параметров движения цели, в выбранной СК на первой части участка наведения. На второй части траектории следует обеспечить плавный переход на наведение по методу пропорциональной навигации.

Для синтеза используем методику аналитического конструирования агрегированных регуляторов (АКАР), основные положения которой изложены в работах [3, 4]. Прикладной пример аналитического синтеза модифицированного МПН в замкнутой форме с применением указанного подхода приведен в статье [5].

В качестве критерия оптимальности будем применять комбинированный критерий: на начальном участке наведения – обеспечение равенства углов ориентации РЛС – ОУ (либо линии относительной дальности ОУ – цель) заданному значению, на конечном участке наведения – поддержание нулевой УСВ линии визирования (относительной дальности) ОУ – цель. Такой подход позволит обеспечить удержание наводимого ОУ в заданном секторе углов относительно нормали к плоскости измерений (при ограниченном секторе сопровождения объекта), образованной тремя позициями воздушного базирования на базе БПЛА. Кроме того, выбранный подход гарантирует минимизацию флуктуационных нормальных ускорений ОУ при наведении на движущуюся цель, в том числе в условиях активных шумовых помех.

Приступим к процедуре синтеза комбинированного управления для ОУ. Сначала получим закон оптимального управления по заданным координатам объекта с использованием уравнений кинематики в СфСК (15). Для этого последовательно определим первую и вторую векторные макропеременные и соответствующее им векторное инвариантное многообразие (ИМ):

$$\Psi_1(\mathbf{x}) = \begin{pmatrix} \Psi_{1z}(\mathbf{x}) \\ \Psi_{1y}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \dot{\omega}_z(\mathbf{x}) - f_{1z}(\mathbf{x}) \\ \dot{\omega}_y(\mathbf{x}) - f_{1y}(\mathbf{x}) \end{pmatrix}, \quad \Psi_2(\mathbf{x}) = \begin{pmatrix} \Psi_{2z}(\mathbf{x}) \\ \Psi_{2y}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \omega_z - f_{2z}(\mathbf{x}) \\ \frac{\omega_y}{\cos(\varepsilon)} - f_{2y}(\mathbf{x}) \end{pmatrix}, \quad (18)$$

где $\Psi_1(\mathbf{x}), \Psi_2(\mathbf{x})$ – ИМ первого и второго этапов редукции соответственно; $f_{1y}(\mathbf{x}), f_{1z}(\mathbf{x})$ – полуопределенные скалярные функции векторной переменной, играющие роль внутренних (промежуточных) управлений для редуцированной системы уравнений, конкретный вид которых определяется после получения редуцированной системы уравнений кинематики желаемого вида (вместо переменных $\dot{\omega}_z(\mathbf{x}), \dot{\omega}_y(\mathbf{x})$ используются выражения для правых частей соответствующих уравнений); $f_{2y}(\mathbf{x}), f_{2z}(\mathbf{x})$ – функции, определяющие вид правой части редуцированной СДУ (желаемую динамику системы) относительно координат ε, β после второго этапа преобразования.

В качестве функционального уравнения, определяющего динамику макропеременной при движении изображающей точки к многообразию $\Psi(\mathbf{x}) = 0$, для векторного экспоненциального процесса используем типовое уравнение

$$\dot{\Psi}_1 + \alpha_1 \Psi_1 = 0, \quad \Psi_1 = \begin{pmatrix} \Psi_{1z}(\mathbf{x}) \\ \Psi_{1y}(\mathbf{x}) \end{pmatrix}, \quad (19)$$

где α_1 – коэффициент сходимости макропеременной к установившемуся значению; $\mathbf{x} = (\varepsilon \beta \omega_z \omega_y W_y W_z)^T$ – вектор фазовых координат системы уравнений (15).

Выполняя подстановку (18), (19) в (15) и выражая искомый вектор управления $\mathbf{U}(\mathbf{x}) = (U_y U_z)^T$, получим полуопределенную редуцированную систему

$$\begin{aligned} \dot{\varepsilon}_0 &= \omega_{z0}, \quad \dot{\beta}_0 = \frac{\omega_{y0}}{\cos(\varepsilon_0)}, \\ \dot{\omega}_{z0} &= f_{1z}(\mathbf{x}), \quad \dot{\omega}_{y0} = f_{1y}(\mathbf{x}). \end{aligned} \quad (20)$$

Запишем соотношения для расчета команд управления:

$$\mathbf{U}(\mathbf{x}) = \begin{pmatrix} U_y(\mathbf{x}) \\ U_z(\mathbf{x}) \end{pmatrix} = \frac{r_0 \alpha_1}{\alpha_p} \Psi_1(\mathbf{x}) + \frac{r_0}{\alpha_p} \Phi_1(\mathbf{x}) - \Phi_2(\mathbf{x}),$$

$$\Phi_1(\mathbf{x}) = \begin{pmatrix} \Phi_{1y}(\mathbf{x}) \\ \Phi_{1z}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \left(\sum_{j=0}^{J-1} \frac{\partial f_{1z}}{\partial x_j} \frac{dx_j}{dt} + \frac{\partial f_{1z}}{\partial t} \right) + \omega_y^2 (1 + \operatorname{tg}(\varepsilon))^2 \dot{\varepsilon} - 2\omega_y \operatorname{tg}(\varepsilon) \dot{\omega}_y \\ - \left(\sum_{j=0}^{J-1} \frac{\partial f_{1y}}{\partial x_j} \frac{dx_j}{dt} + \frac{\partial f_{1y}}{\partial t} \right) - 2v_0 (\omega_z \dot{\omega}_y + \omega_y \dot{\omega}_z) \operatorname{tg}(\varepsilon) - \omega_y \omega_z (1 + \operatorname{tg}(\varepsilon))^2 \dot{\varepsilon} \end{pmatrix}; \quad (21)$$

$$\Phi_2(\mathbf{x}) = \begin{pmatrix} \Phi_{2y}(\mathbf{x}) \\ \Phi_{2z}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \frac{V_0}{r_0 \alpha_p} \left(W_0 \operatorname{th} \left(\frac{W_y}{W_0} \right) - 2V_0 \omega_z + W_{yc} \right) + W_y \left(1 - \operatorname{th} \left(\frac{W_y}{W_0} \right) \right)^2 + \frac{2(\dot{V}_0 \omega_z + V_0 \dot{\omega}_z) + \dot{W}_{yc}}{\alpha_p} \\ \frac{V_0}{r_0 \alpha_p} \left(W_0 \operatorname{th} \left(\frac{W_z}{W_0} \right) - 2V_0 \omega_y + W_{zc} \right) + W_z \left(1 - \operatorname{th} \left(\frac{W_z}{W_0} \right) \right)^2 - \frac{2(\dot{V}_0 \omega_y + V_0 \dot{\omega}_y) + \dot{W}_{zc}}{\alpha_p} \end{pmatrix}, \quad (22)$$

где $\frac{\partial f_{1z}}{\partial x_j}$, $\frac{\partial f_{1y}}{\partial x_j}$ – частные производные внутреннего (промежуточного) управления по фазовым координатам; $\frac{\partial f_{1z}}{\partial t}$, $\frac{dx_j}{dt}$ – частные производные функции внутреннего (промежуточного) управления и фазовой координаты по времени. Вводя второе ИМ (18), получим соотношение для внутреннего управления $\mathbf{f}_1(\mathbf{x})$:

$$\mathbf{f}_1(\mathbf{x}) = \begin{pmatrix} f_{1z}(\mathbf{x}) \\ f_{1y}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \frac{\partial f_{2z}}{\partial \varepsilon} \omega_z + \frac{\partial f_{2z}}{\partial t} - \alpha_2 (\omega_z - f_{2z}) \\ \frac{\partial f_{2y}}{\partial \beta} \omega_y + \frac{\partial f_{2y}}{\partial t} \cos(\varepsilon) - \alpha_2 (\omega_z - f_{2y} \cos(\varepsilon)) - \omega_y \omega_z \tan(\varepsilon) \end{pmatrix}. \quad (23)$$

Выберем в качестве функций $f_{2z}(\mathbf{x})$, $f_{2y}(\mathbf{x})$ соотношения

$$\begin{aligned} f_{2z}(\varepsilon) &= \mu(t)(\varepsilon_\alpha - \varepsilon) + \gamma(t)(\varepsilon_\alpha - \varepsilon)^3, \\ f_{2y}(\beta) &= \mu(t)(\beta_\alpha - \beta) + \gamma(t)(\beta_\alpha - \beta)^3, \end{aligned} \quad (24)$$

где ε_α , β_α – кинематические (заданные) координаты, получаемые из условия наведения объекта в заданную точку встречи с учетом имеющихся ограничений на углы отклонения относительно цели; $\mu(t)$, $\gamma(t)$ – зависящие от времени коэффициенты сходимости, обратно пропорциональные времени сходимости. Введение степенного слагаемого позволяет снизить время отработки ошибки наведения при значительных отклонениях.

Редуцированная СДУ примет экспоненциальную форму с нелинейной коррекцией за счет наличия дополнительного слагаемого:

$$\begin{aligned} \dot{\varepsilon} &= \mu(t)(\varepsilon_\alpha - \varepsilon) + \gamma(t)(\varepsilon_\alpha - \varepsilon)^3, \\ \dot{\beta} &= \mu(t)(\beta_\alpha - \beta) + \gamma(t)(\beta_\alpha - \beta)^3. \end{aligned} \quad (25)$$

Уравнения системы (25) независимы, асимптотически устойчивы и сходятся к опорным значениям ε_α , β_α при $\mu(t)$, $\gamma(t) > 0$ за время $\tau(t) \leq \frac{3}{\mu(t)}$.

Запишем уравнение (23), используя выражения для частных производных с учетом (24):

$$\begin{pmatrix} f_{1z}(\varepsilon, \beta) \\ f_{1y}(\varepsilon, \beta) \end{pmatrix} = \begin{pmatrix} -(\mu(t) + 3\gamma(t)(\varepsilon_\alpha - \varepsilon)^2)\omega_z - \alpha_2(\omega_z - f_{2z}) + \frac{\partial f_{2z}}{\partial t} \\ -(\mu(t) + 3\gamma(t)(\varepsilon_\alpha - \varepsilon)^2)\frac{\omega_y}{\cos(\varepsilon)} - \alpha_2(\omega_y - f_{2y}\cos(\varepsilon)) - \omega_y\omega_z \operatorname{tg}(\varepsilon) + \frac{\partial f_{2z}}{\partial t}\cos(\varepsilon) \end{pmatrix}, \quad (26)$$

$$\frac{\partial f_{2z}}{\partial t} = (\dot{\mu}(t) + \dot{\gamma}(t)(\varepsilon_\alpha - \varepsilon)^2)(\varepsilon_\alpha - \varepsilon) + (\mu(t) + 3\gamma(t)(\varepsilon_\alpha - \varepsilon)^2)\omega_{z\alpha},$$

$$\frac{\partial f_{2y}}{\partial t} = (\dot{\mu}(t) + \dot{\gamma}(t)(\beta_\alpha - \beta)^2)(\beta_\alpha - \beta) + (\mu(t) + 3\gamma(t)(\beta_\alpha - \beta)^2)\frac{\omega_{y\alpha}}{\cos(\varepsilon)},$$

где $\mu(t), \gamma(t)$ – производные коэффициентов сходимости по времени. Частные производные функции внутреннего управления ввиду их громоздкости не приведены.

Опорные (кинематические) координаты $(\varepsilon_\alpha, \beta_\alpha), (\omega_{z\alpha}, \omega_{y\alpha})$ из соотношения (26) могут быть получены несколькими способами:

- расчетом по методу полного или частичного спрямления траектории полета с использованием переменного коэффициента упреждения;
- расчетом по результатам решения задачи о встрече ракеты с целью на основе координат упрежденной точки встречи, оцениваемых темпом формирования команд управления.

В первом случае расчет выполняется согласно формулам

$$\begin{aligned} \varepsilon_\alpha &= \varepsilon_0 + \omega_{z0}\tau_f, \quad \hat{\varepsilon}_\alpha = \hat{\varepsilon}_\alpha + \frac{\Delta\tau}{T_\phi}\hat{\omega}_{z\alpha}, \quad \hat{\omega}_{z\alpha} = \frac{(\varepsilon_\alpha - \hat{\varepsilon}_\alpha)}{\Delta\tau}, \\ \beta_\alpha &= \beta_0 + \omega_{y0}\tau_f, \quad \hat{\beta}_\alpha = \hat{\beta}_\alpha + \frac{\Delta\tau}{T_\phi}\frac{\hat{\omega}_{y\alpha}}{\cos(\varepsilon_\alpha)}, \quad \hat{\omega}_{y\alpha} = \frac{(\beta_\alpha - \hat{\beta}_\alpha)}{\Delta\tau}\cos(\varepsilon_\alpha), \end{aligned} \quad (27)$$

где $\Delta\tau$ – шаг интегрирования сглаживающего фильтра; T_ϕ – постоянная времени сглаживающего фильтра.

Во втором случае кинематические угловые координаты и УСВ линии визирования РЛС – цель рассчитываются на основе информации о параметрах встречи управляемого боеприпаса и цели. Это возможно реализовать путем решения в ускоренном времени упрощенной СДУ, описывающей движение объекта и цели в связанной с РЛС системе координат. Более простой подход предполагает использование оценки времени, которое осталось до встречи с целью, полученной из решения квадратного уравнения

$$\Delta\mathbf{v}_0^T \cdot \Delta\mathbf{r}(\hat{\tau}) = \Delta\mathbf{v}_0 \left(\Delta\mathbf{r}_0 + \Delta\mathbf{v}_0\tau + 0,5\Delta\mathbf{a}_0\tau^2 \right) = 0, \quad (28)$$

где $\Delta\mathbf{a}_0$ – оценка разности векторов ускорений ракеты и цели в СК РЛС; $\Delta\mathbf{v}_0$ – оценка разности векторов скоростей ракеты и цели в СК РЛС; $\Delta\mathbf{r}_0$ – начальная оценка разности координат ракеты и цели в СК РЛС.

Сделаем промежуточные выводы. Во-первых, с использованием методологии АКАР (синергетической теории управления) получены соотношения (21)–(25), реализующие закон оптимального наведения объекта по информации от наземной РЛС или другого измерительного устройства, например МП РЛС наземного или воздушного базирования либо БПЛА. Данный закон наведения является обобщением известных методов телеуправления ракетой (снарядом) при перехвате движущейся цели. Во-вторых, аналогичную методику с некоторыми особенностями можно применять при разработке модифицированного метода параллельного сближения с использованием для оценки опорных углов наклона линии визирования информации, поступающей от алгоритма решения задачи о встрече снаряда и цели.

Рассмотрим вариант синтеза обобщенного закона пропорциональной навигации, который будет использоваться на заключительном этапе наведения ОУ. Синтез будем проводить в пред-

положении о возможности получения достоверных оценок координат объекта и цели, а также их первых производных, применяемых в уравнениях кинематики движения (14).

Согласно методологии АКАР введем первую векторную макропеременную $\Psi_1(\mathbf{x})$, подставим ее в векторное уравнение (19) и выразим из полученных соотношений управление:

$$\dot{\Psi}_1 + \alpha_1 \Psi_1 = 0, \Psi_1 = \begin{pmatrix} \Psi_{1y}(\mathbf{x}) \\ \Psi_{1z}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} f_{1y}(\mathbf{x}) - W_0 \operatorname{th}\left(\frac{W_y}{W_0}\right) \\ f_{1z}(\mathbf{x}) - W_0 \operatorname{th}\left(\frac{W_z}{W_0}\right) \end{pmatrix}, \quad (29)$$

где $f_{1y}(\mathbf{x}), f_{1z}(\mathbf{x})$ – полуопределенная функция, используемая для реализации управления. В результате получим выражение

$$\mathbf{U}(\mathbf{x}) = \begin{pmatrix} U_y(\mathbf{x}) \\ U_z(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \frac{1}{\alpha_p} \left(\nabla_x f_{y1}(\mathbf{x}) + \frac{\partial f_{y1}(t)}{\partial t} \right) + W_y \left(1 - \operatorname{th}\left(\frac{W_y}{W_0}\right)^2 \right) \\ \frac{1}{\alpha_p} \left(\nabla_x f_{z1}(\mathbf{x}) + \frac{\partial f_{z1}(t)}{\partial t} \right) + W_z \left(1 - \operatorname{th}\left(\frac{W_z}{W_0}\right)^2 \right) \end{pmatrix} + \frac{\alpha_2}{\alpha_p} \begin{pmatrix} f_{y1}(t) - W_0 \operatorname{th}\left(\frac{W_y}{W_0}\right) \\ f_{z1}(t) - W_0 \operatorname{th}\left(\frac{W_z}{W_0}\right) \end{pmatrix}. \quad (30)$$

Векторная функция $f_1(t, \mathbf{x})$ выбирается из условия обеспечения требуемой динамики изменения координат ω_z, ω_y , заданных соответствующими дифференциальными уравнениями. Данное условие записывается в виде

$$\begin{pmatrix} -\left(\mu(t)(\omega_z + k_\psi \omega_{zc}) \right) \\ -\left(\mu(t)(\omega_y + k_\psi \omega_{yc}) \right) \end{pmatrix} = \begin{pmatrix} \frac{f_{y1}(\mathbf{x}) - 2v_0 \omega_z + (1 - k_\psi) W_{yc} - \omega_y^2 \operatorname{tg}(\theta)}{D_0} \\ \frac{-f_{z1}(\mathbf{x}) - 2v_0 \omega_y + (1 - k_\psi) W_{zc} + \omega_y \omega_z \operatorname{tg}(\theta)}{D_0} \end{pmatrix}, \quad (31)$$

где ω_{yc}, ω_{zc} – компенсирующие поправки по УСВ ЛВ, учитывающие влияние силы тяжести, продольного ускорения снаряда и маневра цели; W_{yc}, W_{zc} – компенсирующие поправки по нормальному ускорению, учитывающие влияние силы тяжести, продольного ускорения ОУ и ускорения маневра цели; $k_\psi \in (0, 1)$ – коэффициент для выбора способа компенсации возмущающих воздействий, принимающий дискретные значения 0 или 1.

Выражая компоненты функции внутреннего управления $\mathbf{f}_1(\mathbf{x})$, получим формулу

$$\mathbf{f}_1(\mathbf{x}) = \begin{pmatrix} f_{y1}(\mathbf{x}) \\ f_{z1}(\mathbf{x}) \end{pmatrix} = D_0 \begin{pmatrix} -\mu(t)(\omega_z + k_\psi \omega_{zc}) + \omega_y^2 \operatorname{tg}(\theta) \\ \mu(t)(\omega_y + k_\psi \omega_{yc}) + \omega_y \omega_z \operatorname{tg}(\theta) \end{pmatrix} + \begin{pmatrix} 2v_0 \omega_z - (1 - k_\psi) w_{yc} \\ -2v_0 \omega_y - (1 - k_\psi) w_{zc} \end{pmatrix}. \quad (32)$$

Производная функции внутреннего управления $\mathbf{f}_1(\mathbf{x})$ по времени имеет вид

$$\dot{\mathbf{f}}_1(\mathbf{x}) = \begin{pmatrix} \nabla_x f_{y1}(\mathbf{x}) + \frac{\partial f_{y1}(t)}{\partial t} \\ \nabla_x f_{z1}(\mathbf{x}) + \frac{\partial f_{z1}(t)}{\partial t} \end{pmatrix} = \begin{pmatrix} v_0 (-\mu(t)(\omega_z + k_\psi \omega_{zc}) + \omega_y^2 \operatorname{tg}(\theta)) + \varphi_{1y}(\mathbf{x}) \\ v_0 (-\mu(t)(\omega_y + k_\psi \omega_{yc}) + \omega_y \omega_z \operatorname{tg}(\theta)) + \varphi_{1z}(\mathbf{x}) \end{pmatrix} + \begin{pmatrix} \varphi_{2y}(\mathbf{x}) \\ \varphi_{2z}(\mathbf{x}) \end{pmatrix}, \quad (33)$$

$$\begin{pmatrix} \varphi_{1y}(\mathbf{x}) \\ \varphi_{1z}(\mathbf{x}) \end{pmatrix} = D_0 \begin{pmatrix} -\mu(t)(\dot{\omega}_z - k_\psi \dot{\omega}_{zc}) - \dot{\mu}(t)(\omega_z - k_\psi \omega_{zc}) + 2\omega_y \dot{\omega}_y \operatorname{tg}(\theta) + \omega_z \omega_y^2 (1 + \operatorname{tg}(\theta)^2) \\ \mu(t)(\dot{\omega}_y - k_\psi \dot{\omega}_{yc}) + \dot{\mu}(t)(\omega_y - k_\psi \omega_{yc}) + (\dot{\omega}_y \omega_z + \omega_y \dot{\omega}_z) \operatorname{tg}(\theta) + \omega_y \omega_z^2 (1 + \operatorname{tg}(\theta)^2) \end{pmatrix},$$

$$\begin{pmatrix} \varphi_{2y}(\mathbf{x}) \\ \varphi_{2z}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} 2(\dot{v}_0 \omega_z + v_0 \dot{\omega}_z) - (1 - k_\psi) \delta w_{yc} \\ -2(\dot{v}_0 \omega_y + v_0 \dot{\omega}_y) - (1 - k_\psi) \delta w_{zc} \end{pmatrix}.$$

Производные по времени для сигналов компенсации принимаются равными нулю либо оцениваются с использованием сглаживающих фильтров первого порядка. Закон изменения коэффициента сходимости $\mu(t)$ задается соотношением

$$\mu(t) = \frac{\mu_0}{\tau_f(t)}, \quad \mu(t) = \mu_0 \left(1 - 0,9 \exp\left(-\frac{t}{0,3\tau_0}\right) \right), \quad (34)$$

где $\tau_f(t)$ – время, оставшееся до встречи снаряда с целью; τ_0 – оценка полетного времени, полученная на момент начала управления снарядом; $\mu_0 = \mu(0)$ – значение коэффициента сходимости в момент старта.

Сопряжение методов наведения, описываемых соотношениями (21) и (30), может выполняться различными способами. Один из них заключается в плавном переходе с управления (21) на управление (30) с использованием алгоритма

$$\mathbf{U}_0(\mathbf{x}) = (1 - k_x(\Delta\Phi))\mathbf{U}_1(\mathbf{x}) + k_x(\Delta\Phi)\mathbf{U}_2(\mathbf{x}), \quad k_x(\Delta\Phi) = \exp\left(-\frac{\Delta\Phi}{\Delta\Phi_0}\right), \quad (35)$$

где $\mathbf{U}_1(\mathbf{x}), \mathbf{U}_2(\mathbf{x})$ – вектор управления, реализованного по модифицированному методу полного спрямления (ММПС) траектории и модифицированному (обобщенному) методу пропорциональной навигации (ММПН) соответственно; $\Delta\Phi$ – разность между текущей относительной угловой координатой снаряда и граничным углом сектора сопровождения; $\Delta\Phi_0$ – константа, выбираемая из условия полного перехода на управление $\mathbf{U}_1(\mathbf{x})$ при нахождении снаряда на границе сектора сопровождения.

Математическое моделирование и оценка статистических характеристик. Для проведения статистического анализа была разработана и реализована математическая модель контура управления ОУ, в основу которого положена система уравнений динамики материальной точки, устанавливающая связь между нормальными перегрузками, действующими на объект, и его сферическими (декартовыми) координатами (скоростями изменения координат). В качестве методов наведения использовались:

- ММПС, или метод переменного упреждения, описываемый выражениями (21)–(25);
- ММПН, или метод наведения в фактическую (мгновенную) точку встречи, описываемый выражениями (30)–(34);
- комбинированный МПН (КМПН), использующий наведение по ММПС с заданными углами упреждения на начальном участке траектории и наведение по ММПН на конечном участке при малых углах упреждения.

Для оценивания координат использовались измерители координат цели и ОУ, реализованные с помощью уравнений оптимальной нелинейной фильтрации (10)–(12). Шумы измерений задавались в виде нормального гауссовского шума с нулевым математическим ожиданием и среднеквадратическим отклонением (СКО) флуктуационной ошибки измерения УК цели в диапазоне $\sigma_\varphi = (2-6)$ угл. мин, СКО измерения дальности цели – в пределах $\sigma_r = (1-3)$ м, СКО скорости цели – в пределах $\sigma_v = (2-4)$ м/с. На рис. 1–3 показаны временные реализации координат и некоторых параметров цели и ОУ, характеризующие динамику наведения.

Координаты встречи составили $r_m = (11\,200, 17\,500, 14\,500)^T$ и $r_m = (5000, 14\,000, 14\,000)^T$ при начальной скорости цели $V_u = 500$ м/с. Оценивались математическое ожидание и СКО величины мгновенного промаха σ_γ, m_γ и ошибки наведения m_ζ, σ_ζ (линейного расстояния между ракетой и целью) в картинной плоскости.

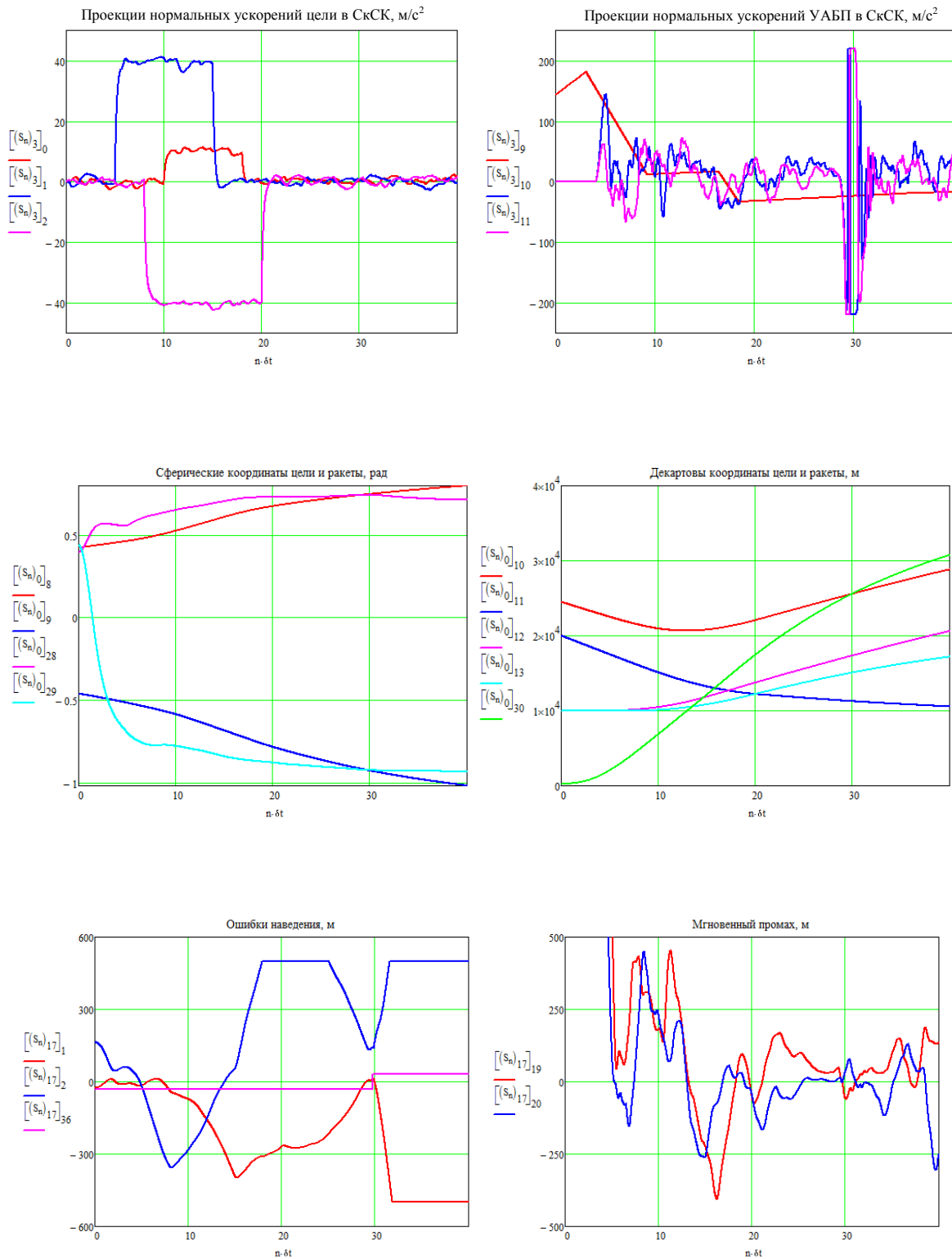


Рис. 1. Координаты и параметры управления ОУ при наведении ММПН на маневрирующую цель без учета ограничений сектора сопровождения

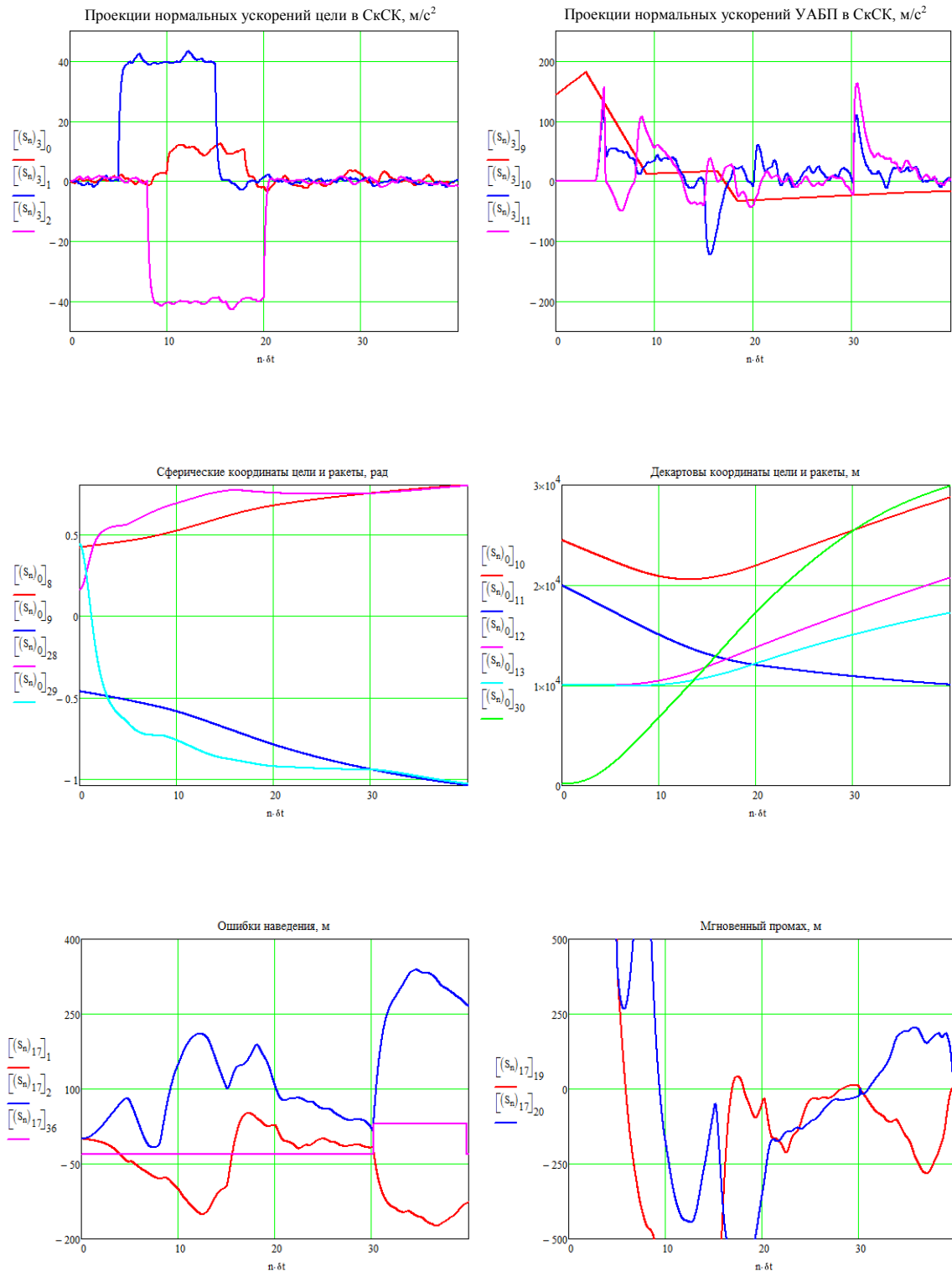


Рис. 2. Координаты и параметры управления ОУ при наведении ММПС на маневрирующую цель без учета ограничений сектора сопровождения

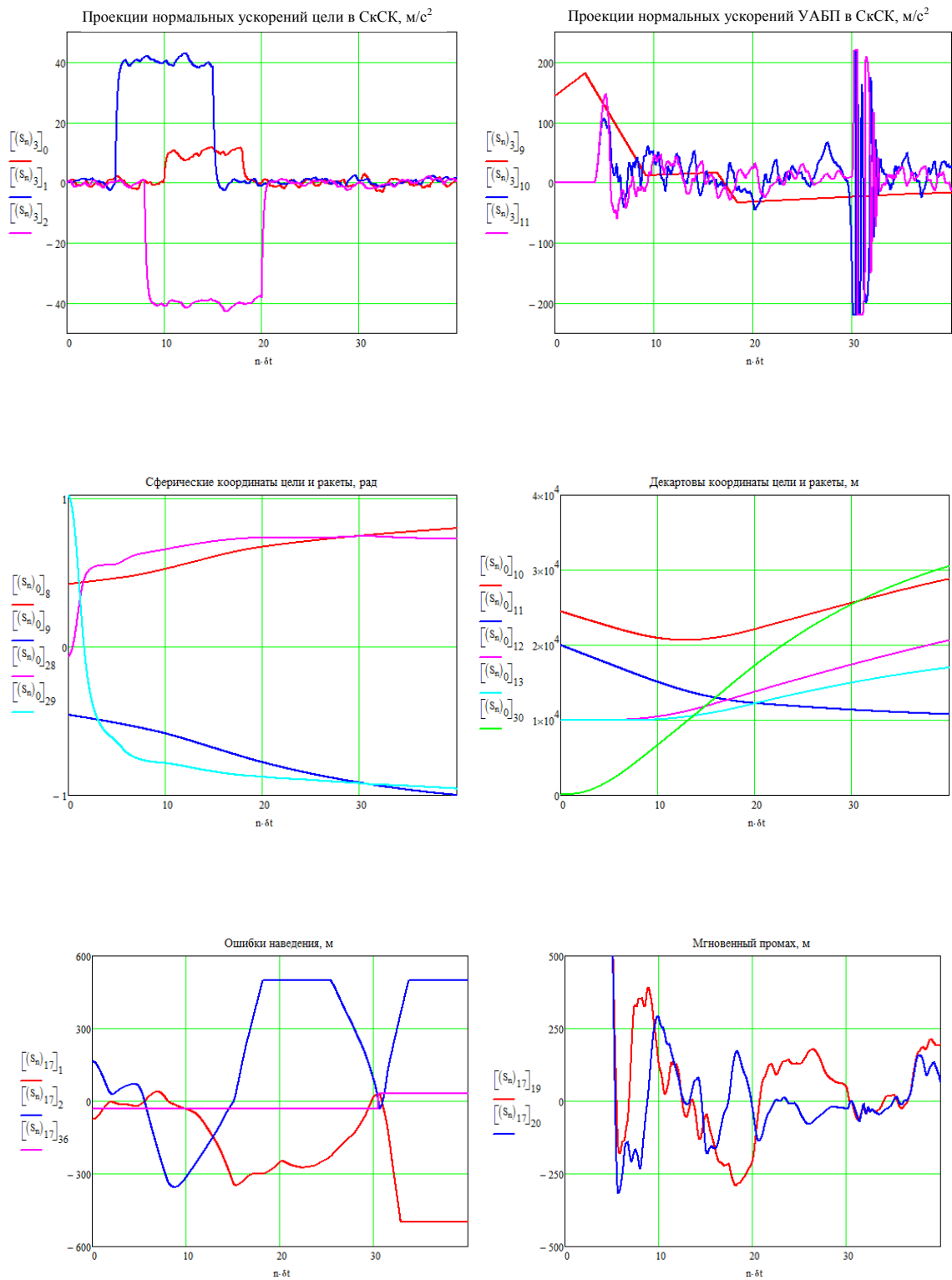


Рис. 3. Координаты и параметры управления ОУ при наведении КМПН на маневрирующую цель с учетом ограничений сектора сопровождения

Сравнительный анализ результатов моделирования, приведенных в таблице, позволяет сделать следующие выводы:

– оценки точности наведения на маневрирующую цель для ММПС, ММПН и КМПН отличаются незначительно, при росте интенсивности флуктуаций по каналу цели предпочтительно использовать ММПН или КМПН как менее чувствительные к флуктуационным ошибкам измерения координат, особенно на заключительном участке траектории;

– для перехвата скоростной цели, летящей на больших высотах и предельном параметре, необходимо использовать КМПН, позволяющий получить приемлемые ошибки наведения при допустимых углах упреждения относительно линии визирования цели;

– при наведении ОУ на маневрирующую цель и при отсутствии возможности оценивания нормальных ускорений цели предпочтение следует отдавать ММПС как менее чувствительному к величине ускорения маневра и его продолжительности.

Ошибки наведения и мгновенный промах ОУ в точке встречи

Вариант расчета	m_ζ, σ_ζ для $\zeta_0, \text{ м}$			m_γ, σ_γ для $\gamma_0, \text{ м}$		
	ММПС	ММПН	КМПН	ММПС	ММПН	КМПН
1	28; 12	25; 8	24; 9	16; 9	12; 7	13; 8
2	18; 11	17; 9	21; 10	14; 8	11; 6	12; 7

Заключение. В работе выполнен краткий обзор существующих и перспективных способов наведения управляемых авиационных (артиллерийских) беспилотных перехватчиков на движущуюся цель. Обоснована необходимость исследования и разработки автономных средств управления наведением ОУ без участия оператора с использованием МП РЛС воздушного базирования.

В качестве основы перспективного алгоритма управления вектором скорости ОУ использован метод пропорциональной навигации, а также его модификации, учитывающие сигналы компенсации возмущающих воздействий.

Обоснован формализованный алгебраический подход для синтеза нелинейных многомерных многосвязных нестационарных оптимальных регуляторов на основе метода АКАР [3, 4].

В ходе исследования получена математическая модель ОУ, описываемая нелинейными кинематическими дифференциальными уравнениями аэродинамического объекта, учитывающая особенности технической реализации объекта и используемая для синтеза структуры оптимального регулятора.

В замкнутой алгебраической форме приведены аналитические выражения для реализации КМПН, обеспечивающего нахождение объекта в пределах заданного сектора сопровождения: на начальном участке при использовании ММПС и на конечном участке – при использовании ММПН. Данные соотношения могут быть реализованы в составе специализированного цифрового вычислителя, входящего в состав бортового оборудования и имеющего ограниченные вычислительные ресурсы.

Список использованных источников

1. Мицель, А. А. Методы оптимизации : учеб. пособие / А. А. Мицель, А. А. Шелестов. – Томск : Томский МЦДО, 2002. – Ч. 1. – 192 с.
2. Марков, Л. Н. Основы построения ЗРК / Л. Н. Марков. – Минск : МВИЗРУ, 1980. – 162 с.
3. Колесников, А. А. Проектирование многокритериальных систем управления промышленными объектами / А. А. Колесников, А. Г. Гельфгат. – М. : Энергоатомиздат, 1993. – 304 с.
4. Колесников А. А. Синергетическая теория управления. Инварианты, оптимизация, синтез / А. А. Колесников. – М. : Энергоатомиздат, Таганрог : Изд-во ТРТУ, 1994. – 344 с.
5. Маркевич, В. Э. Аналитический синтез метода наведения сверхзвукового беспилотного летательного аппарата на основе многомерной нелинейной динамической модели / В. Э. Маркевич // Информатика. – 2017. – № 2(54). – С. 93–112.

References

1. Mizel A. A., Shelestov A. A. Metody optimizacii. *Optimization Methods. Part 1*. Tomsk, Tomskij mezhvuzovskij centr distancionnogo obrazovanija, 2002, chast' 1, 192 p. (in Russian).
2. Markov L. N. Osnovy postroenija ZRK. *Fundamentals of the Design of Antiaircraft Systems*. Minsk, Minskoe vysshee inzhenernoe zenitnoe raketnoe uchilishhe, 1980, 162 p. (in Russian).
3. Kolesnikov A. A., Gelfgat A. G. Proektirovanie mnogokriterial'nyh sistem upravlenija promyshlennymi ob"ektami. *Designing Multi-criteria Control Systems for Industrial Facilities*. Moscow, Energoatomizdat, 1993, 304 p. (in Russian).
4. Kolesnikov A. A. Sinergeticheskaja teorija upravlenija. Invarianty, optimizacija, sintez. *Synergetic Theory of Control. Invariants, Optimization, Synthesis*. Moscow, Energoatomizdat, Taganrog, Izdatel'stvo Taganrogskogo gosudarstvennogo radiotekhnicheskogo universiteta, 1994, 344 p. (in Russian).
5. Markevich V. E. Analiticheskij sintez metoda navedenija sverhzvukovogo bespilotnogo letatel'nogo apparata na osnove mnogomernoj nelinejnoj dinamicheskoj modeli. *Analytical Synthesis of a Method of Guidance of the Supersonic Unmanned Aerial Vehicle on the Basis of Multidimensional Nonlinear Dynamic Model*. Informatika [*Informatics*], 2017, no. 2(54), pp. 93–112 (in Russian).

Информация об авторах

Маркевич Виталий Эдмундович, кандидат технических наук, ведущий научный сотрудник научно-исследовательского сектора, ОАО «АЛЕВКУРП», Минск, Беларусь.

E-mail: mark.vit@tut.by

Легкоступ Виктор Валерьевич, аспирант Белорусского государственного университета информатики и радиоэлектроники, младший научный сотрудник научно-исследовательского сектора, ОАО «АЛЕВКУРП», Минск, Беларусь.

E-mail: legkostupvv@gmail.com

Information about the authors

Vitaliy E. Markevich, Cand. Sci. (Eng.), Leading Researcher, OJSC "ALEVKURP", Minsk, Belarus.

E-mail: mark.vit@tut.by

Victor V. Legkostup, Postgraduate Student of Belarusian State University of Informatics and Radioelectronics, Junior Researcher, OJSC "ALEVKURP", Minsk, Belarus.

E-mail: legkostupvv@gmail.com

Правила для авторов

Редакция журнала «Информатика» просит авторов руководствоваться приведенными ниже правилами:

1. Статьи принимаются в редакцию через электронную систему подачи по адресу <http://inf.grid.by> в формате файлов текстовых редакторов Microsoft Word. Основной текст статьи не должен превышать 17 стр., включая рисунки, таблицы и достаточное количество наиболее актуальных ссылок; обзорной статьи – 10 стр., включая все основные ссылки. Текст набирается с переносами, шрифт Times New Roman 11 пт, интервал между строками одинарный, абзацный отступ 0,5 см, поля по 2,5 см со всех сторон.

Изложенный в статье материал должен быть четко структурированным: введение, цели и задачи, методы, результаты, заключение (выводы).

2. Статьи о результатах работ, проведенных в научных учреждениях, должны иметь разрешение на публикацию (сопроводительное письмо за подписью руководителя или выписку из заседания ученого совета, отдела или кафедры, акт экспертизы).

3. Статья в обязательном порядке должна иметь следующую структуру: индекс по универсальной десятичной классификации (УДК); инициалы и фамилии всех авторов, название статьи, полное название учреждений, где работают авторы, с указанием города, страны, аннотацию (150–250 слов), подрисуночные надписи, названия таблиц и ключевые слова (7–10) на русском и английском языках, адрес электронной почты контактного лица.

4. Аннотация (авторское резюме) должна кратко представлять результаты работы и быть информативной, содержательной. Приветствуется структура аннотации, повторяющая структуру статьи и включающая введение, цели и задачи, методы, результаты, заключение.

5. Формулы, рисунки, таблицы в статье нумеруются в соответствии с порядком их упоминания в тексте. Ссылки на рисунки и таблицы в тексте обязательны. Рисунки должны быть выполнены с хорошим разрешением в масштабе, позволяющем четко различать надписи и обозначения. Подрисуночные подписи с расшифровкой всех позиций, представленных на рисунке, набираются шрифтом гарнитуры основного текста размером 9 пт. Цветные иллюстрации печатаются только в том случае, когда это необходимо для понимания излагаемого материала.

6. Набор формул выполняется в формульном редакторе Microsoft Equation или Math Type. Прямым шрифтом набираются: греческие и русские буквы; математические символы (\sin , \lg , ∞); символы химических элементов (C, Cl, CHCl₃); цифры (римские и арабские); векторы; индексы (верхние и нижние), являющиеся сокращениями слов. Курсивом набираются латинские буквы, символы физических величин (в том числе и в индексе).

7. Сокращения в тексте статьи (за исключением единиц измерения) могут быть использованы только после упоминания полного термина. Единицы измерения физических величин следует приводить в Международной системе единиц (СИ).

8. Цитируемые в статье фамилии авторов теорем, теорий, законов и т. д. следует приводить в скобках на языке оригинала после русского написания. Например, теорема Эйлера (Euler).

9. Список использованной литературы оформляется в соответствии с требованиями Высшей аттестационной комиссии Республики Беларусь (ГОСТ 7.5–2008). Номер литературной ссылки в тексте дается порядковым номером в квадратных скобках. Ссылаться на неопубликованные работы не допускается.

10. Отдельно приводится список цитированных источников в *романском* (латинском) алфавите со следующей структурой: авторы (транслитерация), название статьи в транслитерированном варианте [перевод названия статьи на английский язык в квадратных скобках], название русскоязычного источника (транслитерация) [перевод названия источника на английский язык – парафраз (для журналов можно не делать)], выходные данные с обозначениями на английском языке.

11. Поступившие в редакцию статьи направляются на рецензирование специалистам. Основными критериями целесообразности публикации являются новизна и информативность статьи. Если по рекомендациям рецензента статья возвращается автору на доработку, то переработанная рукопись вновь рассматривается редколлегией. Статьи не по профилю журнала возвращаются авторам после заключения редколлегии.

12. Статьи, направляемые на доработку, должны быть возвращены в исправленном виде с ответами на все замечания.

13. Редакция журнала предоставляет возможность первоочередного опубликования статей, представленных лицами, которые осуществляют послевузовское обучение (аспирантура, докторантура, соискательство) в год завершения обучения.

14. Авторы несут ответственность за направление в редакцию статей, уже опубликованных ранее или принятых к публикации другими изданиями.

15. Редакция оставляет за собой право на редакционные изменения, не искажающие основное содержание статьи. Окончательное решение о публикации принимается редакционной коллегией.

Индексы

00827

для индивидуальных
подписчиков

008272

для предприятий и
организаций